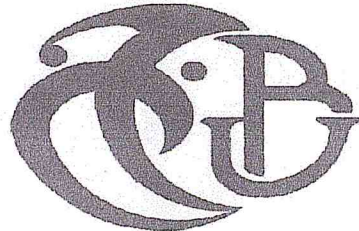


République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida 1

N° D'ordre :



Faculté des sciences

Département d'informatique

Mémoire Présenté par :

Kheris Lilia Hedroug Wassyla

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique
Spécialité : Informatique
Option : Ingénierie de logiciel

Sujet :

***Conception et Développement d'une Application de Simulation
d'Enlèvement de Matière Lors de la Finition des Surfaces Complexes sur
Fraiseuses 05-axes***

Soutenu le : 25/06/2016

N.Toubaline
S.Ferfera
A.Kameche
K.Bouhadja

Président
Examineur
Promoteur
Encadreur

Promotion
2015 / 2016

Dédicaces

Je dédie ce travail à mes chers parents

Kheris Boualem et Hadj Ben Fetima Hinda,

En reconnaissance à leurs sacrifices,

Leurs soutiens et leurs prières quotidiennes.

A ma grand-mère Djohar qui ne m'a pas oublié

Pendant la réalisation de ce travail.

A mes chères sœurs Bouthaina, Saoussen et Lamis

Et cher frère Abdelillah.

A mon amie Amina El Arbi Aissa.

A tous les membres de famille Kheris et Hadj Ben Fetima.

KHERIS LILIA

Dédicaces

A mes chers parents Hadroug Farid et Benmouhamed Haciba

En reconnaissance à leurs sacrifices

Et leurs prières quotidiennes,

A mes chères sœurs

« Nesrine, Ghita Sarah »

A mon chère amie Asma Feddous

A tous les membres de famille Hedroug

Et Benmohammed

HEDROUG WASSYLA

Remerciements

Louanges à Dieu le tout Puissant l'Omnipotent qui nous a fait la faveur de nous donner le courage, la force et la volonté pour mener à bien ce modeste travail.

Nous exprimons notre très profonde gratitude à tous ceux qui nous ont transmis leur savoir en espérant mériter et être à la hauteur de leurs sacrifices consentis à la noble mission d'éduquer tant de générations.

Nous adressons notre très grand respect aux membres du jury qui nous font l'honneur de juger notre travail.

Nous remercions en particulier notre promoteur monsieur KAMACHE Abdallah Hicham et notre encadreur madame BOUHADJA Khadidja et l'équipes CFAO du CDTA qui ne se sont pas ménagés en étant toujours disponibles et à l'écoute de nos préoccupations. Sans leur assistance, leurs orientations et leurs conseils ; ce travail n'aurait pas vu le jour.

Merci infiniment à nos chers parents, frères et sœurs pour leurs écoutes, leurs aides et leurs soutiens durant toute l'année. C'est grâce à eux que nous avons pu bien terminer et avec succès ce travail.

Merci à nos amis qui nous ont soutenus, en particulier Moulay Hamid et Mansouri Khaled.

Merci à tous ceux qui de près ou de loin ont contribué à l'aboutissement de ce projet.

ملخص:

تطوير تقنيات المحاكاة بالقطع على آلات متعددة المحاور (03 محاور و05 محاور) هو المفتاح لتطور الجودة والإنتاجية في صناعة الأجزاء الميكانيكية ذات الأشكال المعقدة التي غالبا ما تكون موجودة في الصناعة (السيارات، والطيران، والقوالب، ... إلخ). يجب أن تلبي هذه القطع المتطلبات الوظيفية و / أو الإبداعية التي تحتاج إلى عناية خاصة في مرحلة الإنتاج. يتركز موضوع دراستنا على تصميم وتنفيذ تطبيق لمحاكاة عملية نزع المادة عن طريق الحفر ذات الخمس محاور لقطع ذات شكل معقد ابتداء من النماذج الحجمية (الدكسل الثلاثي). عملنا في المحاكاة يسمح للصانعين التحقق من التسامح الصناعي لتحقيق حالة سطح مرضية وتلبية الاحتياجات الوظيفية للقطع المصنوعة للربح في الإنتاجية والوقت. يركز عملنا على التمثيل الحجمي باستخدام نموذج الدكسل الثلاثي، و K-مميز لتحسين وقت الحساب في محاكاة عملية نزع المادة و توقع نوعية السطح النهائي.

الكلمات المفتاحية: الدكسل الثلاثي، الحفر ذات الخمس محاور، المحاكاة، سي أف آ أو

Résumé :

Le développement des techniques de simulation d'usinage sur machines multi-axes (03 axes et 05 axes) est la clé de l'évolution de la qualité et de la productivité dans la fabrication de pièces mécaniques de formes complexes, qui sont souvent présents dans l'industrie (automobile, aéronautique, moules, ect.). Ces pièces doivent satisfaire aux exigences fonctionnelles et / ou de style, qui ont besoin d'une attention particulière dans leur phase de production. La thématique de notre étude est axée sur la conception et l'implémentation d'une application de la simulation d'enlèvement de matière au stade finition en usinage 5axes des pièces complexes à partir de leur représentation volumique en Triple Dixel. Notre travail se concentre sur la représentation volumique de la pièce à usiner de forme gauche d'une manière précise et rapide en utilisant le modèle Triple Dixel, et le K-means pour optimiser le temps de calcul dans la simulation d'enlèvement de matière pour la prédiction de la qualité de la surface finie.

Mots-clés : Triple-Dixel, usinage 5-axes, simulation, CFAO.

Abstract :

The development of Machining simulation techniques on multi-axis machines (03 axes and 05 axes) is the key of quality and productivity evolution in the manufacture of mechanical parts with complex shapes, which are often present in the industry (automobile, aerospace, mold, ect.). These parts must meet the requirements functional and / or style, which need special attention in their production phase. The aim of our study is to design and implement an application for material removal simulation in the finition stage in 5axe machining for complex pieces using their volumetric representation (Triple Dixel). Our work focuses on the workpiece volume representation with a complex geometric form in a precise and rapid manner by using the Triple Dixel model from STL model, and the K-means algorithm to optimize the calculation time in the simulation of material removal for the prediction of the finished surface quality.

Key-words : Triple-Dixel, 5-axes machining, simulation, CFAO.

Sommaire

Introduction générale:	1
------------------------------	---

CHAPITRE 1: généralité sur le processus de fabrication de formes complexes

1. Introduction:.....	4
2. Conception et fabrication assisté par ordinateur(CFAO) :	4
2.1. Définition :.....	4
2.2. Conception assistée par ordinateur(CAO) :.....	5
2.3. Fabrication assistée par ordinateur(FAO) :	5
2.4. Processus de CFAO :.....	5
2.4.1. Construction du modèle géométrique :	6
2.4.2. Génération de trajectoire d'outils :.....	8
2.4.3. Validation et simulation :	9
2.4.4. Post-Processeur :	9
3. Fraiseuse à commande numérique :	10
3.1. Les axes de déplacement :	10
3.2. Fraiseuse 3 et 5 axes :	11
4. Génération de trajectoire d'outil :	12
4.1. Définition d'usinage :	12
4.2. Différentes étapes de production :	12
4.2.1. Ébauche :	12
4.2.2. Demi-finition :	13
4.2.3. Finition :	13
4.3. Outils utilisés en finition :	14
4.4. Positionnement de l'outil relativement à la surface à usiner :	14
4.5. Paramètres de calcul de trajectoire d'outil en fraisage en bout :	15
4.5.1. Les paramètres de guidage de l'outil :	15
4.5.2. Paramètre de passe de l'outil :	16
4.6. Stratégies d'usinage des surfaces gauches en finition :	17
4.6.1. Plans parallèles :	17

4.6.2.	Z-constant :	18
5.	simulation d'usinage des formes gauches sur les machines multiaxes :	19
5.1.	Simulation géométrique a l'échelle macroscopique :	21
5.1.1.	Mode filaire :	21
5.1.2.	Mode à base solide :	21
5.1.3.	Modèle à base objet :	21
5.1.4.	Modèle à base espace image :	24
5.2.	Choix de la technique adoptée :	24
5.2.1.	Voxel :	24
5.2.2.	Simple Dixel :	25
5.2.3.	Triple Dixel :	26
6.	Analyse :	28
7.	Conclusion :	29

CHAPITRE 2: conception

1.	Introduction :	31
2.	Analyse des besoins :	31
2.1.	Présentation du sujet :	31
2.2.	Problématique :	31
2.3.	Objectifs du travail :	32
2.4.	Solution proposée de la problématique :	32
2.4.1.	Diagramme d'états transition :	34
2.4.2.	« Lecture de la pièce en Triple-Dexels » :	34
2.4.3.	« Ajout d'une Surépaisseur aux segments »:	37
2.4.4.	« Homogénéisation des segments » :	37
2.4.5.	« Classification des segments » :	40
2.4.6.	Trajectoire d'outil :	43
2.4.7.	« Simulation de l'usinage »:	45
3.	Modélisation de l'application avec UML :	53
3.1.	Diagramme de cas d'utilisation :	53
3.1.1.	Diagramme de cas d'utilisation général :	54
3.1.2.	Diagramme de cas d'utilisation : Acquisition du modèle volumique et la trajectoire d'outil de la pièce:	54

3.1.3.	Diagramme de cas d'utilisation : simulation d'enlèvement de matière :.....	55
3.1.4.	Diagramme de cas d'utilisation : le calcul des erreurs d'usinage:.....	56
3.2.	diagramme de séquence :.....	56
3.2.1.	Récupération et structuration des Triple-Dexels :.....	57
3.2.2.	Récupérer et structurer les points centre d'outil de la trajectoire :	57
3.2.3.	Simulation d'enlèvement de matière :.....	58
4.	Conclusion:.....	59

CHAPITRE 3: implémentation

1.	Introduction :.....	61
2.	Diagramme de paquetage :	61
2.1.	Paquetage Triple Dixel :	61
2.2.	Paquetage Trajectoire d'outil :	63
2.3.	Paquetage simulation d'enlèvement de matière :.....	65
2.3.1.	Diagramme de classes :.....	65
3.	Présentation de l'application :	70
3.1.	Fenêtre principale :.....	70
3.2.	Onglet « Pièce » :	70
3.3.	Onglet Surépaisseur :.....	72
3.4.	Onglet Optimisation :	73
3.5.	Onglet Simulation :.....	76
3.6.	Onglet résultats et statistique :.....	80
3.7.	Test paramétrique :.....	82
4.	Conclusion :	83
	Conclusion générale:.....	84

Liste des figures

CHAPITRE 1

Figure 1: processus de fabrication d'une pièce mécanique [BEA, 13].	6
Figure 2: Flux d'information sur le système de CAO-FAO découplé ou Interfacé [BOU, 13].	7
Figure 3 : Modèle STL d'une surface théorique [BCK, 11].	7
Figure 4: Fichier STL et paramètres d'un triangle [BOU, 13].	8
Figure 5 : Orientation des axes [BOU, 13].	10
Figure 6: comparaison entre l'usinage à trois axes et l'usinage à cinq axes [ZEF, 07].	11
Figure 7: principe et géométries associées à l'étape d'ébauche	13
Figure 8 : principe et géométries associées à l'étape de demi-finition	13
Figure 9 : Principe et géométries associées à l'étape de finition [BOU, 13]	14
Figure 10: différents types d'outils [BOU, 13].	14
Figure 11: représentation d'outil hémisphérique tangent à la surface [BOU, 13].	15
Figure 12: Chemin d'outil montrant l'erreur de flèche [PEC, 06]	16
Figure 13 : chemin d'outil montrant l'erreur de crête [BOU, 13]	17
Figure 14: Découpage par plan parallèle [BOU, 13].	18
Figure 15: Mode de balayage par Z-constant [CHY, 04].	18
Figure 16: Classification des méthodes de simulation [BOU, 13].	20
Figure 17: Méthode Z-map [YUN, 02].	22
Figure 18 : Orientation des vecteurs [ZEF, 07].	22
Figure 19 : Modèle octree [ZEF, 07].	23
Figure 20: Modèle Graftree [KAW, 91].	24
Figure 21 : Model Voxels [KAL, 10].	25
Figure 22 : Identification du type d'un segment.	25
Figure 23: Simulation à base espace image [ARF, 11]	26
Figure 24: représentation d'un dixel sur le plan XY	26
Figure 25 : Représentation en Dixel et en Triple-Dixel [LEN, 11].	27
Figure 26 : Comparaison entre les données en simple-Dixel et en Triple-Dixel [MBZ, 10].	27
Figure 27 : Représentation du simple Dixel dans chaque plan.	28
Figure 28 : Représentation du modèles Triple-Dixel.	28

CHAPITRE 2

Figure 1 : diagramme d'états transition.	34
Figure 2 : représentation de la classification des segments avant l'homogénéité.	38
Figure 3 : représentation de la classification des segments et enveloppes de clusters après l'homogénéité.	38
Figure 4: différents cas de chevauchement sphère cluster	46
Figure 5 : différents cas d'intersection sphère segment.	48

Figure 6 : Tolérance d'usinage.	50
Figure 7 : Démonstration d'usinage de différents segments.	51
Figure 8:Diagramme cas d'utilisation général.	54
Figure 9: Diagramme cas d'utilisation Acquisition du modèle volumique et la trajectoire d'outil de la pièce.	54
Figure 10:Diagramme de cas d'utilisation Simulation d'enlèvement de matière.	55
Figure 11: diagramme de cas d'utilisation vérification de la tolérance.	56
Figure 12 : diagramme de séquence Récupération et structuration des Triple Dixel.	57
Figure 13 : diagramme de séquence Récupérer et structurer les points centre d'outil de la trajectoire.	57
Figure 14 : diagramme de séquence de simulation d'enlèvement de matière.	58

CHAPITRE 3

Figure 1:diagramme de paquetage.	61
Figure 2:Classe Cellule.	62
Figure 3:Classe Droit.	62
Figure 4:Classe Dixel.	62
Figure 5:Classe Centre_Outil.	63
Figure 6:Classe Outil_Stl	63
Figure 7:Classe Contour_Stl.	63
Figure 8:Classe Noeud_Contour_Stl.	64
Figure 9:Classe Liste_Contours_Stl.	64
Figure 10:Classe Trajectoire.	64
Figure 11 : Diagramme de classes.	65
Figure 12:Classe Point.	65
Figure 13:Classe Segment.	66
Figure 14:Classe TDixel.	66
Figure 15:Classe Cluster	67
Figure 16:classe Liste_Cluster.	67
Figure 17:Classe Enveloppe_Cluster.	67
Figure 18:Classe TrajectoireOutil.	68
Figure 19:Classe PointcentreOutil.	68
Figure 20:Classe Simulation.	69
Figure 21:Fonctionnalités de l'onglet pièce.	71
Figure 22:onglet surépaisseur.	72
Figure 23:Visualisation de la surépaisseur d'usinage.	73
Figure 24:Visualisation de la pièce avec surépaisseur.	73
Figure 25:onglet Optimisation.	74
Figure 26 : Visualisation globale des clusters.	75
Figure 27:Exemple : visualisation locale d'une enveloppe d'un cluster.	76
Figure 28:visualisation globale des enveloppes des clusters.	76
Figure 29:onglet simulation.	77

<i>Figure 30: Simulation d'enlèvement de matière.</i>	78
<i>Figure 31 : Après la simulation où l'outil change</i>	79
<i>Figure 32 : Après la simulation d'usinage d'une trajectoire avec une distance entre les plans $z_constant = 4$.</i>	79
<i>Figure 33 : Après la simulation d'usinage d'une trajectoire avec une distance entre les plans $z_constant = 1$.</i>	79
<i>Figure 34 : Onglet résultats et statistiques.</i>	80
<i>Figure 35 : Visualisation des segments bien usinés et les segments mal usinés en enlevant plus de matière.</i>	81
<i>Figure 36 : Visualisation des segments bien usinés et les segments mal usinés en enlevant moins de matière.</i>	81
<i>Figure 37 : Visualisation de tous les segments après le calcul des erreurs.</i>	81

Liste des tableaux

<i>Tableau 1 : Test sur le paramètre de pas de discrétisation.</i>	82
<i>Tableau 2 : Test paramètre de tolérance d'usinage.</i>	83

Liste des algorithmes

<i>Algorithme 1 : Ecriture du fichier Triple-Dexel.</i>	35
<i>Algorithme 2 : Lecture du fichier Triple-Dexel.</i>	36
<i>Algorithme 3 : Ajouter surépaisseur.</i>	37
<i>Algorithme 4 : Homogénéisation des segments</i>	39
<i>Algorithme 5 : Classification des segments par la méthode K-means.</i>	42
<i>Algorithme 6 : Ecriture du fichier trajectoire.</i>	44
<i>Algorithme 7 : Lecture du fichier trajectoire.</i>	45
<i>Algorithme 8 : Calculer intersection sphère segment.</i>	50
<i>Algorithme 9 : Vérification de la tolérance.</i>	52
<i>Algorithme 10 : Simulation.</i>	52

Liste des organigrammes

<i>Organigramme 1 : Architecture globale du système.</i>	33
<i>Organigramme 2 : Organigramme de création et lecture des fichiers Triple-Dexel.</i>	34
<i>Organigramme 3 : Organigramme lecture et écriture fichier CE.</i>	43

INTRODUCTION

GENERALE

Présentation du sujet :

Devant la complexité de la géométrie des pièces à usiner, dans les industries telles que l'automobile, l'aéronautique, les emballages et autre ; il est devenu impératif pour les fabricants de respecter des contraintes et exigences pour produire des pièces fonctionnelles et de bonne qualité de surface

Dans cet ordre d'idées et dans le cadre du programme piloté par l'équipe CFAO au niveau du CDTA (Centre de développement des technologies avancés), il nous a été proposé de concevoir une application de simulation d'enlèvement de matière lors de la finition des surfaces complexes sur des fraiseuses 05 axes. Cette approche est très intéressante vue qu'elle prend en charge un volet très important à savoir la visualisation virtuelle d'un procédé d'usinage. Ce projet est une continuité de plusieurs travaux :

- Modélisation et conception des courbes et des surfaces gauches,
- Finition des surfaces gauches avec différentes stratégies de finition,
- Simulation virtuelle de l'usinage en ébauche et en finition,
- Finition des surfaces gauches à partir d'un nuage de points quelconque.
- La représentation volumique des pièces complexe en Triple Dixel [SEM,15] .

1. Problématique :

La production des pièces mécaniques complexes des surfaces fonctionnelles fait souvent appel à l'enlèvement de matière c'est-à-dire l'usinage. L'usinage de haute précision des pièces gauche (complexe) est l'un des processus clés dans la fabrication moderne, pour rester compétitifs les fabricants cherchent constamment à augmenter la qualité, réduire les couts d'usinage et les délais de production. Cependant, l'usinage réel peut provoquer une perte en productivité si la qualité des pièces est défailante.

La forme finale de la pièce est obtenue en trois opérations: ébauche, semi-finition et la finition, et une éventuelle superfinition. Ces opérations sont liées à des paramètres correspondants (stratégie d'usinage, outil, profondeur de passe, ...etc.). Dans la pratique, les pièces ne peuvent pas être fabriquées aux dimensions idéales exigées par le concepteur étant

donné les imprécisions inhérentes aux procédés de fabrication. Ces imperfections sont reconnues comme des défauts de fabrication.

La problématique est la prédiction de la topographie 3D des surfaces gauches lors de l'opération de finition (usinage en bout) avec des outils hémisphériques sur des fraiseuses numériques à 05-axes en prenant en compte les différents paramètres cités ci-dessus. Le but recherché est l'étude de l'influence de ces paramètres sur la qualité des surfaces (rugosité) et la sélection des paramètres appropriés conduisant à obtenir un meilleur état de surface pour éviter l'opération de super finition afin de réduire les coûts de fabrication.

2. Objectif :

La simulation d'usinage devient de plus en plus indispensable dans diverses industries car elle permet de vérifier la conformité de la pièce avec les exigences du concepteur. Le but de ce travail est le développement d'un module logiciel graphique et interactif sous Windows permettant la simulation d'enlèvement de matière en usinage 05 axes de pièces de formes complexes avec un outil hémisphérique, afin de prédire la topographie (qualité) de la surface finie.

3. Structuration du mémoire :

Le présent mémoire est composé des chapitres suivants :

- Le premier chapitre est consacré à des généralités sur l'usinage 5axes afin d'introduire le domaine d'étude.
- le deuxième chapitre présente l'étude conceptuelle et les algorithmes développés dans notre application logicielle.
- Le dernier chapitre présente l'implémentation, les tests réalisés pour valider les résultats.

CHAPITRE I

GENERALITES SUR LE PROCESSUS DE FABRICATION DES FORMES COMPLEXES

1. Introduction:

Face à la concurrence, les industriels doivent réagir rapidement afin de concevoir de nouveaux produits (automobile, aéronautique... etc.), toujours élaborer de formes variées et de plus en plus complexes. Les concepteurs ont sous la main toute une panoplie de logiciels sophistiqués, répondants à tous les cas de figures de conception et fabrication assistée par ordinateur (CFAO) des systèmes intégrés ou découplés, dont la liaison entre ces derniers est faite par des fichiers de format d'échange. Dans notre travail nous considérons le format STL, c'est ce que nous abordons dans la 1ere partie de ce chapitre. La deuxième partie on entre dans quelques généralités en mécanique concernant les fraiseuses et l'usinage des surfaces complexes. La 3eme partie, nous entamons notre travail concernant la simulation d'usinage en bout des surfaces gauches en abordant la modélisation en Triple-Dexel qui sera le point de départ dans notre application, et nous terminons par une conclusion.

2. Conception et fabrication assisté par ordinateur(CFAO) :

2.1. Définition :

La CFAO est la combinaison de la CAO (Conception Assistée par Ordinateur) et de la FAO (Fabrication assistée par Ordinateur), c'est un procédé qui permet de concevoir et programmer les formes d'une pièce à usiner et d'en définir les différentes opérations d'usinage et agencer les phases successives menant à la réalisation en final d'une pièce conforme aux exigences exprimées par le designer.

L'évolution des systèmes de la CFAO doivent répondre aux critères suivants :

- Très haute rentabilité.
- Automatisation complète de la fabrication.
- Communication et échange de données.
- Perfection et finition des surfaces.
- Contrôle des éventuels points de collisions.
- Ces systèmes doivent procéder à la vérification des performances et de la productivité globale : Applicables à des formes diverses et complexes.
- Applicables à divers modèles de surfaces.
- Traiter des pièces géométriques différentes.
- Respecter la qualité imposée et la validité de la pièce.
- Contrôler l'efficacité de l'usinage.

2.2. Conception assistée par ordinateur(CAO) :

L'ingénieur en charge de la construction possède des informations provenant de calculs ou mesures matérialisés par des points, des courbes, des surfaces, appelées données géométriques fonctionnelles, elles servent de canevas pour la reconstruction de la surface dans un environnement CAO.

En sus de ces données l'ingénieur prendra en compte les contraintes fonctionnelles de construction ne pouvant être exprimées sous forme géométrique (points, droites....) d'une base de données CAO, suivant le cas nous pouvons être confrontés à ces types de contraintes : masses, volumes, surfaces ou de directions de tangence et de concavité.

2.3. Fabrication assistée par ordinateur(FAO) :

Après avoir numérisé la pièce par des données géométriques fonctionnelles à l'aide de la CAO, la production des programmes d'usinage qui contiennent les parcours d'outils et les fonctions annexes (comme la mise en route de la broche, le changement d'outil... etc.) se fait par la FAO. Ces logiciels ne sont pas entièrement automatiques, laissant ainsi à l'utilisateur de fixer le type de trajectoire à utiliser en fonction de l'entité choisie (profil, poche, trou,... etc.).

2.4. Processus de CFAO :

De la conception à la production, le procédé de CFAO est constitué d'une suite d'activités importantes pour la réalisation d'une forme gauche en passant par deux étapes :

- Construction géométrique du modèle (CAO).
- Génération des trajectoires d'outil (FAO).

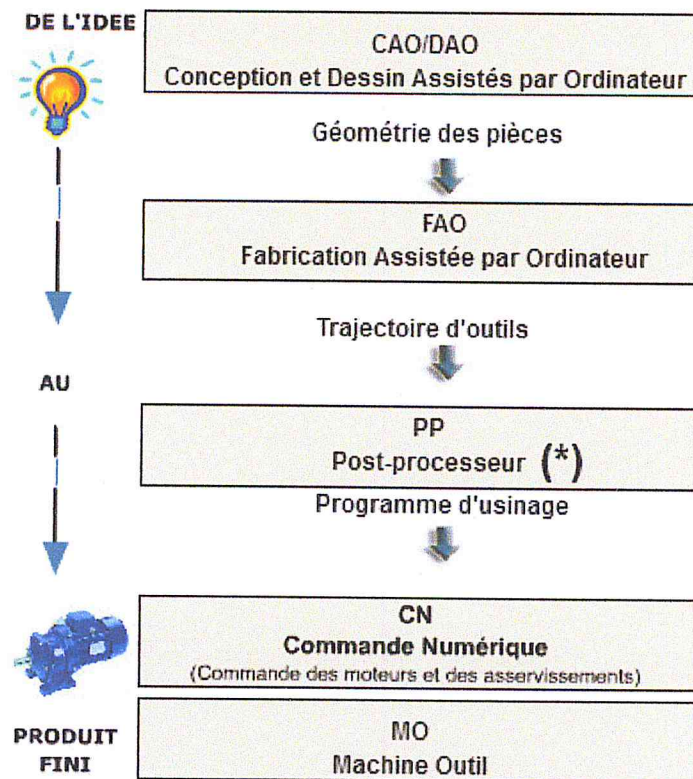


Figure 1: processus de fabrication d'une pièce mécanique[BEA, 13].

2.4.1. Construction du modèle géométrique :

Cette étape a pour but d'exprimer l'ensemble des contraintes fonctionnelles écrites dans le cahier de charges, sous forme de surfaces ou d'éléments géométriques décrits à l'aide d'un système de CAO. Comme il existe deux types de systèmes CFAO, le système intégré et le système découplé ou interfacé (Figure 2).

Dans le système intégré le transfert de données géométriques de la CAO à la FAO se fait systématiquement, par contre pour Le système découplé le passage se fait à l'aide d'un format d'échange de données tel que : IGES, STL, VRML, DWG, DXF, STEP. Dans notre travail nous avons choisit le format STL pour la représentation de pièce de formes complexes.

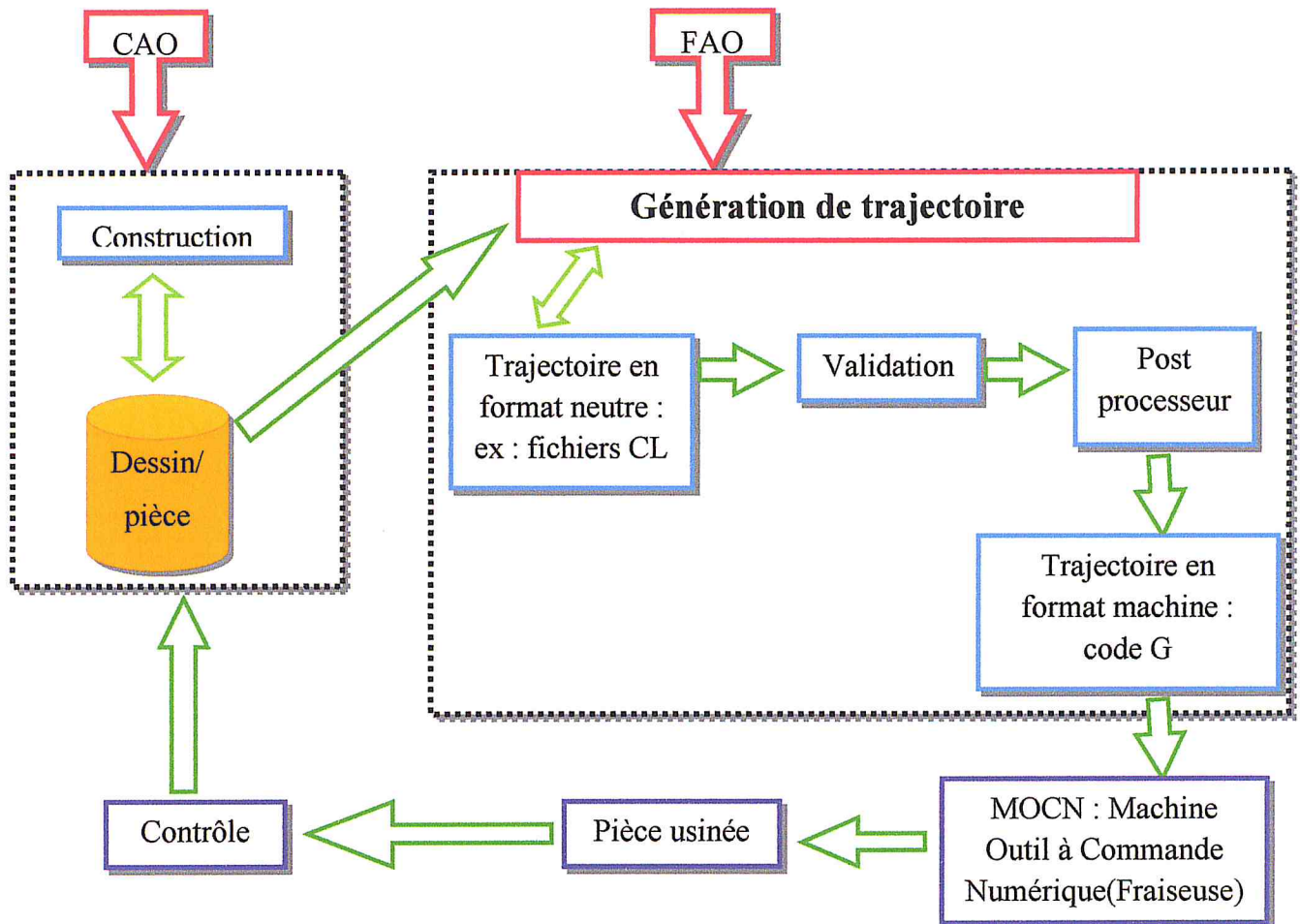


Figure 2: Flux d'information sur le système de CAO-FAO découplé ou Interfacé [BOU, 13].

2.4.1.1. Le format d'échange STL :

Le modèle STL permet de représenter la peau extérieure des objets avec des facettes triangulaires dont le nombre et la taille dépendent de la précision demandée (Figure 3). Par conséquent, ce modèle est composé d'une liste de triangles [BCK, 11].

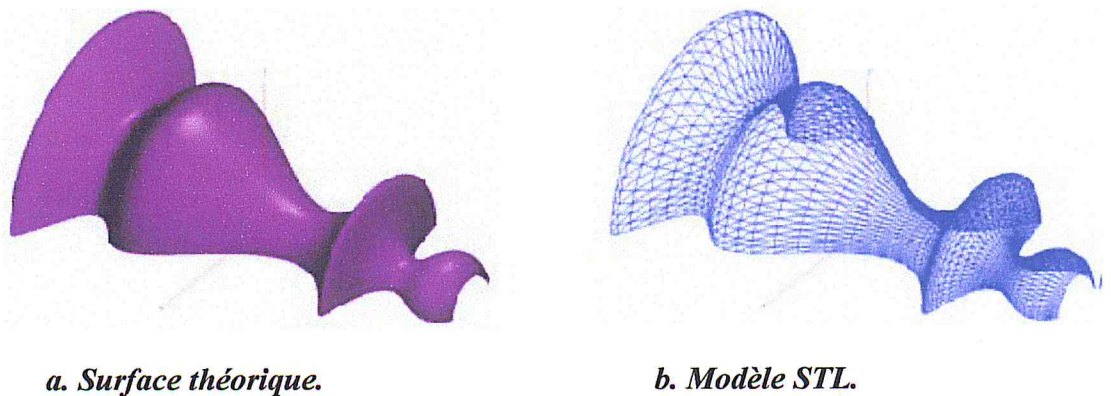
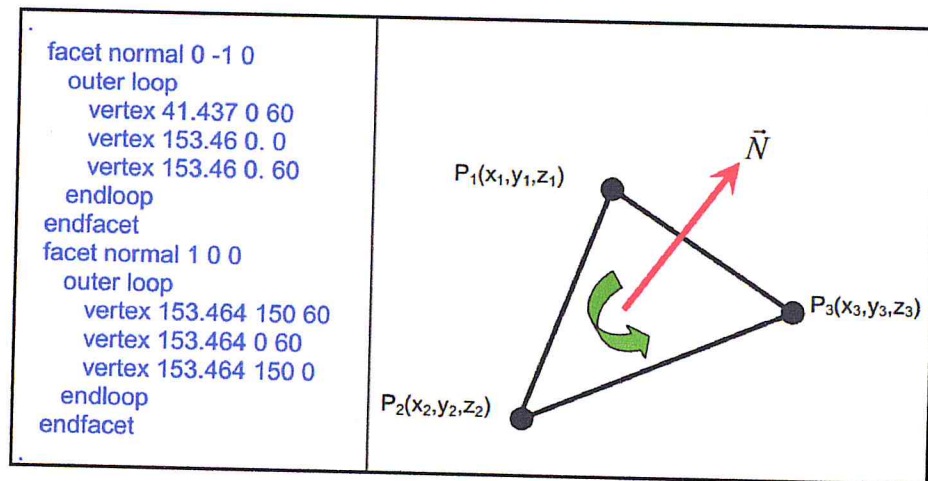


Figure 3 : Modèle STL d'une surface théorique [BCK, 11].

Chaque triangle est défini par les composantes de son vecteur normal unitaire \vec{N} dirigé vers l'extérieur et les coordonnées X, Y et Z de ses trois sommets P1, P2 et P3 (Figure 4).



a. Structure du fichier « STL ».

b. Paramètres d'un triangle.

Figure 4: Fichier STL et paramètres d'un triangle [BOU, 13].

2.4.2. Génération de trajectoire d'outils :

Etape dite de fabrication, elle a pour but de matérialiser les éléments en une forme réelle permettant à l'objet de répondre à sa fonction. Pour déterminer le trajet d'outil il faut d'abord trouver les points de contact entre l'outil avec la pièce à usiner (CC-File), les points d'extrémités de l'outil (CL-File), ou les points centres d'outil.

✚ CC-File :

Le calcul des points de contact CC constituant le fichier CC-File, résultat des intersections des plans de coupe avec la surface de la pièce à usiner s'effectue en deux étapes, à savoir:

- Discrétisation de la surface en facettes.
- Identification des points communs entre les plans de coupe et les facettes de la surface de la pièce.

✚ CL-File :

Ce sont les points représentant l'extrémité de l'outil au cours de l'usinage de la pièce. Dans le fichier de CL, chaque donnée CL contient 6 paramètres (X, Y, Z, I, J, K). Tel que (X, Y, Z) représentant le point CL et (I, J, K) composant un vecteur d'unité représentant l'orientation de coupe. Si les trois autres (I, J, K)

sont omis, cela signifie que dans ce lieu de coupe l'orientation de coupe est inchangée.

2.4.2.1. Trajectoire en format neutre :

Ces trajectoires d'outils exprimées dans un repère lié à la pièce à usiner ne tiennent compte ni de la cinématique de la machine ni de la position de la pièce sur le plateau de la machine. N'étant pas modifiables dans la plupart des cas, à ce stade seul un programmeur expérimenté peut le faire.

Il faut noter que chaque ligne de programme comporte des informations sur les trajectoires et l'attitude de la machine sur chaque point de la trajectoire. Nous devons utiliser un post-processeur ceci afin d'exprimer ces trajectoire dans un repère lié à la machine dans un langage compressible par la commande numérique.

2.4.2.2. Trajectoire en format machine :

Ce sont des trajectoires exprimées dans un langage compréhensible par la commande numérique, il s'agit du Langage ISO.

Langage ISO :

Le langage ISO est une programmation qui s'appuie sur des fonctions préparatoires de type G, et complétées par des fonctions auxiliaires (de type M) et technologiques (F, S...Etc.). Il n'est pas neutre et dépend fortement de la machine réalisant l'usinage. Les fonctions de base de la machine sont communes à toutes les machines l'utilisant.

Par l'intermédiaire de ce code, l'utilisateur communique à la machine un ensemble d'instructions explicites. Un programme en code G est donc la traduction d'une suite d'actions et de déplacements élémentaires (ligne droite ou arc de cercle par exemple) qui permettent de générer des conditions d'usinage ainsi qu'une trajectoire plus ou moins approximée [BET, 12].

2.4.3. Validation et simulation :

C'est l'étape validant le choix de la méthode de calcul des trajectoires d'outils et la stratégie d'usinage optimale avec simulation d'usinage pour détecter les anomalies (problèmes de collisions et d'interférence) et autres problèmes liés au choix de paramètres d'usinage et conditions opératoires.

2.4.4. Post-Processeur :

Programme transformant les trajectoires en format neutre en des trajectoires comprises par la machine en tenant compte de la cinématique de la machine, et ses capacités. Il signale les erreurs et exprime les trajectoires dans le langage spécifique à la machine.

3. Fraiseuse à commande numérique :

On s'intéresse à ces machines où sont enregistrés des ordres de mouvement d'outil pour usiner une pièce complexe (pilotée par un programme informatique en langage ISO) [BOU, 13].

3.1. Les axes de déplacement :

Un système de coordonnées permet de repérer les positions et les déplacements d'un objet par rapport à un point origine. Un système de coordonnées cartésiennes rectangulaires est un trièdre de sens directs constitué de trois axes linéaires X , Y et Z auxquels sont associés trois axes rotatifs A , B et C . La règle des trois doigts permet de retrouver facilement l'orientation des axes X , Y et Z (voir la figure 5) [BOU, 13].

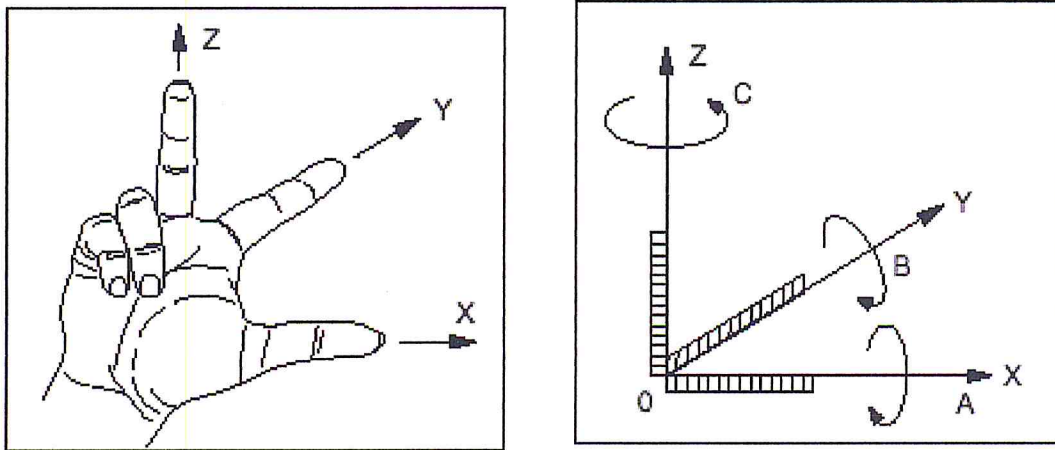


Figure 5 : Orientation des axes [BOU, 13].

L'orientation des axes d'une machine dépend du type de machine et de la disposition des éléments qui la constituent. Pour une fraiseuse :

- L'axe Z est confondu avec l'axe de la broche principale lorsque celui-ci est parallèle à l'une des glissières. Un déplacement dans le sens Z positif accroît la distance entre la pièce et l'outil.
- L'axe X est perpendiculaire à l'axe Z et correspond au plus grand déplacement.
- L'axe Y forme avec les axes X et Z un trièdre de sens direct.

Chaque axe représente un degré de liberté, donc la combinaison des différents axes donne plusieurs types de fraiseuses [ZEF, 07]. Les fraiseuses 3 et 5 axes sont menés d'une cinématique comme suit :

- X, Y, Z simultanés.
- X, Y, Z et A et B (2 rotations) simultanés.

3.2. Fraiseuse 3 et 5 axes :

Par définition, une fraiseuse à cinq axes offre deux degrés de liberté supplémentaires par rapport à une fraiseuse à trois axes. Ces deux degrés de liberté permettent d'orienter l'axe de l'outil. En usinage à trois axes, pour une surface usinable, il existe une unique position de l'outil tangente à la surface en une position donnée. Par contre, en usinage à cinq axes, il existe une infinité de positions de l'outil tangentes à la surface au point de contact. Ces degrés de liberté supplémentaires rendent le problème sous contraint lors du calcul du trajet d'usinage, mais permettent l'optimisation de l'orientation de l'outil en fonction de critères tels que la minimisation des efforts de coupe ou l'augmentation de la zone usinée [ZEF, 07] (voir figure6).

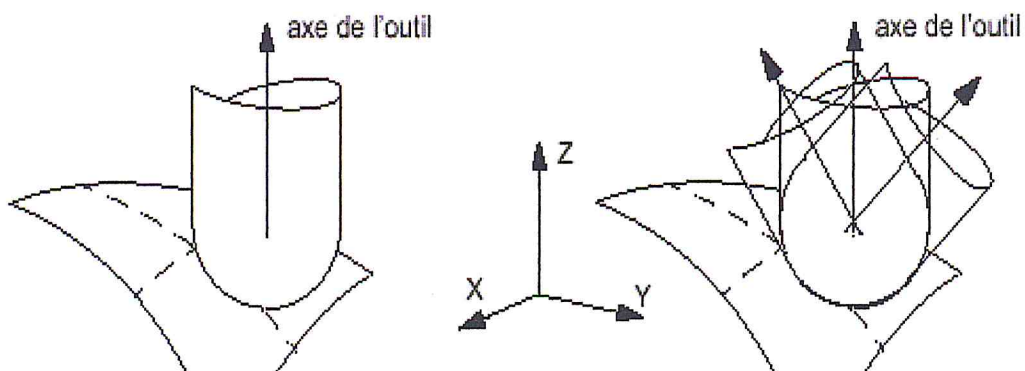


Figure 6: comparaison entre l'usinage à trois axes et l'usinage à cinq axes [ZEF, 07].

Avec les géométries complexes, comme les cavités profondes, les parois hautes et à fortes pentes et les contre-dépouilles, l'usinage 3 axes n'est pas adapté en raison des collisions sauf avec des rallonges d'outils. L'usinage de ces zones exige de nombreuses positions d'outil dans des zones d'usinage parfaitement délimitées. Ces positions peuvent être atteintes avec l'usinage 5axes sans collisions. En fonction de la géométrie et de la cinématique, il est possible de choisir dans l'usinage à 5 axes, l'indexage automatique où un véritable usinage simultané.

L'usinage à cinq axes permet de réaliser des surfaces en ayant un seul point de contact (*usinage en bout*) ou une ligne de contact entre l'outil et la surface nominale (*usinage par flanc*).

4. Génération de trajectoire d'outil :

4.1. Définition d'usinage :

L'usinage est une opération consistant à soumettre une pièce à l'action d'une ou de plusieurs machines-outils, essentiellement dans le cas où cette action consiste à un enlèvement de matière. Le terme usinage, synonyme du mot enlèvement de matière ou enlèvement de copeaux, désigne tous les moyens mis en œuvre pour obtenir une pièce ayant des côtes et des tolérances fixées à l'avance, à partir d'une pièce brute venue de la forge ou de fonderie [GAR, 14].

4.2. Différentes étapes de production :

L'usinage d'une pièce passe par plusieurs étapes sont les suivantes:

4.2.1. Ébauche :

C'est la 1ere étape elle permet l'enlèvement d'un maximum de matière en un minimum de temps (Figure 7). Durant ce type de travail, l'outil devra résister à des efforts de coupe importants: il est donc massif [GAR, 14].

▪ Rôle :

- Eliminer la croûte superficielle liée au procédé d'obtention du brut (calamine), par une opération d'écroûtage.
- Eliminer les fortes surépaisseurs d'usinage (excédent important de matière).

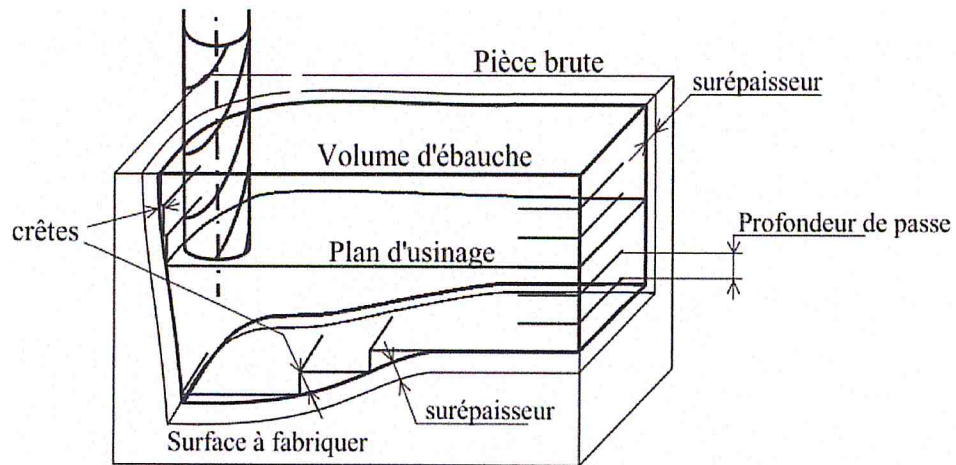


Figure 7 : principe et géométries associées à l'étape d'ébauche [BOU, 13].

4.2.2. Demi-finition :

L'usinage se fait avec une fraise hémisphérique ou torique dans le but d'ôter les escaliers (crêtes) et d'amener la matière restante à une surépaisseur la plus constante possible (Figure 8). Elle utilise souvent une surépaisseur intermédiaire entre celle de l'ébauche et celle de la finition.

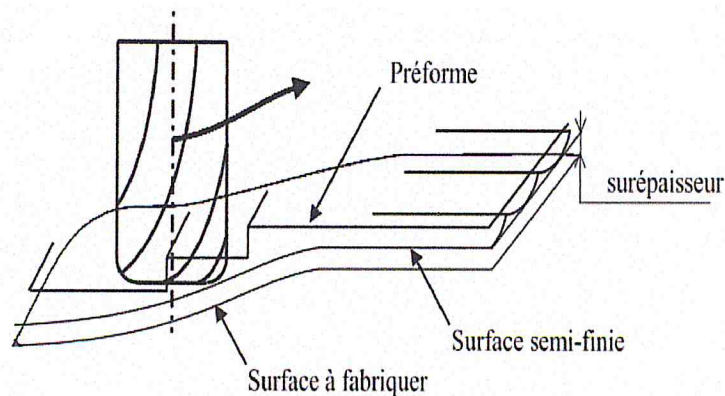


Figure 8 : principe et géométries associées à l'étape de demi-finition [BOU, 13].

4.2.3. Finition :

Pour la finition, des fraise hémisphérique ou fraises toriques sont les plus utilisés, avec la finition on doit arriver dans toutes les parties de la pièce à la côte finale de l'usinage (figure 9).

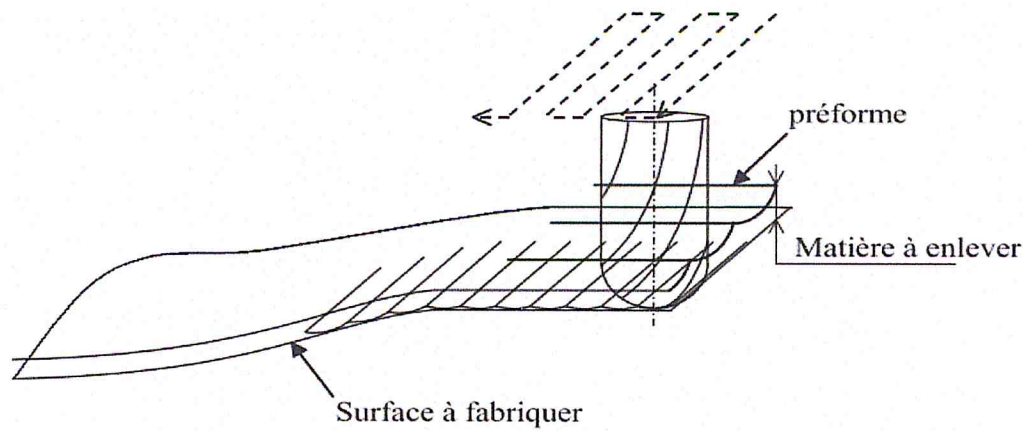


Figure 9 : Principe et géométries associées à l'étape de finition [BOU, 13]

4.3. Outils utilisés en finition :

Contrairement aux autres types d'usinages rencontrés en mécanique générale, il n'existe pas de forme d'outil qui permette l'obtention d'une forme gauche exactement en un seul mouvement élémentaire. Le choix de l'outil est donc le résultat d'un compromis entre :

- la rigidité de l'outil,
- la cinématique de la machine,
- la forme de la pièce à usiner.



Outil cylindrique



Outil torique



Outil hémisphérique

Figure 10: différents types d'outils [BOU, 13].

4.4. Positionnement de l'outil relativement à la surface à usiner :

L'usinage trois à cinq axes permet de réaliser des surfaces en ayant un seul point de contact entre l'outil et la surface nominale. Le positionnement d'une fraise hémisphérique tangente à la surface est donné par la figure 11.

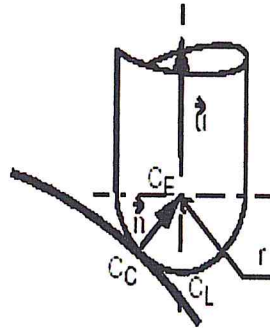


Figure 11: représentation d'outil hémisphérique tangent à la surface [BOU, 13].

On appelle :

- CC (cutter contact) le point de contact entre l'outil et la surface.
- CE le point centre de l'outil.
- CL (cutter location) le point extrémité de l'outil.
- le vecteur \vec{n} , la normale à la surface au point de contact.
- le vecteur \vec{u} , l'axe de l'outil.
- r le rayon de l'outil hémisphérique et le petit rayon de l'outil torique.

4.5. Paramètres de calcul de trajectoire d'outil en fraisage en bout :

Pour générer une séquence d'usinage il faut notamment générer des trajets d'usinage. Les trajectoires suivies par l'outil sont établies en fonction du mode d'usinage, des paramètres de guidage (guidage du point extrémité de l'outil, guidage du point centre d'outil) et des paramètres de passe (pas longitudinal, pas transversal).

4.5.1. Les paramètres de guidage de l'outil :

Le guidage de l'outil le long de la courbe peut se faire de deux manières différentes, soit par guidage du point extrémité de l'outil, soit par guidage du point centre d'outil. En effet, pour l'usinage d'une surface quelconque, le point extrémité d'outil et le point centre d'outil n'ont pas la même trajectoire et les algorithmes de calcul de trajets ne sont pas les mêmes si l'on guide l'extrémité de l'outil ou le point centre d'outil.

4.5.2. Paramètre de passe de l'outil :

Un trajet est représenté par une succession de couples formés par la position de l'extrémité de l'outil et la direction de l'axe d'outil. Générer des trajets d'outil, c'est trouver les positions de l'outil successives qui permettent d'usiner une surface selon des trajectoires définies par le programmeur machine-outil. Ces trajectoires sont en général des courbes 3D.

La stratégie d'usinage définit complètement le mode de balayage de la surface et son interpolation dans un format compréhensible par le directeur de commande numérique. Pour la direction d'usinage, il faut associer deux paramètres d'usinages [KAW, 91] : Le long de la passe et entre deux passe.

4.5.2.1. Paramètres de passe longitudinal :

- **pas longitudinal :** Variable au long du trajet d'usinage d'un point à un autre. Il est dû à l'emploi de l'interpolation linéaire pour décrire le trajet. Il a pour but de garantir le respect de la position relative de la trajectoire par rapport à la surface nominale.
- **erreur de flèche :** L'outil usine une ligne brisée, le pas longitudinal d'usinage est calculé en général par le respect de la tolérance d'usinage qui est la valeur maximale de l'erreur de flèche entre la courbe et chacun des segments de la ligne brisée.

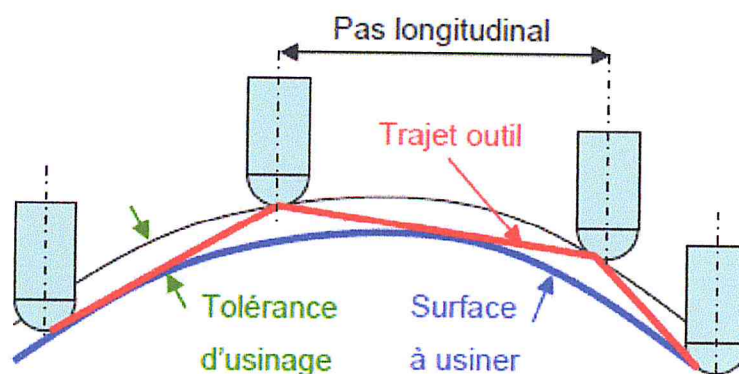


Figure 12: Chemin d'outil montrant l'erreur de flèche[PEC, 06]

4.5.2.2. paramètres de passe transversal :

- **pas transversal :** Variable entre deux passes élémentaires d'usinage. Ce pas est mis en rapport avec la hauteur de la crête que l'on désire laisser sur la pièce du fait du balayage par un outil hémisphérique ou torique.
- **erreur de crête:** Afin d'usiner la totalité de la surface, les passes sont juxtaposés avec une erreur donnée. Le pas d'usinage transversal peut être calculé en respectant soit un critère de distance maximale entre deux passes, soit un critère de hauteur de crête. Ces deux paramètres sont équivalents et liés par le rayon de l'outil. De plus, la distance maximale entre passes peut être établie entre les surfaces de guidage ou sur la surface nominale (Figure 13).

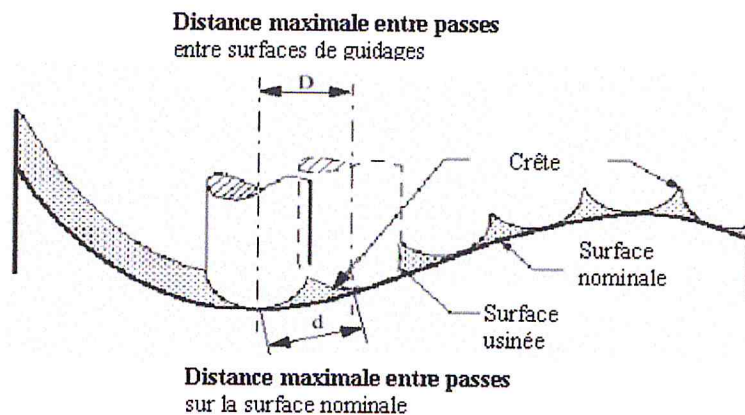


Figure 13 : chemin d'outil montrant l'erreur de crête [BOU, 13]

4.6. Stratégies d'usinage des surfaces gauches en finition :

4.6.4. Plans parallèles :

Les chemins d'outils sont parallèles dans l'espace euclidien (figure 14).

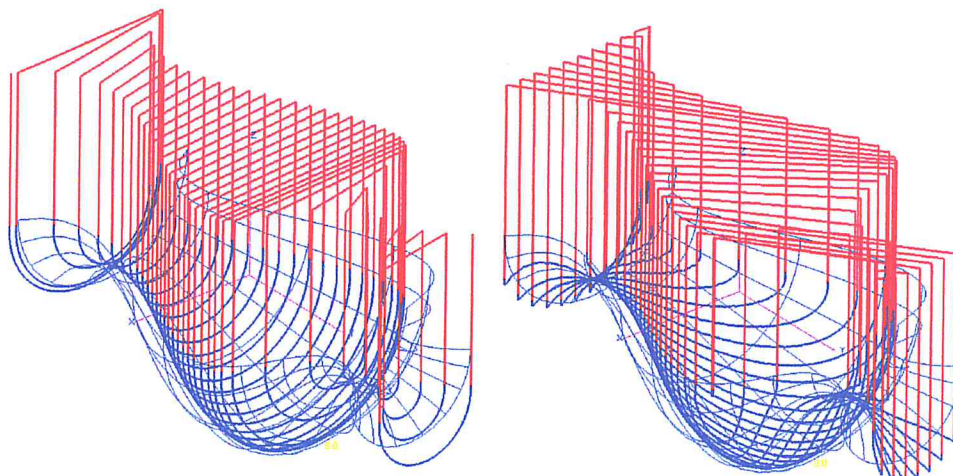


Figure 14: Découpage par plan parallèle [BOU, 13].

4.6.5. Z-constant :

La stratégie Z-Constant est utilisée pour l'usinage des cavités profondes afin d'éviter les longues descentes et montées d'outil. Cela est réalisé par le découpage de la surface en plusieurs plans parallèles à l'axe Z et l'intersection de ces plans avec la surface théorique produit ce qu'on appelle les contours (figure 15).

La stratégie Z-Constant suit le processus suivant :

- Calcul des contours.
- Liaison de ces contours pour construire le trajet d'usinage.

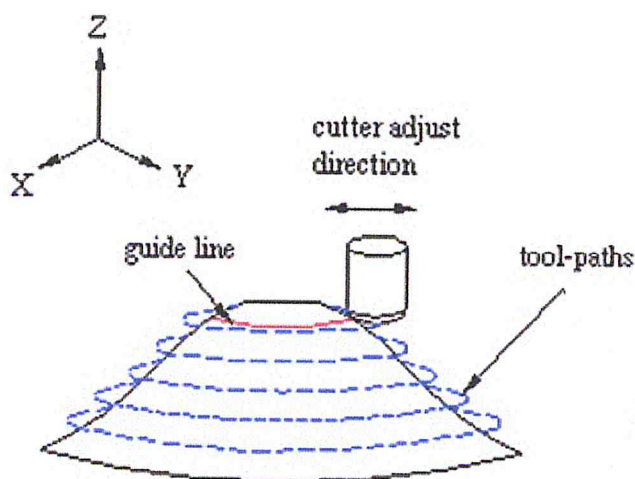


Figure 15: Mode de balayage par Z-constant [CHY, 04].

5. simulation d'usinage des formes gauches sur les machines multiaxes :

La simulation d'usinage est basée sur le principe de l'élaboration d'un plan de coupe le plus réel possible, étape cruciale et incontournable du processus de CFAO, cette technologie au domaine très large touchant différents niveaux souffre encore fortement de nombreuses lacunes pénalisantes dans la connaissance de la coupe. Cette tâche cerne les différentes approches de simulation concises ouvrant la voie à une explication optimale en répondant aux questions clés (figure16) :

- objectifs (simulation géométrique ou physique)
- à quelle échelle (humaine, macroscopique, microscopique)
- comment (modèles géométrique, modèle dynamiques).

Dans notre travail nous allons considérer la simulation géométrique à l'échelle macroscopique.

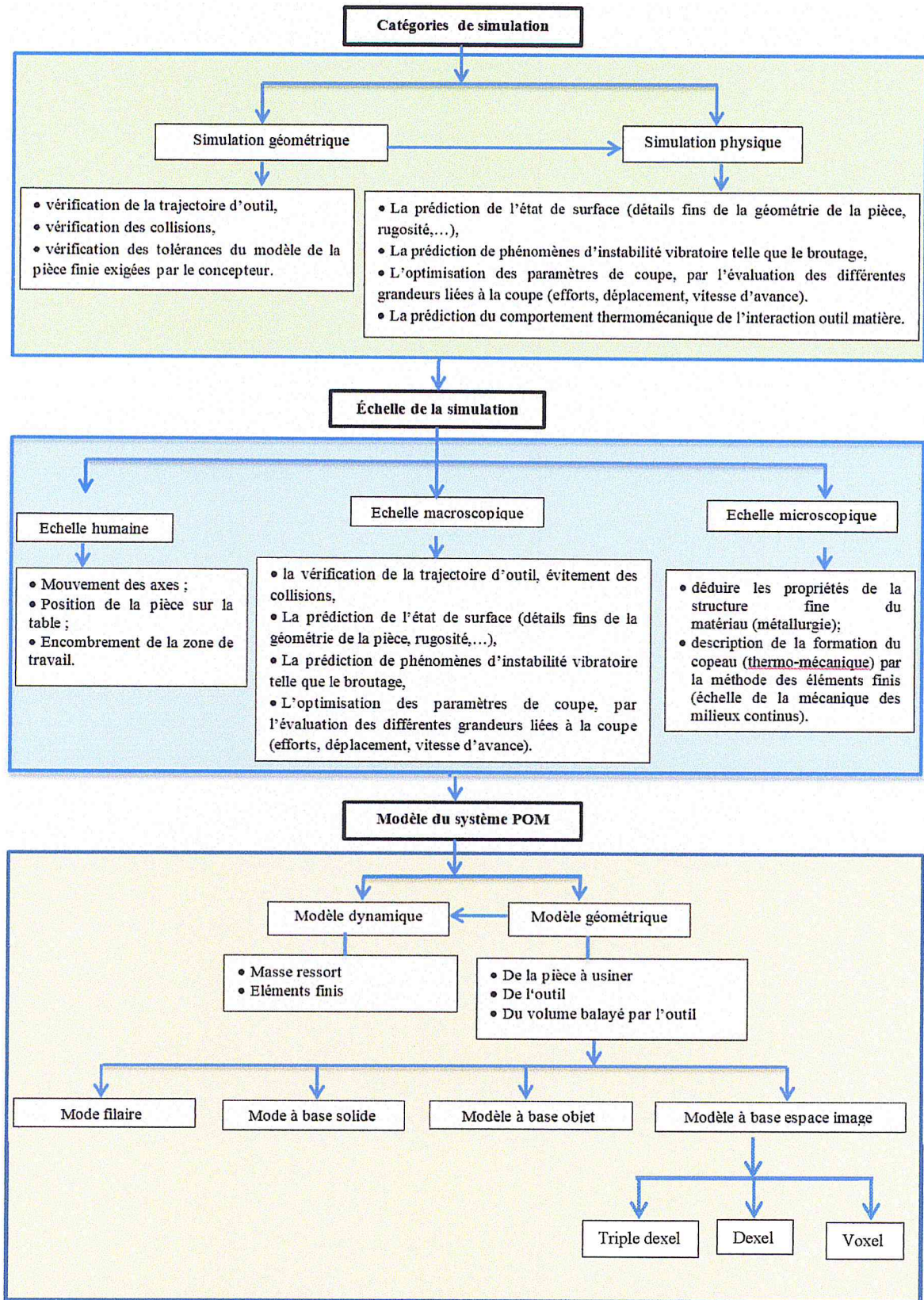


Figure 16: Classification des méthodes de simulation [BOU, 13].

5.1. Simulation géométrique a l'échelle macroscopique :

La littérature montre l'existence de différentes manières de classification pour la représentation géométrique en simulation à l'échelle macroscopique. Les méthodes utilisées sont classées comme suit : mode filaire, mode solide, mode espace objet, mode espace image [ZEF, 07].

5.1.1. Mode filaire :

Dans ce mode de simulation, la trajectoire et la forme de la pièce à usiner sont affichées sous forme de fil de fer. Ce modèle a une structure de données simple et rapide. Il a été largement appliqué au début de la simulation de l'usinage. L'utilisation de ce modèle reste applicable aux pièces de géométrie simple.

5.1.2. Mode à base solide :

Ce mode est une modélisation volumique de représentation 3D plus complète. Il est utilisé pour la simulation géométrique et pour la simulation physique. Ce mode permet une représentation géométrique très précise mais coûteuse [ZEF, 07]. Les deux modèles existants pour ce mode sont :

- ✚ Modèle CSG : il définit la forme constructive d'un modèle 3D en utilisant les volumes primitifs tels que le cylindre, la sphère, ... etc. Bien que les opérations booléennes et la vérification de cohérence soient simples, la visualisation ou l'analyse des données peut exiger une transformation en un autre modèle B-Rep. Le coût approximatif de la simulation à l'aide de CSG est d'ordre $O(n^4)$ où n est le nombre de mouvements de l'outil [VOH, 81]. Donc, la simulation d'usinage des surfaces gauches devient intraitable [JER, 89].
- ✚ Modèle B-Rep : ce modèle est adapté à la visualisation. Contrairement au modèle CSG, le modèle B-Rep définit explicitement le volume par une liste de surfaces, arêtes et sommets. Le coût de calcul est élevé en termes de temps, de stockage de données et de complexité. Pour un total de « n » mouvements de l'outil, le coût de la simulation est estimé à $O(n^{1.5})$ [KAS, 08].

5.1.2. Modèle à base objet :

Dans une simulation d'usinage à base d'espace objet, les pièces sont représentées par un ensemble de points discrets avec des vecteurs ou des surfaces avec des vecteurs ou certains éléments de volume. Il existe trois grandes méthodes de décomposition pour les modèles de simulation d'usinage espace objet :

- ✚ Méthode de Z-map : la méthode de Z_map consiste à décomposer le modèle de la pièce en plusieurs vecteurs 3D (Figure 17). Chaque vecteur commence par la valeur de la hauteur du brut. Pendant le processus de simulation, les mises à jour des hauteurs des vecteurs 3D sont relatives aux mouvements de l'outil. Dans ce cas, les opérations booléennes ont une seule dimension et par conséquent la simulation est très rapide.

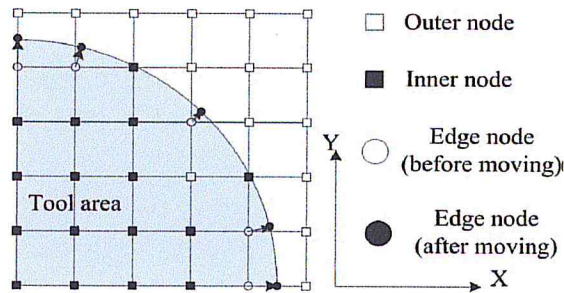
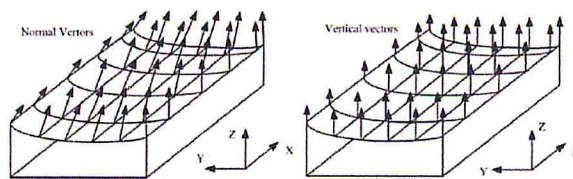


Figure 17: Méthode Z-map [YUN, 02].

- ✚ Méthode de vecteur : cette méthode consiste à découper la surface selon des méthodes spécifiques pour avoir un ensemble de points. Pour chaque point, est associé un vecteur dont les limites sont la surface nominale et la pièce brute. Ses vecteurs peuvent être orientés de deux manières (Figure 18):
 Suivant la normale à la surface (exacte) : dans ce cas, chaque vecteur est indépendant linéairement des autres vecteurs.
 Suivant l'axe Z de l'outil (simplifiée) : dans ce cas, tous les vecteurs sont parallèles à l'axe Z. Ce cas s'adapte à l'usinage 03 axes.



a. Suivant la normale .b. Suivant la verticale.

Figure 18 : Orientation des vecteurs [ZEF, 07].

Pour simuler l'usinage, les intersections des vecteurs avec l'enveloppe du volume balayé par l'outil doivent être calculées au fur et à mesure des déplacements de l'outil. La longueur et le sens des vecteurs sont modifiés pour chaque déplacement

élémentaire de l'outil. Pour détecter les zones non usinées, il suffit de vérifier la direction et la longueur des vecteurs :

- Direction positive : zone non usinée.
- Direction négative : usinage plus que nécessaire.
- Longueurs des vecteurs : s'ils ne sont pas dans les tolérances d'usinage, une correction est indispensable.

✚ Méthode d'octree : la méthode octree représente la pièce à usiner dans une structure arborescente (Figure 19). Chaque nœud d'un arbre est subdivisé récursivement en huit nœuds enfant disjoints jusqu'à ce que la précision exigée à l'objet soit atteinte. Cette représentation en octree hiérarchique offre à la simulation d'usinage CN une simplicité de calcul des opérations booléennes même lorsque la zone de coupe locale est complexe.

Kawashima et al. [KAW, 91], ont élaboré un modèle octree étendu appelé Graftree (Figure 20) pour représenter plus fidèlement les objets 3D dans la simulation géométrique. Dans le cas de Graftree, chaque cellule limite a été décrite dans la forme de CSG avec certaines restrictions. Kim et al. [KIK, 05-KIK, 08] ont utilisé la méthode de super-échantillonnage de l'ordinateur graphique pour améliorer le modèle d'octree.

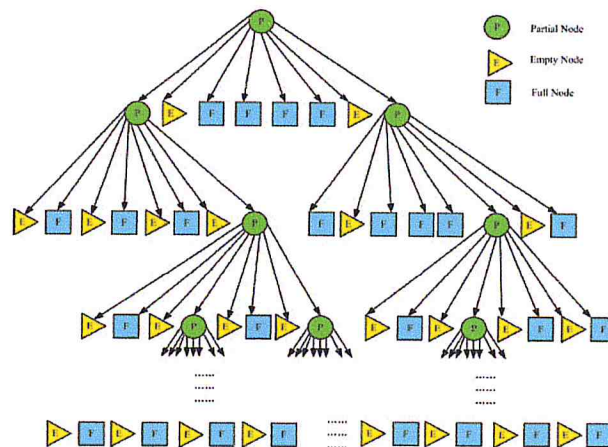


Figure 19 : Modèle octree [ZEF, 07].

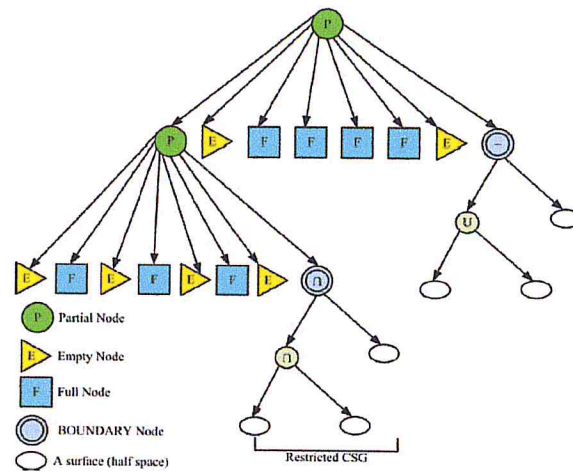


Figure 20: Modèle Graftree [KAW, 91].

5.1.3. Modèle à base espace image :

Dans la simulation d'usinage en espace image, les pièces sont représentées par des profondeurs et des pixels (Dexel). Ce modèle est l'extension du Z-buffer qui est utilisé dans l'élimination des parties cachées en infographie. Ce modèle est particulièrement performant, puisqu'il est issu des méthodes de représentation de l'imagerie et de rendu réaliste [ZEF, 07].

A partir de la classification illustrée sur la figure 16 et ce que présenté dans la section 5.1 de ce chapitre, nous traitons la simulation géométrique à l'échelle macroscopique dans le système pièce, outil et machine (POM) par la représentation à base espace image.

5.2. Choix de la technique adoptée :

Afin d'introduire la notion de matière, les pièces sont modélisées par des volumes élémentaires par des méthodes de discrétisation à savoir : Dexels, Voxels et les Triples Dexels.

5.2.1. Voxel :

Le Voxel (contraction de « volumetric pixel ») est un pixel en 3D (Figure 21). Il consiste à stocker une information colorimétrique avec ses coordonnées spatiales, Ils s'inscrivent plus généralement dans des espaces matriciels bien que les espaces vectoriels leur soient favorables. Leur point faible est de nécessiter énormément de ressources, tant pour leur stockage que pour leur rendu. Aujourd'hui, la

disponibilité de GPU a rendu possible d'obtenir un rendu en temps réel et l'interactivité avec les simulations voxels constitués de plusieurs milliards de voxels.

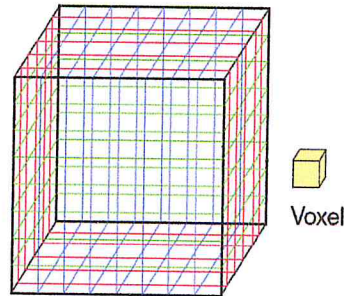


Figure 21 : Model Voxels [KAL, 10].

5.2.2. Simple Dixel :

Définition :

Un Dixel est un parallélépipède caractérisé par sa section représentée par la forme de la cellule, sa hauteur représentée par la longueur du segment de type « matière » (voir figure 22). Ces derniers forment les Dixels de la pièce dans la direction de la droite de la cellule (Figure 23.b). A partir de la section et de la longueur d'un Dixel, le volume du Dixel est calculé. Le volume total de l'objet est la somme des volumes de tous les Dixels. Le modèle de l'objet en Triple-Dixels est obtenu en combinant les Dixels créés dans les trois directions [BOB, 16].

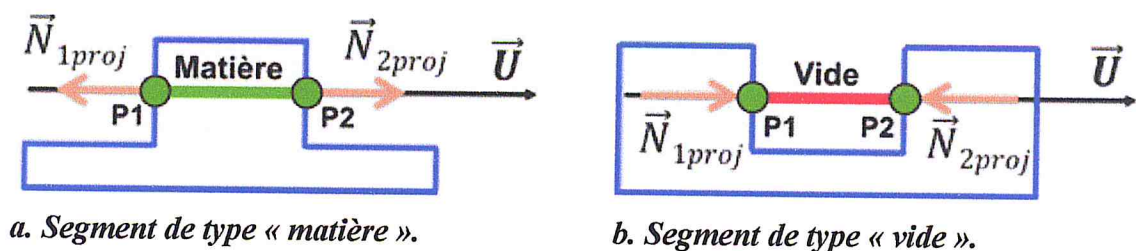
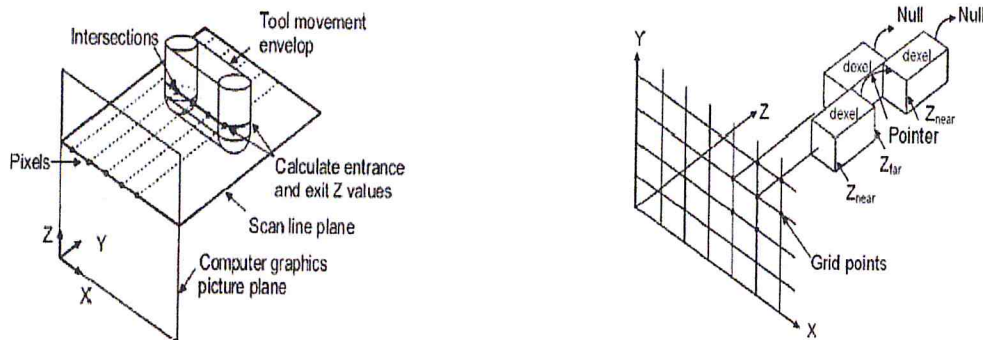


Figure 22 : Identification du type d'un segment [BOB, 16].

Le modèle Dixel est l'extension du Z-buffer qui est utilisé pour l'élimination des surfaces cachées dans l'ordinateur, il découle des limites de Z-map, ce modèle est

particulièrement performant, il est directement issue des méthodes de représentation de l'imagerie, et le rendu réaliste. La base de la méthode est de projeter une grille (soit un écran) selon une direction donnée sur la surface, suivant le point de vue choisi (figure 23). A savoir la géométrie de la pièce il peut y'avoir dans un Dixel plusieurs segments (Figure 24).



a. Intersection de Z-map avec le mouvement de l'outil b. Extension de Z-map à des Dixels

Figure 23: Simulation à base espace image [ARF, 11]

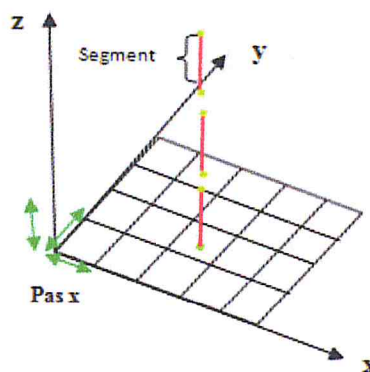
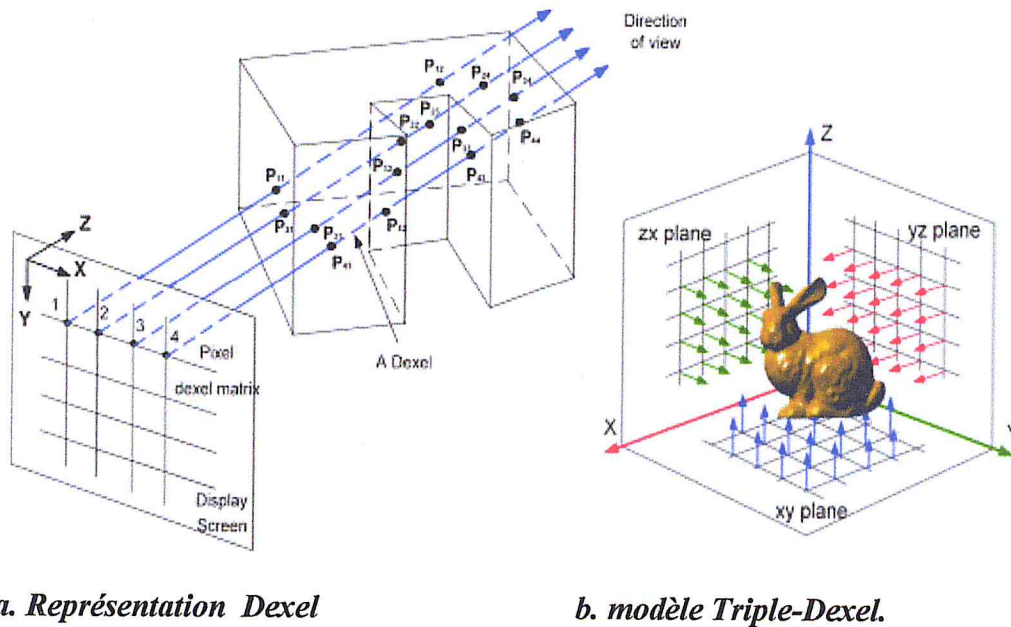


Figure 24: représentation d'un dixel sur le plan XY

5.2.3. Triple Dixel :

Le modèle simple Dixel présenté dans la section 5.2.2 est limitée dans les régions où les surfaces normales sont presque perpendiculaires à la direction du rayon (figure 25.a, figure 26.a et 26.c) [LEN, 11]. Pour surmonter ce problème, une modélisation Triple- Dixel est présentée (figures 26.b, 26.d), qui est une extension de la modélisation simple Dixel, ce modèle est construit par les rayons orientés par les trois directions orthogonales (X, Y et Z) pour discrétiser le modèle (Figure

25.b). Les auteurs de [MBZ, 10] montrent clairement que la surface générée à partir des données Triple-Dexel est plus précise que la surface reconstruite à partir des données simples Dexel, en utilisant la même résolution.



a. Représentation Dexel

b. modèle Triple-Dexel.

Figure 25 : Représentation en Dexel et en Triple-Dexel [LEN, 11].

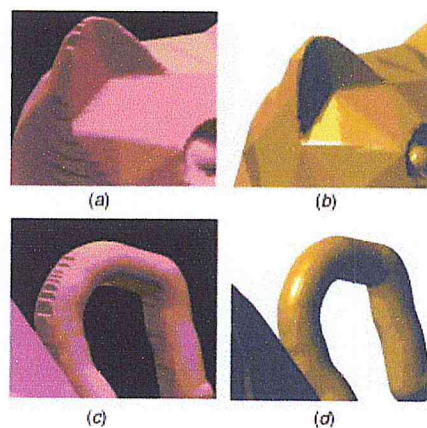


Figure 26 : Comparaison entre les données en simple-Dexel et en Triple-Dexel [MBZ, 10].

La modélisation Triple-Dexel est parfaitement adaptée pour les applications de simulation à base graphique en temps réel tels que la simulation d'usinage et la sculpture virtuelle. Dans [HSY, 93], les auteurs présentent une méthode de reconstruction de surface à partir des données Triple-Dexel. La méthode développée est plus rapide et précise que la méthode de Voxel [HSY, 93].

A partir de cette étude restreinte sur les modèles de représentation 3D des pièces à usinés, nous avons opté pour le modèle Triple-Dexel qui représente une partie dans l'ensemble de traitement de la simulation d'usinage.

6. Analyse :

L'étude des propositions adaptée concernant la simulation d'usinage au niveau de CDTA nous semble non adoptée. Les techniques de représentation volumique comme Dexel ne conviennent pas à notre cas. Cette technique d'approche ayant des limites compte tenu du fait qu'elle ne prend pas en charge de manière cohérente, les courbures et les évidements de la pièce à usiner. La surface reconstruite par simple Dexel n'est pas très précise elle ne permet pas de visualiser correctement la surface de la pièce (Figure 27).

La technique de représentation en Triple-Dexels nous offre une meilleure solution pour le traitement de notre sujet, s'appuyant quant à lui sur un usinage 5axes (Figure 28).

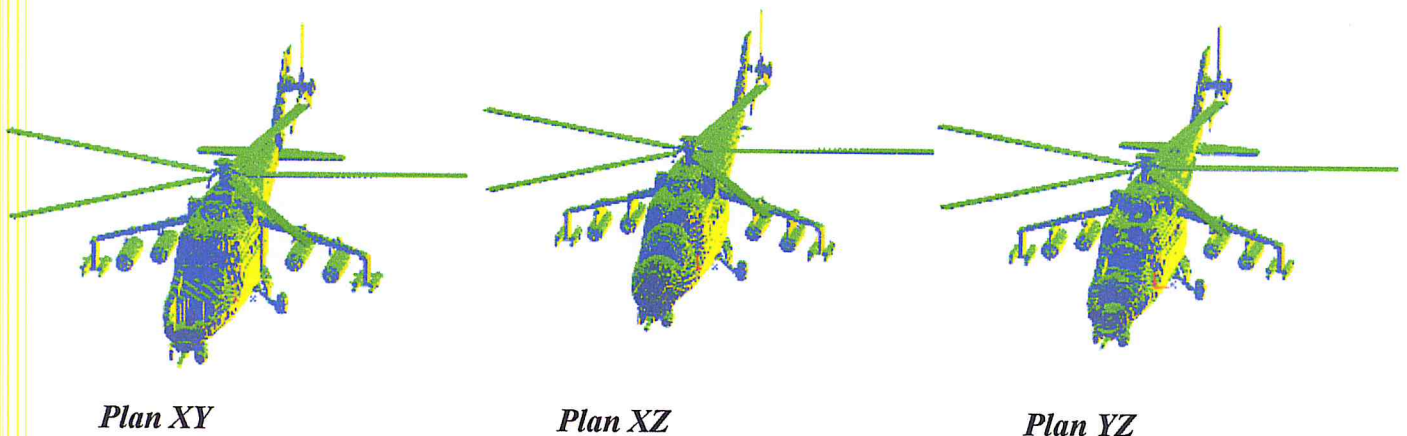


Figure 27 : Représentation du simple Dexel dans chaque plan.



Figure 28 : Représentation du modèles Triple-Dexel.

7. Conclusion :

La réalisation des formes complexes nécessite toute une chaîne pour sa mise en œuvre. Le concepteur prend en compte les données du cahier de charges en les introduisant dans un logiciel de CAO, afin d'obtenir un modèle géométrique qui sera transféré à la FAO sachant qu'il existe deux structures du système CFAO : couplé et découplé, ce dernier nécessitant un format d'échange (ex : STL) qui sera transmis au processus de la FAO qui va permettre de générer la trajectoire d'outil en format neutre résultant les CL-FILE ou CC-FILE qui sont obtenus selon les paramètres de guidage choisis, le fichier CL obtenu doit être post traité pour pouvoir être lu par la commande numérique, cette dernière est exprimée en code G. Concernant la simulation d'usinage des surfaces gauches, nous avons évoqué la simulation géométrique à l'échelle macroscopique et la technique de représentation volumique Triple dexel.

CHAPITRE II

CONCEPTION

1. Introduction :

Dans le précédent chapitre nous avons présenté une situation de l'état de l'art et des généralités utiles pour la compréhension de notre travail. Le second chapitre sera consacré à cerner les problèmes et déterminer les objectifs menant à une bonne conception d'une application répondant aux attentes et besoins de l'ingénieur. Les démarches et étapes pour réaliser ce travail se dérouleront comme suit.

2. Analyse des besoins :

L'Analyse des besoins est une étape où nous présentons de manière générale notre étude, situons ses problèmes et ses objectifs. Ensuite nous proposons des solutions.

2.1. Présentation du sujet :

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

Dans ce projet, nous nous intéressons à la simulation d'enlèvement de matière en usinage 05 axes par un outil hémisphérique et une pièce de forme complexe modélisée par des Triple-Dexels. Le travail consiste à proposer et implémenter une méthodologie permettant la simulation de l'opération d'enlèvement de la matière générée par un outil hémisphérique, pour chaque déplacement de l'outil le long du trajet d'usinage.

2.2. Problématique :

L'usinage par enlèvement de matière est encore à ce jour le procédé de fabrication le plus répandu dans les industries mécaniques de fabrication de formes gauches (automobile, aéronautique, moules...etc.). Ces formes géométriques complexes sont usinables sur des fraiseuses à commande numérique à 05 axes et d'autres. Ces pièces doivent répondre à des exigences fonctionnelles et/ou de style, ce qui impose une attention particulière dans leur mise en production. Les pièces qui ne sont pas conformes sont rejetées cela provoque une perte en productivité. Avant de passer à l'usinage réel sur machine-outil, il est indispensable de passer par la simulation virtuelle de l'usinage afin de vérifier la rugosité de la surface finie, et que les tolérances exigées par l'ingénieur sont respectées.

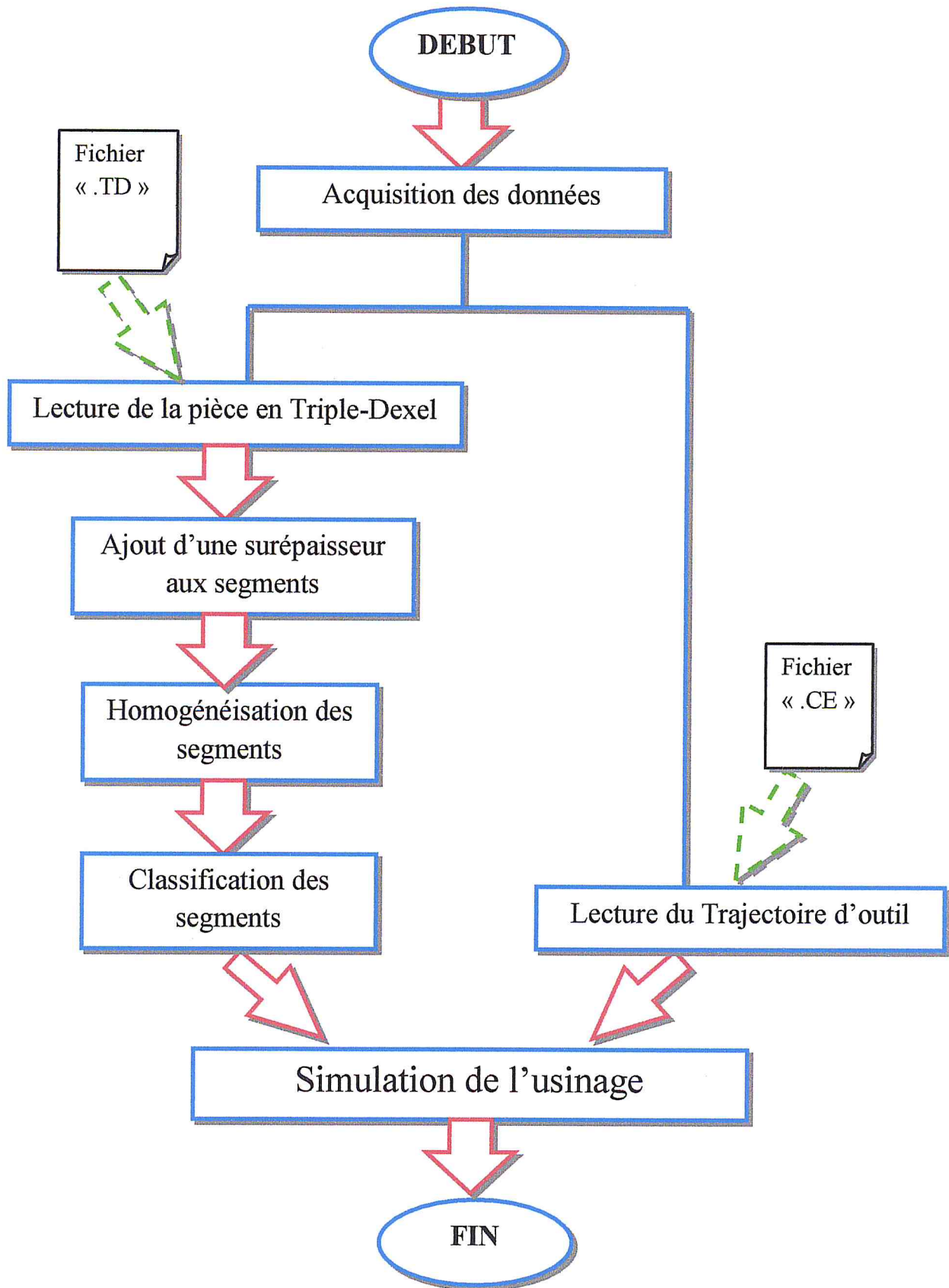
2.3. Objectifs du travail :

L'objectif de ce travail est le développement d'un module logiciel graphique et interactif, sous Windows, permettant la simulation d'enlèvement de matière en usinage 05 axes de pièces de formes complexes avec un outil hémisphérique, afin de vérifier la tolérance de l'usinage du modèle de la pièce finie, exigée par l'ingénieur. Cet objectif peut être subdivisé en plusieurs sous objectifs :

- ✓ Exploitation de l'existant :
 1. Récupération du modèle CAO de la pièce à usiner en Triple-Dexels.
 2. Récupération du trajet d'usinage approprié au modèle de la pièce.
- ✓ Simulation d'usinage :
 1. Optimiser le temps d'usinage en classifiant les Triple-Dexels.
 2. Calculer les intersections entre l'outil et les Triple-Dexels pour chaque point de trajet.

2.4. Solution proposée de la problématique :

Afin d'étudier les résultats de l'usinage des surfaces gauches et vérifier le respect de tolérance, nous allons développer un module logiciel pour la simulation d'enlèvement de matière en finition, sur des fraiseuses 5axes, des surfaces complexes fonctionnant sous environnement Windows. Les démarches de la simulation d'enlèvement de matière sont montrées par l'organigramme 1 en ordre chronologique:



Organigramme 1 : Architecture globale du système.

2.4.1. Diagramme d'états transition :

Les segments passent par différents évènements qui changent leurs états, avec le diagramme d'état transition nous montrons le changement d'état d'un segment en réponse à des événements.

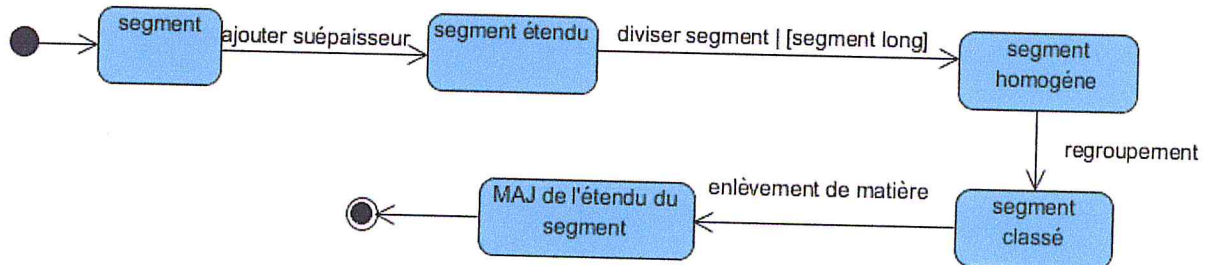
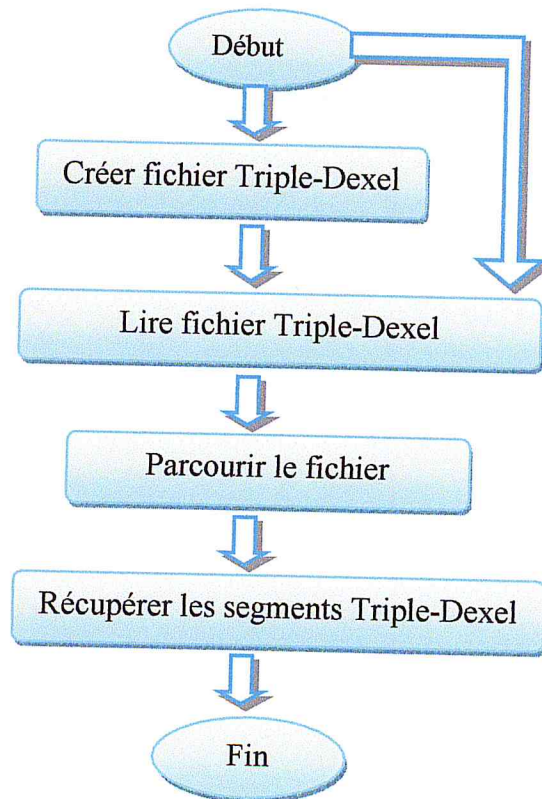


Figure 1 : diagramme d'états transition.

2.4.2. « Lecture de la pièce en Triple-Dexels » :

Dans le cas où il n'existe pas le fichier Triple-Dexels du modèle de la pièce désirée par l'ingénieur, la création du fichier Triple-Dexels est effectuée en passant par l'application <Triple Dixel > qui permet de lire le fichier STL et créer les Triple-Dexels. A ce stade nous récupérons les données pour créer le fichier Triple-Dexels approprié au modèle de la pièce STL. Ensuite passer à la lecture, où la structuration des Triple-Dexels. Nous résumons ces étapes par l'organigramme 2.



Organigramme 2 : Organigramme de création et lecture des fichiers Triple-Dexel.

L'écriture d'un fichier Triple-Dexel se traduit par l'algorithme 1.

dexelSize : le nombre de segments dans la colonne j de la ligne i.

point debut : (x1, y1, z1).

point fin : (x2, y2, z2).

Mat : La valeur de la matière du segment, true : segment matière, false : segment non matière.

DEBUT

Lancer l'exécution de l'application « Triple Dexel »

Créer un fichier « .TD »

Ouvrir le fichier « .TD » en mode écriture

Si le fichier est créé correctement

Début

Ajouter les pas de grille choisis

Lecture de la liste des dexel dans un plan

Pour i vari de 0 à nbr de lignes

Pour j vari de 0 à nbr de colonnes

Lire: dexelSize

Si dexelSize != 0

Écrire dexelSize.

Pour k vari de 0 à dexelSize

Lire : mat

Si mat = true

Écrire (x1, y1, z1), (x2, y2, z2), 1

Sinon

Écrire (x1, y1, z1), (x2, y2, z2), 0

Fin pour.

Fin pour.

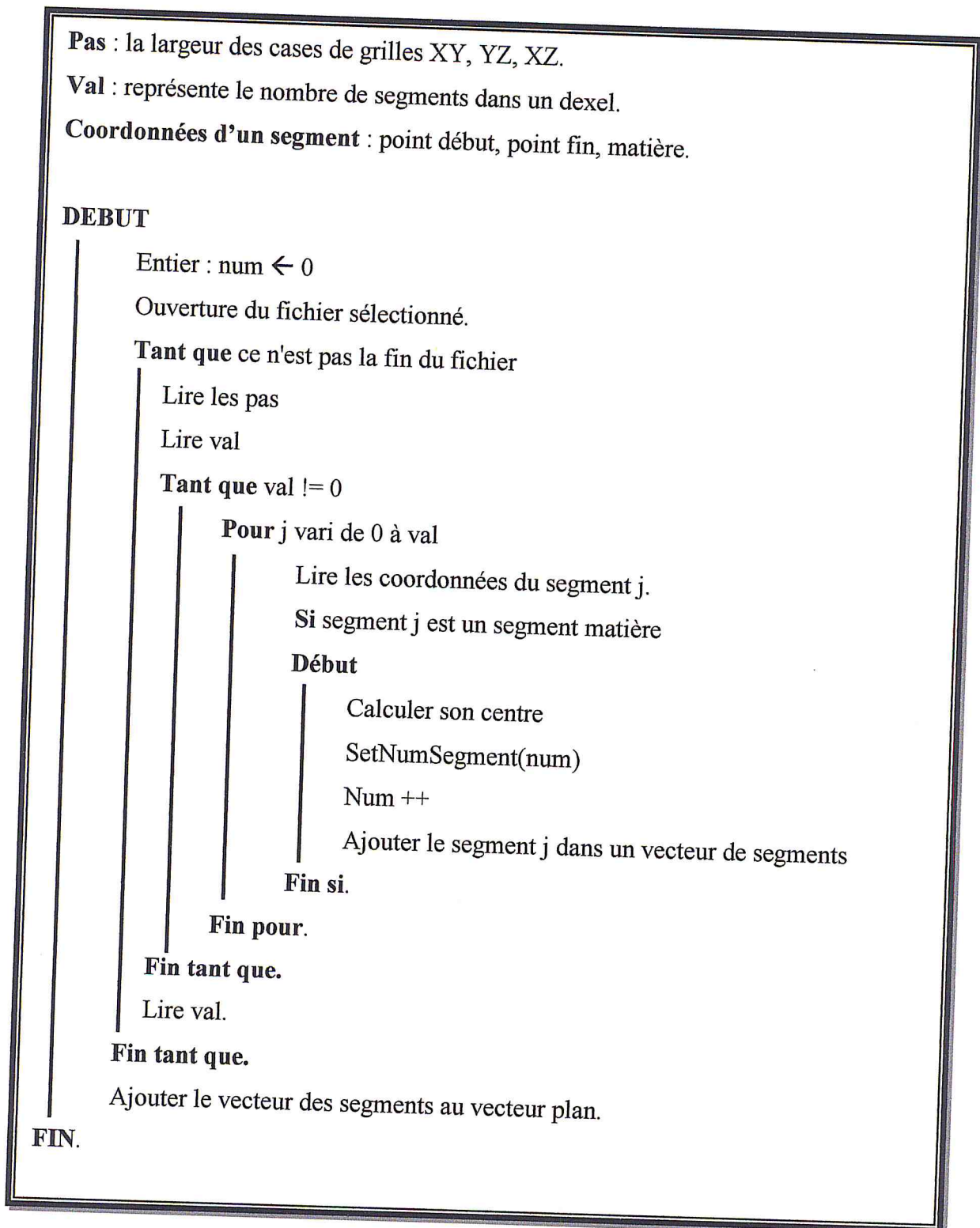
Fin pour.

Ecrire 0 a la fin du plan pour le séparer du prochain plan

Fin si.

FIN.

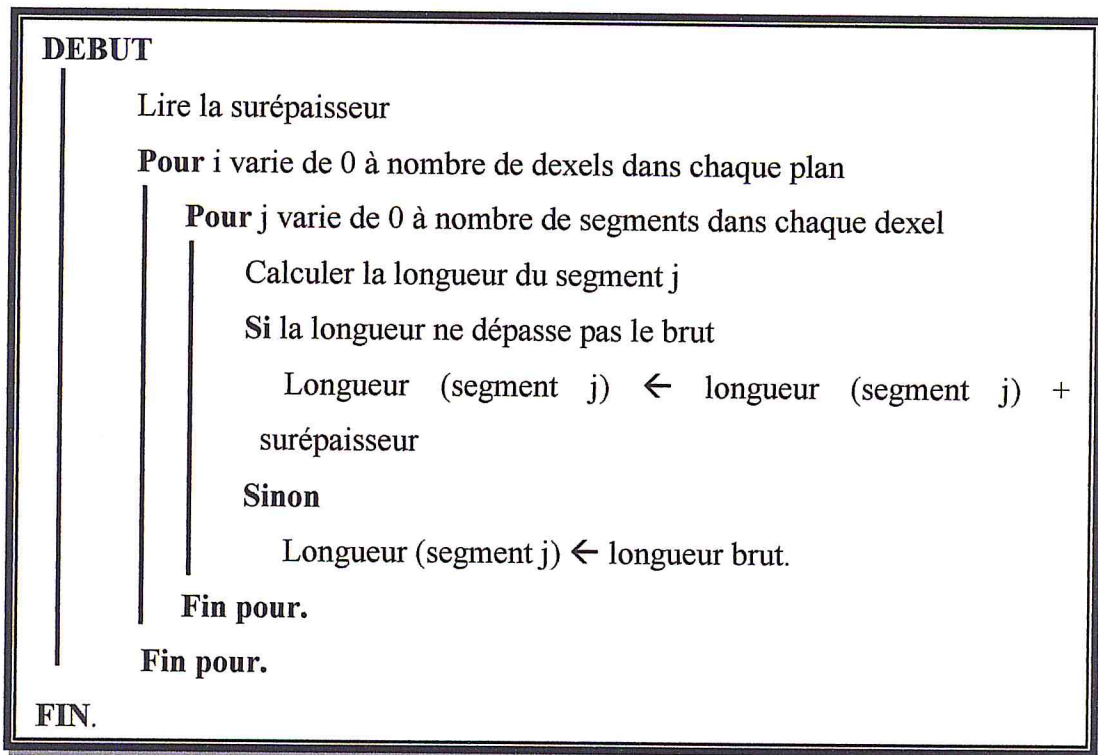
La lecture d'un fichier Triple-Dexel se traduit par l'algorithme 2.



Algorithme 2 : Lecture du fichier Triple-Dexel.

2.4.3. « Ajout d'une Surépaisseur aux segments »:

Cette étape est une estimation de l'état de la pièce avant la finition. Les Triple-Dexels lus du fichier « .TD » représentent le modèle de la pièce final, pour atteindre l'étape avant la finition, il est nécessaire d'ajouter une surépaisseur à chaque Triple-Dexels, comme le démontre l'algorithme 3.



Algorithme 3 : Ajouter surépaisseur.

2.4.4. « Homogénéisation des segments » :

Les formes de pièces où la différence entre les longueurs des segments est grande, lors la classification de ces segments nous avons remarqué que les enveloppes des groupes de segments ne sont pas bien structurés (enveloppe dans une autre...) (figure 2). En vu de régler ce problème nous avons divisé les segments qui ont une longueur supérieur ou égales à 1/3 la longueur du brut (vois algorithme 4), pour les rendre plus homogènes avec les segments courts, nous obtenons une bonne classification à la fin (figure 3).

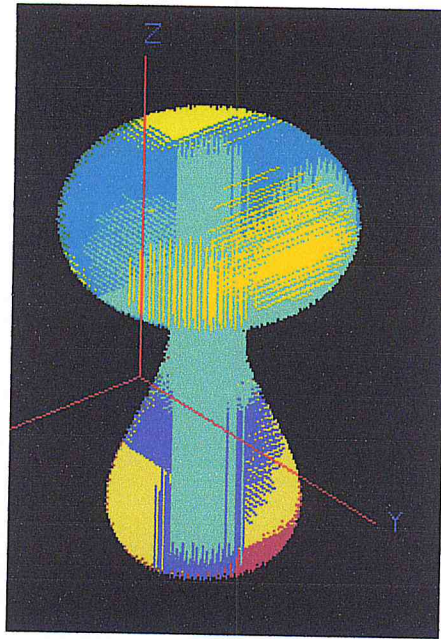
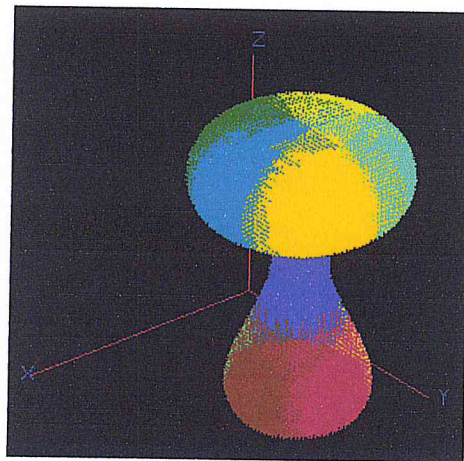
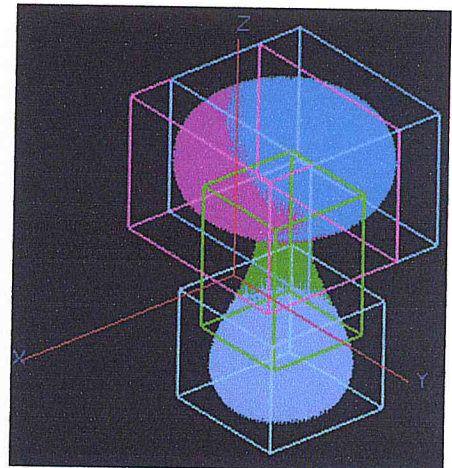


Figure 2 : représentation de la classification des segments avant l'homogénéité.



a. Représentation des segments homogènes



b. Représentation des enveloppes des clusters

Figure 3 : représentation de la classification des segments et enveloppes de clusters après l'homogénéité.

Maxbrut : longueur maximum de la pièce.

DEBUT

Entier : numSegment \leftarrow 0

Pour i vari de 0 à nombre de dexels

Pour j vari de 0 à nombre de segments du dexel i

 Longueur \leftarrow calculerLongueur(segment i)

 Si longueur \geq maxbrut/3

Début

 Diviser le segment j sur 3

Pour k vari de 0 à 3

 CalculerCentre (segment k)

 SetNumSegment (segment k)

 NumSegment ++

 Ajouter segment k à la liste des segments

Fin pour.

Fin si.

Sinon

Début

 SetNumSegment (segment j)

 NumSegment ++

 Ajouter à la liste des segments

Fin sinon.

Fin pour.

Fin pour.

FIN.

Algorithme 4 : Homogénéisation des segments

2.4.5. « Classification des segments » :

Pour calculer l'intersection entre l'outil et les Triple-Dexels, nous devons vérifier si la sphère est en intersection avec tout les Triple-Dexels formant la pièce pour chaque point de la trajectoire.

En classifiant les segments dans des groupes, on peut vérifier l'intersection seulement avec les segments qui appartiennent aux groupes de segments qui chevauchent avec l'enveloppe de la sphère, ainsi on diminue le calcul d'intersection sphère/Triple-Dexel et d'où un gain de temps.

➤ **Définition du clustering :**

Le clustering groupe un ensemble d'objets, de manière que ces derniers, dans le même cluster, sont plus similaires les uns aux autres, que ceux des autres clusters. Le terme important ici est «similaire». D'une certaine manière, nous avons besoin de quantifier «similaire» dans le contexte du regroupement (clustering). Ceci est quantifié par une mesure de distance. Différentes techniques de clustering utilisent différentes mesures de distance pour arriver à l'objectif de former des clusters [JAC, 16]. K-means, une technique de clustering populaire et la plus utilisée.

➤ **k-means :**

❖ **Étapes k-means :**

1. Décider le nombre de cluster : K .
2. Initialiser les centroïdes μ_1, \dots, μ_k .
3. Calculer la distance entre chacune observation et les centroïdes.
4. Attribuer l'observation au cluster du centroïde le plus proche.
5. Recalculer les centres. Le centroïde est la moyenne des observations. Si le nombre d'observation de cluster change en raison de l'affectation des observations à différents groupes à l'étape 3, la moyenne ou le centroïde de ce cluster change.
6. Répétez les étapes 3 et 4 jusqu'à ce que les centroïdes ne se déplacent pas.

❖ **Adaptation de k-means :**

En adaptant l'algorithme à notre cas, classification des segments, chaque segment sera caractérisé par son point début, point fin et son centre :

1. Générer aléatoirement K centroides choisis parmi tous les segments qui existent.
2. Calculer la distance entre les centres des segments et les centres des centroides.
3. Affecter les segments aux clusters des centroides les plus proches.
4. Régénérer les nouveaux centroides. En calculant leurs coordonnées comme suit :

$$X = \frac{1}{N} \sum_{i \in 1}^N x_i$$

$$Y = \frac{1}{N} \sum_{i \in 1}^N y_i$$

$$Z = \frac{1}{N} \sum_{i \in 1}^N z_i$$

Tel que :

Chaque point (début/fin/centre) du centroïde est calculé par les points (début/fin/centre) des segments appartenant au même cluster.

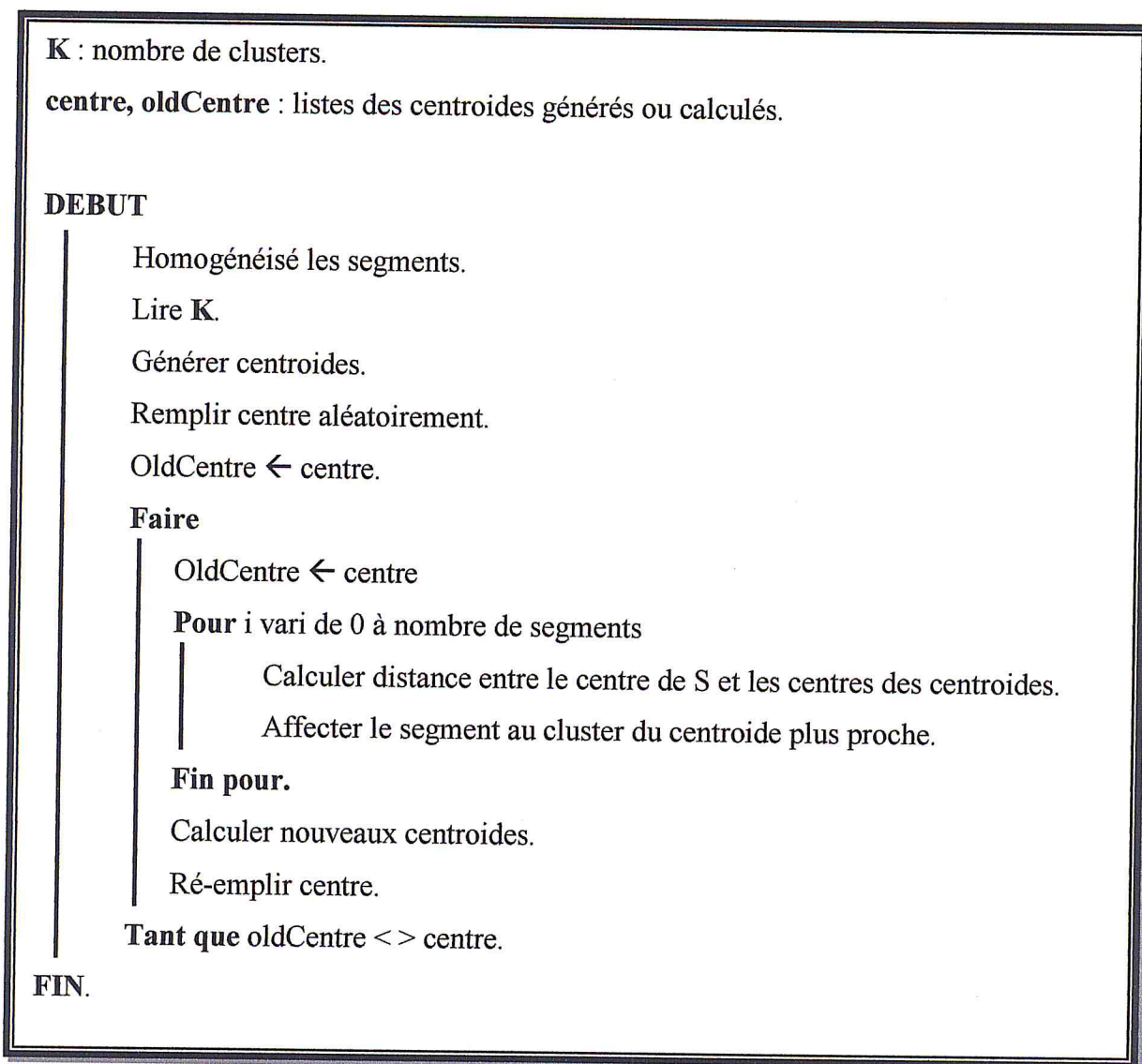
X, Y, Z : les coordonnées des points qui caractérisent le nouveau centroïde.

x_i, y_i, z_i : les coordonnées des points qui caractérisent les segments qui appartiennent au cluster du centroïde.

N : le nombre de segments qui appartiennent au cluster du centroïde.

5. L'algorithme s'arrête lorsque les groupes construits deviennent stables.

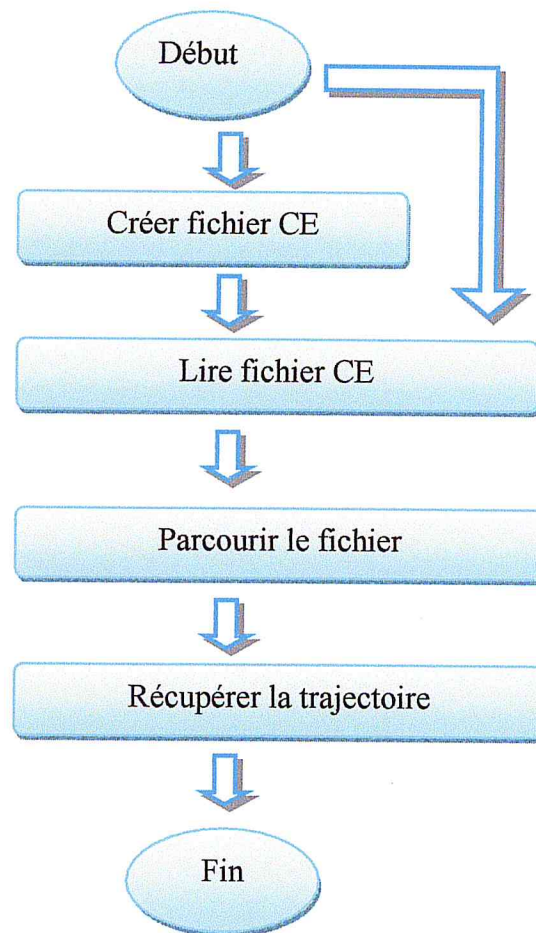
Nous avons implémenté l'algorithme du K-means comme le montre l'algorithme 5.



Algorithme 5 : Classification des segments par la méthode K-means.

2.4.6. Trajectoire d'outil :

En passant par l'application « Trajectoire d'outil », nous récupérons les données nécessaires de la trajectoire et les enregistrons dans un fichier ayant le nom de la pièce, s'il n'existe pas, sinon l'ingénieur procède à la lecture directement (organigramme 3) la lecture et l'écriture du fichier CE est démontré par les algorithmes 6 et 7.



Organigramme 3 : Organigramme lecture et écriture fichier CE.

x, y, z : les coordonnées du point centre : p.

Thêta : orientation autour du segment.

Phi : orientation vers le segment.

r : rayon d'outil dans le point p.

l : longueur d'outil au point p.

DEBUT

Exécuter l'application trajectoire d'outil.

Créer un fichier « .CE ».

Ouvrir le fichier « .CE » en mode écriture.

Si le fichier est créé correctement.

Début

Ecrire le nombre total des centres d'outil.

Pour i vari de 0 à nombre de contours.

Ecrire le nombre de centres dans le contour i.

Pour j vari de 0 à nombre de centres.

Écrire $x, y, z, \text{thêta}, \text{phi}, r, l$.

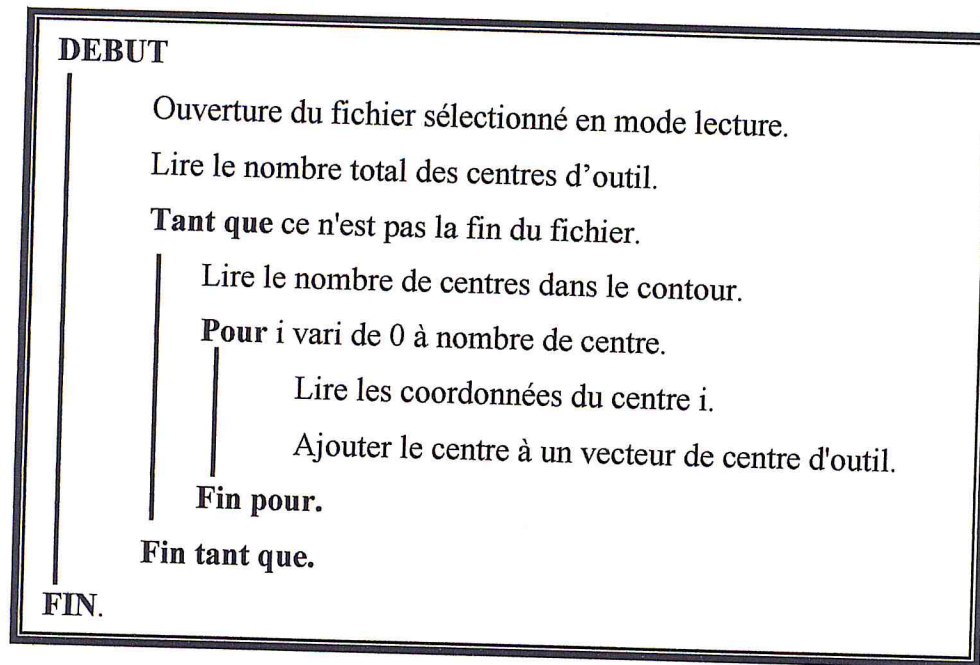
Fin pour.

Fin pour.

Fin si.

FIN.

Algorithme 6 : Ecriture du fichier trajectoire.



Algorithme 7 : Lecture du fichier trajectoire.

2.4.7. « Simulation de l'usinage »:

Après l'introduction de la tolérance d'usinage par l'ingénieur, on passe à la simulation de la pièce suivant le trajet approprié à cette dernière.

La simulation se fait en calculant l'intersection entre les segments et la sphère pour obtenir les nouvelles dimensions des segments, sachant que la pièce est divisée en clusters, chaque cluster groupe un nombre N de segments, donc à chaque point de la trajectoire on calcule l'intersection avec les segments appartenant aux clusters qui chevauchent avec la sphère au niveau de ce point.

2.4.7.1. Vérification du chevauchement sphère \cluster:

La figure (Figure 4) montre tous les différents cas du chevauchement entre sphère cluster. Les clusters chevauchant avec la sphère doivent vérifier les conditions suivantes :

- $X_{minC} < X_{maxCL}$, $X_{maxC} > X_{minCL}$
- $Y_{minC} < Y_{maxCL}$, $Y_{maxC} > Y_{minCL}$
- $Z_{minC} < Z_{maxCL}$, $Z_{maxC} > Z_{minCL}$

Tel que :

X_{minC} , Y_{minC} , Z_{minC} : les limites minimums de l'enveloppe du centre de la sphère.

X_{maxC} , Y_{maxC} , Z_{maxC} : les limites maximums de l'enveloppe du centre de la sphère.

X_{minCL} , Y_{minCL} , Z_{minCL} : les limites minimums de l'enveloppe du cluster.

X_{maxCL} , Y_{maxCL} , Z_{maxCL} : les limites maximums de l'enveloppe du centre de la sphère.

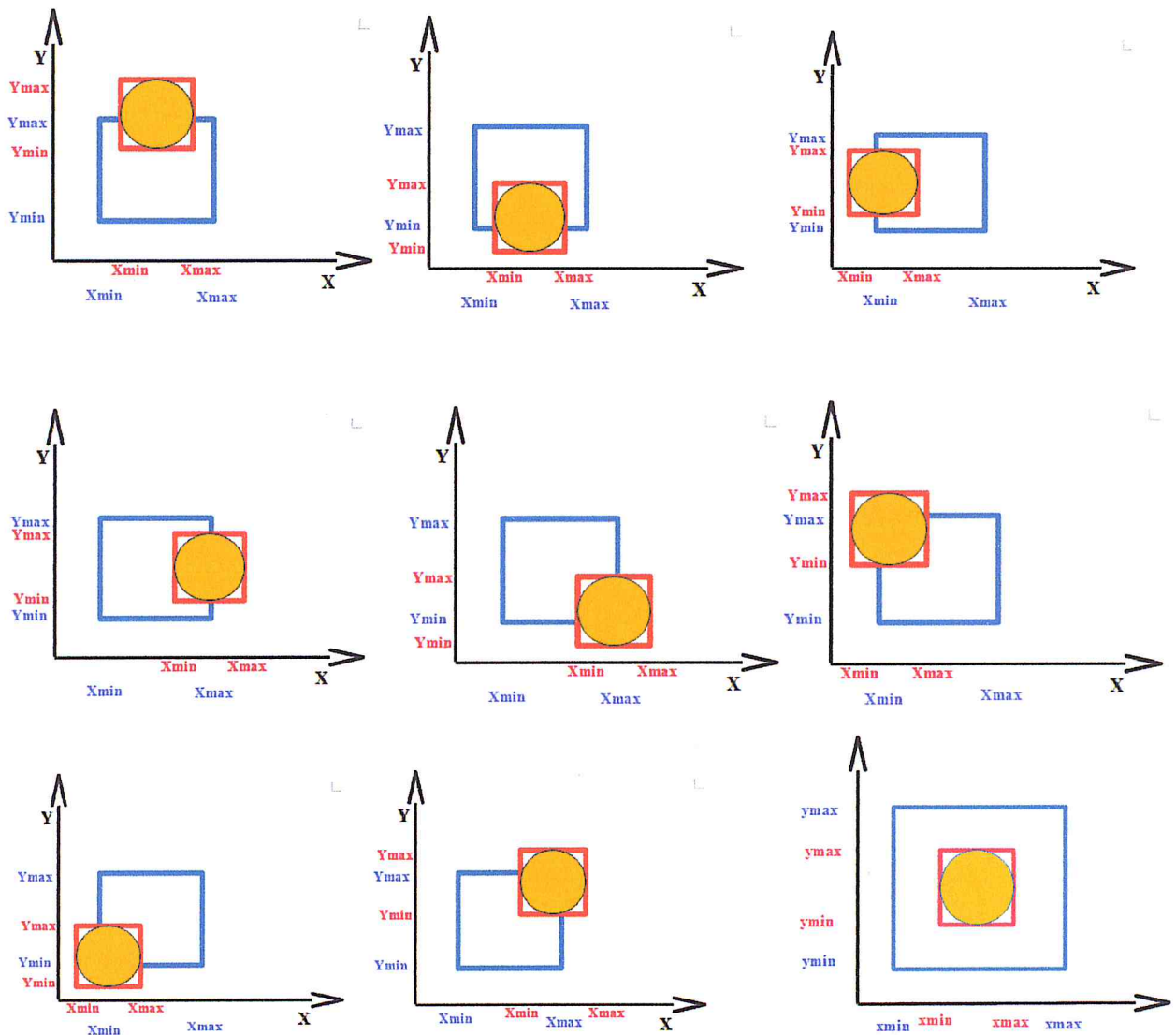


Figure 4: différents cas de chevauchement sphère cluster

2.4.7.2. Résolution de l'équation de la sphère:

Raisonnement:

Prenons exemple le cas dans le plan XY, z est la coordonnée qui se varie dans un segment s.

On a l'équation de la sphère :

$$(z - zc)^2 = R^2 - (x - xc)^2 - (y - yc)^2.$$

Tel que :

xc, yc, zc sont les coordonnées des point centre.

x, y, z coordonnées d'un point qui vérifie l'équation si il appartient à la sphère.

On pose :

$$\text{var} = R^2 - (x - xc)^2 - (y - yc)^2 \Rightarrow (z - zc)^2 = \text{var}.$$

On obtient une équation du 2eme degré:

1. Si $\text{var} < 0$, l'équation n'a pas de solutions

2. Si $\text{var} = 0$,

Il existe une seule solution: $z = zc$ alors un seul point d'intersection.

Le segment se divise en 2 segments S1, S2 tel que :

Le point fin de S1 = le point début de S2 = zc.

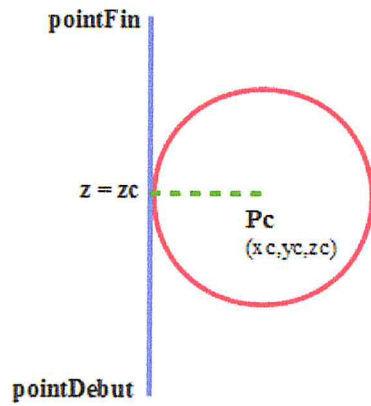
3. Si $\text{var} > 0$,

Il existe deux solutions: $z1 = \sqrt{\text{var}} + zc$ et $z2 = -\sqrt{\text{var}} + zc$ donc deux points d'intersection. On pose $z1 > z2$.

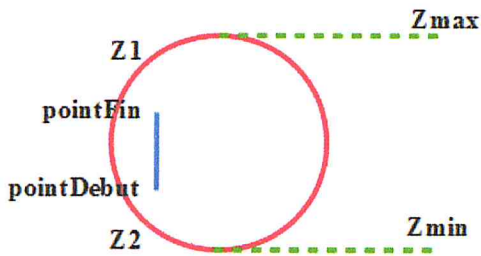
Le segment se divise en 2 segments S1 et S2 tel que :

Le point début de S1 = z2 et le point fin de S2 = z1.

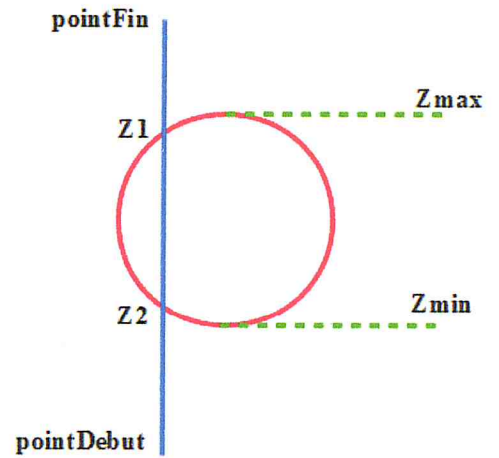
Dans la figure 5 nous avons montré les différents cas d'intersection entre une sphère et un segment.



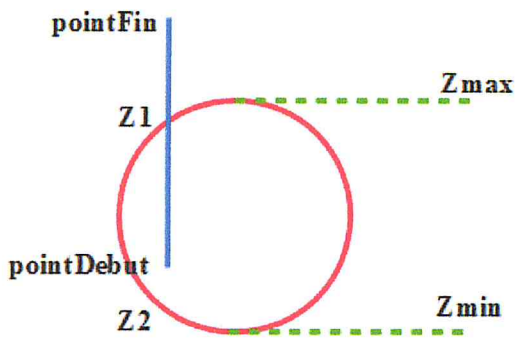
Avec un point



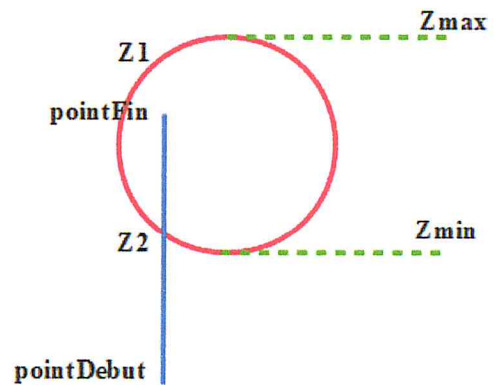
Avec deux points qui n'appartiennent pas au même segment.



Avec deux points qui appartiennent au même segment.



Avec un point appartenant au segment supérieur.



Avec un point appartenant au segment inférieur.

Figure 5 : différents cas d'intersection sphère segment.

Les étapes d'intersection entre une sphère et un segment est dans l'algorithme 8.

S : segment croisant avec la sphère.

S1, S2 : segments résultant de l'intersection de S avec la sphère.

Liste : la nouvelle liste de segments après simulation.

DEBUT

Calculer var

Si var \geq 0

Début

// Il existe un point d'intersection

Si var = 0

Debut

Affecter la valeur de S à S1 et S2.

S1.pointFin.z \leftarrow zc

S2.pointDebut.z \leftarrow zc

Supprimer S de liste

Ajouter S1, S2 à liste.

Fin si

// Deux points d'intersection qui n'appartiennent pas au segment (segment est à l'intérieur de la sphère)

Si z1 \geq S.pointFin.z et z2 \leq S.pointDebut.z

Supprimer le segment

// Il existe deux points d'intersection appartenant au mm segment

Si z1 < S.pointFin.z et z2 > S.pointDebut.z

Début

Affecter la valeur de S à S1 et S2

S1.pointFin.z \leftarrow z2.

S2.pointDebut \leftarrow z1.

Fin si.

// Il y a un point d'intersection appartenant au segment inferieur

Si z1 \geq S.pointFin.z et z2 > S.pointDebut.z et z2 \leq S.pointFin.z.

.....

```

.....

Début
  S.pointFin.z = z2.
  MAJ du segment.
Fin si.
// Il y a un point d'intersection appartenant au segment supérieur
Si z1 < S.pointFin.z et z1 >= S.pointDebut.z et z2 <= S.pointDebut.z
  Début
    s.pointDebut.z = z1.
    MAJ du segment.
  Fin si.
Fin si.
FIN.

```

Algorithme 8 : Calculer intersection sphère segment.

2.4.7.3. Vérification du respect de la tolérance :

La tolérance d'usinage est un chiffre introduit par l'ingénieur en mécanique afin d'accepter les erreurs de l'enlèvement de matière qui ne dépasse pas cette tolérance (Voir figure6 tel que $\|Segment A\| = \|Segment B\| + \text{surépaisseur}$).

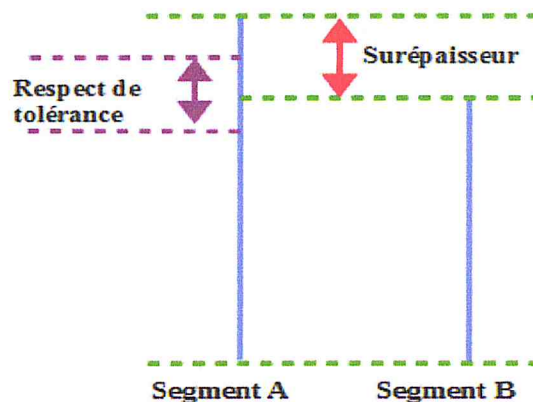


Figure 6 : Tolérance d'usinage.

Dans la figure 7 nous démontrons les différents cas d'usinage :

- Les segments en vert sont les segments usinés ne dépassant pas la tolérance (sont dans l'intervalle orange).
- Le segment en jaune est celui non bien usiné en enlevant moins de matière, dépasse la tol2.
- Le segment en rouge est celui non bien usiné en enlevant trop de matière, dépasse la tol1.

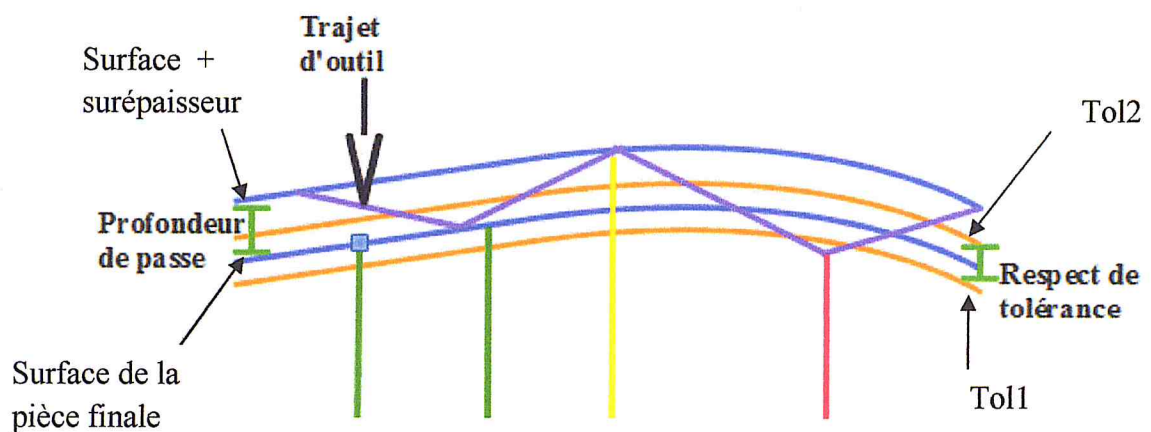


Figure 7 : Démonstration d'usinage de différents segments.

L'algorithme 9 traduit comment nous avons vérifié le respect de tolérance.

oldLongueurS : La longueur du segment avant l'enlèvement de matière.

longueurS : La longueur du segment après l'enlèvement de matière.

Surep : la surépaisseur ajoutée aux segments avant la simulation.

Tol1, tol2 : limites de la tolérance de l'usinage introduit par l'ingénieur.

Tel que $tol1 = tol2 = \text{tolérance} / 2$.

.....

.....

DEBUT

 // Comparaison avec la tolérance en moins ensuite tolérance en plus

Si longueurS \geq tol1 et longueurS \geq tol2

 //Tolérance vérifiée.

 Ajouter le segment à la liste des segments bien usinés.

Sinon //Tolérance dépassée

Si longueurS < tol1 // enlevé trop de matière.

 Ajouter le segment à la liste des segments mal usinés en enlevant plus de matière.

Si longueurS > tol2 // enlevé moins de matière.

 Ajouter le segment à la liste des segments mal usinés en enlevant moins de matière.

FIN.

Algorithme 9 : Vérification de la tolérance.

L'algorithme 10 traduit la simulation d'usinage en utilisant les algorithmes situés en dessus.

DEBUT

 Lancer la simulation.

Pour i vari de 0 à nombre centres total

 Vérifier le chevauchement entre la sphère au centre i et les clusters.

Pour j vari de 0 à nombre clusters chevauchants.

Pour k vari de 0 à nombre de segments dans le cluster j.

 Calculer intersection sphère segment.

 Vérifier le respect de tolérance.

Fin pour.

Fin pour.

Fin pour.

FIN.

Algorithme 10 : Simulation.

3. Modélisation de l'application avec UML :

La conception de développement d'une application nécessite une méthodologie permettant de décrire les fonctionnalités, le comportement et les détails d'un système et son déroulement.

Ainsi, plusieurs langages de modélisation orientés objets on vus le jour puis certains d'entre eux ont été unifiés (OMT, OOSE et BOOCH) sous l'appellation d'UML afin de prendre en charge le développement des applications et logiciels devenant de plus en plus compliqués.

L'utilisation d'UML permet de :

- Faciliter l'analyse, la compréhension et réduit la complexité d'un système.
- Permet de simuler le système et reproduire ses comportements.
- Penser objet dès le départ.

Dans notre projet, nous avons utilisé les diagrammes suivants :

- Diagramme de cas d'utilisation.
- Diagramme de séquence.

3.1. Diagramme de cas d'utilisation :

Les cas d'utilisation (*use case*) permettent de représenter le fonctionnement du système vis-à-vis de l'utilisateur, c'est donc une vue du système dans son environnement extérieur.

Dans ce qui suit nous allons présenter les différentes fonctionnalités de notre système, nous commençons par le diagramme de cas d'utilisation générale qui donne une vue globale de fonctionnement du système.

3.1.1. Diagramme de cas d'utilisation général :

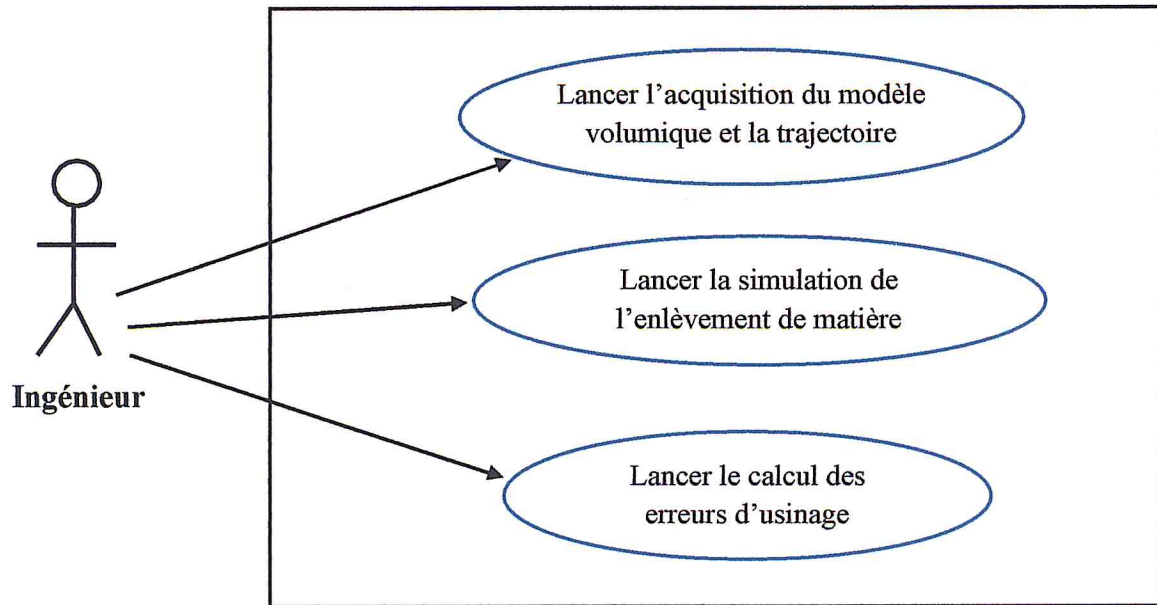


Figure 8: Diagramme cas d'utilisation général.

3.1.2. Diagramme de cas d'utilisation : Acquisition du modèle volumique et la trajectoire d'outil de la pièce:

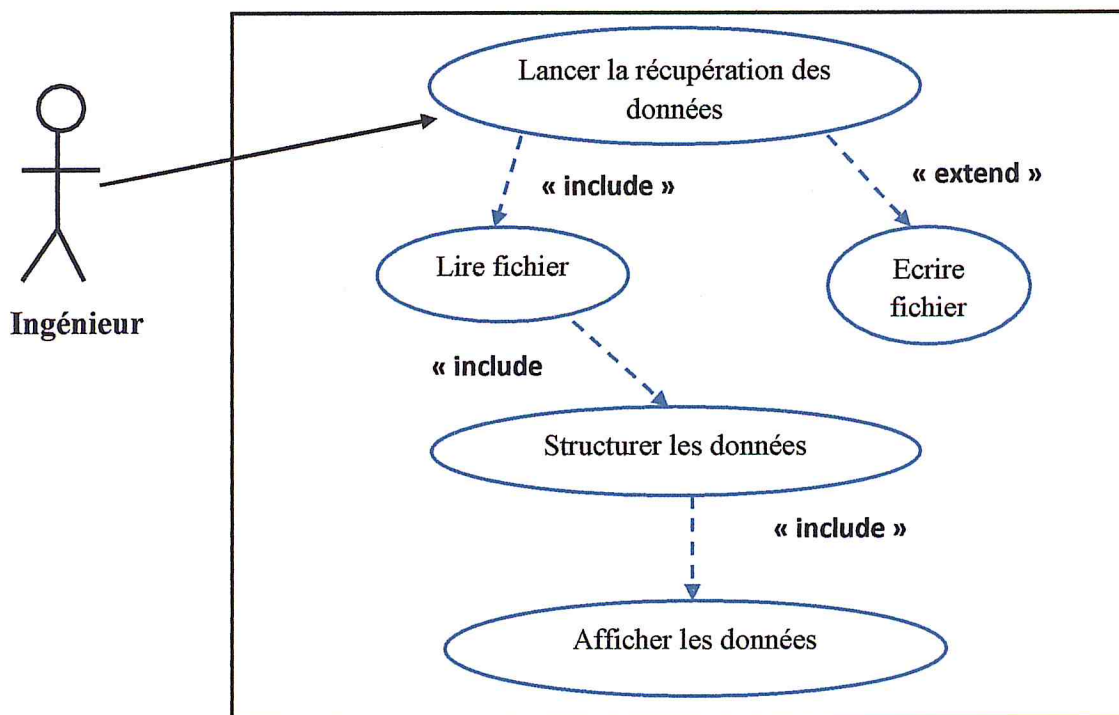


Figure 9: Diagramme cas d'utilisation Acquisition du modèle volumique et la trajectoire d'outil de la pièce.

3.1.3. Diagramme de cas d'utilisation : simulation d'enlèvement de matière :

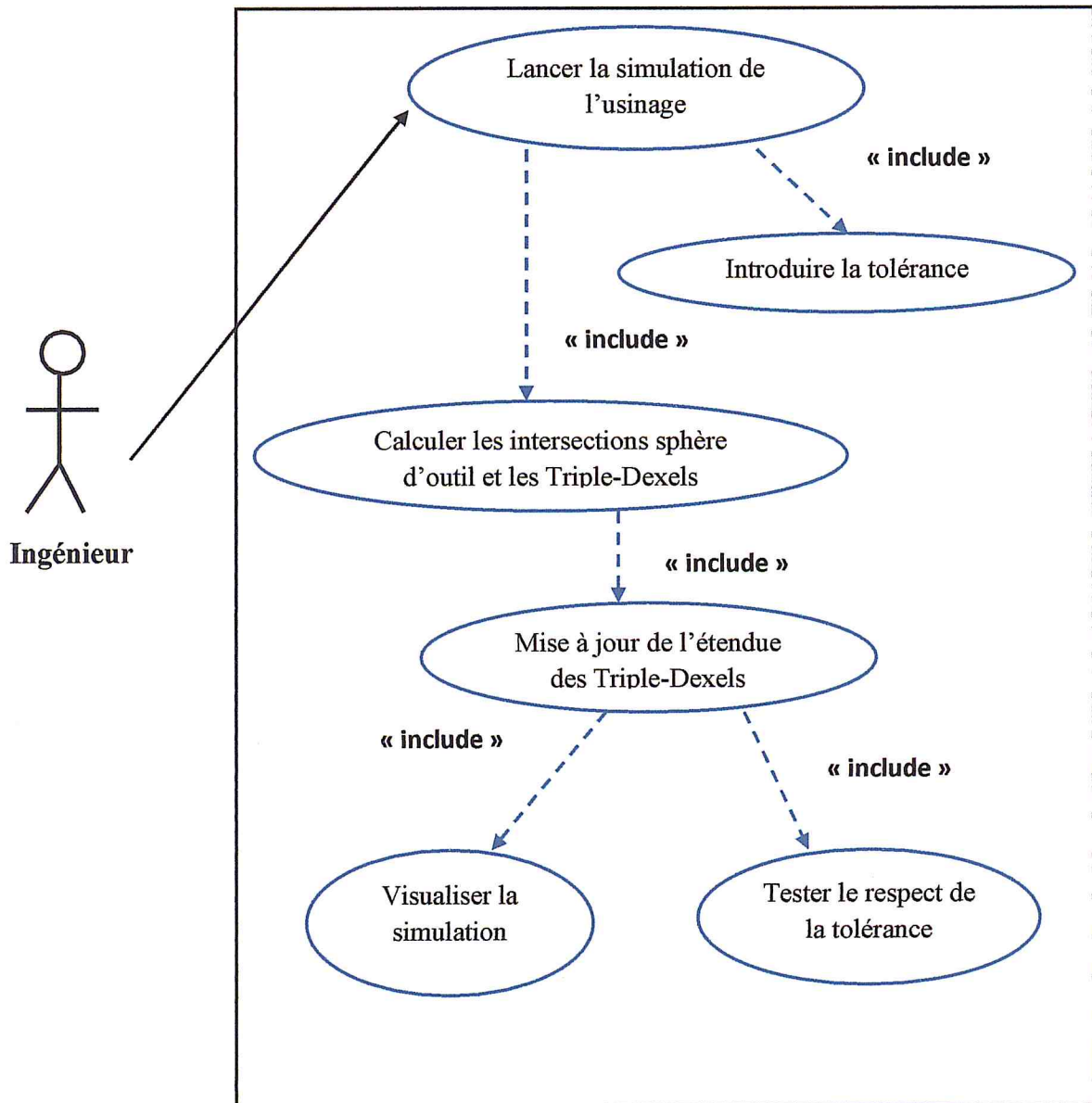


Figure 10: Diagramme de cas d'utilisation Simulation d'enlèvement de matière.

3.1.4. Diagramme de cas d'utilisation : le calcul des erreurs d'usinage:

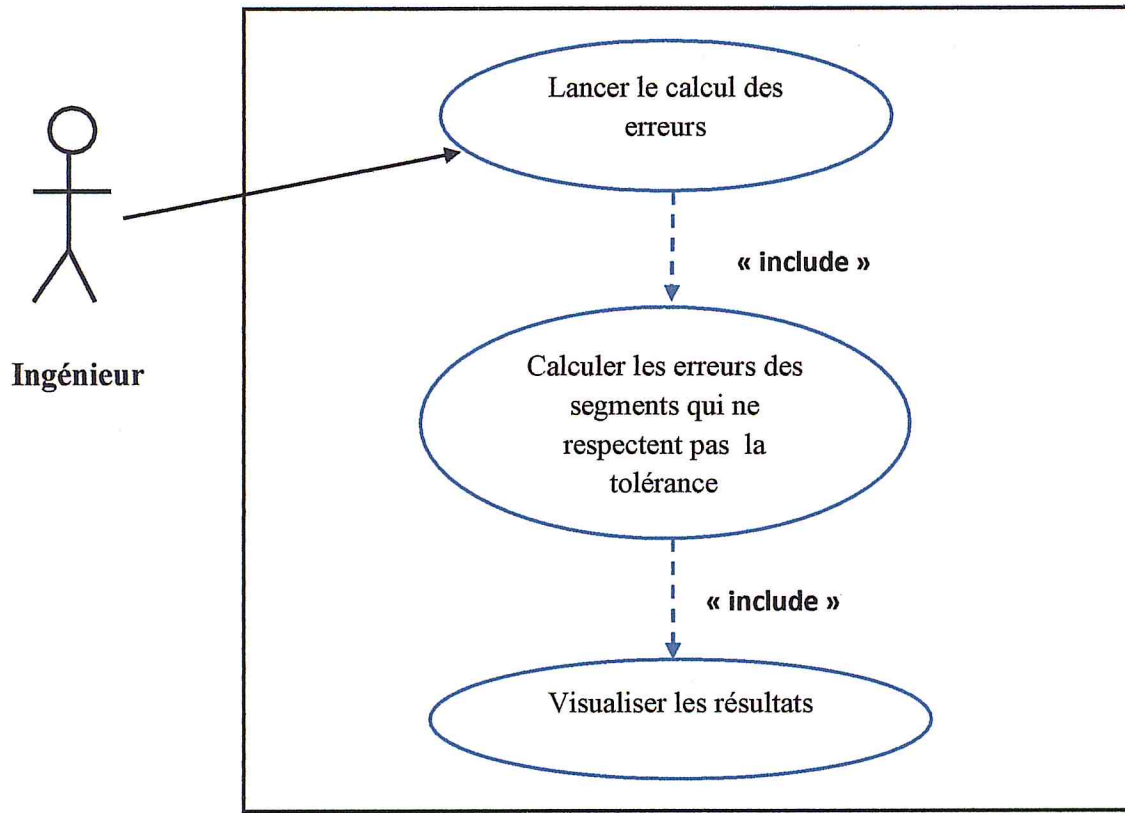


Figure 11: diagramme de cas d'utilisation vérification de la tolérance.

3.2. diagramme de séquence :

Les diagrammes de séquences permettent de décrire COMMENT les éléments du système interagissent entre eux et avec les acteurs.

- Les objets au cœur d'un système interagissent en s'échangeant des messages.
- Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

3.2.1. Récupération et structuration des Triple-Dexels :

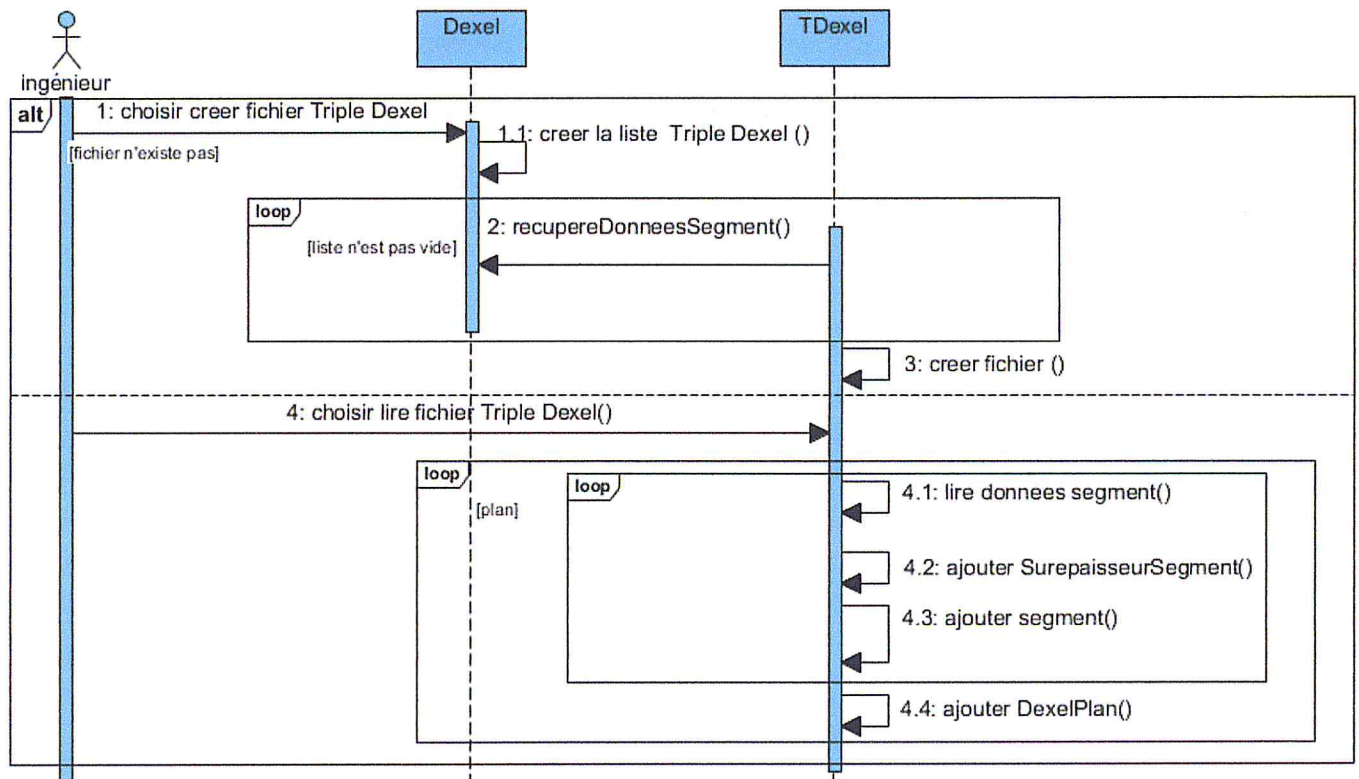


Figure 12 : diagramme de séquence Récupération et structuration des Triple Dexel.

3.2.2. Récupérer et structurer les points centre d'outil de la trajectoire :

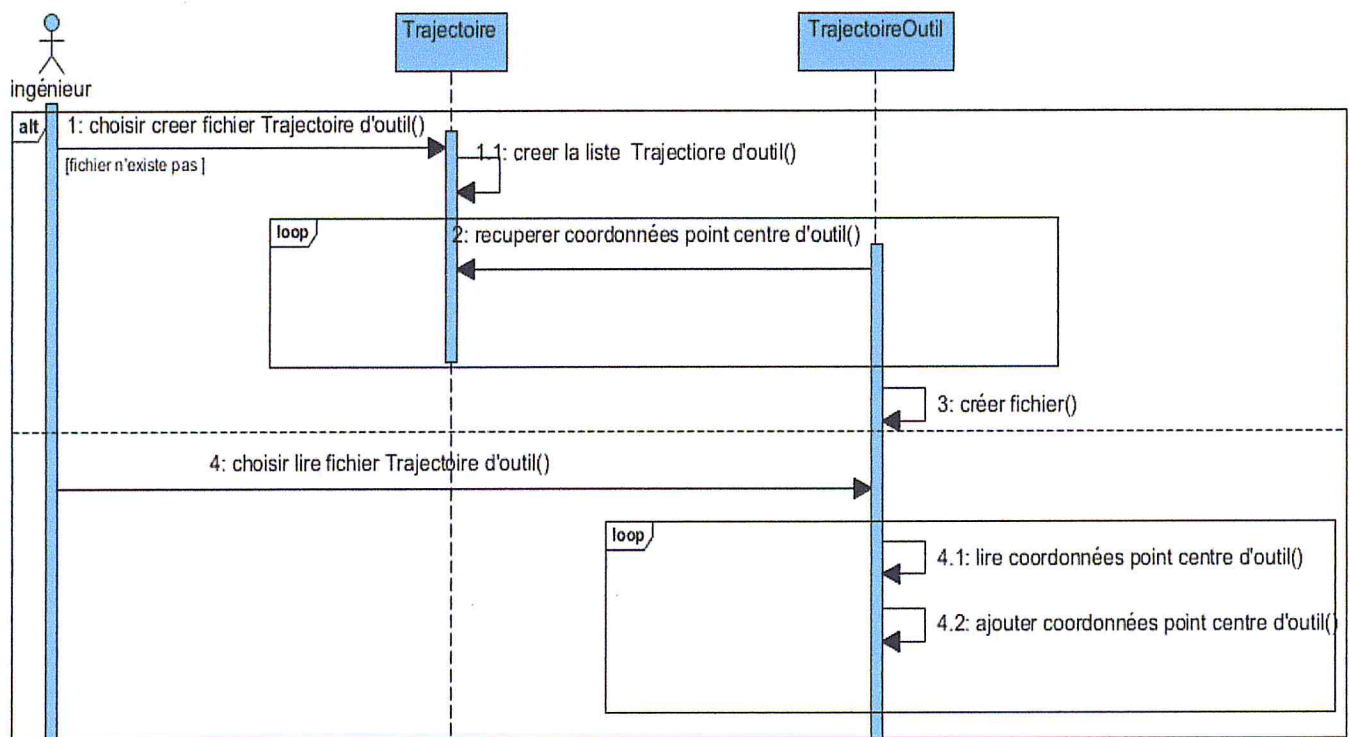


Figure 13 : diagramme de séquence Récupérer et structurer les points centre d'outil de la trajectoire.

3.2.3. Simulation d'enlèvement de matière :

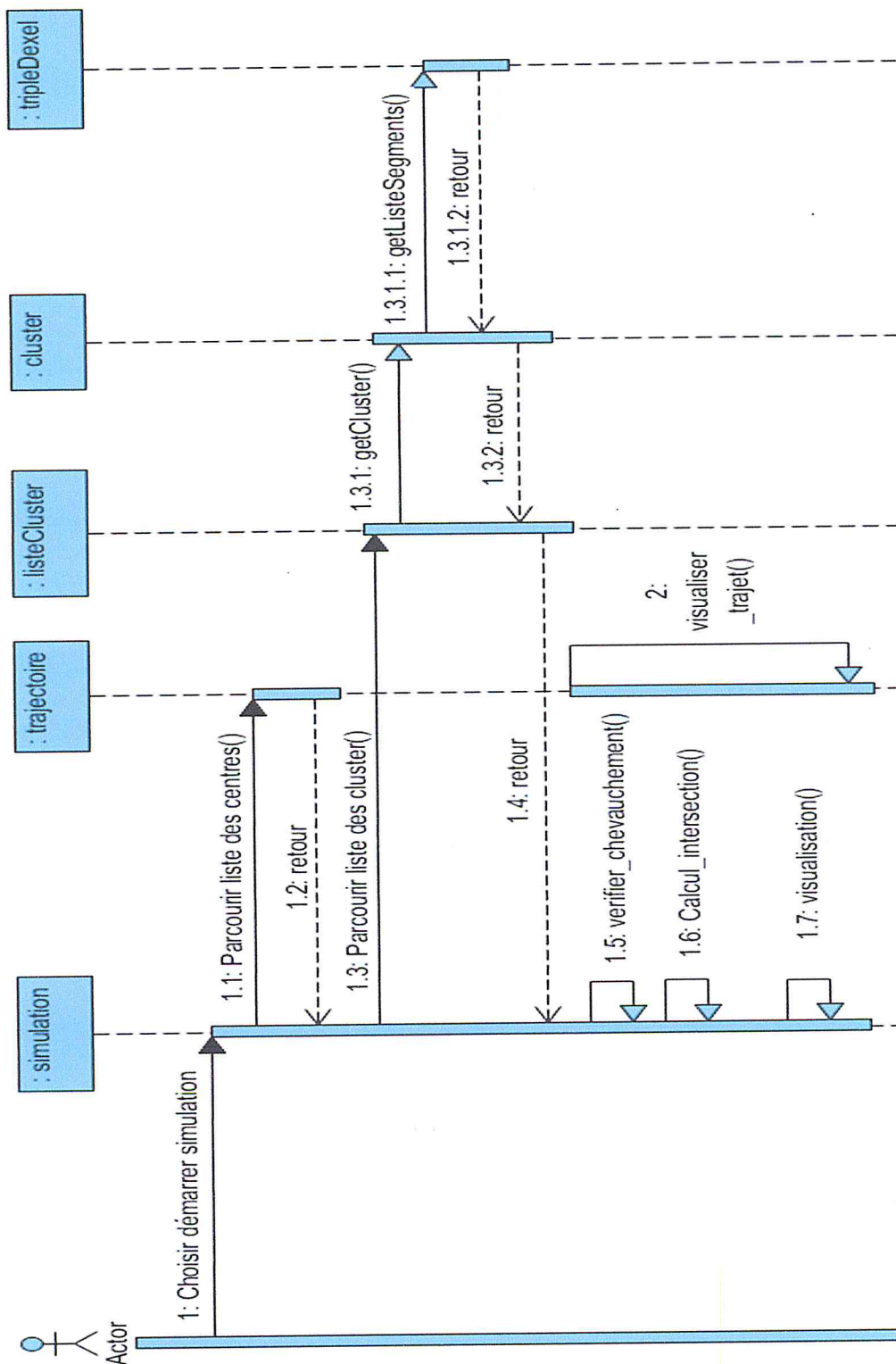


Figure 14 : diagramme de séquence de simulation d'enlèvement de matière.

4. Conclusion:

Dans ce chapitre, nous avons définis la conception de notre application en présentant la problématique, les différents objectifs à atteindre et les algorithmes développés. Ensuite nous présentons la structure de notre application et son déroulement en utilisant le modèle orienté objet et le langage de modélisation UML.

CHAPITRE III

IMPLEMENTATION ET EXPERIMENTATION

1. Introduction :

La conception et la proposition des solutions ayant été traitées par le chapitre précédent. Le présent chapitre a pour objectif : en premier de présenter le diagramme de paquetage mettant en relief des dépendances entre paquetages et d'implémenter l'interface et les fenêtres de l'application, ensuite procéder aux tests sur différentes pièces complexes. Cette application a été conçue avec le logiciel Builder C++ et la bibliothèque graphique OpenGL.

2. Diagramme de paquetage :

Un paquetage est un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML, il sert à représenter les dépendances entre paquetages.

Dans notre projet, on aura besoin de deux paquetages :

- Paquetage Triple-Dexel.
- Paquetage Trajectoire d'outil.

Le diagramme de paquetage est présenté comme suit (figure 1):

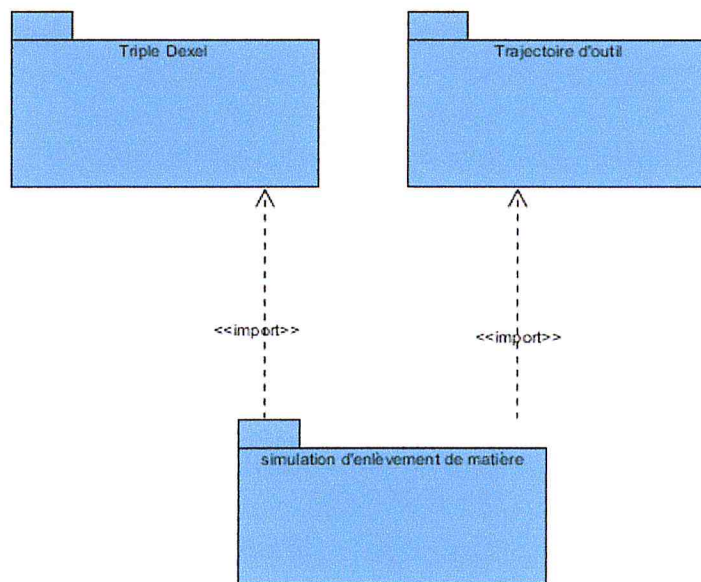


Figure 1:diagramme de paquetage.

2.1. Paquetage Triple Dixel :

Fait pour modéliser les pièces en Triple-Dexels. Les pièces sont représentées par des dexels, chaque dixel est constitués de plusieurs segments, « segments matière » qui représente la partie où se trouve la matière dans la pièce et des « segments non matière » qui représente la partie vide de la pièce. La récupération des Triple-Dexels nécessite les classes :

➤ **Classe cellule :**

Cellule
+centre: Pnt +pmax: Pnt +pmax_droite: Pnt +pmin: Pnt +pmin_droite: Pnt
+Cellule(Pnt, Pnt, Pnt, Pnt, Pnt) +Cellule() +dessiner_cellule(int, int): void +dessiner_centre_cellule(int, int): void +dessiner_droite_cellule(int, int): void +getCentre(): Pnt +getPmax(): Pnt +getPmax_droite(): Pnt +getPmin(): Pnt +getPmin_droite(): Pnt +setCentre(Pnt): void +setPmax(Pnt): void +setPmaxDroite(Pnt): void +setPmin(Pnt): void +setPminDroite(Pnt): void

Figure 2:Classe Cellule.➤ **Classe Droit :**

Droit
-liste_point_intersection: List_Points -p1: Point -p2: Point
+calculer_determinat(double, double, double, double, double, double, double, double, &double, &double, &double, &double) +Droit(Point, Point) +getListe_point_intesection(): List_Points +getP1(): Point +getp2(): point +interseccion_doite_plan(Trg*): void +intersection_droite_segment(Droit): void +intersection_droite_trianle_parallele(Trg*): void +setP1(): void +setP2(): void

Figure 3:Classe Droit.➤ **Classe Dixel :**

Dixel
-matrice: bool -point_debut: Pnt -point_fin: Pnt -supprimer: bool
+dessinerDixel(): void +dessinerVolumeDixel(): void +Dixel(Pnt, Pnt) +getMatrice(): bool +getPointDebut(): Pnt +getPointFin(): Pnt +getSupprimer(): bool +setMatrice(): void +setPointDebut(Pnt): void +setPointFin(Pnt): void +setSupprimer(bool): void

Figure 4:Classe Dixel.

2.2. Paquetage Trajectoire d'outil :

Spécifier pour la génération de la trajectoire d'outil.

➤ Classe Centre_Outil :

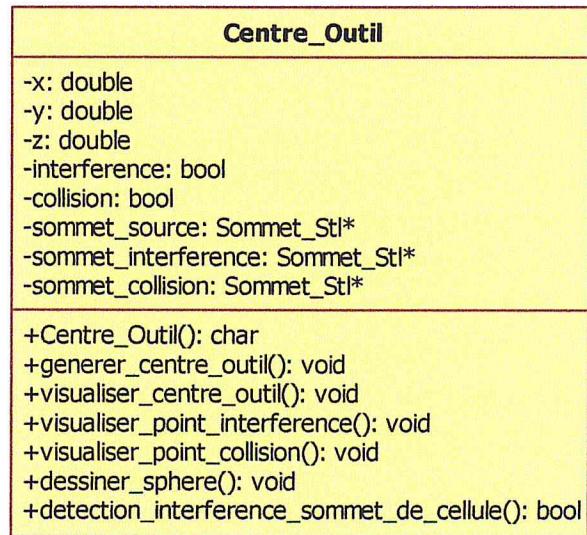


Figure 5: Classe Centre_Outil.

➤ Classe Outil_Stl :

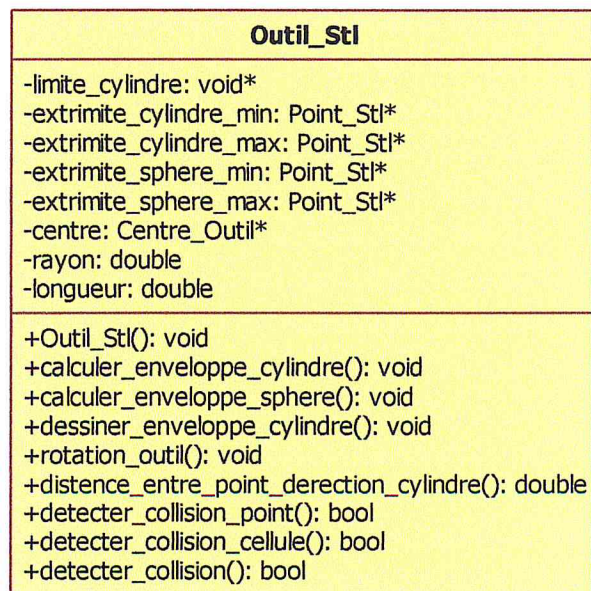


Figure 6: Classe Outil_Stl

➤ Classe Contour_Stl :

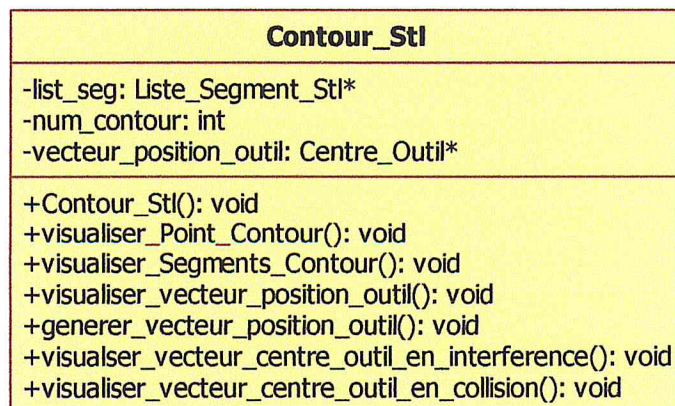
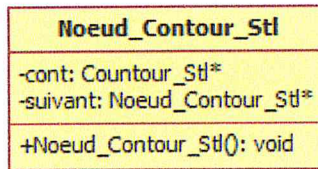
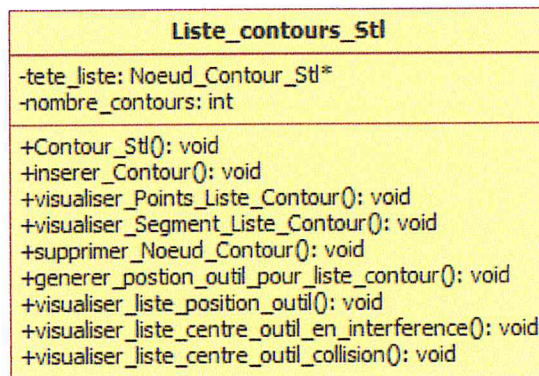
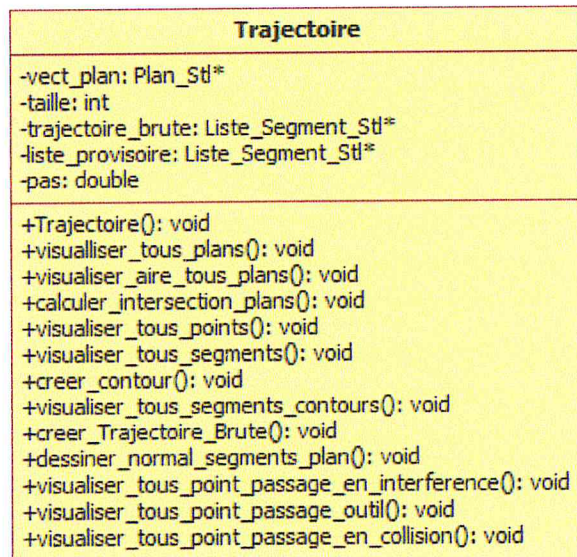


Figure 7: Classe Contour_Stl.

➤ **Classe Noeud_Contour_Stl :***Figure 8:Classe Noeud_Contour_Stl.*➤ **Classe Liste_Contours_stl :***Figure 9:Classe Liste_Contours_Stl.*➤ **Classe Trajectoire :***Figure 10:Classe Trajectoire.*

2.3. Paquetage simulation d'enlèvement de matière :

Les classes du paquetage la simulation de l'enlèvement de matière des pièces représentées par des Triples-Dexel suivant une trajectoire d'outil sont les suivants :

2.3.1. Diagramme de classes :

Le diagramme de classe est une représentation statique des éléments qui composent un système et de leurs relations, permet de spécifier QUI intervient à l'intérieur du système.

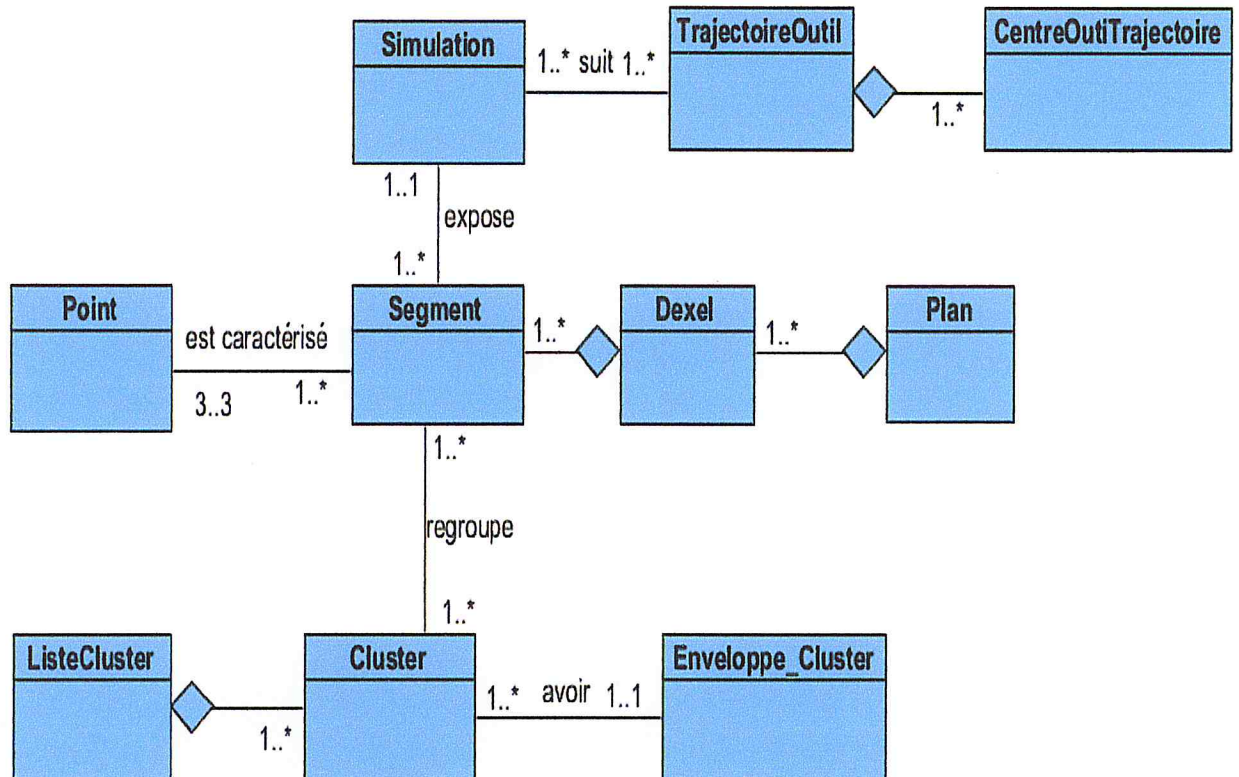


Figure 11 : Diagramme de classes.

➤ **Classe point** : représente les coordonnées d'un point.

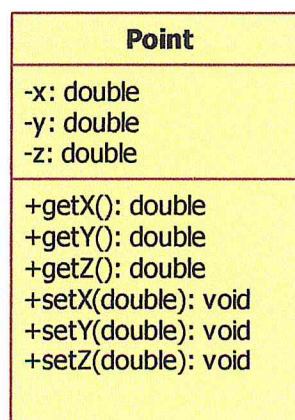


Figure 12: Classe Point.

➤ **Classe segment :**

Caractérise le segment, ses attributs et fonctions sont donnés par la figure 12.

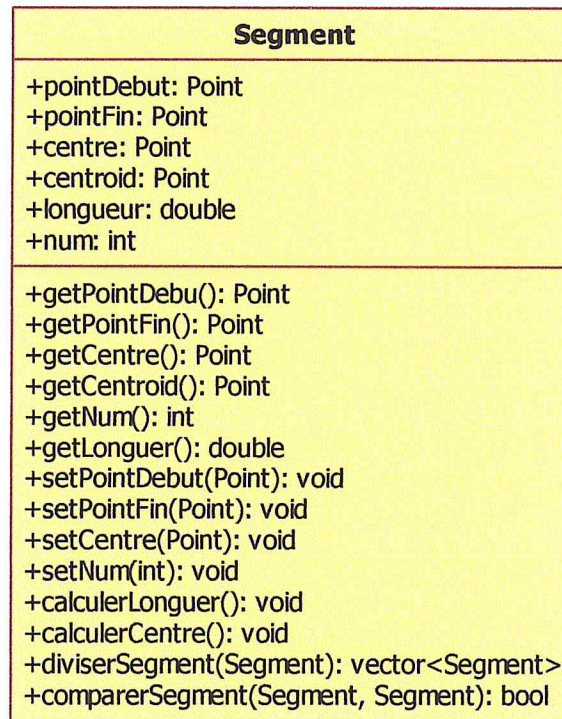


Figure 13:Classe Segment.

➤ **Classe TDexel :**

Définie les dexels, leur attributs et fonction représenter par la figure 13.

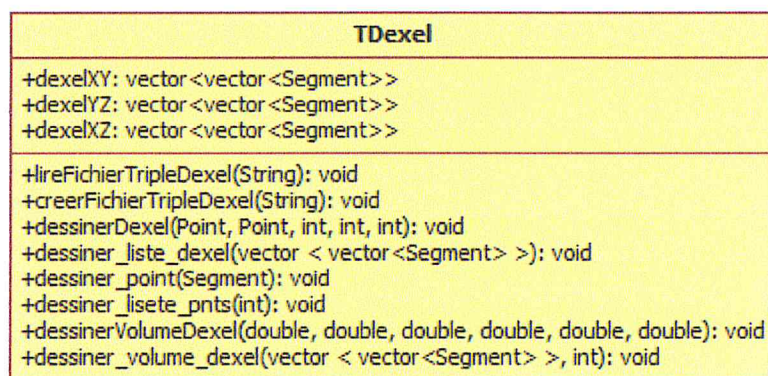
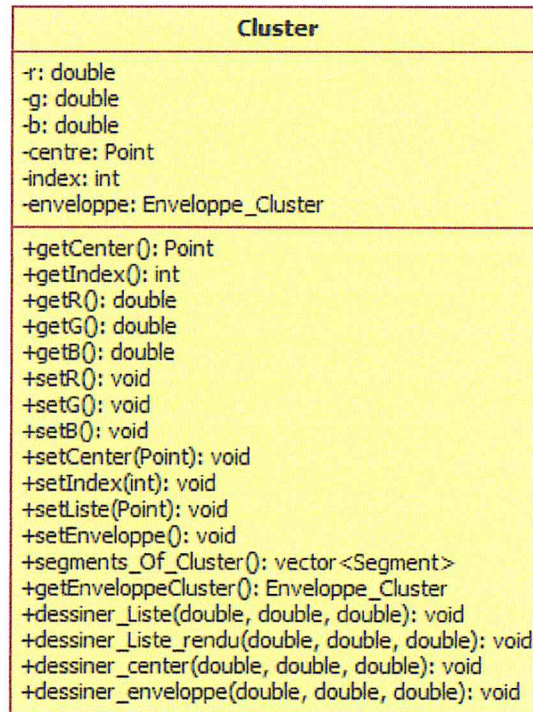
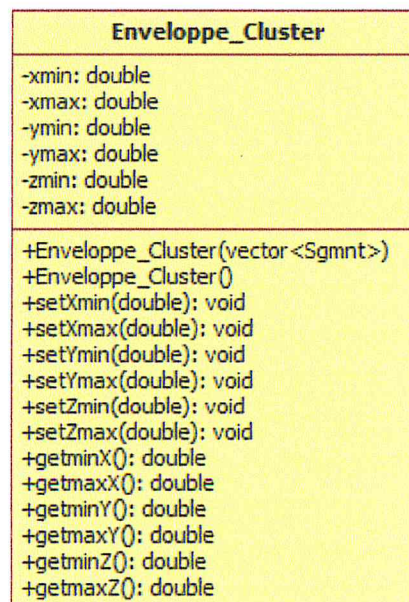


Figure 14:Classe TDexel.

➤ **Classe Cluster :***Figure 15:Classe Cluster*➤ **Liste_Cluster :***Figure 16:classe Liste_Cluster.*➤ **Enveloppe_Cluster :***Figure 17:Classe Enveloppe_Cluster.*

➤ **Classe TrajectoireOutil :**

TrajectoireOutil
-nbre_total_centre: int
+vecteurTrajectoire(): vector <vector <PntCentreOutil> >
+getNbre_total_centre(): int
+setNbre_total_centre(): void
+creerFichierCE(): void
+lireFichierCE(): void
+dessinerOutil(): void
+trajet_outil(): void
+visualiserOutil(): void
+trajet_outil_rapide(): void
+visualiser_outil_rapide(): void

Figure 18:Classe TrajectoireOutil.

➤ **Classe PointCentreOutil :**

PointCentreOutil
-x: double
-y: double
-z: double
-theta: double
-phi: double
-rayon: double
-longueur: double
+getX(): double
+getY(): double
+getZ(): double
+getTheta(): double
+getPhi(): double
+getRayon(): double
+getLongueur(): double
+setX(double): void
+setY(double): void
+setZ(double): void
+setTheta(double): void
+setPhi(double): void
+setLongueur(double): void

Figure 19:Classe PointcentreOutil.

➤ **Classe Simulation :**

Définit les attributs et fonctions implémentés dans la simulation d'enlèvement de matière (figure 19).

Simulation
-tolerance: double +liste_segments_bien_usinee: vector<Segment> +liste0: vector<Sgmt> +liste1: vector<Sgmt> +liste11: vector<Sgmt>
+inetrsection_sphere_cluster(PointCentreOutil, Cluster): bool +enveloppes_chevauchants(PointCentreOutil): vector<Cluster> +calcul_intersection_sphere_cluster(PointCentreOutil): void +cluster_point_proche_centre(vector < vector<Segment> >, Point): vector<Sgmt> +proche_centre(PointCentreOutil, Point, double): bool +getTolerance(): double +setTolerance(): void +grouper_segments(): vector<Segment> +calculer_temps_execution(): void +matiere_enlevee(Segment): double +get_plan(Segment): String +segment_plus_ou_moins(Segment, double): Segment +remplir_listes(): void +appartient_liste0(Segment): void +remplir_liste_segments_bien_usinee(): void +dessiner_matiere(): void +dessiner_volume_matiere(): void +dessiner_matiere_en_plus(): void +dessiner_volume_en_plus(): void +dessiner_matiere_en_moins(): void +dessiner_volume_en_moins(): void

Figure 20:Classe Simulation.

Les fonctions :

calcul_intersection_sphère_cluster(PointCentreOutil p) dans cette fonction nous calculons l'intersection de la sphère de point centre « p » et les segments des clusters qui chevauchent avec l'enveloppe de l'outil en cette position.

Calculer_temps_execution() cette fonction à pour but de calculer le temps d'exécution de l'usinage pour comparer la durée de l'usinage en changeant les différents paramètres associés.

Matière_enlevée(Segment s) calcul le volume de la matière enlevée du segment s.

Les fonctions de dessin : Pour dessiner les segments soit en segment de Dixel (dessiner_matiere) ou en volume Dixel (dessiner_volume).

Dessiner_matiere() Dessiner les Segments de Dixel bien usinés.

Dessiner_volume() Dessiner les volumes de Dixel bien usinés.

Dessiner_matière_en_plus() Dessiner les Segments de Dixel qui ont été mal usinés en enlevant plus de matière.

Dessiner_volume_en_plus() Dessiner les volumes de Dixel qui ont été mal usinés en enlevant plus de matière.

Dessiner_matière_en_moins() Dessiner les Segments de Dixel qui ont été mal usinés en enlevant moins de matière.

Dessiner_volume_en_moins() Dessiner les volumes de Dixel qui ont été mal usinés en enlevant moins de matière.

3. Présentation de l'application :

Dans cette phase nous démontrons les différentes tâches implémentées dans le cadre de notre projet.

3.1. Fenêtre principale :

La fenêtre principale est composée de cinq onglets (parties).

- La 1ere partie : « Triple-Dixels », permet la manipulation des fichiers Triple-Dixels.
- La 2eme partie : « Surépaisseur », pour ajouter une surépaisseur aux Triple-Dixels.
- La 3eme partie : « Optimisation », permet de classier les segments en appliquant l'algorithme de k-means.
- La 4eme partie : « Simulation », permet de voir la simulation de l'enlèvement de matière en usinage 5axes.
- La 5eme partie : « Résultats et statistiques », pour montrer les résultats de la simulation.

Les figures qui suivent, montrent les différentes fonctionnalités de ces onglets, l'utilisateur doit taper sur les boutons indiqués, pour exécuter la procédure écrite en dehors de la fenêtre.

3.2. Onglet « Pièce » :

L'onglet « pièce » (figure 21) permet de faire les tâches suivantes :

- Ouvrir l'application Triple Dixel pour l'exécuter et récupérer les données.
- Créer fichier Triple-Dixels, avec l'extension « .TD ».
- Lire fichier Triple-Dixel.

- Visualiser la pièce en Dixel.
- Visualiser les points début et points fin des segments de la pièce.
- Visualiser le volume des Dixel de la pièce.

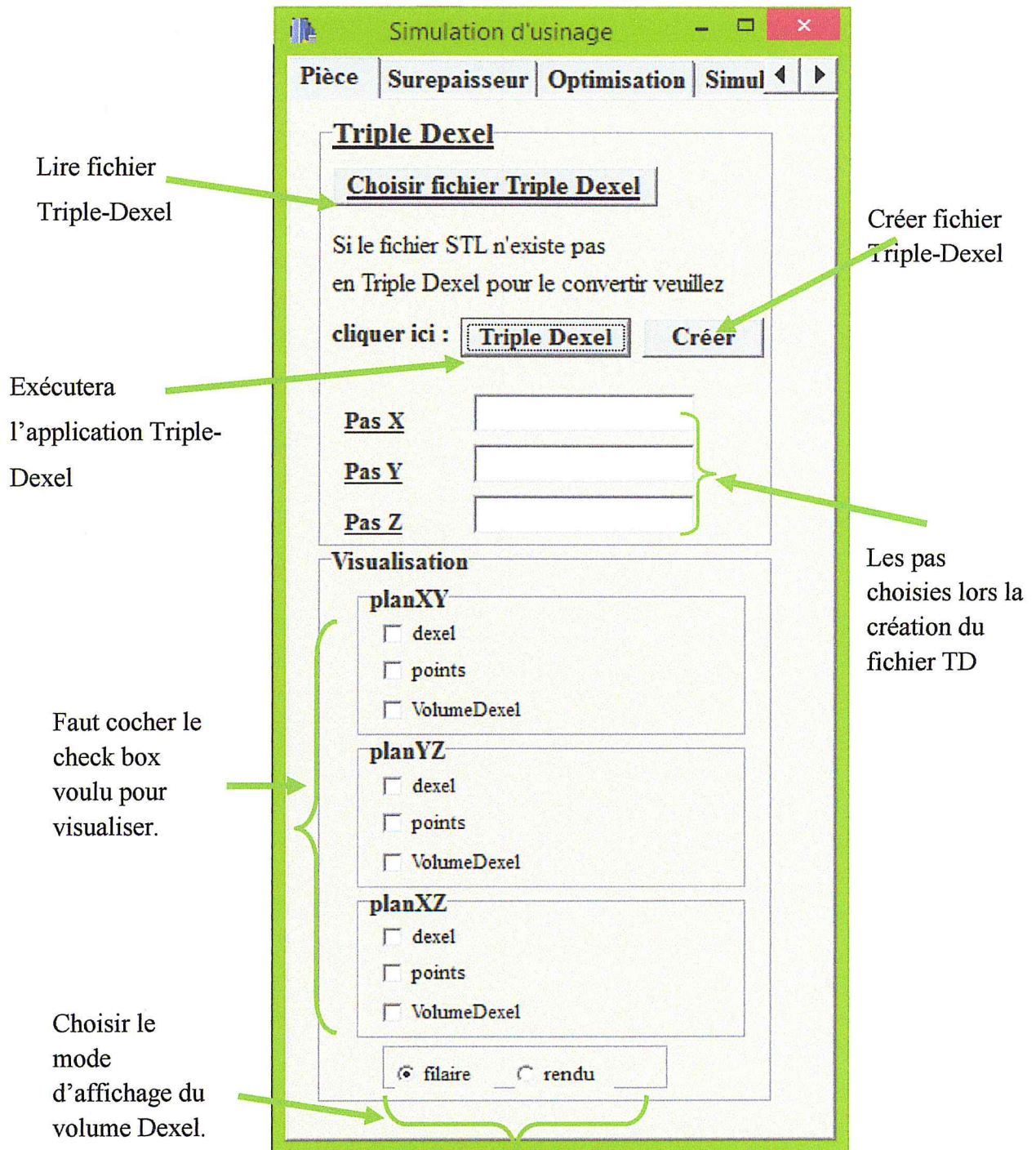


Figure 21: Fonctionnalités de l'onglet pièce.

3.3. Onglet Surepasseur :

Après introduire la surepasseur (figure 22) nous pouvons visualiser:

- La surepasseur de la pièce en Dixel (figure 23.a).
- La surepasseur de la pièce en représentation volumique des Dixel (figure 23.b).
- La pièce à usiner en Dixel (figure 24.a).
- Visualiser la pièce à usiner en représentation volumique des Dixel (figure 24.b).

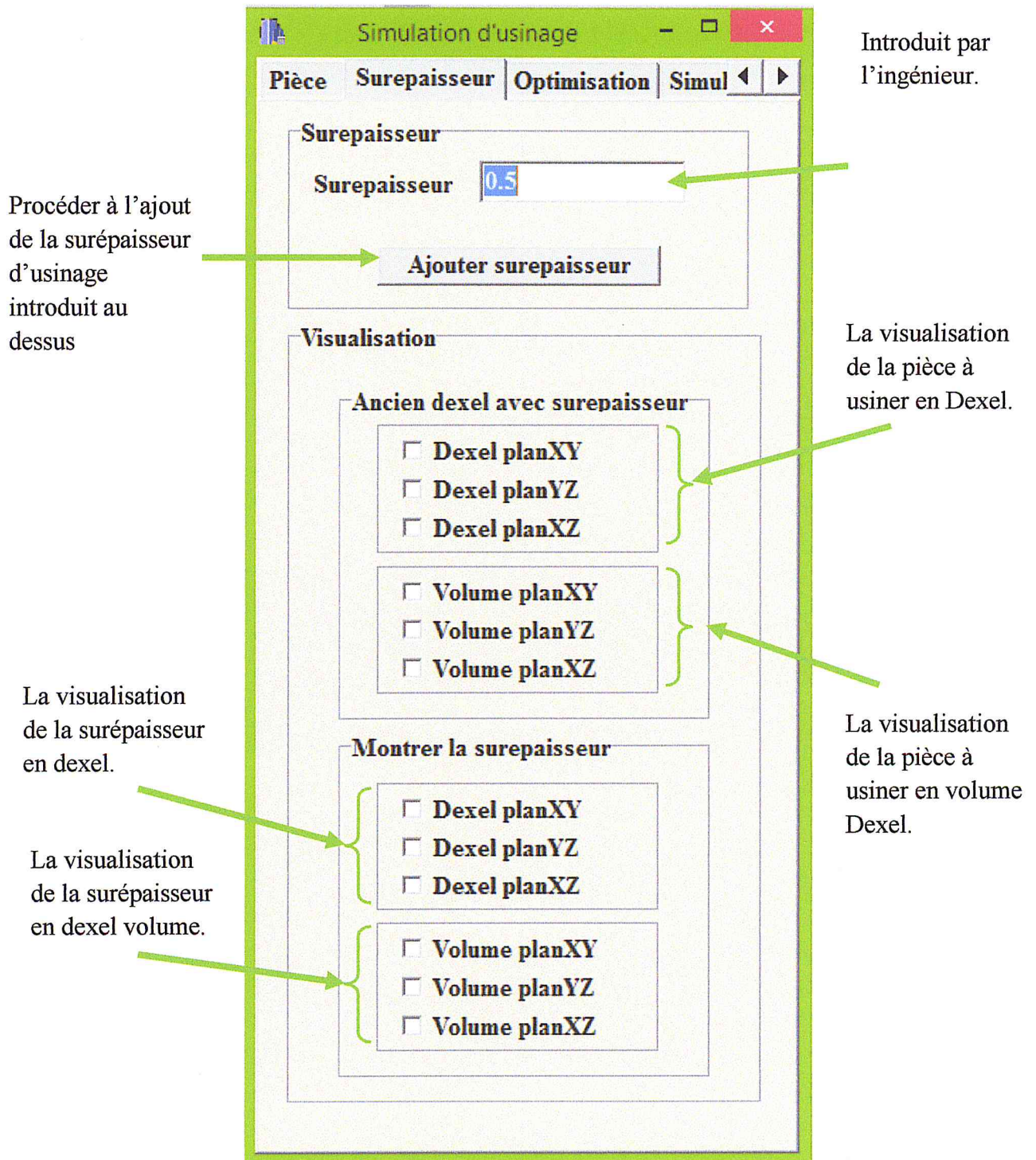
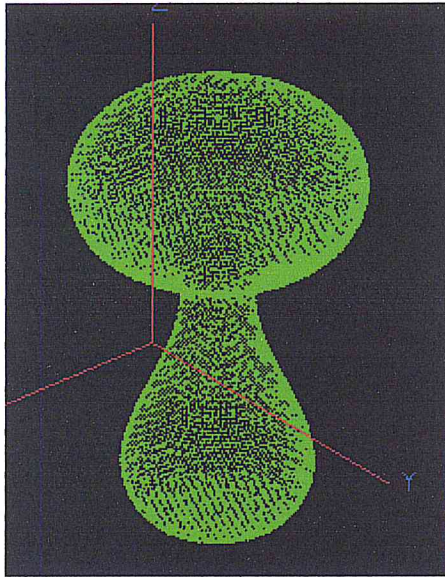
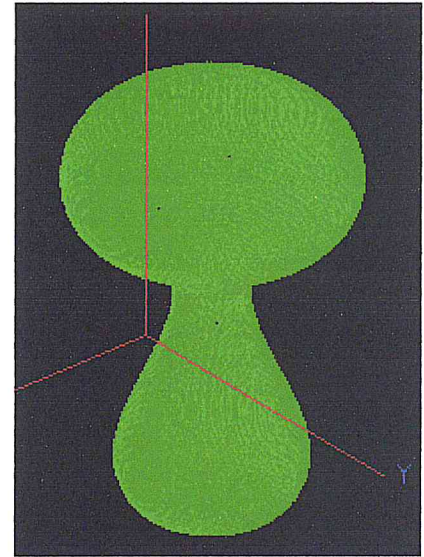


Figure 22:onglet surepasseur.



a- Visualisation des Dexels de la surépaisseur d'usinage



b- Visualisation volumique des Dexels de la surépaisseur d'usinage

Figure 23: Visualisation de la surépaisseur d'usinage.

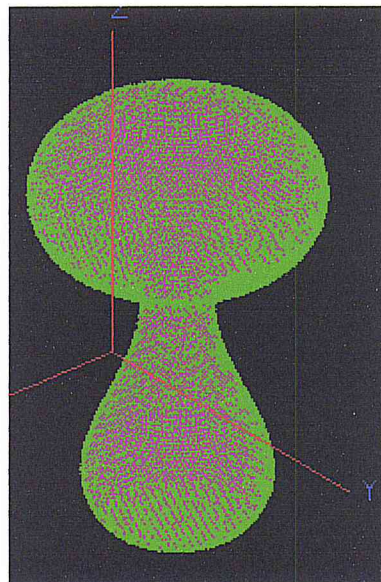


Figure 24: Visualisation de la pièce avec surépaisseur.

3.4. Onglet Optimisation :

L'onglet « optimisation » (figure 25) permet de :

- De classifier les données (segments) dans des clusters, le nombre de ces clusters doit être introduit par l'ingénieur.
- Visualiser la totalité des clusters en filaire (figure 26.a), ou rendu (figure 26.b).
- Visualiser la totalité des enveloppes des clusters (figure 27).
- Visualisation par cluster (figure 28).
- Afficher les limites et les centroides.

Simulation d'usage

Pièce | Surepasseur | **Optimisation** | Simul

KMeans

Creation des clusters

Nombre de Clusters: 10

Nombre d'itérations: [] **Créer**

Visualisation Globale

Centroids: 2

Clusters filaire: 2

Clusters rendus Enveloppes: 2

Visualisation par cluster

numéro cluster: []

nombre de segments: []

Limites

Xmin: [] Xmax: []

Ymin: [] Ymax: []

Zmin: [] Zmax: []

Centroid: 2

coordonnées

x: [] Y: [] Z: []

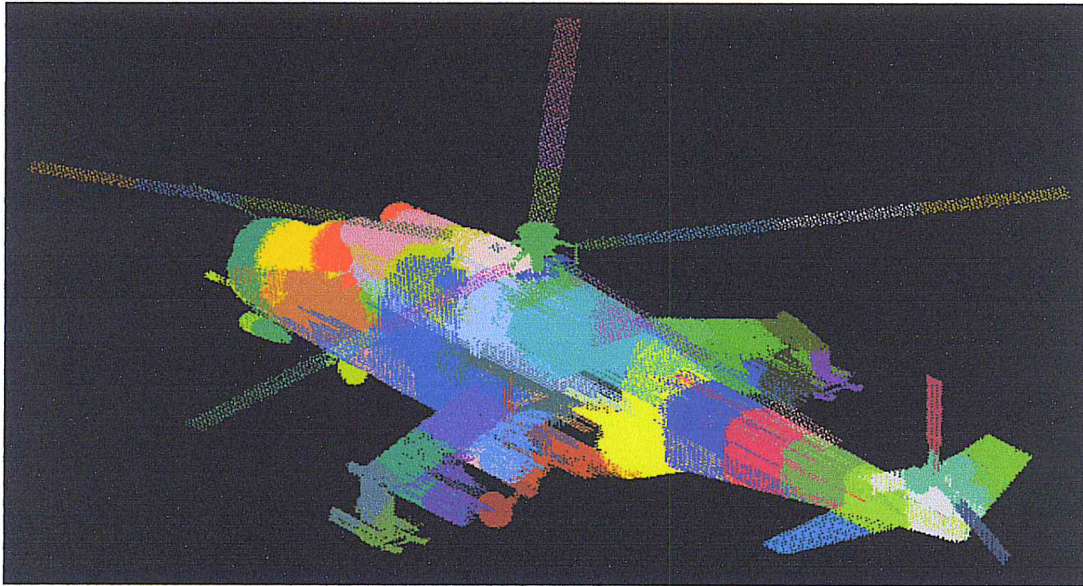
Cluster filaire: 2

Cluster rendu Enveloppe: 2

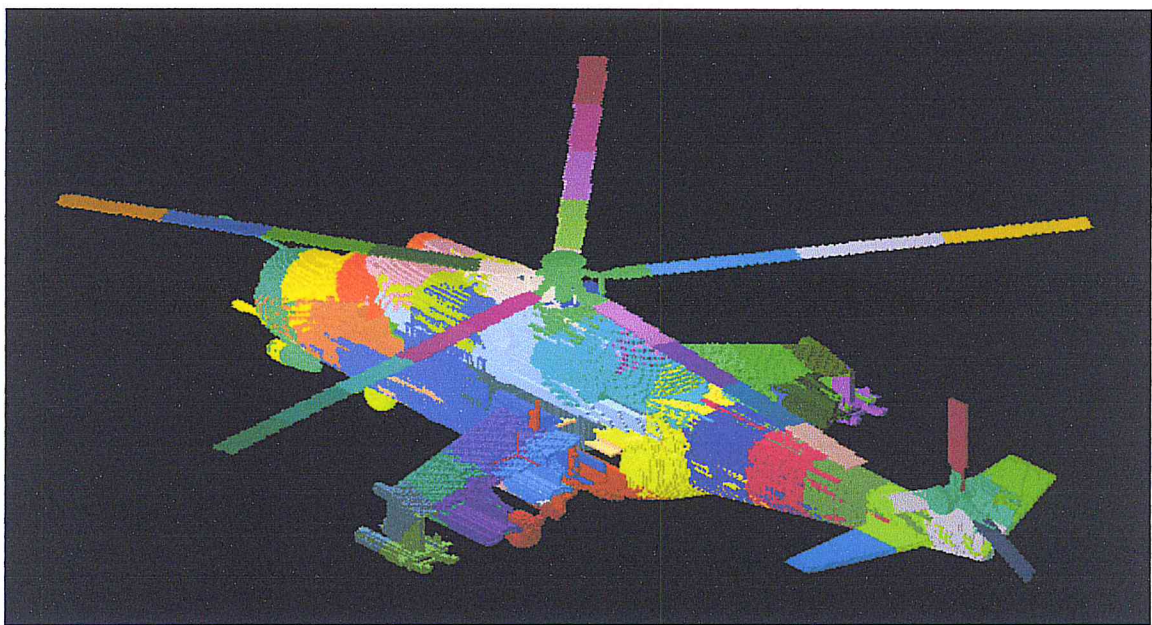
Annotations:

- Introduire le nombre de clusters. (pointe à 'Nombre de Clusters')
- Nombre d'itérations. (S'affiche après le calcul des clusters). (pointe à 'Nombre d'itérations')
- Sélectionner le numéro de cluster désiré pour le visualiser (pointe à 'numéro cluster')
- Les limites de l'enveloppe de cluster sélectionné au-dessus (pointe à 'Limites')
- Les coordonnées du centre de cluster sélectionné au-dessus. (pointe à 'coordonnées')
- Créer les clusters, et leurs affecter les segments (pointe à 'Créer')
- Déterminer la largeur des objets dans l'affichage. (pointe à 'Centroids', 'Clusters filaire', 'Enveloppes')
- Nombre de segments dans le cluster sélectionné (pointe à 'nombre de segments')

Figure 25:onglet Optimisation.



*a- Visualisation globale des clusters
en filaire (80 clusters)*



*b- Visualisation globale des clusters
en rendu. (80 clusters)*

Figure 26 : Visualisation globale des clusters.

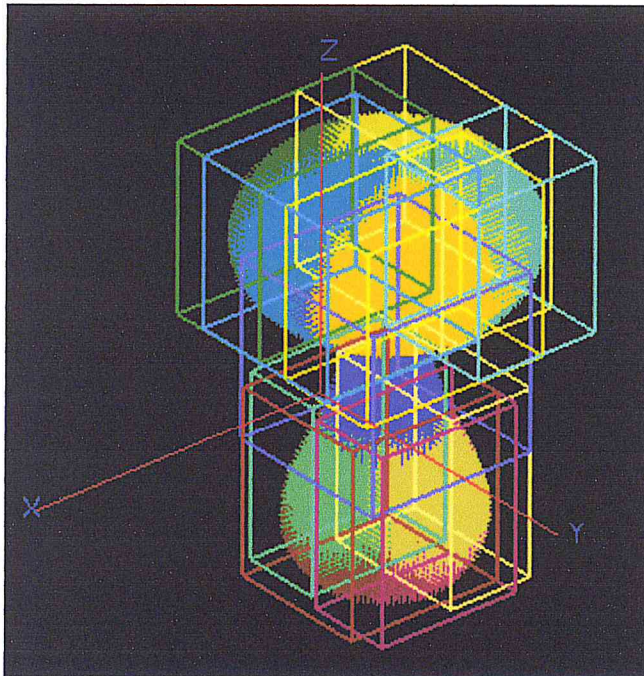


Figure 28: visualisation globale des enveloppes des clusters.

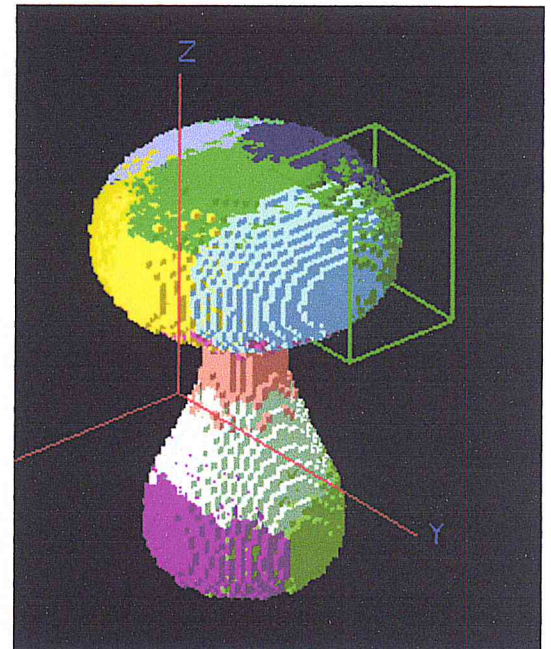


Figure 27: Exemple : visualisation locale d'une enveloppe d'un cluster.

3.5. Onglet Simulation :

L'onglet « simulation » (figure 29) permet de :

- Ouvrir l'application Trajectoire d'outil pour l'exécuter et introduire les paramètres d'outil désiré ensuite récupérer le trajet.
- Créer les fichiers de trajectoire par les points CE (centre d'outil).
- Lire les fichiers CE.
- Visualiser la simulation d'enlèvement de matière (figure 30), et visualisation après la simulation (figures 31, 32, 33).
- Afficher les paramètres de l'outil à chaque position au cours de la simulation d'usinage.
- Le Check Box « Visualiser outil » permet de visualiser l'outil au cours de la simulation rapide de l'usinage, et les Check Box « Outil/ Enveloppe » permettent de visualiser l'outil et l'enveloppe lors la simulation d'usinage par point.

La simulation d'usinage aide à quantifier les erreurs, et bien choisir ses paramètres, tel

qu'en finition le paramètre de distance entre les plans (le z-constant) de l'outil doit être petit pour avoir une surface plus lisse à la fin. Les figures 32 et 33 montrent la différence entre la simulation avec une distance = 1 et une autre = 4.

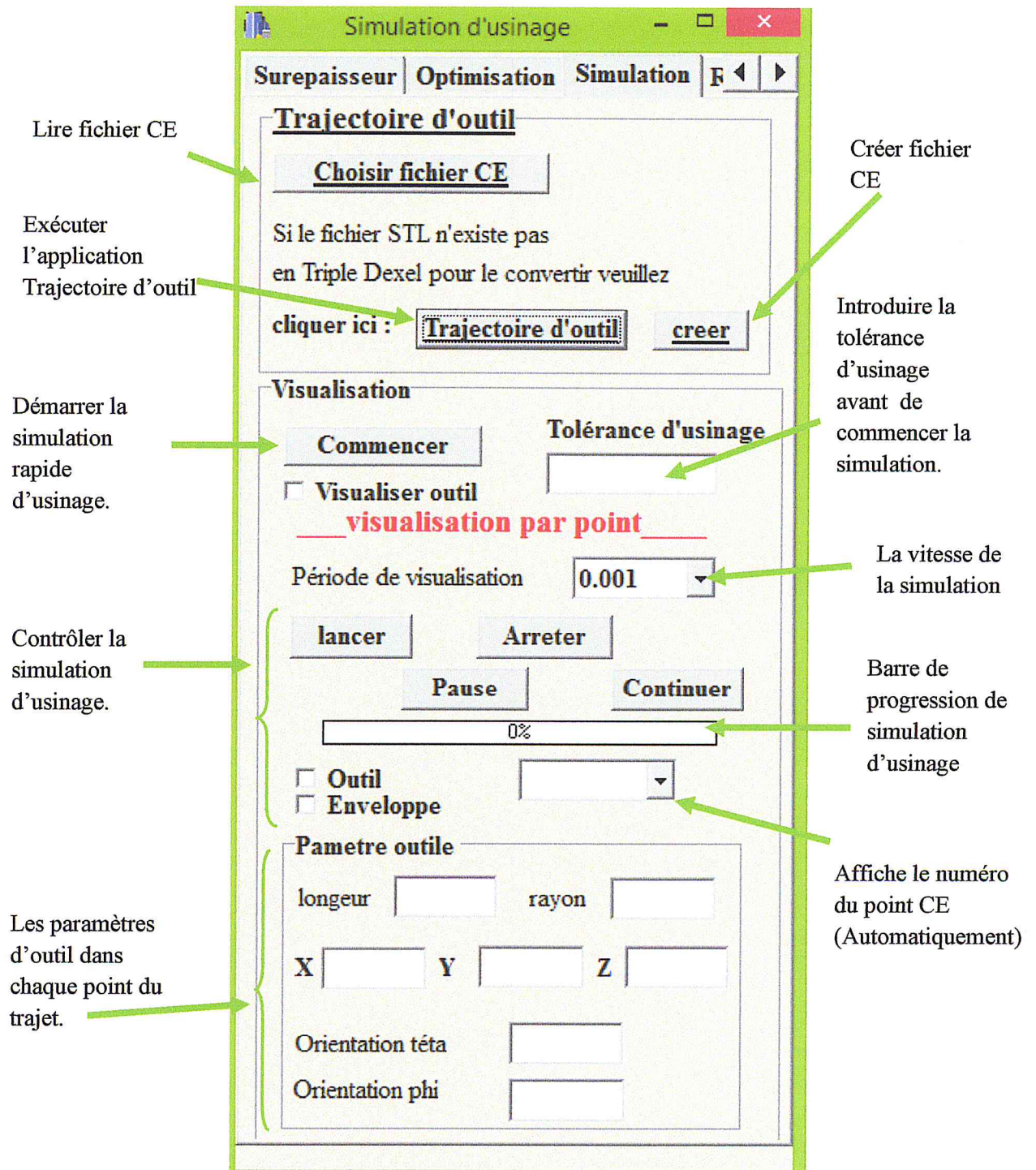
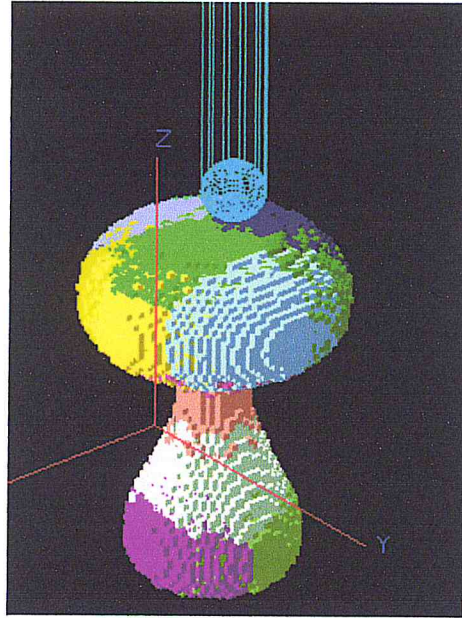
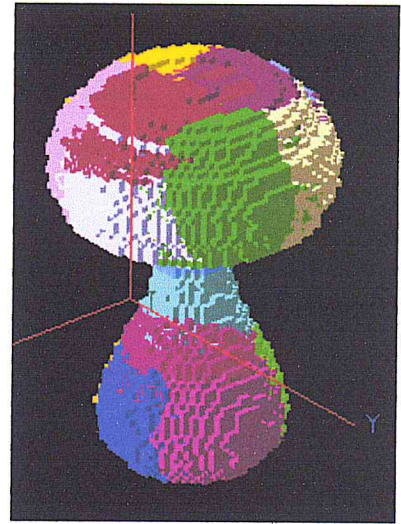


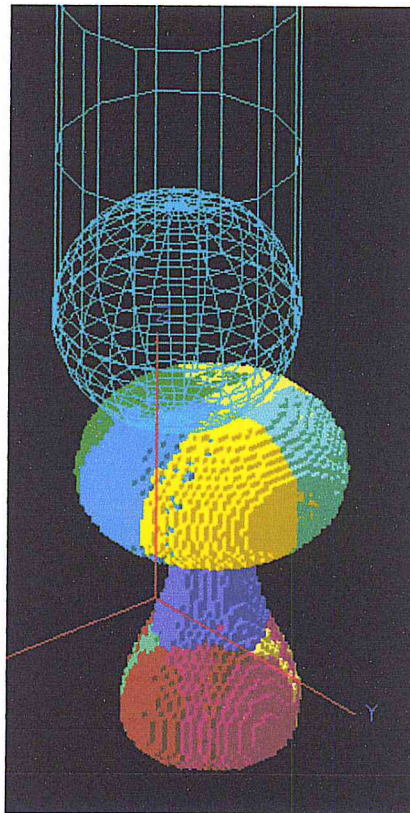
Figure 29:onglet simulation.



Visualisation avec l'outil



Visualisation sans l'outil



Simulation d'usinage où le rayon de l'outil change.

Figure 30: Simulation d'enlèvement de matière.

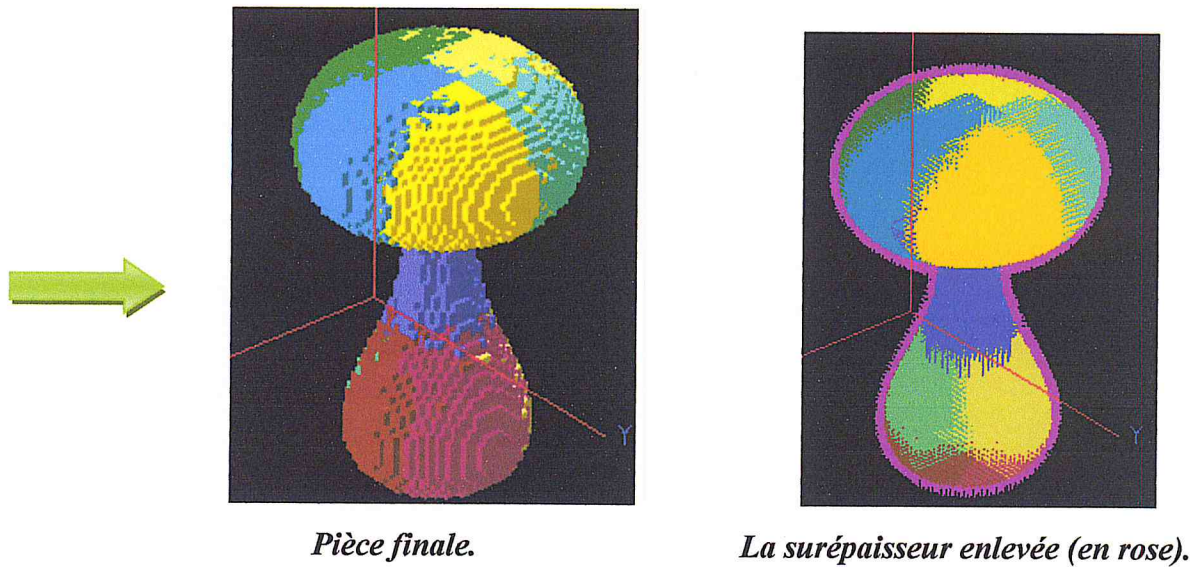


Figure 31 : Après la simulation où l'outil change

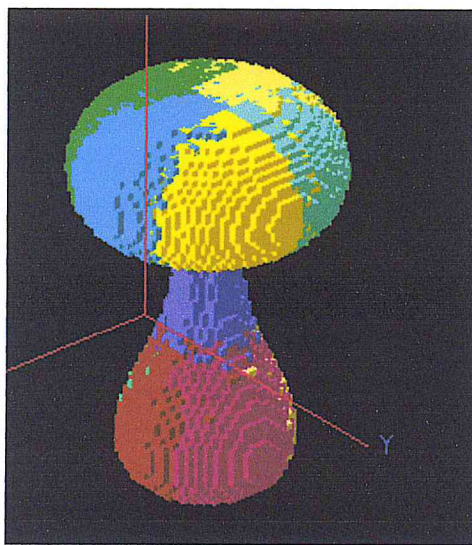


Figure 33 : Après la simulation d'usinage d'une trajectoire avec une distance entre les plans $z_constant = 1$.

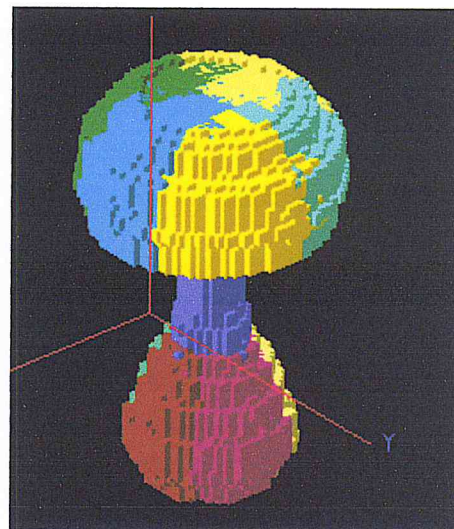


Figure 32 : Après la simulation d'usinage d'une trajectoire avec une distance entre les plans $z_constant = 4$.

3.6. Onglet résultats et statistique :

Cet onglet (figure 34) est fait pour calculer le temps de l'exécution, en cliquant sur le bouton afficher dans la partie « Statistiques » et afficher (partie résultats) et quantifier (partie statistiques) les erreurs de l'usinage de la pièce, montre l'enlèvement de la matière :

- Bien usinée, c'est les segments où la matière enlevée de ces derniers respecte la tolérance déterminée par l'ingénieur.
- En plus, $||\text{segment}|| < ||\text{ancien segment}|| - \text{tolérance}$.
- En moins, $||\text{segment}|| > ||\text{ancien segment}|| + \text{tolérance}$.

La génération des listes de segments mal usinés.

La génération de la liste des segments bien usinés.

Cocher le check Box pour afficher soit en filaire (Dexel) ou en rendu (volume Dexel)

Le temps d'exécution de la simulation d'usinage.

L'affichage du volume des segments mal usinés.

Figure 34 : Onglet résultats et statistiques.

La pièce ci-dessous montre les résultats après la simulation de la pièce « poigné » en choisissant les paramètres :

- Pas de discrétisation de Triple-Dexel = 0.7 mm
- Distance entre les plans (z-constant) = 1 mm
- Profondeur de passe (surépaisseur) = 1mm

Sachant que : les segments bien usinés sont en vert, les segments mal usinés en enlevant moins de matière : en jaune, les segments mal usinés en enlevant plus de matière : en rouge. En montrant les segments jaune et vert dans la figure36, et rouge et vert dans la figure35, et tous les segments dans la figure37, nous remarquons que la pièce ne respecte pas vraiment la tolérance introduit

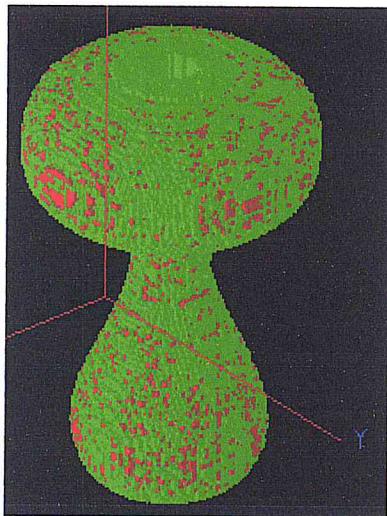


Figure 35 : Visualisation des segments bien usinés et les segments mal usinés en enlevant plus de matière.

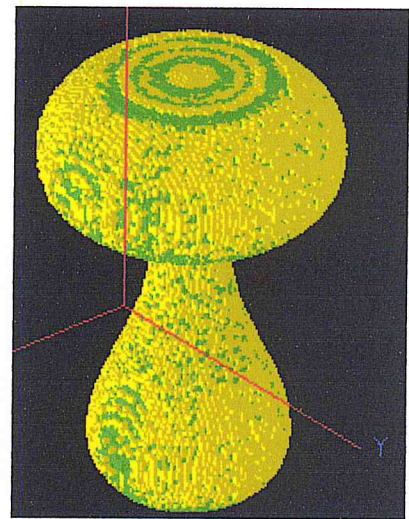


Figure 36 : Visualisation des segments bien usinés et les segments mal usinés en enlevant moins de matière.

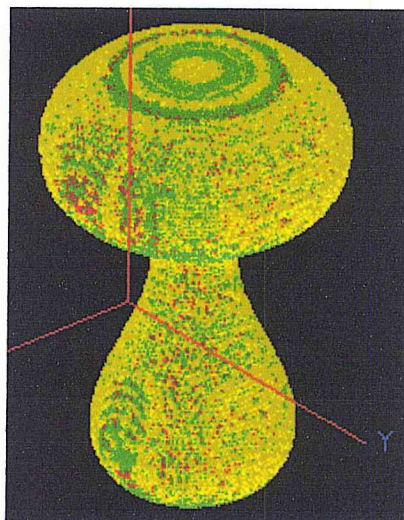


Figure 37 : Visualisation de tous les segments après le calcul des erreurs.

3.7. Test paramétrique :

Nous avons introduit 3 paramètres :

- La surépaisseur.
- Le pas de discrétisation des Triple-Dexels: influence sur le temps.
- La tolérance d'usinage : influence sur la perfection d'usinage.

Le tableau1 montre le changement du temps d'exécution de la pièce « poignée » en variant le paramètre de pas de discrétisation du Triple-Dexel et en fixant les autres paramètres, tel que : la tolérance d'usinage = 0.5, la profondeur de passe = 0.5 dans les deux pièces.

Pièce	Pas	Temps d'exécution	Nombre de segments bien usinés	Nombre de segments mal usinés
Poigné	1	5 minutes	21383	54166
	2	2 minutes	11127	23448
	3	18 secondes	2574	5293
	4	10 secondes	2616	3243

Tableau 1 : Test sur le paramètre de pas de discrétisation.

Le tableau 2 montre le changement des segments bien usinés et mal usinés de la même pièce, en variant le paramètre de tolérance d'usinage et en fixant les autres paramètres, tel que le pas de discrétisation = 2, la profondeur de passe = 2 dans les deux pièces.

Surépaisseur	Tolérance	Temps d'exécution	Nombre de segments bien usinés	Nombre de segments mal usinés	% des segments bien usinés dans la pièce
0.5	0.25	36 s	8968	13848	37,94 %
	0.5	38 s	11362	12824	46,97 %
	1	40 s	20579	3487	85,51 %
0.25	0.25	35 s	8471	12751	39,91 %
	0.5	39 s	20979	3379	86,12 %
	1	40 s	23787	0	100 %

1	0.25	38 s	6661	13746	32,64 %
	0.5	40 s	9072	12702	41,66 %
	1	38 s	11599	12014	49,12 %

Tableau 2 : Test paramètre de tolérance d'usinage.

4. Conclusion :

Ce dernier chapitre était réservé aux tests de notre application en mettant en évidence le déroulement des différentes étapes qui mènent à la simulation d'enlèvement de matière en mode 5axes. Pour la détermination des erreurs d'usinage nous avons testé notre application sur des pièces mécaniques de formes gauches usinés sur des fraiseuses 5-axes (hélicoptère et poignée). Pour différents paramètres d'usinage liés à la trajectoire d'outil (distance entre plans, dimensions d'outil), au modèle volumique de la pièce (pas de discrétisation en Triple-Dxel) et à la profondeur de passe (surépaisseur), selon les résultats présentés nous remarquons l'importante influence de ces paramètres sur la qualité de la pièce usinée, l'intérêt de cette simulation est d'aider l'opérateur à sélectionner les paramètres adéquats.

CONCLUSION

GENERALE

Conclusion Générale :

L'objectif de notre sujet d'étude est de réaliser une application logicielle ayant pour but la simulation d'enlèvement de matière au stade de finition en fraisage 5axes conformément aux normes de tolérances requises et exigées par l'ingénieur ou l'utilisateur.

La finalité de notre travail connue, ainsi, nous avons présenté les démarches tout au long de notre thème pour réaliser une application qui prend en charge les tâches suivantes :

- ✓ L'Optimisation du temps de calcul avec l'utilisation de k-means ;
- ✓ La détermination des intersections entre l'outil hémisphérique et les triple dexels ;
- ✓ La simulation virtuelle de l'enlèvement de matière en finition ;
- ✓ Et enfin, la vérification de la tolérance exigée par l'ingénieur.

De part ce qui précède, nous pouvons en conclure que la simulation virtuelle de l'enlèvement de matière est un outil de vérification nécessaire avant tout passage à l'opération d'usinage réel. La sélection des paramètres de coupe appropriés conduisent à l'obtention d'une meilleure qualité de surface pour éviter l'opération de super finition afin de réduire les couts de fabrication.

En perspective de notre travail, nous recommandons de traiter les problèmes suivants :

- Localiser les points de contact outils-pièce et le calcul de l'effort de coupe afin d'optimiser les conditions de coupe qui influent directement sur la qualité de surface.
- Faire introduire d'autres paramètres tels que la vitesse d'avance et la vitesse de coupe
- Utiliser d'autre stratégie d'usinage.

REFERENCES

BIBLIOGRAPHIQUE

BIBLIOGRAPHIE :

[ARF, 11] E. Aras, H.Y. Feng, Vector model-based workpiece update in multi-axis milling by moving surface of revolution, *Manuf Technol*, 52(2011), p.913–927.

[BEA ,13] J. Beaudic. <http://www.imetaux.net/cfao/comprendre/definition.php>(2013).

[BOU, 13] K. Bouhadja , Fabrication des surfaces de forme gauche.Rapport de recherche (2013) .

[BOB, 16] K. Bouhadja, M.Bey, Modélisation Volumique des Pièces de Formes Complexes par Triple-Dexels, 10^{èmes} Journées de Mécanique de l'EMP (JM'10–EMP) (2016).

[BCK, 11] M. Bey, Z. Chantchane, L. Kheidri and N. Benhenda ,Automatisation de l'Opération de Tréflage.Rapport de recherche (2011).

[BET, 12] F. Bettine : Etude du processus d'usinage des pièces mécanique sur les machines CNC à 5 axes.In : <https://library.csc.dz/abstract.php?docId=320&docType=t> (2012) .

[CHY, 04]C.M. Chuang, H.T. Yuma, new approach to z-level contour machining of triangulated surface models using fillet endmills (2004).

[GAR, 14] S. Gara, productique - Procédés d'usinage - Tournage, fraisage, perçage, rectification (niveau A), ellipses, (2014).

[GER, 13] P. Gérard, <http://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours05.html>- (2013).

[HSY, 93] P.L. Hsu and W.T.Yang, Real-time 3D simulation of 3-axis milling using isometric projection, *Computer-Aided Design*, 25 (4) (1993), p.215–224.

[JER, 89] R.B. Jerard and al, Methods for detecting errors in numerically controlled machining of sculptured surfaces, *IEEE Computer Graphics and Application* (1989), p.26-39.

[JOS, 16] J. Joseph, How to Detect Outliers Using Parametric and Non- Parametric Methods : Part II. In : <https://blog.clevertap.com/how-to-detect-outliers-using-parametric-and-non-parametric-methods-part-ii/>(2016).

[KAS, 08] K.P Karunakaran and R. Shringi, A solid modelbased off-line adaptive controller for feed rate scheduling for milling process, *Journal of Materials Processing Technology*, 204 (1–3) (2008), p.384–396.

[KAW, 91] Y. Kawashima, and al. , A flexible quantitative method for NC machining verification using a space-division based solid model, *The Visual Computer*, 7 (2–3) (1991), p.149–157.

[KIK, 05] Y.H. Kim, and S.L. Ko, Development of a machining simulation system using the Octree algorithm, *Lecture Notes in Computer Science*, 3482 (III) (2005), p.1089–1098,

[KIK, 08] Y.H. Kim and S.L. Ko, Improvement of cutting simulation using the octree method, *International Journal of Advanced Manufacturing Technology*, 28 (11–12) (2008), p.1152–1160.

[KAL, 10] F.Kalay, Simulation numérique de l'usinage-Application à l'aluminium AU4G (A2024-T351) ,*Technique De L'ingénieur, l'expertise technique de référence*, (2010).

[LEN, 11] S.W.Lee, A.Nestler, Complete swept volume generation — Part II: NC simulation of self-penetration via comprehensive analysis of envelope profiles, *Computer-Aided Design*, 43(2011), p.442–456.

[MBZ, 10] S.Mann, S.Bedi, G.Israeli, X.Zhou, Machine models and tool motions for simulating five-axis machining, *Computer-Aided Design*, 42(2010), p.231-237.

[PEC, 06] P.Y. PECHARD, Usinage de pièces de formes complexes: méthode de choix de stratégie d'usinage U.G.V., (2006).

[PIL, 13] J.F. Pillou, [http://www.commentcamarche.net/contents/1142-modelisation-avec-uml\(2\)](http://www.commentcamarche.net/contents/1142-modelisation-avec-uml(2)).

[VOH, 81] H.B. Volcker and W.A. Hunt, The role of solide modeling in machining process modeling and NC verification, *SAE Technical Paper 810195* (1981).

[YUN, 02] W.S.Yun and al. , Development of a virtual machining system, part 3: cutting process simulation in transient cuts, *International Journal of Machine Tools and Manufacture*, 42 (15) (2002), p.1617–1626.

[ZEF, 07] C.Zezhong, Q.Fu ,A Partical Approach Generating Steepest Ascent Tool-Paths for Three-Axis Finiche Milling of Compound NURBS Surfaces(2007).

[SEM, 15] K.Sebti, A.Moulay, Modélisation Volumique des Pièces de Formes Complexes par Triple Dexels à Partir de Leurs Modèles STL(2015).