



**République Algérienne Démocratique et Populaire**

**Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique**

**CDTA**

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

**MEMOIRE DE FIN D'ETUDE**

**Pour l'obtention**

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

**Option : Systèmes Informatiques et Réseaux**

**Thème :**

**Mise en Place d'une Application Logicielle pour  
la Simulation Géométrique des Fraiseuses 03-  
Axes et Génération du « G-Code »**

**Réalisé par :**

M<sup>elle</sup>. ABDALLAH Meriem

M<sup>elle</sup>. MESSAOUDI Amel

**Soutenu devant :**

Mr. BEY Mohamed	CDTA	Encadreur
Mr. BENDIFALLAH Hassène	CDTA	Encadreur
Mr. HAMMOUDA Mohamed	USDB	Promoteur
Mme. TOUBALINE Nesrine	USDB	Présidente
Mr. KAMECHE Abdallah Hicham	USDB	Examineur

2019/2020

## **Résumé :**

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

Dans ce projet, nous nous intéressons à la modélisation géométrique et à la simulation virtuelle des mouvements des fraiseuses numériques à 03-axes. Il s'agit de concevoir, de développer et d'intégrer à la plateforme logicielle, un module logiciel graphique et interactif pour la modélisation géométrique des différentes configurations des fraiseuses numériques à 03-axes, la simulation des mouvements et la génération du programme « G-Code ».

**Mots clés :** Pièce Complexe, Modèle STL, Simulation, Modélisation Géométrique, Collision, Fraiseuse 03-axes, G-Code.

## **Abstract:**

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

In this project, we are interested in the geometric modeling and virtual simulation of the movements of 03-axis CNC milling machines. This involves designing, developing and integrating into the software platform, a graphical and interactive software module for the geometric modeling of the various configurations of 03-axis CNC milling machines, the simulation of movements and the generation of the "G -Code".

**Keywords:** Complex Part, STL Model, Simulation, Geometric Modeling, Collision, 03-axis Milling Machine, G-Code.

## **ملخص:**

هذا العمل يندرج في إطار تطوير منصة برمجية لإنتاج الأسطح ذات الأشكال المعقدة التي بدأها فريق التصميم والتصنيع بمساعدة الكمبيوتر (CFAO) بقسم التصنيع والروبوتات (DPR) التابع لمركز تطوير التكنولوجيات المتطورة (CDTA).

في هذا المشروع، نحن مهتمون بالتمذجة الهندسية والمحاكاة الافتراضية لحركات أجزاء الآلات الرقمية ذات ثلاث محاور. يتضمن ذلك تصميم وتطوير ودمج في منصة البرامج، وحدة برمجية تفاعلية ورسومية للتمذجة الهندسية لمختلف الإعدادات المستخدمة في أجهزة التفريز ثلاثية المحاور، ومحاكاة الحركات وتوليد برنامج "G-Code".

**الكلمات المفتاحية:** القطع ذات الأشكال المعقدة، نموذج «STL»، المحاكاة، النمذجة الهندسية، آلات ذات 3 محاور (الآلات التفريز)، G-Code.

## **Remerciements**

*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 tenons à exprimer notre profonde gratitude à nos encadreurs Mr. Bey Mohammed et Mr. Bendifallah Hassène pour nous avoir aidé, guidé, et suivi pendant tout le long de notre travail.*

*Nous remercions également notre promoteur pour avoir accepté de diriger ce travail, pour son sens d'écoute et d'échange et aussi pour nous avoir corrigé et conseillé, il s'agit plus particulièrement de notre enseignant Mr. Hammouda Mohamed.*

*Ainsi que tout enseignant ayant enrichi nos connaissances et aidé pendant tout notre cursus à l'université Saad Dahleb-BLIDA et tous les étudiants qui nous ont aidé de près ou de loin.*

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

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

*Merci*

Je dédie ce modeste travail

À mes chers parents qui ne cessent de me donner avec amour le nécessaire pour que je puisse arriver à ce que je suis aujourd'hui, que Dieu vous protège et que la réussite soit toujours à ma portée pour que je puisse vous combler de bonheur.

A mon petit frère, Yacine, que les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour lui.

À mes frères et sœurs « Hannen, Lotfi, Asmaa »;

Que Dieu vous bénisse.

Et

« À tous ceux qui, par un mot, m'ont donné la force de continuer ».

ABDALLAH Meriem

Je dédie ce modeste travail :

*À mes chers parents*

Aucune dédicace ne saurait exprimer l'amour, la reconnaissance et le respect que j'ai pour vous.

Je tiens à vous remercier pour votre patience, soutien et encouragements tout le long de ces années.

J'implore Dieu, tout puissant, de vous accorder une bonne santé, une longue vie et beaucoup de bonheur.

*À ma sœur « Amina »*

Les mots sont insuffisants pour exprimer ma profonde estime, et ma gratitude de t'avoir comme sœur. Merci pour tes encouragements pendant toutes ces années

Je te dédie ce travail à toi et à ton mari que Dieu vous bénisse et vous garde la petite princesse « Sydra ».

Je vous souhaite tout le bonheur du monde.

*À ma sœur « Meriem »*

Aucun mot ne peut exprimer mon amour, et ma gratitude de t'avoir comme sœur. Merci pour ton soutien et tes conseils.

Je te souhaite une vie pleine de joie, de succès et de prospérité.

À tous les membres de ma famille, en particulier ma tante Houria et ses enfants « Lylia, Salem et Tarek »

À toute personne qui se reconnaîtra par ces mots, m'ayant aidé et encouragé de près ou de loin pendant mon cursus.

**Amel MESSAOUDI**

# Table des matières

Résumé .....	1
Remerciements.....	2
Table des matières .....	5
Liste des figures .....	8
<b>INTRODUCTION GENERALE.....</b>	<b>11</b>
<b>Chapitre I Étude bibliographique.....</b>	<b>13</b>
Introduction .....	14
I. Processus de production des pièces de formes complexes.....	14
I.1 Pièces de formes complexes .....	14
I.2 Processus d'usinage des pièces de formes complexes.....	14
II. Conception assistée par ordinateur .....	16
II.1 Modélisation géométrique .....	16
II.1.1 Modèle discret .....	16
II.1.2 Modèle continu .....	16
II.2 Formats d'échange de données .....	17
II.2.1 Format STEP .....	18
II.2.2 Format IGES.....	18
II.2.3 Format STL .....	18
III. Fabrication assistée par ordinateur .....	20
III.1. Fraisage.....	20
III.2. Machine-outil à commande numérique .....	21
III.2.1 G-Code .....	21
III.2.2 Techniques de modélisation des « MOCN ».....	22
III.2.3 Modélisation géométrique .....	22
III.2.4 Modélisation cinématique.....	23
III.3. Fraiseuses à commande numérique .....	23
III.3.1. Fraiseuses à commande numérique à 03-axes.....	24
III.3.2. Outils de fraisage .....	25
III.3.3. Configuration des fraiseuses numériques à 03-axes.....	26
III.4 Problèmes d'usinage .....	26
III.5 Simulation virtuelle de la géométrie de la machine.....	27
III.5.1 Simulation.....	27
III.5.2 Catégories de simulation .....	27

III.5.3	Nécessité de simulation .....	28
III.5.4	Processus de simulation virtuelle .....	28
Conclusion	.....	28
<b>Chapitre II</b>	<b>Étude Conceptuelle.....</b>	<b>29</b>
Introduction	.....	30
I.	Problématique et solution proposée .....	30
I.1	Problématique .....	30
I.2	Solution.....	31
I.2.1	Récupération du modèle STL de la pièce à usiner .....	31
I.2.2	Définition du trajet d'outils .....	32
I.2.3	Définition des différents organes de la machine-outil et configuration .....	33
I.2.4	Orientation de la machine .....	35
I.2.5	Positionnement de la pièce et du trajet d'outils sur la table .....	37
I.2.6	Simulation géométrique des mouvements de la machine .....	38
I.2.7	Génération du programme « G Code » et du fichier d'outils.....	40
II.	Modélisation UML .....	42
II.1	Diagramme de cas d'utilisation .....	43
II.1.1	Diagramme de cas d'utilisation général .....	43
II.1.2	Diagramme de cas d'utilisation « Importer le modèle de la machine et positionner la pièce ».....	43
II.1.3	Diagramme de cas d'utilisation « Simuler la géométrie de la machine » .....	44
II.1.4	Diagramme de cas d'utilisation « Générer le programme G-Code et le fichier d'outils ».....	45
II.2	Diagramme d'activité .....	45
II.2.1	Diagramme d'activité général .....	45
II.2.2	Diagramme d'activité de cas « Importer le modèle de la machine et positionner la pièce ».....	47
II.2.3	Diagramme d'activité de cas « Simuler la géométrie de la machine ».....	47
II.2.4	Diagramme de cas « Générer le programme G-Code et le fichier d'outils » .....	48
II.3	Diagramme de classes.....	48
II.3.1	Diagramme de classes général.....	48
II.3.2	Classes utilisées .....	50
Conclusion	.....	54
<b>Chapitre III</b>	<b>Implémentation et validation.....</b>	<b>55</b>
Introduction	.....	56
I.	Présentation des outils utilisés .....	56
I.1	Présentation du langage C++ .....	56

I.2	Présentation de l'OpenGL .....	57
I.3	Présentation de l'Embarcadero C++ Builder Seattle .....	57
II.	Présentation de l'environnement « CFAO ».....	57
III.	Présentation de l'application développée .....	58
III.1	Onglet « Pièce » .....	59
III.2	Onglet « Trajet » .....	60
III.3	Onglet « Machine ».....	61
III.4	Onglet « Simulation » .....	62
III.5	Onglet « G-Code » .....	63
IV.	Test et validation .....	65
	Conclusion.....	74
	<b>CONCLUSION GENERALE.....</b>	<b>75</b>
	<b>Références bibliographiques.....</b>	<b>77</b>



# Liste des figures

## Chapitre I : Etude bibliographique.

Figure 1. Processus d'élaboration des pièces de formes complexes [4].....	15
Figure 2. Processus d'usinage des pièces de formes complexes.....	16
Figure 3. Localisation d'un point sur une surface paramétrique. ....	17
Figure 4. Représentation d'une sphère. ....	18
Figure 5. Types de triangulation.....	19
Figure 6. Vecteur normal et orientation des sommets [12]. ....	19
Figure 7. Modes du fichier STL [10].....	20
Figure 8. Modes de fraisage. ....	21
Figure 9. Espace articulaire et espace des tâches. ....	22
Figure 10. Passage d'un solide au solide suivant. ....	23
Figure 11. Fraiseuses à 03-axes.....	24
Figure 12. Actions possibles et impossibles avec les fraiseuses 3-axes [17]. ....	24
Figure 13. Formes d'outils de coupe.....	25
Figure 14. Positionnement de l'outil. ....	26
Figure 15. Types d'interférences [12]. ....	27
Figure 16. Catégories de simulation [25]. ....	27

## Chapitre II : Etude conceptuelle

Figure 1. Organigramme général de la solution proposée.....	31
Figure 2. Limites et dimensions du brut.....	32
Figure 3. Paramètres géométriques des outils. ....	32
Figure 4. Trajet et syntaxe du fichier « trajet d'outils » d'une pièce.....	33
Figure 5. Point centre et point origine d'un organe.....	34
Figure 6. Organes mal orientés.....	35
Figure 7. Axes et angles de rotation. ....	36
Figure 8. Positionnement de la pièce au-dessus de la « Table ». ....	38
Figure 9. Point origine de la broche. ....	39
Figure 10. Géométrie de la machine.....	40
Figure 11. Origines proposés du programme « G-Code ».....	40
Figure 12. Calcul des positions d'outils pour le programme « G-Code ». ....	41
Figure 13. Optimisation du fichier « G-Code ». ....	42
Figure 14. Paramètres des outils utilisés. ....	42
Figure 15. Diagramme de cas d'utilisation général. ....	43
Figure 16. Diagramme de cas « Importer le modèle de la machine et positionner la pièce ».....	44
Figure 17. Diagramme de cas « Simuler la géométrie de la machine ». ....	44
Figure 18. Diagramme de cas « Générer le programme G-Code et le fichier d'outils ». ....	45
Figure 19. Diagramme d'activité général.....	46
Figure 20. Diagramme d'activité de cas « Importer le modèle de la machine et positionner la pièce ». ....	47

Figure 21. Diagramme d'activité de cas « Simuler la géométrie de la machine ».....	47
Figure 22. Diagramme d'activité de cas « Générer le programme G-Code et le fichier d'outils ».....	48
Figure 23. Diagramme de classes général. ....	49
Figure 24. Classe « STL ». ....	50
Figure 25. Classe « Brut ». ....	50
Figure 26. Classe « Triangle_STL ». ....	51
Figure 27. Classe « Normal_Triangle ». ....	51
Figure 28. Classe « Point_STL ». ....	51
Figure 29. Classe « Outil ». ....	52
Figure 30. Classe « Position ». ....	52
Figure 31. Classe « Trajet ». ....	52
Figure 32. Classe « Organe_Machine ». ....	53
Figure 33. Classe « Paramètres G-code ». ....	53
Figure 34. Classe « Machine ». ....	54

### **Chapitre III : Implémentation et validation**

Figure 1. Fenêtre principale de la plateforme. ....	58
Figure 2. Lancement du module logiciel développé.....	59
Figure 3. Onglets du module logiciel développé. ....	59
Figure 4. Onglet « Pièce ». ....	60
Figure 5. Onglet « Trajet ». ....	61
Figure 6. Onglet « Machine ». ....	62
Figure 7. Onglet « Simulation ». ....	63
Figure 8. Onglet « G-Code ». ....	64
Figure 9. Ouverture d'un fichier STL non valide. ....	64
Figure 10. Ouverture d'un fichier trajet non valide.....	64
Figure 11. Saisi d'un caractère non autorisé.....	65
Figure 12. Génération d'un fichier « G-Code » sans saisi des paramètres.....	65
Figure 13. Exemple de test. ....	66
Figure 14. Paramètres géométriques du modèle STL.....	66
Figure 15. Points de passage du trajet d'outils. ....	67
Figure 16. Machine complète. ....	67
Figure 17. Principaux organes de la machine.....	68
Figure 18. Point CL de la broche.....	68
Figure 19. Positionnement de la pièce sur la table. ....	68
Figure 20. Début de la première simulation. ....	69
Figure 21. Première simulation en cours. ....	69
Figure 22. Fin de la première simulation.....	69
Figure 23. Emplacement des organes mobiles. ....	69
Figure 24. Début et fin du premier fichier « G-Code ». ....	70
Figure 25. Premier fichier d'outils. ....	70
Figure 26. Réorientation de la machine.....	71

Figure 27. Machine complète réorientée. ....	71
Figure 28. Positionnement de la pièce sur la table. ....	72
Figure 29. Simulation de la deuxième machine. ....	72
Figure 30. Début et fin du deuxième fichier « G-Code ». ....	73
Figure 31. Deuxième fichier d'outils. ....	73
Figure 32. Comparaison entre les fichiers G-Code. ....	73
Figure 33. Comparaison entre les fichiers d'outils. ....	74

## Liste des tableaux

Tableau 1. Configurations des fraiseuses à 03-axes .....	25
--	----

# INTRODUCTION GENERALE

## Présentation du sujet :

L'outil informatique a connu une évolution très importante ces dernières années où il est introduit dans tous les domaines, en particulier dans les domaines de la conception et de la fabrication assistées par ordinateur (CFAO) tels que l'automobile, l'aéronautique...etc. Son utilisation a permis d'augmenter la productivité, de réduire les coûts et d'améliorer la qualité des pièces réalisées. Les pièces de formes géométriques très complexes sont de plus en plus utilisées dans notre vie quotidienne. En raison de leurs complexités géométriques et des problèmes d'accessibilité, ces pièces sont usinées, généralement, sur des fraiseuses numériques multi-axes de 03-axes à 05-axes. La forme finale de la pièce nécessite le passage par trois phases d'usinage : ébauchage, demi-finition et finition. Chaque phase nécessite la génération du trajet d'outils par la prise en compte de plusieurs paramètres (géométrie de la machine, outils, stratégies d'usinage, conditions de coupe, problèmes d'usinage...etc.). Le résultat de chaque phase, est un fichier texte appelé « CL-File » contenant toutes les positions de passage des outils avec les vitesses d'avance associées. L'obtention de ces pièces avec les caractéristiques prescrites dans le cahier des charges nécessite parfois l'utilisation successive de différents procédés de fabrication.

L'usinage effectif est réalisé après la transcription des trajectoires d'outils en un ensemble d'instructions appelées « G-Code » exécutables par la commande numérique de la machine-outil. Ces machines ont plusieurs configurations possibles et sont modélisées selon le modèle géométrique et le modèle cinématique.

Afin d'optimiser le temps d'usinage et de réduire le coût tout en améliorant la qualité des pièces, il est nécessaire d'effectuer une simulation avant de passer à l'usinage effectif.

## Situation du problème :

L'usinage consiste à réaliser des pièces de formes complexes à partir d'un brut par enlèvement de matière. La géométrie complexe des surfaces à usiner rend cette tâche très difficile ce qui nécessite une attention particulière lors de la fabrication.

L'usinage des pièces complexes est très coûteux. Donc, il est préférable d'éviter de rater les premières pièces et de s'affranchir de nombreuses reprises d'usinage. Pour rester compétitifs, les fabricants cherchent constamment à accroître la productivité, la qualité et à réduire les coûts et les délais de production.

Avant de passer à l'usinage effectif, il est indispensable de simuler virtuellement les organes mobiles de la machine quelle que soit sa configuration et quel que soit le modèle de la pièce et le trajet d'outil utilisés afin de vérifier l'absence des problèmes de collisions dans l'environnement d'usinage. Cette simulation permet de confirmer que pour chaque position du trajet que les organes de la machine sont en bonne position suivant les trois (03) axes de

déplacement. Si le trajet ne présente aucun problème, le code exécutable par la machine-outil « G-Code » est généré en fonction de la configuration de la machine sur laquelle l'usinage sera réalisé.

### Objectif du travail :

Ce travail, intitulé « *mise en place d'une application logicielle pour la simulation géométrique des fraiseuses 03-axes et génération du « G-Code »* », s'insère dans le cadre de développement d'une plateforme logicielle graphique et interactive sous Windows pour la production des pièces de formes complexes initié par l'équipe Conception et Fabrication Assistées par Ordinateur « CFAO » de la Division Productique et Robotique « DPR » du Centre de Développement des Technologies Avancées « CDTA ».


Dans ce projet nous nous intéressons à la **vérification de l'usinabilité** des pièces de formes complexes sur des fraiseuses numériques à 03-axes en considérant les deux aspects liés à la simulation virtuelle de la fraiseuse : les propriétés géométriques et cinématiques des machines-outils d'une part, et le trajet d'usinage d'autre part. Il s'agit donc de concevoir, de développer et d'intégrer à la plateforme logicielle de l'équipe « CFAO » du « CDTA » un module logiciel graphique et interactif permettant :

- La modélisation géométrique de toutes les configurations possibles des fraiseuses numériques à 03-axes.
- La simulation du trajet d'outil par le calcul des déplacements des organes mobiles de la fraiseuse.
- Le développement d'un post-processeur permettant la transcription du trajet d'outils « CL-File » en un programme « G-Code » exploitable par les machines-outils en fonction de leurs configurations.

### Structure du mémoire :

Le présent mémoire est composé des parties suivantes :

- Le premier chapitre est consacré à l'étude bibliographique du processus d'usinage des pièces de formes complexes sur des fraiseuses 03-axes.
- Le deuxième chapitre est réservé à l'étude conceptuelle de notre application.
- Le dernier chapitre détaille l'implémentation informatique et les tests réalisés pour valider l'approche proposée.



# Chapitre I

## Étude

# bibliographique

Introduction

- I. Processus de production des pièces de formes complexes
- II. Conception assistée par ordinateur
- III. Fabrication assistée par ordinateur

Conclusion

### Introduction

Avec l'évolution de l'industrie dans plusieurs domaines tels que l'énergétique, l'aéronautique, l'automobile...etc. Les pièces de base sont devenues de plus en plus de formes complexes. La production de ces pièces exige de plus en plus d'équipements spéciaux complexes et chers ainsi que le recours à des méthodes de réalisation plus adaptées ainsi qu'une haute précision. Cette évolution devait être aussi suivie par une évolution des machines-outils, des outils, des appareils et des méthodes de mesure et de contrôle.

La fabrication d'une pièce 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 pour concevoir un moule ou un prototype, reproduire des pièces d'origine ou en produire de nouvelles, c'est-à-dire la modélisation complète du produit et de son procédé dans un système de « CFAO » « Conception et Fabrication Assistées par Ordinateur ». La qualité de ces pièces est déterminée par les erreurs de forme et le degré de finition de la surface produite par le processus de fabrication. L'état des surfaces joue un rôle important dans la résistance à l'usure et à la fatigue des pièces usinées. 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 [1].

Ce chapitre est organisé en trois parties. La première partie, une présentation du processus de production des pièces de formes complexes. La deuxième partie est consacrée à 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 réservée à la fabrication assistée par ordinateur « FAO » en 03-axes et les moyens utilisés.

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

### I.1 Pièces de formes complexes

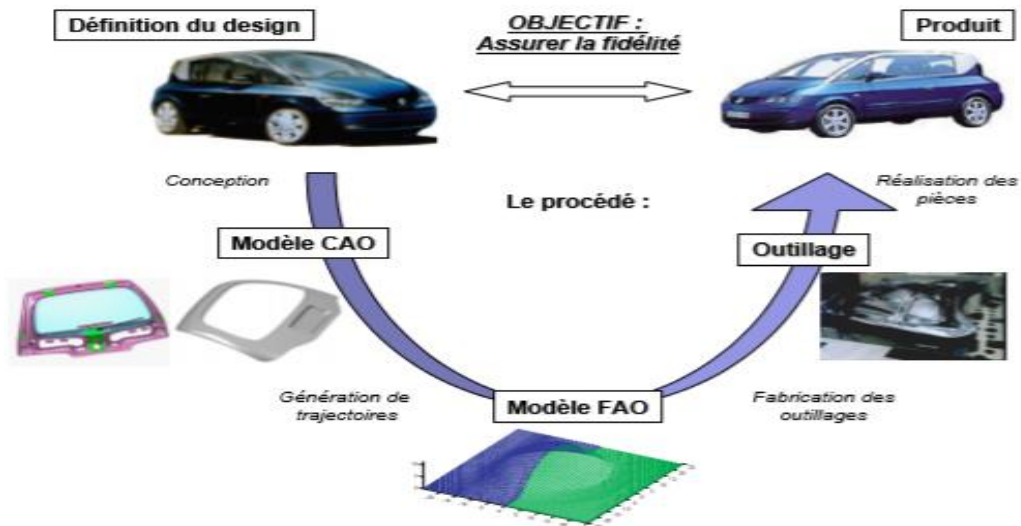
Une pièce de formes complexes est appelée aussi « pièce avec des surfaces de formes libres » ou « pièce avec des surfaces gauches ». Ces surfaces sont caractérisées par une topologie complexe composée d'un ensemble de surfaces concaves et/ou convexes. On parle de pièces complexes lorsqu'il y a une présence de topologies qui demandent de passer par de nombreuses trajectoires d'usinage, souvent avec risque de collisions ainsi que des zones difficilement accessibles [2]. La réalisation des pièces de formes complexes est soumise à plusieurs facteurs associés qui lui rendent cette réalisation difficile, longue et coûteuse. Parmi ces facteurs : la forme, les dimensions, les matériaux utilisés, le risque d'échauffement...etc.

### I.2 Processus d'usinage des pièces de formes complexes

Le processus de production des pièces de formes complexes est un ensemble d'étapes nécessaires permettant de réaliser les formes désirées à partir d'une pièce brute par

l'enlèvement de matière, tout en respectant les formes géométriques et les précisions exigées par le cahier des charges [3].

- **Processus d'élaboration des pièces de formes complexes** : il doit permettre de garantir la fidélité entre la pièce et les spécifications fonctionnelles exprimant l'idée initiale du concepteur (Figure 1) [4].



**Figure 1.** Processus d'élaboration des pièces de formes complexes [4].

- **Processus d'usinage** : avant l'usinage réel, nous devons passer par l'usinage virtuel des pièces composées de surfaces gauches. Cette étape permet d'éviter les risques d'accidents, de détecter et de corriger les erreurs et de diminuer les coûts. Le processus d'usinage débute par un cahier des charges et se termine par l'obtention des pièces selon les exigences du client. Le processus d'usinage passe par trois parties (Figure 2).

**Étape 1** : identification de l'objet à réaliser.

**Étape 2** : conception et fabrication assistées par ordinateur « CFAO ». La « CFAO » permet de concevoir et de réaliser des pièces pour les assembler pour en faire un système. L'idée générale de ce procédé est d'utiliser les capacités de l'ordinateur afin de dessiner la pièce en 02 dimensions (2D) ou, mieux encore, en 03 dimensions (3D) pour la fabriquer [5]. Cette étape comprend deux sous-étapes : la conception « CAO » et la fabrication « FAO ».

**Étape 3** : l'usinage réel sur machine-outil en utilisant le procédé de fraisage. La forme finale de la pièce est obtenue en trois étapes :

- **Ébauchage** : consiste à enlever le maximum de matière.
- **Demi-finition** : consiste à se rapprocher de la forme souhaitée en corrigeant les défauts géométriques résultants de l'opération d'ébauchage.
- **Finition** : consiste à obtenir la forme finale conforme aux spécifications techniques et géométriques.



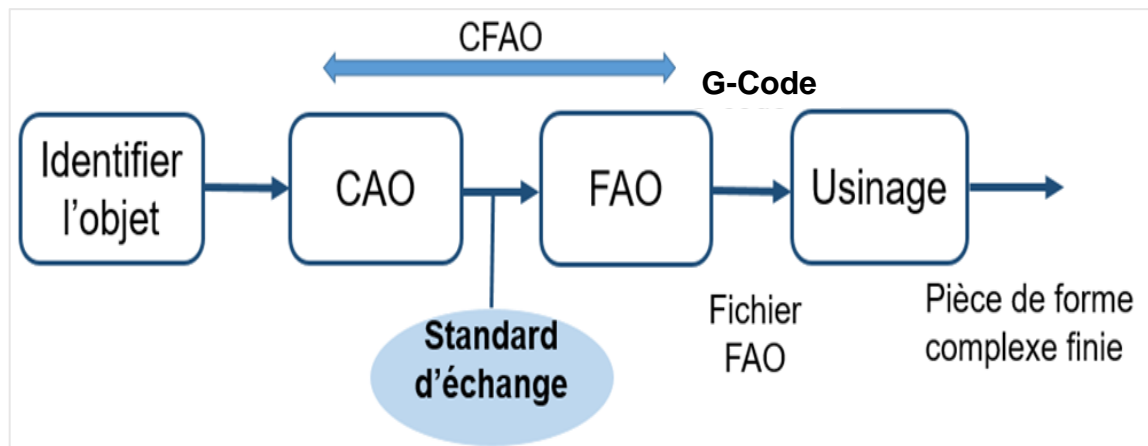


Figure 2. Processus d'usinage des pièces de formes complexes.

## II. Conception assistée par ordinateur

Les environnements « CAO » fournissent des outils puissants pour la programmation **graphique** et la manipulation d'entités **géométriques** complexes. Pour la représentation des surfaces en « CAO », le modèle géométrique peut être utilisé. Il est ensuite aisé de transmettre les informations à la « FAO » « Fabrication Assistée par Ordinateur » en utilisant des standards d'échange de données.

### II.1 Modélisation géométrique

C'est la représentation virtuelle (numérique) d'un objet réel. Les modèles géométriques représentent la surface frontière (volume) d'un objet ou d'une forme. Ils doivent être le plus proche possible de la réalité qu'ils cherchent à reproduire. Ainsi, un objet peut avoir une représentation plus ou moins complexe ou plus ou moins simplifiée selon l'application pour laquelle il est destiné [6]. Dans le cadre de notre étude, nous nous penchons plus particulièrement sur la modélisation géométrique appliquée au domaine de la « CAO ». La modélisation géométrique est composée de plusieurs modèles (discret et continu).

#### II.1.1 Modèle discret

La modélisation est réalisée en utilisant un ensemble d'entités géométriques simples (point, carré, triangle...etc.) pour représenter un objet donné. La modélisation est plus rapide et sa fidélité dépend des précisions d'approximation exigées. Ce modèle comporte le modèle polyédrique et le modèle de subdivision.

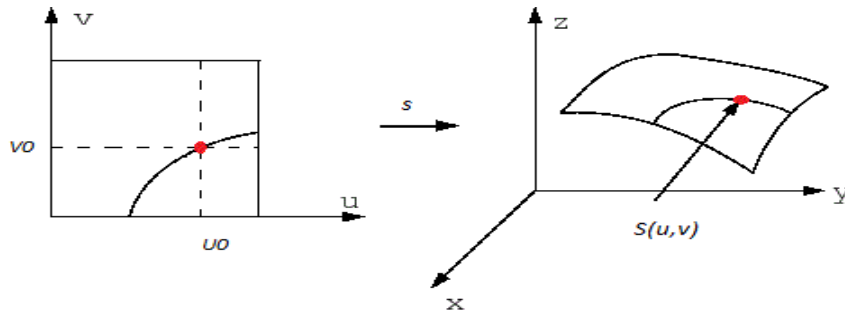
#### II.1.2 Modèle continu

La modélisation est réalisée en se basant sur des équations mathématiques, utilisée surtout en « CAO ». Ce modèle comporte le modèle paramétrique (courbes et des surfaces). Nous nous intéressons dans notre recherche aux surfaces paramétriques parce qu'elles permettent de modéliser les objets 3D continus et en raison de ses propriétés.

➤ **Définition d'une surface paramétrique :** on appelle surface paramétrique  $S$  la représentation définie par les équations paramétriques suivantes (Figure 3) :

$$S = \begin{cases} x = f(u, v) \\ y = g(u, v) \\ z = h(u, v) \end{cases} \quad \text{Avec } (u, v) \in [0,1]^2 \quad (1)$$

$f$ ,  $g$  et  $h$  sont des fonctions réelles et  $u$  et  $v$  ses paramètres. Les fonctions paramétriques permettent de se déplacer sur la surface en faisant varier les paramètres  $u$  et  $v$ .



**Figure 3.** Localisation d'un point sur une surface paramétrique.

➤ **Propriétés de la surface paramétrique :** les surfaces paramétriques sont caractérisées par les propriétés suivantes :

- Point de la surface : le point  $M_0$  de paramètres  $u_0$  et  $v_0$  a pour coordonnées :

$$M_0 \begin{cases} x_0 = f(u_0, v_0) \\ y_0 = g(u_0, v_0) \\ z_0 = h(u_0, v_0) \end{cases} \quad (2)$$

- Vecteurs tangents : les vecteurs tangents  $\vec{r}_0$  et  $\vec{s}_0$  suivant les directions  $u$ ,  $v$  respectivement au point  $M_0$  sont donnés par :

$$\vec{r}_0 = \begin{pmatrix} \frac{\partial f}{\partial u} \\ \frac{\partial g}{\partial u} \\ \frac{\partial h}{\partial u} \end{pmatrix} \quad \text{et} \quad \vec{s}_0 = \begin{pmatrix} \frac{\partial f}{\partial v} \\ \frac{\partial g}{\partial v} \\ \frac{\partial h}{\partial v} \end{pmatrix} \quad (3)$$

- Vecteur normal : le vecteur normal au point  $M_0$  est obtenu par le produit vectoriel des vecteurs tangents. Il permet d'orienter la surface.

$$\vec{n} = \frac{\vec{r}_0 \wedge \vec{s}_0}{\|\vec{r}_0 \wedge \vec{s}_0\|} \quad (4)$$

## II.2 Formats d'échange de données

Les formats d'échange de données aussi appelés standards d'échange de données, sont des passerelles qui assurent la fiabilité des informations contenues dans les données lors du transfert de ces dernières. Il existe plusieurs formats d'échange (STEP, IGES, STL, ...etc.).

### II.2.1 Format STEP

Abréviation de « *Standard for Exchange of Product model data* ». Son objectif principal est de représenter sans aucune ambiguïté les données liées à l'ingénierie et aux produits, dans un format neutre, interprétable par tout système informatique, sans perdre l'intégrité des données durant tout le cycle de vie d'un produit [7].

### II.2.2 Format IGES

Abréviation de « *Initial Graphical Exchange System* ». Le premier standard d'échange de données « CAO », son principal objectif est l'échange des dessins techniques. L'échange entre deux systèmes est réalisé en deux étapes [7] :

- Ecriture : un pré-traitement réalisé sur le système émetteur permettant de générer un fichier IGES.
- Lecture : un post-traitement effectué sur le système récepteur qui consiste à traduire le fichier IGES selon le langage propre à ce système.

### II.2.3 Format STL

STL est l'abréviation de « *Standard Tessellation Language* ». C'est un format d'échange de données supporté par de nombreux systèmes « CAO ». Il a été développé par la société 3D Systèmes. 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 3 dimensions par un ensemble de triangles (Figure 4) [8]. Il est utilisé afin de stocker des ensembles de mailles incolores car ce type de fichier n'est pas adapté pour les impressions en couleur [9]. C'est le format de fichier le plus utilisé pour l'impression 3D.



**Figure 4.** Représentation d'une sphère.

**1. Règles spéciales pour le format STL :** le fichier STL représente l'objet par son enveloppe extérieure. Cette représentation doit être sous forme de faces triangulaires, où chaque triangle doit partager deux sommets avec le triangle adjacent, autrement dit deux arêtes avec les triangles adjacents, ainsi, une arête est commune à deux triangles [10].

- ❖ **Règle des sommets :** chaque triangle doit partager deux sommets avec ses triangles voisins (Figure 5.b). Cette règle doit être impérativement respectée lors de l'approximation des surfaces d'un objet 3D [8].



Figure 5. Types de triangulation.

❖ **Règle d'orientation** : chaque triangle est défini par les composantes X, Y et Z de son vecteur normal  $\vec{N}$  orienté vers l'extérieur et de ses trois sommets orientés dans le sens trigonométrique (Figure 6) [8].

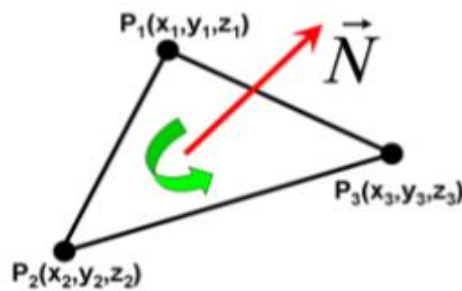


Figure 6. Vecteur normal et orientation des sommets [11].

**2. 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 des triangles sont stockées mais de deux manières différentes :

➤ **Fichier STL ASCII** : le fichier commence toujours par la ligne « solid » et se termine par « endsolid » (Figure 7.a). Chaque triangle commence par « facet » et se termine par « endfacet ». Le vecteur normal est inclus dans la ligne « facet » et il est identifié par le mot clé « normal ». Les trois sommets du triangle sont délimités par « outer loop » et « endloop ». Chaque sommet est décrit avec un vertex de coordonnées X, Y et Z [7].

Si la triangulation comporte de nombreux petits triangles, le fichier ASCII STL devient volumineux. C'est pourquoi il existe une version binaire plus compacte.

➤ **Fichier STL binaire** : le fichier STL binaire commence par une en-tête de 80 caractères, puis le nombre total de triangles est indiqué à l'aide d'un entier non signé de 4 octets. Les informations sur les triangles suivent, chaque triangle est représenté par douze nombres à virgule flottante de 32 bits. Tout comme le fichier ASCII STL, 3 nombres correspondent aux coordonnées cartésiennes 3D de la normale du triangle et les 9 autres chiffres correspondent aux coordonnées des sommets (trois chacun) (Figure 7.b). Après chaque triangle, il y a une séquence de 2 octets appelée « nombre d'octets d'attribut ». Dans la plupart des cas, il est mis à zéro et agit comme un espaceur entre deux triangles [9].

a. Fichier STL ASCII.

b. Fichier STL binaire.

Figure 7. Modes du fichier STL [9].

### 3. Avantages et inconvénients du modèle STL

#### ➤ Avantages

- Fichier plus simple puisqu'il utilise le triangle pour la description de l'objet.
- Il est universel et pris en charge par presque tous les logiciels « CAO ».
- Ne comportant que le strict minimum d'informations sur le modèle 3D.
- Facile à trancher.

#### ➤ Inconvénients

- Le fichier généré est souvent très volumineux.
- Redondance de l'information.
- Impossible d'inclure des métadonnées telles que la paternité.
- Le support des couleurs n'est pas standard et n'est souvent pas possible.

## III. Fabrication assistée par ordinateur

Appelé aussi préparation à la fabrication. Elle comprend des outils informatiques pour définir les trajectoires et les outils de coupe (matière et dimensions du brut, types de fraises, etc.) à partir du fichier « CAO » et finir par la génération d'un fichier « G-Code » de pilotage pour décrire les trajectoires et les mouvements d'outils pour réaliser une pièce précise.

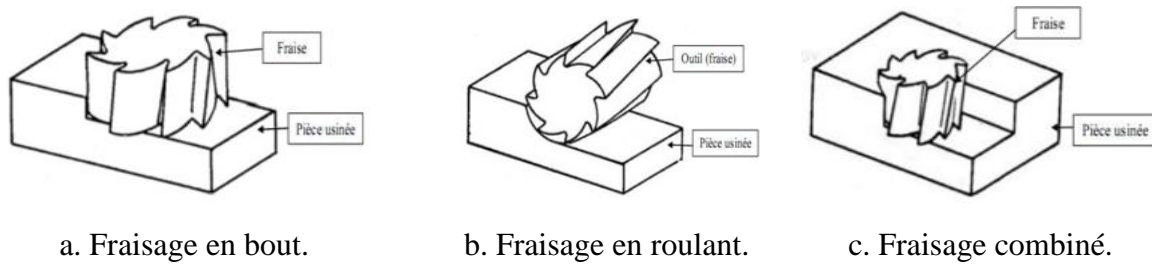
### III.1. Fraisage

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 et avance rectiligne de la pièce à usiner. Il est possible de distinguer trois modes de fraisage :

➤ **Fraisage en bout** : le fraisage en bout est le plus répandu car il permet de réaliser des surfaces complexes [12]. L'axe de la fraise est placé perpendiculairement à la surface à usiner (Figure 8.a). La fraise coupe avec son diamètre, mais aussi avec sa partie frontale.

➤ **Fraisage en roulant** : l'axe de la fraise est placé parallèlement à la surface à usiner (Figure 8.b). La fraise coupe avec son diamètre [13].

➤ **Fraisage combiné** : la fraise coupe avec son diamètre et sa partie frontale (Figure 8.c) [15].



**Figure 8.** Modes de fraisage.

### III.2. Machine-outil à commande numérique

Une machine-outil à commande numérique « MOCN » est une machine programmable équipée d'une commande numérique par ordinateur « CNC ». Une « MOCN » assure la réalisation automatisée des pièces où les mouvements nécessaires sont décrits dans un programme. Ces machines sont composées essentiellement de deux (02) parties :

- **Partie opérative** : composée d'organes qui agissent directement sur le produit final.
- **Partie commande** : contrôle la machine en envoyant des consignes et récupère des informations sur le déroulement des opérations.

Pour le pilotage des machines-outils à commande numérique le langage « G-Code » est utilisé actuellement et ses principes sont regroupés dans la norme ISO 6983 [19].

#### III.2.1 G-Code

G-Code ou « code géométrique », c'est le langage utilisé pour contrôler une MOCN. Il sert à programmer les mouvements que la machine va effectuer. La suite d'instructions que la machine va effectuer « programme d'usinage » est contenue dans un fichier texte. Il se compose d'un certain nombre de « commandes » spécifiques, indiquant à la machine quel type de mouvements elle doit exécuter (droite, arc de cercle, etc.), et d'indications de coordonnées sur les axes X, Y et Z (cas de 3-axes).

Un programme G-Code d'une pièce à usiner est composé de blocs et de mots soumis à des règles de structure, syntaxe ou format.

### III.2.2 Techniques de modélisation des « MOCN »

#### 1. Pourquoi modéliser une MOCN ?

Par définition, le pilotage d'une « MOCN » est réalisé par un ordinateur qui établit une stratégie de commande des axes (sortie) en fonction des déplacements mesurés de la machine (entrée). Pour représenter la machine dans l'espace, la modélisation dynamique de la partie opérative est nécessaire au calcul de l'asservissement. De même, la modélisation géométrique de la machine est nécessaire à sa commande par l'opérateur [20].

#### 2. Principe de la cellule élémentaire d'usinage (CEU)

On considère que la situation relative de l'outil par rapport à la pièce est assurée par un « assemblage » de corps solides, autorisant un mouvement de rotation ou de translation. On distingue deux espaces de description : l'espace des tâches et l'espace articulaire. La géométrie des surfaces à usiner et les dimensions du porte pièce sont décrites dans l'espace des tâches. Les mouvements relatifs des organes de la machine et leurs dimensions intrinsèques sont représentés dans l'espace articulaire [20].

### III.2.3 Modélisation géométrique

Une fois les repères géométriques définis, on détermine le modèle géométrique qui permet de passer d'une configuration de l'espace articulaire vers une position (de l'outil par rapport à la pièce) de l'espace des tâches (Figure 9). Pour cela, on crée des modèles géométriques associés à chaque élément. Pour la chaîne cinématique, on utilise les modèles classiques de la robotique pour décrire les positions respectives des solides. Pour l'outil, on définit un modèle caractérisant la partie active de l'outil. À la fin, on modélise les liaisons complètes par des matrices de changement de repère ou des torseurs de petits déplacements. Deux niveaux de finesse sont possibles pour modéliser la machine-outil.

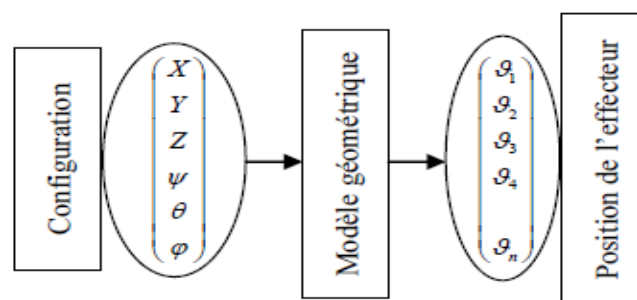


Figure 9. Espace articulaire et espace des tâches.

➤ **Modélisation globale de la machine :** la modélisation complète de la machine a pour but d'étudier précisément la phase d'usinage et les défauts géométriques associés. La machine est modélisée comme un changement de repère qui permet de passer d'un repère associé à l'outil à un repère associé à la pièce. On utilise pour cela les coordonnées homogènes (possibilité d'appliquer les mêmes opérateurs aux points et aux vecteurs) et les matrices de changement de repère (pour exprimer les rotations et les translations).

➤ **Modélisation par axe de la machine** : dans ce cas, on cherche à obtenir un modèle qui permet le passage du repère associé à un solide au repère associé au solide suivant. Ce passage est paramétré par deux distances «  $d_i$  » et «  $a_i$  » ainsi que deux angles «  $\alpha_i$  » et «  $\theta_i$  » entre les deux solides (Figure 10) [20].

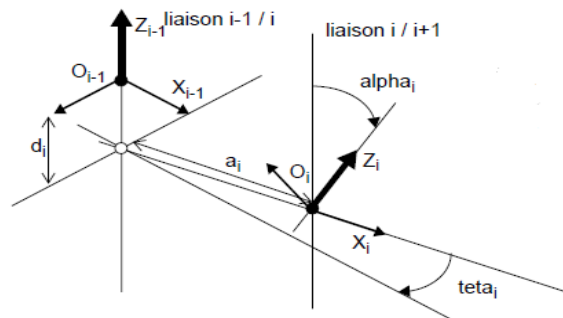


Figure 10. Passage d'un solide au solide suivant.

Les deux niveaux de finesse peuvent être utilisés en modélisation directe ou inverse :

- **Modélisation géométrique directe (MGD)** : pour des consignes données aux axes, on trouve la position de l'extrémité outil dans le repère lié à la pièce.
- **Modélisation géométrique inverse (MGI)** : pour un jeu de coordonnées de l'extrémité de l'outil exprimées dans le repère associé à la pièce, on trouve les consignes à programmer sur les axes. Ce modèle permet la transformation des coordonnées de l'espace des tâches à l'espace articulaire.

### III.2.4 Modélisation cinématique

Consiste à traiter les deux organes mobiles de la machine-outil : la table et la broche.

- **Modélisation cinématique directe (MCD)** : le modèle cinématique direct permet de passer de la vitesse articulaire de la machine à la vitesse dans le domaine cartésien.
- **Modélisation cinématique inverse (MCI)** : l'objectif du modèle cinématique inverse des machines « CNC » est le calcul des vitesses des organes mobiles à partir de la vitesse de l'outil [21].

### III.3. Fraiseuses à commande numérique

Une fraiseuse est une machine-outil utilisée pour usiner des pièces mécaniques. Elle est équipée d'un outil coupant appelé une fraise. La fraiseuse est capable d'enlever de la matière d'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. Un axe est le degré de liberté de déplacement de la machine.



### III.3.1. Fraiseuses à commande numérique à 03-axes

Une fraiseuse à 03-axes comporte une broche soit verticale (Figure 11.a) soit horizontale (Figure 11.b) ou universelle, et c'est à la table d'être positionnée en fonction du but recherché sur la pièce [15]. 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.
- L'axe Y forme le trièdre direct avec les axes X et Z.



a. Fraiseuse verticale.



b. Fraiseuse horizontale.

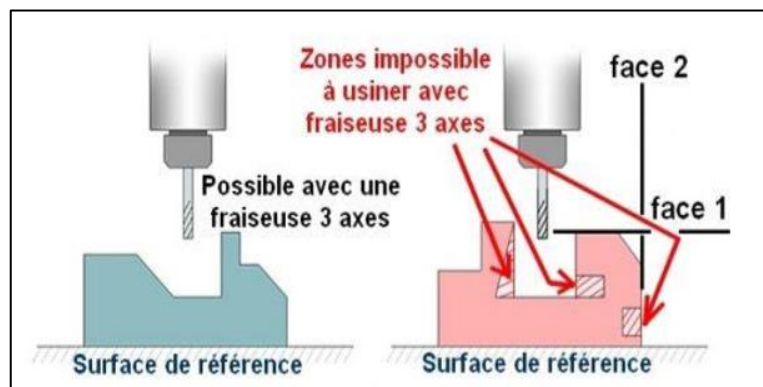
**Figure 11.** Fraiseuses à 03-axes.

Les fraiseuses à commande numérique à 03-axes permettent :

- Des découpes sur des surfaces 2D ou 3D.
- Des gravures sur des surfaces 2D ou 3D.
- De créer des formes 3D sans dépouilles négatives.

Cependant, les actions suivantes sont impossibles (Figure 12) :

- Des usinages nécessitant un axe d'outil perpendiculaire à plusieurs plans.
- Des usinages de surfaces avec dépouilles négatives.



**Figure 12.** Actions possibles et impossibles avec les fraiseuses 3-axes [16].

Avec l'expression des nouveaux besoins (Figure 12), de nouvelles technologies d'usinage ont donc été développées telles que les fraiseuses à 04-axes et à 05-axes.

### Avantages des fraiseuses à commande numérique

- Réalisation des pièces de formes complexes en temps réduit.
- Fonctionnement sans aucune assistance pendant l'usinage.
- Possibilité de changement du programme d'usinage.

### III.3.2. Outils de fraisage

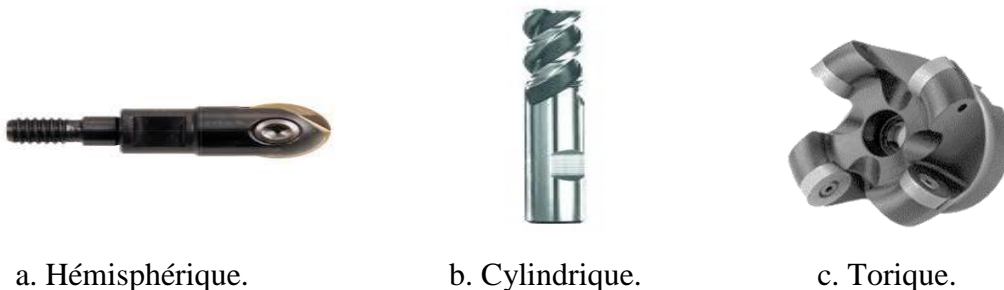
L'outil de coupe « fraise » est un organe monté sur la tête de la machine-outil. Il sert à enlever la matière (copeaux) de la pièce à usiner. Il est défini par des critères géométriques (forme et dimensions) et technologiques (conditions de coupe et modes d'attachement) [17].

**1. Parties d'un outil de coupe :** les différentes parties de l'outil sont :

- **Corps :** il assure la liaison de l'outil avec le porte-outil.
- **Queue :** elle permet la liaison de l'outil avec le porte-outil.
- **Partie active :** c'est la partie qui agit sur la pièce pour provoquer un enlèvement de métal. Elle est caractérisée par sa forme et son matériau.

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

- **Fraise hémisphérique :** cet outil est le plus utilisé vu sa simplicité et sa géométrie (Figure 13.a).
- **Fraise cylindrique :** avec leurs faces planes, elles conviennent aux travaux d'ébauche et de demi-finition (Figure 13.b).
- **Fraise torique :** les fraises toriques constituent un bon compromis entre les fraises cylindriques et les fraises hémisphériques (Figure 13.c).



**Figure 13.** Formes d'outils de coupe.

**3. Positionnement de l'outil :** Le positionnement de l'outil sur la pièce s'effectue en deux étapes :

- La définition du point de contact au niveau de l'outil et de la pièce.
- L'orientation de l'axe de l'outil.

Le positionnement de l'outil 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 (Figure 14).

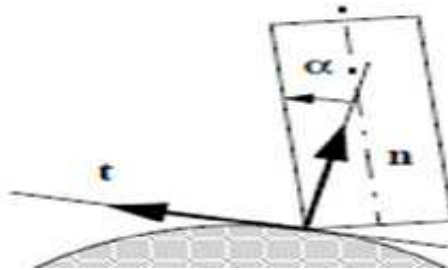


Figure 14. Positionnement de l'outil.

### III.3.3. Configuration des fraiseuses numériques à 03-axes

Ce type de fraiseuses se caractérise par trois **degrés de liberté de translation** (directions autorisés), et donc huit (08) configurations possibles. Une configuration de fraiseuse est la définition des axes de translation pour la table et les axes de translation pour la broche. Ces configurations peuvent être divisées en trois (03) catégories :

- **Déplacement de la table** : dans ce cas, la broche est immobile et la table se déplace suivant les trois (03) axes.
- **Déplacement de la broche** : dans ce cas la table est immobile et seule la broche se déplace suivant les trois (03) axes.
- **Déplacement des deux organes** : combinaison des deux déplacements précédents.

### III.4 Problèmes d'usinage

Lors de la phase d'usinage des pièces de formes complexes, plusieurs problèmes peuvent se poser, au niveau du positionnement de l'outil ou de son déplacement, les collisions représentent le problème majeur en usinage 03-axes. Ces problèmes peuvent être une source de génération de défauts géométriques importants. Ils sont classés comme suit :

- **Interférences locales** : elles traduisent un enlèvement de matière excessif par l'outil sur la surface à usiner (Figure 15.a).
- **Interférences vers l'arrière** : elles sont des pénétrations intempestives de l'arrière de l'outil dans la surface à usiner (Figure 15.b).
- **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...} (Figure 15.c) [22].

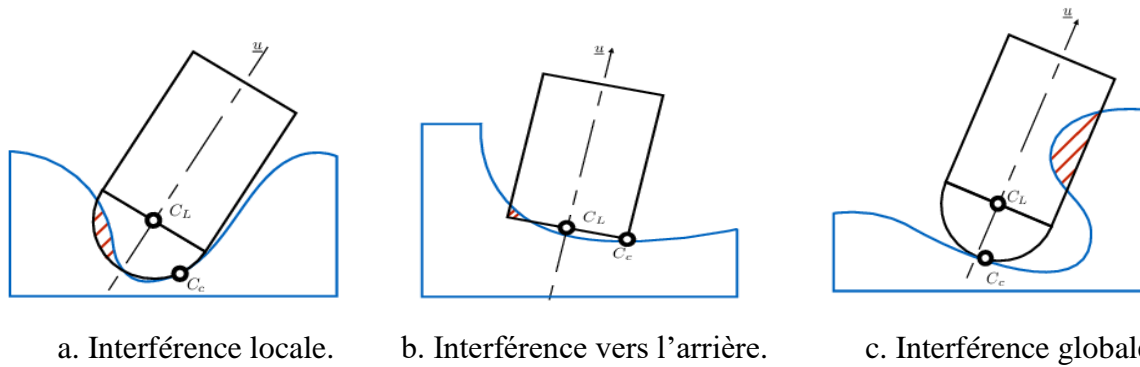


Figure 15. Types d'interférences [11].

### III.5 Simulation virtuelle de la géométrie de la machine

Afin d'éviter les problèmes de l'usinage mentionnés dans le point précédent, il est indispensable de simuler virtuellement les déplacements des organes mobiles de la machine.

#### III.5.1 Simulation

Le mot « simulation » veut dire faire apparaître comme réel. La simulation d'usinage est une **représentation virtuelle** des actions de fraisage tel que le mouvement de l'outil, l'enlèvement de la matière...etc. C'est une opération qui s'effectue en FAO avant de passer à l'usinage réel sur machine-outil.

#### III.5.2 Catégories de simulation

Deux catégories de simulation de l'usinage sont distinguées : simulation géométrique et simulation physique (Figure 16).

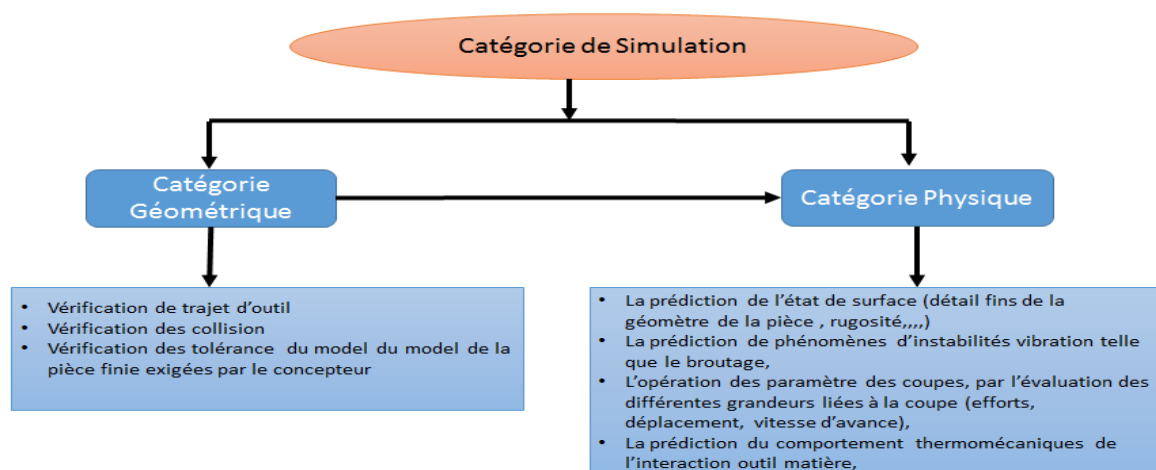


Figure 16. Catégories de simulation [23].

- **Simulation géométrique** : elle est utilisée pour vérifier graphiquement les interférences et les collisions, le respect des tolérances exigées dans le cahier des charges par le constructeur. En outre, elle peut fournir des informations géométriques telles que l'angle d'attaque de l'outil à la simulation physique.

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

### III.5.3 Nécessité de simulation

Avant de passer à l'usinage effectif sur machine, il est indispensable de simuler virtuellement les déplacements des organes mobiles de la machine afin de réaliser les actions suivantes :

- Vérifier l'usinabilité des surfaces complexes.
- Vérifier un programme avant son exécution, en remédiant aux erreurs et en palliant aux risques de collision.
- Evaluer avec précision les temps d'exécution en vue d'élaborer des devis précis.
- Optimiser le choix entre plusieurs procédures d'exécution et ainsi parvenir aux choix économiques garantissant la qualité exigée.

Pour qu'une simulation fournisse les résultats escomptés, il est nécessaire de connaître les principes directeurs mis en œuvre et les règles qui ont conduit aux objectifs proposés, c'est-à-dire une gamme de production opérationnelle [18].

### III.5.4 Processus de simulation virtuelle

La simulation de l'usinage des pièces de formes complexes s'effectue en deux étapes importantes [18] :

- Faire entrer le brut à usiner soit en donnant les dimensions du brut ou bien en utilisant les dimensions minimales du brut.
- Faire entrer la trajectoire de l'outil.

## Conclusion

Dans ce chapitre et en première partie, nous avons introduit le processus de fabrication des pièces de formes libres et nous avons présenté l'outil « CFAO ». En deuxième partie, nous avons exposé les différentes méthodes de modélisation des surfaces gauches et les formats d'échange de données en mettant l'accent sur le format « STL ». Par la suite, nous avons présenté le procédé de fraisage, les machines-outils à commande numérique, les fraiseuses numériques à 03-axes, les problèmes d'usinage et à la fin le processus de simulation de la géométrie de la machine.

Le prochain chapitre fera l'objet de la présentation de la conception de notre application

# Chapitre II

## Étude Conceptuelle

Introduction

I. Solution proposée

II. Modélisation UML

Conclusion

### Introduction

Après avoir présenté dans le chapitre précédent un état de l'art et des généralités indispensables pour la compréhension de notre travail. Le présent chapitre sera consacré à cerner les étapes entreprises pour simuler virtuellement les déplacements des organes mobiles des fraiseuses numériques à 03-axes. Ces étapes sont, en premier lieu, présentées dans l'architecture générale de la solution proposée. Par la suite, nous allons exploiter le formalisme UML (Unified Modeling Language) pour présenter en détail notre conception.

## I. Problématique et solution proposée

### I.1 Problématique

Afin d'éviter de rater les premières pièces et de perdre le temps et d'augmenter les coûts de production, il est indispensable de simuler virtuellement les déplacements des organes mobiles de la machine d'usinage. Pour atteindre cet objectif, il est impératif de récupérer l'ensemble des parties introduites dans la simulation virtuelle des mouvements {pièce, trajet d'outils, organes de la machine}, pour tous les outils, toutes les opérations d'usinage, toutes les stratégies de balayage des outils et pour toutes les configurations possibles des fraiseuses numériques à 03-axes. L'axe de la broche doit être vertical et l'enveloppe de la table de la machine doit être parallèle aux axes de déplacement X et Y. Avant la simulation, la pièce et le trajet doivent être positionnés au centre de la face supérieure de la table.

Pour cela, les questions suivantes doivent être posées :

- Comment unifier la récupération des modèles de la pièce et des différents organes de la fraiseuse ?
- Comment récupérer le trajet d'outils ?
- Comment orienter correctement les différents organes de la machine ?
- Comment positionner la pièce et le trajet d'outils par rapport à la table ?
- Comment faire pour considérer toutes les configurations possibles des fraiseuses numériques à 03-axes ?
- Comment déplacer les organes mobiles de la machine pour chaque position du trajet d'outils ?
- Comment générer un programme « G-Code » optimisé en fonction de la configuration de la fraiseuse ?
- Quelles sont les formes et les dimensions des outils utilisés ?
- Quels sont les emplacements des outils dans le magasin d'outils de la fraiseuse ?

## I.2 Solution

Pour résoudre les problèmes mentionnés précédemment, nous avons proposé une solution comprenant les étapes suivies, présentées dans l'organigramme général donné par la Figure 1.

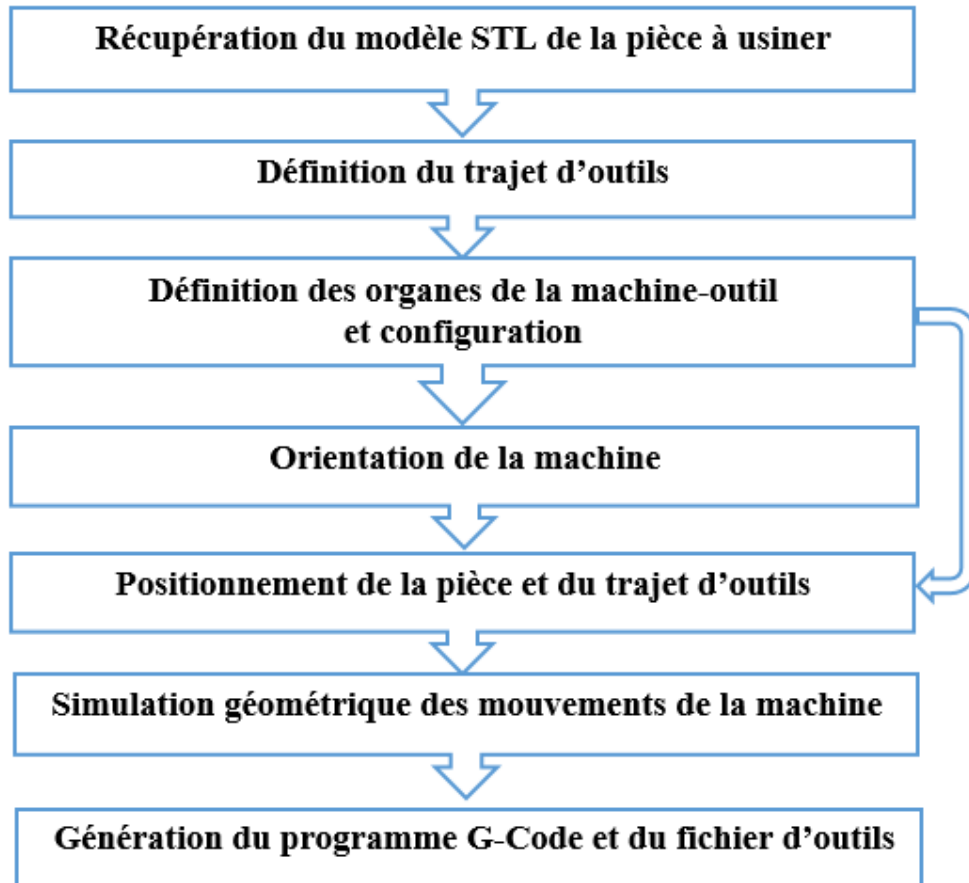


Figure 1. Organigramme général de la solution proposée.

### I.2.1 Récupération du modèle STL de la pièce à usiner

Le fichier STL décrivant la pièce à usiner doit être analysé. Cette analyse consiste à vérifier son extension et sa syntaxe. La lecture du fichier STL consiste à récupérer les informations des sommets et des triangles. Par la suite, les dimensions minimales du brut de la pièce doivent être calculées. La création du brut a pour but d'englober tous les points du modèle STL dans une enveloppe parallélépipédique des faces parallèles aux axes X, Y et Z. Il s'agit de déterminer les coordonnées  $X_{\min}$ ,  $X_{\max}$ ,  $Y_{\min}$ ,  $Y_{\max}$ ,  $Z_{\min}$  et  $Z_{\max}$  et par la suite ses dimensions minimales (Longueur, Largeur et Hauteur) (Figure 2) [11].

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



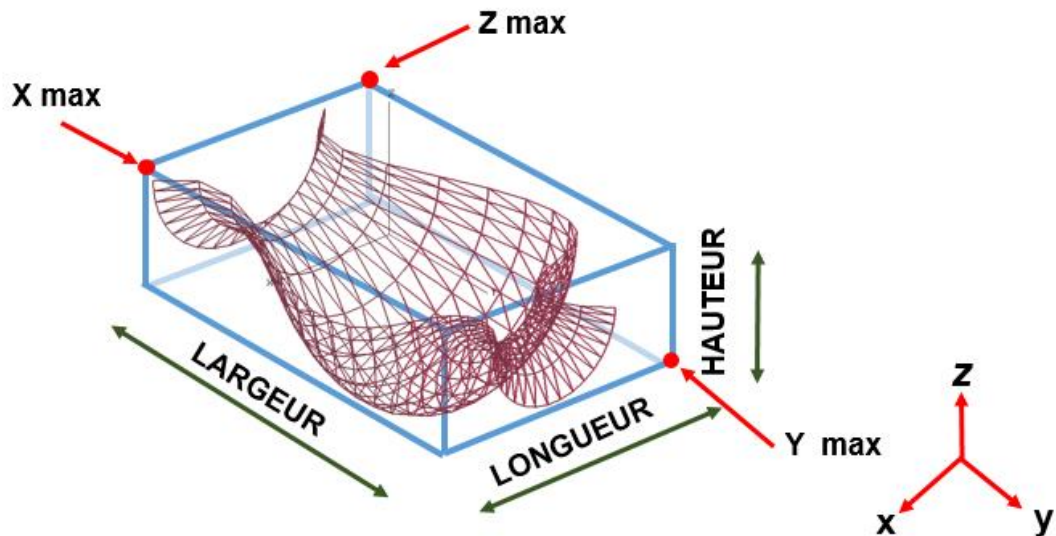


Figure 2. Limites et dimensions du brut.

### I.2.2 Définition du trajet d'outils

Pour la solution proposée, un trajet d'outils est déterminé par la définition de :

- **Forme et dimensions d'outils** : plusieurs formes d'outils sont utilisées dans l'usinage à 03-axes parmi eux les trois formes mentionnées dans le chapitre précédent (l'outil hémisphérique, l'outil torique, l'outil cylindrique). Ces outils sont caractérisés par une longueur  $L$  et un rayon  $R$  (Figure 3).

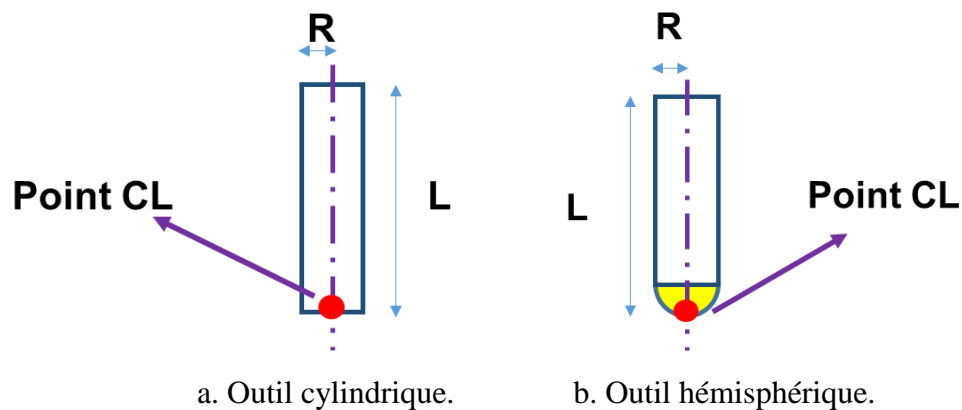
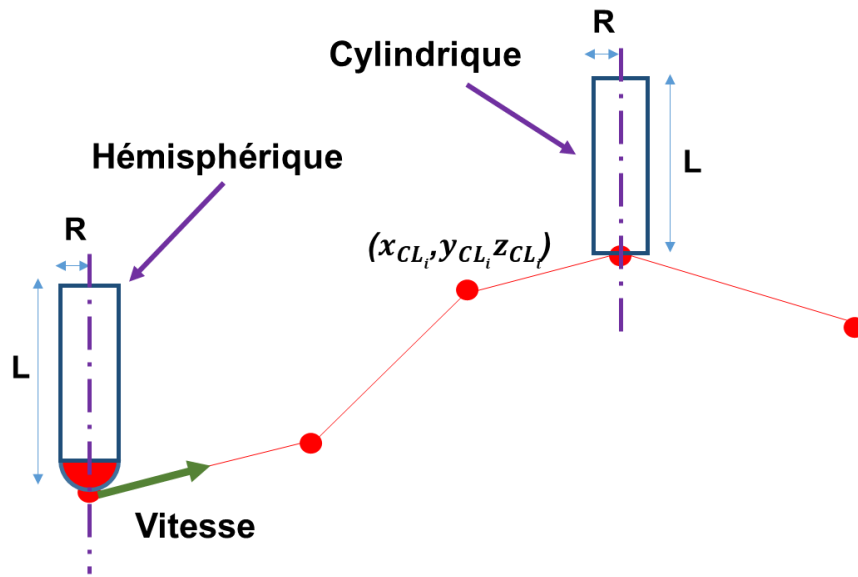


Figure 3. Paramètres géométriques des outils.

- **Coordonnées du point CL** : c'est le point extrémité de l'outil et en même temps c'est le point permettant de piloter la fraiseuse (Figure 3).

Le fichier trajet d'outils « CL-File » est composé d'un ensemble de positions. Chaque position contient la forme et les dimensions de l'outil utilisé dans cette position et les coordonnées  $(x_{CL}, y_{CL}, z_{CL})$  du point CL. Elle contient aussi le mode de déplacement et la vitesse d'avance qui vont être utilisés lors de la génération du programme « G-Code » de la

pièce (Figure 4). Il est à noter que le fichier trajet d'outils utilisé est indépendant de toute opération ou stratégie d'usinage.



a. Trajet d'outils.

Nombre de points (positions)	Rayon	Longueur	$x_{CL}$	$y_{CL}$	$z_{CL}$	Mode	vitesse
100							
Forme							
	50	97	-108.422	100	47.07315	0	1
	50	97	-108.422	100	7.07315	0	1
	50	97	-108.422	100	-2.92685	1	1
	50	97	-88.4218	100	-2.92685	1	1
	50	97	-88.4218	100	7.07315	0	1

b. Syntaxe du fichier « trajet d'outils ».

Figure 4. Trajet et syntaxe du fichier « trajet d'outils » d'une pièce.

### I.2.3 Définition des différents organes de la machine-outil et configuration

Pour la solution proposée, la définition des différents organes de la machine consiste à identifier les organes fixes et les organes mobiles. Pour unifier les traitements, les modèles de ces organes sont de type STL. Donc, toutes les propriétés du modèle STL de la pièce peuvent être calculées (brut, limites, dimensions,...etc.). De plus, deux points caractéristiques sont insérés : le point centre de l'organe et le point origine (Figure 6).

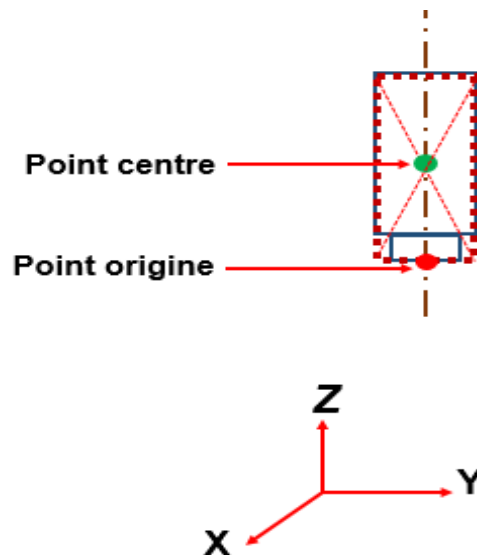


Figure 5. Point centre et point origine d'un organe.

- **Organes fixes** : c'est le bâti et ses accessoires. Il assure le guidage des axes de mouvements et l'avancement des autres organes de la machine. Parmi ses accessoires, le magasin d'outils, les portes, ...etc.
- **Organes mobiles** : c'est les organes les plus importants de la machine. Grâce à leurs déplacements l'usinage s'effectue. Les organes mobiles sont :
  - **Broche** : elle crée le mouvement de coupe nécessaire à l'usinage. Elle assure donc la mise en rotation de l'outil. Elle peut se déplacer suivant les axes X, Y et Z.
  - **Table** : ou support de pièce. Elle est perpendiculaire à l'outil de coupe, La table peut se déplacer suivant les axes X, Y et Z.
- **Géométrie de la machine-outil** : les fraiseuses numériques à 03-axes ont  $2^3$  configurations possibles. Chaque configuration est représentée par les mouvements autorisés de la broche et les mouvements autorisés de la table. Le Tableau 1 montre ces configurations.

Pour la solution proposée, la machine-outil doit être importée en premier lieu et ensuite sa configuration doit être spécifiée de la manière suivante :

- La définition de la « Table » et de la « Broche » de la machine. Ces dernières sont utilisées lors du positionnement de la « Pièce », du « Trajet » et de l'« Outil ».
- La définition des axes de déplacement des organes mobiles de la machine.

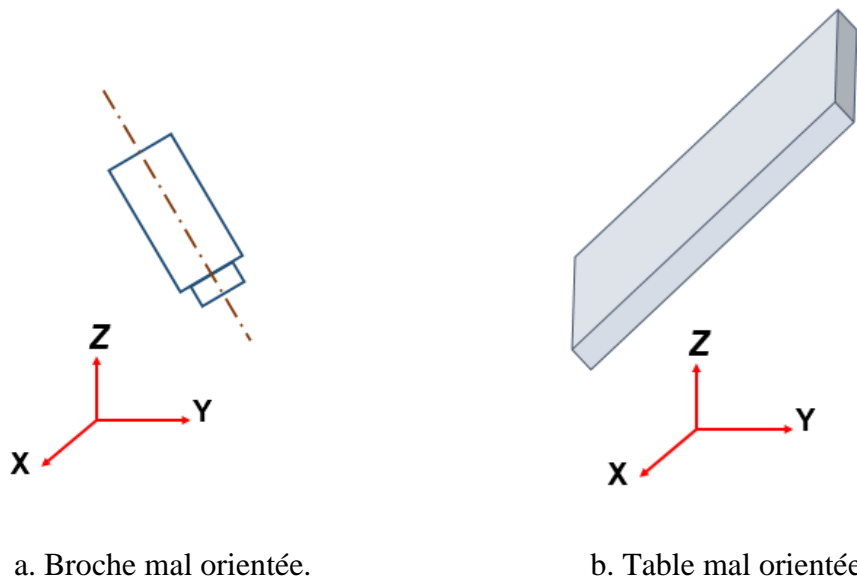
**Tableau 1.** Configurations des fraiseuses à 03-axes.

Cas	Broche Axes de translation	Table Axes de translation
1	0	X, Y, Z
2	X	Y, Z
3	Y	X, Z
4	Z	X, Y
5	X, Z	Y
6	X, Y	Z
7	Y, Z	X
8	X, Y, Z	0

### I.2.4 Orientation de la machine

Après l'importation de la machine-outil, si l'axe de la « Broche » n'est pas parallèle à l'axe Z (Figure 6.a) ou la « Table » n'est pas parallèle aux axes X ou Y (Figure 6.b), il est nécessaire de réorienter ces organes pour obtenir la bonne disposition de la machine. Pour cela, **les matrices de rotation** doivent être utilisées.

La rotation s'effectue autour d'un **axe** et nécessite un **angle  $\theta$**  et le sens de rotation. Un angle positif indique une rotation dans le sens trigonométrique par rapport à l'axe choisi.



**Figure 6.** Organes mal orientés.

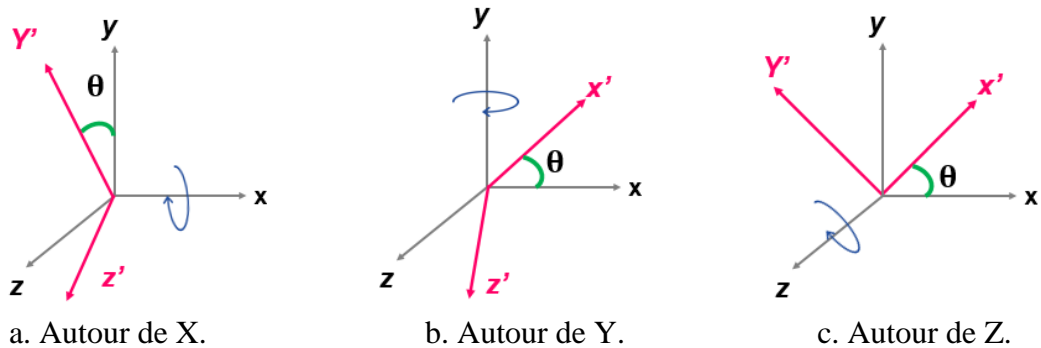
Les matrices de rotation autour des axes X, Y et Z sont données respectivement par :

$$\mathbf{R}_{ox} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix} \quad (2)$$

$$\mathbf{R}_{oy} = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \quad (3)$$

$$\mathbf{R}_{oz} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

La rotation autour de X est donnée par la Figure 7.a, la rotation autour de l'axe Y est donnée par la Figure 7.b et la rotation autour de l'axe Z est donnée par la Figure 7.c.



**Figure 7.** Axes et angles de rotation.

Les nouvelles coordonnées de chaque point des triangles du modèle STL des organes de la fraiseuse sont données par :

- Autour de l'axe X :

$$\begin{cases} x_{\text{apres}} = x_{\text{avant}} \\ y_{\text{apres}} = y_{\text{avant}} * \cos(\theta_x) + z_{\text{avant}} * \sin(\theta_x) \\ z_{\text{apres}} = y_{\text{avant}} * \sin(\theta_x) + z_{\text{avant}} * \cos(\theta_x) \end{cases} \quad (5)$$

- Autour de l'axe Y :

$$\begin{cases} x_{\text{apres}} = x_{\text{avant}} * \cos(\theta_y) + z_{\text{avant}} * \sin(\theta_y) \\ y_{\text{apres}} = y_{\text{avant}} \\ z_{\text{apres}} = -x_{\text{avant}} * \sin(\theta_y) + z_{\text{avant}} * \cos(\theta_y) \end{cases} \quad (6)$$

- Autour de l'axe Z :

$$\begin{cases} x_{\text{apres}} = x_{\text{avant}} * \cos(\theta_z) - y_{\text{avant}} * \sin(\theta_z) \\ y_{\text{apres}} = x_{\text{avant}} * \sin(\theta_z) + y_{\text{avant}} * \cos(\theta_z) \\ z_{\text{apres}} = z_{\text{avant}} \end{cases} \quad (7)$$

### I.2.5 Positionnement de la pièce et du trajet d'outils sur la table

Avant de commencer la simulation, la pièce et le trajet d'outils doivent être positionnés au centre de la face supérieure de la table (Figure 8). Le positionnement est réalisé à partir des étapes suivantes :

**a. Calcul des coordonnées** ( $x_{\text{centre\_pièce}}, y_{\text{centre\_pièce}}, z_{\text{centre\_pièce}}$ ) **du centre de la face basse de la pièce.**

$$\begin{cases} x_{\text{centre\_pièce}} = (x_{\text{min}} + x_{\text{max}})/2 \\ y_{\text{centre\_pièce}} = (y_{\text{min}} + y_{\text{max}})/2 \\ z_{\text{centre\_pièce}} = z_{\text{min}} \end{cases} \quad (8)$$

Avec :

$x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}}, z_{\text{min}}$  sont les limites de la pièce.

**b. Calcul des coordonnées** ( $x_{\text{table}}, y_{\text{table}}, z_{\text{table}}$ ) **du centre de la face supérieure de la « table » :**

$$\begin{cases} x_{\text{table}} = (x_{\text{min}} + x_{\text{max}})/2 \\ y_{\text{table}} = (y_{\text{min}} + y_{\text{max}})/2 \\ z_{\text{table}} = z_{\text{max}} \end{cases} \quad (9)$$

Avec :

$x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}}, z_{\text{max}}$  sont les limites de l'enveloppe de la table.

**c. Calcul des translations** : les nouvelles coordonnées sont calculées en utilisant les translations calculées comme suit :

$$\begin{cases} \text{deltax} = x_{\text{table}} - x_{\text{centre\_pièce}} \\ \text{deltay} = y_{\text{table}} - y_{\text{centre\_pièce}} \\ \text{deltaz} = z_{\text{table}} - z_{\text{centre\_pièce}} \end{cases} \quad (10)$$

**d. Translation de la pièce et du trajet**

Les nouvelles coordonnées de chaque point des triangles du modèle STL de la pièce sont données par :

$$\begin{cases} x_{\text{new\_pièce}} = x_{\text{pièce}} + \text{deltax} \\ y_{\text{new\_pièce}} = y_{\text{pièce}} + \text{deltay} \\ z_{\text{new\_pièce}} = z_{\text{pièce}} + \text{deltaz} \end{cases} \quad (11)$$

Avec ( $x_{\text{pièce}}, y_{\text{pièce}}, z_{\text{pièce}}$ ) sont les coordonnées des points du modèle STL de la pièce avant positionnement et ( $x_{\text{x\_new\_pièce}}, y_{\text{y\_new\_pièce}}, z_{\text{z\_new\_pièce}}$ ) sont les coordonnées des points du modèle STL de la pièce après positionnement.

De même pour le « Trajet » (sachant que le trajet est par défaut positionné par rapport à la pièce) :

$$\begin{cases} x_{\text{new\_trajet}} = x_{\text{trajet}} + \text{deltax} \\ y_{\text{new\_trajet}} = y_{\text{trajet}} + \text{deltay} \\ z_{\text{new\_trajet}} = z_{\text{trajet}} + \text{deltaz} \end{cases} \quad (12)$$

Avec  $(x_{\text{trajet}}, y_{\text{trajet}}, z_{\text{trajet}})$  sont les coordonnées des points de passage d'outils avant positionnement et  $(x_{\text{new\_trajet}}, y_{\text{new\_trajet}}, z_{\text{new\_trajet}})$  sont les coordonnées des points de passage d'outils après positionnement.

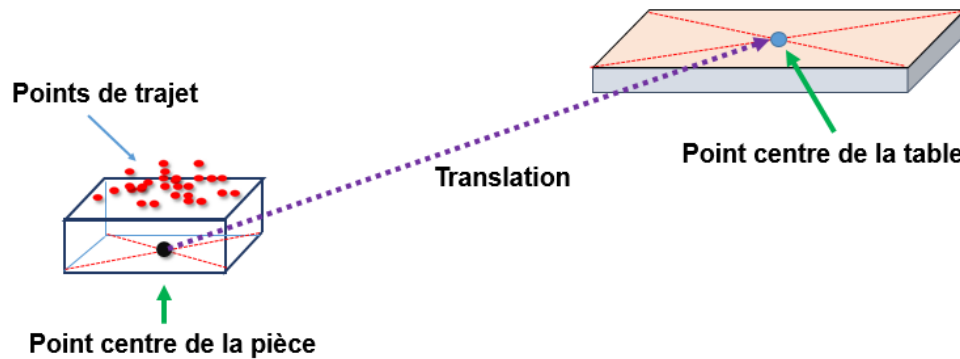


Figure 8. Positionnement de la pièce au-dessus de la « Table ».

### I.2.6 Simulation géométrique des mouvements de la machine

Le positionnement des différents organes de la machine pour chaque position du trajet d'outils est effectué suite au calcul des **positions articulaires** en utilisant le MGI (Modèle Géométrique Inverse).

Pour pouvoir calculer les positions articulaires, et par la suite positionner les organes pour chaque position du trajet, il faut effectuer ces étapes :

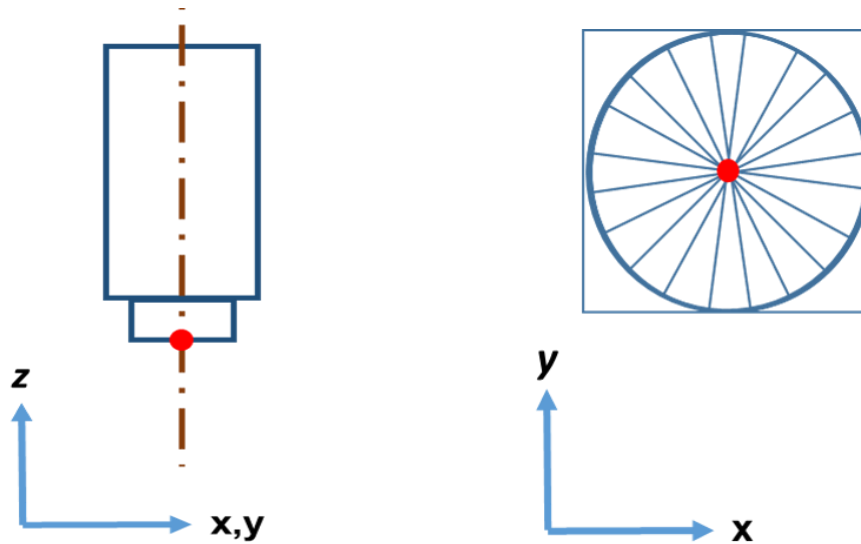
**a. Récupération d'une position du trajet :** récupérer depuis le trajet d'outils, une position (forme, longueur, rayon, coordonnées du point CL).

**b. Récupération des coordonnées du point origine de la « Broche » et ses déplacements suivant les 03 axes :** la broche est la base choisie pour trouver l'ensemble des positions articulaires.

Pour trouver le point origine de la broche (Figure 9), il faut :

- Récupérer le  $z_{\min}$  du modèle STL de la « Broche »
- Ensuite, chercher tous les points des triangles du modèle STL ayant le même  $z_{\min}$ , et récupérer les valeurs des limites  $x_{\min}$ ,  $y_{\min}$ ,  $x_{\max}$ ,  $y_{\max}$ .
- Calculer le point origine

$$\begin{cases} \text{origine\_broche}_x = (x_{\min} + x_{\max})/2 \\ \text{origine\_broche}_y = (y_{\min} + y_{\max})/2 \\ \text{origine\_broche}_z = z_{\min} \end{cases} \quad (13)$$



a. Projection sur le plan XZ/YZ.      b. Projection sur le plan XY.

**Figure 9.** Point origine de la broche.

**c. Calcul des translations suivant les trois axes X, Y et Z :** pour cela, la géométrie de la fraiseuse doit être considérée (Figure 10) :

Si la broche est fixe suivant l'axe X :

$$\text{deltax} = \text{origine\_broche}_x - x \quad (14)$$

Sinon

$$\text{deltax} = -(\text{origine\_broche}_x - x) \quad (15)$$

Si la broche est fixe suivant l'axe Y :

$$\text{deltay} = \text{origine\_broche}_y - y \quad (16)$$

Sinon

$$\text{deltay} = -(\text{origine\_broche}_y - y) \quad (17)$$

Si la broche est fixe suivant l'axe Z :

$$\text{deltaz} = -(z - \text{origine\_broche}_z) + \text{longueur\_outil} \quad (18)$$

Sinon

$$\text{deltaz} = z - \text{origine\_broche}_z + \text{longueur\_outil} \quad (19)$$

Avec  $(\text{origine\_broche}_x, \text{origine\_broche}_y, \text{origine\_broche}_z)$  sont les coordonnées du point origine de la broche, longueur\_outil est la longueur de l'outil utilisé à cette position, et x, y, z sont les coordonnées de la position.



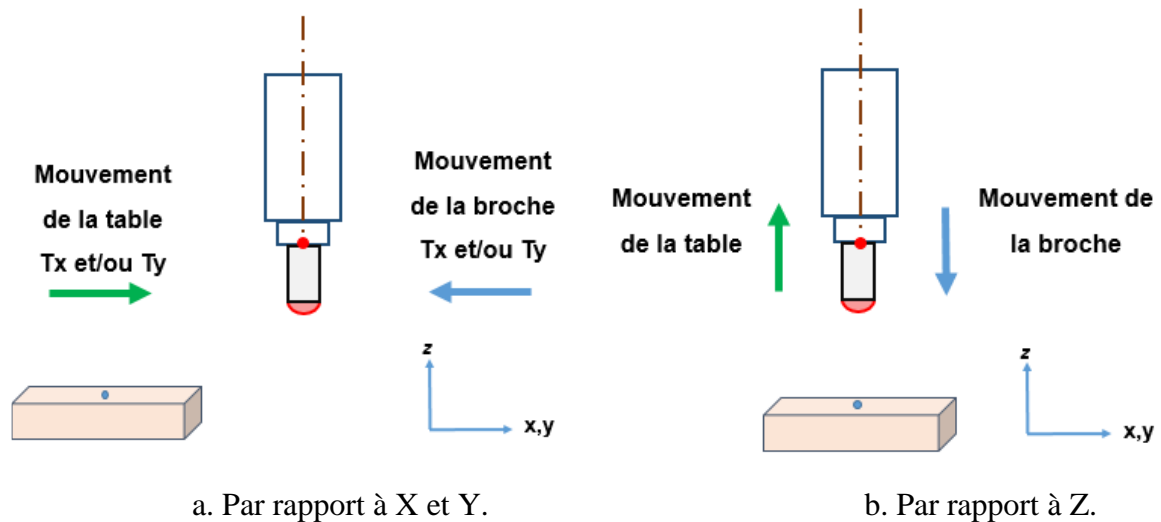


Figure 10. Géométrie de la machine.

**d. Translation des organes mobiles, de la pièce et du trajet :** les organes mobiles, pièce et trajet (sachant que la pièce et le trajet ont le même déplacement que la table) se déplacent selon leurs axes de déplacement en utilisant les translations calculées.

Les méthodes de translation pour traduire le trajet, la pièce et les organes de la machine doivent traiter toutes les configurations possibles de la machine (08 configurations).

### I.2.7 Génération du programme « G Code » et du fichier d'outils

Pour notre système, après la fin de la simulation virtuelle des mouvements des organes mobiles de la machine, le programme « G-Code » de la pièce simulée est généré à partir de son fichier de trajet. Pour générer celui-ci, il faut d'abord définir :

- Le numéro du programme.
- Les unités utilisées pour l'ensemble du programme.
- L'incrément entre les lignes du programme.
- L'origine du programme parmi les cinq (05) origines proposés (Figure 11).
- Le sens de rotation de l'outil.
- L'arrosage.
- La vitesse de rotation de la broche (outil).

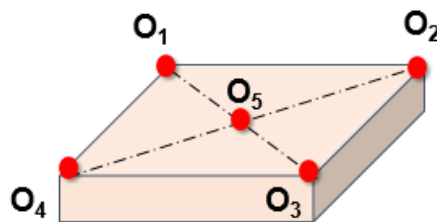


Figure 11. Origines proposés du programme « G-Code ».

Après la définition de ces paramètres, les nouvelles positions que doit occuper l'outil en usinage réel sont calculées tout en prenant en considération la géométrie de la machine (Figure 12). Cela, est traduit par :

Si le trajet est fixe selon X

$$x_{CL\_new} = origine_x - x_{CL} \quad (20)$$

Sinon

$$x_{CL\_new} = -(origine_x - x_{CL}) \quad (21)$$

Si le trajet est fixe selon Y

$$y_{CL\_new} = origine_y - y_{CL} \quad (22)$$

Sinon

$$y_{CL\_new} = -(origine_y - y_{CL}) \quad (23)$$

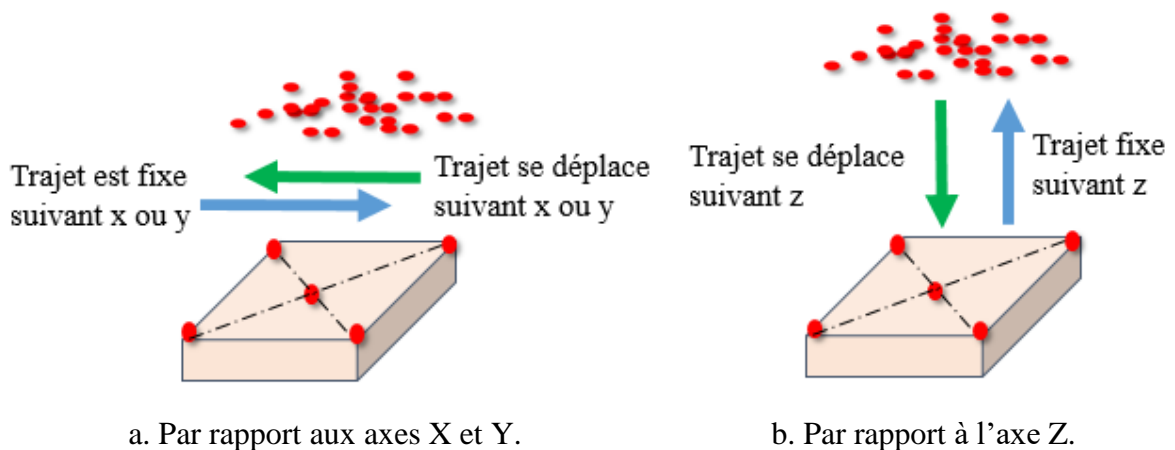
Si le trajet est fixe selon Z

$$z_{CL\_new} = origine_z - z_{CL} \quad (24)$$

Sinon

$$z_{CL\_new} = -(origine_z - z_{CL}) \quad (25)$$

Avec  $(x_{CL}, y_{CL}, z_{CL})$  les coordonnées des points CL du trajet,  $(x_{CL\_new}, y_{CL\_new}, z_{CL\_new})$  les nouvelles positions et  $(origine_x, origine_y, origine_z)$  sont les coordonnées de l'origine programme choisi.



**Figure 12.** Calcul des positions d'outils pour le programme « G-Code ».

Le fichier « G-Code » obtenu doit être optimisé pour minimiser sa taille. Pour cela, si une coordonnée se répète en plusieurs lignes successives (ensemble de lignes), elle sera mentionnée seulement dans la première ligne de l'ensemble, la suite des lignes ne contient

pas cette coordonnée jusqu'à ce que cette dernière change (Figure 13). De même pour la vitesse d'avance de l'outil, l'indice de l'outil utilisé et les fonctions modales (G0 et G1).

Pour accomplir cette tâche, il faut vérifier pour chaque position que ses coordonnées, sa vitesse d'avance, son mode de déplacement et son outil sont différents de ceux de la position précédente.

N10 G00	X10 Y30	Z 50	N10 G00 X10 Y30 Z 50
N20 G00	X10 Y30	Z 60	N20 Z 60
N30 G00	X10 Y30	Z 70	N30 Z 70
N40 G00	X50 Y25	Z 15	N40 X50 Y25 Z 15

a. Sans optimisation.

b. Avec optimisation.

**Figure 13.** Optimisation du fichier « G-Code ».

Une fois le programme G-Code généré, il est indispensable de connaître les outils utilisés (forme et dimensions) ainsi que leurs emplacements dans le magasin d'outils afin d'éviter les risques d'interférences et de collisions pendant l'usinage réel de la pièce (Figure 14).

Outil 1  
Forme cylindrique  
Rayon 50  
Longueur 150

Outil 2  
Forme hémisphérique  
Rayon 25  
Longueur 135

a. Premier outil.

b. Deuxième outil.

**Figure 14.** Paramètres des outils utilisés.

## II. Modélisation UML

Le langage UML « *Unified Modeling Language* » est un langage de modélisation orientée objet développé en réponse à l'appel aux propositions lancées 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 classes.

## II.1 Diagramme de cas d'utilisation

C'est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système. Le diagramme de cas d'utilisation permet de recueillir, d'analyser et d'organiser les besoins de ce dernier. Il permet d'identifier les interactions entre le système et les acteurs, **c'est-à-dire toutes les fonctionnalités que doit fournir le système.**

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

Dans ce qui suit, les grandes fonctionnalités du système à développer sont présentées dans un diagramme de cas d'utilisation (Figure 15).

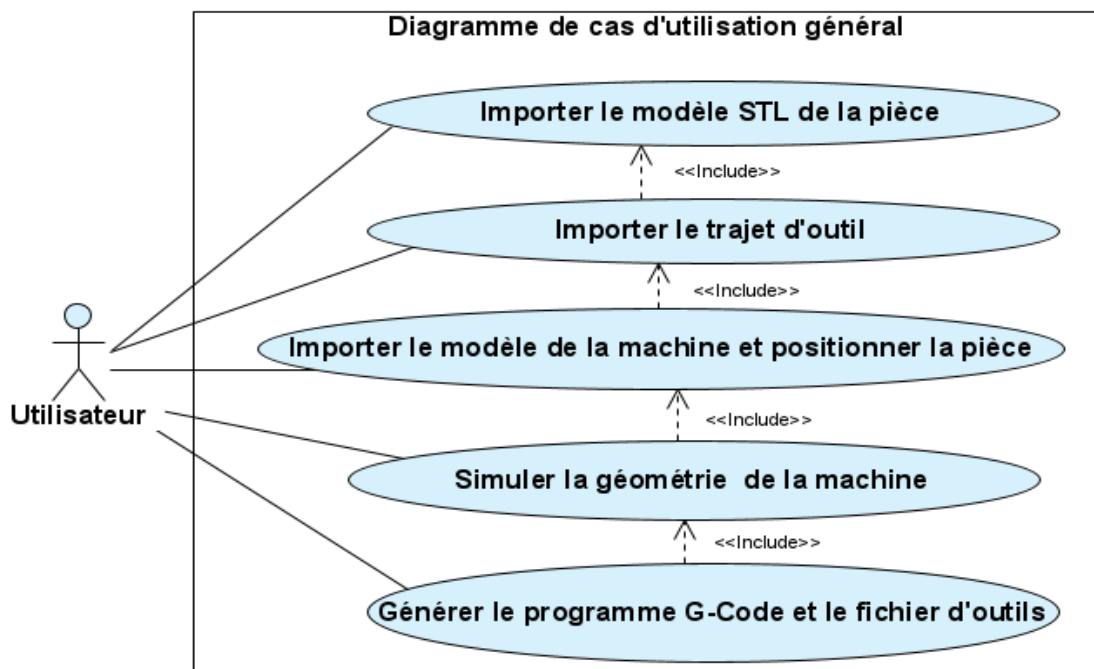


Figure 15. Diagramme de cas d'utilisation général.

### II.1.2 Diagramme de cas d'utilisation « Importer le modèle de la machine et positionner la pièce »

Ce cas comme montre la Figure 16, consiste à :

- Calculer les limites et les dimensions de la machine à simuler.
- Visualiser la machine et changer son orientation selon les trois (03) axes.
- Sélectionner un organe, visualiser son brut et son origine.
- Configurer la machine.
- Positionner la pièce.

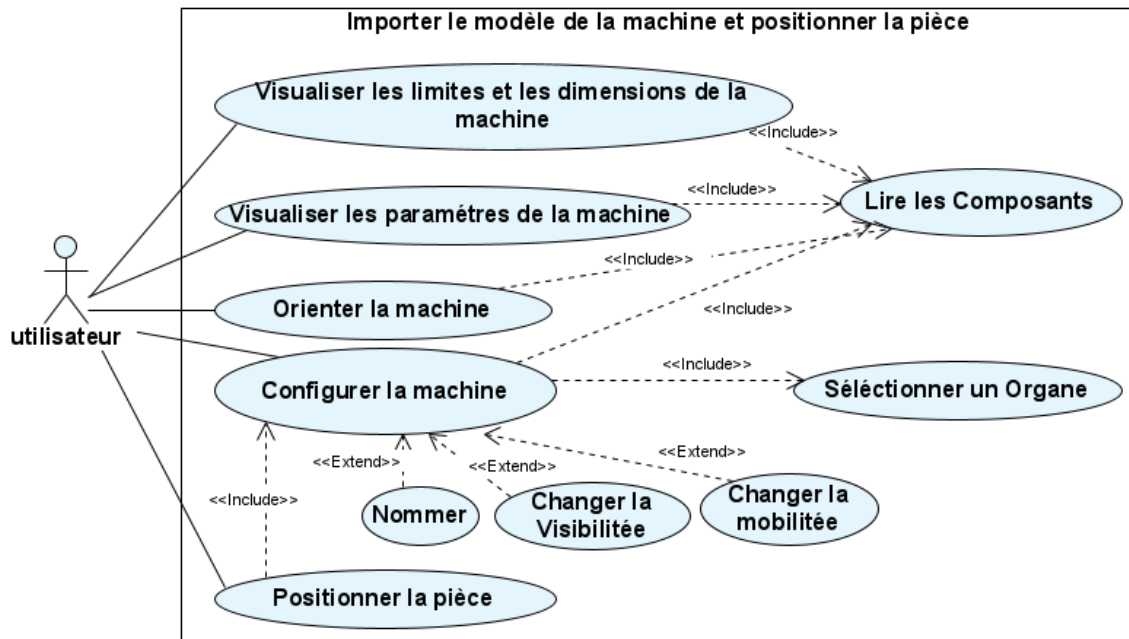


Figure 16. Diagramme de cas « Importer le modèle de la machine et positionner la pièce ».

### II.1.3 Diagramme de cas d'utilisation « Simuler la géométrie de la machine »

Dans ce cas, l'utilisateur peut déplacer les organes mobiles suivant les trois axes et simuler manuellement ou automatiquement les mouvements des organes mobiles (Figure 17).

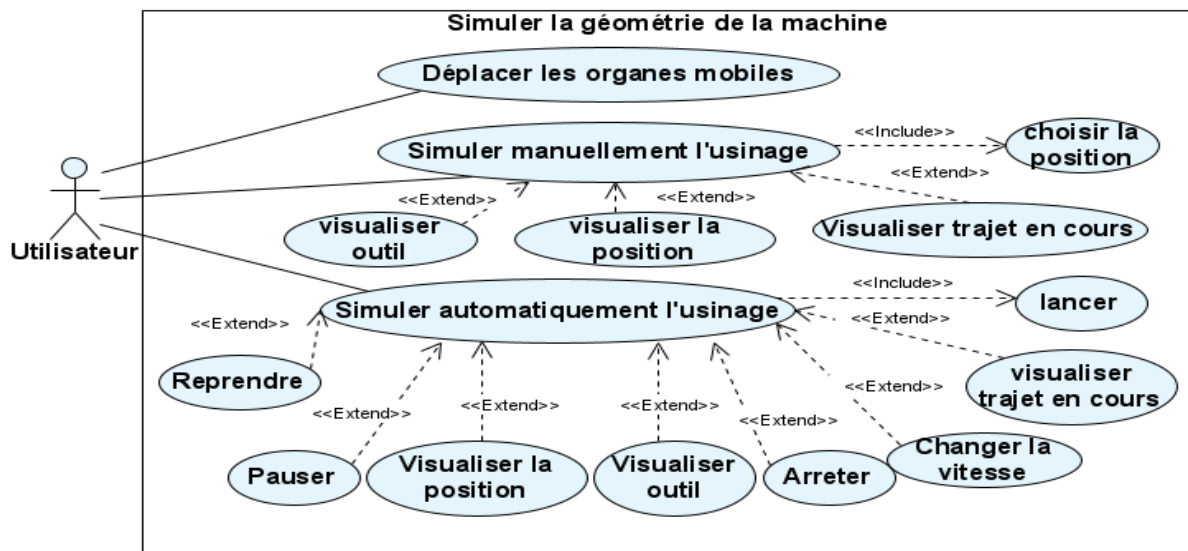


Figure 17. Diagramme de cas « Simuler la géométrie de la machine ».

### II.1.4 Diagramme de cas d'utilisation « Générer le programme G-Code et le fichier d'outils »

Ce cas consiste à générer le fichier « G-Code » du trajet d'outils associé à la pièce simulée et à générer le fichier des outils utilisés lors de l'usinage.

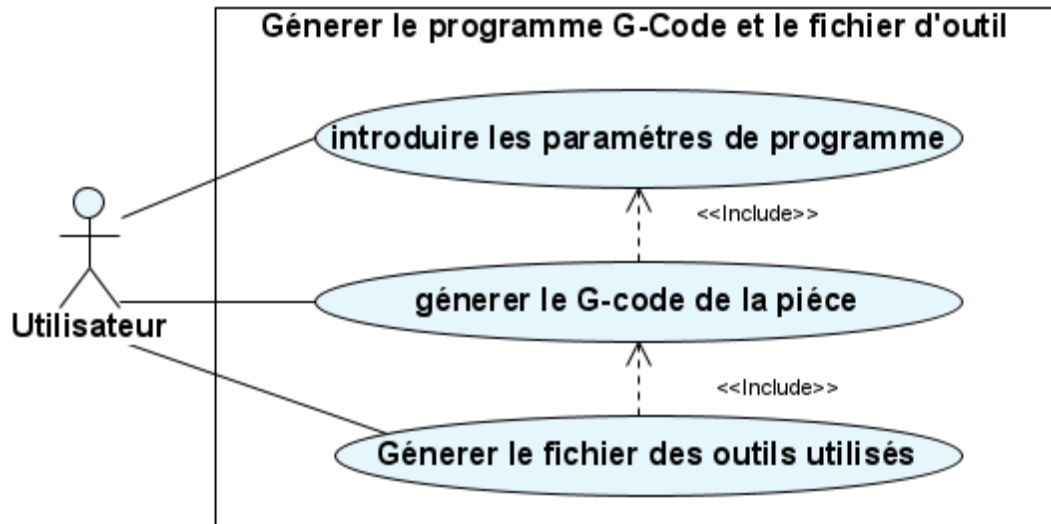


Figure 18. Diagramme de cas « Générer le programme G-Code et le fichier d'outils ».

## II.2 Diagramme d'activité

Un diagramme d'activité permet de modéliser un processus interactif, global ou partiel pour un système donné. Nous l'avons utilisé pour mettre l'accent sur les activités, leurs relations et leurs impacts sur les objets. Il permet aussi de représenter les transitions entre les activités ainsi que leurs synchronisations. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

### II.2.1 Diagramme d'activité général

Dans ce qui suit, le scénario général du système à développer est présenté par le diagramme d'activité général (Figure 19).

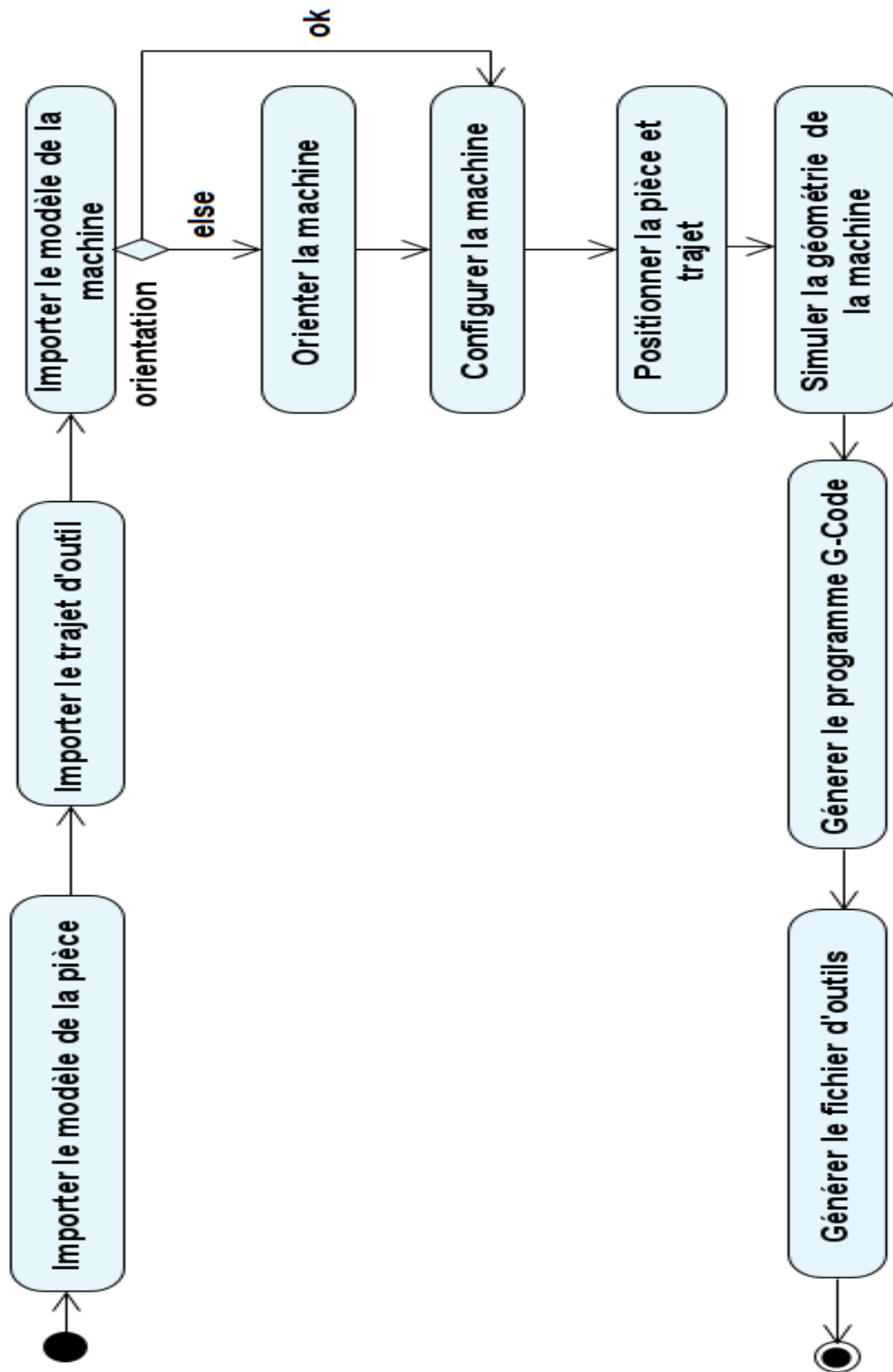
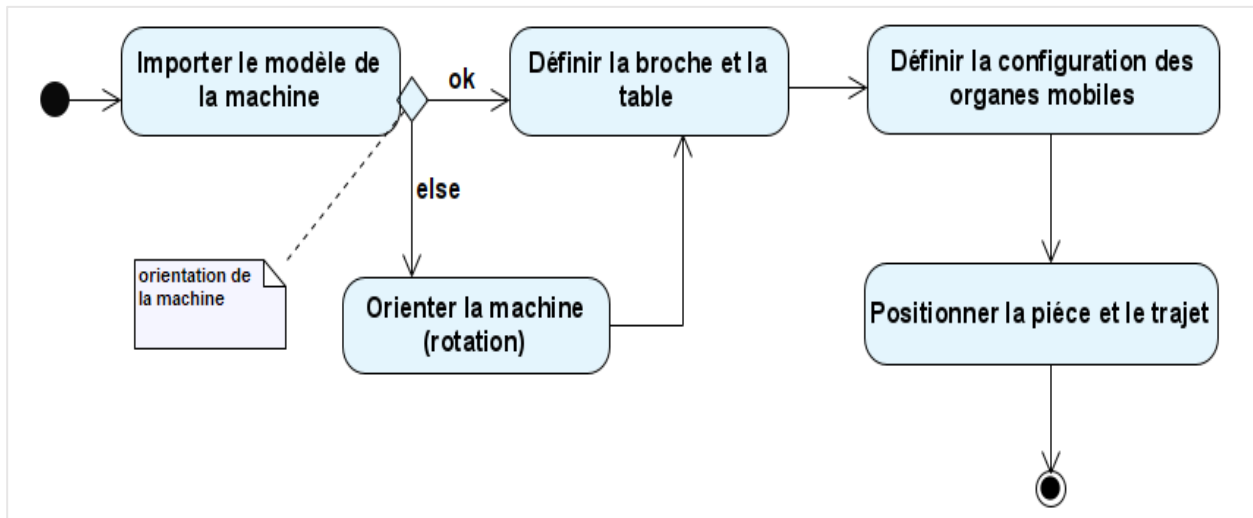


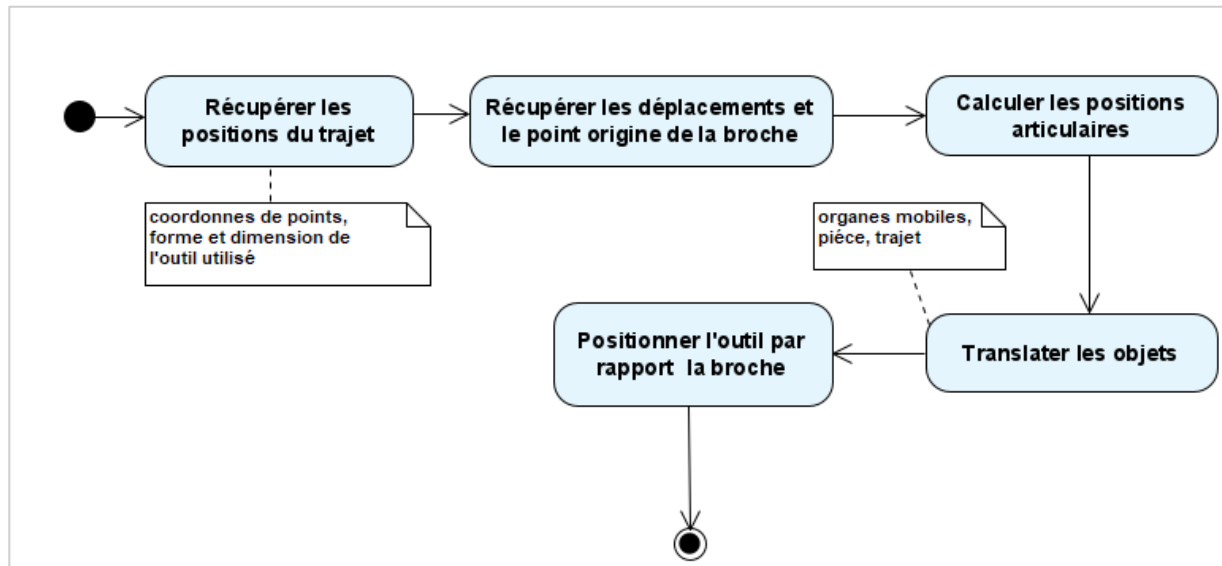
Figure 19. Diagramme d'activité général.

**II.2.2 Diagramme d'activité de cas « Importer le modèle de la machine et positionner la pièce »**



**Figure 17.** Diagramme d'activité de cas « Importer le modèle de la machine et positionner la pièce ».

**II.2.3 Diagramme d'activité de cas « Simuler la géométrie de la machine »**



**Figure 21.** Diagramme d'activité de cas « Simuler la géométrie de la machine ».



### II.2.4 Diagramme de cas « Générer le programme G-Code et le fichier d'outils »

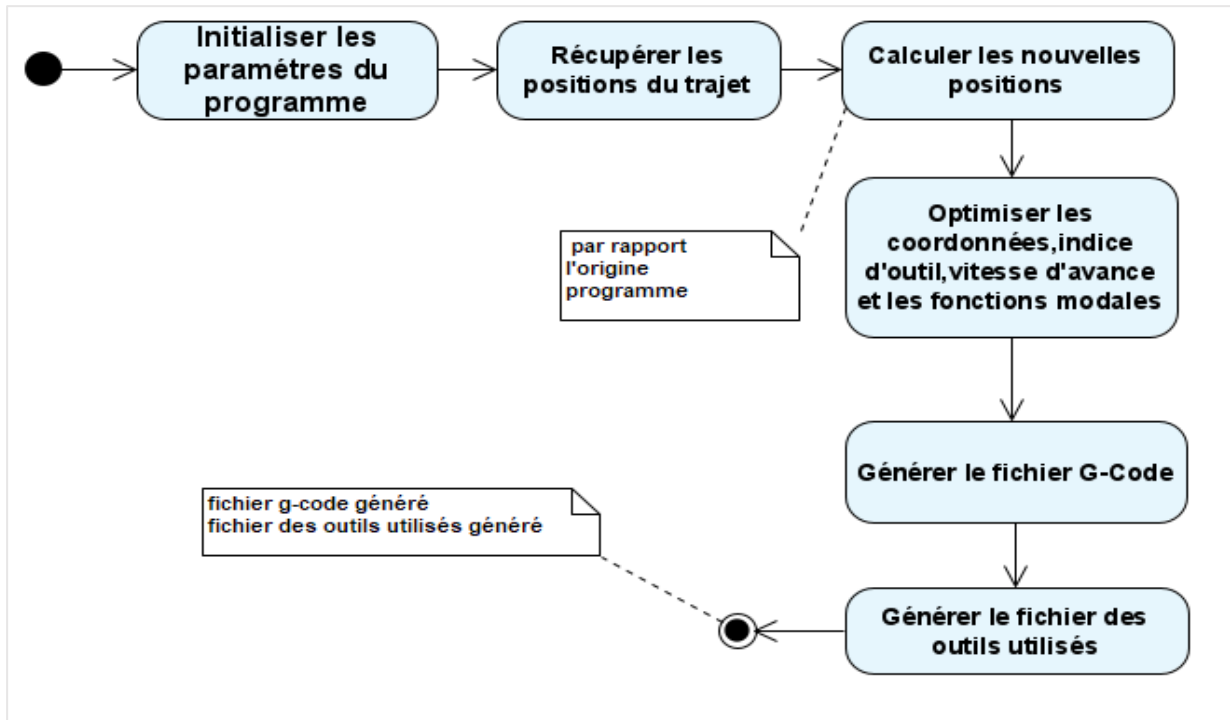


Figure 22. Diagramme d'activité de cas « Générer le programme G-Code et le fichier d'outils ».

## II.3 Diagramme de classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet. Il décrit clairement **la structure interne** d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets. Donc, il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.

### II.3.1 Diagramme de classes général

Dans ce qui suit, le diagramme de classes d'analyse est présenté en tenant compte des relations entre les classes (Figure 23).

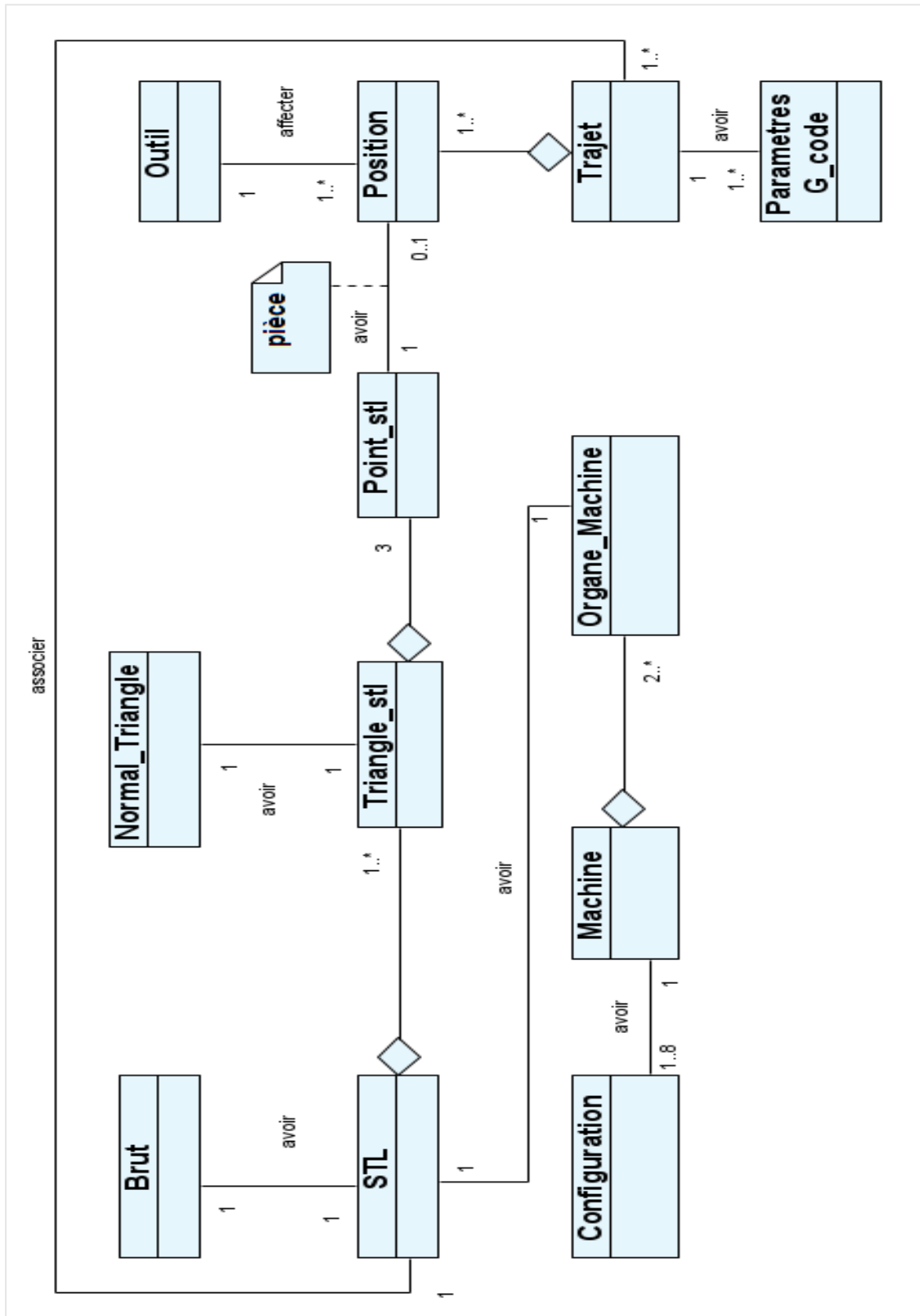


Figure 23. Diagramme de classes général.

### II.3.2 Classes utilisées

Les classes qui définissent notre système sont détaillées comme suit :

- **Classe « STL »** : c'est la classe principale. Elle regroupe tous les sommets et les triangles du modèle STL, ainsi que les informations du brut de ce dernier. Ses méthodes permettent de récupérer les informations du modèle STL, de dessiner le modèle et de changer son repère (translation/rotation) (Figure 24).

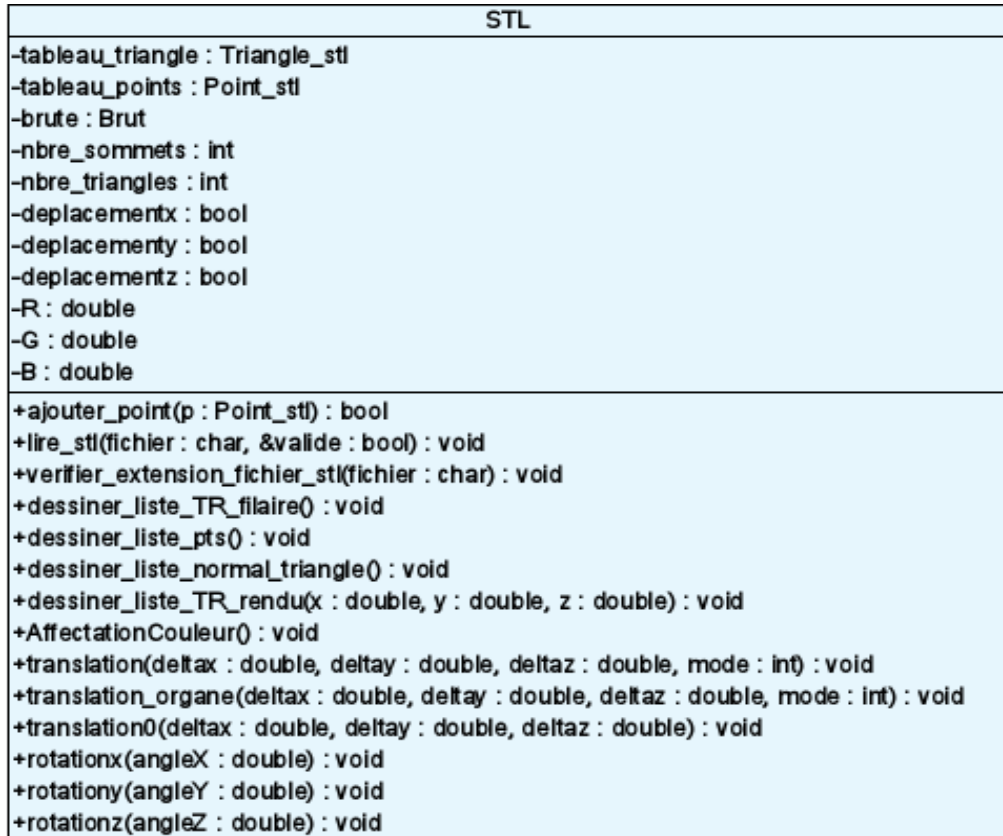


Figure 24. Classe « STL ».

- **Classe « Brut »** : cette classe détermine les dimensions minimales d'un modèle STL (Figure 25).

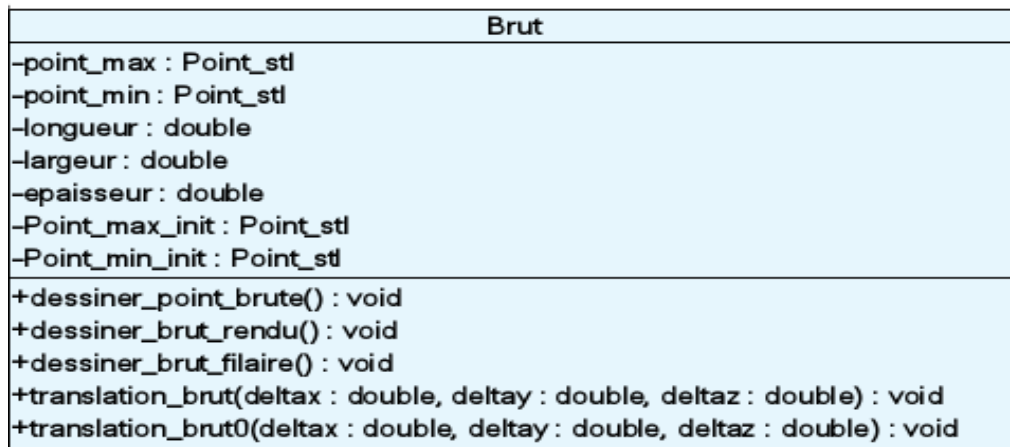


Figure 25. Classe « Brut ».

- **Classe « Triangle\_stl »** : cette classe englobe les différentes caractéristiques d'un triangle. Ses fonctions sont données pour représenter ce dernier (Figure 26).

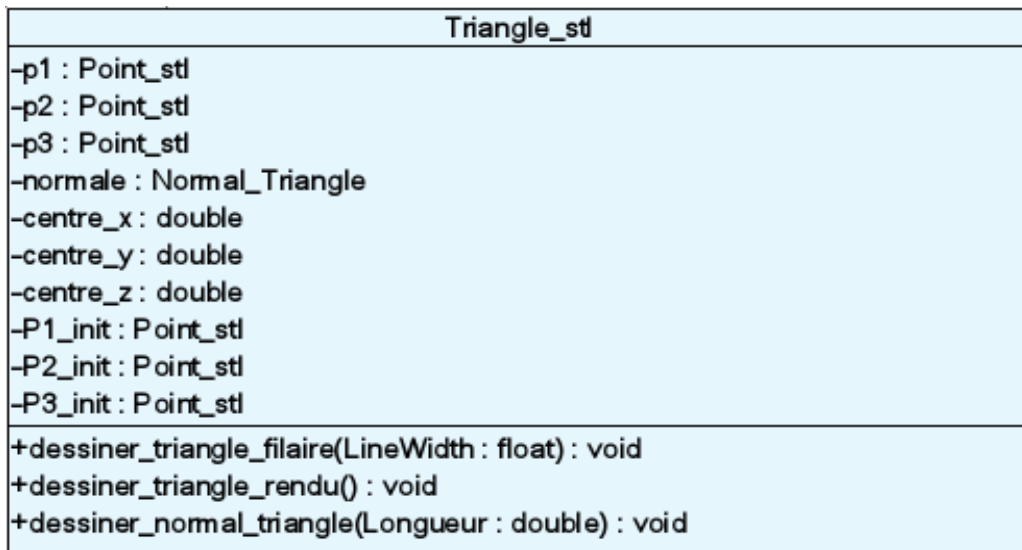


Figure 26. Classe « Triangle\_STL ».

- **Classe « Normal\_Triangle »** : elle représente la normale d'un triangle (Figure 27).

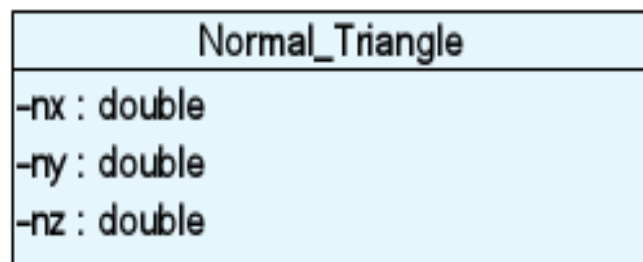


Figure 27. Classe « Normal\_Triangle ».

- **Classe « Point\_stl »** : elle regroupe toutes les caractéristiques d'un point (sommet) (Figure 28).

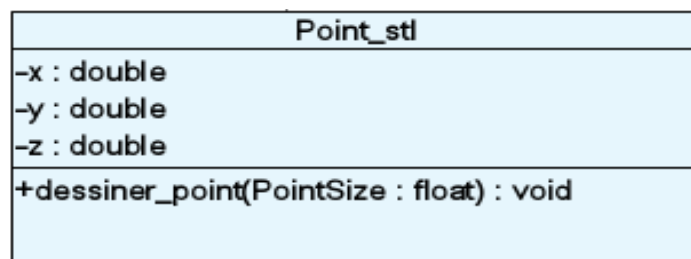


Figure 28. Classe « Point\_STL ».

- **Classe « Outil »** : elle représente les paramètres d'un outil (Figure 29).

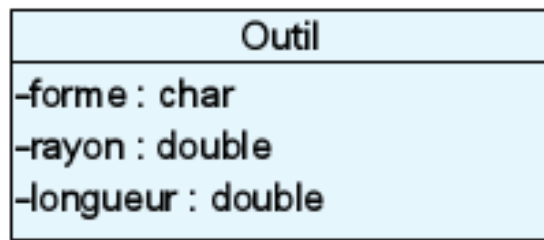


Figure 29. Classe « Outil ».

- **Classe « Position »** : elle représente une position du fichier trajet, c'est-à-dire les coordonnées du point CL et l'outil qui lui est affecté. Ses méthodes permettent de dessiner l'outil et la position. (Figure 30).

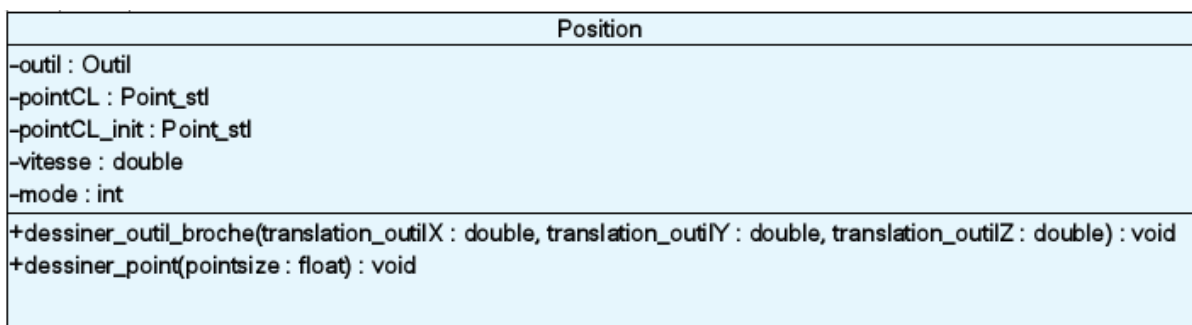


Figure 30. Classe « Position ».

- **Classe « Trajet »** : c'est l'ensemble des positions d'outils. Elle permet de lire un fichier de trajet. Ses méthodes permettent de dessiner le trajet, de changer son repère et de générer le fichier « G-Code » (Figure 31).

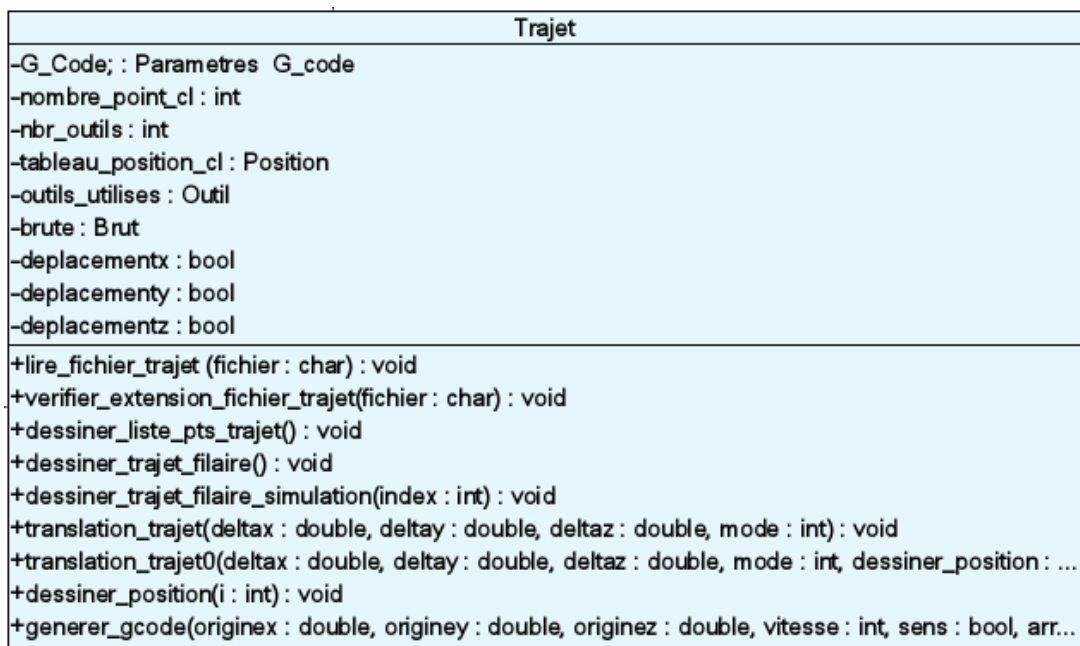


Figure 31. Classe « Trajet ».

- **Classe « Organe\_Machine »** : cette classe représente un organe de la machine. Ses méthodes permettent de représenter chaque organe, de changer son repère (translation/rotation), de dessiner plusieurs paramètres (origine, organe, ...etc.) (Figure 32).

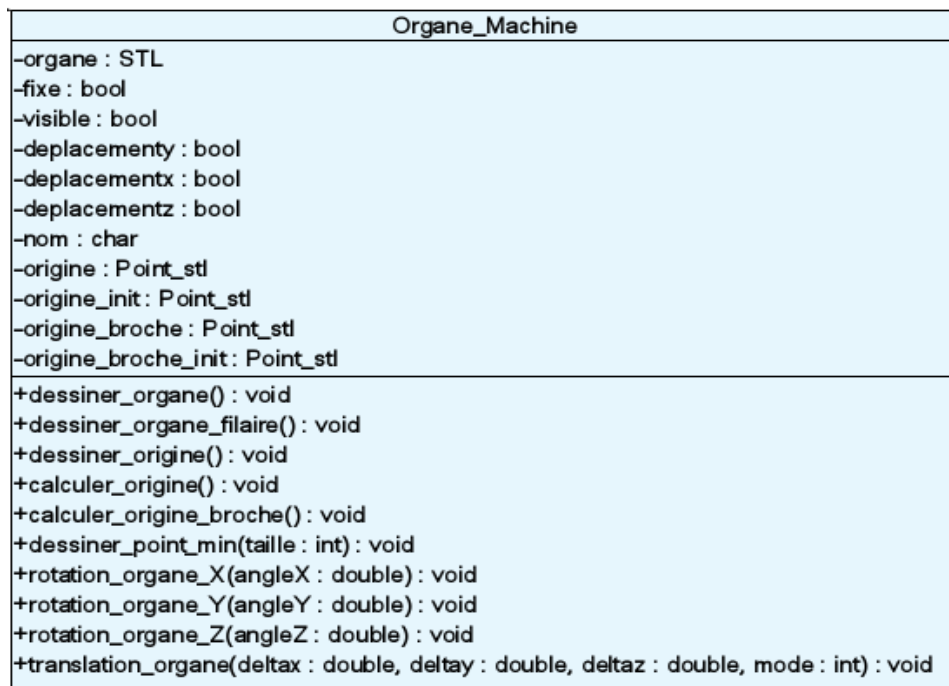


Figure 32. Classe « Organe\_Machine ».

- **Classe « G-Code »** : cette classe contient l'ensemble des paramètres du fichier « G-Code » et du fichier d'outils (Figure 33).

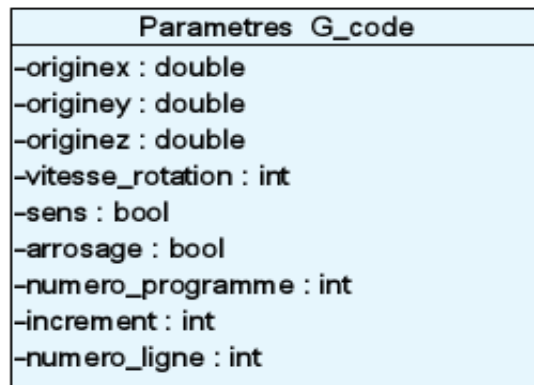


Figure 33. Classe « Paramètres G-code ».

- **Classe « machine »** : regroupe l'ensemble des organes de la machine. Ses méthodes permettent de dessiner la machine, de la configurer et de changer son repère (rotation/translation) (Figure 34).

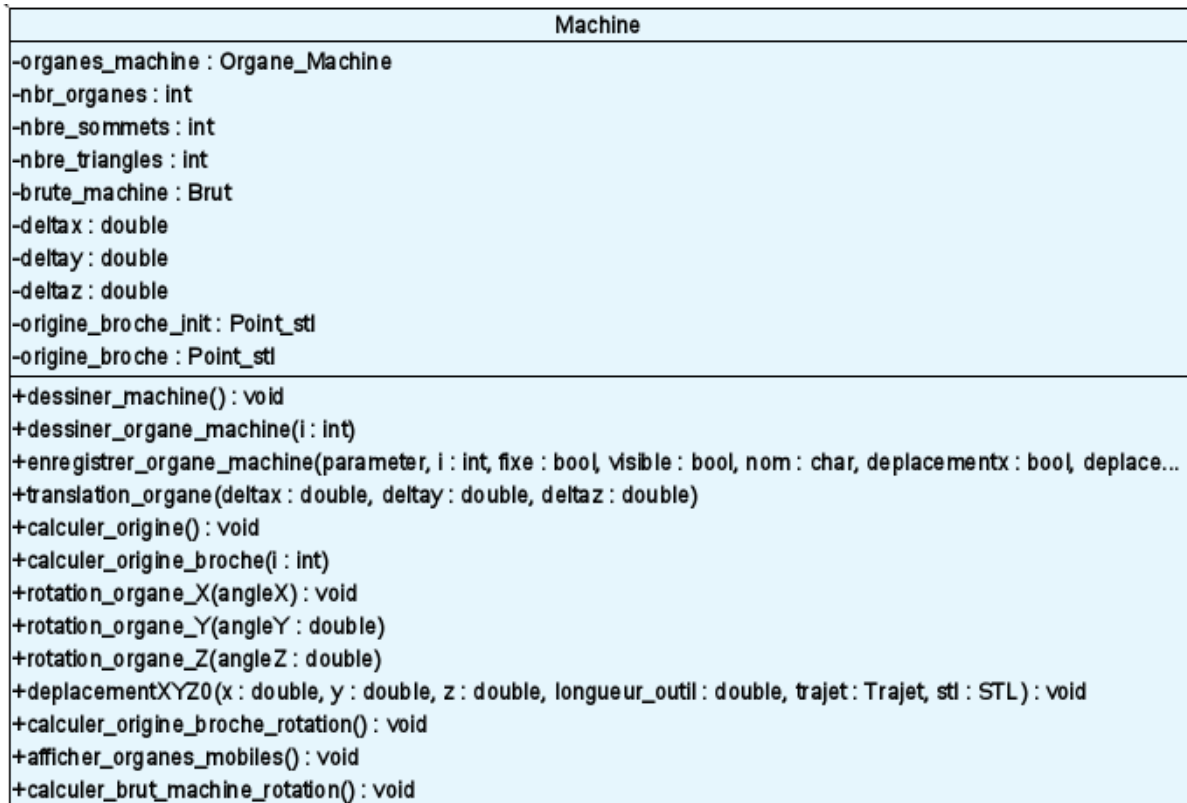


Figure 34. Classe « Machine ».

## Conclusion

Dans ce chapitre, nous avons présenté l'architecture générale de notre application d'un point de vue conceptuel, en spécifiant toutes les étapes suivies de la solution proposée.

Ensuite, nous avons présenté la structure de notre application et son déroulement en utilisant le langage de modélisation UML avec les diagrammes de cas d'utilisation, les diagrammes d'activité et les diagrammes de classes.

L'implémentation, le test et la validation de notre application seront traités dans le chapitre suivant.



# Chapitre III

## **Implémentation et validation**

Introduction

I. Présentation des outils utilisés

II. Présentation de l'environnement « CFAO »

III. Présentation de l'application développée

IV. Test et validation

Conclusion



### Introduction

Après avoir présenté la conception de notre système dans le deuxième chapitre, l'objectif de ce chapitre est l'illustration des différentes tâches que notre système peut effectuer. Nous commençons par présenter l'application conçue ainsi que l'environnement de développement et les outils utilisés pour sa mise en œuvre. Par la suite, une présentation de l'interface utilisateur et une validation du système à travers des exemples de pièces.

Le but de ce chapitre est de donner une vue à la fois intelligible et concrète sur l'application en présentant les informations manipulées à travers les exemples de validation.

### I. Présentation des outils 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 objet 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, etc. 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, etc.) et les algorithmes les plus usuels.

Parmi les environnements de développement nous citons : AnjutaDevStudio, C++ Builder, Code : Blocks (open-source), Dev-C++, Eclipse (open-source), Microsoft Visual C++, etc. [24].

### I.2 Présentation de l'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. Elle favorise l'innovation et la simplicité d'utilisation en offrant aux développeurs de nombreuses fonctionnalités de visualisation. Etant libre, ouverte et portable, la bibliothèque graphique a pu se vendre dans le milieu scientifique et industriel, et être utilisée dans de nombreuses applications artistiques ou de traitement d'images. Du fait de ses performances, l'API OpenGL est aussi présente dans différents domaines 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 programmation différents dont des langages de script comme Python ou Ruby.



### I.3 Présentation de l'Embarcadero C++ Builder Seattle

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 ++ [25].



## II. Présentation de l'environnement « CFAO »

C'est une plateforme logicielle, développée par l'équipe « CFAO » du « CDTA », dédiée à la production numérique des pièces de formes complexes sur fraiseuses numériques multiaxes (03-axes et 05-axes). La fenêtre principale de la plateforme 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 les 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 aussi tous les modules logiciels intégrés à la plateforme logicielle.



**Figure 18.** Fenêtre principale de la plateforme.

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 surfaces 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.

### III. Présentation de l'application développée

La partie contenant notre module logiciel développé dans le cadre de ce travail est intégré à la plateforme logicielle, et est lancée à partir de « Simulation Fraiseuse 03-Axes et G-Code » qui se trouve dans la rubrique « Option » (Figure 2).

Le module logiciel développé permet le lancement de plusieurs fonctionnalités complémentaires. Il est composé de cinq (05) onglets (Figure 3).

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

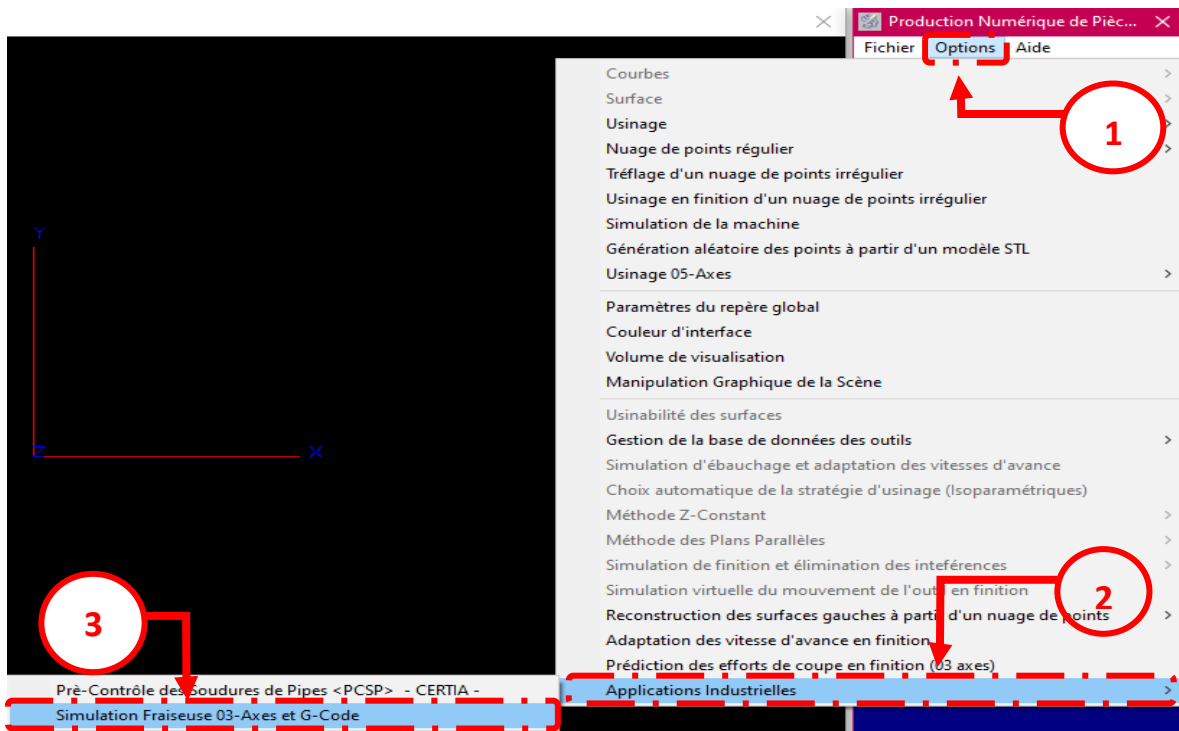


Figure 19. Lancement du module logiciel développé.

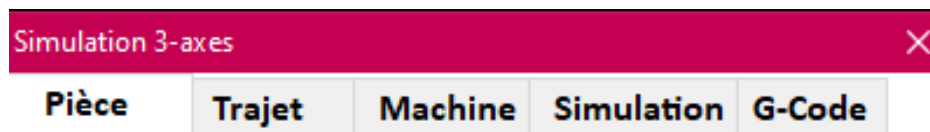


Figure 20. Onglets du module logiciel développé.

### III.1 Onglet « Pièce »

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 de la pièce. Ceci est réalisé par un simple clic sur le bouton « Lire fichier STL » (Figure 4) suivi par la sélection du fichier à ouvrir depuis la boîte de dialogue qui apparaît. Cette dernière est personnalisée et n'affiche que les fichiers de type « STL ».

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

- Les dimensions minimales de la pièce brute.
- Le nombre total de points et de triangles.

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

- Les sommets du brut.
- Le brut du modèle en deux modes filaire et rendu.
- Les sommets et les normales des triangles du modèle.
- Les triangles du modèle en deux modes filaire et rendu.

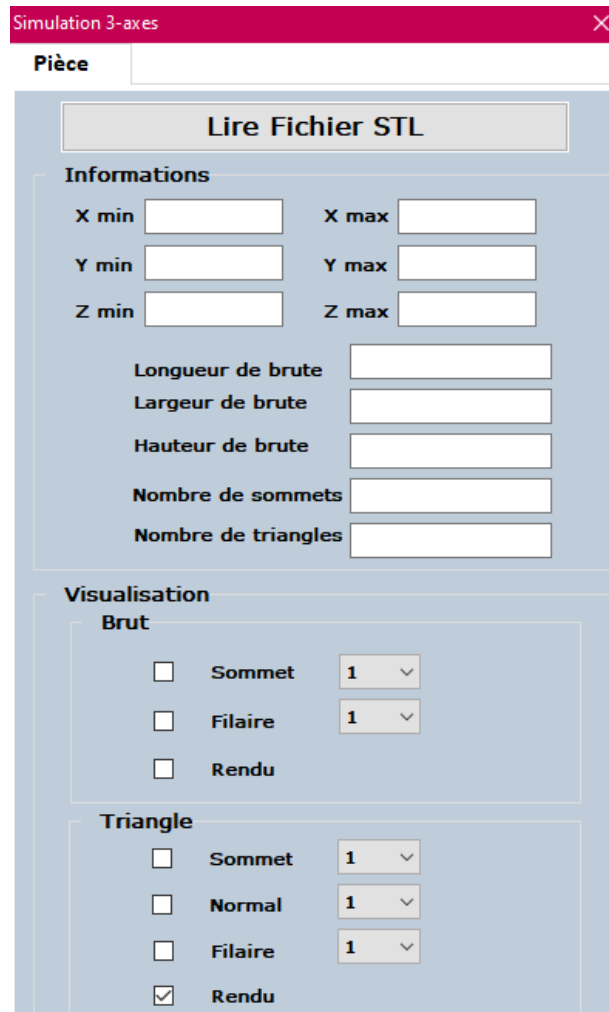


Figure 4. Onglet « Pièce ».

### III.2 Onglet « Trajet »

Le deuxième onglet permet la lecture du fichier texte contenant le trajet d'outils. En important le trajet à partir du bouton « lire le trajet d'outils » (Figure 5), suivi par la sélection du fichier à ouvrir depuis la boîte de dialogue qui apparaît. Cette dernière est personnalisée et n'affiche que les fichiers de type « txt ».

Il permet aussi d'afficher :

- Le nombre total de points de passage.
- Les points de passage.
- Le trajet d'outils.

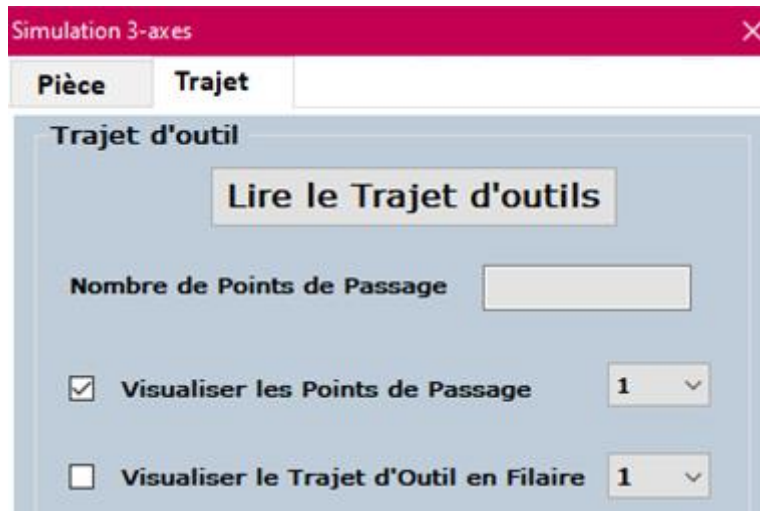


Figure 21. Onglet « Trajet ».

### III.3 Onglet « Machine »

Le troisième onglet permet la lecture et la visualisation des paramètres STL de l'ensemble des organes de la machine (Figure 6). Ceci est réalisé par un simple clic sur le bouton « **Lire les composants** » suivi par la sélection des fichiers à ouvrir filtrés.

Une fois les fichiers lus, il est possible de visualiser les informations suivantes :

- Les dimensions minimales de la machine.
- Le nombre total de points et de triangles de la machine.
- Le nombre d'organes lus.
- Le nombre total d'organes de la machine.

En plus de ces informations, l'utilisateur peut réorienter la machine selon les trois axes X, Y et Z avec les angles voulus.

Par la suite, il doit choisir le nom de l'organe courant et modifier sa visibilité. Si l'organe est mobile, l'utilisateur doit spécifier les axes de déplacement qui apparaissent en décochant « Fixe ». Cliquer sur le bouton « Modifier » pour que ces changements soient effectifs.

Il est aussi possible de visualiser :

- Tous les organes.
- Les organes mobiles seulement.
- Le brut de la machine pour confirmer que les informations sont correctes.
- Un organe, son brut et son origine après avoir choisi son indice.
- Le point CL de la broche.

À la fin, l'utilisateur clique sur le bouton « Positionner pièce » qui permet de déplacer la pièce et le trajet afin de les positionner au centre de la face supérieure de la table.

Figure 22. Onglet « Machine ».

### III.4 Onglet « Simulation »

Le quatrième onglet permet de simuler manuellement et automatiquement les mouvements de la machine complète avec la pièce et l'outil bien positionnés (Figure 7).

Au début, l'utilisateur peut déplacer manuellement les organes sur leurs axes de déplacement, puis les repositionner en cliquant sur « Réinitialiser ». Ensuite, dans la simulation manuelle, c'est l'utilisateur qui doit changer les positions sur le trajet d'outils à partir d'une liste déroulante (**ComboBox**), tout en visualisant l'outil en deux modes, la position et le trajet en cours qui se dessine à chaque avancement.

Pour la simulation automatique, l'utilisateur choisit la fréquence d'affichage et lance la simulation tout en visualisant le pourcentage d'avancement. À n'importe quel moment, l'utilisateur peut mettre la simulation en pause, la reprendre là où elle s'est arrêtée ou bien l'arrêter. Il peut aussi modifier dans les onglets précédemment mentionnés. Tout en visualisant l'outil, la position et le trajet en cours qui se dessine à chaque avancement, les coordonnées des points CL, les translations et les informations sur l'outil.

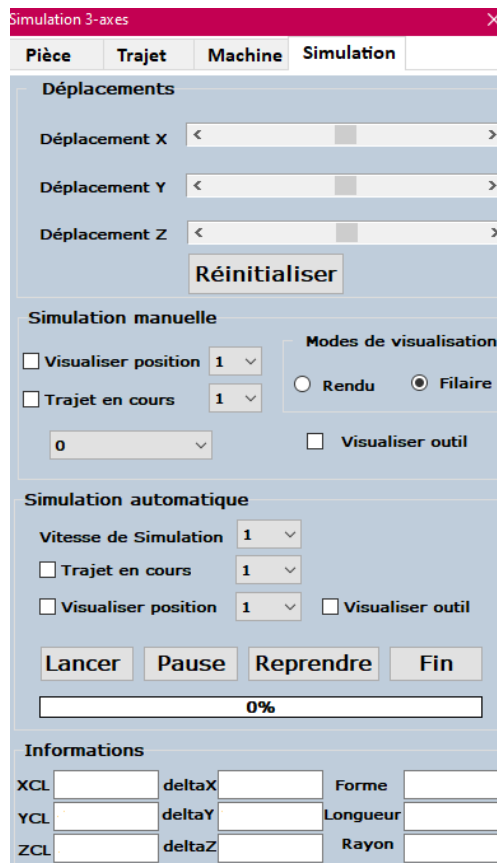


Figure 23. Onglet « Simulation ».

### III.5 Onglet « G-Code »

Le cinquième onglet permet de générer le fichier texte du « G-Code » et celui des outils utilisés (Figure 8).

L'utilisateur doit d'abord choisir un point origine parmi les cinq (05) origines proposés sur l'illustration, pour que la machine prenne ce point comme repère de la pièce. Les coordonnées de ce dernier sont affichées. Ensuite, il doit introduire les paramètres du programme : numéro du programme, incrément de lignes, vitesse de rotation de l'outil, sens de rotation et activation de l'arrosage. Dans le cas où l'utilisateur ne saisit pas ces informations, les paramètres par défaut seront appliqués.

Finalement, en cliquant sur le bouton « Fichier G-Code », la boîte de dialogue apparaît pour choisir l'emplacement, nommer et enregistrer le fichier « G-Code ». Cette dernière enregistre automatiquement dans un fichier de type « txt ». En cliquant par la suite sur le bouton « Fichier d'outils », le fichier des outils utilisés est généré.



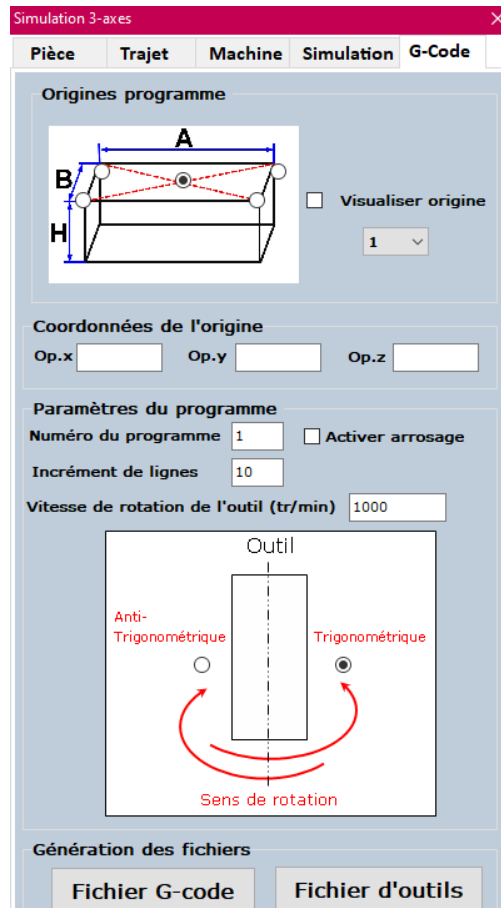


Figure 24. Onglet « G-Code ».

L'application gère aussi les cas où l'utilisateur essaie d'ouvrir des fichiers d'extensions inadéquates, saisit des valeurs incorrectes ou ne saisit aucun paramètre du « G-Code », en lui affichant un message.

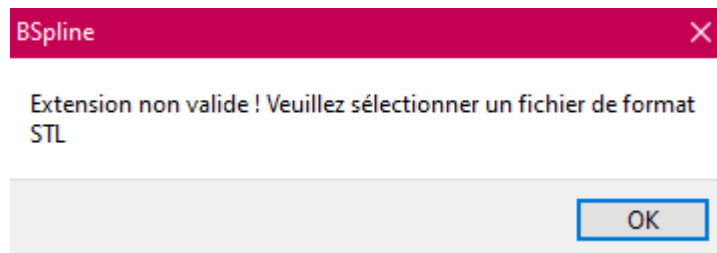


Figure 25. Ouverture d'un fichier STL non valide.

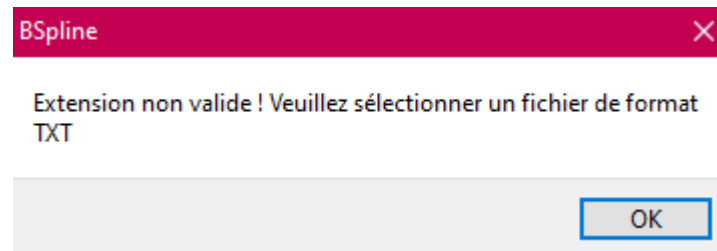


Figure 26. Ouverture d'un fichier trajet non valide.

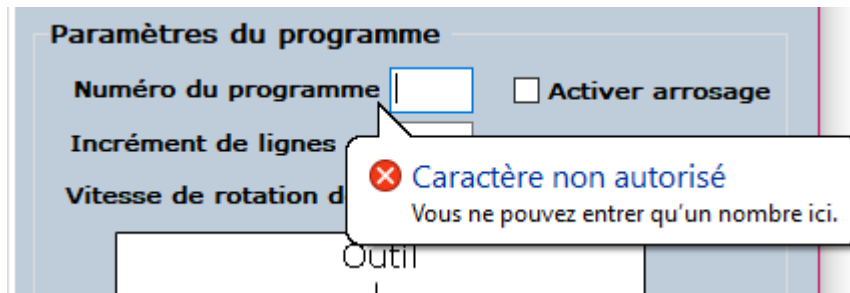


Figure 27. Saisi d'un caractère non autorisé.

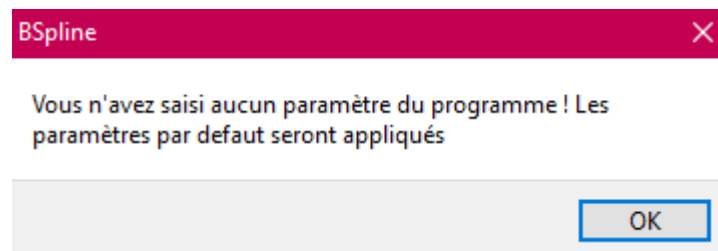


Figure 28. Génération d'un fichier « G-Code » sans saisi des paramètres.

## IV. Test et validation

Arrivé au terme de la présentation de l'application, la phase de validation du travail et les différents résultats auxquels nous avons abouti sont présentés dans la suite.

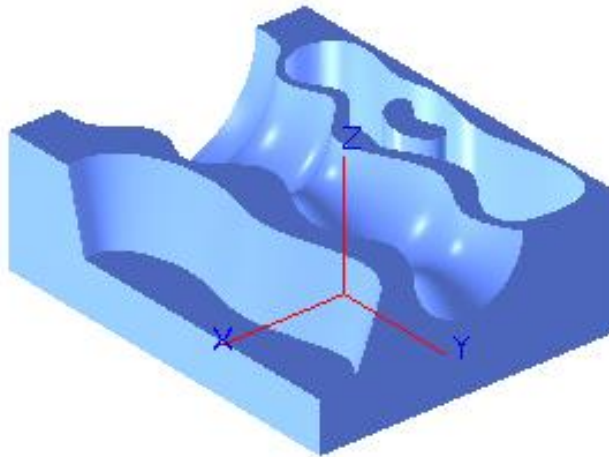
Le module logiciel que nous avons intégré a été testé sur un PC dont les caractéristiques sont les suivantes :

- Windows 10.
- Processeur : Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz 2.81 GHz.
- Mémoire installée (RAM) : 16.0 Go (15.9 Go utilisable).
- Type de système : système d'exploitation 64 bits, processeur x64.

La validation du module logiciel s'est faite sur plusieurs modèles STL de pièces et machines différents. La pièce donnée par la Figure 13 et deux modèles de machines sont pris comme support de validation dans la suite de ce travail.

La pièce est de forme très complexe, adéquate pour visualiser les mouvements des outils.

Le premier modèle de machine vise à montrer la simulation géométrique et la génération du fichier « G-Code » et le fichier d'outils. Cette machine représente six cas possibles de configurations des fraiseuses. Par contre, le deuxième modèle de machine présente le cas d'une table fixe et une broche qui se déplace suivant les trois (03) axes, très utilisé dans l'usinage des pièces lourdes.



**Figure 29.** Exemple de test.

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

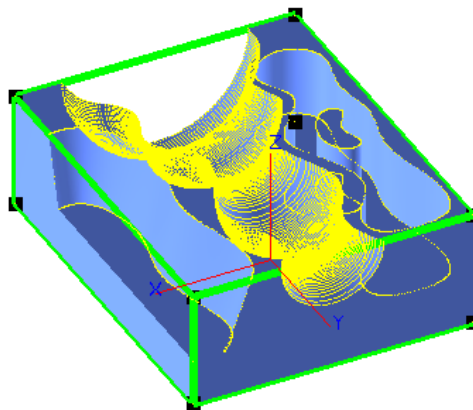
Xmin = -49,871 mm      Ymin = -77,904 mm      Zmin = 0,000 mm

Xmax = 70,129 mm      Ymax = 72,096 mm      Zmax = 50,000 mm

Longueur = 120,000 mm    Largeur = 150,000 mm    Hauteur = 50,000 mm

Nombre total de sommets = 23560      Nombre total de triangles = 47116

Les autres paramètres qui sont visualisés sont donnés par la Figure 14. Cette figure présente le brut en filaire et ses sommets ainsi que les sommets des triangles.



**Figure 30.** Paramètres géométriques du modèle STL.

- ❖ **Etape 2 :** lecture du fichier trajet qui contient des informations (forme et dimensions d'outils, déplacements, vitesse et mode de déplacement). Le nombre de points de passage obtenu est 66184 (Figure 15)

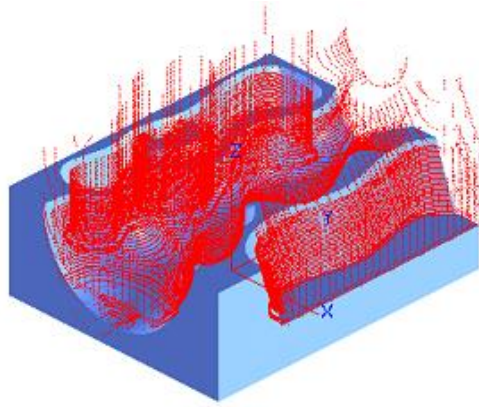


Figure 31. Points de passage du trajet d'outils.

#### IV.1 Premier modèle de machine

- ❖ **Etape 3 :** importation des fichiers STL des organes de la machine. Cette machine est composée de trois organes mobiles : une broche et deux tables (Figure 16).

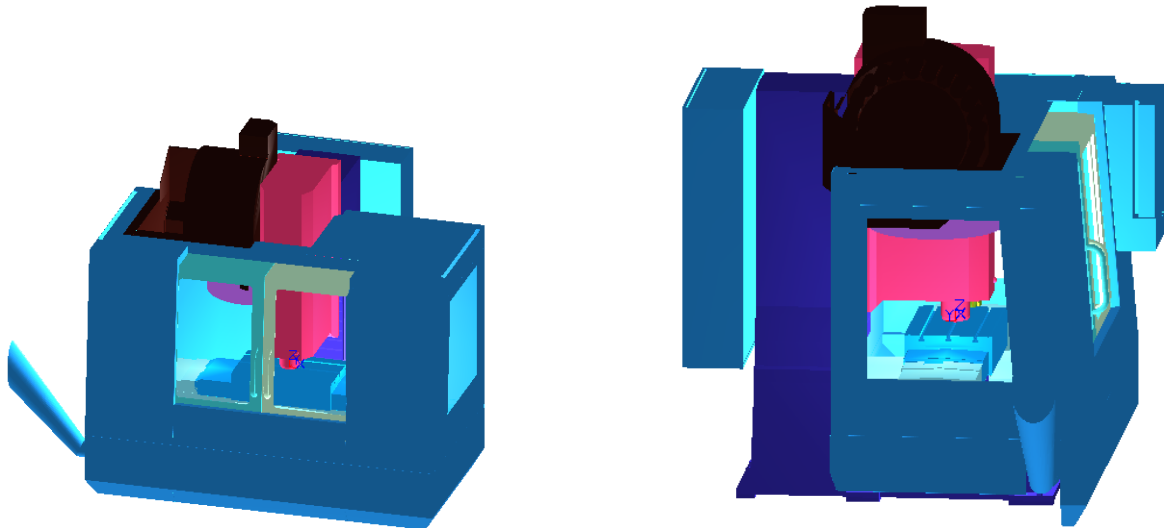


Figure 32. Machine complète.

Une fois les fichiers lus, il est possible de visualiser les informations suivantes :

Xmin = -11.127 mm      Ymin = -8.267 mm      Zmin = -7.412 mm

Xmax = 122.392 mm      Ymax = 90.941 mm      Zmax = 81.537 mm

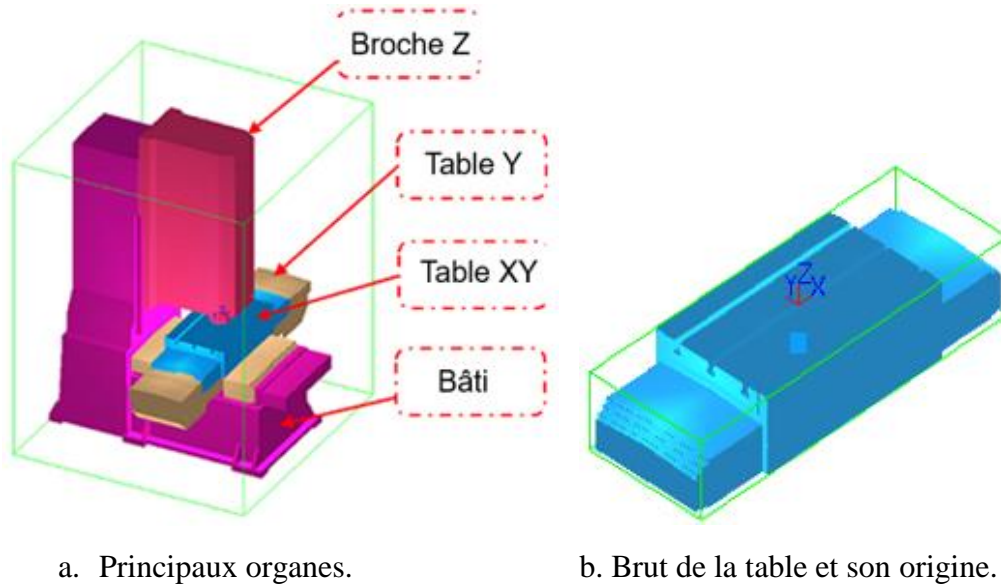
Longueur = 111.265 mm      Largeur = 82.673 mm      Hauteur = 74.125 mm

Nombre total de sommets = 18318      Nombre total de triangles = 36222

Nombre d'organes = 11

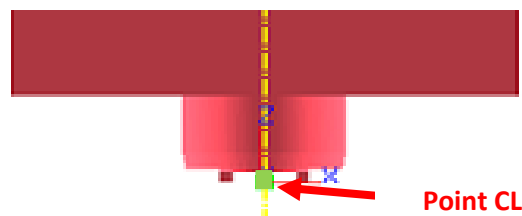
Pour s'assurer que les dimensions précédentes sont correctes, il suffit de cliquer sur « Brut machine ». L'utilisateur a aussi la possibilité de visualiser chaque organe en rendu, en filaire, son brut et son origine (Figure 17).

Cette première machine ne nécessite aucune rotation, donc nous passons directement au nommage des organes, spécification de leur visibilité et leurs déplacements. L'utilisateur peut visualiser la machine complète ou seulement les organes mobiles. À n'importe quel moment, il peut modifier la visibilité de chaque organe. Cette fonctionnalité est très efficace lors de la simulation.



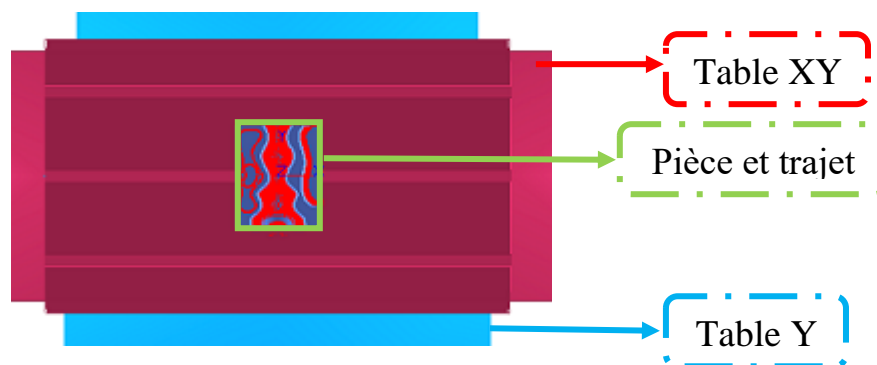
**Figure 33.** Principaux organes de la machine.

Pour la broche, le point CL (le point le plus bas) est calculé et visualisé. L'outil doit être fixé à cet emplacement (Figure 18).



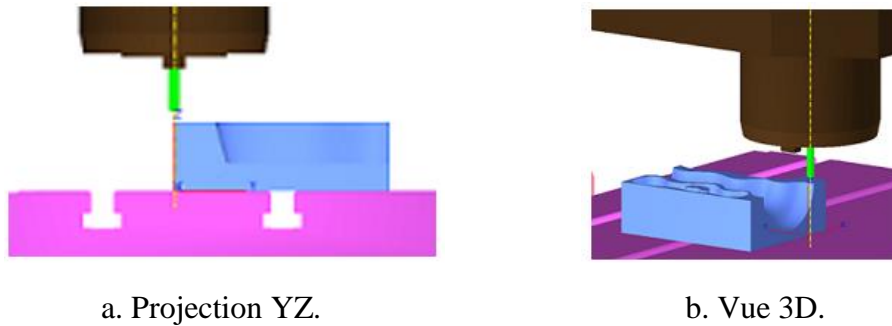
**Figure 34.** Point CL de la broche.

Puis, positionner la pièce et son trajet d'outils sur le centre de la table en appuyant sur le bouton requis (Figure 19).



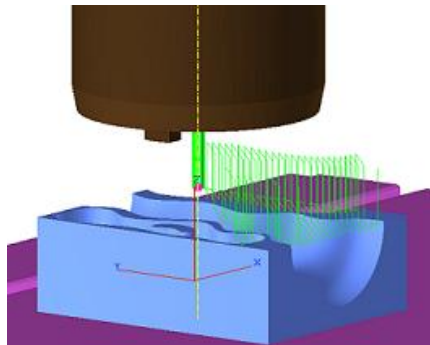
**Figure 35.** Positionnement de la pièce sur la table.

- ❖ **Etape 4 :** Simulation géométrique manuelle et automatique des mouvements des organes lors de l'usinage. Cette dernière permet à l'utilisateur de visualiser l'outil, la position actuelle, le trajet parcouru et les déplacements de la machine.

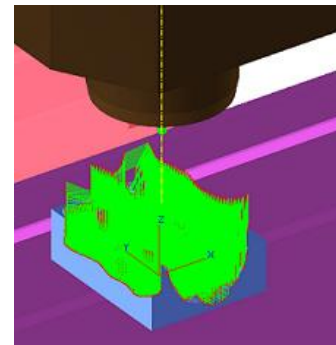


**Figure 36.** Début de la première simulation.

Dans cette étape, l'utilisateur confirme que l'outil est passé par tous les points du trajet, en restant fixé au point CL de la broche (Figure 20). Figure 21 et Figure 22 montrent la simulation du trajet d'outils en une position intermédiaire et à la fin du trajet respectivement.

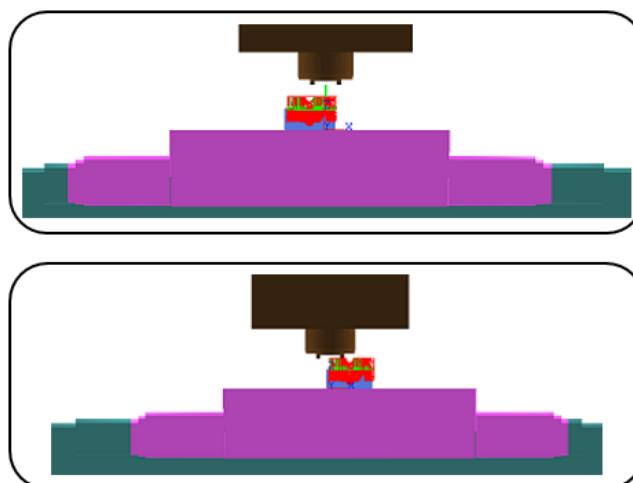


**Figure 37.** Première simulation en cours.



**Figure 38.** Fin de la première simulation.

Pendant la simulation, l'utilisateur visualise les mouvements des organes mobiles qui diffèrent pour chaque point de passage (Figure 23).



**Figure 39.** Emplacement des organes mobiles.

- ❖ **Etape 5 :** après avoir confirmé que la simulation est correcte, l'utilisateur passe à la dernière phase qui est la génération du fichier « G-Code » et du fichier d'outils (Figure 24).

```

Fichier  Edition  Format  Affichage  Aide
%1
N1 G40 G80 G17 G90 G71 G94 G97      N662031 G01 X-1.007 Y-71.985
N11 T1 M06                          N662041      Z30.6
N21 S1000 M3 M8                      N662051 G00 Z9.6
N31 G00 X34.544 Y-75 Z-28.9          N662051 M09 M05 M02
                                       XOFF
    
```

- a. Début.
- b. Fin.

**Figure 40.** Début et fin du premier fichier « G-Code ».

Le fichier des outils utilisés est optimisé puisqu'il ne contient aucune redondance. Il comporte quatre (04) outils différents (Figure 25).

```

Fichier  Edition  Format  Affichage  Aide
Liste des outils utilisés

Outil 1

Forme:cylindrique
Rayon:3
Longueur:30

Outil 2

Forme:cylindrique
Rayon:1
Longueur:10
    
```

**Figure 41.** Premier fichier d'outils.

## IV.2 Deuxième modèle de machine

Les deux premières étapes sont identiques à celles de la première machine.

- ❖ **Etape 3 :** importation des fichiers STL des organes de la machine. Ce deuxième modèle est composé de quatre principaux organes : une table fixe et trois organes mobiles. La broche se déplace sur les trois axes X, Y et Z.

Cette machine est mal orientée, donc elle nécessite une rotation de 90 degrés autour de l'axe X (Figure 26 et Figure 27).

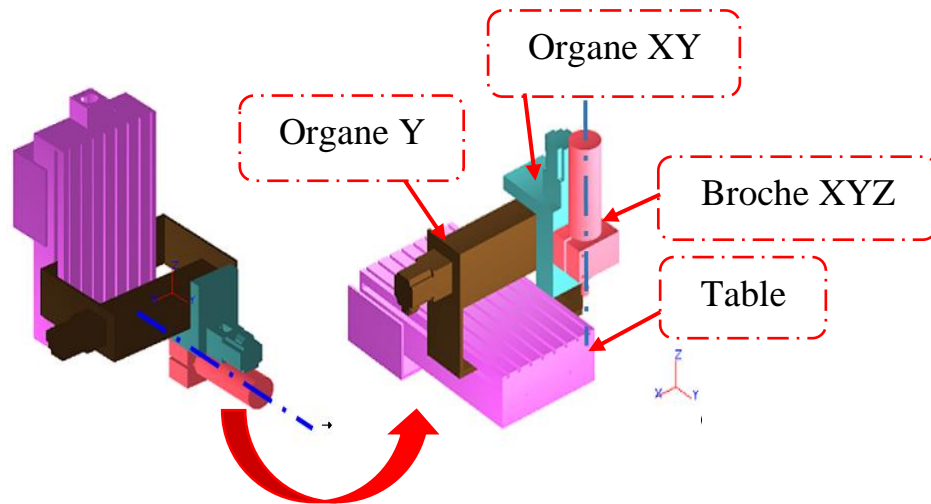


Figure 42. Réorientation de la machine.

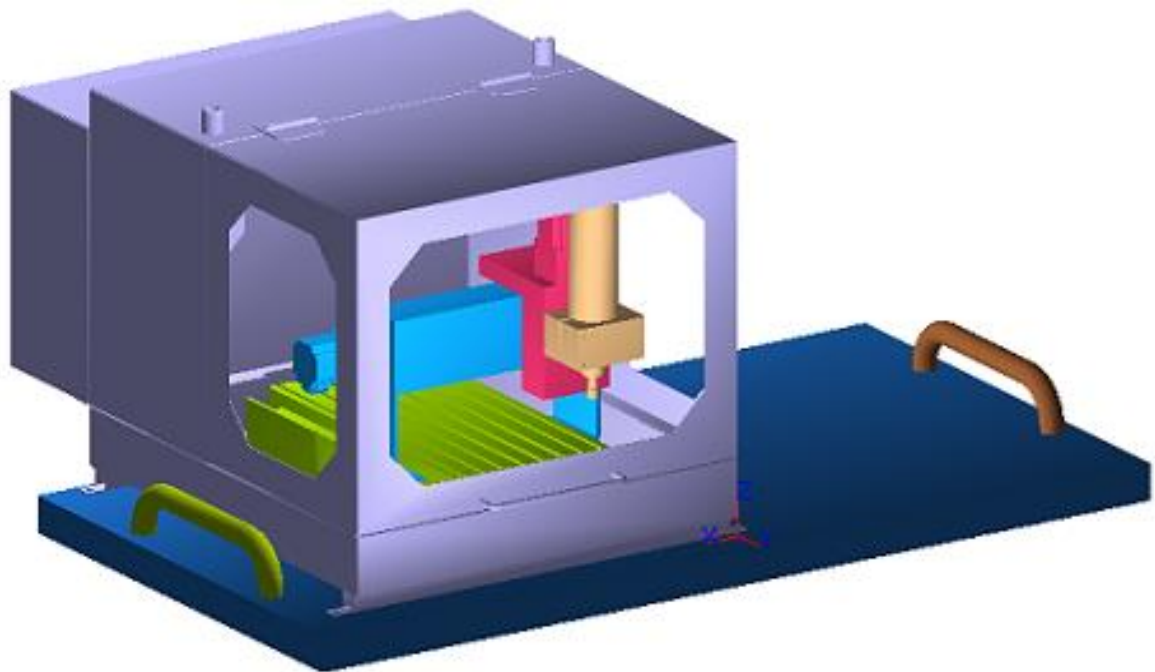


Figure 43. Machine complète réorientée.

Une fois les fichiers lus, les informations suivantes sont visualisées :

Xmin = -731.520 mm      Ymin = -820.364 mm      Zmin = -129.541 mm

Xmax = 792.480 mm      Ymax = 645.570 mm      Zmax = 906.719 mm

Longueur = 1270.000 mm      Largeur = 628.650 mm      Hauteur = 863.550 mm

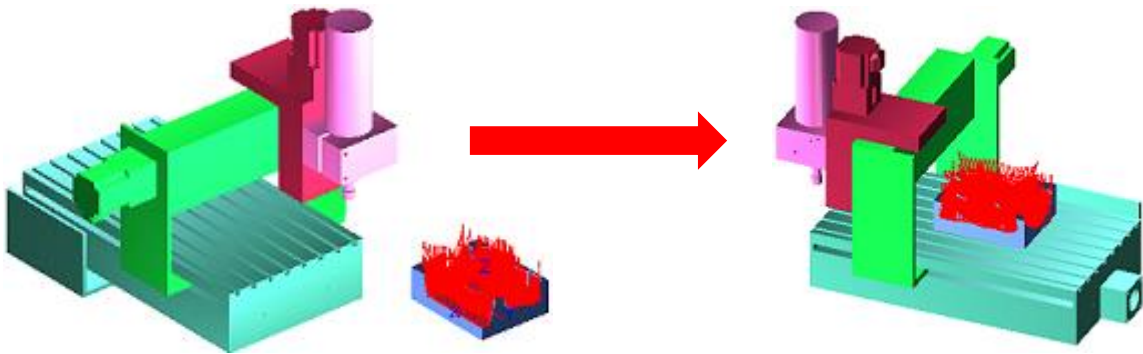
Nombre total de sommets = 42727      Nombre total de triangles = 85474

Nombre d'organes = 8

Les mêmes fonctionnalités de visualisation que la machine précédente sont possibles (brut machine, brut organe, organes mobiles, etc.).

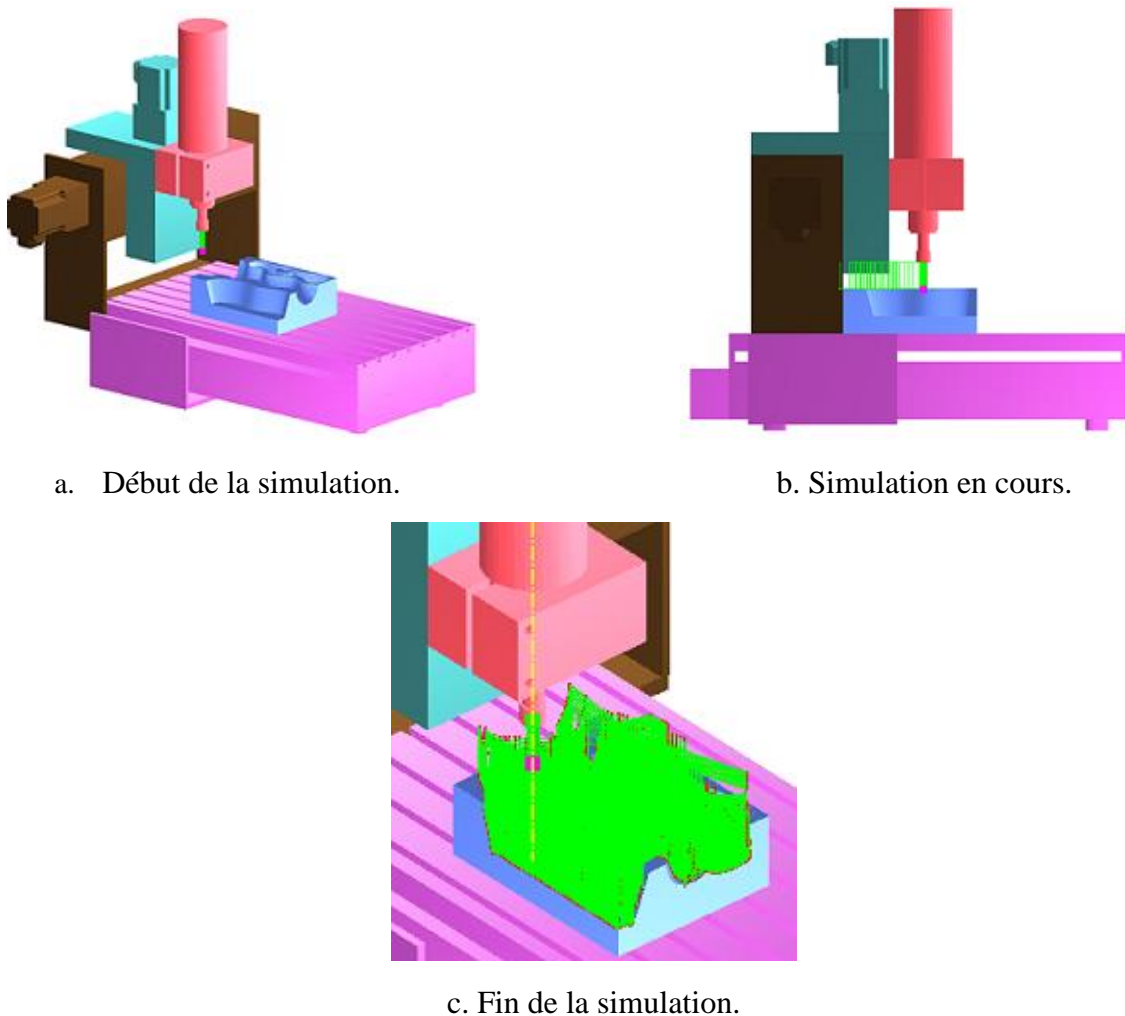


Après avoir nommé et spécifié les déplacements et la visibilité de chaque organe, l'utilisateur positionne la pièce sur la table (Figure 28).



**Figure 44.** Positionnement de la pièce sur la table.

- ❖ **Etape 4 :** Simulation géométrique manuelle et automatique des mouvements des organes lors de l'usinage (Figure 29).



**Figure 45.** Simulation de la deuxième machine.

Après avoir terminé la simulation, nous constatons que l'outil est bien positionné sur la broche et est passé par tous les points de passage du trajet.

- ❖ **Etape 5 :** après s'être assuré que la simulation est correcte, nous passons à la génération du « G-Code » (Figure 30) et le fichier d'outils (Figure 31).

Nous avons gardé les paramètres par défaut afin de comparer les résultats.

<pre style="margin: 0;">G-code machine 2 - Bloc-notes Fichier Edition Format Affichage Aide %1 N1 G40 G80 G17 G90 G71 G94 G97 N11 T1 M06 N21 S1000 M3 M8 N31 G00 X-34.544 Y75 Z-28.9</pre>	<pre style="margin: 0;">N662031 G01 X1.007 Y71.985 N662041      Z30.6 N662051 G00 Z9.6 N662051 M09 M05 M02 XOFF</pre>
--	---

b. Début.

b. Fin.

**Figure 46.** Début et fin du deuxième fichier « G-Code ».

Liste des outils utilisés

Outil 1

Forme: cylindrique  
Rayon: 3  
Longueur: 30

Outil 2

Forme: cylindrique  
Rayon: 1  
Longueur: 10

**Figure 47.** Deuxième fichier d'outils.

<pre style="margin: 0;">G-code machine 1 - Bloc-notes Fichier Edition Form* Affichage Aide %1 N1 G40 G80 G17 G90 G71 G94 G97 N11 T1 M06 N21 S1000 M3 M8 N31 G00 X34.544 Y-75 Z-28.9 N31      Z-7.9 N41 G01 Z1.1 F6 N51      X35.957 Y 73.587 N61      X35.791 Y-69.511</pre>	<pre style="margin: 0;">G-code machine 2 - Bloc-notes Fichier Edition Format Affichage Aide %1 N1 G40 G80 G17 G90 G71 G94 G97 N11 T1 M06 N21 S1000 M3 M8 N31 G00 X-34.544 Y75 Z-28.9 N31      Z-7.9 N41 G01 Z1.1 F6 N51      X-35.957 Y73.587 N61      X-35.79 Y69.511</pre>
--	--

**Figure 48.** Comparaison entre les fichiers G-Code.

D'après l'illustration précédente nous constatons que les coordonnées des points changent (Figure 32), c'est-à-dire le « G-Code » change en fonction de la géométrie de la machine.

Par contre les fichiers d'outils utilisés sont identiques, vu qu'ils ne dépendent pas de la géométrie de la machine (Figure 33).

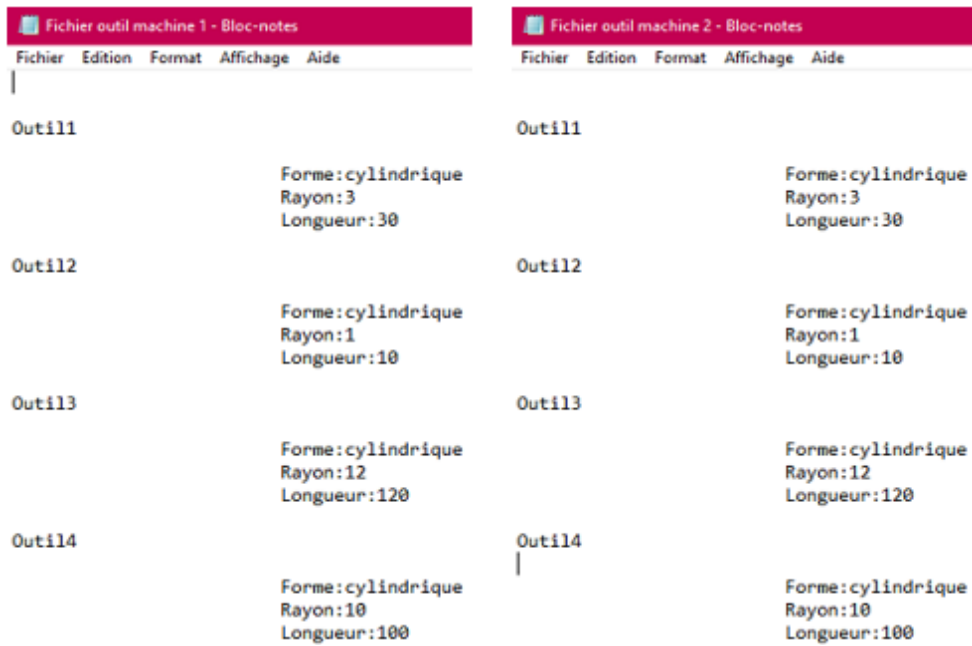


Figure 49. Comparaison entre les fichiers d’outils.

## Conclusion

Au cours de ce chapitre, nous avons présenté notre application et les résultats de test obtenus. Dans un premier temps, nous avons présenté le langage de programmation utilisé, l’environnement de développement de notre application ainsi que les interfaces pour décrire les différentes fonctionnalités. Dans un second temps, nous avons testé les différentes fonctionnalités de notre application à travers un exemple de pièce complexe et deux modèles de machines, depuis la lecture des paramètres STL jusqu’à la génération du fichier « G-Code » et le fichier d’outils. D’après les résultats obtenus nous pouvons conclure qu’il est possible de simuler n’importe quelle fraiseuse numérique à 03-axes et générer les fichiers « G-Code » et outils utilisés.

## CONCLUSION GENERALE

Ce mémoire est le résultat de notre projet de fin d'études mené au niveau de l'équipe « CFAO » de la Division Productique et Robotique du « CDTA ». Le travail réalisé consiste à concevoir et à implémenter un module logiciel graphique et interactif permettant la simulation **géométrique** des mouvements de fraiseuses numériques à 03-axes, pour toutes les configurations possibles, à partir du trajet d'outils et de la pièce complexe définie par son modèle STL.

Lors de la réalisation de ce projet, nous avons commencé par une étude bibliographique sur le processus de production des pièces mécaniques et les modèles de représentation des surfaces complexes tout en focalisant notre analyse sur les surfaces continues. Ensuite, nous avons mené une étude sur les différents formats d'échange de données et le modèle STL en particulier. Par la suite, nous avons présenté les machines-outils à commande numérique et le processus de simulation. Cette étude a été complétée par une étude conceptuelle de l'application et une implémentation informatique des solutions proposées. La réalisation de cette approche a été menée sous Windows en utilisant le langage de programmation « C++ », l'environnement de développement « Embarcadero C++ Builder Seattle » et la bibliothèque graphique « OpenGL ». À la fin, l'approche a été validée sur deux configurations de fraiseuses 3-axes.

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

- ✓ Récupération des principaux paramètres du modèle de la pièce.
- ✓ Récupération des principaux paramètres du trajet d'outils.
- ✓ Récupération des principaux paramètres de la machine.
- ✓ Changement d'orientation de la machine.
- ✓ Configuration de la machine.
- ✓ Positionnement de la pièce et du trajet.
- ✓ Positionnement des organes mobiles, de la pièce, du trajet et des outils utilisés pour chaque position du trajet d'outils.
- ✓ Simulation automatique des mouvements des organes mobiles de la fraiseuse.
- ✓ Génération du fichier « G-Code » optimisé du trajet d'outils.
- ✓ Génération du fichier optimisé des outils utilisés lors de l'usinage.

En perspectives, nous recommandons de traiter les thématiques suivantes :

- Développement d'une approche permettant la détection des collisions entre l'outil de l'usinage et l'environnement d'usinage.
- Modélisation des différentes configurations de la fraiseuse selon le modèle cinématique et le modèle dynamique.
- Extension de l'approche proposée aux fraiseuses numériques 04-axes et 05-axes.
- Automatisation de l'orientation de la machine.
- Génération du « G-Code » en 05-axes.
- Correction du trajet d'outils.

# Références bibliographiques

- [1]. B. Bloul, B. Aour, A. Bourdim, R. Harhout, « Modélisation de l'état de surface d'une pièce rectifiée en utilisant ANN », 31er Congrès International sur les Ingénieries Civile, Mécanique et Electrique pour l'Energie, CMEEE 2015- Marrakech.
- [2]. <https://blog.topsolid.fr/programmation-pieces-complexes>. [Consulté le 01/12/2019]
- [3]. Sonia Djebali, « Optimisation Globale du Processus d'Usinage des Surfaces Gauches », Thèse de Doctorat, Université de Toulouse 3 Paul Sabatier, 2014.
- [4]. Christophe Tournier, « Contribution à la conception des formes complexes : La surface d'usinage en fraisage 5 axes isocrête », Thèse de Doctorat, Laboratoire Universitaire de Recherche en Production Automatisée Ecole Normale Supérieure de CACHAN, 2001.
- [5]. « C.F.A.O. (conception et fabrication assistées par ordinateur) ». In Universalis Junior [en ligne]. Encyclopædia Universalis, consulté le 3 décembre 2019. Disponible sur <http://junior.universalis.fr/encyclopedie/c-f-a-o-conception-et-fabrication-assistees-par-ordinateur/>.
- [6]. Raphaël La Greca, « Approche déclarative de la modélisation de surfaces. Interface homme-machine [cs.HC] ». Université de la Méditerranée - Aix-Marseille II, 2005.
- [7]. M. Mahiddini, « Echange des données entre systèmes de CFAO : application à la conversion par approximation des courbes et surfaces de Bézier de degré élevé », Mémoire fin d'études, Département Génie mécanique, Université M'hamed Bougara Boumerdes 2006.
- [8]. <https://all3dp.com/what-is-stl-file-format-extension-3d-printing> [consulté le 6 Février 2020].
- [9]. Y. Bendjerad, « Impression 3D : Vers une personnalisation massive », Mémoire fin d'études, Département Mathématique et informatique, Université Abdelhamid Ibn Badis Mostaganem 2016.
- [10]. <https://www.digicad.fr/> [consulté le 6 Février 2020].
- [11]. 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.
- [12]. 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.
- [13]. Gilles Prod'Homme, « Commande Numérique des Machines-Outils », Technique de l'Ingénieur, Traité Génie Mécanique, B 7130, pp.3, 1997.
- [14]. D. Veeramani, Y. Gau, « Technologie de la Machine-outil », éd. CIP Tram élan, 1997.
- [15]. <https://www.usimm.ca/quels-sont-les-differents-types-de-fraiseuses/> [consulté le 29 Février 2020].
- [16]. M. Mouzoui, Y. Tazamoucht, « Réalisation et automatisation d'une machine à commande numérique », Mémoire fin d'études, Département de Génie Électrique, Université A.MIRA-BEJAIA, 2018.
- [17]. S. Bensaada, « Coupe des métaux ». Cours, Université Mohamed Khider, Biskra.
- [18]. Loudjedi Fatma Zohra, Zahra Fatma Zahra, « Optimisation du processus de fabrication par simulation de l'outil de production intégrant le couplet « modèle géométrique pièce /modèle cinématique de MOCN » », Mémoire fin d'études, département d'Informatique, Université Saad Dahleb-Blida ,2004-2005
- [19]. B. Aguenini, H. Messaoudi, « Simulation et vérification de l'opération d'ébauchage des surfaces gauches sur des fraiseuses à commande numérique à 3-axes » Mémoire fin d'études, Département d'Informatique, Université Saad Dahlab, Blida, 2005/2006.
- [20]. Emmanuel Duc, Emmanuel Lefur, « Machines-outils à commande numérique, structure, modélisation et réglage ». Mémoire de DEA à l'école normale supérieur de Cachan, France, septembre 1997.
- [21]. BETTINE FARID, « Etude de processus d'usinage des pièces mécaniques sur les machines cnc à 5-axes ». Mémoire de Magister en : Génie Mécanique, Département de Génie Mécanique, Université Mohamed Khider – Biskra, 2012.
- [22]. M. Bey, H. Bendifallah, M. Dermache, N. Abdelhamid, « Finition des surfaces gauches par combinaison de différentes formes et dimensions d'outils ». Huitièmes Journées de

Mécanique, JM'08, 10 au 11 Avril 2012, Ecole Militaire Polytechnique « EMP », Alger, Algérie.

[23]. K. Bouhadja, M. Bey, «Classification of simulation methods in machining on multi-axis machines », The 2014 International Conference of Manufacturing Engineering and Engineering Management, ICMEEM'14, 02 au 04 Juillet 2014, Londres, Angleterre, pp 992-997.

[24]. <https://www.dalgo.fr/techniques-usinage/> [consulté le 07 août 2020].

[25]. <https://www.embarcadero.com/> [consulté le 07 août 2020].





# **Annexes**

## Annexe 1 : G-Code

C'est le langage de programmation des machines-outils. Un programme G-Code est composé d'un ensemble de blocs et chaque bloc est composé de mots.

### Fonctions modales

Une fonction est dite modale lorsqu'elle reste active jusqu'à sa révocation.

<b>Code</b>	<b>Désignation</b>
<b>G00</b>	Interpolation linéaire à vitesse rapide.
<b>G01</b>	Interpolation linéaire à vitesse d'avance programmée.
<b>G17</b>	Choix du plan XY.
<b>G20/G70</b>	Entrée en pouces.
<b>G21/G71</b>	Entrée en mm.
<b>G40</b>	Annulation de correction de rayon.
<b>G90</b>	Programmation absolue par rapport à l'origine programme.
<b>G91</b>	Programmation relative par rapport au point de départ du bloc.
<b>G94</b>	Vitesse d'avance exprimée en millimètre, pouce ou degré par minute.
<b>G95</b>	Vitesse d'avance exprimée en millimètre ou en pouce par tour.

### Fonctions non modales

Une fonction est dite non modale si elle reste active seulement dans une ligne.

<b>Code</b>	<b>Désignation</b>
<b>M0</b>	Arrêt programmé.
<b>M2</b>	Fin de programme.
<b>M3</b>	Rotation de broche sens anti trigonométrique.
<b>M4</b>	Rotation de broche sens trigonométrique.

## Annexes

<b>M5</b>	Arrêt de broche.
<b>M6</b>	Appel d'outil.
<b>M8</b>	Arrosage numéro 1.
<b>M9</b>	Arrêt d'arrosage.
<b>M30</b>	Fin de programme.

## Autres fonctions

<b>Code</b>	<b>Désignation</b>
<b>T</b>	Numéro d'outil.
<b>D</b>	Appel du correcteur d'outil.
<b>S</b>	Vitesse de la broche.
<b>F</b>	Vitesse d'avance.
<b>N</b>	Numéro du bloc.
<b>X</b>	Coordonnée X.
<b>Y</b>	Coordonnée Y.
<b>Z</b>	Coordonnée Z.

## Annexe 2 : Fraiseuse numérique à 03-axes

Les machines-outils utilisées lors de la simulation sont les suivantes :

### Première machine

Pour la configuration de cette machine, la table se déplace suivant X et Y tandis que la broche se déplace suivant Z.

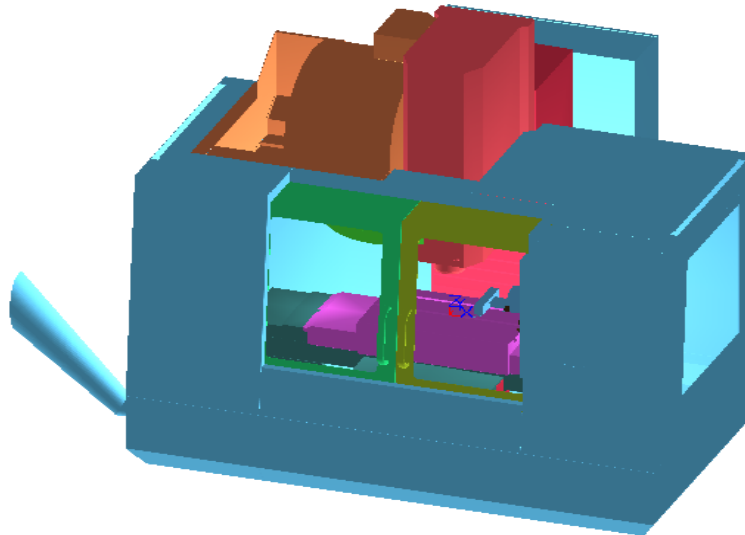


Figure 1. Machine 1 complète.

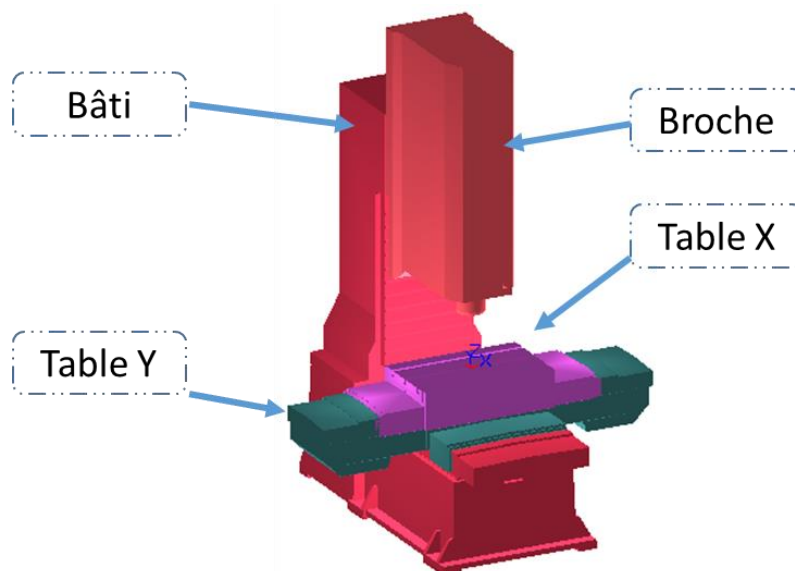


Figure 2. Principaux organes de la machine 1.

## Deuxième machine

Pour la configuration de cette machine, la table est fixe et la broche se déplace suivant les axes X, Y et Z.

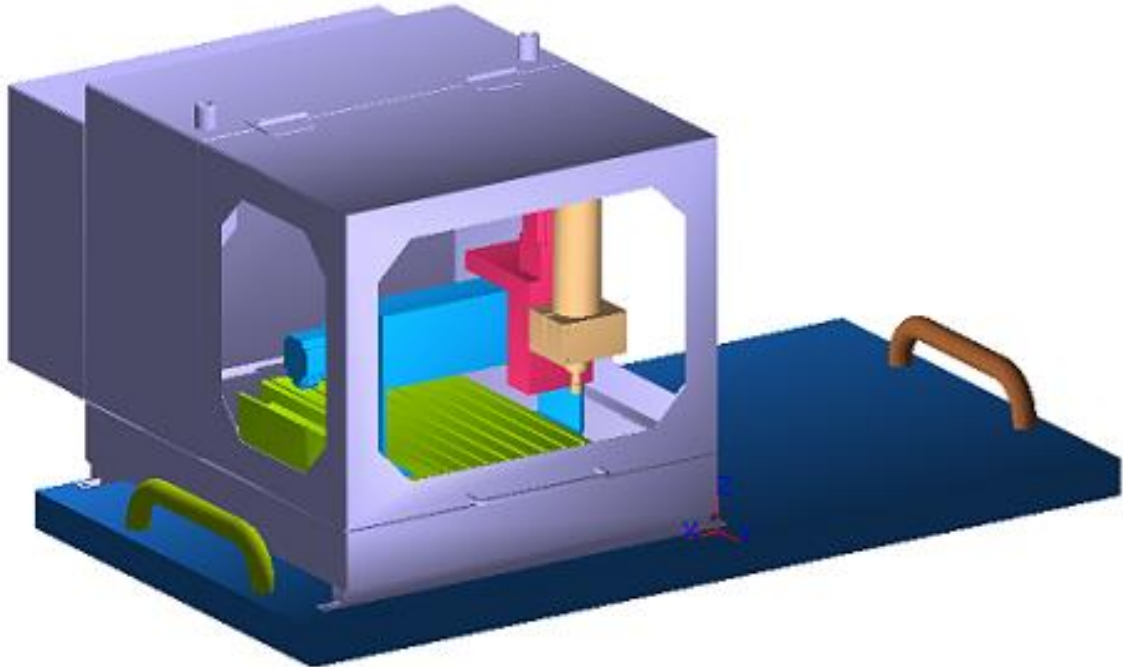


Figure 3. Machine 2 complète.

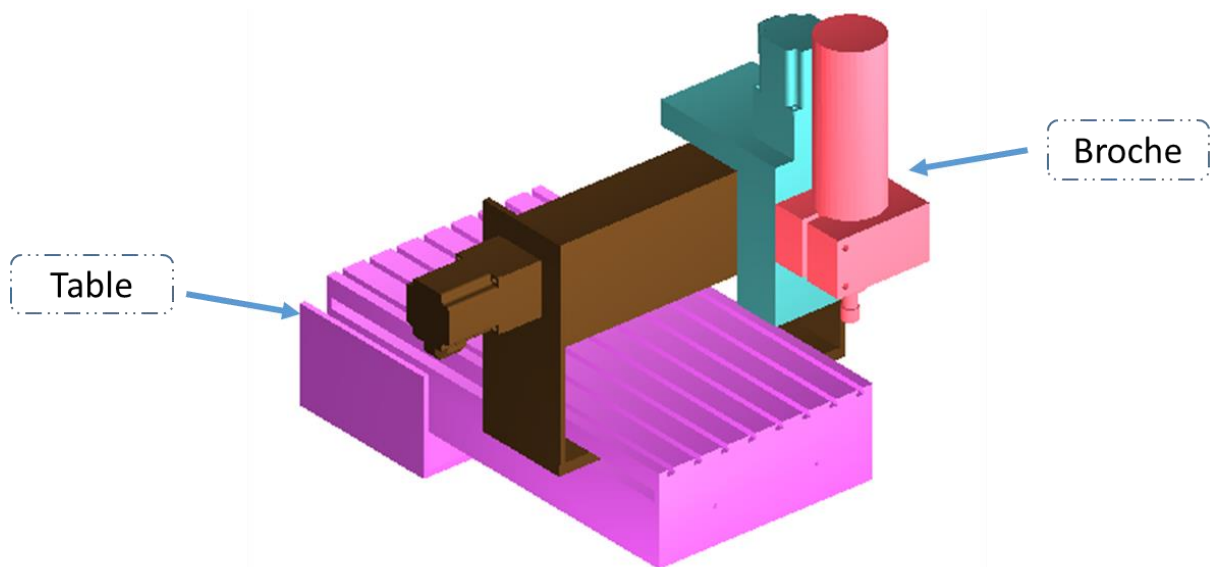
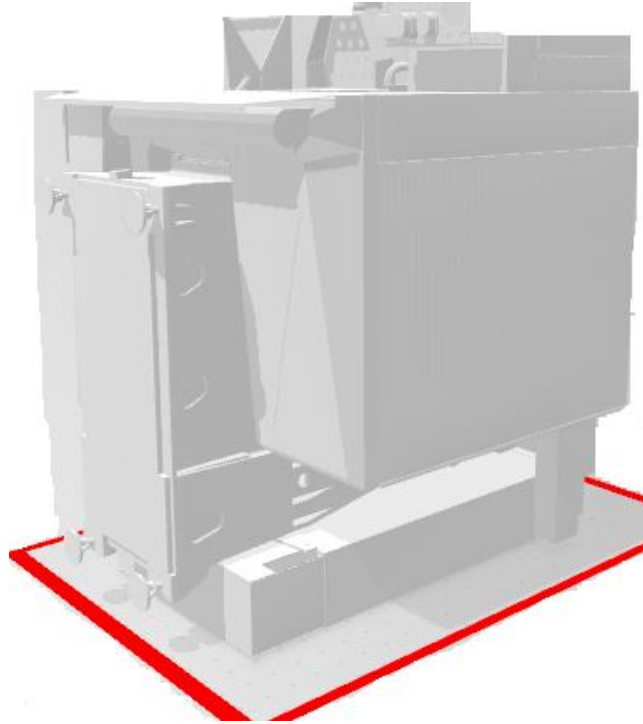


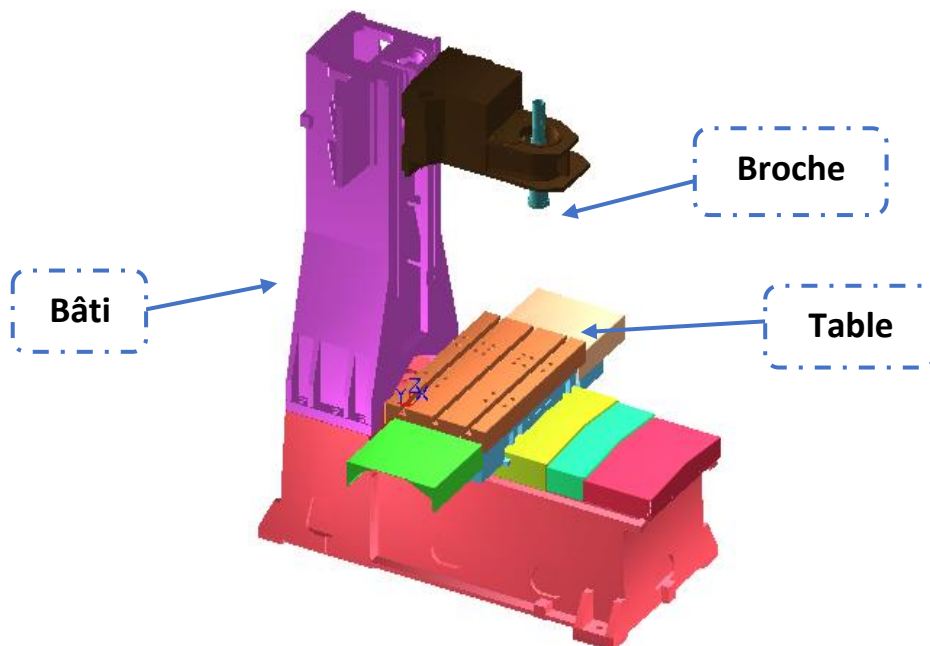
Figure 4. Principaux organes de la machine 2.

### Troisième machine

Pour la configuration de cette machine, la table se déplace suivant X et Y tandis que la broche se déplace suivant Z.



**Figure 5.** Machine 3 complète.



**Figure 6.** Principaux organes de la machine 3.