



Département Génie Mécanique  
Centre de Développement des Technologies Avancées

**Mémoire de projet de fin d'études**  
pour l'obtention du diplôme d'ingénieur d'état en Mécanique

Partitionnement intelligent des pièces de forme libre.  
Application simulation de l'usinage des pièces de forme complexe.

**Younes Abdeldjalil TOUMI**

Sous la direction de :

Khadija BOUHADJA MR-B CDTA  
Djamel SAIDI MC-B ENP

Présenté et soutenu publiquement le 24/06/2023

**Composition du Jury :**

Président :	Said RECHAK	Professeur	ENP
Promoteurs :	Khadija BOUHADJA	MR-B	CDTA
	Djamel SAIDI	MC-B	ENP
Examineur :	Hamid SEDJAL	MA-A	ENP





Département Génie Mécanique  
Centre de Développement des Technologies Avancées

**Mémoire de projet de fin d'études**  
pour l'obtention du diplôme d'ingénieur d'état en Mécanique

Partitionnement intelligent des pièces de forme libre.  
Application simulation de l'usinage des pièces de forme complexe.

**Younes Abdeldjalil TOUMI**

Sous la direction de :

Khadija BOUHADJA MR-B CDTA  
Djamel SAIDI MC-B ENP

Présenté et soutenu publiquement le 24/06/2023

**Composition du Jury :**

Président :	Said RECHAK	Professeur	ENP
Promoteurs :	Khadija BOUHADJA	MR-B	CDTA
	Djamel SAIDI	MC-B	ENP
Examineur :	Hamid SEDJAL	MA-A	ENP

تهدف رسالة المذكرة هذه إلى اقتراح نهج للتصنيف يقوم بتقسيم أجزاء الشكل الحر التي تم تشكيلها على آلات التفريز ذات المحاور المتعددة بشكل ذكي. يمكن وصف هندسة هذه الأجزاء باستخدام نماذج مستمرة أو نماذج متقطعة. النماذج المتقطعة أسهل في التنفيذ من النماذج المستمرة ولكنها تتطلب تقسيمًا كثيفًا لتمثيل القطع بدقة. ومع ذلك، يؤدي هذا النهج إلى زيادة في وقت المعالجة. للتغلب على هذه الصعوبة، يتم اقتراح نهج يقوم بتحسين تجميع العناصر الهندسية إلى أجزاء باستخدام طريقة التصنيف « K-Means » المستمدة من التعلم غير المراقب للذكاء الاصطناعي. يتم حل مشكلة الوقت والجودة المرتبطة بتقسيم النماذج المتقطعة من خلال التهيئة الأولية « k-means++ » والطرز التغير « Mini Batch K-Means » يتم التحقق من صحة النتائج من خلال تطوير وحدة برمجية رسومية تفاعلية تقوم أولاً بتوليد نموذج يسمى « Triple Dixel » ، وثانياً بمحاكاة تشكيل هذا النموذج مع تقييم جودة سطحه. أظهرت المقارنة مع طريقة التصنيف التقليدية « Cell » نتائج ملحوظة، مما يبرز فعالية الدراسة في تحسين محاكاة التشكيل الهندسي المتقطع. علاوة على ذلك، يمكن توحيد الطريقة المستخدمة وتطبيقها على أي برنامج يتعامل مع نماذج هندسية متقطعة (Meshes) .

الكلمات المفتاحية: سطح حر، محاكاة القطع ، تقطيع ذو محاور متعددة، الذكاء الاصطناعي، التصنيف.

## Abstract

The purpose of this master thesis is to propose a classification approach that intelligently partitions free-form machined parts on multi-axis milling machines. The geometry of these parts can be described using continuous or discrete models. Discrete models are easier to implement than continuous models but require dense discretization to accurately represent the cutting phenomenon. However, this approach increases processing time. To overcome this difficulty, an approach is proposed that optimizes the grouping of geometric elements into partitions using the K-Means classification method derived from unsupervised AI learning. The problem of time and quality related to partitioning dense models is solved through the "k-means++" initialization and the "Mini Batch K-Means" variant. The validation of the results is carried out by developing a graphical and interactive software module that generates, firstly, a model called Triple Dixel, and secondly, simulates the machining of this model with an evaluation of the quality of its surface. The comparison with the classical "cell" classification method has yielded remarkable results, highlighting the effectiveness of the study in improving discrete geometric machining simulation. Furthermore, the method used can be standardized and applied to any software dealing with discrete geometric models (meshes).

**Keywords :** Free-form surface, Machining simulation, Multi-axis machining, artificial intelligence, Classification

## Résumé

L'objet de ce mémoire est de proposer une démarche de classification, permettant de partitionner intelligemment les pièces de formes libres usinées sur des fraiseuses multi-axes. La géométrie de ces pièces peut être décrite à l'aide de modèles continus ou discrets. Les modèles discrets sont plus faciles à implémenter que les modèles continus, mais nécessitent une discrétisation dense pour représenter fidèlement le phénomène de coupe. Cependant, cette approche entraîne une augmentation du temps de traitement. Pour surmonter cette difficulté, une approche est proposée, qui consiste à optimiser le regroupement des éléments géométriques en partitions, et ceci en utilisant la méthode de classification K-Means issue de l'apprentissage non supervisé de l'IA. Le problème lié au temps et qualité de partitionnement des modèles denses est résolu par l'initialisation "k-means++" ainsi que la variante "Mini Batch K-Means". La validation des résultats est faite par le développement d'un module logiciel graphique et interactif permettant de générer, dans un premier lieu, un modèle dit Triple Dixel, et dans un second lieu, de simuler l'usinage de ce modèle, en évaluant la qualité de sa surface. La comparaison avec la méthode classique de classification "cellule" a donné des résultats remarquables, mettant en évidence l'efficacité de l'étude réalisée pour améliorer la simulation d'usinage géométrique discrète. De plus, la méthode utilisée peut être standardisée pour être appliquée à n'importe quel logiciel traitant des modèles géométriques discrets (maillages).

**Mots-clés :** Surface libre, Simulation d'usinage, Usinage multi-axes, intelligence artificielle, Classification.

# Dédicaces

*À mes parents,  
À mon frère et ma sœur,  
À mes grand parents,*

# Remerciements

*Je tiens à exprimer ma sincère gratitude envers Mme Khadidja BOUHADJA pour avoir partagé avec moi ses connaissances, expériences, et son expertise au quotidien. Sa disponibilité, ses conseils, son temps partagé, et son aide ont grandement contribué à la réalisation de ce projet.*

*Mes remerciements vont également à M. SAIDI pour ses enseignements précieux qui m'ont permis d'acquérir les compétences et connaissances nécessaires pour mener à bien ce modeste projet de fin d'études.*

*Je souhaite exprimer ma gratitude envers l'ensemble des membres du jury : Pr. RECHAK et M. SEDJAL, d'avoir accepté d'examiner ce rapport. Leur participation et évaluation contribueront à mon développement académique et professionnel.*

*J'aimerais également remercier chaleureusement toute l'équipe du CDTA pour leur accueil chaleureux et leur esprit d'équipe. Leur collaboration et soutien ont rendu cette expérience enrichissante.*

*Enfin, je tiens à exprimer ma profonde gratitude et mes sincères remerciements à toutes les personnes qui ont contribué, ne serait-ce qu'un minimum, à la réussite de ce projet. Je n'aurai jamais assez de mots pour remercier ma chère famille et mes proches qui m'ont accompagné, encouragé et motivé tout au long de mon parcours universitaire.*

# Table des matières

Liste des figures

Liste des tableaux

Liste d'abréviations

<b>Introduction générale</b>	<b>11</b>
<b>1 Généralités et état de l'art</b>	<b>13</b>
1.1 Introduction . . . . .	14
1.2 Processus de production des pièces de forme libre . . . . .	14
1.2.1 Pièces de formes libres . . . . .	15
1.2.2 Machine de fraisage multi-axes . . . . .	16
1.2.3 Usinage 5 axes CNC . . . . .	17
1.3 Méthodes de simulation virtuelle d'usinage multiaxes . . . . .	20
1.3.1 Simulation d'usinage multiaxes sur pièces de formes libres . . . . .	20
1.3.2 Techniques d'approximation des modèles pour la simulation d'usinage . . . . .	22
1.3.3 CFAO . . . . .	25
1.3.4 Stratégies d'usinages . . . . .	27
1.4 Défauts de Fabrication . . . . .	29
1.4.1 Identification des défauts . . . . .	29
1.4.2 Analyse des défauts d'usinage . . . . .	30
1.5 Quantification des défauts de fabrication . . . . .	32
1.5.1 Prédiction de la topographie de surface . . . . .	32
1.5.2 Paramètres de l'état de surface . . . . .	32
1.6 Conclusion . . . . .	33
<b>2 Outils mathématiques</b>	<b>34</b>
2.1 Introduction . . . . .	35
2.2 Intelligence artificielle et Apprentissage automatique . . . . .	35
2.2.1 Apprentissage automatique . . . . .	36
2.3 Partitionnement de données . . . . .	37
2.3.1 Types de partitionnement . . . . .	37
2.3.2 Critère de distance . . . . .	38
2.3.3 Qualité de partitionnement . . . . .	39
2.4 Classification de données Par K-Means . . . . .	46
2.4.1 Classification par Mini Batch K-Means . . . . .	48
2.5 Problèmes liés Initialisation des centroides K-Means . . . . .	50
2.5.1 Variantes des K-Means . . . . .	50
2.5.2 Algorithme k-means++ . . . . .	51
2.6 Conclusion . . . . .	52
<b>3 Analyse et conception</b>	<b>53</b>
3.1 Introduction . . . . .	54
3.2 Solution proposée . . . . .	54
3.3 Démarche de l'application . . . . .	54
3.3.1 Acquisition du modèle triple Dixel . . . . .	56
3.3.2 Simulation d'usinage d'enlèvement de matière . . . . .	63
3.4 Conclusion . . . . .	70

<b>4</b>	<b>Implémentation et réalisation</b>	<b>71</b>
4.1	Introduction . . . . .	72
4.2	Environnement de travail . . . . .	72
4.2.1	Environnement matériel . . . . .	72
4.2.2	Environnement logiciel et langage . . . . .	72
4.3	Présentation graphique . . . . .	74
4.3.1	Application de génération d'un modèle TD . . . . .	74
4.3.2	Application de Simulation d'enlèvement de matière . . . . .	79
4.4	Scénario de fonctionnement . . . . .	84
4.4.1	Récupération du modèle STL . . . . .	84
4.4.2	Récupération du trajet d'outil .CL . . . . .	84
4.4.3	Scénario de fonctionnement du module logiciel . . . . .	85
4.5	Validation du module logiciel et résultat . . . . .	86
4.5.1	Validation de l'optimisation de génération par partitionnement . . . . .	86
4.5.2	Validation de l'optimisation de la simulation par partitionnement . . . . .	90
4.5.3	Validation de la prédiction de la topographie de surface . . . . .	91
4.5.4	Optimisation du partitionnement K-Means basique . . . . .	93
4.6	Détermination du nombre de partitions optimal . . . . .	95
4.7	Conclusion . . . . .	96
	<b>Conclusion générale</b>	<b>97</b>
	<b>Bibliographie</b>	<b>98</b>
	<b>Annexes</b>	<b>100</b>
A.	Intersection Triangle-Segment . . . . .	101
B.	Intersection Sphère-Segment . . . . .	102

# Table des figures

1.1	Types d'usinage par enlèvement de matière [1]. . . . .	14
1.2	Exemple d'usinage d'un vilebrequin. . . . .	14
1.3	Exemple de pièces de formes libre. . . . .	15
1.4	Surfaces basées sur l'utilisation des courbes. . . . .	16
1.5	Représentation d'une fraise [6]. . . . .	16
1.6	Représentation des types de processus de fraisage [7]. . . . .	17
1.7	Comparaison entre l'inclinaison de l'outil : trois axes et cinq axes [9]. . . . .	18
1.8	Différents types de machines 5 axes. . . . .	18
1.9	Types d'outils utilisés dans le fraisage 5 axes [3]. . . . .	19
1.10	Réorientation des différents types d'outils [12]. . . . .	19
1.11	Catégories de simulation [13]. . . . .	20
1.12	Catégories des échelles de simulation [13]. . . . .	21
1.13	Catégories des modèles de simulation [13]. . . . .	21
1.14	Représentation d'un triangle issu de fichier .STL. . . . .	22
1.15	Différence entre le modèle CAO et le modèle STL. . . . .	22
1.16	Comparaison entre trois modèle STL (1556, 8442 et 79600 triangles respectivement). . . . .	22
1.17	Représentation d'un fichier .STL ASCII et Binaire [16]. . . . .	23
1.18	Représentation d'un Voxel. . . . .	23
1.19	Projection des grilles sur un modèle [13]. . . . .	24
1.20	Projection des grilles sur un modèle surfacique [13]. . . . .	24
1.21	Représentation d'un modèle d'exel et triple d'exel [12]. . . . .	25
1.22	Démarche générale du processus de CFAO . . . . .	26
1.23	Trajet d'outil issu d'une stratégie isoparamétrique. . . . .	27
1.24	Trajet d'outil one-way (droite) et Zig-zag (gauche) . . . . .	27
1.25	Trajet d'outil issu d'une stratégie concentrique. . . . .	28
1.26	Trajet d'outil issu d'une stratégie par plan parallèle. . . . .	28
1.27	Trajet d'outil issu d'une stratégie z-constant. . . . .	28
1.28	Diagramme d'Ishikawa - Écart de reprise [21]. . . . .	29
1.29	Diagramme d'Ishikawa - Écart de d'usinage [21]. . . . .	29
1.30	Représentation schématique du processus d'élaboration UGV [22]. . . . .	30
1.31	Passes longitudinales et transversales. . . . .	30
1.32	Erreur de flèche [3]. . . . .	31
1.33	Erreur de crête [3]. . . . .	31
2.1	Apprentissage automatique et Intelligence artificielle. . . . .	35
2.2	Catégories de l'apprentissage automatique. . . . .	36
2.3	Problèmes de l'apprentissage supervisé. . . . .	36
2.4	Exemple de partitionnement non-supervisé. . . . .	37
2.5	Types de partitionnement non-supervisé. . . . .	37
2.6	Représentation des vecteurs $\vec{p}$ et $\vec{q}$ . . . . .	38
2.7	Qualité de partitionnement. . . . .	39
2.8	Représentation du critère du coude. . . . .	40
2.9	Représentation des coefficients $a_{(i,j)}$ et $b_{(i,j)}$ . . . . .	40
2.10	Représentation du critère de la silhouette. . . . .	41
2.11	Détermination de l'enveloppe d'objets. . . . .	42
2.12	Chevauchement entre deux enveloppes. . . . .	43
2.13	Chevauchement entre deux enveloppes pour les cas particuliers. . . . .	44
2.14	Représentation graphique du calcul de chevauchement. . . . .	45

2.15	Chevauchement d'enveloppes d'un objet .STL. . . . .	45
2.16	Représentation de différents groupes par K-Means . . . . .	46
2.17	Logigramme du partitionnement par K-Means . . . . .	47
2.18	Étapes de l'algorithme des K-Means . . . . .	47
2.19	Problème d'initialisation aléatoire. . . . .	50
3.1	Architecture d'acquisition du modèle .TD. . . . .	55
3.2	Architecture de simulation d'enlèvement de matière. . . . .	55
3.3	Fichier .TD type ASCII. . . . .	56
3.4	Fichier .STL type ASCII. . . . .	56
3.5	Processus du partitionnement par cellules d'un modèle STL [3]. . . . .	57
3.6	Processus du partitionnement par K-Means d'un modèle STL [3]. . . . .	58
3.7	Enveloppe d'un modèle STL . . . . .	59
3.8	Grille suivant les trois plans XY, YZ et XZ. . . . .	59
3.9	Intersection Segments-Triangles . . . . .	60
3.10	Translation et transformation du triangle [39] . . . . .	60
3.11	Processus de création des segments type matière. . . . .	61
3.12	Processus de création d'un Dixel volumique a partir d'un segment. . . . .	62
3.13	Ajout d'une sur-épaisseur sur un modèle .TD [3]. . . . .	63
3.14	Processus du partitionnement par cellules d'un modèle TD. . . . .	63
3.15	Processus du partitionnement par K-Means d'un modèle TD . . . . .	64
3.16	Fichier .KM type ASCII. . . . .	64
3.17	Différence entre un partitionnement généré et acquit. . . . .	65
3.18	Conversion du fichier .CL issu de <i>CAMWorks 2018</i> en .CE. . . . .	67
3.19	Représentation du point CE et CL. . . . .	67
3.20	Approximation d'un outil hémisphérique. . . . .	67
3.21	Différents cas possible d'intersection Segment-Sphère. . . . .	68
3.22	Ecart obtenu après simulation [3]. . . . .	69
3.23	Tolérance après usinage [3]. . . . .	69
4.1	Présentation de la fenêtre générale du module logiciel. . . . .	74
4.2	Présentation de la fenêtre de génération du modèle TD. . . . .	74
4.3	Représentation de l'onglet "1- Modèle STL". . . . .	75
4.4	Représentation de l'onglet "2- Classification Modèle STL". . . . .	76
4.5	Représentation de la fenêtre d'aide au choix du nombre de partitions. . . . .	77
4.6	Représentation de l'onglet "3- Création des Grilles". . . . .	77
4.7	Représentation de l'onglet "4- Génération Modèle TD". . . . .	78
4.8	Présentation de la fenêtre de simulation d'enlèvement de matière. . . . .	79
4.9	Représentation de l'onglet "1- Modèle TD". . . . .	80
4.10	Représentation de l'onglet "2- Ajout surépaisseur". . . . .	80
4.11	Représentation de l'onglet "3- Partitionnement". . . . .	81
4.12	Représentation de l'onglet "3- Partitionnement par K-Means". . . . .	81
4.13	Représentation de l'onglet "4- Trajet outil". . . . .	82
4.14	Représentation de l'onglet "5- Simulation". . . . .	82
4.15	Représentation de l'onglet "6- État de surface". . . . .	83
4.16	Processus de conception d'une surface gauche via SolidWorks. . . . .	84
4.17	Fonctionnalités permettant l'obtention du trajet .CL. . . . .	84
4.18	Trajet .CL suivant les deux courbes paramétriques. . . . .	84
4.19	Scénario de fonctionnement de l'application de génération du modèle TD. . . . .	85
4.20	Scénario de fonctionnement de l'application de simulation d'usinage. . . . .	85
4.21	Modèles utilisés pour la validation du module. . . . .	86
4.22	Comparaison entre K-Means et Cellules pour les trois modèles. . . . .	88
4.23	Variation du temps de calcul et du chevauchement pour les trois modèles. . . . .	89
4.24	Graphes pour la validation de l'optimisation de la simulation. . . . .	90
4.25	Graphes obtenus pour la validation de la prédiction de la topographie. . . . .	91
4.26	Topographie de la surface (IT = 0.3 mm). . . . .	92
4.27	Comparaison du temps de partitionnement entre l'initialisation aléatoire et 'k-means++'. . . . .	93
4.28	Comparaison du temps de partitionnement entre la méthode de K-Means et Mini Batch K-Means . . . . .	94
4.29	Détermination du nombre de partitions optimal. . . . .	95

# Liste des tableaux

4.1	Résumé des différentes informations relatives aux modèles utilisés pour la validation du module. . . . .	86
4.2	Résultats sur le partitionnement pour le modèle double vague. . . . .	87
4.3	Résultats du partitionnement pour le modèle poignée. . . . .	87
4.4	Résultats du partitionnement pour le modèle aube de turbine. . . . .	87
4.5	Résultats de partitionnement pour la simulation d'un modèle de simple complexité. . . . .	90
4.6	Résultats de partitionnement pour la simulation d'un modèle de moyenne complexité et de fortes courbures. . . . .	90
4.7	Résultats de la prédiction de la topographie de la surface. . . . .	91
4.8	Résultat de comparaison des deux initialisation K-Means. . . . .	93
4.9	Résultat de comparaison des deux algorithmes K-Means et Mini Batch K-Means. . . . .	94

# Liste d'abréviations

ASCII	American Standard Code for Information Interchange
CAO	Conception Assistée par Ordinateur
CFAO	Conception et Fabrication Assistée par Ordinateur
CNC	Computer Numerical Control
CL	Cutter Location
CE	Centre d'outil
FAO	Fabrication Assistée par Ordinateur
IA	Intelligence Artificielle
ISO	International Organization for Standardisation
KM	KMeans
STL	Stereolithography
TD	Triple Dixel
UGV	Usinage à Grande Vitesse

# Introduction générale

L'évolution technologique permanente des produits dans les divers domaines d'aéronautique, automobile et mécanique (à titre d'exemple des structures d'avions, de carrosserie d'automobile, d'aubes de turbines, etc.) a accentué la concurrence entre les concepteurs et les exigent à aboutir à des designs de plus en plus complexes avec une meilleure qualité. Ces designs sont principalement constitués de surfaces complexes, appelées aussi surfaces gauches, elles sont caractérisées par une topologie complexe composée d'un ensemble de surfaces concaves, convexes ou à double courbure. La fabrication de ces surfaces se fait généralement par le procédé d'enlèvement de matière le plus souvent par des fraiseuses multiaxes. Les opérations de fraisage liées à la production des pièces de formes gauches peuvent représenter une difficulté en raison de la considération de l'ensemble des formes libres. Le concepteur doit donc faire face à deux challenges, l'optimisation de qualité de surface de la pièce afin répondre aux exigences fonctionnelles et esthétiques tout en la réalisant en un temps minimum, pour réduire dans la mesure du possible les pertes économiques liées à sa fabrication. La conception et la fabrication assistées par ordinateur (CFAO) est essentielle pour automatiser la production de pièces complexes. Elles intègrent les fonctions de conception, de planification et de fabrication dans le processus de production.

Dans cet ordre d'idées ce sujet de projet de fin d'études a été proposé par l'équipe CFAO de la division productive et robotique du CDTA, afin de mener des développements dans le domaine de fabrication des surfaces gauches par enlèvement de matière en portant sur le triptyque indispensable en usinage : pièce, outil et machine-outil.

## Problématique

Les dimensions de la pièce obtenues par fraisage ne peuvent pas toujours correspondre exactement aux dimensions idéales exigées par le concepteur, en raison des inévitables imprécisions liées aux procédés de fabrication. Une pièce non conforme entraîne une perte de productivité, de temps et d'argent. Afin d'éviter cela, il est essentiel de prédire la qualité de la surface finie pour garantir le respect des dimensions et la qualité de surface, avant la production sur la machine-outil. Trois approches de prédictions existent : l'approche analytique, l'approche empirique et l'approche par simulation d'enlèvement de matière.

La précision de la simulation d'usinage dépend de la représentation la plus fidèle du phénomène de coupe. Cette dernière peut se faire géométriquement et physiquement. Cependant, la représentation physique nécessite une représentation géométrique préalable, ce qui souligne l'importance de la représentation géométrique. Elle consiste à modéliser la pièce à usiner, le trajet d'outil ainsi que le volume balayé par l'outil. La modélisation discrète est préférée par rapport à la modélisation continue vu sa simplicité et son implémentation directe. La précision de cette dernière est liée au nombre d'éléments géométriques de discrétisation, qui est étroitement lié au temps de calcul.

L'optimisation du temps de calcul est donc un enjeu majeur pour les industriels qui cherchent à réduire les délais de production et les coûts associés. Notre problématique consiste à optimiser les temps liés à la simulation d'enlèvement de matière afin d'assurer une prédiction qualitative de la topographie 3D des surfaces gauches, lors de l'opération de finition en usinage en bout avec des outils hémisphériques sur des fraiseuses multiaxes.

L'auteur K. BOUHADJA propose des techniques permettant l'optimisation du temps de calcul en utilisant des approches de classification. Ce présent travail est une suite de ses travaux, qui consiste à développer une meilleure approche de classification permettant de partitionner intelligemment les pièces de formes libres usinées sur des fraiseuses multiaxes, et ceci en utilisant des techniques d'IA.

## Objectif de travail

L'objectif de ce travail est de proposer une démarche de classification permettant l'optimisation du temps de calcul. Ceci en développant un module logiciel graphique et interactif permettant d'introduire différentes données et paramètres liés à la simulation géométrique discrète et d'interpréter les résultats obtenus en termes de qualité de partitionnement, temps de calcul et d'état de surface. Ce module exploitera différentes méthodologies de classifications des éléments géométriques afin d'optimiser leurs générations, puis seront par la suite exploités pour simuler l'enlèvement de matière lors de l'opération de finition.

## Organisation du document

Le présent document est constitué de quatre chapitres :

**Chapitre 1 Généralités et état de l'art :** Ce premier chapitre introduit les concepts fondamentaux nécessaires dans le processus de production des pièces de formes libres ainsi que les méthodes de simulation virtuelle en usinage multiaxes. Ce chapitre analysera par la suite, les défauts liés à la fabrication ainsi que la prédiction de la qualité d'une surface.

**Chapitre 2 Outils mathématiques :** Ce deuxième chapitre explore les outils mathématiques permettant le partitionnement intelligent de données, en abordant les techniques spécifiques d'apprentissage automatique utilisées. Ensuite une étude détaillée le partitionnement par K-Means est proposé en mettant l'accent sur ces critères et variantes.

**Chapitre 3 Analyse et conception :** Ce troisième chapitre présente la démarche chronologique générale du module logiciel développé, en présentant les différents algorithmes, entrées et sorties, et approches utilisées pour atteindre l'objectif.

**Chapitre 4 Implémentation et réalisation :** Ce quatrième et dernier chapitre aborde la phase concrète de la mise en œuvre du module logiciel, en présentant le scénario de fonctionnement général ainsi que des discussions sur les résultats obtenus.

Une conclusion générale clôturera ce travail.

# Chapitre 1

## Généralités et état de l'art

## 1.1 Introduction

Ce chapitre se concentre sur trois sujets principaux relatifs à la production des pièces de formes libres. Nous allons tout d'abord examiner les processus de fabrication, en nous intéressant particulièrement sur le fraisage 5 axes. Ensuite, nous étudierons les différentes méthodes de simulation virtuelle d'enlèvement de matière, les catégories de simulation, les techniques d'approximation d'un modèle par des éléments géométrique (Triangles, Dexels et triple Dexels), ainsi que les stratégies d'usinage associées. Enfin, nous terminerons par l'identification et l'analyse des défauts de surface des pièces de forme libre, avec les approches utilisées pour quantifier ces derniers.

## 1.2 Processus de production des pièces de forme libre

Le processus d'usinage par enlèvement de matière est un type de processus de fabrication dans lequel le produit final est obtenu en éliminant l'excès de métal de la matière première [1]. Cette technique est réalisée à l'aide d'une machine-outil et permet d'obtenir des pièces de grandes précisions .

### 1.2.0.1 Types d'usinage par enlèvement de matière

Les processus d'usinage par enlèvement de matière sont principalement divisés en deux types : traditionnels et non traditionnels [1].

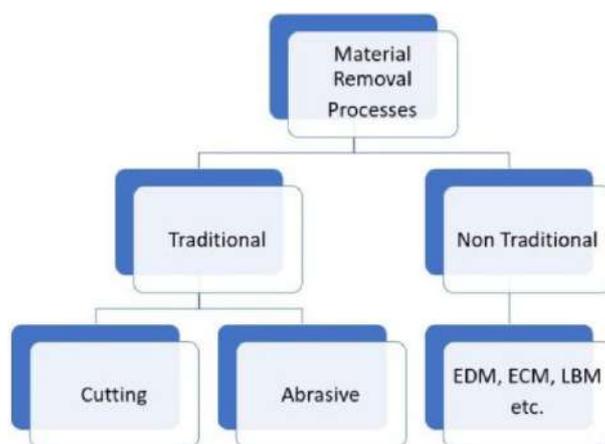


FIGURE 1.1 – Types d'usinage par enlèvement de matière [1].

- **Les processus d'usinage par enlèvement traditionnel** : Ils sont utilisés pour une large gamme de fabrication de produits généraux. Des exemples de ces processus sont *le tournage, le perçage, le lappage, etc.*
- **Les processus d'usinage par enlèvement de matière non traditionnel** : Processus utilisés pour des travaux d'ingénierie de très haute qualité. Des exemples de processus d'usinage par enlèvement de matière non traditionnel incluent *l'usinage électrochimique, l'usinage par décharge électrique, et d'autres.*



FIGURE 1.2 – Exemple d'usinage d'un vilebrequin.

### 1.2.0.2 Avantages et limitations de l'usinage par enlèvement de matière

Le processus d'usinage par enlèvement de matière offre plusieurs avantages par rapport aux autres processus de fabrication, malgré les limitations présentes [1].

#### A. Avantages

- Une grande précision dimensionnelle pour des pièces de qualité supérieure ;
- Une obtention de coins droits qui ne sont pas réalisables avec d'autres techniques de production ;
- Une Polyvalence du processus, permettant de produire une grande variété de pièces avec différentes formes et tailles ;
- La Rentabilité du processus, permettant de produire de grands volumes de pièces avec une qualité constante.

#### B. Limitations

- Limitation de la taille des pièces et composants ;
- L'exigence d'un opérateur hautement qualifié lors de l'usinage manuel ;
- Temps de production relativement long par rapport aux autres processus ;
- Génération de déchets (copeaux) important, pouvant être couteux à éliminer ;
- Problèmes liés à la dureté et usinabilité du matériau.

### 1.2.1 Pièces de formes libres

Une surface gauche est une surface engendrée par le mouvement d'une droite, dont deux positions successives ne sont généralement pas dans un même plan [2]. Les surfaces gauches sont utilisées dans l'usinage de pièces mécaniques pour obtenir des formes complexes et précises.



FIGURE 1.3 – Exemple de pièces de formes libre.

Les techniques de conception de surfaces non développables en CAO sont divisées en deux grandes catégories : les méthodes qui utilisent des courbes et celles qui se basent sur des points [3] [4], la figure 1.4 illustre les surfaces basées sur l'utilisation des courbes.

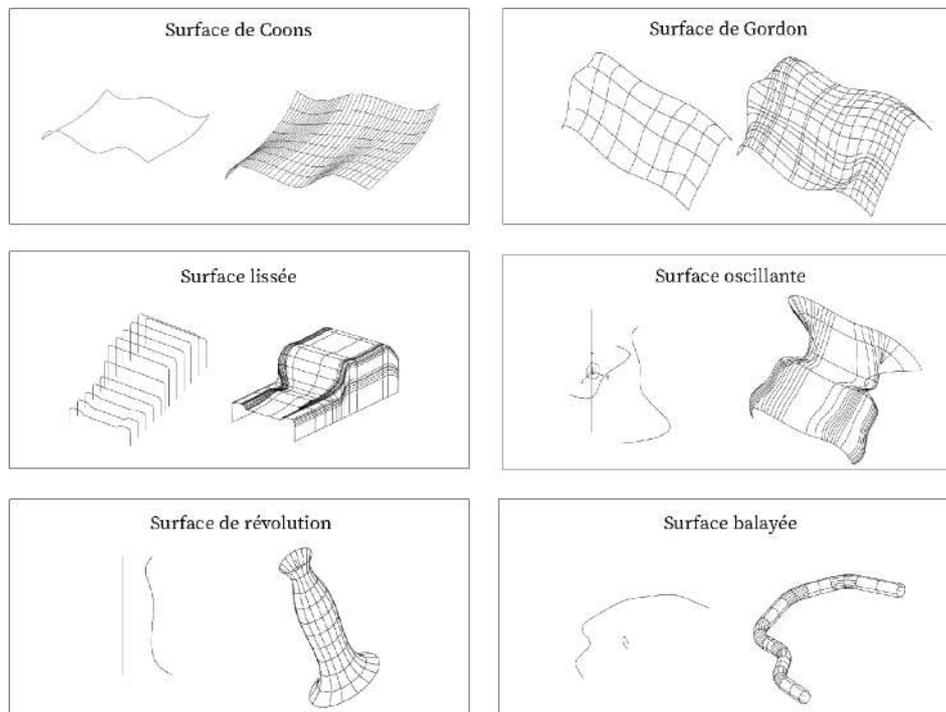


FIGURE 1.4 – Surfaces basées sur l'utilisation des courbes.

### 1.2.2 Machine de fraisage multi-axes

Le fraisage est un processus d'usinage couramment utilisé qui consiste à enlever de la matière d'une pièce en faisant avancer une fraise rotative [5]. Ce processus comprend une variété d'opérations et de machines, allant du fraisage de petites pièces individuelles à des opérations de fraisage de groupe pour de grands composants. En raison de sa capacité à produire des pièces personnalisées avec des tolérances précises, le fraisage est considéré comme l'un des processus d'usinage les plus utilisés.

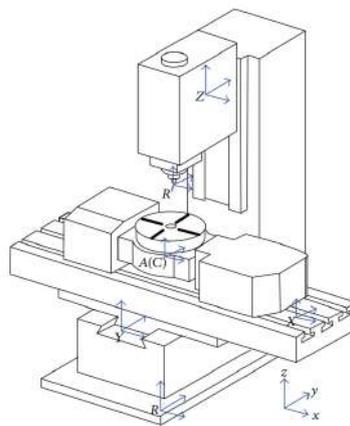


FIGURE 1.5 – Représentation d'une fraise [6].

Pour réaliser ce processus, il est nécessaire d'avoir quatre composants essentiels : une machine de fraisage, une pièce à travailler, un gabarit et une fraise. La pièce à travailler est un matériau préformé qui est fixé à un dispositif, monté ensuite sur une plate-forme à l'intérieur de la machine de fraisage. La matière est retirée en effectuant de nombreuses petites coupes à l'aide d'une fraise à plusieurs dents, qui est soit tournée à grande vitesse tout en faisant avancer lentement la matière à travers la fraise, soit une combinaison des deux techniques.

### 1.2.2.1 Types de processus de fraisage

Le processus de fraisage est largement classé en deux catégories principales [5] :

- La première catégorie est le fraisage de face, où l'action de coupe primaire se produit aux coins de l'extrémité de la fraise. Le fraisage de face est principalement utilisé pour couper des surfaces plates dans la pièce ou créer des cavités à fond plat.
- La deuxième catégorie est le fraisage périphérique, où l'action de coupe se produit principalement le long de la périphérie de la fraise. Par conséquent, la forme de la surface fraisée ressemble à celle de la fraise. Ici, les lames de la fraise sont responsables de l'enlèvement de matière de la pièce. Le fraisage périphérique est particulièrement adapté pour couper des rainures profondes, des filetages et des dents d'engrenage.

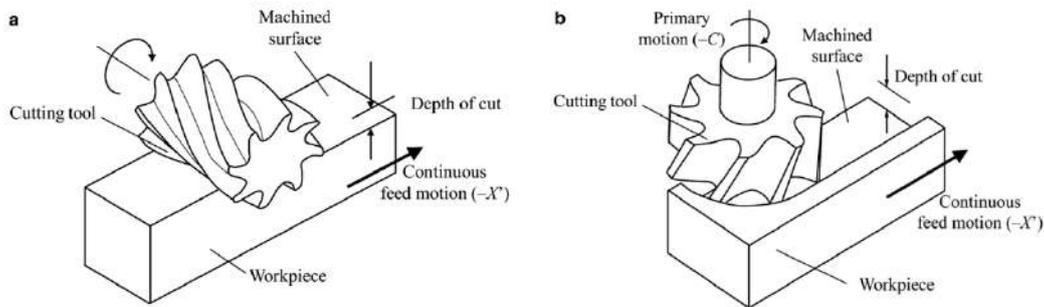


FIGURE 1.6 – Représentation des types de processus de fraisage [7].

### 1.2.2.2 Types de machines

Il existe plusieurs types de fraisage en fonction des spécifications de la pièce à fabriquer, y compris l'usinage en 2.5 axes, en 3 axes, en 4 axes et en 5 axes. À mesure que le nombre d'axes augmente, la complexité et la précision des pièces qui peuvent être fabriquées augmentent également [8].

- Usinage 2.5 axes : Il s'agit d'une forme d'usinage en 2D qui implique l'utilisation d'un outil de coupe pour enlever de la matière d'une pièce en deux dimensions (X et Y) tout en se déplaçant verticalement le long de l'axe Z. L'outil de coupe ne peut se déplacer qu'en haut et en bas le long de l'axe Z et ne peut pas basculer ou pivoter. Ce type d'usinage est souvent utilisé pour des pièces simples qui ne nécessitent pas de formes ou d'angles complexes.
- Usinage 3 axes : L'outil de coupe peut se déplacer le long des trois axes, mais il ne peut ni basculer, ni pivoter. L'usinage en 3 axes est couramment utilisé pour la fabrication de pièces simples, le perçage de trous et la création de surfaces planes.
- Usinage 4 axes : L'ajout d'un quatrième axe permet à l'outil de coupe de basculer ou de pivoter, permettant la création de pièces plus complexes avec des surfaces courbes ou angulaires.
- Usinage 5 axes : L'ajout d'un cinquième axe permet à l'outil de coupe de basculer et de pivoter dans plusieurs directions, permettant la création de pièces encore plus complexes avec des caractéristiques telles que des contre-dépouilles et des angles composés.

Les machines avec plus d'axes peuvent être plus coûteuses et plus difficiles à programmer, donc le choix de la forme d'usinage à utiliser dépend des exigences spécifiques de la pièce à fabriquer.

### 1.2.3 Usinage 5 axes CNC

L'usinage 5-axes offre une infinité de possibilités en terme de taille et de forme des pièces que l'on peut produire. Le terme "5 axes" se réfère au nombre de directions dans lesquelles l'outil de coupe peut se déplacer. Sur un centre d'usinage 5 axes, l'outil de coupe se déplace le long des axes linéaires  $X$ ,  $Y$  et  $Z$  et tourne autour de deux des trois axes  $A$ ,  $B$  et  $C$  pour travailler sur la pièce dans toutes les directions. Offrant un niveau de flexibilité et de précision inégal par rapport à d'autres méthodes d'usinage [9].

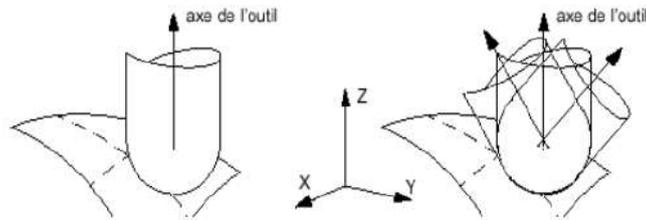


FIGURE 1.7 – Comparaison entre l'inclinaison de l'outil : trois axes et cinq axes [9].

### 1.2.3.1 Types de machines

Différents types de machines à 5 axes existent, avec différentes conceptions architecturales, permettant l'orientation de l'outil par rapport à la pièce, elles sont catégorisées en trois familles [10] :

- **Machine R' R'** : Deux axes de rotation pour orienter la table.
- **Machine R R** : Deux axes de rotation pour orienter l'outil.
- **Machine R' R** : Un axe de rotation pour orienter la table et un autre pour l'outil.

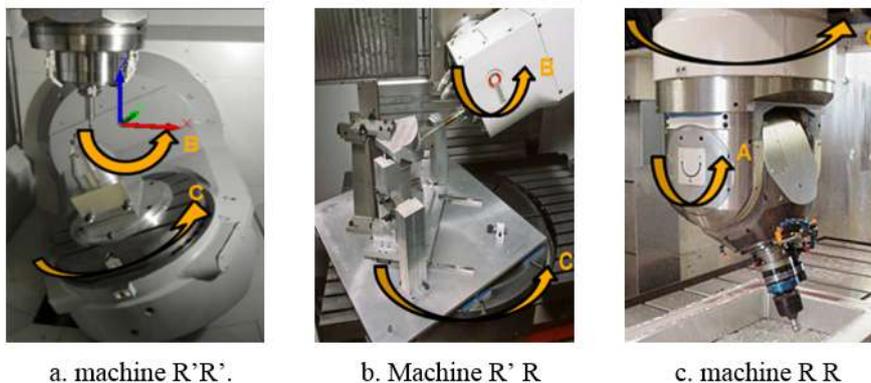


FIGURE 1.8 – Différents types de machines 5 axes.

### 1.2.3.2 Avantages de la CNC

La technique d'usinage à 5 axes offre de nombreux avantages en termes de flexibilité, de précision et de temps d'usinage. Elle permet d'usiner des formes complexes en une seule opération avec un seul programme CNC, ce qui améliore considérablement la productivité tout en réduisant les temps de traitement et de cycle [11]. Cette technique, appelée DONE-IN-ONE, offre des économies en temps et en argent pour la préparation des outils. L'usinage à 5 axes permet aussi d'obtenir une précision accrue des pièces, du fait que la pièce ne se déplace pas à travers plusieurs postes de travail pour être finie.

### 1.2.3.3 Opérations d'usinage

En usinage, les outils jouent un rôle crucial dans le processus d'enlèvement de matière. Pour obtenir une pièce finie de haute qualité, il est souvent nécessaire de réaliser plusieurs opérations successives, chacune ayant des spécificités particulières : d'ébauche, semi-finition, finition et éventuelle superfinition [3].

- **Opération d'ébauche** : Elle consiste à retirer une grande quantité de matière rapidement, afin de rapprocher la pièce de sa forme finale. Les outils utilisés pour l'ébauche doivent être solides et capables de résister à des forces importantes.
- **Opération de semi-finition** : Elle vise à éliminer les irrégularités laissées par l'ébauche et à obtenir une surface plus régulière. Les outils utilisés pour la semi-finition sont plus fins et plus précis que ceux utilisés pour l'ébauche.
- **Opération de finition** : Elle consiste à obtenir une surface lisse et précise sur la pièce. Les outils utilisés pour la finition sont encore plus fins et plus précis que ceux utilisés pour la semi-finition.

- **Opération de super finition** : Il s'agit d'une opération facultative qui permet d'obtenir une surface extrêmement lisse et précise sur la pièce. Les outils utilisés pour la super finition sont très fins et sont souvent associés à des techniques de polissage.

#### 1.2.3.4 Types d'outil d'usinage de finition

Pour l'opération de finition, il existe principalement trois outils : cylindrique, torique et hémisphérique. Le choix de l'outil approprié se fait en fonction des paramètres de rigidité de l'outil, la cinématique de la pièce à usiner ainsi que la forme de la pièce à usiner [3].

- **Outil cylindrique** : Utilisé dans l'usinage des épaulements. Ses faces planes sont utilisées pour l'usinage de surface, tandis que ses côtés sont utilisés pour l'usinage de contour.
- **Outil torique** : Il s'agit d'un outil intermédiaire entre l'outil cylindrique et hémisphérique. Il est de plus en plus utilisé dans l'industrie en raison de son principal avantage : sa partie active, le tore, situé sur un rayon à distance de l'axe de l'outil, dont la vitesse est non nulle [3]. Cependant, il présente le même problème que l'outil cylindrique : si l'orientation de l'axe de l'outil est modifiée, la rotation est effectuée autour de ce point et, par conséquent, la zone de contact outil/matériau est modifiée [12].
- **Outil hémisphérique** : Il s'agit d'un outil qui convient pour la finition de tout matériau, quel que soit sa dureté. Il peut travailler à des vitesses de coupe beaucoup plus élevées que les autres outils. Son utilisation est moins contrainte en raison de sa forme de pointe sphérique qui est invariante par rotation autour de son centre.



FIGURE 1.9 – Types d'outils utilisés dans le fraisage 5 axes [3].

Les opérations de fraisage nécessitent l'utilisation d'outils spécifiques associés à des trajectoires simples. Cependant, pour l'usinage de surfaces complexes, il n'existe pas d'outils qui permettent d'obtenir des formes en un seul mouvement élémentaire. Les outils les plus simples pour l'usinage des surfaces gauches sont les fraises sphériques, mais le choix entre la pointe sphérique et torique est souvent arbitraire ou conditionné par la géométrie locale (courbure) des surfaces [12].

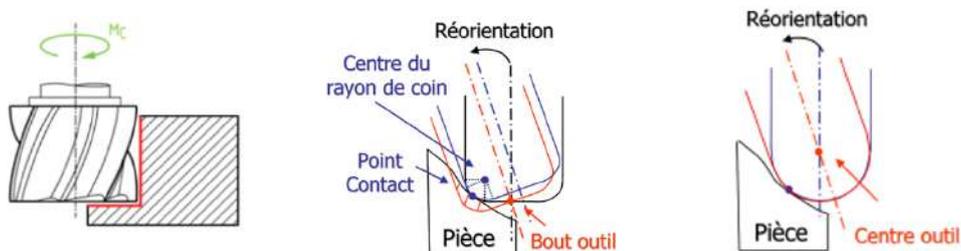


FIGURE 1.10 – Réorientation des différents types d'outils [12].

## 1.3 Méthodes de simulation virtuelle d'usinage multiaxes

La simulation d'usinage multiaxes consiste en la quantification de la matière enlevée par l'outil de coupe, ainsi que la détermination de la trace de l'outil sur la surface usinée. Ce processus est particulièrement complexe pour les surfaces de formes gauches.

### 1.3.1 Simulation d'usinage multiaxes sur pièces de formes libres

La littérature abonde en travaux de recherche portant sur les différents problèmes liés à la simulation d'usinage [13], notamment la simulation géométrique ou physique, à différentes échelles (humaine, macroscopique, microscopique) et avec différents types de modèles (géométriques, dynamiques).

#### 1.3.1.1 Catégorie de simulation

La simulation se divise en deux catégories principales la simulation géométrique ainsi que la simulation physique :

- **Simulation géométrique** : Elle permet de vérifier graphiquement les interférences, les collisions et les tolérances imposées par le concepteur, tout en fournissant des informations géométriques utiles pour la simulation physique.
- **Simulation physique** : Elle permet de caractériser les aspects physiques du processus d'usinage, tel que les forces de coupe, les vibrations, la rugosité de surface, la température d'usinage et l'usure de l'outil. Elle s'appuie sur la simulation géométrique et sur le choix adéquat du matériau de l'outil de coupe.

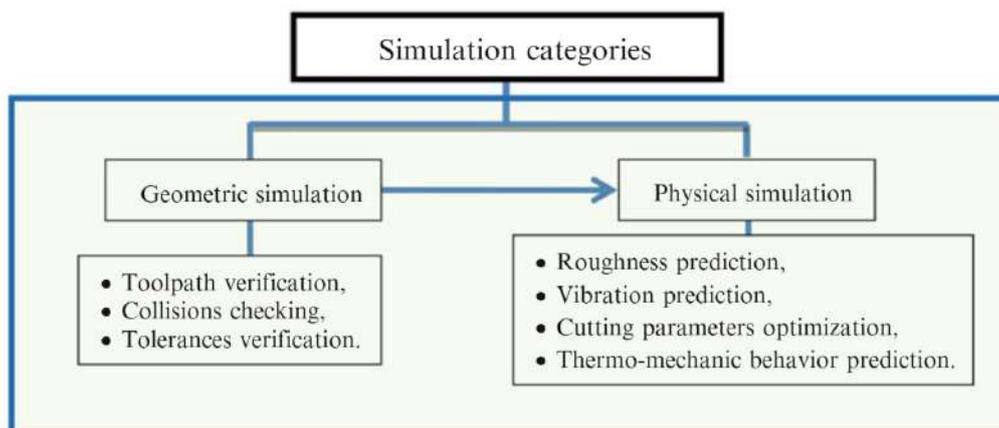


FIGURE 1.11 – Catégories de simulation [13].

#### 1.3.1.2 Échelle de simulation

Afin d'étudier l'usinage de manière efficace, il est courant d'utiliser une approche multiéchelle qui permet de réduire la complexité en limitant le nombre de phénomènes à considérer et en adaptant la taille du modèle à chaque échelle. Cette approche peut être appliquée à trois niveaux d'analyse différents :

- **Échelle humaine** : Elle a pour objectif de prédire le comportement des moyens de production et de préparer le processus d'usinage en considérant les mouvements des axes, la position de la pièce à usiner sur la table et l'espace de travail disponible.
- **Échelle macroscopique** : Elle permet de déterminer le volume de matière enlevé pour chaque mouvement d'outil lors de l'usinage de la pièce.
- **Échelle microscopique** : Elle permet de déduire certaines propriétés de la structure matérielle, notamment la loi de comportement du matériau utilisé.

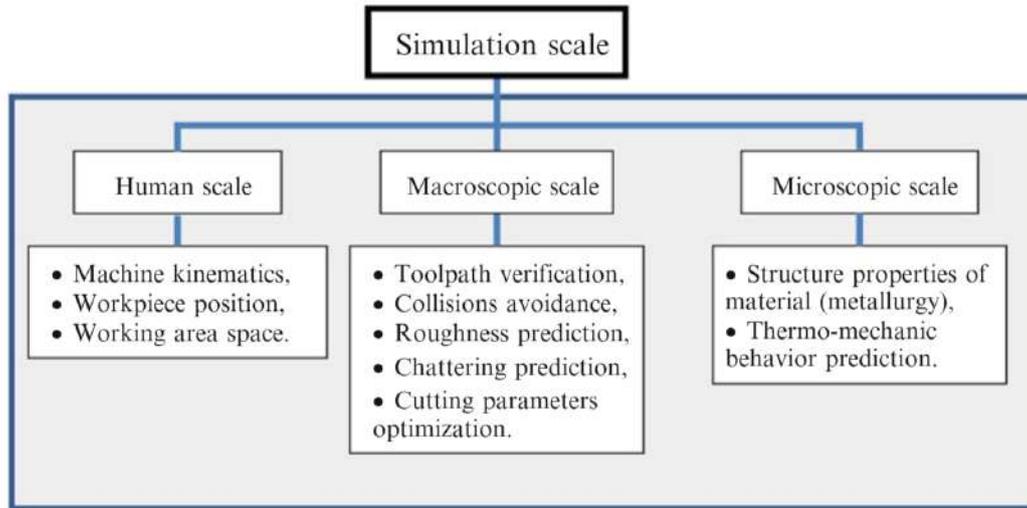


FIGURE 1.12 – Catégories des échelles de simulation [13].

### 1.3.1.3 Modèle de simulation

Afin de représenter les interactions entre la pièce, l'outil et la machine à l'échelle macroscopique, un modèle dynamique et géométrique sont utilisés dans la simulation du système Pièce-Outil-Machine.

- **Modèle dynamique** : Il est utilisé pour le système Porte-Outil-Machine pouvant être associé à un modèle simple de type masse-ressort ou un modèle complexe utilisant la méthode des éléments finis.
- **Modèle géométrique** : Différentes classifications existent dans la littérature pour la représentation géométrique en simulation à l'échelle macroscopique. Les méthodes utilisées peuvent être classées en quatre catégories : le mode filaire, le mode solide, le mode espace objet et le mode espace image.

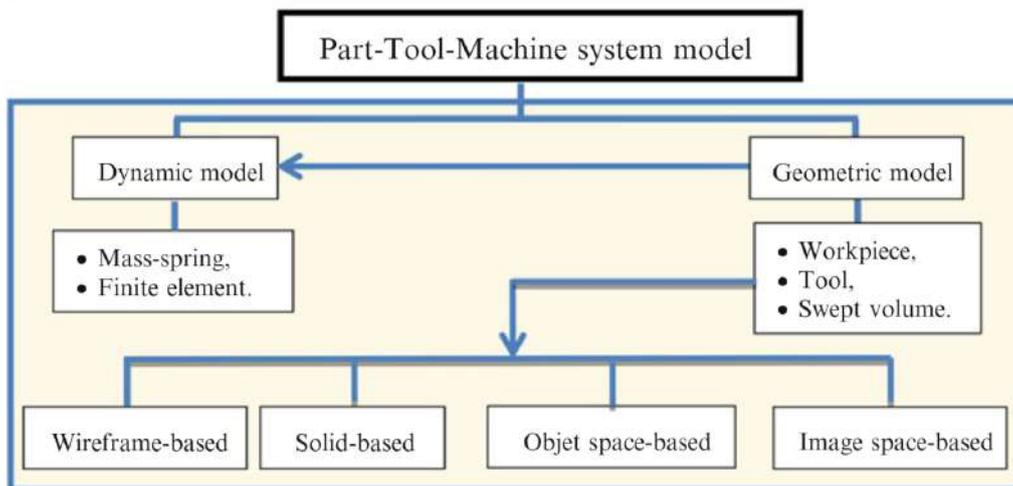


FIGURE 1.13 – Catégories des modèles de simulation [13].

Dans le contexte de la simulation d'usinage à l'échelle macroscopique, il existe une méthode de représentation géométrique appelée "espace image". Dans cette méthode, les pièces sont représentées par des pixels en profondeur (Dexel), qui est une extension du modèle Z-buffer utilisé dans la suppression des parties cachées en infographie.

### 1.3.2 Techniques d'approximation des modèles pour la simulation d'usinage

Les modèles discrets présentent de nombreux avantages pour le suivi des simulations d'usinage. Leur complexité peut varier, allant de représentations simples à des modèles plus complexes. Ces modèles discrets offrent une représentation plus détaillée de la géométrie de la pièce et de l'outil, ce qui se traduit par une simulation plus précise de l'usinage [14].

#### 1.3.2.1 Approximation de la surface gauche par fichier .STL

L'extension de fichier .STL est un format natif du logiciel de CAO stéréolithographique créé par 3D Systems, appelé "Standard Triangle Language". Ces fichiers contiennent une description géométrique d'une surface en trois dimensions sous la forme de facettes triangulaires [15].

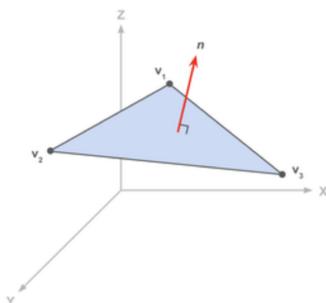


FIGURE 1.14 – Représentation d'un triangle issu de fichier .STL.

Les fichiers .STL contiennent une liste de données de facettes qui définissent la surface de l'objet en trois dimensions. Chaque facette est définie par trois sommets et une normale unitaire qui est perpendiculaire à la surface du triangle. Les coordonnées de chaque sommet sont spécifiées dans un système de coordonnées cartésiennes à trois dimensions, et la normale est définie comme une ligne perpendiculaire au triangle.

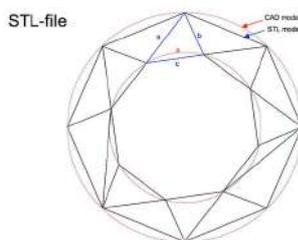


FIGURE 1.15 – Différence entre le modèle CAO et le modèle STL.

Chaque triangle doit partager deux sommets avec chacun de ses triangles adjacents. En d'autres termes, un sommet d'un triangle ne peut pas se trouver sur le côté d'un autre triangle.

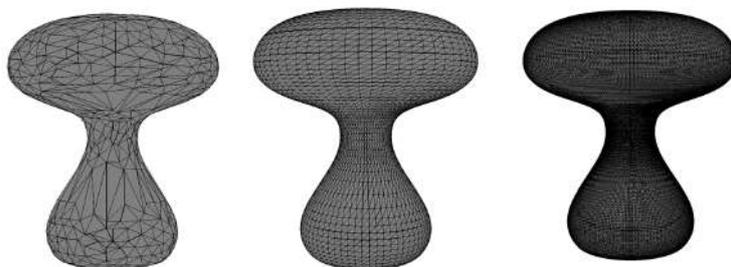


FIGURE 1.16 – Comparaison entre trois modèles STL (1556, 8442 et 79600 triangles respectivement).

#### • Extension ASCII et Binaire

Il existe deux formats différents pour les fichiers .STL, binaire et ASCII, qui diffèrent en fonction de leur objectif et de leur structure [16].

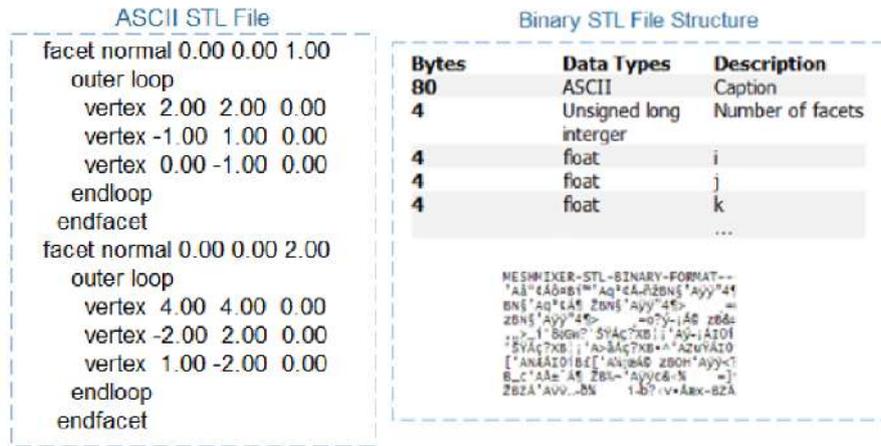


FIGURE 1.17 – Représentation d'un fichier .STL ASCII et Binaire [16].

- La principale différence technique entre les fichiers .STL binaires et ASCII réside dans leur taille, les fichiers .STL binaires étant bien plus optimisés et prenant moins d'espace que les fichiers .STL ASCII en raison de leur optimisation.
- L'autre aspect qui différencie ces deux types de codage est la lisibilité des fichiers. Alors que les fichiers .STL binaires ne peuvent pas être lus directement par l'utilisateur, les fichiers .STL ASCII peuvent être ouverts dans un éditeur de texte et leur code peut être lu et compris.

### 1.3.2.2 Approximation d'un modèle volumique

L'approximation des modèles volumiques permet une suivie suivie de l'usinage par enlèvement de matière, l'approximation se fait généralement par des Voxels, Dexels ou Triple Dexels.

#### A. Voxel

Dans le domaine de l'infographie, un Voxel (ou "pixel 3D") est un cube de trois dimensions positionné sur une grille tridimensionnelle utilisée pour créer des modèles 3D. Chaque Voxel contient des informations de position ainsi qu'une valeur de couleur attribuée, et tous les Voxels individuels ont la même taille. Les Voxels peuvent ensuite être combinés dans la grille à différentes positions avec des couleurs différentes pour créer un modèle de Voxel [17].

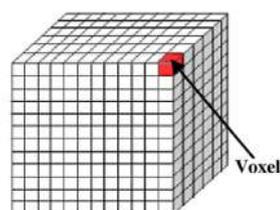


FIGURE 1.18 – Représentation d'un Voxel.

- **Avantages** : Les modèles de Voxel ont des propriétés quasiment isotropes et offrent une description volumétrique correcte du volume de l'objet [18].
- **Limitations** : Les modèles de Voxel nécessitent un effort de calcul plus important pour modifier la géométrie en raison du nombre d'interactions Voxel individuelles [18].

## B. Dixel

Le Dixel est une méthode de représentation issue de l'imagerie, qui est souvent utilisé pour obtenir une présentation volumique d'un modèle surfacique (ex STL). Pour cette méthode, il est nécessaire de projeter une grille sur la surface dans une direction donnée en fonction du point de vue choisi. Les Dexels sont alors représentés par des parallélépipèdes ayant une section spécifique et une hauteur équivalente à la longueur du segment de type matière [14].

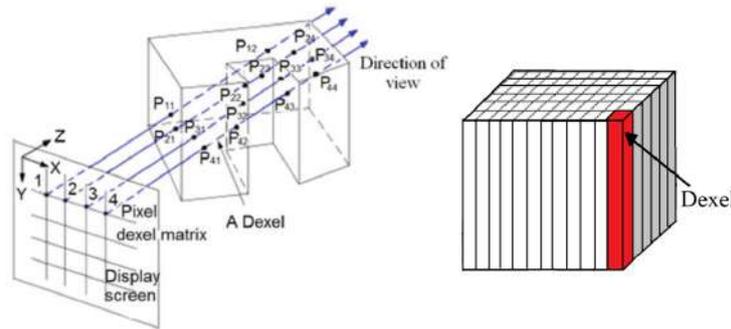


FIGURE 1.19 – Projection des grilles sur un modèle [13].

## C. Triple Dixel

Le modèle Dixel trouve ses limites dans le cas où les surfaces du modèle sont quasi-perpendiculaire à la direction de projection. Pour y remédier, une modélisation Triple-Dixel est proposée, qui utilise des rayons dans les trois directions orthogonales pour discrétiser le modèle. Cette méthode a été démontrée comme étant plus précise que le modèle simple Dixel pour la reconstruction de surfaces, avec une même résolution [14].

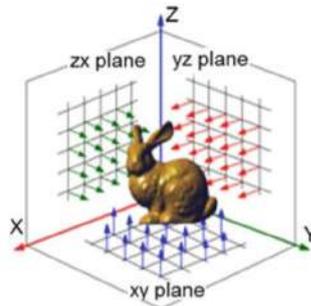


FIGURE 1.20 – Projection des grilles sur un modèle surfacique [13].

En ce qui concerne la simulation d'enlèvement de matière, il est crucial de choisir une méthode de modélisation précise et efficace. Dans ce contexte, le Triple-Dixel est souvent privilégié en raison de sa précision et de son efficacité. Contrairement au Voxel qui nécessite de stocker des informations pour toutes les zones intérieures du modèle, le Triple-Dixel permet une utilisation plus efficace de la mémoire en consommant moins d'espace. Cela est particulièrement avantageux pour les modèles complexes avec de grandes zones intérieures. En optant pour le Triple-Dixel pour la simulation d'usinage, des résultats plus précis sont obtenus, tout en assurant une efficacité accrue dans l'utilisation de la mémoire [12].

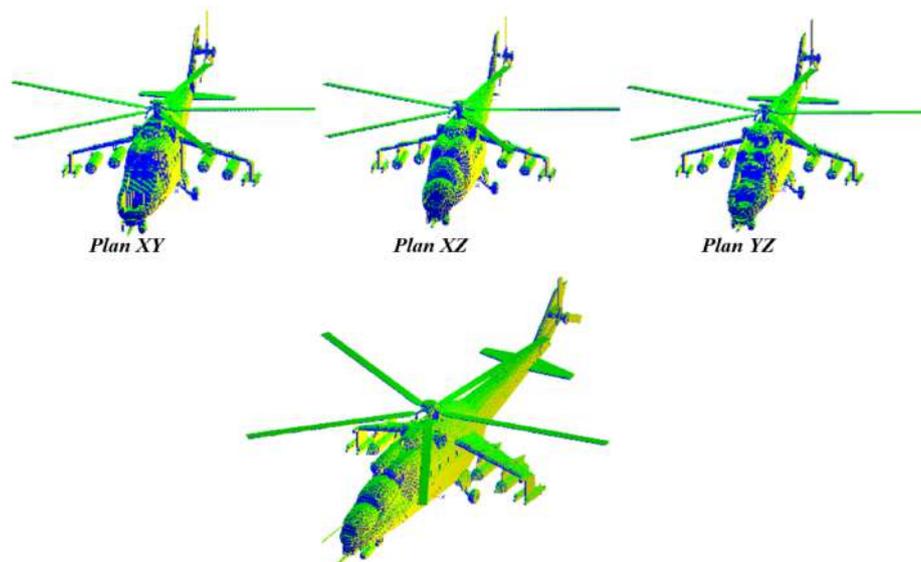


FIGURE 1.21 – Représentation d'un modèle dexel et triple dexel [12].

### 1.3.3 CFAO

La CFAO représente une superposition de la CAO et de la FAO, elle joue un rôle crucial dans l'usinage des pièces de formes gauches, permettant la visualisation et simulation du procédé.

#### 1.3.3.1 CAO

La Conception Assistée par Ordinateur CAO (*Computer Aided Design, CAD*) comprend l'ensemble des logiciels et des techniques de modélisation géométrique permettant de concevoir, de tester virtuellement et des techniques de simulation numérique. Elle est principalement utilisée pour créer des modèles 2D et 3D hautement précis.

Les logiciels de CAO permettent de visualiser des propriétés telles que la hauteur, la largeur, la distance, la couleur et le matériau, ce qui permet de construire des modèles complets pour un vaste domaine application : L'architecture, la construction, l'ingénierie, l'impression 3D, la menuiserie et la fabrication de métaux représentent une partie des nombreuses industries qui utilisent la CAO [19].

#### 1.3.3.2 FAO

La Fabrication Assistée par Ordinateur FAO (*Computer Aided Manufacturing, CAM*) désigne généralement l'utilisation de logiciels de commande numérique pour créer des instructions détaillées (code G) qui pilotent les machines-outils à commande numérique par ordinateur CNC pour la fabrication de pièces [20].

L'utilisation de la FAO présente plusieurs avantages lorsqu'il s'agit de créer des composants dans divers domaines. Par rapport aux machines manuelles, la FAO offre généralement [20] :

- Une plus grande rapidité dans la production de composants.
- Une précision et une cohérence accrues.
- Une plus grande efficacité, éliminant le besoin de pauses.
- Une grande sophistication en termes de suivi de motifs complexes.

### 1.3.3.3 Processus de la CFAO

La CFAO est une technique d'automatisation de la production qui intègre la conception, la planification et la fabrication en deux étapes principales : la construction d'un modèle géométrique et la génération de trajectoires d'outils, essentielle à la production [3].

- **Construction du modèle géométrique** : La première étape de la CFAO consiste à créer un modèle géométrique qui exprime toutes les contraintes fonctionnelles décrites dans le cahier des charges à l'aide d'un système de CAO. Ce modèle est constitué de surfaces ou d'éléments géométriques.
- **Génération de trajectoires d'outils** : La deuxième étape de la CFAO, appelée fabrication, consiste à transformer les éléments géométriques décrits à l'aide d'un système CAO en une forme physique qui permet à l'objet de remplir sa fonction. Cela se fait à l'aide d'un système FAO qui génère des trajectoires d'outils pour la fabrication.
- **Trajectoires en format neutre** : Trajectoires d'outils exprimées dans le repère de la pièce à usiner, mais ne prennent pas en compte la cinématique de la machine ni la position de la pièce sur le plateau. Elles sont non modifiables et nécessitent un postprocesseur pour être compréhensibles par la commande numérique, avec le format neutre le plus courant étant le CL-FILE.
- **Les trajectoires en format "machine"** : Trajectoires d'outils exprimées dans un langage compréhensible par la commande numérique, généralement le langage ISO (aussi G-Code ou M-Code) . Fortement dépendant de la machine. Le langage ISO contient deux types d'ordres : les ordres "G" pour définir la trajectoire et les ordres "M" pour d'autres fonctions. Ces trajectoires sont modifiables, car le langage de programmation est simple et compact.
- **Postprocesseur** : programme qui convertit les trajectoires de format neutre (CL-FILE) en trajectoires compréhensibles par la machine en prenant en compte ses capacités et sa cinématique. Il détecte les erreurs, exprime les trajectoires dans le langage spécifique à la machine et chaque machine possède son propre postprocesseur.
- **Validation et simulation** : Il s'agit d'une phase qui permet de vérifier la méthode de calcul des trajectoires d'outils et la stratégie d'usinage optimale. Une simulation est réalisée pour détecter tout problème tel que des collisions ou des interférences, ainsi que pour résoudre d'autres problèmes liés aux paramètres d'usinage et aux conditions opératoires.

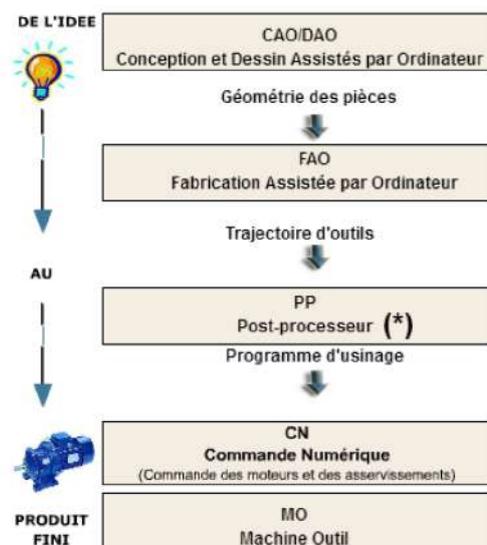


FIGURE 1.22 – Démarche générale du processus de CFAO .

### 1.3.4 Stratégies d'usinages

Les stratégies d'usinage sont des méthodes utilisées pour exécuter des opérations d'usinage précises et efficaces. Les trajets d'usinage déterminent le mouvement de l'outil de coupe par rapport à la pièce. Les fichiers .CL et G-CODE sont utilisés pour programmer les machines-outils à commande numérique pour l'usinage des pièces.

#### 1.3.4.1 Trajet d'usinage

- **Stratégies isoparamétriques** : Elle consiste à balayer une surface  $S(u, v)$  en suivant une succession de courbes isoparamétriques  $P(v)$ , obtenues en fixant l'un des paramètres  $u$  ou  $v$  de la surface. Les paramètres de discrétisation de la trajectoire suivent alors les variations  $\Delta u$  et  $\Delta v$  dans les deux directions [3].

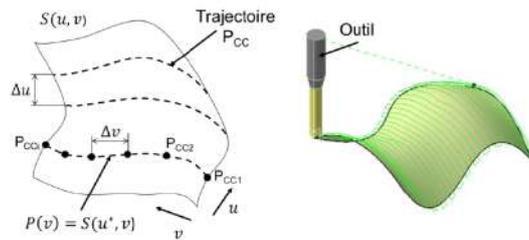


FIGURE 1.23 – Trajet d'outil issu d'une stratégie isoparamétrique.

- **Aller simple (One-Way)** : Elle consiste à suivre une isoparamétrique  $u$  ou  $v$  jusqu'à ce qu'il atteigne sa fin, puis de déplacer l'outil selon l'autre isoparamétrique. Ce processus est répété pour usiner l'ensemble de la surface.
- **Aller-retour (Zig-Zag)** : Similaire à l'aller simple, elle consiste à suivre une isoparamétrique  $u$  ou  $v$  jusqu'à ce que l'outil atteigne, puis se déplacer selon l'autre isoparamétrique de manière continue, jusqu'à l'usinage total de la surface.

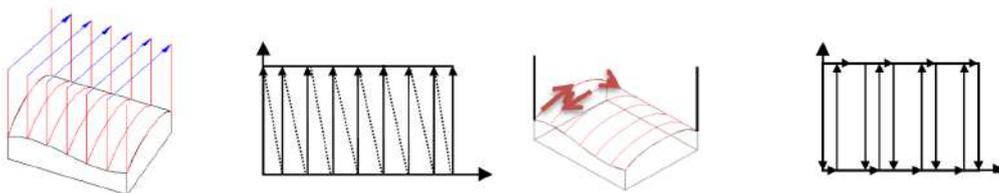


FIGURE 1.24 – Trajet d'outil one-way (droite) et Zig-zag (gauche)

- **Concentrique** : Elle consiste à suivre une trajectoire circulaire autour de l'axe de la pièce à usiner tout en se déplaçant simultanément vers l'intérieur ou l'extérieur de la pièce.
  - **Spiral in** : La trajectoire de l'outil d'usinage suit une spirale pour couvrir toute la surface, en partant de l'extérieur en se déplaçant progressivement vers l'intérieur.
  - **Spiral out** : La trajectoire de l'outil d'usinage suit une spirale pour couvrir toute la surface, en partant de l'intérieur en se déplaçant progressivement vers l'extérieur.
- **Radial** : La surface est usinée en partant du centre en se déplaçant jusqu'au bord, jusqu'à ce que toute la surface soit usinée.

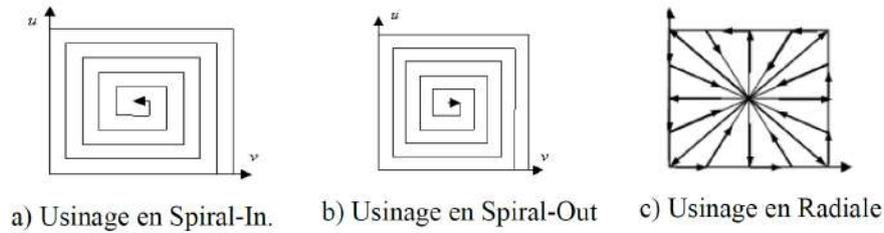


FIGURE 1.25 – Trajet d'outil issu d'une stratégie concentrique.

- **Stratégie par plan parallèle** : Elle consiste à balayer une surface  $S(u, v)$  en suivant une succession de courbes parallèles  $P(v)$ , les chemins d'outils sont parallèles dans l'espace euclidien [3].

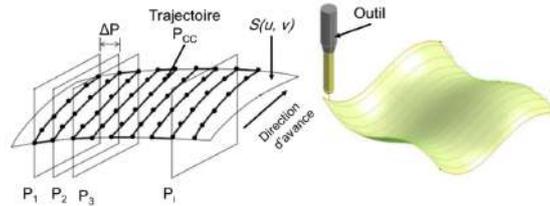


FIGURE 1.26 – Trajet d'outil issu d'une stratégie par plan parallèle.

- **Stratégie Z-constant** : Elle consiste à diviser la surface en plusieurs plans parallèles à l'axe Z. Cette méthode permet de minimiser les longues descentes et montées d'outil, ce qui peut réduire le temps d'usinage et l'usure de l'outil [3].

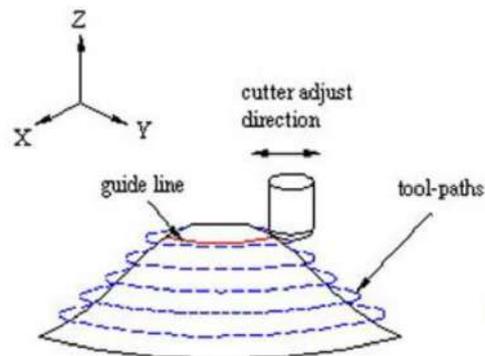


FIGURE 1.27 – Trajet d'outil issu d'une stratégie z-constant.

## 1.4 Défauts de Fabrication

Lors des processus d'usinage, les dimensions des pièces fabriquées peuvent ne pas correspondre aux dimensions du modèle CAO exigées par le concepteur en raison des imprécisions inévitables qui se produisent pendant la fabrication. Ces imprécisions peuvent se manifester sous forme d'imperfections qui entraînent des défauts de fabrication provenant de différentes sources [3].

### 1.4.1 Identification des défauts

L'origine des défauts de fabrication de la pièce est classée en deux catégories principales : les écarts de reprise et les écarts d'usinage [3].

#### 1.4.1.1 Écarts de reprise

Il s'agit des écarts de positionnement de la pièce pendant la mise en position et le maintien dans le porte-pièce qui sont à l'origine des défauts de reprise en fabrication [21].

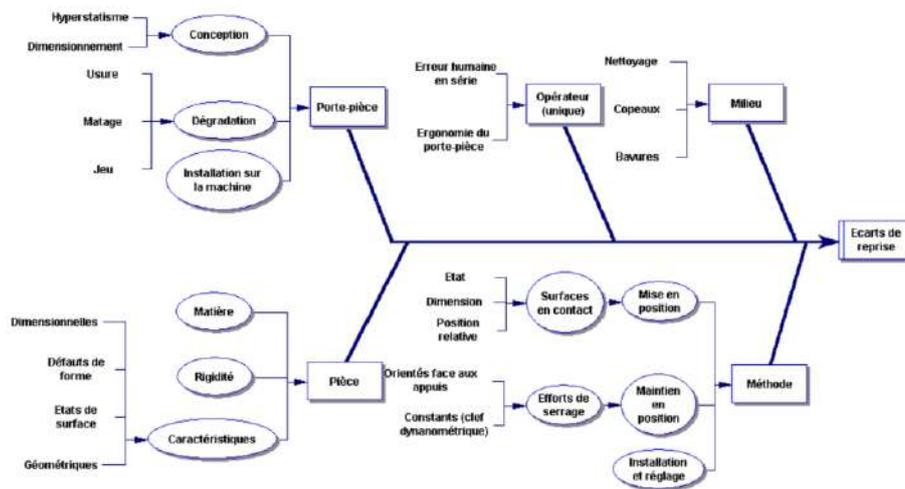


FIGURE 1.28 – Diagramme d'Ishikawa - Écarts de reprise [21].

#### 1.4.1.2 Écarts d'usinage

Les écarts d'usinage sont des différences entre la surface visée lors de l'opération d'usinage et la surface réellement obtenue. Ces écarts sont causés par des phénomènes qui se produisent pendant l'usinage et peuvent dépendre de plusieurs paramètres liés à l'opération, l'outil, le milieu, la machine, etc. [21].

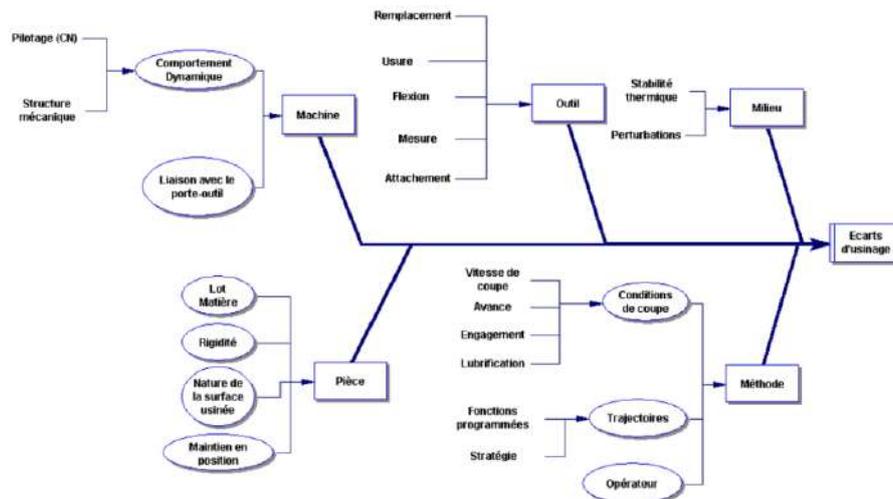


FIGURE 1.29 – Diagramme d'Ishikawa - Écarts de d'usinage [21].

### 1.4.2 Analyse des défauts d'usinage

L'analyse des défauts d'usinage est une étape importante dans le processus de CFAO, car chaque étape de ce processus peut générer des imperfections et des erreurs. Pour identifier ces défauts, l'auteur [22] propose une classification en trois écarts, à savoir  $e_1$ ,  $e_2$  et  $e_3$ . Ces écarts sont définis comme suit :

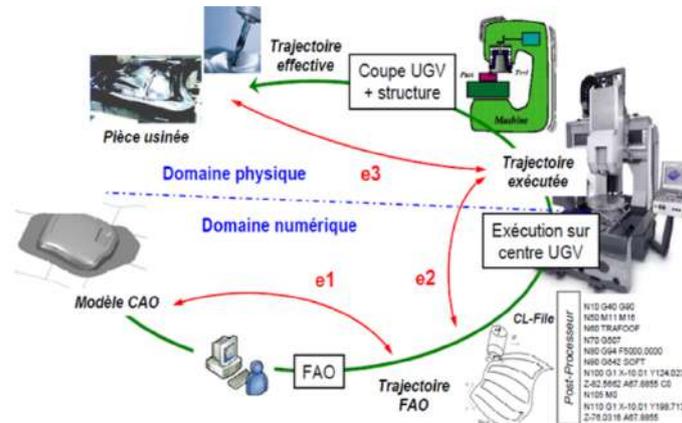


FIGURE 1.30 – Représentation schématique du processus d'élaboration UGV [22].

- **Écart de FAO  $e_1$**  : L'écart  $e_1$  est dû aux erreurs numériques qui résultent des approximations liées au positionnement de l'outil sur la pièce, à la stratégie d'usinage, etc. Après l'étape de FAO, l'enveloppe de la trajectoire de l'outil générée constitue une approximation de la surface nominale.
- **Écart d'exécution  $e_2$**  : L'écart  $e_2$  est un écart d'exécution qui résulte de la génération des consignes de position des axes par la commande numérique, ainsi que de la structure de commande et des composants mécaniques de transformation de mouvement. Les erreurs engendrées par la structure d'asservissement sont couplées numériques/physiques, avec des effets d'inertie, de frottement, etc.
- **Écart  $e_3$**  : Les écarts  $e_3$  résultent de la combinaison de différents phénomènes purement physiques, tels que la géométrie non idéale de la structure de la machine, les dynamiques associées à la coupe, les déformations des composants et les conditions de coupe utilisées. Ces écarts peuvent influencer la qualité finale de la pièce usinée.

#### 1.4.2.1 Écart de FAO $e_1$

La première étape de la chaîne numérique pour l'usinage est la génération de trajectoire à partir du modèle CAO. Notamment deux erreurs peuvent apparaître lors de cette étape : les erreurs de flèche et les erreurs de crête. Ces erreurs sont liées aux paramètres d'usinage tels que la passe longitudinale et transversale, qui définissent respectivement la tolérance d'usinage et la hauteur de crête.

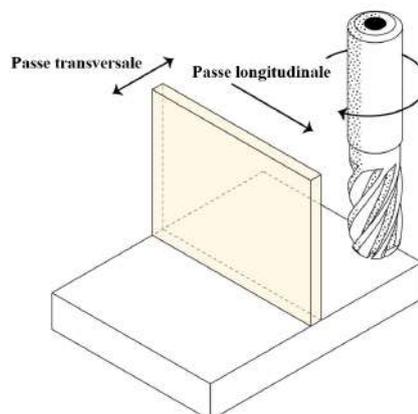


FIGURE 1.31 – Passes longitudinales et transversales.

### A. Erreur de flèche

L'erreur de flèche est générée par les écarts entre la trajectoire d'outil réelle et la trajectoire d'outil nominale. L'outil subit des contraintes mécaniques et thermiques qui entraînent des déformations, notamment de flèche, qui se traduisent par des écarts entre la surface usinée et la surface nominale [3].

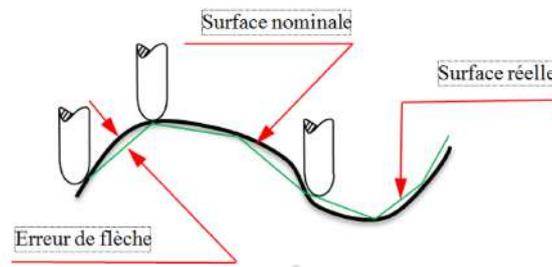


FIGURE 1.32 – Erreur de flèche [3].

### B. Erreur de crête

L'erreur de crête est une source d'erreur qui est étroitement liée à la passe transversale, qui est la distance entre deux passes successives de l'outil. Lors de l'usinage, la passe transversale influence la hauteur des crêtes sur la surface usinée. Si la passe est trop grande, cela peut entraîner des crêtes élevées et des vallées profondes, créant une surface rugueuse et irrégulière. À l'inverse, si la passe est trop petite, cela peut conduire à une surface très lisse, mais le processus d'usinage sera plus lent [3].

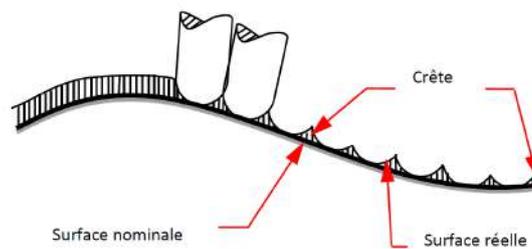


FIGURE 1.33 – Erreur de crête [3].

## 1.5 Quantification des défauts de fabrication

Pour garantir la conformité d'une pièce usinée avec les exigences du concepteur, il est nécessaire de soumettre le produit fini à un contrôle de la topographie de la surface, en prenant en compte les défauts de forme et d'état de surface. Les pièces non conformes sont rejetées, ce qui entraîne une perte de productivité. Afin de remédier à ce problème et de trouver un compromis entre la qualité et le prix, il est essentiel de prédire la topographie de surface. À cet égard, plusieurs approches ont été étudiées dans la littérature [3].

### 1.5.1 Prédiction de la topographie de surface

Diverses classifications ont été présentées dans la littérature, telles que celles proposées par les auteurs de [23]. Ces classifications se basent sur quatre grandes catégories :

- **Approches basées sur la théorie de l'usinage** : Elles consistent à développer des modèles analytiques et/ou des algorithmes informatiques pour représenter la surface usinée.
- **Approches axées sur l'expérimentation et l'analyse des résultats** : Elles examinent les effets de divers facteurs en réalisant des expériences et en analysant les résultats obtenus.
- **Approches utilisant des expériences conçues** : Elles mettent en œuvre des expériences spécifiquement conçues pour étudier la prédiction de la topographie de surface.
- **Approches basées sur l'intelligence artificielle** : Elles utilisent des techniques d'intelligence artificielle pour prédire la topographie de surface.

### 1.5.2 Paramètres de l'état de surface

La norme ISO 25178 [24] définit une série de paramètres permettant de quantifier et d'évaluer les propriétés géométriques des surfaces usinées, il s'agit de l'étendue des paramètres classiques 2D. Les paramètres d'états de surface sont regroupés en quatre catégories principales : Paramètres d'amplitude, Paramètres fonctionnels, Paramètres spatiaux et Paramètres hybrides. Les paramètres d'amplitudes sont définis comme suit :

- **Hauteur moyenne arithmétique de la surface  $S_a$**  : Il s'agit de la moyenne arithmétique de la valeur absolue de l'ordonnée à l'intérieur d'une zone de définition  $A$ .

$$S_a = \frac{1}{A} \iint_A |z(x, y)| \, dx \, dy \quad (1.1)$$

- **Hauteur efficace de la surface  $S_q$**  : Il s'agit de la moyenne quadratique des valeurs des ordonnées à l'intérieur d'une zone de définition  $A$ .

$$S_q = \sqrt{\frac{1}{A} \iint_A z^2(x, y) \, dx \, dy} \quad (1.2)$$

- **Hauteur maximale de pic de la surface  $S_p$**  : Il s'agit de la plus grande valeur de hauteur de pic à l'intérieur d'une zone de définition  $A$ .

$$S_p = \max_A(z(x, y)) \quad (1.3)$$

- **Hauteur maximale de fosse de la surface  $S_v$**  : Il s'agit de la plus petite valeur de hauteur de fosse à l'intérieur d'une zone de définition  $A$ .

$$S_v = \min_A(z(x, y)) \quad (1.4)$$

- **Hauteur maximale de la surface  $S_z$**  : Il s'agit de la somme de la valeur de la hauteur maximale de pic et de la valeur de la hauteur maximale de fosse à l'intérieur d'une zone de définition  $A$ .

$$S_z = S_p + S_v \quad (1.5)$$

## 1.6 Conclusion

Ce chapitre a introduit les concepts fondamentaux nécessaires dans le processus de production des pièces de formes libres en usinage multi-axes. Ensuite, nous nous sommes concentrés sur les méthodes de simulation virtuelle d'usinage multi-axes, en mettant l'accent sur les différentes catégories de simulation ainsi que les techniques d'approximation des modèles utilisés. Le prochain chapitre se concentrera sur les outils mathématiques utilisés pour le partitionnement de données permettant de résoudre une partie des problèmes liés à la simulation d'usinage des pièces de formes complexes.

## Chapitre 2

# Outils mathématiques

## 2.1 Introduction

Le partitionnement intelligent des pièces de forme libre est une étape important dans la simulation d'usinage, il implique de diviser les pièces en partitions plus petites facilement accessibles par la machine.

Ce chapitre traite des détails le processus de partitionnement intelligent, en commençant par un aperçu général sur l'intelligence artificielle et l'apprentissage automatique (Machine Learning). Ensuite, les différents types de partitionnement seront examinés, ainsi que les critères d'évaluation de la qualité d'un partitionnement. Par ailleurs, les problèmes liés à l'initialisation des centroïdes K-Means sont discutés, et les possibles solutions et variantes pour contourner ce problème.

## 2.2 Intelligence artificielle et Apprentissage automatique

Selon [25], l'intelligence artificielle se définit comme étant "Un Ensemble de théories et de techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine."

L'objectif principal de la discipline de l'intelligence artificielle est de recréer et reproduire l'intelligence humaine afin d'automatiser l'exécution de certaines tâches quotidiennes avec une précision suffisante. Cela est réalisé en combinant différentes disciplines existantes telles que la logique mathématique, les statistiques, les probabilités, la neurobiologie computationnelle et l'informatique.

### 2.2.0.1 Catégories d'intelligence Artificielle

L'intelligence artificielle est un domaine vaste qui englobe une large gamme de techniques, d'algorithmes et de technologies. À un niveau élevé, l'IA peut être largement classée en trois types principaux, basés sur leurs capacités et leurs applications potentielles [26], [27] :

- **Intelligence artificielle étroite** : Artificial Narrow Intelligence, elle est considérée comme une IA "faible", tandis que les deux autres types sont classés comme des IA "fortes". Elle est définie par sa capacité à accomplir une tâche très spécifique, comme gagner une partie d'échecs ou identifier une personne spécifique dans une série de photos.
- **Intelligence artificielle générale** : Artificial General Intelligence, elle est définie par sa capacité à être comparée à celle des humains. L'IA générale serait capable de performer sur un niveau équivalent à un humain.
- **Intelligence artificielle super** : Intelligence artificielle super, elle est connue sous le nom de "superintelligence", dépasse l'intelligence et les capacités humaines.

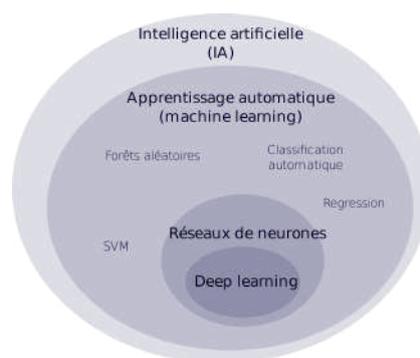


FIGURE 2.1 – Apprentissage automatique et Intelligence artificielle.

## 2.2.1 Apprentissage automatique

L'apprentissage automatique se définit selon [28] par "Le Mode d'apprentissage par lequel un agent évalue et améliore ses performances et son efficacité sans que son programme soit modifié, en acquérant de nouvelles connaissances et aptitudes à partir de données et/ou en réorganisant celles qu'il possède déjà."

Il s'agit d'un champ d'étude de l'intelligence artificielle se fondant sur des approches mathématiques et statistiques, pour donner aux ordinateurs la capacité d'apprendre à partir de données fournies. Les algorithmes d'apprentissage se catégorisent selon le mode d'apprentissage qu'ils emploient :

- Apprentissage supervisé.
- Apprentissage non supervisé.
- Apprentissage semi-supervisé.

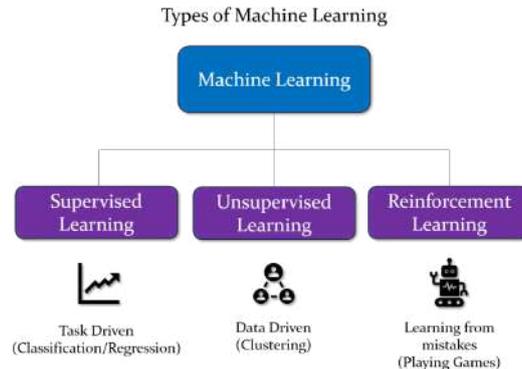


FIGURE 2.2 – Catégories de l'apprentissage automatique.

### 2.2.1.1 Apprentissage supervisé

Lorsque les classes sont prédéterminées et les exemples connus, le système apprend à classer selon un modèle de classification ou de classement. Les problèmes d'apprentissage supervisé peuvent être subdivisés en deux [29] :

- **Problème de classification** : Il s'agit de la prédiction d'une sortie discrète.
- **Régression** : Il s'agit de la prédiction de résultats pour sortie continue.

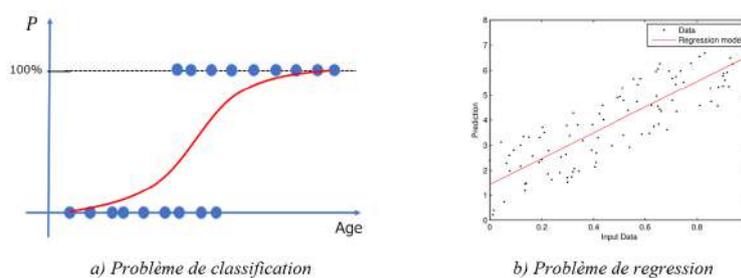


FIGURE 2.3 – Problèmes de l'apprentissage supervisé.

### 2.2.1.2 Apprentissage non-supervisé

Lorsque le système ou l'opérateur ne dispose que d'exemples non étiquetés, et que leur nature n'a pas été prédéterminée, il s'agit d'un apprentissage non supervisé (Clustering). Aucun expert n'est requis. L'algorithme doit découvrir par lui-même la structure des données. Le partitionnement de données (Data Clustering) est un algorithme d'apprentissage non supervisé.

- **Problème de partitionnement** : Il s'agit de la dérivation de la structure des données en les regroupant en plusieurs sous-groupes.

## 2.3 Partitionnement de données

Le partitionnement de données est une méthode en analyse des données. Elle vise à diviser un ensemble de données en différents groupes homogènes, en ce sens que les données de chaque sous-ensemble partagent des caractéristiques communes [30], qui correspondent le plus souvent à des critères de proximité (similarité informatique) définie en introduisant des mesures et classes de distance entre objets.

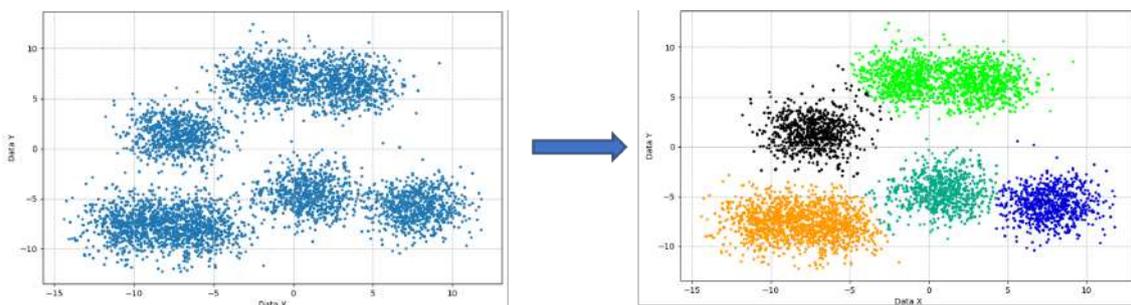


FIGURE 2.4 – Exemple de partitionnement non-supervisé.

### 2.3.1 Types de partitionnement

Il existe deux principaux types de méthode dans le partitionnement de données : La méthode hiérarchique ainsi que la méthode de partition des données [31].

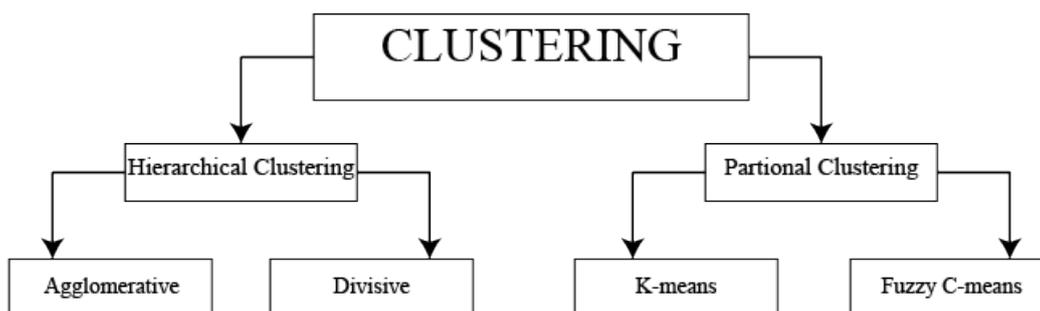


FIGURE 2.5 – Types de partitionnement non-supervisé.

#### 2.3.1.1 Méthode Hiérarchique

Ce type de partitionnement utilise une structure arborescente, avec une relation parent/enfant de manière hiérarchique organisée divisée en deux sous-catégories :

- **Partitionnement Agglomératif** : Chaque élément est traité comme un groupe individuel, puis progressivement fusionné pour former des groupes plus grands et plus significatifs.
- **Partitionnement Divisif** : l'Ensemble de données est traité comme un grand groupe initial, puis divisé progressivement en groupes plus petits.

#### 2.3.1.2 Méthode de Partition

Ce type de partitionnement classe les données en plusieurs groupes, de façon à ce que les données au sein de chaque groupe ont les mêmes traits. Il est divisé en deux sous-catégories :

- **Partitionnement K-Means** : Il s'agit d'une méthode de partitionnement simple, mais efficace, qui ne permet pas la redondance des objets dans de multiples groupes.
- **Partitionnement Fuzzy C-Means** : Similaire à la méthode des K-Means, la différence réside dans le fait que cette méthode permet la multiple appartenance.

### 2.3.2 Critère de distance

Le critère de distance permet de déterminer le niveau de similitude entre les points. Il représente le critère principal pour déterminer la qualité d'un partitionnement. Plusieurs méthodes de mesure de distance existent [31] :

- **Mesure euclidienne** : Il s'agit d'une ligne droite ordinaire reliant les deux points dans l'espace euclidien.
- **Mesure de la distance du cosinus** : Il s'agit de l'angle entre les deux vecteurs formés en joignant le point d'origine.
- **Mesure euclidienne au carré** : Il s'agit d'une mesure de distance identique à la distance euclidienne, mais qui ne nécessite pas l'extraction de la racine carrée à la fin du calcul.
- **Mesure de Manhattan** : Il s'agit de la somme simple des composantes horizontales et verticales, ou encore à la distance entre deux points mesurée le long d'axes perpendiculaires.

Le tableau suivant résume les formules utilisées pour le calcul du critère de distance, avec un exemple de calcul de distance entre deux vecteurs  $\vec{p}$  et  $\vec{q}$  définis par :

$$\vec{p} = \begin{pmatrix} 2.0 \\ 3.0 \\ 5.0 \end{pmatrix}$$

$$\vec{q} = \begin{pmatrix} 4.0 \\ 1.0 \\ 2.0 \end{pmatrix}$$

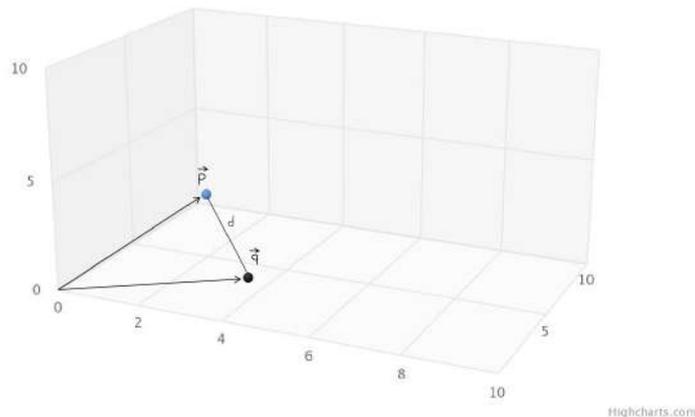


FIGURE 2.6 – Représentation des vecteurs  $\vec{p}$  et  $\vec{q}$

Methode	Formule	Exemple
Mesure euclidienne	$d = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$	d = 4.12
Mesure de la distance du cosinus	$d = \frac{\vec{p} \cdot \vec{q}}{  \vec{p}   \cdot   \vec{q}  }$	d = 0.74
Mesure euclidienne au carré	$d = \sum_{i=1}^n (q_i - p_i)^2$	d = 17.00
Mesure de Manhattan	$d = \sum_{i=1}^n  q_i - p_i $	d = 7.00

La distance entre deux points décrit la similarité entre eux, et par conséquent plus la distance  $d \rightarrow 0$ , similaire seront les deux points.

### 2.3.3 Qualité de partitionnement

Le choix de la mesure pour quantifier le degré de variation au sein d'un groupe est une considération importante lors de l'analyse de partitionnement. Différentes mesures conviennent à différents types de données et d'algorithmes de partitionnement, et la sélection d'une mesure appropriée peut avoir un impact significatif sur la qualité du résultat de partitionnement. La plupart des stratégies impliquent d'exécuter l'algorithme avec différents nombres de groupes puis de trouver la meilleure combinaison en utilisant des critères qui décrivent la qualité du partitionnement [32] :

- Critère du Coude;
- Critère de la silhouette;
- Critère du chevauchement.

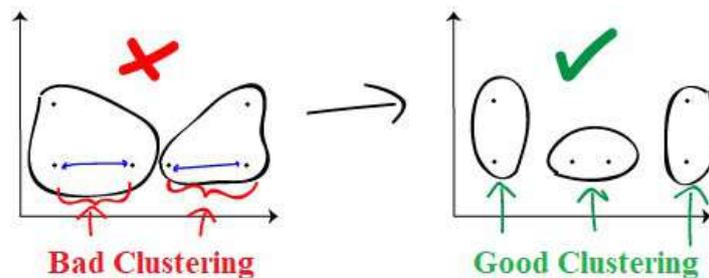


FIGURE 2.7 – Qualité de partitionnement.

#### 2.3.3.1 Critère du Coude

Le Critère du coude consiste à trouver un point appelé **point de coude** généralement utilisé pour le partitionnement de données 2D. La qualité du partitionnement s'améliore rapidement lors de la modification du nombre de groupes, mais finira par converger. Le point où l'amélioration relative n'est plus pertinente est le point de coude, il s'agit plus d'un critère visuel [32].

La détermination analytique du point de coude, se fait par vérification de la distance moyenne entre groupes via la moyenne des sommes de carrés intra-cluster mesure la cohésion des groupes.

$$AWCSS = \frac{1}{N} \sum_{i=1}^K \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2$$

- $N$  : Nombre de points total.
- $\bar{x}_i$  : Centroid du  $i$  groupe.
- $x_{ij}$  :  $j$  point appartenant au groupe  $i$ .
- $K$  : Nombre de groupes.
- $n_i$  : Nombre de points du groupe  $i$ .

Cette mesure évalue à quel point les données au sein d'un même groupe sont similaires. Elle est souvent utilisée pour évaluer la qualité de la partition obtenue par un algorithme de partitionnement. Une faible somme de carrés intra-cluster indique que les données à l'intérieur du même groupe sont très similaires, tandis qu'une somme de carrés intra-cluster élevée indique que les données à l'intérieur du groupe sont très différentes, ce qui peut suggérer que le groupe ne devrait pas exister ou qu'il est mal défini [32].

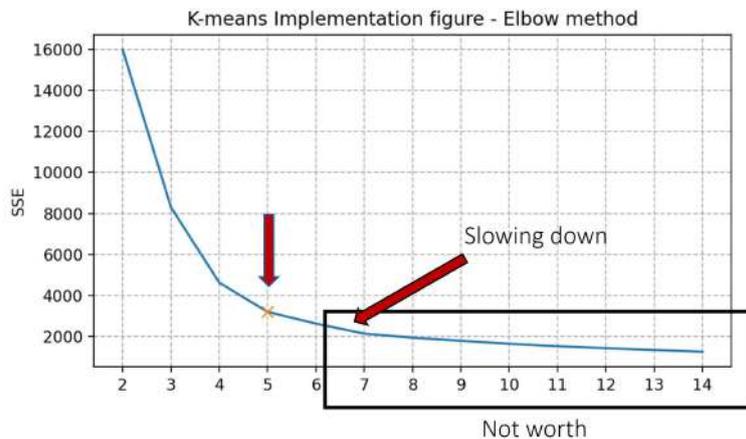
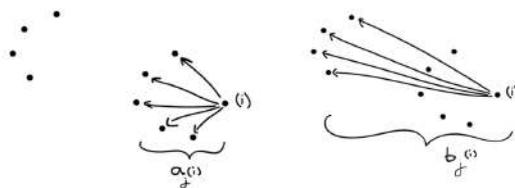


FIGURE 2.8 – Représentation du critère du coude.

### 2.3.3.2 Critère de la silhouette

Le critère de la silhouette calcule le coefficient de silhouette  $s(i)$  de chaque point, ce critère mesure la similarité d'un point appartenant à son propre groupe (cohésion) par rapport aux autres groupes (séparation). La valeur du score de silhouette varie de  $[-1 : +1]$ . Le calcul du coefficient de silhouette  $s$  s'effectue en suivant les trois étapes suivantes [33] :

1. **Calcul du coefficient  $a(i, j)$**  : Il représente la distance moyenne du point  $i$ , appartenant au groupe  $j$ , avec tous les autres points se retrouvant dans le groupe  $j$  ;
2. **Calcul du coefficient  $b(i, j)$**  : Il représente la distance moyenne du point  $i$ , appartenant au groupe  $j$ , avec tous les autres points se retrouvant dans le groupe le plus proche du groupe  $j$  ;

FIGURE 2.9 – Représentation des coefficients  $a_{(i,j)}$  et  $b_{(i,j)}$ .

3. **Calcul du coefficient  $s(i, j)$**  : Il est calculé par la relation suivante :

$$s(i, j) = \frac{b(i, j) - a(i, j)}{\max\{b(i, j), a(i, j)\}}$$

4. **Calcul de la moyenne de tous les  $s(i, j)$**  : ou  $m_j$  représente le nombre de points dans le groupe  $j$ , et  $n$  représente le nombre total de groupes.

$$s = \frac{1}{n} \sum_{j=1}^n \frac{1}{m_j} \sum_{i=1}^{m_j} s(i, j)$$

Dans l'analyse du coefficient de silhouette, trois cas sont à distinguer [33] :

- **Le coefficient final  $s \rightarrow +1$**  : De façon générale l'ensemble des groupes sont éloignés de leurs groupes voisins, il s'agit du cas le plus optimal.
- **Le coefficient final  $s \rightarrow 0$**  : Les groupes sont relativement proches les uns des autres, selon le cas, cela peut être acceptable ou non.
- **Le coefficient final  $s \rightarrow -1$**  : Les points ont été affecté au mauvais groupe.

La valeur pour laquelle le coefficient  $s$  est maximal représente la meilleure configuration de partitionnement.

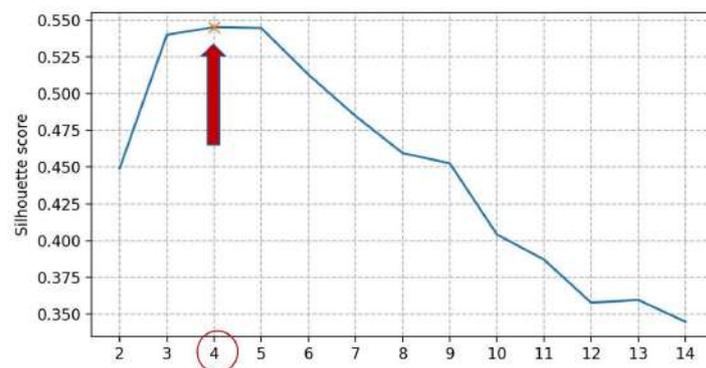


FIGURE 2.10 – Représentation du critère de la silhouette.

### 2.3.3.3 Critère de chevauchement

Le critère de chevauchement consiste à estimer le taux de chevauchement entre les différentes enveloppes des groupes de partitionnement, afin de quantifier l'éloignement des groupes entre eux. De façon générale, un taux de chevauchement réduit se traduit par une meilleure qualité de partitionnement. Le chevauchement total entre groupes est obtenu par les étapes suivantes :

#### 1. Détermination de l'enveloppe d'un ensemble de points :

L'enveloppe d'un point de vue géométrique, est le plus petit cadre englobant un ensemble de points  $S$  en  $N$  dimensions, qui est le boîtier avec la plus petite mesure (surface, volume ou hyper-volume dans les dimensions supérieures) dans lequel se trouvent tous les points.

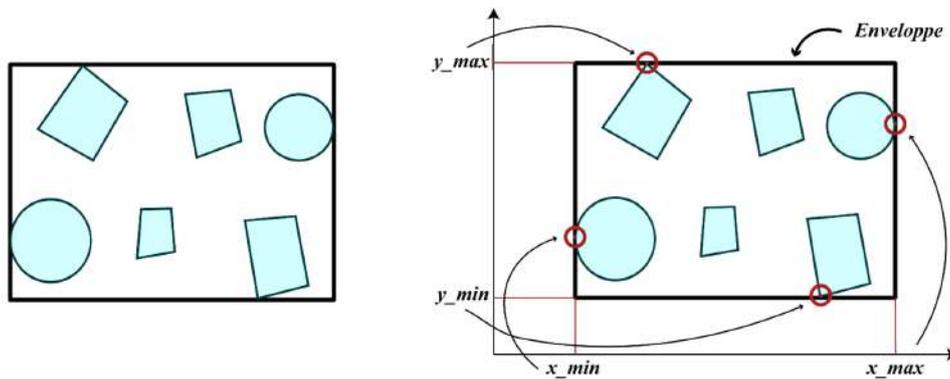


FIGURE 2.11 – Détermination de l'enveloppe d'objets.

Il suffit uniquement deux points  $p_1, p_2$  (de coordonnées  $=\{x_1, x_2, \dots, x_N\}$ ) pour tracer une enveloppe. Pour tracer une enveloppe qui inclut l'intégralité des points, il suffit de trouver les deux points minimum et maximum  $p_{min}(x_{min}, y_{min}, \dots), p_{max}(x_{max}, y_{max}, \dots)$  de telle sorte que :

$$\begin{aligned} x_{min} &= \min(x_{p_1}, x_{p_2}, \dots, x_{p_N}) \\ y_{min} &= \min(y_{p_1}, y_{p_2}, \dots, y_{p_N}) \end{aligned}$$

$$\begin{aligned} x_{max} &= \max(x_{p_1}, x_{p_2}, \dots, x_{p_N}) \\ y_{max} &= \max(y_{p_1}, y_{p_2}, \dots, y_{p_N}) \end{aligned}$$

## 2. Vérification de la présence du chevauchement entre deux enveloppes :

Le principe de cette méthode consiste à évaluer le centre de chaque enveloppe et de vérifier la distance entre le vecteur résultant entre les deux centres, et les dimensions des deux enveloppes.

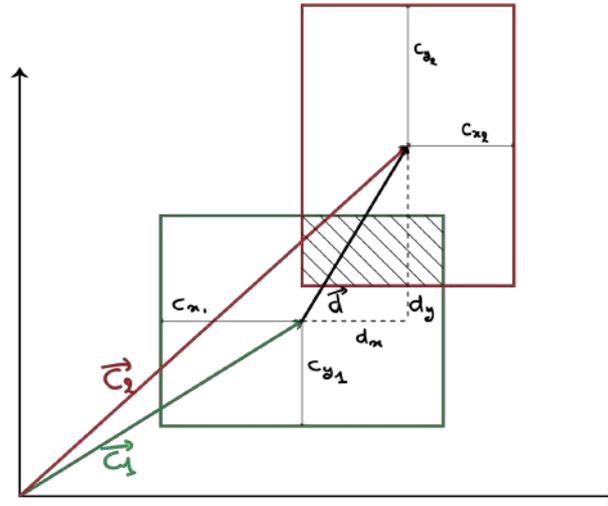


FIGURE 2.12 – Chevauchement entre deux enveloppes.

Le rectangle vert désigne la première enveloppe, et le deuxième désigne l'enveloppe rouge. Avec :

- $\vec{C}_1, \vec{C}_2$  : Respectivement le vecteur reliant le centre à l'origine du repère.
- $\vec{C}_2$  : Vecteur reliant les vecteurs  $\vec{C}_1$  et  $\vec{C}_2$  de tel sorte que  $\vec{C}_2 = \vec{C}_1 + \vec{d}$ .
- $d_x$  et  $d_y$  : Coordonnées du vecteur  $\vec{d}$ .
- $C_{x,i}, C_{y,i}$  : Respectivement les dimensions de l'enveloppe  $i$  divisé par deux.

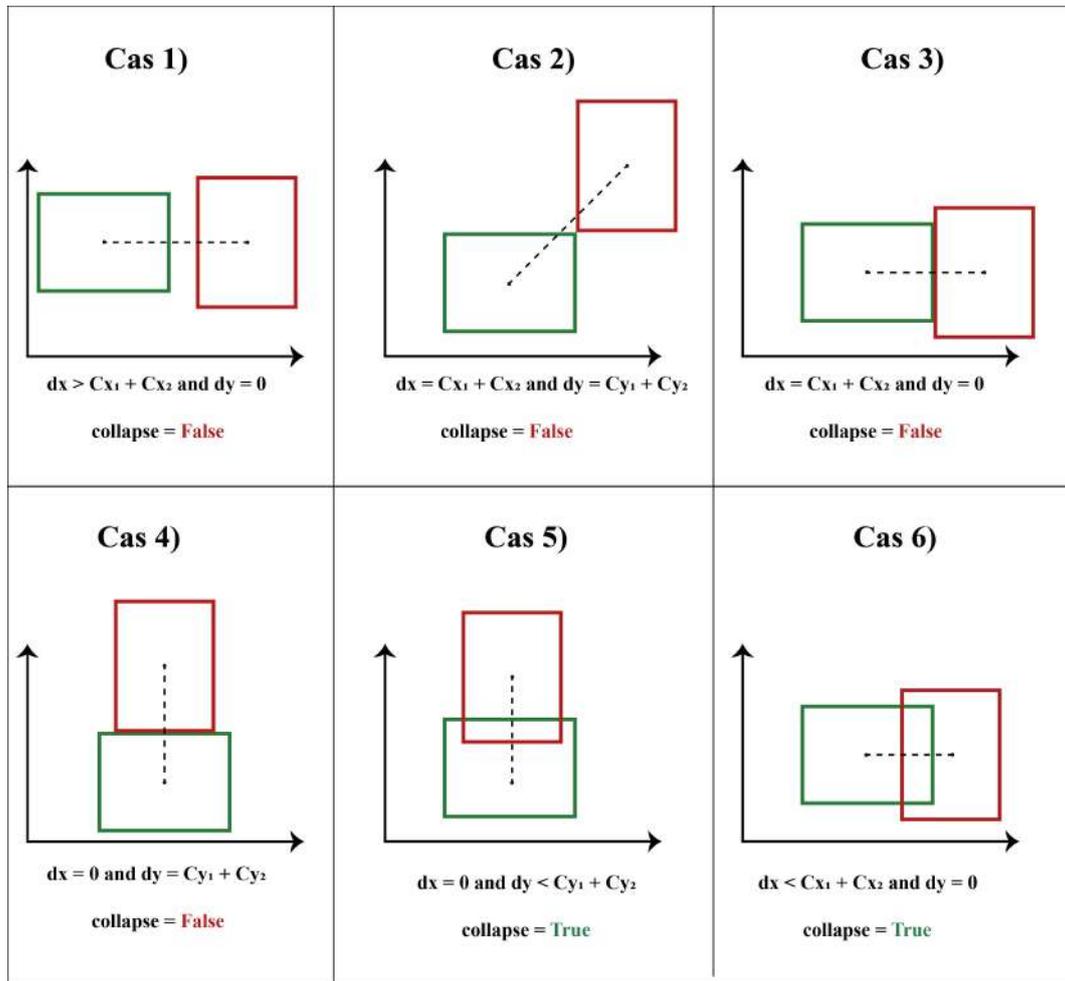


FIGURE 2.13 – Chevauchement entre deux enveloppes pour les cas particuliers.

### 3. Quantification du chevauchement entre deux enveloppes :

La quantification du chevauchement enveloppe-enveloppe est obtenue par multiplication des chevauchements selon les directions  $x$ ,  $y$  et  $z$  :

$$S_i = S_x \times S_y \times S_z$$

$$S_j = C_{j,1} + C_{j,2} - d_j \quad \text{with } j = \{x, y, z\}$$

---

**Algorithm 1** Evaluate overlap between two 3D bounding boxes

---

**Require:**  $c_1 = (cx_1, cy_1, cz_1)$ ,  $c_2 = (cx_2, cy_2, cz_2)$ ,  $d$

**Ensure:** The overlap between the two bounding boxes

```

1: function COLLAPSEVAL( $c_1, c_2, d$ )
2:   if  $dx < cx_1 + cx_2$  and  $dy < cy_1 + cy_2$  and  $dz < cz_1 + cz_2$  then
3:      $collapse\_x \leftarrow cx_1 + cx_2 - dx$ 
4:      $collapse\_y \leftarrow cy_1 + cy_2 - dy$ 
5:      $collapse\_z \leftarrow cz_1 + cz_2 - dz$ 
6:     return  $collapse\_x \times collapse\_y \times collapse\_z$ 
7:   else
8:     return 0
9:   end if
10: end function

```

---

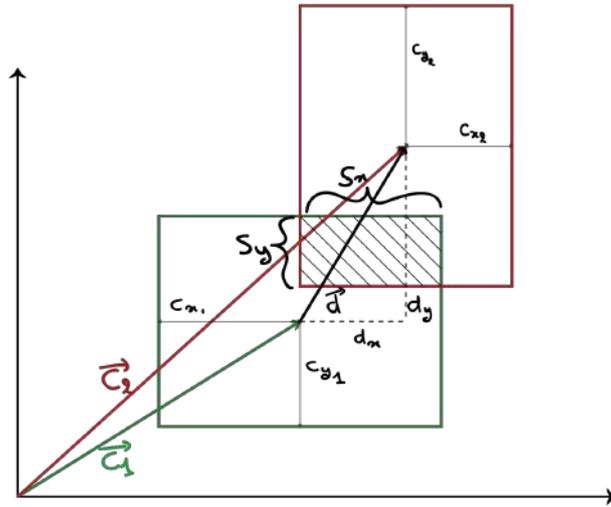


FIGURE 2.14 – Représentation graphique du calcul de chevauchement.

La quantification du chevauchement enveloppe-enveloppe total est obtenue par la sommation des chevauchements de l'ensemble des groupes K entre eux :

$$S = \sum_{i=1}^K \sum_{j=i+1}^K S_{ij}$$

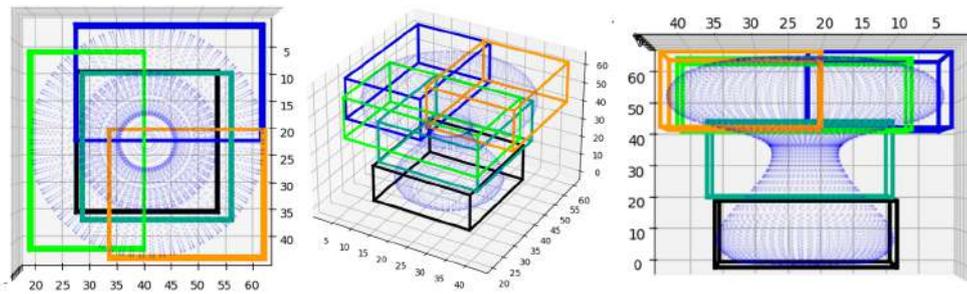


FIGURE 2.15 – Chevauchement d'enveloppes d'un objet .STL.

## 2.4 Classification de données Par K-Means

La méthode des K-Means est une méthode de partitionnement des données en K-groupes, c'est une méthode qui utilise un algorithme d'apprentissage non-supervisé, les données d'entrée n'étant pas étiquetées. Importante dans le domaine de la data-science et les industries, cette méthode permet de regrouper des milliers voire millions de données en un nombre fini de groupes similaire [32].

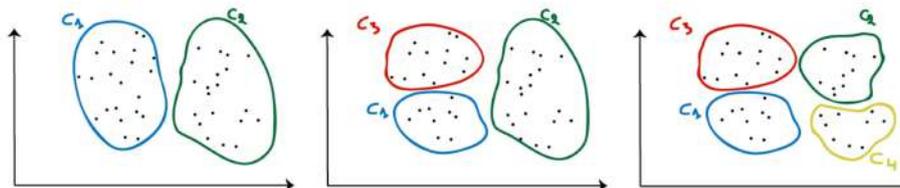


FIGURE 2.16 – Représentation de différents groupes par K-Means

Étant donné un ensemble de points  $(x_1, x_2, \dots, x_n)$  de dimension  $N$ , devant être partitionné en  $K$  ensemble  $S = \{S_1, S_2, \dots, S_k\} (K \in \mathbb{N} \leq N)$ . Il est nécessaire de minimiser la distance entre les points à l'intérieur de chaque partition par rapport au centroïde  $\mu$  de chaque ensemble  $j$  :

$$J(C, \mu) = \sum_{i=1}^K \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \quad (2.1)$$

### 2.4.0.1 Algorithme

L'algorithme de classification de données K-Means est considéré comme l'un des plus simples. Il consiste à choisir un ensemble de centres au hasard, puis à chercher itérativement le partitionnement optimal. Chaque donnée est affectée au centre le plus proche. Une fois toutes les données affectées, la moyenne de chaque groupe est calculée et devient le nouveau représentant du groupe, ce processus continue jusqu'à ce que l'état stationnaire soit atteint, plus aucune donnée ne change de groupe [32], [31].

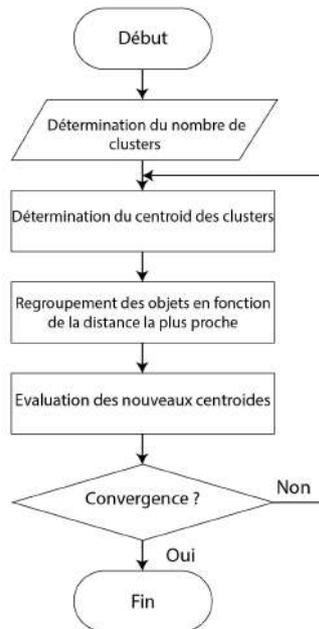


FIGURE 2.17 – Logigramme du partitionnement par K-Means

L'approche utilisée pour exécuter l'algorithme de partitionnement Kmeans peut-être résumé dans les 5 étapes suivantes. La figure 2.18 montre les différentes étapes de l'algorithme et le résultat de chaque étape.

1. Choisir une valeur K aléatoire comme point de départ (généralement  $K = 2$ ) - **Fig (a)** ;
2. Attribuer K points aléatoires comme centroïdes - **Fig (b)** ;
3. Déterminer la distance entre les centroïdes et les points de données, puis affectées - **Fig (c)** ;
4. Réévaluer le centroïde réel des clusters - **Fig (d)** ;
5. Répéter les étapes 3 et 4 jusqu'à la convergence des centroïdes vers une position fixe - **Fig (e, f)**.

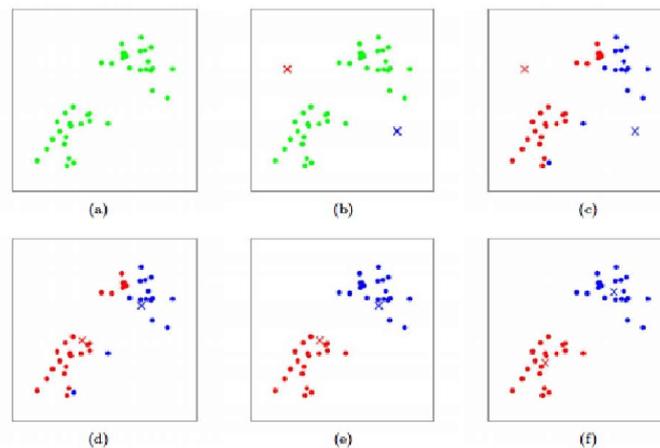


FIGURE 2.18 – Étapes de l'algorithme des K-Means

**Algorithm 2** K-means algorithm**Require:**  $X$  : dataset,  $k$  : number of clusters,  $max\_iter$  : maximum number of iterations**Ensure:**  $C$  : cluster centers,  $L$  : cluster labels

```

1: function K-MEANS( $X, k, max\_iter$ )
2:   Initialize  $C$  with  $k$  random data points from  $X$ 
3:    $L \leftarrow$  empty array of length  $|X|$ 
4:   for  $i \leftarrow 1$  to  $max\_iter$  do
5:     for  $j \leftarrow 1$  to  $|X|$  do
6:        $l_j \leftarrow$  index of the nearest center in  $C$ 
7:     end for
8:     for  $c \leftarrow 1$  to  $k$  do
9:        $C_c \leftarrow$  mean of all data points in cluster  $c$ 
10:    end for
11:  end for
12:  return  $C, L$ 
13: end function

```

### 2.4.1 Classification par Mini Batch K-Means

Bien que l'implémentation de la méthode des K-Means soit simple, elle nécessite que l'utilisateur dispose de ressources informatiques suffisantes, notamment de la RAM, pour stocker les données et les calculs intermédiaires en mémoire. Toutefois, les fichiers .STL générés par les logiciels de CAO peuvent être très volumineux, atteignant parfois des centaines de mégaoctets. Pour des données de cette ampleur, la méthode des K-Means peut être très lent ou échouer complètement si l'utilisateur ne dispose pas de ressources informatiques adéquates [34].

Afin de résoudre les problèmes liés à l'utilisation de K-Means avec de grands ensembles de données, deux solutions sont proposées [34] :

- Les approches de parallélisation exploitent généralement une combinaison de concepts MapReduce pour traiter un grand volume de données dans un environnement informatique distribué, et l'utilisation de processeurs multicœurs.
- Les approches de sous-échantillonnage, telles que l'algorithme Mini Batch K-Means, travaillent sur de petits sous-échantillons de données aléatoires ("mini-batches") qui peuvent tenir en mémoire sur des ordinateurs standards.

#### 2.4.1.1 Principe

L'algorithme de partitionnement Mini-Batch K-Means est une version de l'algorithme K-Means qui peut être utilisée à la place de l'algorithme K-Means pour le partitionnement de grands ensembles de données, pour fichier .STL dépassant les dix mille triangles. Cette méthode fonctionne mieux que l'algorithme K-Means standard lorsqu'elle travaille sur des ensembles de données volumineux, car il ne parcourt pas l'ensemble des données à chaque itération. Elle crée des lots de données aléatoires à stocker en mémoire, puis un lot de données aléatoire est collecté à chaque itération pour mettre à jour les groupes [35].

Le partitionnement Mini-Batch K-Means suit une approche itérative similaire à l'algorithme de Lloyd. Cependant, à chaque itération  $t$ , un nouveau sous-ensemble aléatoire  $M$  de taille  $b$  est utilisé et cela continue jusqu'à convergence. Si nous définissons le nombre de centroïdes comme  $k$  et la taille du mini-Batch comme  $b$  (ce que nous appelons la "taille du lot") [36].

#### 2.4.1.2 Algorithme

Mini Batch K-Means utilise de petits lots d'exemples aléatoires d'une taille fixe pour stocker en mémoire. Chaque itération utilise un nouvel échantillon aléatoire de l'ensemble de données pour mettre à jour les groupes jusqu'à la convergence. Le taux d'apprentissage diminue avec le nombre d'itérations et est l'inverse du nombre d'exemples assignés à un groupe. La convergence est détectée lorsqu'aucun changement dans les groupes ne se produit pendant plusieurs itérations consécutives [36].

**Algorithm 3** Mini-Batch Kmeans

---

**Require:**  $X$  : data set,  $k$  : number of clusters,  $b$  : batch size,  $max\_iter$  : maximum number of iterations**Ensure:**  $C$  : set of cluster centroids

```
1: function K-MEANS( $X, k, max\_iter$ )
2:   Randomly select  $k$  points from  $D$  as initial centroids  $C$ 
3:    $t \leftarrow 0$ 
4:   while  $t < max\_iter$  do
5:      $B \leftarrow$  random subset of  $b$  points from  $X$ 
6:     for  $x \in B$  do
7:        $c_i \leftarrow \operatorname{argmin}_{c_j \in C} |x - c_j|^2$ 
8:        $n_i \leftarrow n_i + 1, m_i \leftarrow m_i + (x - c_i)$ 
9:     end for
10:    for  $c_i \in C$  do
11:       $c_i \leftarrow c_i + \eta \cdot \frac{m_i}{n_i}$ , where  $\eta$  is the learning rate
12:       $n_i \leftarrow 0, m_i \leftarrow 0$ 
13:    end for
14:     $t \leftarrow t + 1$ 
15:  end while
16:  return  $C$ 
17: end function
```

---

## 2.5 Problèmes liés Initialisation des centroides K-Means

La classification par K-Means a pour principale limite sa forte dépendance aux centres initiaux choisis aléatoires, ce qui engendre des solutions différentes (optimum local) pour chaque initialisation, pouvant s'éloigner de la solution optimale (optimum global) dans certains cas. Une approche naïve pour résoudre ce problème consiste à exécuter plusieurs fois l'algorithme de K-means de base avec différents centres initiaux et à conserver la meilleure partition obtenue [37]. Les auteurs [38] proposent une variante, appelée "k-means++" qui choisit les centres au hasard parmi les points de données, mais qui pondère les points de données en fonction de leur distance au carré par rapport au centre le plus proche déjà choisi.

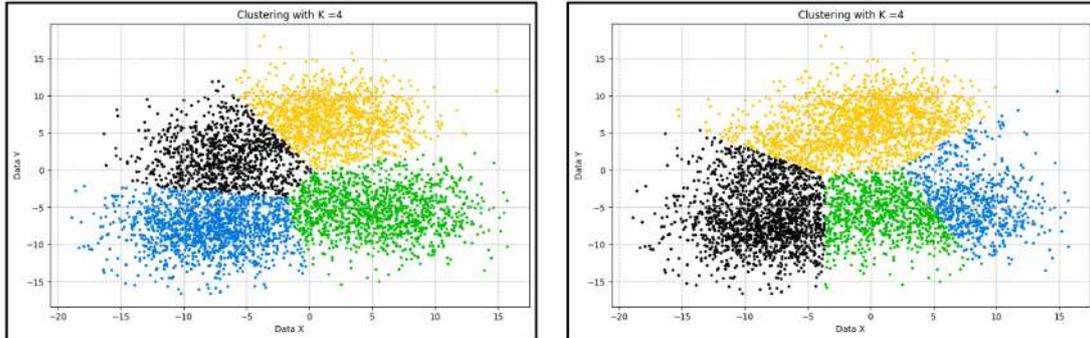


FIGURE 2.19 – Problème d'initialisation aléatoire.

### 2.5.1 Variantes des K-Means

Les variantes des K-Means sont des algorithmes qui améliorent les performances, la précision et réduisent le temps de calcul de l'algorithme de base. Les variantes incluent le K-Means global, le K-Means global rapide ainsi que le K-Means incrémental.

#### 2.5.1.1 Partitionnement par Global K-Means

La méthode Global k-means est une technique pour résoudre le problème d'initialisation du k-means en se basant sur les données pour obtenir une solution optimale globale. Elle utilise un clustering incrémental qui ajoute dynamiquement un nouveau centre suivi de l'application du k-means jusqu'à convergence. Les centres sont choisis un par un en fonction d'une méthode spécifique : le premier centre est le centre de gravité de toutes les données (résultat de l'application du k-means avec  $k=1$ ), et les autres centres sont choisis parmi les données, où chaque donnée est candidate pour devenir un centre. Chaque candidat est testé avec le reste de l'ensemble, et le meilleur candidat est celui qui minimise la fonction objectif [37].

---

#### Algorithm 4 Global K-means algorithm

---

**Require:**  $X$  : dataset,  $k$  : number of clusters

**Ensure:**  $C$  : cluster centers,  $L$  : cluster labels

```

1: function GLOBALKMEANS( $X, k$ )
2:   Randomly select  $k$  points from  $X$  as the initial cluster centers
3:    $L \leftarrow \emptyset^n$ 
4:   repeat
5:     for  $i = 1$  to  $n$  do
6:        $l_i \leftarrow \operatorname{argmin}_j = 1^k d(x_i, c_j)^2$ 
7:     end for
8:     for  $j = 1$  to  $k$  do
9:        $c_j \leftarrow \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i$ 
10:    end for
11:  until convergence
12:   $C \leftarrow c_1, c_2, \dots, c_k$ 
13:  return  $C, L$ 
14: end function

```

---

### 2.5.1.2 Partitionnement par Fast Global K-Means

Il a été constaté que la solution de Global K-Means était lente en raison de la stratégie utilisée pour choisir le nouveau centre [37]. Une technique alternative a été proposée, appelée Fast global K-Means, qui accélère le processus. Cette méthode conserve la même philosophie que la précédente, en considérant que toutes les données peuvent devenir un centre, mais elle évite d'affecter les données aux centres les plus proches et de calculer l'erreur quadratique. Au lieu de cela, le nouveau centre est choisi en fonction d'un taux qui mesure la diminution de l'erreur quadratique en fonction du nombre de centres. Le candidat qui maximise ce taux est choisi comme nouveau centre [37].

---

#### Algorithm 5 Fast Global K-Means Algorithm

---

**Require:** Dataset  $X$ , number of clusters  $k$ , number of iterations  $T$ , threshold  $\epsilon$

**Ensure:** Cluster centroids  $C$

```

1: function GLOBALKMEANS( $X, k, T, \epsilon$ )
2:   Choose  $k$  random points as initial centroids  $C$ 
3:   for  $t = 1$  to  $T$  do
4:     Assign each point to the nearest centroid
5:     Compute the new centroids for each cluster
6:      $C_{new} \leftarrow$  Apply PCA on  $X$  to reduce its dimensionality
7:     if  $\frac{\|C_{new} - C\|}{\|C\|} < \epsilon$  then
8:       break
9:     end if
10:     $C \leftarrow C_{new}$ 
11:   end for
12:   return  $C$ 
13: end function

```

---

### 2.5.1.3 Partitionnement par Incremental K-Means

Les auteurs [37] proposent une méthode de classification incrémentale similaire à celle du Global K-Means, mais avec quelques différences :

- L'utilisation deux points initiaux plutôt qu'un seul.
- La recherche du nouveau centre se limite à la recherche de l'élément le plus mal classé.

---

#### Algorithm 6 Incremental K-Means Algorithm

---

**Require:** Initial centroids  $C_0$ , data points  $X$ , number of clusters  $k$ , learning rate  $\eta$

**Ensure:** Final centroids  $C$

```

1: function INCREMENTALKMEANS( $C_0, X, k, \eta$ )
2:    $C \leftarrow C_0$ 
3:   for  $x \in X$  do
4:      $c_i \leftarrow$  the closest centroid to  $x$  in  $C$ 
5:      $c_i \leftarrow c_i + \eta(x - c_i)$ 
6:   end for
7:   return  $C$ 
8: end function

```

---

## 2.5.2 Algorithme k-means++

L'algorithme k-means++ propose une méthode spécifique pour sélectionner les centres qui vise à améliorer la qualité du regroupement. Il commence par définir  $D(x)$  comme la distance la plus courte entre un point de données et le centre le plus proche qui a déjà été choisi. L'algorithme se déroule comme suit [38] :

1. Choisir un centre  $c_1$  uniformément au hasard dans l'ensemble de données  $X$ .
2. Pour chaque centre suivant  $c_i$ , sélectionner un nouveau point  $x \in X$  avec une probabilité proportionnelle à son poids  $D_2$ , qui est calculé comme  $D(x)^2 / \sum D(x)^2$  sur tous les points de  $X$ .
3. Répéter l'étape 2 jusqu'à ce que  $k$  centres aient été sélectionnés.
4. Procéder avec l'algorithme standard de K-Means pour attribuer les points de données restants à leurs centres les plus proches et mettre à jour itérativement les centres jusqu'à ce que la convergence soit atteinte.

---

**Algorithm 7** K-means++ Algorithm
 

---

**Require:**  $n$  points  $x_1, x_2, \dots, x_n \in R^d$ ; the number  $k$  of clusters to form

**Ensure:** a set of  $k$  cluster centers  $C = c_1, c_2, \dots, c_k$

- 1: **function** KMEANSPP( $X, k$ )
  - 2:     Choose an initial center  $c_1$  uniformly at random from the set of data points
  - 3:     **for**  $i = 2$  to  $k$  **do**
  - 4:         Compute  $D(x)$ , the distance between each data point  $x$  and the closest center that has already been chosen
  - 5:         Choose a new center  $c_i$  from the data points with probability proportional to  $D(x)^2$
  - 6:     **end for**
  - 7:     Assign each data point to the nearest center to form clusters **return**  $C$
  - 8: **end function**
- 

L'utilisation du poids  $D_2$  à l'étape 2 garantit que l'algorithme est orienté vers la sélection de points de données qui sont plus éloignés des centres existants, ce qui permet de répartir les centres initiaux et d'éviter un regroupement sous-optimal [38].

## 2.6 Conclusion

Ce chapitre a exploré en détail les techniques de partitionnement de l'intelligence artificielle utilisées pour la classification des données, en présentant les algorithmes, et les critères d'évaluations, ainsi que les solutions vis-à-vis du problème de l'initialisation des centroïdes.

La mise en œuvre d'une application de simulation est une étape importante qui nécessite la création d'une solution logicielle robuste et efficace, capable de gérer de grandes quantités de données et de réaliser des calculs complexes. Ce chapitre a posé les bases pour le développement de cette application en fournissant une compréhension approfondie sur les techniques de partitionnement et des défis associés à leur mise en œuvre.

## Chapitre 3

# Analyse et conception

### 3.1 Introduction

Après avoir étudié en détail les outils mathématiques utilisés pour le partitionnement intelligent, ce chapitre se concentre sur l'analyse et la conception d'une application de simulation d'usinage qui exploite ces outils mathématiques. La démarche chronologique du module logiciel sera présentée en détaillant sur les algorithmes et les procédures utilisés.

### 3.2 Solution proposée

Afin de réaliser l'objectif énoncé précédemment, il est recommandé de suivre un certain nombre d'étapes et de procédures pour garantir la réussite de la mise en œuvre. Dans ce contexte, il est proposé de mettre en place les actions suivantes :

- **La modélisation volumique de pièce de forme libres.**
  1. Classification du modèle STL afin d'optimiser le temps de calcul.
  2. Génération des grilles ayant un pas de Dixel.
  3. Calcul des intersections entre les droit des grilles et le modèle STL afin de généré un modèle volumique TD.
  
- **Simulation d'enlèvement de matière.**
  1. Optimisation du temps de calcul en classifiant le modèle TD.
  2. Calcul des intersections entre l'outil et le modèle TD.
  3. Estimation de la qualité de l'état de surface.

### 3.3 Démarche de l'application

Un module logiciel a été développé pour la simulation d'enlèvement de matière en opération de finition, sous environnement Windows. Les étapes de cette simulation sont organisées selon un organigramme chronologique. La réalisation d'une approche de prédiction de la topographie 3D des surfaces libres usinées par fraisage nécessite des interactions fortes entre de nombreuses activités. Chaque étape de cette approche doit garantir la conformité de la pièce complexe et le respect du cahier des charges. Les activités que couvre le nouveau module logiciel sont les suivantes :

- **Modélisation volumique de pièce de forme libres.**
  - Lecture du modèle TD existant
    1. Acquisition du modèle .TD.
  - Génération du modèle TD.
    1. Acquisition du modèle STL via un logiciel CAO.
    2. Classification du modèle STL par des méthodes de partitionnement.
    3. Création des grilles suivant les trois plans XY, YZ et XZ.
    4. Calcul d'intersections Segments (grilles)-Triangles .
    5. Filtre des segments résultant de l'intersection et création du modèle .TD.
  
- **Simulation d'enlèvement de matière.**
  1. Acquisition du modèle TD.
  2. Ajout de la surépaisseur sur chaque Dixel.
  3. Classification du modèle TD épaissi par des méthodes de partitionnement.
  4. Acquisition du trajet d'outil d'usinage par des logiciels de CFAO.
  5. Simulation d'enlèvement de matière.
  6. Topographie de surface de la pièce volumétrique usinée.

L'architecture générale de l'application permettant l'acquisition du modèle TD est présentée dans les deux figures 3.1 et 3.2, la première représente une architecture simple et directe qui consiste à simplement lire un modèle TD déjà existant dans une base de données, tandis que la deuxième architecture décrit le processus de génération de ce dernier.

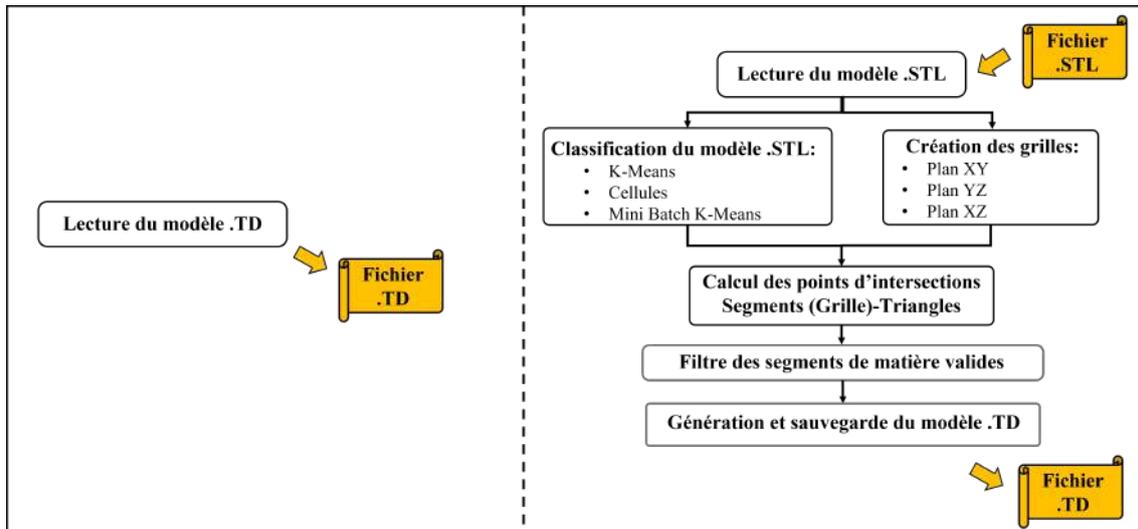


FIGURE 3.1 – Architecture d'acquisition du modèle .TD.

L'architecture générale de l'application de simulation d'enlèvement de matière est présentée dans la figure 3.2.

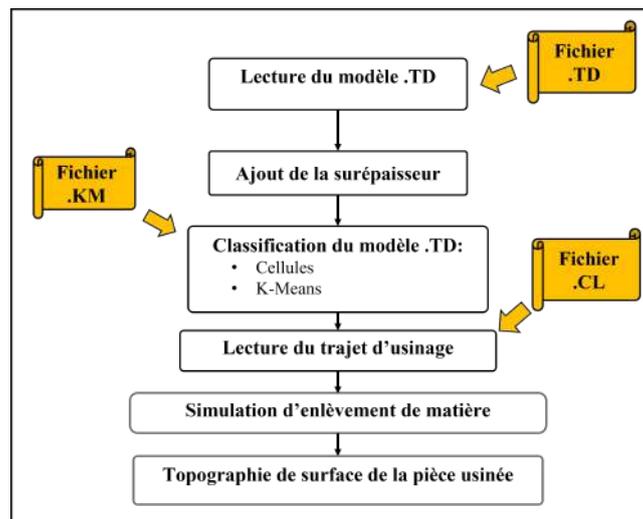


FIGURE 3.2 – Architecture de simulation d'enlèvement de matière.

### 3.3.1 Acquisition du modèle triple Dixel

Si le modèle TD issu du modèle STL a déjà été généré au préalable, l'acquisition du modèle TD se fait directement par l'ouverture et lecture de ce dernier dans l'application de simulation, dans le cas contraire il est nécessaire de le générer et de le sauvegarder pour l'utiliser dans l'application de simulation.

#### 3.3.1.1 Lecture du modèle triple Dixel

L'acquisition du modèle TD se fait par la lecture d'un fichier d'extension .TD type ASCII ou binaire qui contient toutes les informations nécessaires :

- Pas du modèle suivant les trois axes X, Y et Z.
- Les dimensions et coordonnées de l'enveloppe du modèle.
- Les coordonnées des Dixels ainsi que le type de segment.

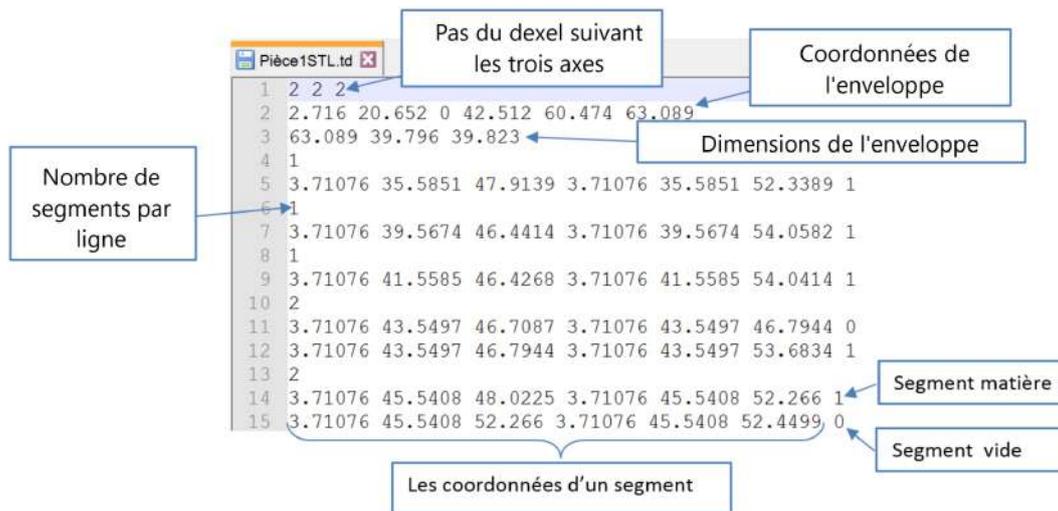


FIGURE 3.3 – Fichier .TD type ASCII.

#### 3.3.1.2 Génération du modèle triple dixel

Dans le cas où le modèle TD n'existe pas, il faudrait passer par l'étape de la génération de ce dernier. Ceci via l'application de "Génération modèle Triple Dixel".

#### A. Acquisition du modèle .STL

L'acquisition du modèle STL se fait par la lecture d'un fichier d'extension .STL ASCII ou binaire, issu de logiciels de modélisation, CAO, etc. Chaque triangle est défini par trois sommets reliés entre eux, ainsi qu'une normale unitaire.

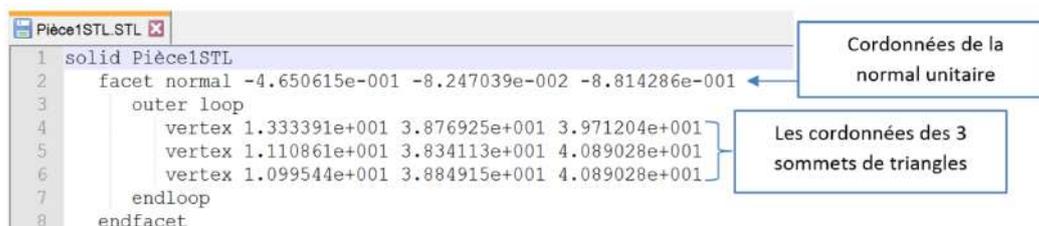


FIGURE 3.4 – Fichier .STL type ASCII.

#### B. Classification du modèle STL

La classification du modèle .STL a pour but la réduction considérable du temps global d'exécution et de génération, le regroupement des différents triangles en zones "Cluster" (Groupe), diminue fortement

le nombre d'opérations que doit effectuer le système. La classification se fait principalement par deux méthodes de partitionnement, Cellules et K-Means.

— Cellules :

Cette méthode est couramment utilisée dans les logiciels graphiques et consiste à diviser l'espace en sous-espaces de même taille appelés cellules. Chaque cellule est définie par une longueur, une largeur et une hauteur, et caractérisée par ses limites, son centre ainsi que l'axe représentant sa direction. La classification du modèle .STL via les Cellules se fait en deux étapes [3] :

1. La création des cellules se fait à partir de l'enveloppe du modèle .STL à discrétiser, en spécifiant le nombre de cellules selon les axes X, Y et Z. Les dimensions et les coordonnées des points extrêmes de chaque cellule sont ensuite calculées.
2. L'étape suivante consiste à affecter les triangles aux cellules. Les triangles qui sont entièrement contenus dans une cellule sont facilement déterminés, tandis que les triangles qui se chevauchent partiellement peuvent être affectés en vérifiant si l'un de leurs sommets appartient à la cellule.

---

#### Algorithm 8 Cell Partitioning of STL Files

---

**Require:** Array of cell bounding boxes  $C$  of size  $n$ , array of triangles  $T$  of size  $m$

**Ensure:** Array of affected triangles  $A$

```

1: Initialize empty array  $A$ 
2: for  $i = 1$  to  $m$  do
3:   Get bounding box of triangle  $t_i$ 
4:   for  $j = 1$  to  $n$  do
5:     if  $t_i$  is inside bounding box of cell  $C_j$  then
6:       Assign  $t_i$  to cell  $C_j$ 
7:       Add  $t_i$  to  $A$ 
8:       break
9:     end if
10:  end for
11: end for
12: return  $A$ 

```

---

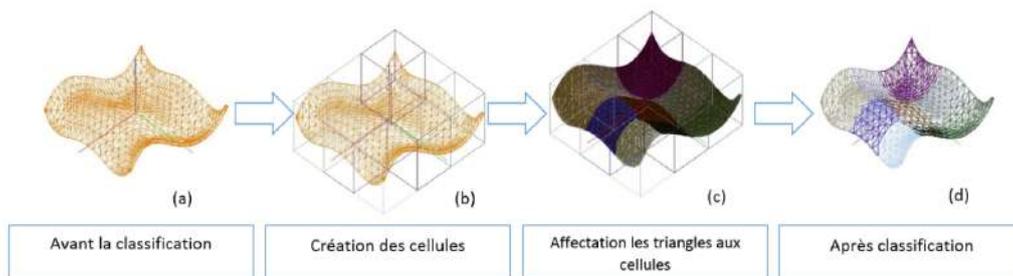


FIGURE 3.5 – Processus du partitionnement par cellules d'un modèle STL [3].

— Kmeans :

Le partitionnement par K-Means des pièces de forme libre regroupe les triangles en zones appelées "Clusters". Chaque Cluster est constitué d'une liste de triangles et d'un point spécifique appelé "Centroïde", qui correspond au centre de gravité des triangles. Les étapes du processus sont les suivantes [3] :

1. Déterminer le nombre de clusters (K) et le nombre d'itérations (i) à utiliser dans le processus.
2. Calculer le centre de gravité pour chaque triangle du modèle STL.
3. Initialiser les clusters en choisissant arbitrairement les coordonnées des K centroïdes.
4. Affecter chaque triangle à son cluster respectif en fonction de la distance minimale entre le centroïde et le centre de gravité du triangle.
5. Mettre à jour les coordonnées du centroïde de chaque cluster en fonction des triangles qu'il contient.
6. Vérifier la stabilité du processus en calculant les écarts entre les nouveaux et anciens centroïdes de chaque cluster.
7. Arrêter le processus si la stabilité est atteinte ou si le nombre d'itérations a été atteint.

---

#### Algorithm 9 KMeans Partitioning of STL Files

---

**Require:** Array of kmeans bounding boxes  $C$  of size  $n$ , array of triangles  $T$  of size  $m$

**Ensure:** Array of affected triangles  $A$

```

1: Initialize empty array  $A$ 
2: for  $i = 1$  to  $m$  do
3:   Get bounding box of triangle  $t_i$ 
4:   for  $j = 1$  to  $n$  do
5:     if  $t_i$  is inside bounding box of cluster  $C_j$  then
6:       Assign  $t_i$  to cluster  $C_j$ 
7:       Add  $t_i$  to  $A$ 
8:     break
9:   end if
10:  end for
11: end for
12: return  $A$ 

```

---

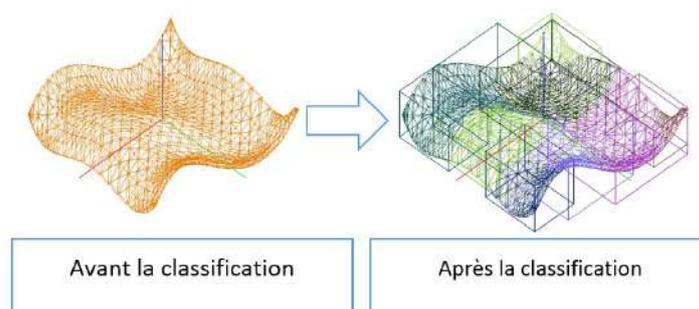


FIGURE 3.6 – Processus du partitionnement par K-Means d'un modèle STL [3].

### C. Création des grilles

La création des grilles pour la génération du modèle TD, se fait par l'intermédiaire de l'enveloppe du modèle .STL ainsi que le pas Dixel.

— Détermination de l'enveloppe :

L'enveloppe est obtenue par la détermination des coordonnées min et max de l'ensemble des sommets suivant les trois axes. Soit l'ensemble de sommets  $\{P\} = \{P_1, P_2, \dots, P_n\}$ , où  $n$  représente le nombre de sommets unique d'un modèle STL. Les coordonnées  $P_{min}$ ,  $P_{max}$  qui décrit l'enveloppe du modèle sont :

$$P_{i,min} = \min(P_{i,1}, P_{i,2}, \dots, P_{i,n}), \quad i = \{x, y, z\}$$

$$P_{i,max} = \max(P_{i,1}, P_{i,2}, \dots, P_{i,n}), \quad i = \{x, y, z\}$$

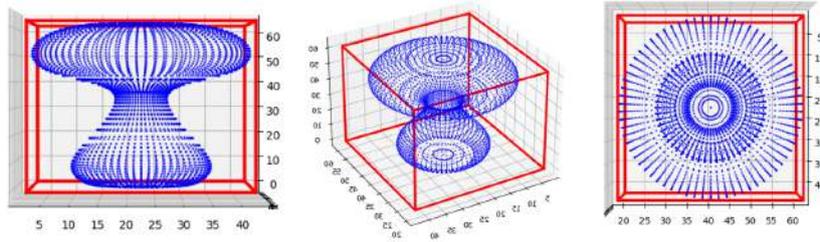


FIGURE 3.7 – Enveloppe d'un modèle STL

Les dimensions de l'enveloppe sont par la suite obtenues par la différence entre les deux points  $P_{min}$  et  $P_{max}$  :

$$Dimension = P_{max} - P_{min}$$

#### — Création des grilles :

Cette étape consiste à créer des grilles suivant les trois plans XY, YZ et XZ. Ayant un pas de discrétisation suivant les trois axes X, Y et Z ainsi que l'enveloppe du modèle STL, les grilles sont obtenues par subdivision des trois plans. Chaque cellule de grille possède les mêmes caractéristiques qu'une cellule normale, à savoir les limites, un centre et une direction.

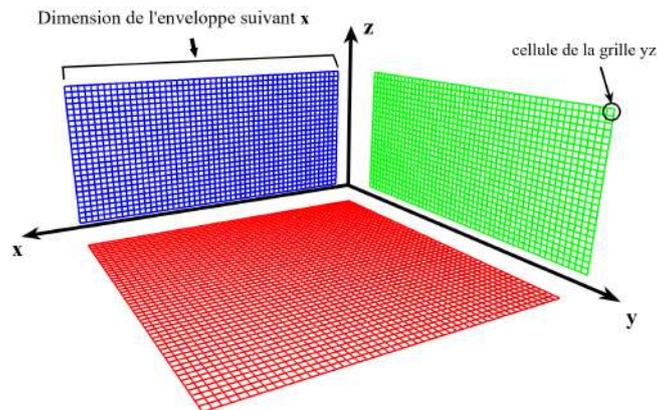


FIGURE 3.8 – Grille suivant les trois plans XY, YZ et XZ.

## D. Calcul d'intersections Segments (grilles)-Triangles

Le calcul de l'intersection entre les segments des grilles et les triangles du modèle STL permet de déterminer les zones où la droite traverse le triangle, ce qui permet de limiter les segments des grilles et d'obtenir un modèle de segment semblable au modèle .STL.

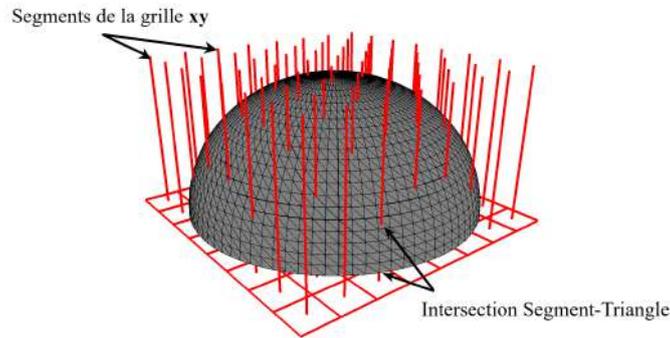


FIGURE 3.9 – Intersection Segments-Triangles

Plusieurs techniques existent pour le calcul de l'intersection Segment-Triangle, actuellement les algorithmes les plus simples et efficaces sont :

- Algorithme de Badouel.
- Algorithme de Möller-Trumbore.
- Algorithme de Segura.

L'algorithme qui a été utilisé dans le module logiciel pour le calcul d'intersection est l'algorithme de Möller-Trumbore, selon [39] il s'agit de la méthode la plus rapide pour la détermination des points d'intersection. Tant dis que l'algorithme de Segura est plus efficace pour déterminer l'existence d'intersection ou non.

### • Algorithme de Möller-Trumbore :

L'algorithme de Möller-Trumbore est une version plus avancée et optimisée de l'algorithme de Badouel. Le triangle est translaté de sorte que son origine coïncide avec l'origine du système de coordonnées. Ensuite, il est mis à l'échelle pour devenir un triangle unitaire suivant  $yz$ , tandis que le rayon est aligné avec l'axe des  $x$ . Cette transformation simplifie les calculs nécessaires pour déterminer si le segment intersecte le triangle, rendant l'algorithme plus efficace. L'annexe [A] détaille sur la détermination des points d'intersections.

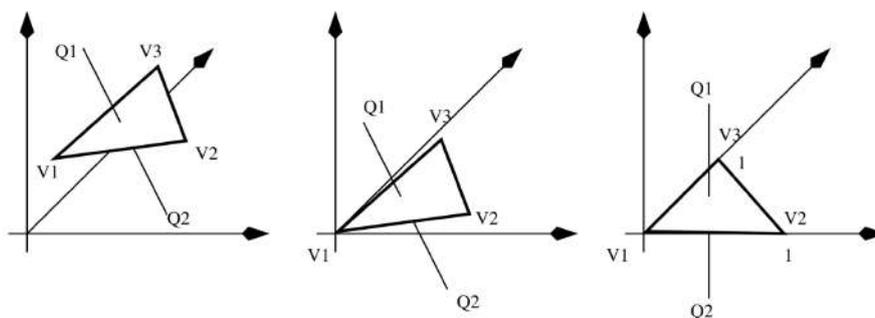


FIGURE 3.10 – Translation et transformation du triangle [39] .

Les calculs sont accélérés, car il n'est nécessaire d'effectuer qu'une seule division, deux produits vectoriels et quatre produits scalaires dans le pire des cas. L'utilisation de cette méthode permet une optimisation des calculs nécessaires pour déterminer l'intersection entre un segment et un triangle. Contrairement à

d'autres méthodes, elle nécessite moins d'opérations mathématiques et ne nécessite pas le stockage d'informations supplémentaires. Cela permet de gagner du temps et de l'espace mémoire dans l'implémentation de l'algorithme [39].

---

**Algorithm 10** Trumbore-Moller Algorithm
 

---

```

1: function TRUMBOREINTERSECTION( $Q_1, Q_2, V_1, V_2, V_3$ )
2:    $D \leftarrow Q_2 - Q_1$ 
3:    $E_1 \leftarrow V_2 - V_1$ 
4:    $E_2 \leftarrow V_3 - V_1$ 
5:    $T \leftarrow Q_1 - V_1$ 
6:    $P \leftarrow D \times E_2$ 
7:    $det \leftarrow E_1 \cdot P$ 
8:   if  $det < 0$  then
9:     return No intersection
10:  end if
11:   $invDet \leftarrow 1/det$ 
12:   $Tvec \leftarrow T \times E_1$ 
13:   $u \leftarrow (Tvec \cdot P) \times invDet$ 
14:  if  $u < 0$  or  $u > 1$  then
15:    return No intersection
16:  end if
17:   $Qvec \leftarrow T \times E_2$ 
18:   $v \leftarrow (D \cdot Qvec) \times invDet$ 
19:  if  $v < 0$  or  $u < 0$  or  $u + v > 1$  then
20:    return No intersection
21:  end if
22:   $t \leftarrow (E_2 \cdot Qvec) \times invDet$ 
23:   $P \leftarrow Q_1 + tD$ 
24:  return Intersection point  $P$ 
25: end function

```

---

## E. Création du modèle triple Dixel TD

Afin de générer le modèle Triple-Dixels, il est nécessaire de créer des Dixels suivant les trois directions X, Y et Z. Pour cela, il faut identifier les zones représentant la matière ou le vide. Le processus de création des Dixels se déroule en plusieurs étapes :

1. Trie de la liste des points d'intersection sur une même droite de cellule de grille de façon à les ordonner de manière croissante.
2. Identification du type de chaque segment selon la normal du triangle dans lequel se trouve le point d'intersection, si deux normales successives sont dos à dos, le segment représente de la matière, sinon il s'agit d'un segment vide.
3. Création des Dixels à partir des segments type "matière".

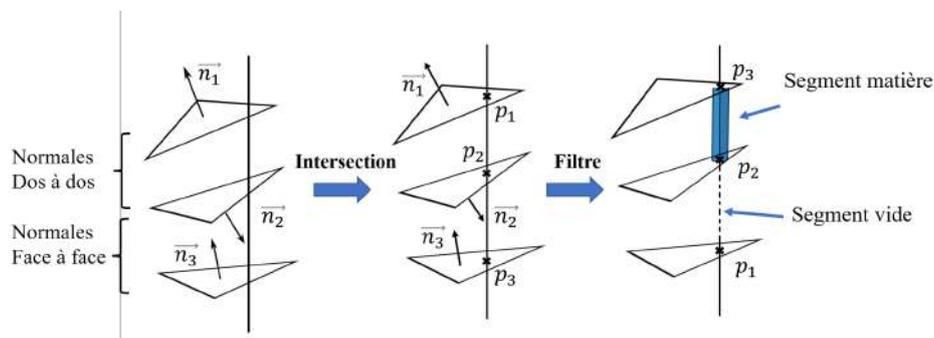


FIGURE 3.11 – Processus de création des segments type matière.

**Algorithm 11** Filter Intersections**Require:** intersections : list of intersections and normals**Ensure:** valid\_line : list of valid line segments

```

1: function FILTERINTERSECTIONS(intersections)
2:   valid_line  $\leftarrow$  []
3:   unique_intersections  $\leftarrow$  list of unique intersections
4:   if len(unique_intersections) > 1 then
5:     for  $i \leftarrow 0$  to len(unique_intersections) - 1 do
6:       delta  $\leftarrow$  unique_intersections[ $i + 1$ ][0] - unique_intersections[ $i$ ][0]
7:       N0  $\leftarrow$  unique_intersections[ $i$ ][1]
8:       N1  $\leftarrow$  unique_intersections[ $i + 1$ ][1]
9:       dp0  $\leftarrow$  dotProduct(delta, N0)
10:      dp1  $\leftarrow$  dotProduct(delta, N1)
11:      if dp0 < 0 and dp1 > 0 then
12:        valid_line.append([unique_intersections[ $i$ ][0], unique_intersections[ $i + 1$ ][0]])
13:      end if
14:    end for
15:  end if
16:  return valid_line
17: end function

```

Le Dixel volumique est généré à partir du segment matière par l'intermédiaire d'un parallélépipède défini par une hauteur et les deux arêtes de la base étant les mêmes que les arêtes de la cellule de la grille.

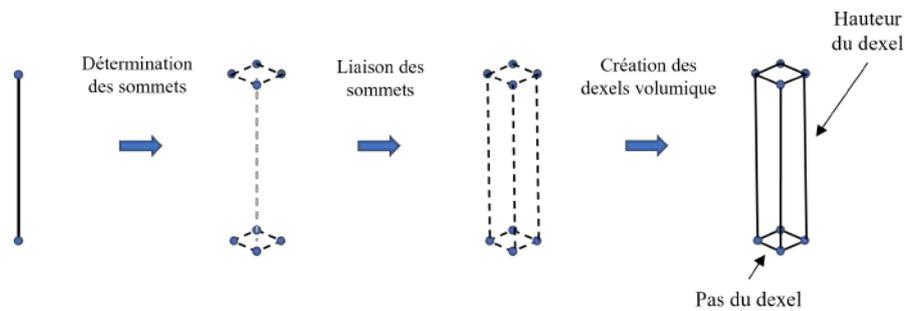


FIGURE 3.12 – Processus de création d'un Dixel volumique à partir d'un segment.

### 3.3.2 Simulation d'usinage d'enlèvement de matière

La simulation d'usinage d'enlèvement de matière consiste à usiner un modèle volumique .TD traité spécifiquement. Ce traitement permet de les préparer afin d'être correctement et efficacement exploitées dans le module logiciel de simulation.

#### 3.3.2.1 Ajout de la surépaisseur sur chaque Dixel

Ayant acquis le modèle .TD par la lecture directe ou bien la génération, la première étape du traitement pour la simulation d'enlèvement de matière consiste à ajouter une certaine surépaisseur  $e$ . Le but de cet ajout est de représenter la surface de la pièce avant l'opération de finition ainsi que la profondeur de passe.

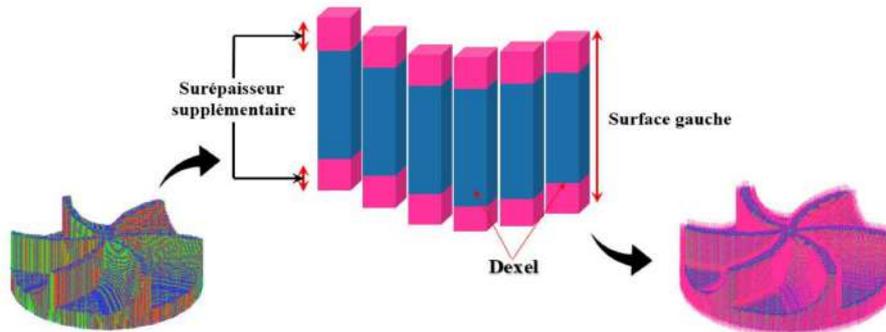


FIGURE 3.13 – Ajout d'une sur-épaisseur sur un modèle .TD [3].

#### 3.3.2.2 Classification du modèle TD épaissi

La classification du modèle TD, a pour but la réduction considérable du temps global d'exécution et de simulation. Les regroupements des segments font que les intersections seront calculées uniquement pour les segments qui se superposent avec l'enveloppe de la sphère. La classification se fait par cellules ou K-Means.

##### A. Partitionnement par cellules

Le partitionnement par cellules du modèle .TD suit le même principe que le partitionnement du modèle STL, la différence réside dans le fait que les éléments à classifier sont des segments et non des triangles.

1. Création des cellules : à partir de l'enveloppe du modèle TD à discrétiser, en spécifiant le nombre de cellules selon les axes X, Y et Z. Les dimensions et les coordonnées des points extrêmes de chaque cellule sont ensuite calculées.
2. Affectation des segments aux cellules : elle se fait par comparaison d'un segment avec les limites d'une cellule. Dans le cas où une partie d'un segment appartient à une cellule, ce dernier sera subdivisé et affecter à la cellule correspondante.

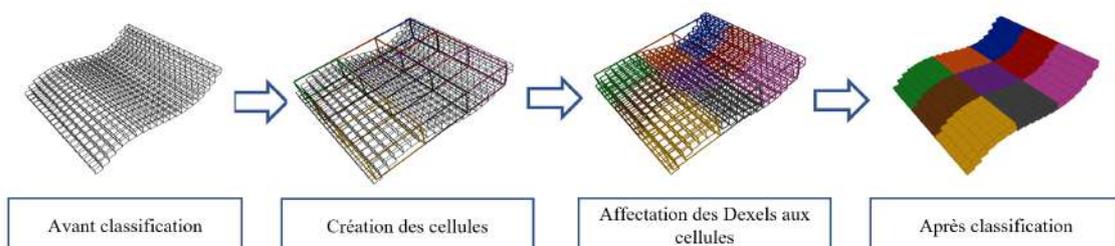


FIGURE 3.14 – Processus du partitionnement par cellules d'un modèle TD.

## B. Partitionnement par K-Means

Deux approches s'offrent à l'utilisateur en ce qui concerne le partitionnement par K-Means, soit de générer un nouveau partitionnement du modèle TD ou d'importer un partitionnement existant issu du partitionnement du modèle STL.

### • Génération d'un nouveau partitionnement K-Means

Le partitionnement par K-Means du modèle TD regroupe les Dexels en zones. Chaque zone est constituée d'une liste de Dexels et d'un point spécifique appelé "Centroïde", qui correspond au point centre des Dexels. Les étapes du processus sont similaires à celles que le partitionnement par K-Means du modèle STL, dans ce cas, les éléments à classifier sont les centres de Dexels.

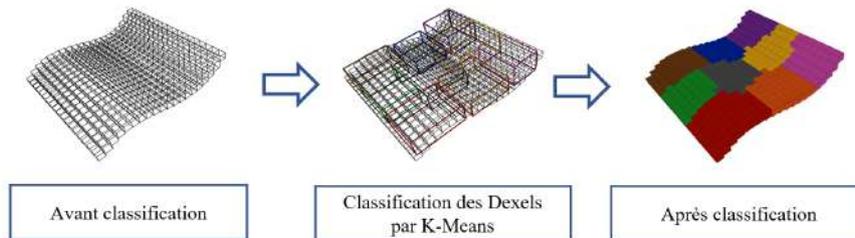


FIGURE 3.15 – Processus du partitionnement par K-Means d'un modèle TD

### • Acquisition d'un partitionnement K-Means existant

La classification des Dexels suivant leurs centres peut présenter quelques anomalies pour certains modèles TD, pour y remédier, il est possible d'importer un partitionnement K-Means existant sous fichier .KM (K-Means). Ce fichier contient des informations relatives aux paramètres de partitionnement, ainsi que les coordonnées des différentes enveloppes. Ayant les enveloppes, les segments seront affectés et subdivisés dans chaque enveloppe dans laquelle ils appartiennent, en utilisant le même algorithme que pour l'intersection segment-cellule.

```

partitionnement_km - Notepad
File Edit Format View Help
Nom modèle .STL : 'poignet_average.STL'
Méthode de partitionnement : 'K-Means'
Méthode d'initialisation : 'k-means++'
Nombre de partitions (K-cluster) : 5
Critère d'arrêt : 'Itération Max', 100
Chevauchement [mm^3]: 11848.40968465805

Enveloppes (p_min; p_max) :
1.393305, 21.08279, 39.7142; 31.58577, 40.68089, 62.18723
8.544223, 8.640293, 0.00311972; 33.43707, 33.53314, 18.51311
20.99065, 3.257182, 39.71347; 40.58799, 37.56988, 62.18718
6.595047, 7.27181, 18.5117; 33.3747, 34.17898, 42.04402
1.393305, 1.486209, 39.71309; 29.13168, 23.87254, 62.18695

```

FIGURE 3.16 – Fichier .KM type ASCII.

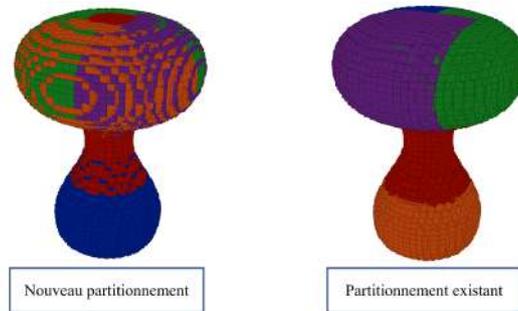


FIGURE 3.17 – Différence entre un partitionnement généré et acquis.

**Algorithm 12** Segment-Box Intersection**Require:** bounding\_boxes : a list of bounding boxes**Ensure:** affected\_dexel : segment within the intersection, affected\_dexel\_label : label corresponding to affected\_dexel

```

1: function PARTITIONDEXEL(dexel, sub_bounding_boxes)
2:   for each index and box in enumerate(bounding_boxes) do
3:     P_0 as the starting point of segment
4:     P_1 as the ending point of segment
5:     if P_0 and P_1 are entirely inside box then
6:       Add p_0 and P_1 to affected_dexel
7:       Add index to affected_dexel_label
8:       Continue to the next segment
9:     end if
10:    if P_0 and P_1 are not inside box then
11:      Continue to the next segment
12:    end if
13:    Calculate the intersection of segment with each face of box
14:    Calculate the intersection points tmin and tmax
15:    Set  $P_{intersection_0}$  as  $P_0 + tminx(P_1 - P_0)$ 
16:    Set  $P_{intersection_1}$  as  $P_1 + tmaxx(P_1 - P_0)$ 
17:    if  $P_{intersection_0}$  and  $P_{intersection_1}$  are included in the segment then
18:      affected_dexel  $\leftarrow [P_{intersection_0}, P_{intersection_1}]$ 
19:      affected_dexel_label  $\leftarrow$  index
20:    end if
21:  end for
22:  Return affected_dexel and affected_dexel_label
23: end function

```

**3.3.2.3 Acquisition du trajet d'outil d'usinage par des logiciels de CFAO**

L'acquisition du trajet d'outil d'usinage se fait par la lecture d'un fichier d'extension .CE ASCII. Les fichiers .CE représentent un sous-ensemble des fichiers .CL. Les fichiers .CE sont spécialement dédiés aux outils hémisphériques et contiennent des informations élémentaires telles que les coordonnées du centre de l'outil, sa longueur et son rayon.

**A. Acquisition du fichier .CL**

Les fichiers .CL / .CLT (cutter location files) sont des fichiers générés à partir des trajectoires d'outils, ils font référence à la position dans laquelle une fraiseuse CNC a été chargée de tenir l'outil d'usinage.

L'acquisition d'un fichier type .CL se fait par des logiciels de FAO tel que SolidWorks-CamWorks, ils comprennent la position de l'outil et l'orientation de l'outil par rapport au système de coordonnées de la pièce, mais aussi des informations relatives aux vitesses de coupes, les dimensions de l'outil utilisé ainsi que le nom de l'opération.

## B. Conversion du fichier .CL en fichier .CE

Afin d'être exploitable dans l'application de simulation, le fichier .CL doit être converti en fichier .CE, afin de permettre une implémentation précise et simple du trajet d'outil. La conversion se fait par l'extraction des informations significatives à la simulation (position de l'outil, rayon, rotation, etc.).

---

### Algorithm 13 .CL to .CE Conversion

---

**Require:** *file\_path* is a valid file path to a .CL file

**Ensure:** *tool\_contour* Array containing the converted data in .CE format

```

1: function CONVERTCLTOCE(file_path)
2:   tool_path ← []
3:   tool_contour ← []
4:   rotations ← []
5:   open file_path in read mode
6:   for all line in file_path do
7:     split line into individual words
8:     if line[0] is 'CUTTER/' then
9:       extract tool informations from line
10:    end if
11:    if line[0] is 'ROTABL/' then
12:      extract rotations from line and affect it to
13:    end if
14:    if line[0] is 'GOTO/' then
15:      extract the coordinate values from line
16:      append coordinate to tool_contour
17:    end if
18:    if line[0] is 'RAPID/' then
19:      if length of tool_contour > 0 then
20:        append the tool_contour to tool_path
21:        reset tool_contour
22:      end if
23:    end if
24:  end for
25:  close the file
26:  apply rotation transformations and origin translation to tool_contour and compute tool angles
27:  return tool_contour
28: end function

```

---

## C. Acquisition du fichier .CE

Ayant obtenu le fichier .CE, l'acquisition de ce dernier se fait par la lecture de chaque ligne du fichier. La relation entre le point CE et le point CL est exprimée par la formule suivante :

$$\overrightarrow{OC}_L = \overrightarrow{OC}_E - r \cdot \vec{u} \quad (3.1)$$

Avec :

- CC : Cutter contact, représente le point de contact entre l'outil et la surface.
- CE : Représente le point centre de l'outil.
- CL : Cutter Location, représente le point d'extrémité de l'outil.
- $\vec{u}$  : représente l'axe de l'outil.
- r : représente le rayon de l'outil hémisphérique.

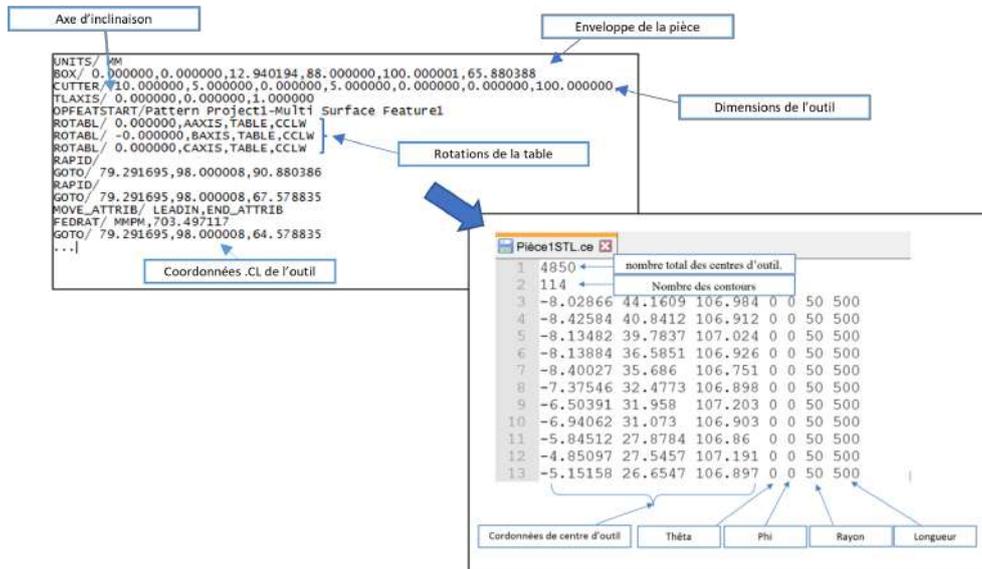


FIGURE 3.18 – Conversion du fichier .CL issu de *CAMWorks 2018* en .CE.

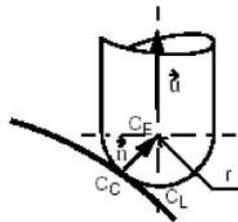


FIGURE 3.19 – Représentation du point CE et CL.

### 3.3.2.4 Simulation enlèvement de matière

La simulation d'usinage consiste à représenter le mouvement de l'outil le long du trajet d'usinage ainsi que d'évaluer les différentes intersections entre les segments de Dexels et l'outil hémisphérique afin d'obtenir les nouvelles longueurs des segments simulant ainsi l'enlèvement de matière.

#### A. Représentation de l'outil hémisphérique

En raison de la rotation à grande vitesse de l'outil, il est souvent pratique de considérer l'outil hémisphérique comme une sphère pour simplifier les calculs et les analyses.

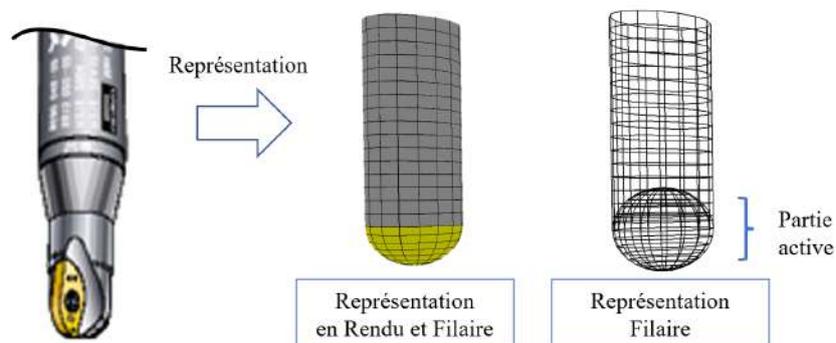


FIGURE 3.20 – Approximation d'un outil hémisphérique.

Cette approximation simplifie les calculs de la géométrie de coupe. Les propriétés de la sphère, telles que le rayon et le centre, peuvent être utilisées pour décrire efficacement l'outil hémisphérique. Les outils de modélisation et de simulation peuvent ainsi prendre en compte cette simplification pour optimiser les paramètres de coupe et améliorer les performances de l'usinage.

## B. Calcul de l'intersection outil-segment

Le calcul d'intersection entre les segments du modèle TD et l'outil hémisphérique permet de déterminer les zones où l'outil entre en contact avec de la matière afin de simuler par la suite l'enlèvement de matière. L'algorithme qui a été utilisé dans le module logiciel pour le calcul d'intersection est l'algorithme "Intersecting Segment Against Sphere" retrouvé dans [40]. L'annexe [B] détaille sur la procédure de détermination des points d'intersections.

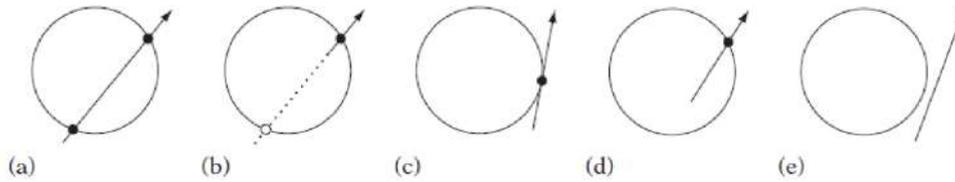


FIGURE 3.21 – Différents cas possible d'intersection Segment-Sphère.

---

### Algorithm 14 Line-Sphere Intersection

---

**Require:** Line parameters :  $P_0, P_1$ ; Sphere parameters :  $C, r$

**Ensure:** Intersection point :  $P_{\text{intersection}}$  or "No intersection"

```

1: function LINESPHEREINTERSECTION( $P_0, P_1, C, r$ )
2:    $\mathbf{d} \leftarrow P_1 - P_0$ 
3:    $\mathbf{oc} \leftarrow P_0 - C$ 
4:    $a \leftarrow \mathbf{d} \cdot \mathbf{d}$ 
5:    $b \leftarrow 2 \cdot (\mathbf{d} \cdot \mathbf{oc})$ 
6:    $c \leftarrow \mathbf{oc} \cdot \mathbf{oc} - r^2$ 
7:    $\Delta \leftarrow b^2 - 4ac$ 
8:   if  $\Delta < 0$  then
9:     return "No intersection"
10:  else
11:    if  $\Delta = 0$  then
12:       $t \leftarrow -\frac{b}{2a}$ 
13:       $P_{\text{intersection}} \leftarrow P_0 + t \cdot \mathbf{d}$ 
14:      return  $P_{\text{intersection}}$ 
15:    else
16:       $t_1 \leftarrow \frac{-b + \sqrt{\Delta}}{2a}$ 
17:       $t_2 \leftarrow \frac{-b - \sqrt{\Delta}}{2a}$ 
18:       $P_{\text{intersection1}} \leftarrow P_0 + t_1 \cdot \mathbf{d}$ 
19:       $P_{\text{intersection2}} \leftarrow P_0 + t_2 \cdot \mathbf{d}$ 
20:      return  $P_{\text{intersection1}}, P_{\text{intersection2}}$ 
21:    end if
22:  end if
23: end function

```

---

### 3.3.2.5 Topographie de surface de la pièce volumique usinée

Après avoir réalisé la simulation d'enlèvement de matière, la dernière étape consiste à évaluer et analyser la topographie de la surface usinée. Cette analyse se déroulera en deux volets principaux. Tout d'abord, il sera procédé à la quantification des cinq paramètres d'état de surface clé définie dans le chapitre 1. Ensuite, une approche supplémentaire consistera à vérifier le respect d'un intervalle de tolérance prédéfini. L'objectif de cette analyse est d'évaluer la conformité de la pièce usinée.

#### A. Calcul des paramètres d'état de surface

Le calcul des paramètres d'état de surface se fait par le calcul des écarts obtenus après l'usinage de du modèle. Un écart est représenté par la différence entre la distance du segment idéal et le segment usiné.

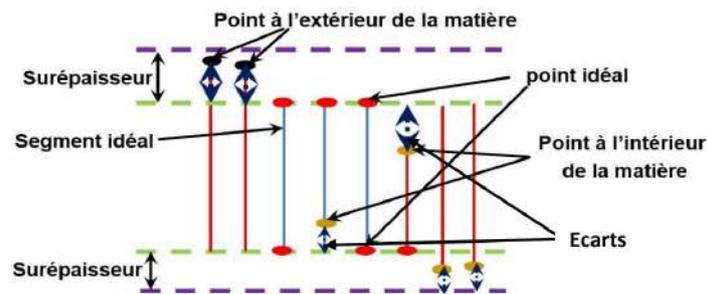


FIGURE 3.22 – Ecarts obtenus après simulation [3].

$$e = d_f - d_i \quad (3.2)$$

Où  $d_f$  représente la distance du segment usinée final et  $d_i$  représente le segment idéal.

#### B. Vérification du respect de tolérance

La vérification du respect de la tolérance s'effectue en examinant si les points d'extrémité du segment usiné appartiennent à l'intervalle de tolérance défini. Pour cela, une couleur est attribuée au segment en fonction de son appartenance ou non à l'intervalle spécifié. Cette approche visuelle permet de rapidement identifier si le segment respecte les critères de tolérance définis, facilitant ainsi l'évaluation de la conformité de la pièce usinée. Trois cas possibles se présentent :

- Le point d'extrémité se retrouve à l'intérieur de l'intervalle, le segment correspondant sera représenté par la couleur verte.
- Le point d'extrémité se retrouve à l'extérieur de l'intervalle et supérieur à la valeur maximum de l'intervalle, le segment correspondant sera représenté par la couleur bleue.
- Le point d'extrémité se retrouve à l'extérieur de l'intervalle et inférieur à la valeur minimum de l'intervalle, le segment correspondant sera représenté par la couleur rouge.

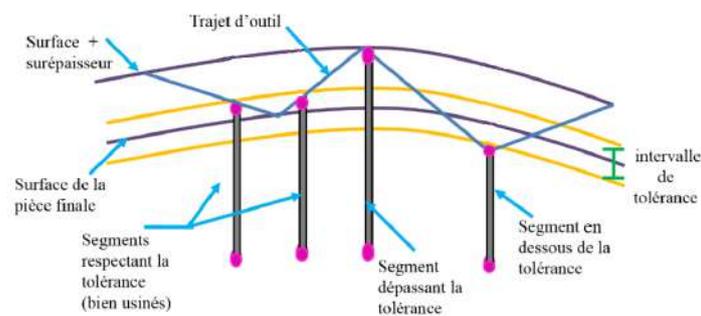


FIGURE 3.23 – Tolérance après usinage [3].

## 3.4 Conclusion

Ce chapitre a été consacré à l'étude approfondie de la conception, de l'analyse et de l'architecture générale du module logiciel que nous avons développé. La démarche de l'application a été présentée en détail en couvrant les différents algorithmes et procédures clés de notre module logicielle. Grâce à cette analyse, nous avons pu atteindre un niveau de détail suffisant, essentiel pour passer à la phase de réalisation et implémentation.

## Chapitre 4

# Implémentation et réalisation

## 4.1 Introduction

Dans ce dernier chapitre, la phase concrète de mise en œuvre du module logiciel sera abordée, suite à l'élaboration de sa conception. Dans un premier temps, nous détaillerons sur les outils, les langages et les techniques spécifiques utilisés. Ensuite, nous présenterons les scénarios de fonctionnement les plus généraux du module logiciel, illustrés par des captures d'écran. Dans une troisième partie, nous exposerons et discuterons les résultats de génération de modèles TD obtenus ainsi que les résultats de simulations d'enlèvement de matière.

## 4.2 Environnement de travail

La démarche de résolution, ainsi que le développement de l'application logicielle ont été réalisés dans un environnement de travail précis comprenant à la fois du matériel et du logiciel.

### 4.2.1 Environnement matériel

L'environnement matériel représente tout matériel utilisé lors de la réalisation du module logiciel. Un ordinateur portable classique a été utilisé pour la réalisation de l'application, ayant pour caractéristiques les suivantes :

- **Processeur** : Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz ;
- **Cœurs de processeur** : 2 Cores ;
- **RAM** : 16.0 GB ;
- **Système d'exploitation** : Windows 10 Pro ;
- **Carte graphique** : Intel(R) HD Graphics 620.

### 4.2.2 Environnement logiciel et langage

L'environnement logiciel et langage représente l'ensemble des logiciels, outils et langages de programmation utilisés pour la réalisation de l'application.

#### 4.2.2.1 Visual Studio Code



Visual Studio Code (VSCoDe) est un environnement de développement intégré (IDE) populaire et largement utilisé par les développeurs pour créer une variété d'applications. Conçu et maintenu par Microsoft, Visual Studio Code est une application gratuite et open source qui offre une grande flexibilité et une richesse de fonctionnalités.

#### 4.2.2.2 OpenGL



OpenGL (Open Graphics Library) est une bibliothèque de programmation graphique largement utilisée pour créer des applications interactives 2D et 3D. Elle offre un ensemble de fonctions et de commandes qui permettent aux développeurs d'accéder aux fonctionnalités graphiques des systèmes informatiques.

OpenGL est conçu pour être portable et compatible avec différentes plateformes, elle peut être utilisé sur une variété de systèmes d'exploitation tels que Windows, macOS, Linux et des environnements mobiles.

### 4.2.2.3 PyQt5 Designer



PyQt5 Designer est un outil visuel convivial qui fait partie de la bibliothèque PyQt5, un ensemble de modules Python permettant de créer des interfaces graphiques. PyQt5 Designer facilite la conception et la création d'interfaces utilisateur en adoptant une approche de développement visuel.

Avec PyQt5 Designer, les développeurs peuvent créer des interfaces graphiques en utilisant des composants prédéfinis tels que les boutons, les champs de texte, les cases à cocher, les listes déroulantes, etc. Il permet également de personnaliser l'apparence des éléments d'interface en ajustant les propriétés, les styles et les mises en page.

### 4.2.2.4 SolidWorks - CAMWorks



CAMWorks est la première solution entièrement intégrée de Fabrication assistée par ordinateur (FAO) spécifiquement conçue pour fonctionner avec SolidWorks. Elle offre des fonctionnalités d'usinage associatif et basé sur la connaissance directement dans SolidWorks. Les modules d'usinage comprennent le fraisage 2.5 axes, le fraisage 3 axes, le tournage, le fraisage-tournage avec prise en charge de l'usinage simultané à 5 axes, l'usinage simultané à 4/5 axes et l'électroérosion par fil.

Il s'agit d'un add in qui permet un contrôle total de la simulation de fabrication d'une pièce SolidWorks. Il permet la gestion des trajectoires d'usinage, la définition des distances d'engagement, la configuration de différents types d'outils tels que les forets, les fraises cylindriques, toriques et hémisphériques.

### 4.2.2.5 Langage Python



Python est connu pour sa syntaxe claire et lisible, il favorise la simplicité et la lisibilité du code, ce qui permet de développer rapidement des applications et de prototyper des idées. Il dispose d'une vaste bibliothèque standard qui offre de nombreuses fonctionnalités prêtes à l'emploi, ce qui facilite le développement d'une large gamme d'applications, allant des scripts simples aux projets complexes.

### 4.2.2.6 Langage C



Le langage C est un langage de programmation de bas niveau, souvent considéré comme un langage de programmation procédurale. Il est réputé pour sa performance, sa proximité avec le matériel et sa portabilité. Le langage C fournit un contrôle précis sur la mémoire et permet de programmer efficacement des systèmes et des logiciels bas niveau, tel que les systèmes d'exploitation, les pilotes de périphériques et les applications qui nécessitent des performances optimales.

Une caractéristique clé du langage C est sa compilation préalable. Lorsqu'un programme écrit en langage C est compilé, il est traduit en langage machine spécifique à une plateforme particulière. Cette étape de compilation transforme le code source en un exécutable binaire qui peut être directement exécuté par la machine cible, ce qui rend le langage C extrêmement rapide en termes d'exécution. La compilation préalable permet d'optimiser le code et d'exploiter efficacement les ressources matérielles disponibles.

### 4.3 Présentation graphique

Le module logiciel est constitué de deux sous applications, chacune possédant un rôle bien précis, la première permet la génération du modèle TD tandis que la deuxième permet la simulation d'usinage de ce dernier.

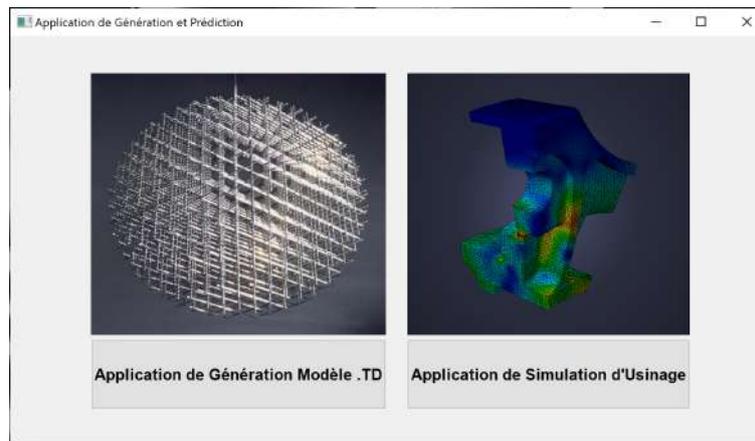


FIGURE 4.1 – Présentation de la fenêtre générale du module logiciel.

#### 4.3.1 Application de génération d'un modèle TD

L'application de génération du modèle TD est composée de deux interfaces principales. La première interface comprend toute fonctionnalités d'entrée et de sortie. La deuxième interface permet de visualiser le modèle sous OpenGL avec la possibilité de modifier différents paramètres liés la scène. La figure 4.2 présente une vue globale sur l'application :

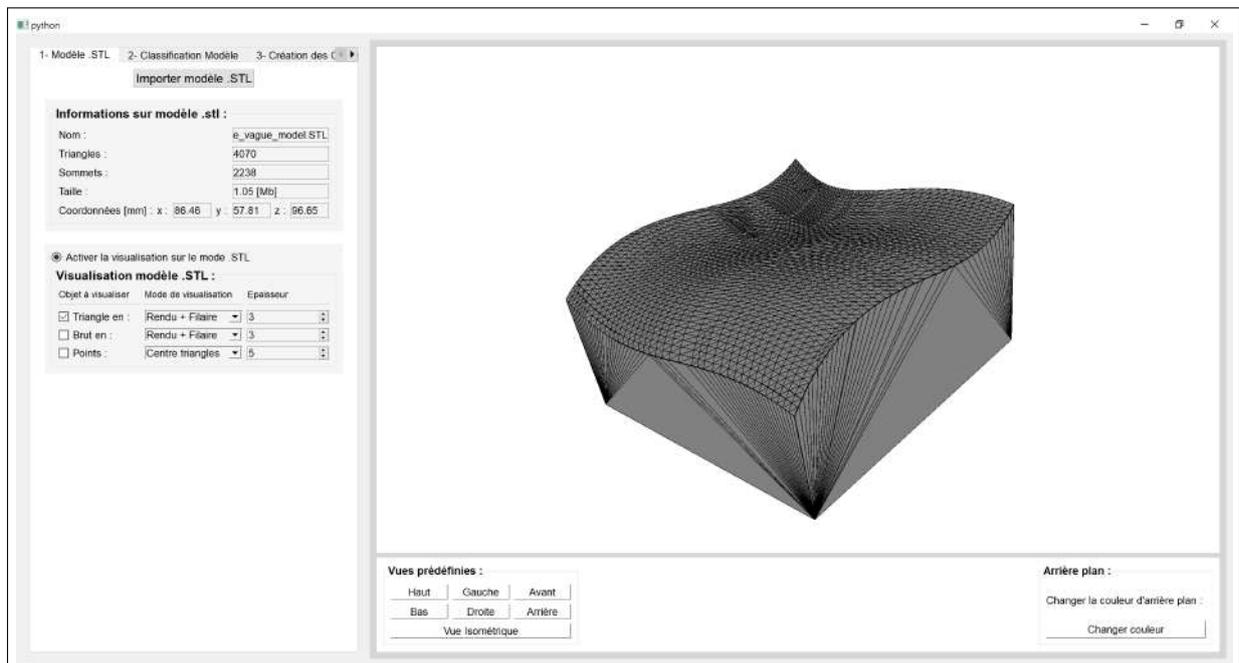


FIGURE 4.2 – Présentation de la fenêtre de génération du modèle TD.

Afin générer un modèle TD, l'utilisateur doit suivre les quatre onglets dans un ordre chronologique, les suivants :

1. Onglet "Modèle STL" ;
2. Onglet "Classification Modèle STL" ;
3. Onglet "Création des Grilles" ;
4. Onglet "Génération TD".

#### 4.3.1.1 Onglet "1- Modèle STL"

L'onglet "1- Modèle STL" a pour principal objectif l'acquisition du modèle STL issu de logiciels de conception et modélisation (CAO), en affichant un ensemble d'informations utiles à l'utilisateur. Cet onglet donne aussi la possibilité de visualiser les différents éléments du modèle (triangles, sommets, centres, etc.) séparément ou ensemble.

Objet à visualiser	Mode de visualisation	Epaisseur
<input type="checkbox"/> Triangle en :	Rendu + Fileire	3
<input type="checkbox"/> Brut en :	Rendu + Fileire	3
<input type="checkbox"/> Points :	Centre triangles	5

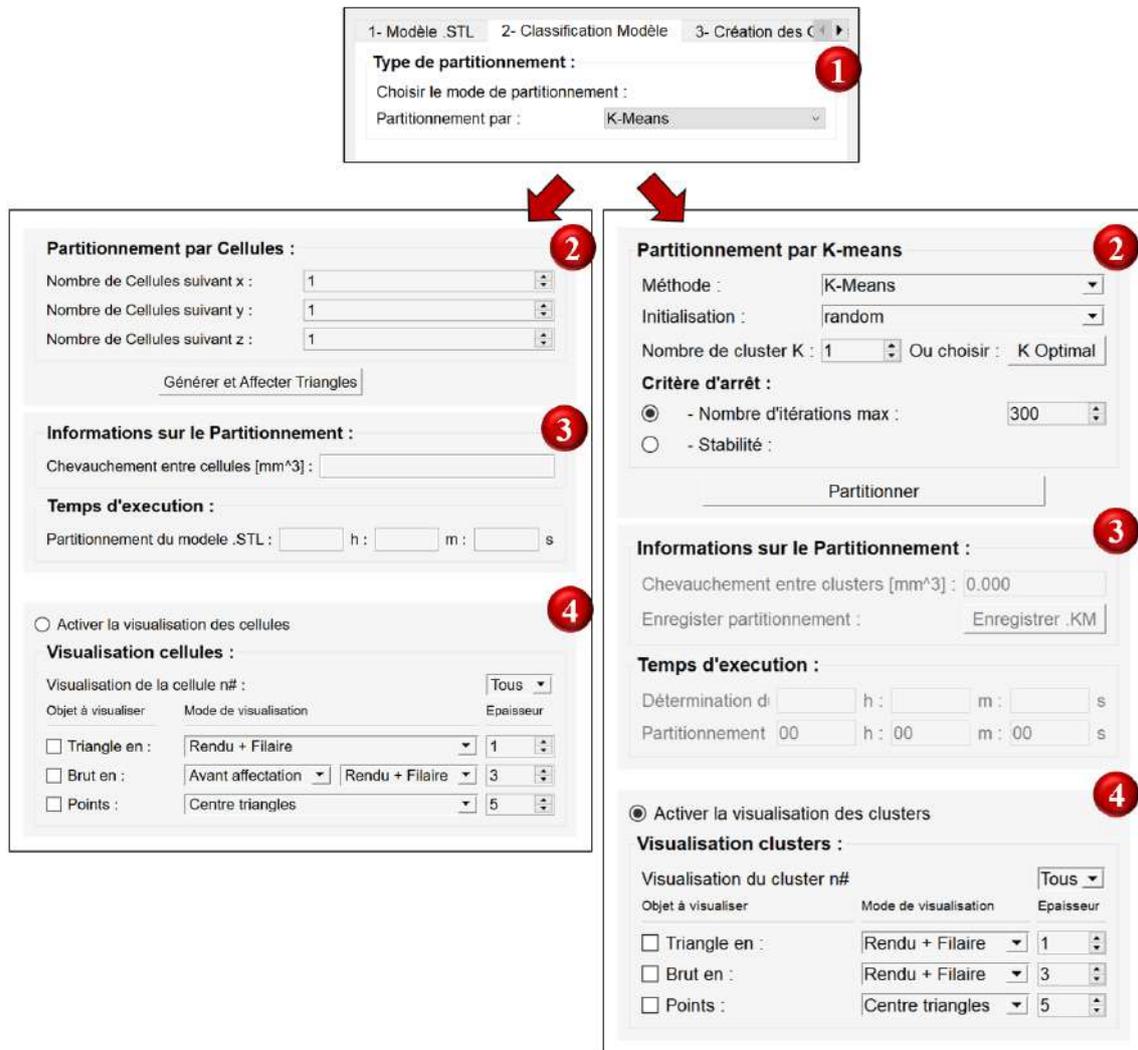
- 1) Acquisition du modèle .STL.
- 2) Informations relatives au modèle .STL.
- 3) Visualisation du modèle .STL via OpenGL.

FIGURE 4.3 – Représentation de l'onglet "1- Modèle STL".

### 4.3.1.2 Onglet "2- Classification Modèle STL"

L'onglet "2- Classification Modèle STL" a pour objectif la classification du modèle STL acquis, donnant la possibilité à l'utilisateur de choisir entre le partitionnement par Cellules ou par K-Means.

Selon le choix de l'utilisateur, une fenêtre de partitionnement apparaît, cette fenêtre contient les paramètres d'entrées et sorties liés au partitionnement, et donne la possibilité de visualiser le modèle STL partitionné.



- 1) Choix de la méthode de partitionnement.
- 2) Introduction des différents paramètres liés au partitionnement.
- 3) Informations sur la qualité et le temps de partitionnement.
- 4) Visualisation des partitions via OpenGL.

FIGURE 4.4 – Représentation de l'onglet "2- Classification Modèle STL".

Si l'utilisateur est peu expérimenté dans le choix du nombre de partitions (Nombre de clusters K), une fenêtre d'aide est disponible, lui permettant de trouver le nombre de partitions optimal en utilisant le critère de chevauchement.

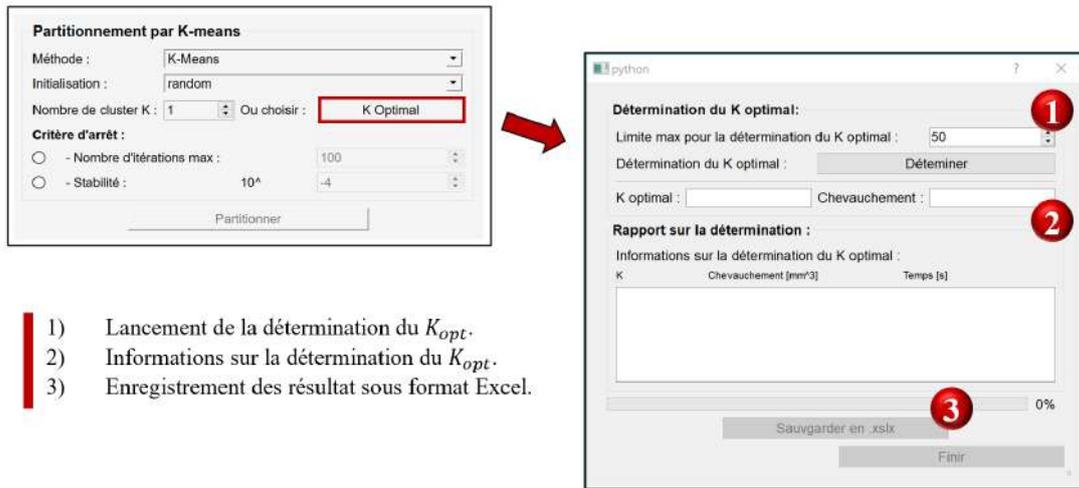


FIGURE 4.5 – Représentation de la fenêtre d’aide au choix du nombre de partitions.

### 4.3.1.3 Onglet ”3- Création des Grilles”

L’onglet ”3- Création des Grilles” a pour objectif la création des grilles qui sera par la suite utilisée pour la génération du modèle TD, en affichant un ensemble d’informations utiles à l’utilisateur. Cet onglet donne aussi la possibilité de visualiser les différents éléments de la grille (Grilles, Segments et Centres) séparément ou ensemble.

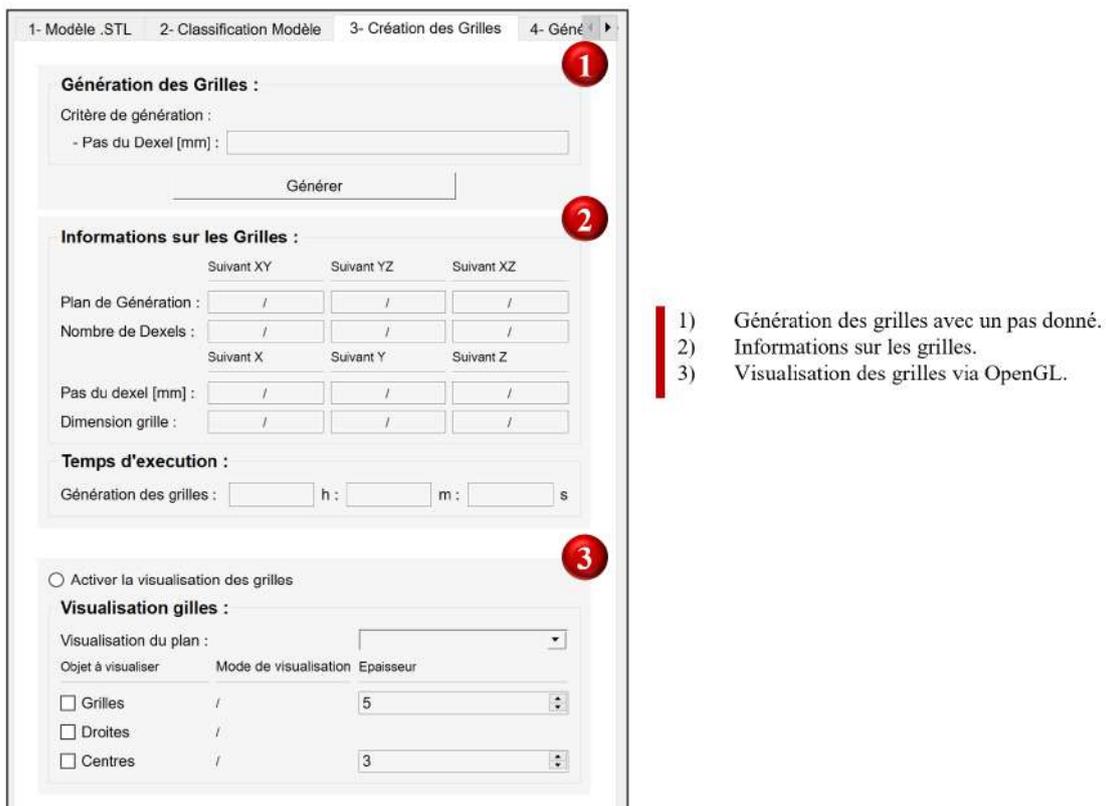


FIGURE 4.6 – Représentation de l’onglet ”3- Création des Grilles”.

#### 4.3.1.4 Onglet "4- Génération Modèle TD"

L'onglet "4- Génération Modèle TD" a pour objectif la génération et la sauvegarde du modèle TD qui sera par la suite utilisée pour la simulation d'usinage. Cet onglet donne aussi la possibilité de visualiser les différents éléments du Dexels (Dexels, Segments et Points) séparément ou ensemble.

Objet à visualiser	Mode de visualisation	Epaisseur
<input type="checkbox"/> Dixel en :	Rendu + Filaire	1
<input type="checkbox"/> Segment en :	/	3
<input type="checkbox"/> Points :	/	5

- 1) Génération du modèle Triple Dixel.
- 2) Informations sur les temps d'exécutions.
- 3) Visualisation du modèle .TD via OpenGL.

FIGURE 4.7 – Représentation de l'onglet "4- Génération Modèle TD".

### 4.3.2 Application de Simulation d'enlèvement de matière

L'application de Simulation d'enlèvement de matière est composée de deux interfaces principales, identique à la première application. La première interface comprend toutes les fonctionnalités d'entrée et de sortie. La deuxième interface permet de visualiser le modèle sous OpenGL avec la possibilité de modifier différents paramètres liés la scène. La figure 4.8 présente une vue globale sur l'application.

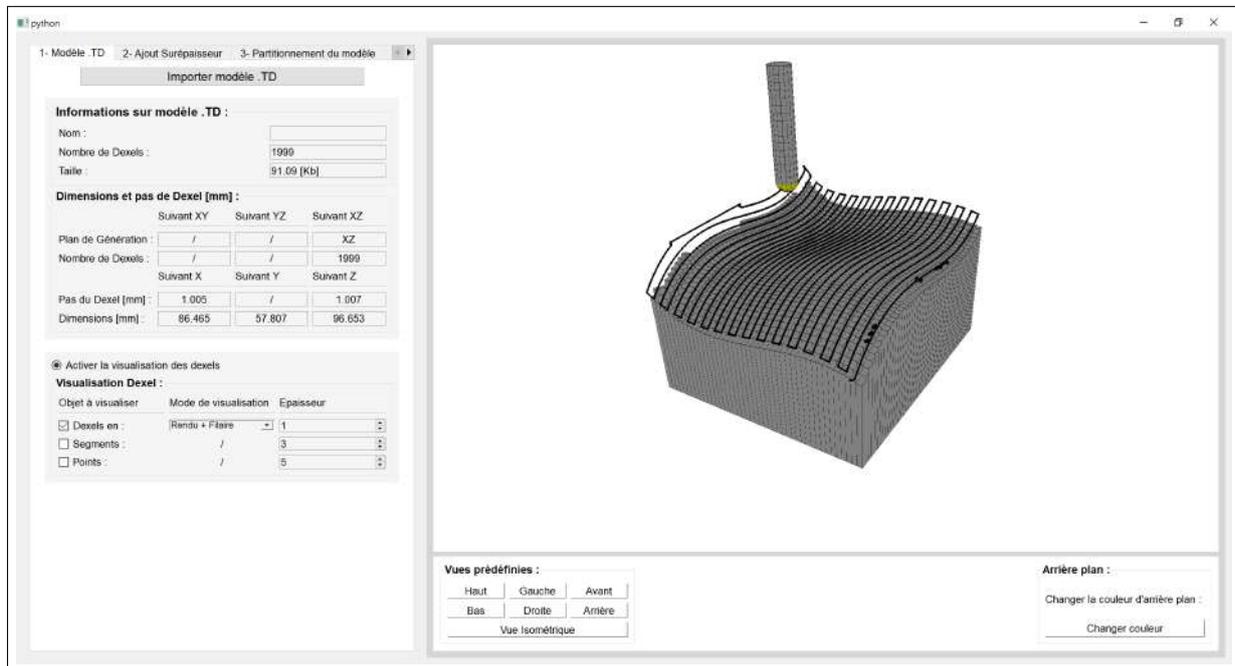


FIGURE 4.8 – Présentation de la fenêtre de simulation d'enlèvement de matière.

Afin simuler l'usinage d'enlèvement de matière du modèle TD et d'obtenir l'état de surface de ce dernier, l'utilisateur doit suivre les six onglets dans un ordre chronologique :

1. Onglet "1- Modèle TD" ;
2. Onglet "2- Ajout surépaisseur" ;
3. Onglet "3- Partitionnement modèle"
4. Onglet "4- Trajet outil"
5. Onglet "5- Simulation"
6. Onglet "6- État de surface"

#### 4.3.2.1 Onglet "1- Modèle TD"

L'onglet "1- Modèle TD" a pour principal objectif l'acquisition du modèle TD issu de l'application de génération de modèle triple Dixel, en affichant un ensemble d'informations utiles à l'utilisateur. Cet onglet donne aussi la possibilité de visualiser les différents éléments du modèle (Dexels, Segments, Points, etc.) séparément ou ensemble.

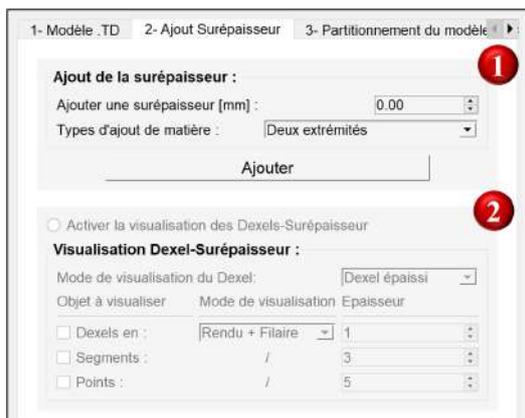


- 1) Acquisition du modèle .TD.
- 2) Informations relatives au modèle .TD.
- 3) Visualisation du modèle .TD via OpenGL.

FIGURE 4.9 – Représentation de l'onglet "1- Modèle TD".

#### 4.3.2.2 Onglet "2- Ajout surépaisseur"

L'onglet "2- Ajout surépaisseur" a pour principal objectif l'ajout d'une certaine surépaisseur sur le modèle TD acquit, selon l'extrémité du segment choisit. Cet onglet donne aussi la possibilité de visualiser le modèle TD épaissi (Dexels, Segments, Points, etc.).



- 1) Ajout d'une surépaisseur au modèle .TD.
- 2) Visualisation du modèle .TD épaissi via OpenGL.

FIGURE 4.10 – Représentation de l'onglet "2- Ajout surépaisseur".

### 4.3.2.3 Onglet "3- Partitionnement"

L'onglet "3- Partitionnement Modèle TD" a pour objectif la classification du modèle TD acquit, donnant la possibilité à l'utilisateur de choisir entre le partitionnement par Cellules ou par K-Means.

Selon le choix de l'utilisateur, une fenêtre de partitionnement apparaît, cette fenêtre contient les paramètres d'entrées et sorties liés au partitionnement, et donne la possibilité de visualiser le modèle TD partitionné.

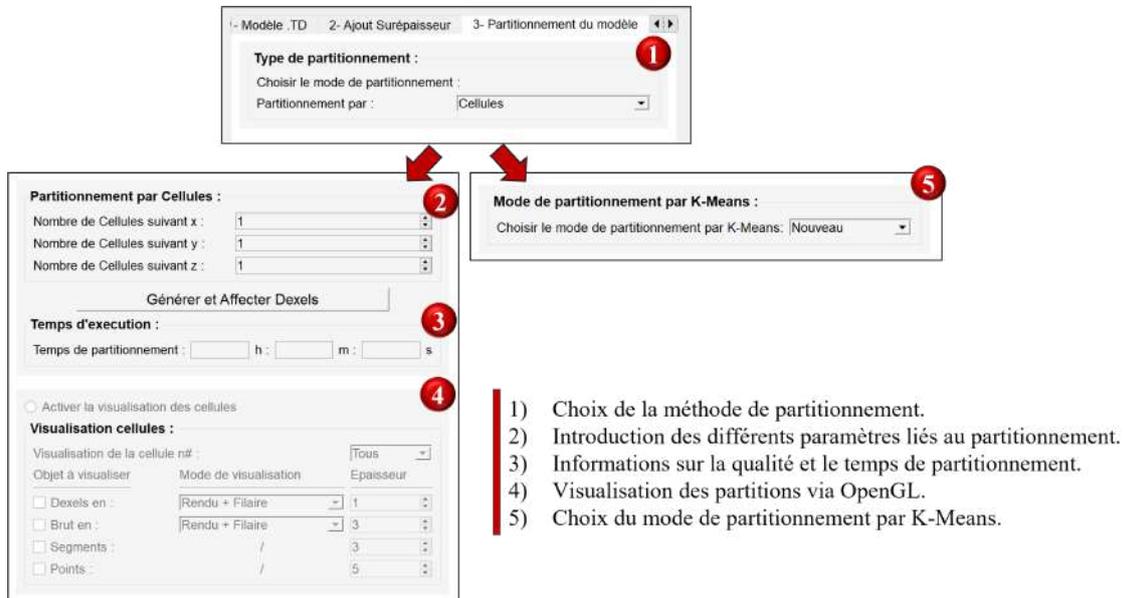


FIGURE 4.11 – Représentation de l'onglet "3- Partitionnement".

Dans le cas où l'utilisateur n'est pas satisfait du nouveau partitionnement K-Means obtenu, il est possible d'importer un partitionnement existant .KM issu de l'application de génération du modèle TD.

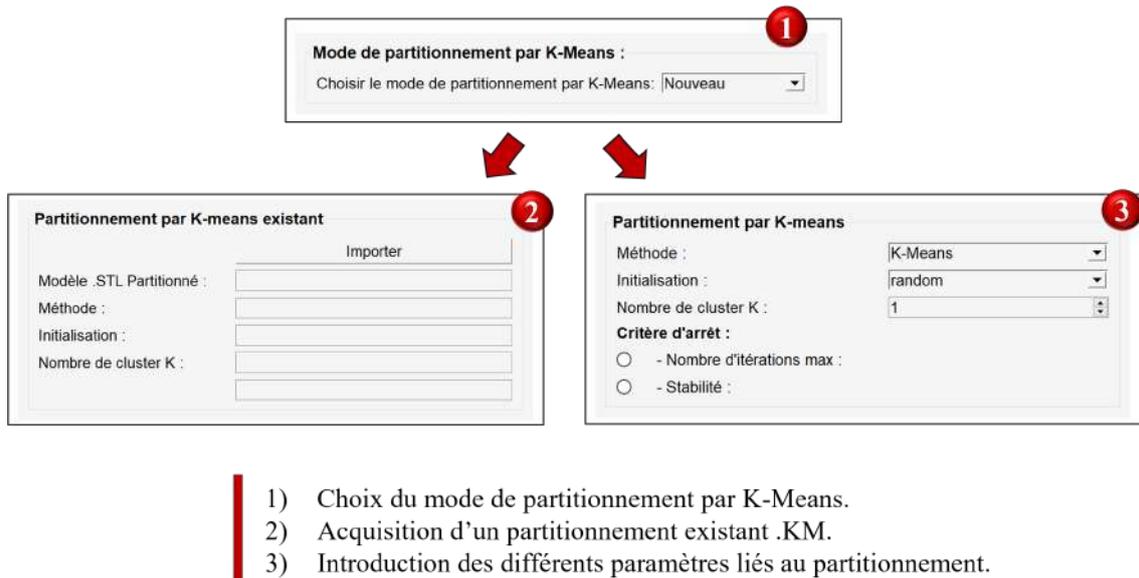
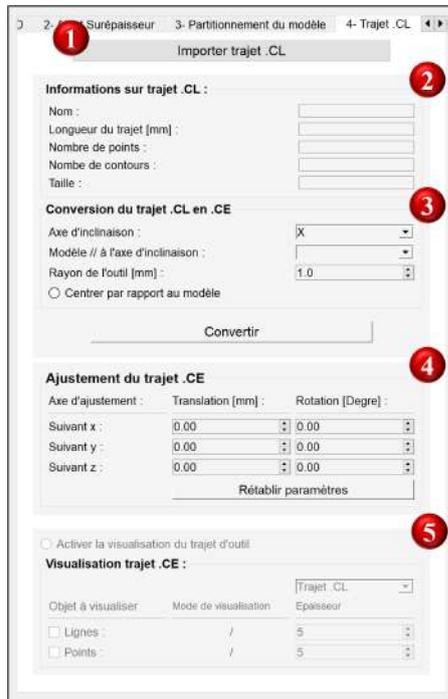


FIGURE 4.12 – Représentation de l'onglet "3- Partitionnement par K-Means".

#### 4.3.2.4 Onglet "4- Trajet outil"

L'onglet "4- Trajet outil" a pour objectif l'acquisition d'un fichier de trajet d'outil d'usinage .CL, de le convertir en un trajet .CE exploitable par le module logiciel et permet l'ajustement du trajet selon les trois directions si l'utilisateur le désire. Cet onglet permet aussi la visualisation des lignes et points du trajet d'outil d'usinage.

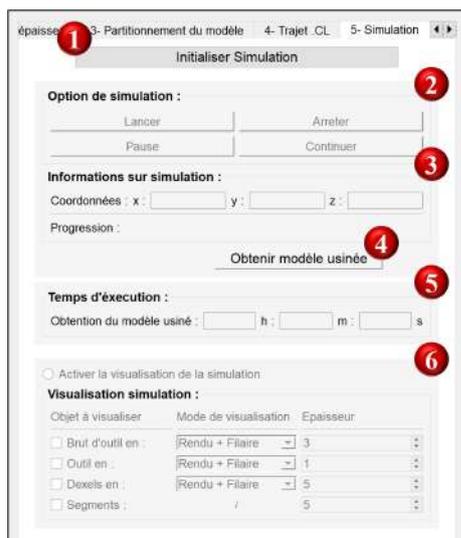


- 1) Acquisition du trajet d'outil d'usinage.
- 2) Informations relatives au trajet d'usinage.
- 3) Paramètres de conversion du trajet d'usinage .CL vers .CE.
- 4) Ajustement du trajet .CE.
- 5) Visualisation des trajets d'usinage via OpenGL.

FIGURE 4.13 – Représentation de l'onglet "4- Trajet outil".

#### 4.3.2.5 Onglet "5- Simulation"

L'onglet "5- Simulation" regroupe les différentes fonctionnalités et paramètres liés à la simulation d'usinage, en donnant des informations relatives à la position de l'outil en chaque instant. Cet onglet permet aussi la visualisation des différents éléments et objets liés à la simulation à savoir le modèle à usiner, l'outil, etc.



- 1) Initialisation de la simulation d'usinage.
- 2) Paramètres liés à la simulation.
- 3) Informations relatives au positionnement de l'outil.
- 4) Obtention du modèle usiné après simulation d'usinage.
- 5) Informations sur le temps d'exécution.
- 6) Visualisation des éléments liés à la simulation via OpenGL.

FIGURE 4.14 – Représentation de l'onglet "5- Simulation".

#### 4.3.2.6 Onglet "6- État de surface"

L'onglet "6- État de surface" représente le dernier onglet de l'application et a pour objectif le calcul des paramètres de l'état de surface du modèle TD usiné. Cet onglet permet à la fois le calcul de l'état de surface mais aussi la visualisation de la topographie de surface, en donnant divers informations significatives relatives aux différents volumes. Cet onglet permet aussi la visualisation de la topographie du modèle usiné et d'enregistrer les résultats obtenus.

- 1) Calcul des écarts après usinage.
- 2) Informations relatives aux paramètres d'état de surface.
- 3) Informations relatives au volume du modèle.
- 4) Détermination de la topographie de surface.
- 5) Visualisation de la topographie via OpenGL.
- 6) Enregistrement des résultats liés à la simulation d'usinage.

FIGURE 4.15 – Représentation de l'onglet "6- État de surface".

## 4.4 Scénario de fonctionnement

Afin de tester le module logiciel, deux entrées principales doivent être fournies, le modèle STL et le trajet d'outil d'usinage CL.

### 4.4.1 Récupération du modèle STL

Le modèle de la pièce utilisé pour réaliser les essais sur le module logiciel développé, a été obtenu à l'aide du logiciel de conception *SolidWorks 2018*. Il s'agit d'un modèle généré par l'extrusion d'une surface gauche, obtenue par deux courbes guidées différentes, par la suite le modèle SolidWorks sera enregistré sous extension *.STL* pour être exploitable dans notre module logiciel.

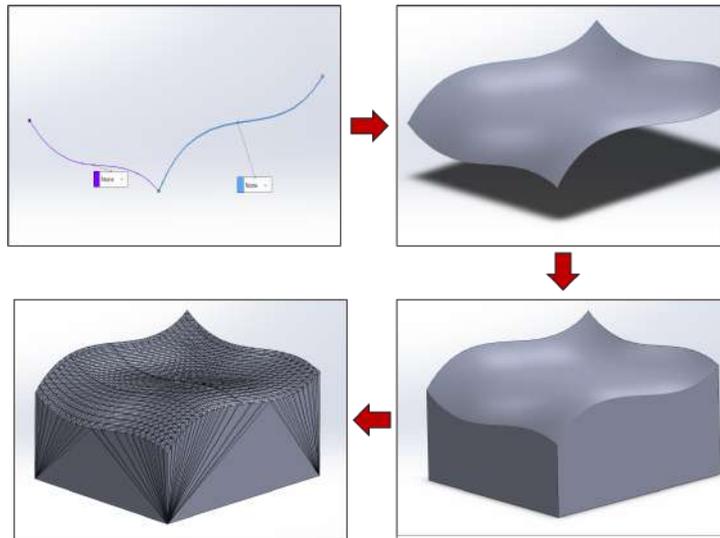


FIGURE 4.16 – Processus de conception d'une surface gauche via SolidWorks.

### 4.4.2 Récupération du trajet d'outil *.CL*

La récupération du trajet d'outil a été faite à l'aide de l'extension FAO de SolidWorks, *CAMWorks 2018*. L'obtention du trajet se fait par la création d'un plan d'opération pour définir le référentiel de travail. Ensuite, les paramètres d'usinage ont été configurés de manière personnalisée, offrant un contrôle complet sur des aspects tels que les dimensions de l'outil, la stratégie d'usinage, le dégagement et l'engagement, la vitesse d'avance, le pas d'avance et la profondeur de passe. Une fois les paramètres définis, le logiciel a calculé le parcours de l'outil, permettant la visualisation du trajet et d'éventuelles modifications. Enfin, le fichier *CL* contenant toutes les informations nécessaires pour l'usinage a été enregistré en vue d'une utilisation dans le module logiciel.

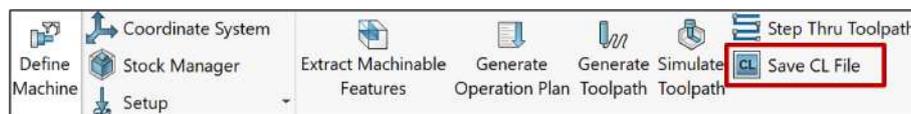


FIGURE 4.17 – Fonctionnalités permettant l'obtention du trajet *.CL*.

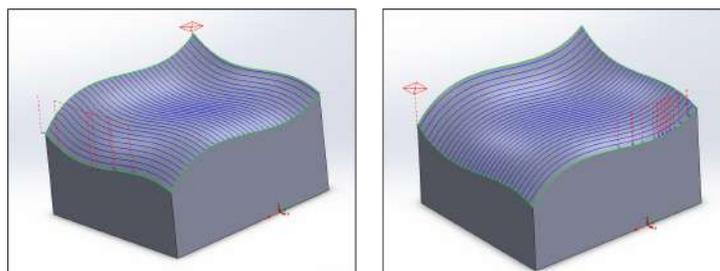


FIGURE 4.18 – Trajet *.CL* suivant les deux courbes paramétriques.

### 4.4.3 Scénario de fonctionnement du module logiciel

La figure 4.19 résume le scénario de fonctionnement de l'application de génération du modèle TD. Le modèle décrit dans 4.16 a été utilisé pour tester l'application.

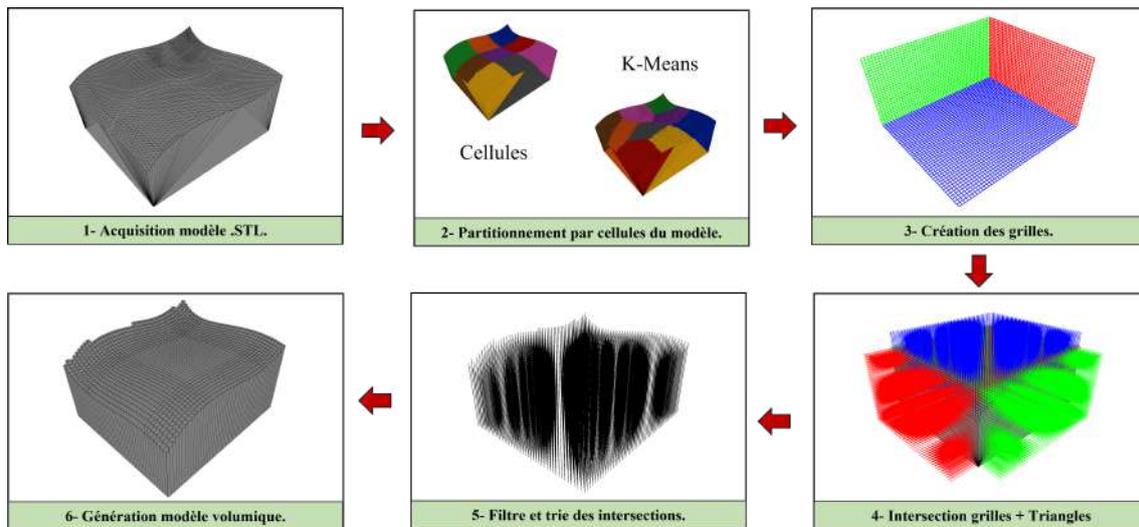


FIGURE 4.19 – Scénario de fonctionnement de l'application de génération du modèle TD.

La figure ?? résume le scénario de fonctionnement de l'application de simulation d'usinage du modèle TD. Le modèle utilisé pour tester l'application de simulation est celui obtenu via la validation de génération.

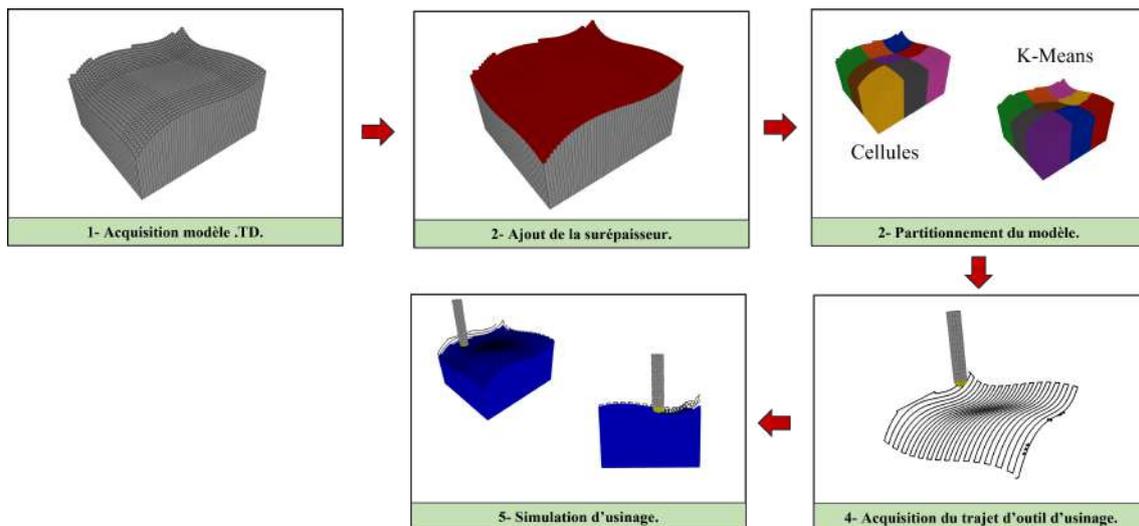


FIGURE 4.20 – Scénario de fonctionnement de l'application de simulation d'usinage.

## 4.5 Validation du module logiciel et résultat

La validation du module logiciel se fera principalement en deux étapes :

- Validation de l’optimisation du temps par l’usage de classification par la comparaison des résultats de génération et simulation obtenus sur trois modèles STL et deux modèles TD respectivement.
- Validation qualitative de la topographie de la surface avec un même modèle pour différents paramètres d’usinage.

### 4.5.1 Validation de l’optimisation de génération par partitionnement

La validation de l’optimisation du temps de calcul par le partitionnement est faite par la comparaison des temps de calculs obtenus via le partitionnement par Cellules et par K-Means. Dans ce contexte, le temps de calcul pour la génération représente la sommation du temps d’exécution de l’algorithme d’intersection Segments-Triangles avec celui du filtre des segments. La variante "k-means++" sera utilisé pour la validation, avec comme critère d’arrêt une stabilité dans l’ordre de  $10^{-6}$ .

Trois principaux modèles ont été choisis pour la validation du module logiciel, afin d’englober différents cas possibles :

- **Modèle de faible complexité** : Le modèle choisit est semblable a une double vague, obtenu par l’extrusion d’une surface gauche.
- **Modèle de moyenne complexité avec de fortes courbures** : Le modèle choisit est semblable a une poignée, il s’agit surface de révolution avec de fortes courbures.
- **Modèle de forte complexité** : Le modèle choisit est celui de l’aube de turbine.

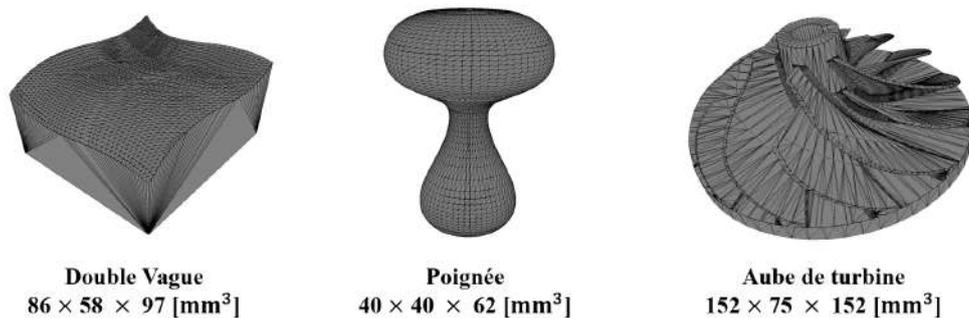


FIGURE 4.21 – Modèles utilisés pour la validation du module.

TABLE 4.1 – Résumé des différentes informations relatives aux modèles utilisés pour la validation du module.

Double Vague STL	Nombre de Triangles	Dimensions		
	4070	X : 86.46 mm	Y : 57.81 mm	Z : 96.65 mm
Poignée STL	Nombre de Triangles	Dimensions		
	9076	X : 40.08 mm	Y : 40.04 mm	Z : 62.11 mm
Aube de turbine STL	Nombre de Triangles	Dimensions		
	26512	X : 152.00 mm	Y : 75.00 mm	Z : 152.00 mm

Les tableaux 4.2, 4.3 et 4.4 représentent une comparaison entre le partitionnement par Cellules et K-Means pour les trois modèles de complexité différents, avec différentes partitions. La dernière ligne de chaque tableau représente la rapidité du partitionnement K-Means par rapport aux Cellules en pourcentage.

TABLE 4.2 – Résultats sur le partitionnement pour le modèle double vague.

Informations sur les Grilles	Pas de discrétisation			Nombre de Dexels par plan			Triangles
	X : 0.3 mm	Y : 0.3 mm	Z : 0.3 mm	XY : 40755	YZ : 45540	XZ : 68172	
Nombre de Partitions	1	2	4	8	12	18	27
<b>Partitionnement par Cellules</b>							
Temps de calcul [s]	179.5	263.4	277.95	313.7	329.7	407.5	436.0
Chevauchement [ $mm^3$ ] $\times 10^{-3}$	0	464.0	1365.7	2248.31	3059.71	2532.19	2600.0
<b>Partitionnement par K-Means</b>							
Temps de calcul [s]	179.5	175.5	199.2	184.5	218.3	330.4	404.2
Chevauchement [ $mm^3$ ] $\times 10^{-3}$	0	275.91	1084.71	1321.58	22295.67	1579.65	2360.27
Accélération par K-Means [%]	0.0 %	33.4 %	28.3 %	41.2 %	33.8 %	18.9 %	7.3 %

TABLE 4.3 – Résultats du partitionnement pour le modèle poignée.

Informations sur les Grilles	Pas de discrétisation			Nombre de Dexels par plan			Triangles
	X : 0.3 mm	Y : 0.3 mm	Z : 0.3 mm	XY : 40755	YZ : 45540	XZ : 68172	
Nombre de Partitions	1	2	4	8	12	18	27
<b>Partitionnement par Cellules</b>							
Temps de calcul [s]	379.0	208.6	240.9	208.8	194.5	264.36	246.8
Chevauchement [ $mm^3$ ] $\times 10^{-3}$	0	2.1	5.8	6.1	7.69	25.39	26.00
<b>Partitionnement par K-Means</b>							
Temps de calcul [s]	379.0	189.1	192.1	136.9	142.5	169.3	198.9
Chevauchement [ $mm^3$ ] $\times 10^{-3}$	0	0	26.0	9.1	18.7	22.5	28.0
Accélération par K-Means [%]	0.0 %	9.3 %	20.2 %	34.4 %	26.7 %	35.9 %	19.4 %

TABLE 4.4 – Résultats du partitionnement pour le modèle aube de turbine.

Informations sur les Grilles	Pas de discrétisation			Nombre de Dexels par plan			Triangles
	X : 0.3 mm	Y : 0.3 mm	Z : 0.3 mm	XY : 40755	YZ : 45540	XZ : 68172	
Nombre de Partitions	1	2	4	8	12	18	27
<b>Partitionnement par Cellules</b>							
Temps de calcul [s]	1405.90	815.7	791.2	526.9	561.4	623.1	635.24
Chevauchement [ $mm^3$ ] $\times 10^{-3}$	0	79.54	1966.9	2227.3	2725.1	3797.8	2883.86
<b>Partitionnement par K-Means</b>							
Temps de calcul [s]	1405.90	781.70	494.5	498.3	516.73	525.0	527.7
Chevauchement [ $mm^3$ ] $\times 10^{-3}$	0	353.88	498.17	1648.65	3583.25	4835.95	5686.82
Accélération par K-Means [%]	0.0 %	4.2 %	31.2 %	7.7 %	7.9 %	15.7 %	16.9 %

La figure 4.22 représente les courbes de comparaison du partitionnement par K-Means et par Cellules pour chaque modèle.

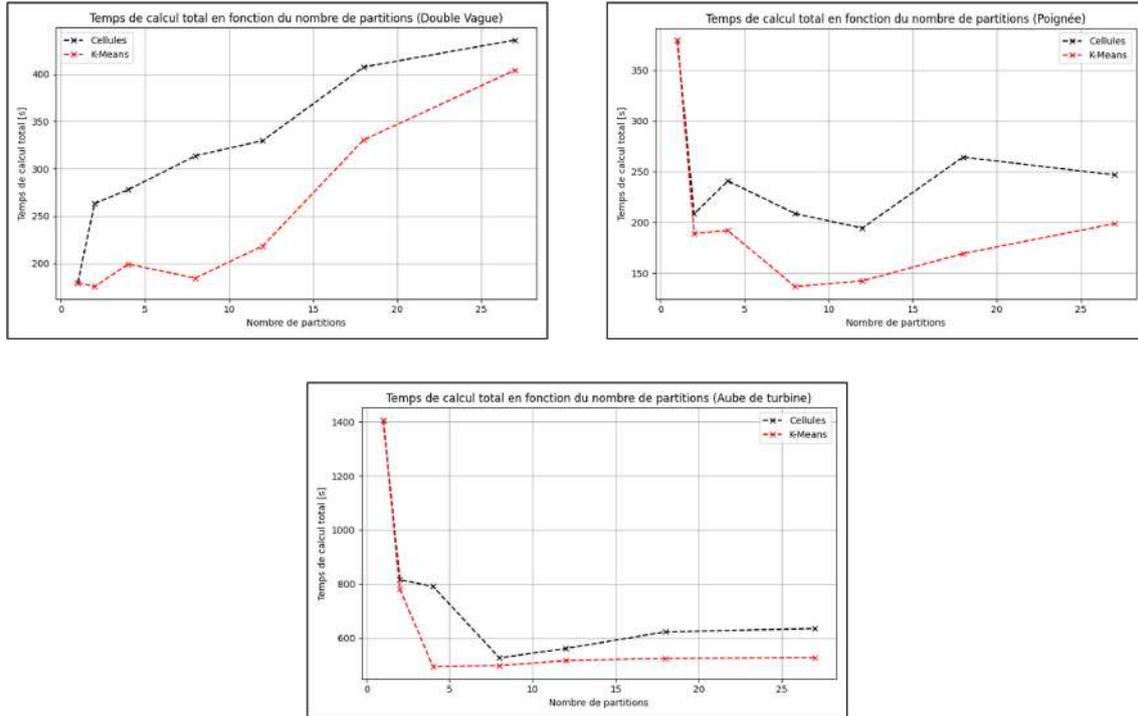
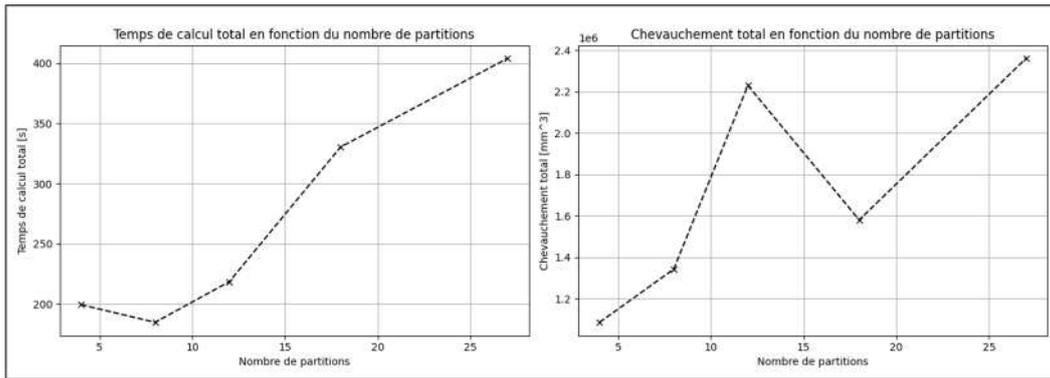


FIGURE 4.22 – Comparaison entre K-Means et Cellules pour les trois modèles.

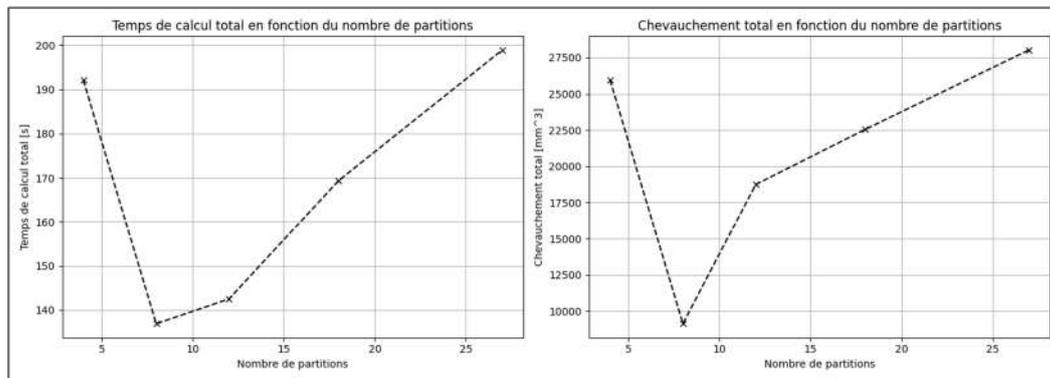
#### • Discussions

À partir des courbes obtenues, nous pouvons constater que la méthode des K-Means est nettement meilleure que celle des Cellules et ceci pour les trois types de complexité. En effet, comme le montrent les tableaux 4.2, 4.3 et 4.4 cette Méthode est de 4.2 % jusqu'à 41.2 % plus rapide que la méthode des Cellules, l'optimisation par le partitionnement est d'autant plus efficace plus la complexité de la pièce augmente.

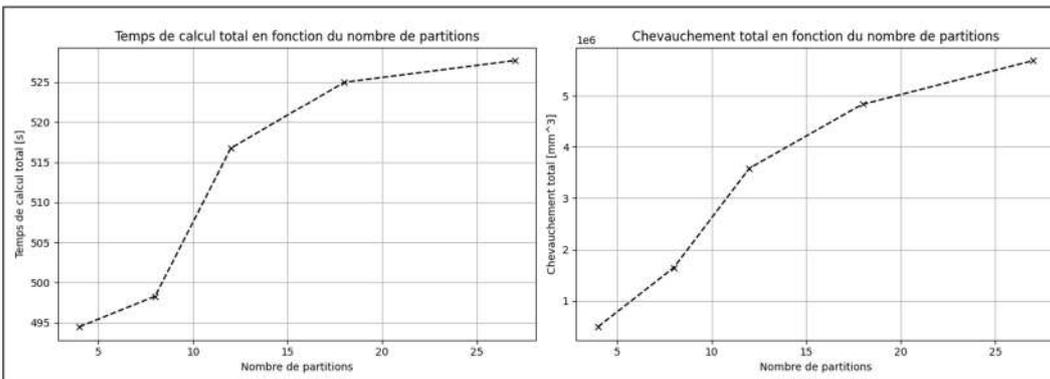
Les courbes de la figure 4.23 représentent l'évolution du temps de calcul de la méthode des K-Means (k-means++) (figure de gauche) ainsi que l'évolution le chevauchement (figure de droite) en fonction du nombre de partitions.



a) Double Vague.



b) Poignée



c) Aube de turbine.

FIGURE 4.23 – Variation du temps de calcul et du chevauchement pour les trois modèles.

• Discussions

À partir des graphes obtenus, nous pouvons constater que l'allure de la courbe du temps de calcul a tendance à suivre celle du chevauchement, et ceci pour les trois complexités. En effet à partir de quatre partitions, nous pouvons constater que la proportionnalité est présente. L'avantage de ce résultat permet d'avoir une idée sur le temps de calcul minimal, et donc éviter de lancer des simulations arbitraire.

### 4.5.2 Validation de l'optimisation de la simulation par partitionnement

La validation de l'optimisation du temps de calcul par le partitionnement, est faite par la comparaison des temps de calcul obtenus via le partitionnement par Cellules et par K-Means. Dans ce contexte, le temps de calcul pour la simulation représente la sommation du temps d'exécution de l'algorithme d'intersection Segments-Sphères avec l'algorithme de partitionnement. La variante "k-means++" sera utilisé pour la validation, avec comme critère d'arrêt une stabilité dans l'ordre de  $10^{-6}$ .

TABLE 4.5 – Résultats de partitionnement pour la simulation d'un modèle de simple complexité.

Nombre de partitions	1	2	4	6	9	12	16	25
Temps total Cellules [s]	238.40	134.97	75.17	53.10	44.30	46.90	50.60	68.1
Temps total K-Means [s]	238.40	123.21	60.29	35.02	24.19	20.24	17.86	17.91
Différence [%]	0.0 %	8.7 %	19.8 %	34.0 %	45.4 %	56.8 %	64.7 %	73.0 %

TABLE 4.6 – Résultats de partitionnement pour la simulation d'un modèle de moyenne complexité et de fortes courbures.

Nombre de partitions	1	2	4	8	12	18	27	37
Temps total Cellules [s]	140.4	109.3	64.16	49.3	48.8	67.5	69.7	84.5
Temps total K-Means [s]	140.4	91.6	65.3	53.0	50.5	42.3	40.4	41.47
Différence [%]	0.0 %	16.2 %	(-) 1.8 %	(-) 7.5 %	(-) 3.5 %	37.3 %	42.0 %	50.9 %

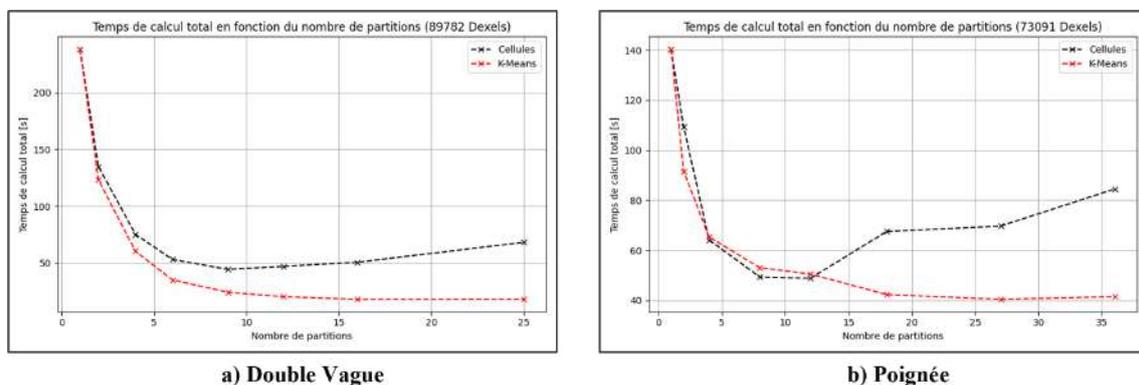


FIGURE 4.24 – Graphes pour la validation de l'optimisation de la simulation.

#### • Discussions

À partir des courbes obtenues, nous pouvons constater que la méthode des K-Means reste toujours meilleure que celle des Cellules dans la partie de simulation et ceci pour les deux types de complexité. En effet, comme le montrent les tableaux 4.5 et 4.6 cette méthode est de 8.7 % jusqu'à 73.0 % plus rapide que la méthode des Cellules, l'optimisation par le partitionnement est d'autant plus efficace plus le nombre de partitions de la pièce augmente.

### 4.5.3 Validation de la prédiction de la topographie de surface

La validation de la prédiction de la topographie se fait par la comparaison des différents paramètres de surfaces 3D obtenus après simulation d'usinage. Deux tests seront effectués, dans un premier temps en faisant varier le pas d'avances de l'outil, puis le rayon de l'outil en lui même.

TABLE 4.7 – Résultats de la prédiction de la topographie de la surface.

	Pas d'avance	Sa	Sq	Sp	Sv	Sz
Rayon de l'outil : 4 mm	6 mm	465.94	645.61	1500	-11.13	1488.87
	4 mm	219.83	346.13	1500	-11.13	1488.87
	2 mm	43.75	58.37	516.63	-11.13	505.50
	Rayon de l'outil	Sa	Sq	Sp	Sv	Sz
Pas d'avance : 4 mm	4mm	219.83	346.13	1500	-11.13	1488.87
	6mm	131.34	210.80	1500	-214.35	1285.65
	8mm	109.15	159.25	1319.61	-214.60	1105.01

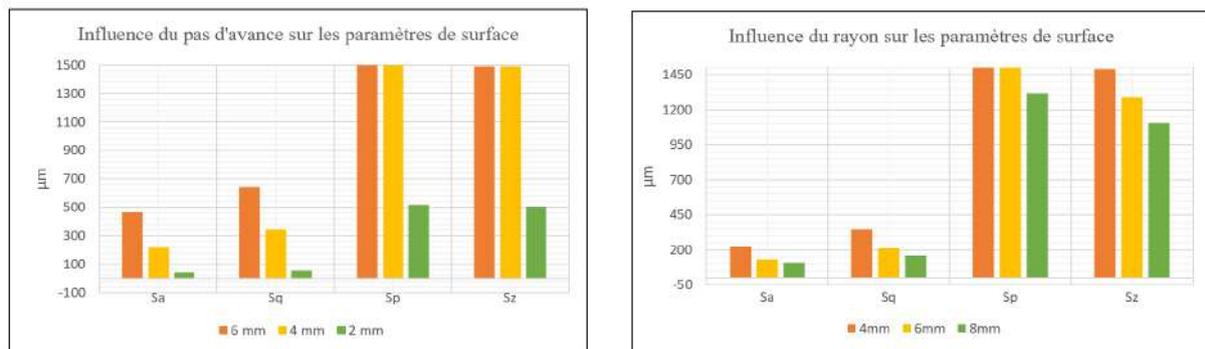


FIGURE 4.25 – Graphes obtenus pour la validation de la prédiction de la topographie.

#### • Discussions

À partir des deux graphes, nous pouvons constater que la variation d'un paramètre d'usinage affecte les paramètres d'état de surface. Comme le démontrent les graphes de la figure 4.25, pour un rayon d'outil constant, le pas d'avance et les paramètres de rugosité sont proportionnels. Pour un pas d'avance constant, le rayon de l'outil est inversement proportionnel avec les paramètres de surfaces. Cela s'explique par la réduction du phénomène de crête, chose qui valide la prédiction qualitative de la topographie de surface. La figure 4.26 illustre les résultats relatifs au respect de tolérance de 0.3 mm.

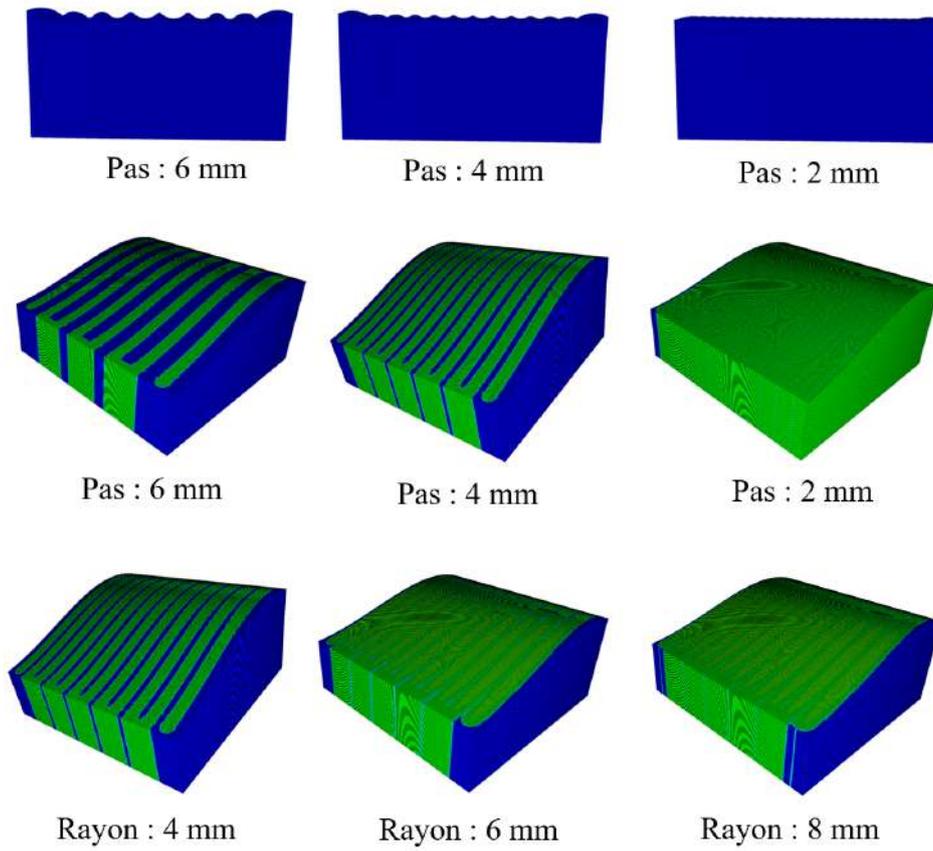


FIGURE 4.26 – Topographie de la surface (IT = 0.3 mm).

#### 4.5.4 Optimisation du partitionnement K-Means basique

Deux nouvelles approches ont été implémentées pour optimiser et améliorer les performances de l'algorithme des K-Means, la première consiste à réduire le nombre d'itérations nécessaire pour la convergence de l'algorithme, la deuxième consiste à optimiser les ressources informatiques pour partitionner des modèles de grandes densités, les performances de ces deux nouvelles approches seront comparées avec l'algorithme des K-Means classique.

##### 4.5.4.1 Comparaison entre les deux méthodes d'initialisation des centres

L'algorithme K-Means basique a été amélioré par l'implémentation d'une nouvelle technique d'initialisation des centres nommée 'k-means++'. Comme décrit dans le Chapitre 2 "*Outils Mathématiques*", il s'agit d'une méthode qui permet de sélectionner les centres de telle sorte que le temps de convergence ainsi que la qualité du regroupement sont grandement améliorés. La figure 4.27 compare les performances des deux initialisations, et ceci pour les trois cas de complexité. Le tableau 4.8 compare le temps de la détermination du  $K_{opt}$ .

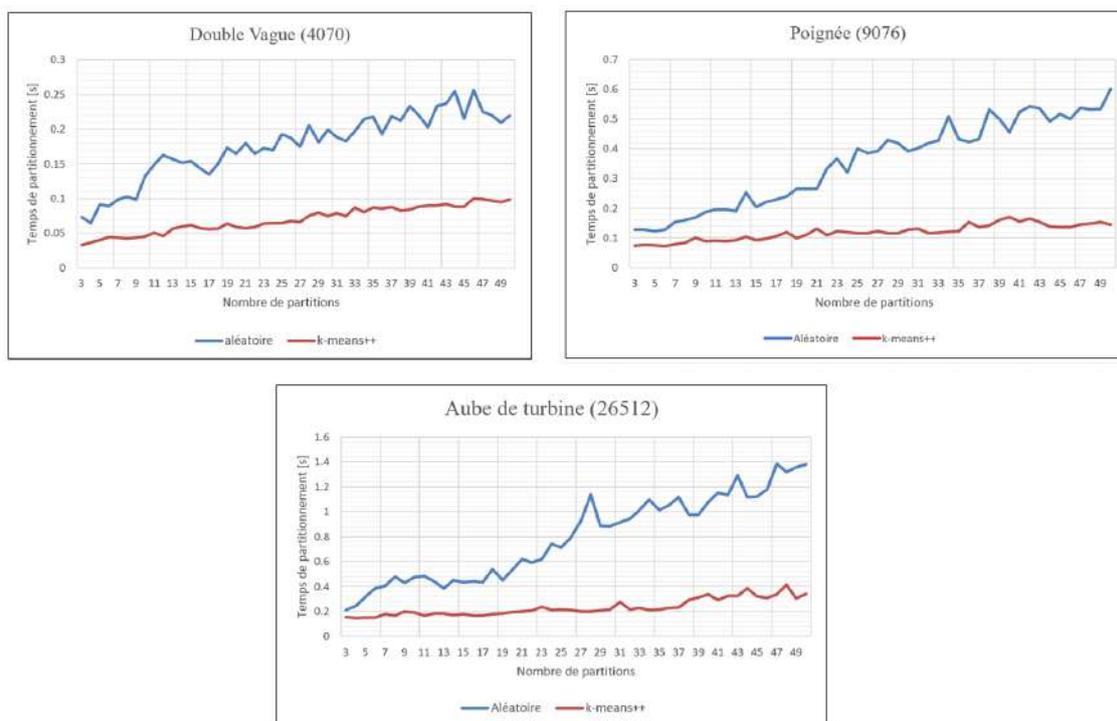


FIGURE 4.27 – Comparaison du temps de partitionnement entre l'initialisation aléatoire et 'k-means++'

TABLE 4.8 – Résultat de comparaison des deux initialisation K-Means.

	Double Vague	Poignée	Aube de turbine
Aléatoire [s]	8.47	17.00	38.09
'k-means++' [s]	3.35	5.70	11.12
Différence [%]	60.4 %	66.5 %	70.8 %

#### • Discussion

À partir des trois graphes obtenus dans la figure 4.27, nous pouvons constater que l'initialisation par la nouvelle approche réduit grandement le temps de partitionnement. En effet, comme le montre le tableau 4.8, cette nouvelle approche permet de réduire de plus de deux fois le temps nécessaire pour déterminer le nombre optimal de partitions.

#### 4.5.4.2 Comparaison entre K-Means et Mini Batch K-Means

L'algorithme K-Means basique a été amélioré davantage par l'implémentation d'une variante de ce dernier, appelée l'algorithme Mini Batch K-Means, il s'agit d'un algorithme qui permet d'optimiser le partitionnement en termes de ressources informatiques, particulièrement efficace pour les modèles de très grande densité. La figure 4.28 compare les performances des deux algorithmes et ceci pour les trois cas de complexité en choisissant l'initialisation 'k-means++'. Le tableau 4.9 compare le temps de la détermination du  $K_{opt}$ .

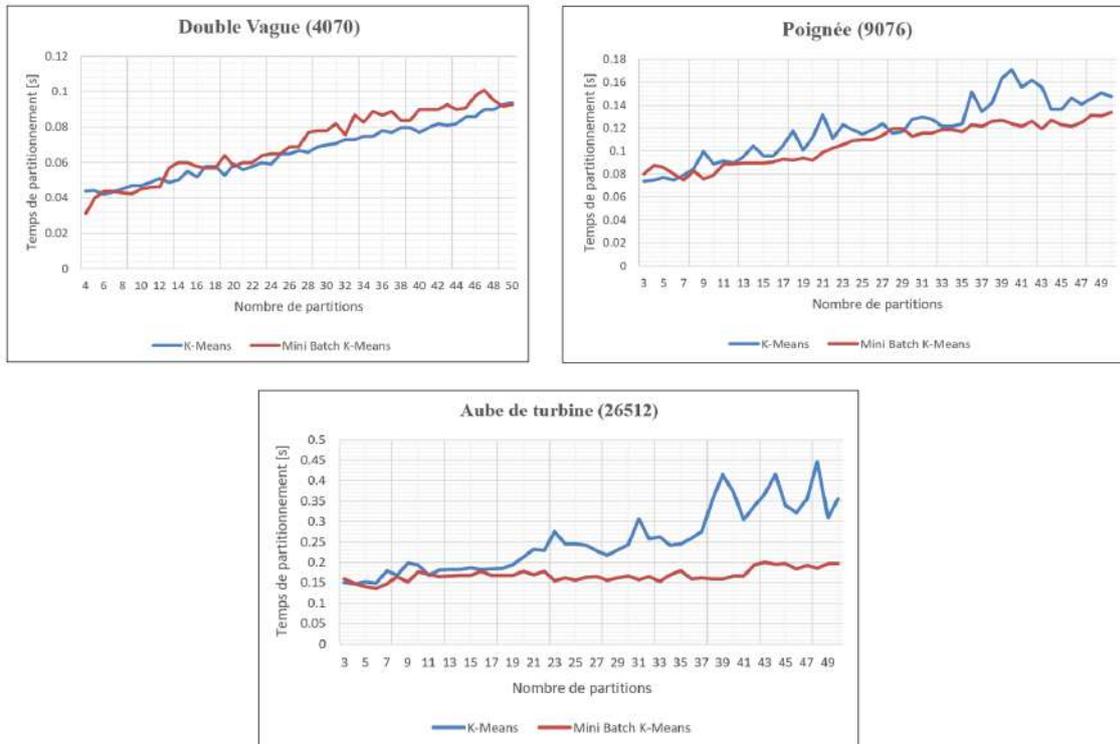


FIGURE 4.28 – Comparaison du temps de partitionnement entre la méthode de K-Means et Mini Batch K-Means

TABLE 4.9 – Résultat de comparaison des deux algorithmes K-Means et Mini Batch K-Means.

	Double Vague	Poignée	Aube de turbine
K-Means [s]	3.14	5.72	12.12
Mini Batch K-Means [s]	4.56	5.12	8.09
Difference [%]	(-) 45.22 %	10.5 %	33.2 %

#### • Discussion

À partir des trois graphes obtenus 4.28, nous pouvons constater que le nouvel algorithme a de meilleures performances plus le nombre d'éléments géométriques à classer est élevé. Cependant son utilisation n'est pas conseillée lorsque le modèle à partitionner est moins dense ( $< 10000$  Triangles).

## 4.6 Détermination du nombre de partitions optimal

La détermination du nombre optimal de partitions  $K_{opt}$ , se fait par l'usage de stratégies qui impliquent d'exécuter l'algorithme K-Means avec différents nombres de groupes, puis de trouver la meilleure combinaison en utilisant des critères qui décrivent la qualité du partitionnement. Le plus souvent un critère de distance est utilisé, tel que le critère de coude ou de silhouette.

Dans notre contexte le critère de chevauchement est utilisé, car il permet d'avoir une idée sur la redondance de l'exécution de l'algorithme d'intersection. Donc la minimisation du chevauchement réduit en théorie le nombre d'opérations répétitives le plus souvent insignifiantes. La figure 4.29 illustre le résultat de la détermination du nombre de partitions optimal pour chaque modèle.

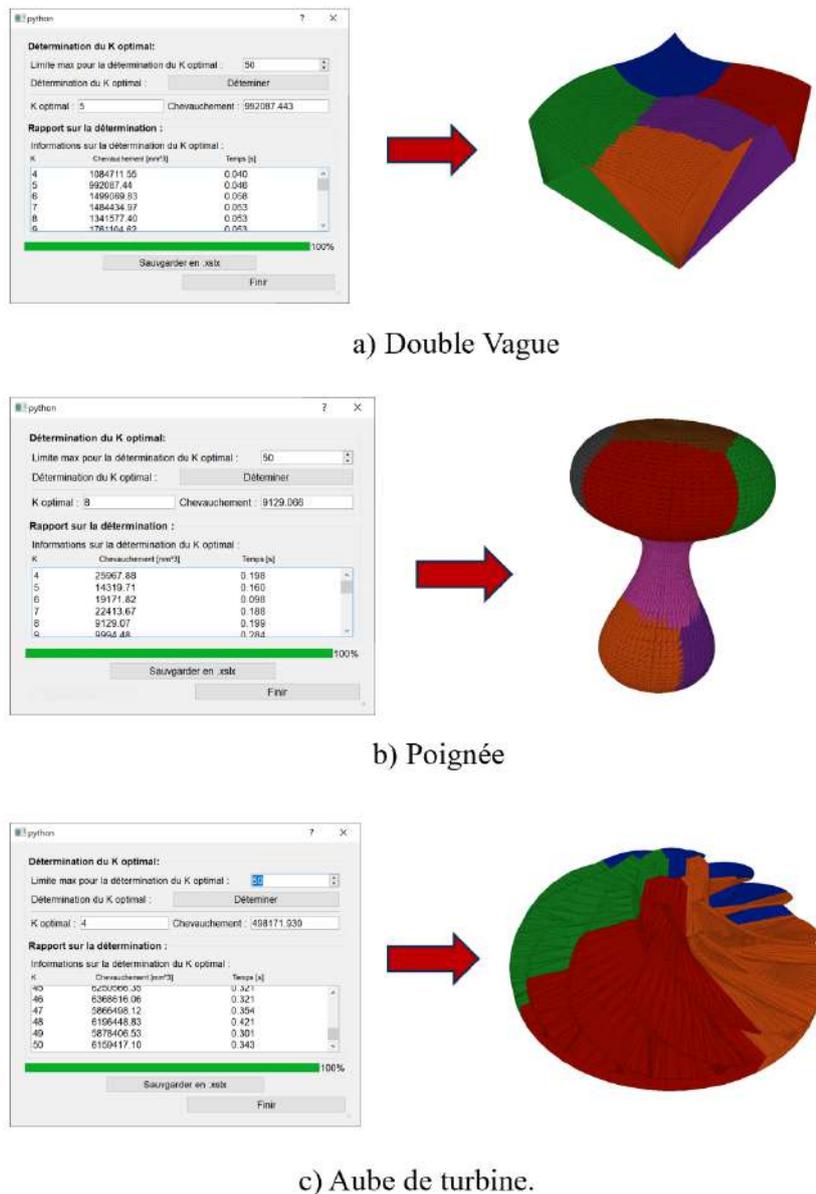


FIGURE 4.29 – Détermination du nombre de partitions optimal.

## 4.7 Conclusion

Dans ce dernier chapitre, nous avons mis en œuvre les méthodes décrites et étudiées dans les précédents chapitres. Nous avons présenté les outils et les langages utilisés, en illustrant les étapes essentielles de notre implémentation par des captures d'écran en décrivant le scénario de fonctionnement général du module logiciel. Enfin, les résultats obtenus ont été discutés et en mettant l'accent sur :

- Validation de l'optimisation du temps de génération par les méthodes de partitionnements des triangles du modèle STL.
- Validation de l'optimisation du temps de simulation par les méthodes de partitionnements des Dixel du modèle TD.
- Validation de la prédiction de la topographie de la surface de la pièce usinée.
- Optimisation du partitionnement K-Means par l'usage de nouvelles approches de classifications intelligentes.

# Conclusion générale

La simulation virtuelle de l'enlèvement de matière représente un outil de vérification important, elle permet de prédire la qualité de la surface avant la production sur la machine-outil. La sélection des paramètres appropriés en termes d'outil et de trajet conduisent à l'obtention d'une meilleure qualité, qui assure la conformité aux normes imposées, et éviter les possibles pertes.

L'objet du présent travail intitulé "Partitionnement intelligent des pièces de forme libre" attribué par l'équipe de recherche CFAO du CDTA, avait pour objectif de proposer une démarche de classification, permettant d'optimiser le temps de calcul pour la simulation d'usinage d'enlèvement de matière. Un module logiciel graphique et interactif a été réalisé pour atteindre cet objectif, donnant la possibilité à l'ingénieur d'introduire des paramètres et d'interpréter les résultats obtenus. Les principales tâches effectuées lors de ce travail sont les suivantes :

- Développement d'un module logiciel permettant la génération d'un modèle volumique et de simuler l'enlèvement de matière.
- Proposition et implémentation d'une nouvelle approche de partitionnement des éléments géométriques du modèle STL et TD.
- Proposition et implémentation d'une approche de détermination du nombre de partitions optimal, par l'usage du critère de chevauchement.
- Optimisation des temps de calculs par l'usage de technique de partitionnement Cellules et K-Means.
- Conversion des fichiers de trajet d'outil d'usinage .CL vers une extension .CE exploitable par les logiciels de simulation.
- Prédiction de la qualité de la surface de la pièce usinée par les paramètres de la norme ISO 25178-2 de 2012 , ainsi que l'intervalle de tolérance.

L'efficacité de la méthode des K-Means a été validée en effectuant une étude comparative entre les deux méthodes en termes de temps de calcul pour différentes complexités du modèle. Les résultats ont montré que cette méthode est en moyenne 25 % plus rapide que la méthode des Cellules. Les temps de calcul pour la génération du modèle volumique ont montré une certaine proportionnalité avec le chevauchement, ce dernier peut représenter une aide à la décision sur le partitionnement optimal, qui permet d'éviter le lancement de simulation répétitive.

La prédiction de la qualité de la surface a été validée en effectuant une comparaison qualitative des différents paramètres des surfaces 3D pour différents trajet et rayon d'outil. Les résultats ont montré que la variation sur les paramètres d'usinage influe sur la qualité de la surface usinée.

L'utilisation de techniques de partitionnement intelligent a montré un grand potentiel pour améliorer la précision et l'efficacité de la simulation d'usinage. En divisant la pièce en régions homogènes, le processus de simulation peut être optimisé pour chaque région. Ce travail peut être étendu dans les directions suivantes :

- La possibilité de standardiser les techniques proposées à tout type de logiciels traitant des modèles géométriques discrets (Elément fini).
- Exploitation des modèles volumiques générés dans d'autres domaines d'applications (Tests de collisions/crash, Terrain virtuelle 3D, etc.).

# Bibliographie

- [1] CLUBTECHNICAL, “Material removal process — types, advantages, limitations, applications.”, [Consulté le 05/2023], Disponible sur : <https://clubtechnical.com/material-removal-process#What-is-a-material-removal-process>.
- [2] LexiqueDeMathématique, “Surface gauche.”, [Consulté le 05/2023], Disponible sur : <https://lexique.netmath.ca/surface-gauche/>.
- [3] K. Bouhadja, “Contribution à l’étude de la topographie 3d des surfaces gauches usinées sur des machines 5 axes,”, Thèse de doctorat : Génie Mécanique, Alger : Ecole Nationale Polytechnique, 2023, 164p.
- [4] D. Salamon, *Curves and Surfaces for Computer Graphics*. Springer, 2006.
- [5] E. CHOICE, “What is milling? - definition, process, and operations.”, [Consulté le 05/2023], Disponible sur : <https://www.engineeringchoice.com/what-is-milling/>.
- [6] W. Tian, W. Gao, W. Chang, and Y. Nie., “Error modeling and sensitivity analysis of a five-axis machine tool,” *Mathematical Problems in Engineering*, 2014.
- [7] I. Girsang and J. Dhupia, “Machine tools for machining,” in *Handbook of Manufacturing Engineering and Technology* (A. Nee, ed.), Springer, 2015.
- [8] A. GROUP, “Quelle est la différence entre le fraisage 3axes, 4 axes et 5 axes.”, [Consulté le 05/2023], Disponible sur : <https://3axis-group.com/fr/quelle-est-la-difference-entre-le-fraisage-3-axes-4-axes-et-5-axes/>.
- [9] E. Duc, “Usinage de formes gauches; contribution à l’amélioration de la qualité des trajectoires d’usinage,”, Thèse de doctorat : , Paris : Ecole Normale Supérieure de Cachan, 1998, 253p.
- [10] MACHINEMFG, “Mastering 5 axis machining : Your ultimate guide.”, [Consulté le 05/2023], Disponible sur : <https://www.machinemfg.com/5-axis-machining/>.
- [11] WAYKEN, “What is 5-axis machining? working principles, types, and benefits.”, [Consulté le 05/2023], Disponible sur : <https://waykenrm.com/blogs/5-axis-cnc-machining/>.
- [12] F. Z. Saad and A. Boubekeur, “Conception et développement d’une approche de prédiction de la topographie 3d des surfaces libres usinées sur machine 05-axes,”, Mémoire de master : systèmes informatiques intelligents, Alger : Université des sciences et de la technologie Houari-Boumediène, 2019, 100p.
- [13] K. Bouhadja and M. Bey, *Survey on Simulation Methods in Multi-axis Machining*. Springer, Janvier 2015.
- [14] K. Bouhadja and M. Bey, “Modélisation volumique des pieces de formes complexes par triple-dexels,” in *Proceeding of the 10ème Journées de Mécanique de l’EMP*, Avril 2016.
- [15] FABBERS, “The stl format.”, [Consulté le 05/2023], Disponible sur : <https://www.fabbers.com/the-stl-format/>.
- [16] A. A. Suzen, Z. Yildiz, K. Kayaalp, and O. Ceylan, “Customized hallux valgus splint design for 3d printer,” *International Journal of 3D Printing Technologies and Digital Industries*, Juillet 2018.
- [17] MEGA VOXELS, “What is a voxel?.”, [Consulté le 05/2023], Disponible sur : <https://www.megavoxels.com/learn/what-is-a-voxel/>.
- [18] F. Schons, D. Hartmann, B. Obst, and G. Glashagen, “Gpu accelerated voxel-based machining simulation,” *The International Journal of Advanced Manufacturing Technology*, Mai 2021.
- [19] 3DSOURCED, “What is cad? definition, advantages and uses explained.”, [Consulté le 05/2023], Disponible sur : <https://www.3dsourced.com/guides/what-is-cad-guide-modeling/>.
- [20] TRIMBLE, “What is cam (computer aided manufacturing)?.”, [Consulté le 05/2023], Disponible sur : <https://constructible.trimble.com/construction-industry/what-is-cam-computer-aided-manufacturing>.

- 
- [21] S. Tichadou, “Modélisation et quantification tridimensionnelles des écarts de fabrication pour la simulation d’usinage.”, Thèse de doctorat : Mécanique. Nantes : Ecole Centrale de Nantes, 2005, 271p.
- [22] D. Prévost, “Modélisation avancée du couple machine outil / commande numérique dans un contexte ugv pour l’optimisation du calcul de trajectoires multi axes.”, Thèse de doctorat : Génie Mécanique, Paris : Ecole Normale Supérieure de Cachan, 2011, 238p.
- [23] P. Benardos and G. C. Vosniakos, “Predicting surface roughness in machining : a review,” *International Journal of Machine Tools and Manufacture*, vol. 43, Juin 2003.
- [24] ISO25178, “Geometrical product specifications – surface texture : Areal.”, International Organization for Standardization, [Consulté le 05/2023], Disponible sur : <https://www.iso.org/obp/ui/#iso:std:iso:25178:-2:ed-1:v1:en>.
- [25] LAROUSSE, “Intelligence artificielle.”, [Consulté le 05/2023], Disponible sur : [https://www.larousse.fr/encyclopedie/divers/intelligence\\_artificielle/187257](https://www.larousse.fr/encyclopedie/divers/intelligence_artificielle/187257).
- [26] MEDIUM, “Artificial intelligence : Its types and implications.”, [Consulté le 05/2023], Disponible sur : <https://medium.com/codex/artificial-intelligence-its-types-and-implications-b95d5a4b9959>.
- [27] IBM, “Ai vs. machine learning vs. deep learning vs. neural networks : What’s the difference?.”, [Consulté le 05/2023], Disponible sur : <https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks>.
- [28] QUEBEC, “Apprentissage automatique.”, [Consulté le 05/2023], Disponible sur : [https://vitrinelinguistique.oqlf.gouv.qc.ca/fiche-gdt/fiche/8395061/apprentissage-automatique?utm\\_campaign=Redirection%20des%20anciens%20outils&utm\\_content=id\\_fiche%3D8395061&utm\\_source=GDT](https://vitrinelinguistique.oqlf.gouv.qc.ca/fiche-gdt/fiche/8395061/apprentissage-automatique?utm_campaign=Redirection%20des%20anciens%20outils&utm_content=id_fiche%3D8395061&utm_source=GDT).
- [29] COURSERA, “Machine learning specialization.”, [Consulté le 05/2023], Disponible sur : <https://www.coursera.org/specializations/machine-learning-introduction#courses>.
- [30] ANALYSTANSWERS, “Data cluster : Definition, example, & cluster analysis.”, [Consulté le 05/2023], Disponible sur : <https://analystanswers.com/data-cluster-definition-example-cluster-analysis/>.
- [31] SIMPLILEARN, “K-means clustering algorithm : Applications, types, & how does it work?.”, [Consulté le 04/2023], Disponible sur : <https://www.simplilearn.com/tutorials/machine-learning-tutorial/k-means-clustering-algorithm>.
- [32] TOWARDSMACHINELEARNING, “K-means algorithm and how it works.”, [Consulté le 05/2023], Disponible sur : <https://towardsmachinelearning.org/k-means/>.
- [33] TOWARDSDATASCIENCE, “Silhouette method — better than elbow method to find optimal clusters.”, [Consulté le 05/2023], Disponible sur : <https://towardsdatascience.com/silhouette-method-better-than-elbow-method-to-find-optimal-clusters-378d62ff6891>.
- [34] S. C. Hicks, R. Liu, Y. Ni, E. Purdom, and D. Risso, “mbkmeans : Fast clustering for single cell data using mini-batch k-means,” *PLoS Comput Biol*, vol. 17, Janvier 2021.
- [35] THECLEVERPROGRAMMER, “Mini-batch k-means clustering.”, [Consulté le 05/2023], Disponible sur : <https://thecleverprogrammer.com/2021/09/10/mini-batch-k-means-clustering-in-machine-learning/>.
- [36] J. Bejar, “K-means vs mini batch k-means : A comparison,” 2013.
- [37] G. Zaouaoui and L. Zaoui, “Proposition d’une solution au problème d’initialisation cas du k-means,” in *Conférence Internationale sur l’Informatique et ses Applications*, (Saida, Algerie), Mai 2009.
- [38] D. Arthur and S. Vassilvitskii, “k-means++ : The advantages of careful seeding,” in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Janvier 2007.
- [39] J. J. Jiménez, R. J. Segura, and F. R. Feito, “A robust segment/triangle intersection algorithm for interference tests : Efficiency study,” *Computers & Graphics*, vol. 33, no. 4, pp. 438–449, 2009.
- [40] C. Ericson, *Real-Time Collision Detection*. San Francisco : Morgan Kaufmann, 2005.

# Annexes

## A. Intersection Triangle-Segment

Une ligne (Ray)  $\vec{R}_{(t)}$  est défini par un point d'origine  $\vec{O}$  et une direction normalisée  $\vec{D}$ , exprimé par :

$$\vec{R}_{(t)} = \vec{O} + t \cdot \vec{D}$$

À présent, la représentation d'un segment est basée sur deux points, le point début du segment  $\vec{Q}_1$  et le point fin  $\vec{Q}_2$ , exprimé par :

$$\vec{R}(t) = \vec{Q}_1 + t \cdot (\vec{Q}_2 - \vec{Q}_1) \quad (4.1)$$

De même, un triangle est défini par ses sommets  $\vec{V}_1$ ,  $\vec{V}_2$  et  $\vec{V}_3$ . En utilisant les coordonnées barycentriques d'un point par rapport à un triangle, si un point  $S(u, v)$  appartient à ce triangle en vérifiant les conditions suivantes :

$$\begin{aligned} u &> 0 \\ v &> 0 \\ u + v &< 1 \end{aligned}$$

Le point  $S(u, v)$  est défini par la formule :

$$S(u, v) = (1 - u - v) \cdot \vec{V}_1 + u \cdot \vec{V}_2 + v \cdot \vec{V}_3 \quad (4.2)$$

En comparant l'équation du segment  $\vec{R}(t)$  avec celle du point de triangle  $\vec{S}(u, v)$ , le point d'intersection de ces deux éléments est calculé en résolvant l'équation :

$$\vec{Q}_1 + t \cdot (\vec{Q}_2 - \vec{Q}_1) = (1 - u - v) \cdot \vec{V}_1 + u \cdot \vec{V}_2 + v \cdot \vec{V}_3$$

Après résolution de l'équation, les paramètres  $t$ ,  $u$  et  $v$  sont obtenus par :

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{(\vec{D} \times \vec{E}_2) \cdot \vec{E}_1} \begin{bmatrix} (\vec{T} \times \vec{E}_1) \cdot \vec{E}_2 \\ (\vec{D} \times \vec{E}_2) \cdot \vec{T} \\ (\vec{T} \times \vec{E}_1) \cdot \vec{D} \end{bmatrix} \quad (4.3)$$

Ou les vecteurs  $D$ ,  $E_1$ ,  $E_2$  et  $T$  sont définis par :

- $\vec{D} = \vec{Q}_2 - \vec{Q}_1$
- $\vec{E}_1 = \vec{V}_2 - \vec{V}_1$
- $\vec{E}_2 = \vec{V}_3 - \vec{V}_1$
- $\vec{T} = \vec{Q}_1 - \vec{V}_1$

## B. Intersection Sphère-Segment

Une ligne (Ray)  $\vec{R}(t)$  est défini par un point d'origine  $\vec{O}$  et une direction normalisée  $\vec{D}$ , exprimé par :

$$\vec{R}(t) = \vec{O} + t \cdot \vec{D}$$

À présent, la représentation d'un segment est basée sur deux points, le point début du segment  $\vec{Q}_1$  et le point fin  $\vec{Q}_2$ , exprimé par :

$$\vec{R}(t) = \vec{Q}_1 + t \cdot (\vec{Q}_2 - \vec{Q}_1) \quad (4.4)$$

À présent, la représentation géométrique d'une sphère se fait par le vecteur de son centre  $\vec{C}$  et de son rayon  $R$ , ou  $\vec{X}$  est un point qui appartient a la sphère :

$$\|\vec{X} - \vec{C}\|^2 = R^2 \quad (4.5)$$

L'intersection désigne désigne l'existante dans point appartenant à la sphère et au segment en même temps. En remplaçant l'équation du segment  $\vec{R}(t)$  dans l'équation de la sphère, les points d'intersection peuvent être déterminés en résolvant l'équation suivante :

$$\|\vec{Q}_1 + t \cdot (\vec{Q}_2 - \vec{Q}_1) - \vec{C}\|^2 = R^2 \quad (4.6)$$

$$(\vec{Q}_1 + t \cdot (\vec{Q}_2 - \vec{Q}_1) - \vec{C}) \cdot (\vec{Q}_1 + t \cdot (\vec{Q}_2 - \vec{Q}_1) - \vec{C}) = R^2$$

$$(\vec{Q}_2 - \vec{Q}_1) \cdot (\vec{Q}_2 - \vec{Q}_1) t^2 + 2 (\vec{Q}_2 - \vec{Q}_1) \cdot (\vec{Q}_1 - \vec{C}) t + (\vec{Q}_1 - \vec{C}) \cdot (\vec{Q}_1 - \vec{C}) - R^2 = 0 \quad (4.7)$$

L'équation peut être développée pour obtenir une forme quadratique :

$$at^2 + bt + c = 0 \quad (4.8)$$

Avec :

- $a = (\vec{Q}_2 - \vec{Q}_1) \cdot (\vec{Q}_2 - \vec{Q}_1)$
- $b = 2 (\vec{Q}_2 - \vec{Q}_1) \cdot (\vec{Q}_1 - \vec{C})$
- $c = (\vec{Q}_1 - \vec{C}) \cdot (\vec{Q}_1 - \vec{C}) - R^2$

Trois cas possibles se présentent selon le déterminant  $\Delta = b^2 - 4 a \times c$ , de l'équation quadratique :

- $\Delta < 0$  : Il n'existe aucun point d'intersection entre le segment et la sphère.
- $\Delta = 0$  : Le segment intersecte la sphère dans un point double,  $t$ .
- $\Delta > 0$  : Le segment intersecte la sphère dans deux points distincts,  $t_1$  et  $t_2$ .