

# Table des matières

<b>Introduction</b>	<b>9</b>
<b>1 Généralités et prérequis</b>	<b>11</b>
1.1 Généralités sur l'usinage en bout des surfaces gauches . . . . .	11
1.1.1 L'usinage en bout . . . . .	11
1.1.2 Les surfaces gauches . . . . .	12
1.2 Types d'outils . . . . .	14
1.3 Stratégies d'usinage . . . . .	17
1.4 Éléments de statistique . . . . .	22
1.4.1 Moyenne et médiane . . . . .	22
1.4.2 Variance et écart-type . . . . .	23
1.4.3 Covariance et corrélation . . . . .	24
1.4.4 Analyse en composantes principales . . . . .	25
1.5 Optimisation de boite noire et sans dérivées . . . . .	27
1.6 Calcul des temps et longueur d'usinage . . . . .	28
1.6.1 Longueur de trajectoire . . . . .	29
1.6.2 Temps d'usinage . . . . .	29
<b>2 Approximation des surfaces gauches</b>	<b>33</b>
2.1 Introduction et motivation . . . . .	33
2.2 Étude des surfaces planes . . . . .	34
2.2.1 Problématique de choix d'outil . . . . .	35
2.2.2 Développement analytique . . . . .	36
2.2.2.1 Outil torique selon la direction de plus grande pente . . . . .	37
2.2.2.2 Outil sphérique selon la direction principale . . . . .	38
2.2.2.3 Valeurs critiques des paramètres facteur de forme et pente . . . . .	39
2.2.3 Validation numérique . . . . .	41
2.2.3.1 Considérations préliminaires . . . . .	42
2.2.3.2 Validation de l'approximation analytique du temps d'usinage . . . . .	43

2.2.3.3	Validation des paramètres critiques . . . . .	44
2.3	Approximation des surfaces gauches . . . . .	45
2.3.1	Considérations préliminaires . . . . .	45
2.3.2	Utilisation de la matrice de covariance pour définir la direction principale . . . . .	46
2.3.3	Utilisation des valeurs propres pour définir un facteur de forme approximatif d'une surface gauche . . . . .	47
2.3.4	Pente approximée d'une surface gauche . . . . .	47
2.4	Application au choix de l'outil . . . . .	48
2.4.1	Description de la procédure du choix d'outil . . . . .	48
2.4.2	Application à des cas tests . . . . .	49
2.4.2.1	Cas test 1 . . . . .	49
2.4.2.2	Cas test 2 . . . . .	51
2.5	Discussion . . . . .	53
2.5.1	Sur l'étude analytique des surfaces planes . . . . .	53
2.5.2	Sur l'approximation des surfaces gauches par l'ACP . . . . .	54
2.5.3	Sur la méthodologie d'aide au choix d'outil . . . . .	55
2.5.3.1	De l'efficacité de la méthode proposée . . . . .	55
2.5.3.2	Sur la mise en œuvre de la méthode . . . . .	56
2.5.3.3	De l'influence des paramètres auxiliaires . . . . .	56
2.6	Conclusion . . . . .	58
<b>3</b>	<b>Partitionnement de surfaces gauches pour l'usinage</b>	<b>61</b>
3.1	Motivation et intérêt . . . . .	61
3.2	État de l'art . . . . .	62
3.3	Discrétisation et création du data-set . . . . .	64
3.3.1	Maillage et <i>sample-points</i> . . . . .	64
3.3.2	<i>Feature-vector</i> et <i>data-set</i> . . . . .	64
3.4	Algorithmes de <i>clustering</i> . . . . .	66
3.4.1	L'algorithme <i>K-means</i> . . . . .	67
3.4.2	L'algorithme RPCL . . . . .	68
3.4.3	La classification hiérarchique ascendante . . . . .	69
3.5	Métriques classiques . . . . .	69
3.5.1	Distance euclidienne . . . . .	70
3.5.2	Distance de Mahalanobis . . . . .	70
3.5.3	Distance basée sur les écarts-types . . . . .	71
3.5.4	Distance de Ward . . . . .	71
3.6	Métrique basée sur l'ACP . . . . .	72
3.6.1	Considérations préliminaires . . . . .	72

3.6.2	Définition de la métrique . . . . .	73
3.6.3	Quelques remarques . . . . .	74
3.7	Détails d'implémentation . . . . .	75
3.7.1	Partition initiale . . . . .	75
3.7.2	Calcul incrémental de la matrice de covariance . . . . .	75
3.7.3	Critères d'arrêt . . . . .	76
3.7.4	Recherche de composantes connexes . . . . .	77
3.8	Comparaison des résultats . . . . .	79
3.8.1	Protocole de tests . . . . .	79
3.8.2	Analyse des résultats . . . . .	80
3.9	Discussion . . . . .	83
3.9.1	Avantages et limites . . . . .	83
3.9.2	Comparaison avec d'autres travaux . . . . .	85
3.10	Conclusions et perspectives . . . . .	87
<b>4</b>	<b>Optimisation de l'usinage de surfaces gauches</b>	<b>89</b>
4.1	Introduction . . . . .	89
4.2	Formulation du problème d'optimisation . . . . .	90
4.3	Fonctions <i>surrogates</i> . . . . .	93
4.4	Résultats . . . . .	99
4.4.1	Modèle 3-axes . . . . .	100
4.4.2	Modèle 3+2-axes . . . . .	102
4.5	Heuristique 3+2-axes . . . . .	104
4.5.1	Considérations générales . . . . .	104
4.5.2	Définition de l'heuristique basée sur l'ACP . . . . .	105
4.5.3	Remarques sur l'optimalité de l'heuristique . . . . .	108
4.5.4	Résultats de l'heuristique . . . . .	111
4.6	Discussion et comparaisons . . . . .	112
4.6.1	Discussion des résultats obtenus . . . . .	113
4.6.2	Comparaison avec d'autres articles . . . . .	114
4.7	Conclusion et perspectives . . . . .	115
<b>Conclusion générale et perspectives</b>	<b>117</b>	
<b>Bibliographie</b>	<b>119</b>	
<b>Annexes</b>	<b>125</b>	
A	Étude du paramètre critique $\mu_c = f(s)$ . . . . .	127
A.1	Expression analytique de $\mu_c$ . . . . .	127

## Table des matières

---

A.2	Analyse numérique du paramètre critique $\mu_c$	128
B	Preuve que l'approximation du facteur de forme par les valeurs propres est exacte pour un plan	131
C	Preuve de la décroissance des fonctions critiques	133
D	Preuve des formules utilisées dans le calcul incrémental de la matrice de covariance	135
D.1	Cas d'ajout d'une maille	135
D.2	Cas de retrait d'une maille	136
E	Calcul analytique du temps d'usinage pour la surrogate 3	139
<b>Table des figures</b>		<b>143</b>

# Introduction

Les surfaces gauches sont utilisées actuellement dans divers domaines d'activité tels que la définition de moules ou de matrices d'emboutissage. Vu la complexité de la géométrie des surfaces utilisées pour ce type d'applications, ces surfaces exigent un niveau de qualité élevé et des défauts de formes réduits. Ceci rend leur usinage long, coûteux et est peu optimisé à ce jour.

Bien que plusieurs représentations mathématiques des surfaces existent, la représentation paramétrique est utilisée la plupart du temps, pour sa commodité et sa généralité. Pour cela, plusieurs modèles mathématiques ont été développés depuis le siècle dernier, à savoir les surfaces de Bézier, les B-splines et les surfaces NURBS. Ces modèles permettent de décrire des surfaces, tel que la sphère, que la représentation algébrique ne permet pas.

Les avancées technologiques réalisées durant le siècle dernier, dans le domaine de la Conception et Fabrication Assistées par Ordinateur (CFAO) permettent aujourd'hui de modéliser et d'usiner de telles surfaces à l'aide de Machine-Outil à Commande Numérique (MOCN). Ces machines permettent d'enlever la matière à l'aide d'un outil qui peut parcourir la surface en se déplaçant selon trois axes de translation, ce qui lui permet d'accéder de façon unique à n'importe quel point de la surface. Cette opération définit l'usinage en 3-axes. En plus des translations, certaines machines offrent la possibilité d'un déplacement relatif de l'outil par rapport à la surface selon 2 axes de rotations, ce qui permet à l'outil d'accéder à n'importe point de la surface par une infinité de façons. Cette opération définit l'usinage en 5-axes.

Parmi les avantages de l'usinage 5-axes on peut mentionner une visibilité accrue de la surface pièce par l'outil. La visibilité d'une pièce est définie par la capacité de l'outil à accéder à cette pièce dans une configuration pièce-machine donnée. La configuration pièce-machine inclut les positions relatives de la pièce, du bâti de la machine, du montage d'usinage, de l'outil et du porte-outil. L'étude de la visibilité a pour but d'éviter l'interférence globale ; c'est à dire l'interférence entre les divers éléments de la configuration d'usinage hors de la proximité immédiate du point de contact outil-pièce. Par opposition, l'interférence locale correspond à l'interférence entre l'outil et la pièce à proximité du point de contact outil-pièce.

En dépit de ses avantages, l'usinage 5-axes comporte des limitations cinématiques et dynamiques causées par les axes de rotations, qui, lorsqu'ils travaillent simultanément avec les axes de translation, diminuent la rigidité de ces derniers. De plus, l'outil risque de ne pas atteindre sa vitesse de consigne

dans ce cas.

Les MOCN offrent différents modes d’usinage. Le mode d’usinage est défini par la partie de l’outil qui est en contact avec la pièce à usiner.

- L’usinage en bout consiste à usiner avec l’extrémité de l’outil.
- L’usinage en roulant consiste à usiner avec le corps de l’outil.
- L’usinage combiné consiste à usiner en même temps avec le corps et l’extrémité de l’outil.

Le mode utilisé dans cette étude est l’usinage en bout. De plus, seule la phase de finition est étudiée, car elle est la phase la plus critique du processus d’usinage et exige une grande précision afin de produire une surface avec une erreur géométrique maîtrisée.

Afin de produire la surface, il est nécessaire de planifier le parcours d’outil, avec comme objectif de contrôler l’erreur géométrique tout en minimisant le temps d’usinage. L’erreur géométrique ne peut pas être complètement annulée sur les surfaces gauches, à part certaines surfaces prismatiques et de révolution. Il est aussi important que le temps de calcul nécessaire pour planifier le parcours d’outil soit maîtrisé. Le critère de temps de calcul peut revêtir une importance aussi capitale que la minimisation du temps d’usinage. En effet, les méthodes d’optimisation proposées dans cette étude nécessitent un nombre assez conséquent d’évaluations du processus d’usinage ce qui peut rendre le temps de la planification de parcours d’outil très grand. De plus, les systèmes de CFAO doivent aujourd’hui permettre à l’utilisateur de travailler interactivement; ce qui implique une réponse en temps réel ou quasi-réel des algorithmes utilisés. Nous nous sommes donc placés dans ce cadre pour tous nos développements.

Dans un premier chapitre, les prérequis à connaître sur le problème d’usinage de surfaces gauches sont présentés, ainsi que les notions de bases nécessaires à la compréhension de la suite de ce travail. Le second chapitre est consacrée à une nouvelle méthodologie d’approximation des surfaces gauches par des plans, à l’aide de l’Analyse en Composantes Principales (ACP). Diverses utilisations de cette approximation seront utilisés tout au long de nos travaux. Dans le troisième chapitre, le partitionnement des surfaces gauches est traité. Une nouvelle métrique est introduite permettant de prendre en compte le comportement cinématique de la machine. Dans le dernier chapitre, l’usinage de surfaces gauches est formulé comme un problème d’optimisation où on cherche à minimiser le temps d’usinage sous contrainte de hauteur de crête. Une formulation originale intégrant les étapes de partitionnement et d’usinage est proposée. Le problème d’optimisation ainsi obtenu est de type boîte noire et résolu par le logiciel NOMAD [2]. Une approche heuristique combinant ACP et optimisation est également proposée, cette approche permet d’obtenir des gains considérables par rapport aux résultats que nous avons obtenus jusqu’ici et ceux existant dans la littérature.

# Chapitre 1

## Généralités et prérequis

Dans ce premier chapitre, les prérequis à connaître sur le problème d’usinage de surfaces gauches sont présentés, ainsi que les notions de bases nécessaires à la compréhension de la suite de ce travail.

### 1.1 Généralités sur l’usinage en bout des surfaces gauches

La première partie de cette section décrit l’usinage en bout alors que la seconde partie traite des surfaces gauches.

#### 1.1.1 L’usinage en bout

L’usinage entre dans la gamme de fabrication des pièces mécaniques. Il consiste à réaliser des pièces par enlèvement de matière en respectant l’état de surface, les tolérances de formes et la géométrie spécifiée par le dessin de définition. À chaque phase de la gamme de fabrication, le bureau des méthodes choisit la machine, le type d’usinage à réaliser, l’outil ainsi que le support de pièce permettant l’obtention de tous les éléments de cotation de la surface considérée. Il existe principalement deux types d’opérations d’usinage : le tournage et le fraisage. Le tournage consiste à usiner une surface avec un mouvement de coupe obtenu par rotation de la pièce serrée entre les mors d’un mandrin et un mouvement d’avance obtenu par le déplacement de l’outil coupant. Cette opération est utilisée pour usiner des surfaces de révolution. Le fraisage résulte quant à lui d’un mouvement de rotation de l’outil de coupe d’une part, et de l’avance de la pièce à usiner d’autre part.

Le fraisage présente plusieurs avantages tels qu’un rendement élevé, un bon fini de surface, une haute précision et une grande souplesse au niveau de la génération de différentes formes. Trois types de fraisages peuvent être distingués : le fraisage en bout, le fraisage en roulant et le fraisage combiné. Le fraisage en bout (figure 1.1) est le plus répandu en industrie, car il permet de réaliser des surfaces complexes obtenues par un travail d’enveloppe généré par la rotation conjuguée au déplacement de l’outil. Dans ce procédé d’usinage, l’outil est tangent à la surface à usiner au point de contact. La partie

de l'outil jouant un rôle important dans ce type de fraisage est le bout de l'outil. En effet, selon la forme de l'outil (torique, sphérique, plat — voir figure 1.5, page 16), la trace laissée sur la surface et la quantité de matière enlevée diffèrent. L'aspect visuel d'une surface usinée en fraisage en bout est caractérisé par une série de surfaces sécantes correspondant aux traces laissées par les dents de la fraise sur la pièce.

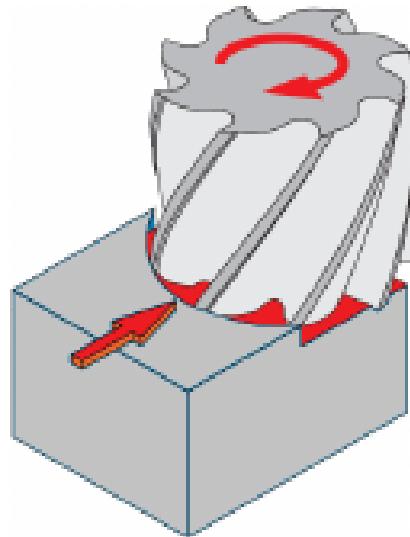


Figure 1.1 – Principe du fraisage en bout

Le fraisage en bout peut être réalisé sur des machines-outils à commande numérique (MOCN) 3-axes ou 5-axes. Pour les machines 3-axes, l'outil se déplace suivant trois axes orthogonaux **X**, **Y** et **Z**, (figure 1.2) le positionnement de l'outil pour accéder à un point de la surface est unique. Ce positionnement peut être défini par un seul point appelé point centre-outil [64]. L'usinage avec des MOCN 5-axes offre plus de liberté de positionnement de l'outil sur la surface (figure 1.2). Il existe une infinité de solutions pour amener l'outil tangent à la surface au point de contact.

### 1.1.2 Les surfaces gauches

Il existe plusieurs définitions pour les surfaces gauches ; nous choisissons celle qui consiste à définir une surface gauche comme toute surface non-développable. Pour définir les surfaces développables, il faut d'abord définir les surfaces réglées : Par chaque point d'une surface réglée passe une droite, appelée génératrice, contenue dans la surface. La figure 1.3 illustre une surface réglée.

Soit  $C_1$  et  $C_2$  deux courbes paramétriques,  $f$  une fonction de  $[0, 1]$  dans  $[0, 1]$ , l'équation 1.1 donne la représentation paramétrique d'une surface réglée quelconque.

$$\forall (u, v) \in [0, 1]^2, \quad S(u, v) = (1 - v)C_1(u) + vC_2(u + f(u)) \quad (1.1)$$

Une surface réglée telle que son plan tangent reste le même le long d'une génératrice est dite développable. En pratique une surface développable peut être aplatie sans modifier la distance géodésique [13]

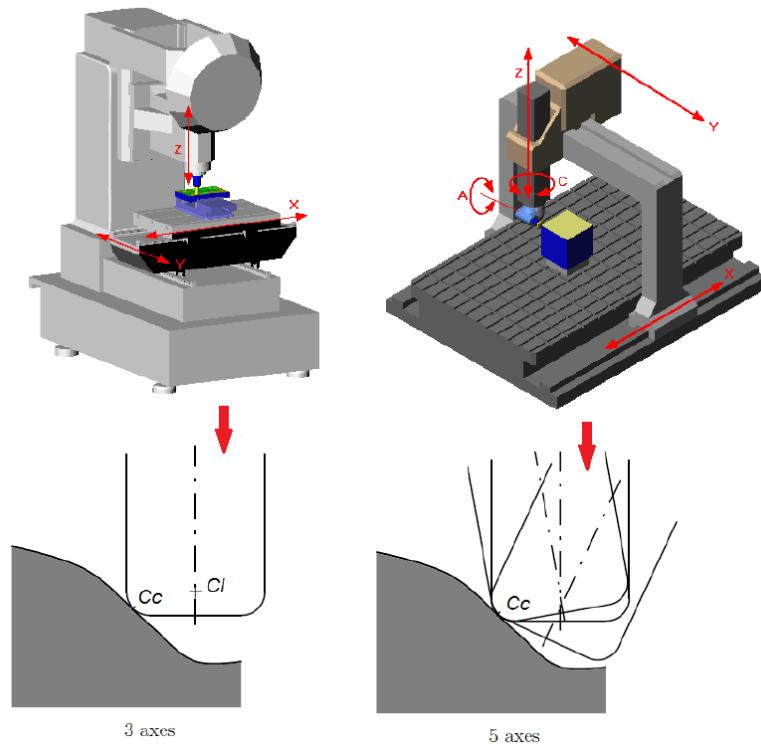


Figure 1.2 – Fraisage en bout en 3-axes et en 5-axes

entre aucune paire de points. Dans ce sens les surfaces gauches sont les surfaces qu’on ne peut pas aplatis sans modifier la distance entre au moins deux points.

Dans l’industrie, les surfaces gauches sont souvent des surfaces à forte valeur ajoutée (moules et matrices d’emboutissage). Ces surfaces peuvent être définies de plusieurs façons, par une propriété géométrique ou par une relation analytique entre les coordonnées d’un point [30] :

- Surfaces algébriques (explicite) :  $z = f(x, y)$
- Surfaces algébriques (implicite) :  $f(x, y, z) = 0$
- Surfaces paramétriques :  $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f(u, v) \\ g(u, v) \\ h(u, v) \end{pmatrix}$
- Surfaces en coordonnées semi-polaires :  $z = f(\theta, \rho)$  avec  $\theta$  et  $\rho$  l’angle polaire et le rayon-vecteur.

En CFAO (Conception et Fabrication Assistée par Ordinateur), la forme paramétrique est majoritairement utilisée et les surfaces à usiner sont modélisées avec des carreaux de Bézier (équation 1.2) ou des NURBS (*Non Uniform Rational B-Spline*).

Un carreau de Bézier est défini à l’aide d’un réseau de points de contrôle  $(P_{ij})_{0 \leq i \leq n, 0 \leq j \leq m}$  de taille  $(n+1) \times (m+1)$  :

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m \binom{n}{i} u^i (1-u)^{n-i} \binom{m}{j} v^j (1-v)^{m-j} P_{ij} \quad (1.2)$$

Un carreau NURBS (équation 1.4) est défini à partir de fonctions de base (équations 1.3) définies par

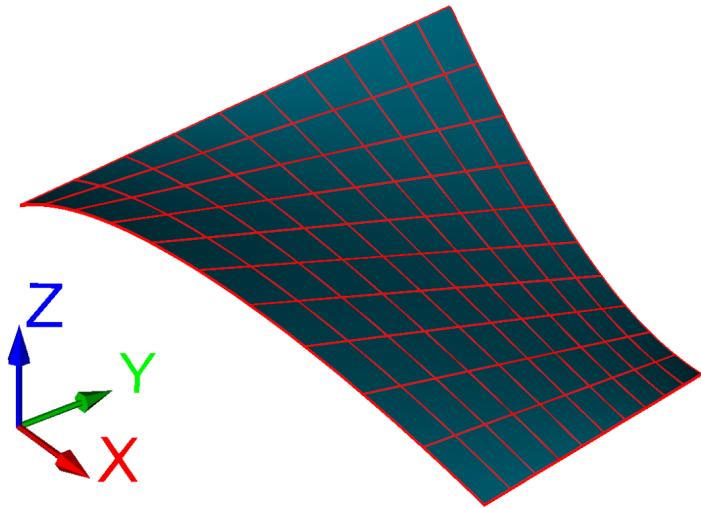


Figure 1.3 – Exemple de surface réglée

récurrence, de degré  $n_u$  et  $n_v$ , un nombre de nœuds  $m_u$  (respectivement  $m_v$ ) en  $u$  (respectivement  $v$ ) un réseau de points de contrôle ( $P_{ij}$ ) et leurs poids associés ( $\omega_{ij}$ ).

$$\begin{cases} N_j^0(t) = \begin{cases} 1, & \text{si } t_j \leq t \leq t_{j+1} \\ 0 & \text{sinon} \end{cases} \\ N_j^k(t) = \frac{t-t_j}{t_{j+k}-t_j} N_j^{k-1}(t) + \frac{t_{j+k+1}-t}{t_{j+k+1}-t_{j+1}} N_{j+1}^{k-1}(t) \end{cases} \quad (1.3)$$

$$S(u, v) = \frac{\sum_{i=0}^{m_u-n_u-1} \sum_{j=0}^{m_v-n_v-1} N_i^{n_u}(u) N_j^{n_v}(v) \omega_{ij} P_{ij}}{\sum_{i=0}^{m_u-n_u-1} \sum_{j=0}^{m_v-n_v-1} N_i^{n_u}(u) N_j^{n_v}(v) \omega_{ij}} \quad (1.4)$$

## 1.2 Types d'outils

Les fraises les plus utilisées pour l'usinage de surfaces gauches sont des fraises à rayon. La figure 1.4 montre un paramétrage, proposé par [51], qui décrit dans le cas général la géométrie des outils. En fraisage en bout, on distingue trois outils principaux, illustrés dans la figure 1.5, à savoir :

- l'outil plat, moins utilisé en raison des traces nettes qu'il laisse sur la surface de la pièce.
- l'outil sphérique, ou boule, qui est le plus utilisé dans l'industrie, car il permet de calculer facilement les trajectoires d'outils.
- l'outil torique, dont il a été prouvé qu'il pouvait donner de meilleurs résultats que l'outil sphérique, bien qu'il soit beaucoup plus difficile à manier dans les calculs [12].

Pour un outil donné, le rayon effectif est défini comme le rayon de la courbe définissant la trace laissée par l'outil dans la matière, au point de contact. Le rayon effectif est un paramètre très important, car il a une grande influence sur la performance de l'usinage. En effet, il est intéressant de maximiser (dans la

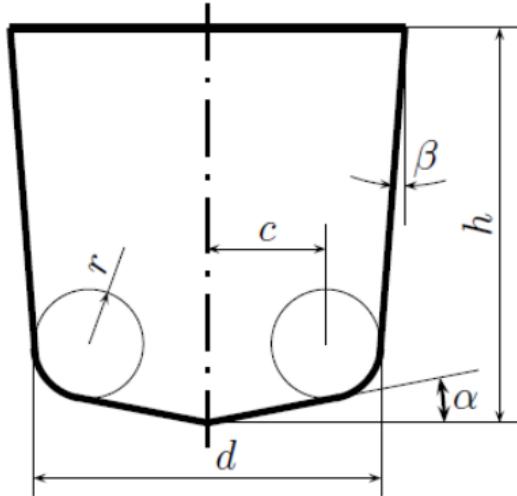


Figure 1.4 – Paramétrage géométrique des outils

mesure du possible) le rayon effectif en chaque point de la surface, ceci résultera en des passes (passage de l'outil sans être relevé) de plus en plus espacées, ce qui réduit la longueur de la trajectoire d'usinage. Cette notion a été initialement introduite dans [72]. Le rayon effectif  $R_{eff}$  peut être calculé comme suit :

- pour l'outil plat : son rayon effectif dépend de l'angle d'inclinaison  $\phi$ . Comme défini dans [72] :  $R_{eff} = \frac{R}{\sin \phi}$ , où  $R$  est le rayon de l'outil.
- pour l'outil sphérique : son rayon effectif est égal au rayon de l'outil quelle que soit son orientation :  $R_{eff} = R$ .
- pour l'outil torique : son rayon effectif dépend du rayon de l'outil  $R$ , du rayon du tore  $r$ , de la pente la plus forte  $s$  et de l'angle entre la direction d'usinage et la direction de la pente la plus forte  $\alpha$ . L'équation 1.5 proposée par [62] détaille l'expression de  $R_{eff}$  :

$$R_{eff} = \frac{(R - r) \cos^2 \alpha}{\sin s (1 - \sin^2 \alpha \sin^2 s)} + r \quad (1.5)$$

L'outil sphérique est largement utilisé dans l'industrie, car il ne nécessite pas de calcul du rayon effectif, ce qui rend la génération de trajectoires d'usinage assez simple. En outre, il laisse des marques moins prononcées sur la surface qu'un outil à bout plat, ce qui entraîne une moindre rugosité. Cependant, la fraise à bout plat permet un rayon effectif plus grand que son homologue à bout sphérique. La fraise torique, telle qu'illustrée dans [12], hérite des mérites des deux fraises mentionnées précédemment. En fait, elle peut permettre, selon la direction de l'usinage et les caractéristiques géométriques de la surface, un rayon effectif supérieur à celui de l'outil sphérique, tout en laissant des traces sur la surface plus lisses que celles d'une fraise plate de même rayon.

De nos jours, les fraises à bout plat ne sont, pour ainsi dire, plus utilisées pour la finition des surfaces gauches en raison des traces prononcées qu'elles laissaient sur la surface, entraînant une rugosité élevée dans le sens de l'avance [20, 39]. Par conséquent, elles ne sont pas prises en compte dans la présente

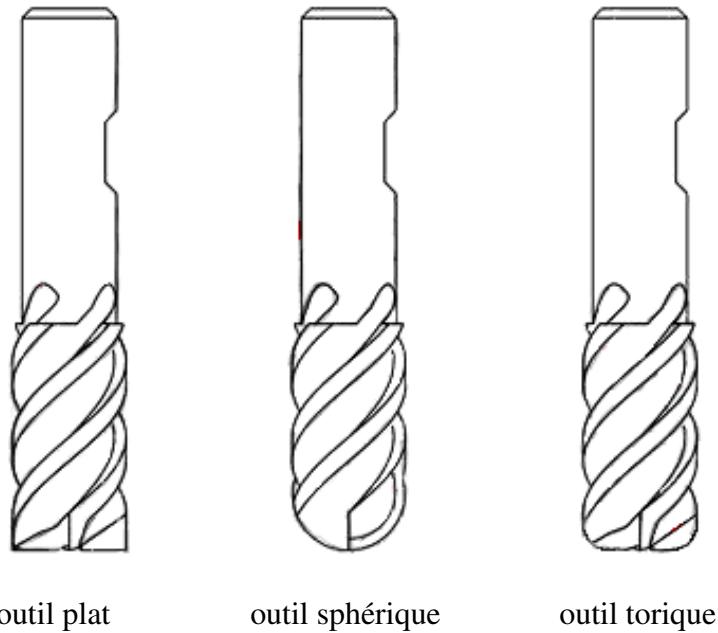


Figure 1.5 – Les principaux outils utilisés pour le fraisage en bout

étude.

La figure 1.6 montre les pas transversaux permis par la fraise sphérique (à gauche) et la fraise torique (à droite), lors de l’usinage dans la direction de la pente la plus raide (fraise bleue) ou dans une direction perpendiculaire à celle-ci (fraise verte). Toutes les hauteurs de crête sont les mêmes sur l’ensemble de la figure. Un simple plan a été choisi comme surface pour faciliter la lecture de la figure, mais les mêmes résultats sont obtenus pour des surfaces gauches.

Sur cette figure, on peut voir que pour la fraise sphérique, le rayon effectif, et donc la distance entre trajectoires adjacentes, sont les mêmes quelle que soit la direction d’usinage choisie. Il est également clair que l’utilisation d’une fraise torique dans la direction de la pente la plus forte conduit à de meilleurs résultats que l’utilisation d’une fraise à boule dans les mêmes conditions, tandis que l’utilisation d’une fraise torique dans la direction perpendiculaire à la direction de la pente la plus forte conduit à de moins bons résultats que ceux obtenus avec une fraise à boule dans les mêmes conditions.

Il a été prouvé dans [25] que le rayon effectif d’une fraise torique de rayons  $R$  et  $r$  est supérieur à celui obtenu par un outil sphérique de rayon  $R$ , lorsque l’angle  $\alpha$  est compris dans l’intervalle  $[-35^\circ, 35^\circ]$ . Cette condition est suffisante quelles que soient les valeurs de  $R$ ,  $r$  et  $s$ .

Une fois le rayon effectif connu, la hauteur de crête est assez facile à calculer. Pour les fraises toriques, l’utilisation de l’équation (1.5) permet un calcul beaucoup plus rapide que les méthodes numériques précédemment employées.

La contrainte de qualité de la surface est exprimée en terme de hauteur de crête maximale ; la hauteur de crête, notée  $h_c$ , correspond, comme le montre la figure 1.7, à l’épaisseur résiduelle non usinée laissée par l’outil lors de son déplacement suivant deux trajectoires adjacentes. La hauteur de crête peut être

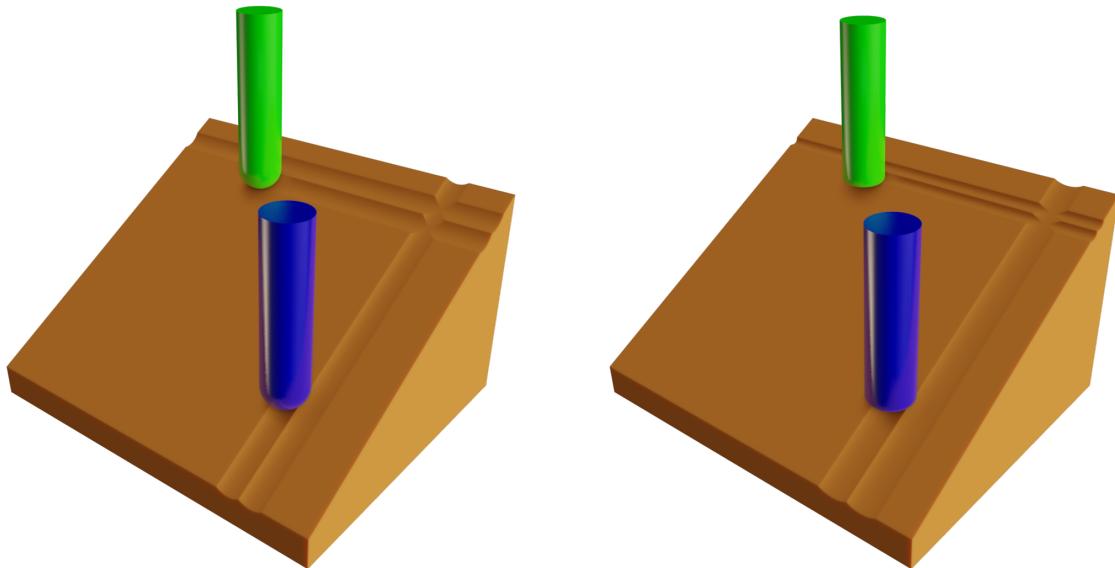


Figure 1.6 – Comparaison du pas transversal des outils sphérique (gauche) et torique (droite)

calculée soit en effectuant des approximations sur la géométrie de la surface et de l'outil, comme le font les auteurs de [76] et [72], ou en utilisant des méthodes numériques, comme dans [61] qui déduit la hauteur de crête à partir de deux trajectoires consécutives, un pas transversal, une surface nominale, un outil et un positionnement, et cela en s'intéressant aux intersections de l'outil dans deux positions adjacentes données.

## 1.3 Stratégies d'usinage

La planification de trajectoires d'usinage est un problème très vaste et délicat à modéliser. En effet, l'objectif est de trouver une trajectoire qui minimise le temps d'usinage sous contraintes de qualité ; il est cependant difficile de décrire n'importe quelle trajectoire avec un nombre raisonnable de variables. De plus, la hauteur de crête doit être évaluée sur un grand nombre de points, ce qui rend l'évaluation des contraintes dans le cas général très coûteuse.

En pratique, des stratégies pour générer des trajectoires d'outil sont utilisées. Ces stratégies sont des heuristiques qui permettent de réduire considérablement l'espace de recherche et le nombre de paramètres décrivant la trajectoire, ce qui facilite la planification. Bien que les solutions trouvées par ces stratégies soient sous-optimales, elles demeurent nettement meilleures qu'une solution triviale, car les stratégies d'usinages contiennent souvent de la connaissance métier. Parmi les stratégies d'usinage les plus connues, on peut citer :

### — Surface guide

Cette stratégie définit la trajectoire de l'outil à partir d'une surface intermédiaire appelée surface guide. L'outil se déplace tout en restant en contact avec la surface à usiner et la surface guide (figure 1.8). La position de l'outil est déterminée avec un calcul numérique itératif [10, 11] assez lourd en termes de

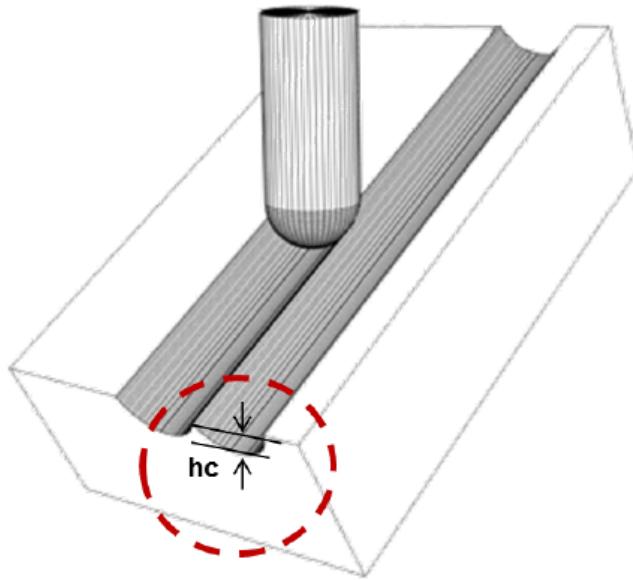


Figure 1.7 – Hauteur de crête

temps de calcul. Les auteurs de [40] préconisent une surface guide assez facile à construire et surtout compatible avec la surface à usiner.

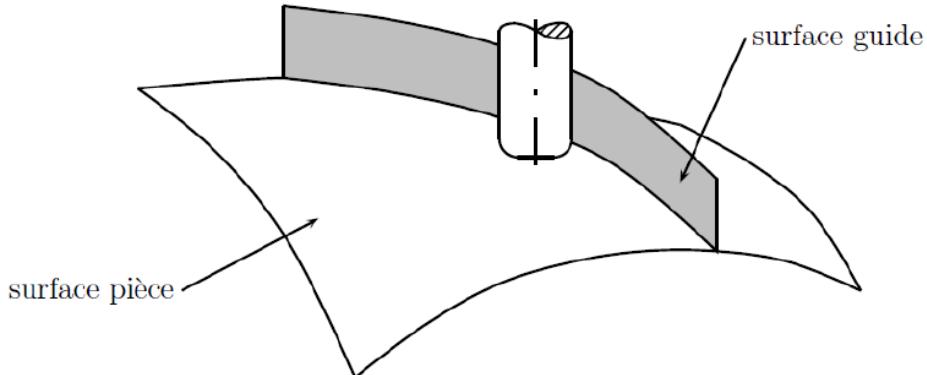


Figure 1.8 – Stratégie d'usinage par surface guide

### — Plans parallèles

Cette stratégie est la plus utilisée et maîtrisée en industrie, de par sa simplicité d'implémentation et son efficacité. Elle consiste à usiner la surface selon des plans parallèles (figure 1.9) ; la distance entre deux plans successifs est définie à partir de la contrainte de hauteur de crête. La stratégie plans parallèles en 3 axes s'appuie sur :

- la direction d'usinage, qui détermine l'orientation des plans ;
- le pas transversal, issu de la trajectoire précédente en satisfaisant la contrainte de hauteur de crête en tout point ;
- le pas longitudinal ; Pour une erreur de corde fixe, plus la surface est courbée plus le pas est petit, ce qui augmente le nombre de points d'interpolation de la trajectoire outil ;

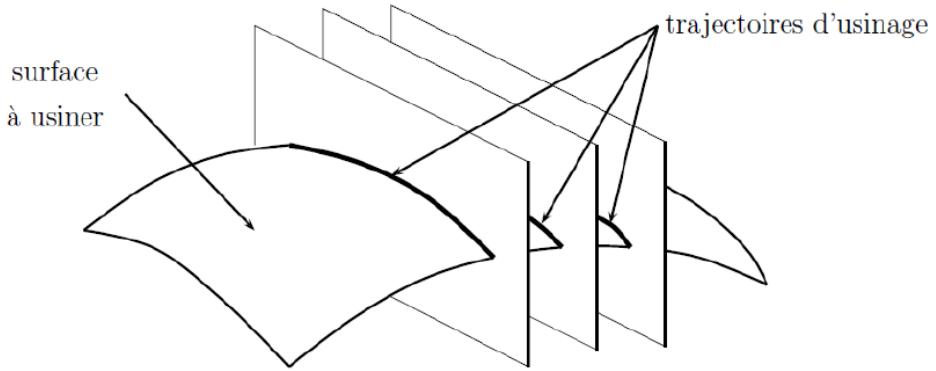


Figure 1.9 – Stratégie d'usinage par plans parallèles

- la géométrie de l'outil, qui détermine, pour une orientation donnée, le rayon effectif de l'outil et par la suite le pas transversal.

Lorsqu'on usine en plans parallèles selon une direction donnée, il est possible d'usiner tous les plans dans le même sens, ce qui définit une stratégie de type *one way*. On peut aussi, alterner le sens d'un plan à l'autre, ce qui définit une stratégie de type *zigzag*.

En fonction du directeur de commande numérique (DCN) utilisé, les trajectoires peuvent être exprimées sous forme de segments de droite, d'arcs de cercle, de courbes B-spline ou de NURBS. Pour optimiser la stratégie plans parallèles, il est possible de choisir le premier plan parallèle à la frontière de plus grande longueur, ou le plan avec la passe de largeur maximale. Ensuite, on peut aussi découper la surface en zones fortement inclinées, qui seront usinées selon des plans normaux à l'axe de l'outil, et des zones faiblement inclinées, usinées selon des plans contenant l'axe de l'outil [52]. Ce découpage donne un résultat largement sous optimal mais néanmoins souvent meilleur que l'usinage de toute la surface en une seule zone.

#### — Isoparamétriques

Cette stratégie consiste à usiner la surface selon des courbes isoparamétriques, illustrées par la figure 1.10. Une courbe isoparamétrique d'une surface  $S(u, v)$  est la courbe de la surface telle que le paramètre  $u$  (ou  $v$ ) est constant. Il s'agit donc de trouver une séquence du paramètre  $u$  ou  $v$  dans  $[0, 1]$  telle que toute la surface soit usinée en respectant la contrainte de hauteur de crête.

La stratégie isoparamétrique est plus rapide en terme de temps de calcul, car elle ne nécessite pas de calcul d'intersections entre surfaces contrairement aux stratégies surface guide et plans parallèles. Cependant, le résultat dépend du choix du paramètre à fixer ( $u$  ou  $v$ ). Le problème majeur de la stratégie isoparamétrique tient au fait que des trajectoires redondantes risquent d'apparaître, ce qui affecte la qualité de l'usinage. De plus, la mise en œuvre de cette stratégie est compliquée pour les surfaces composées de plusieurs carreaux paramétriques.

Une amélioration de la stratégie isoparamétrique est proposée par [29], en utilisant des iso-courbes adaptatives.

#### — Recouvrement de surfaces ou space filling curves (SFC) [24]

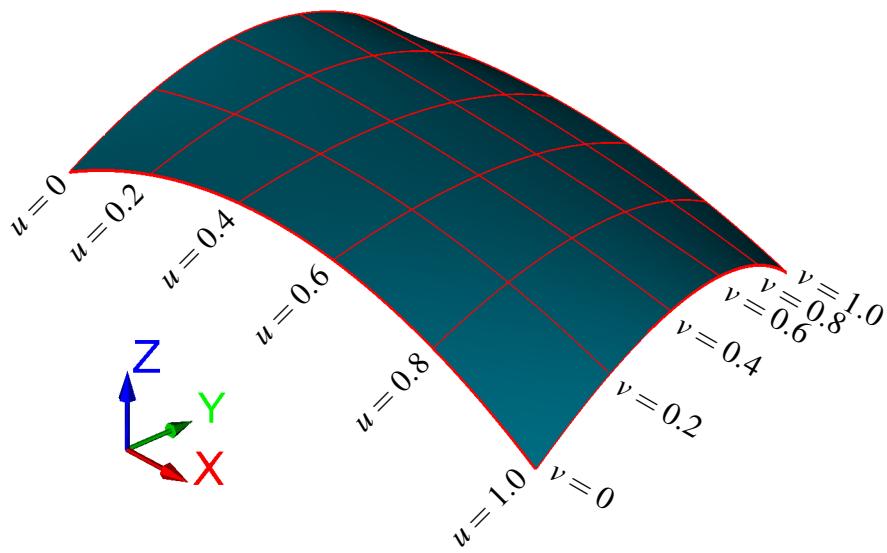


Figure 1.10 – Courbes isoparamétriques

Cette stratégie consiste à utiliser des courbes fractales continues pour recouvrir toute la surface à longueur minimale, tout en respectant la contrainte de hauteur de crête. La figure 1.11 montre le résultat de l'approche de [34] où la trajectoire d'usinage est générée en utilisant des courbes de Hilbert. Cette stratégie génère des trajectoires avec de nombreux et très importants changements de direction, ce qui pénalise la dynamique de la machine et fait qu'en pratique elle n'est jamais utilisée.

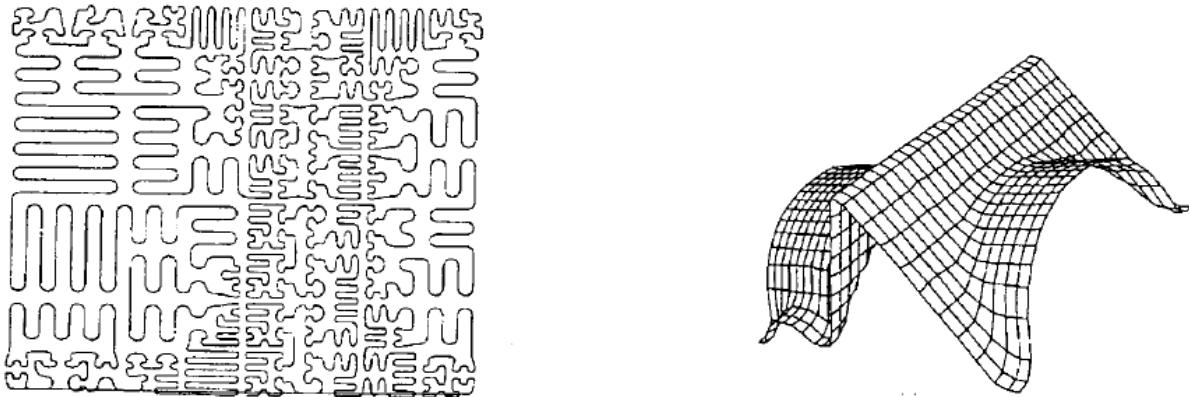


Figure 1.11 – Génération de trajectoire d'usinage avec des courbes de Hilbert

### — Isocrête

Les travaux de [71], [65], [69], [75] et [45] se sont intéressés à la stratégie isocrête. Cette stratégie nécessite une trajectoire initiale, dont le résultat final est énormément dépendant. Le concept de *machining potential field* est utilisé par [19] pour trouver une trajectoire initiale en suivant la plus grande pente. L'inconvénient de cette approche est que la première trajectoire peut présenter des changements d'orientation qui seront propagés et amplifiés sur les trajectoires suivantes.

Trouver une direction d'usinage et une trajectoire initiale peut être traité comme un problème d'optimisation. Un algorithme génétique est utilisé dans [3] pour trouver une première trajectoire de la forme  $v = a_0 + a_1u + a_2u^2 + a_3u^3$ . Dans [9], un réseau de neurones est utilisé pour prédire la stratégie d'usinage la plus adaptée en fonction de la surface à usiner. Pour entraîner le réseau, chaque surface est représentée par un vecteur contenant les coordonnées réelles des points de la surface choisis à partir d'une grille uniforme. Un tel vecteur est assez grand de taille et exige un grand nombre de surfaces pour entraîner le réseau. D'autre part, afin usiner proprement et correctement les surfaces gauches, un ensemble de six conditions nécessaires et suffisantes, portant sur les paramètres géométriques et cinématiques de l'usinage, sont proposées dans [60].

Une revue de littérature assez récente peut être trouvée dans [43]. Cet article décrit les développements proposés dans le domaine d'usinage de surfaces gauches, depuis 1997 jusqu'à 2008, cette revue s'intéresse particulièrement à trois aspects :

- la génération de la trajectoire d'outil
- l'identification de l'orientation de l'outil
- le choix de la géométrie de l'outil

Une trajectoire d'outil doit satisfaire les contraintes de qualités suivantes : la hauteur de crête doit être inférieure au maximum toléré, et il ne doit pas y avoir de *gouging*, c'est-à-dire d'interférences entre l'outil et la surface en dehors du point de contact. On peut distinguer trois types de *gouging* : local, *rear* et global.

Il existe plusieurs topologies possibles pour une trajectoire d'outil : à directions parallèles, à contours parallèles et les *curvilinear space fitting curve* (SFC), modélisé comme un problème de voyageur de commerce TSP) [6].

Même si elle est la plus utilisée dans l'industrie, la stratégie plans parallèles présente de la surqualité au niveau de la contrainte de hauteur de crête, ce qui donne nécessairement des trajectoires sous-optimales. La stratégie isocrête permet d'éviter ce problème avec moins de points de contact surface-outil (**Cc point**). Cependant, elle souffre de possibles dérives qui se propagent de la courbe *maîtresse* jusqu'à la dernière courbe de la trajectoire. La courbe maîtresse (*master cutter path*) peut être identifiée grâce au concept de *curvature matching* [37] qui impose que le rayon effectif de l'outil doit être inférieur ou égal au rayon de courbure de la surface à chaque point de contact surface-outil (**Cc point**). Ceci permet d'avoir une première courbe qui satisfait la contrainte de *local gouging*.

Quelle que soit la stratégie d'usinage employée, il peut s'avérer utile de partitionner la surface en des régions regroupant les points avec un vecteur normal similaire dans une même région. Par conséquent, chaque région est composée de points avec des directions de plus grande pente proches, ce qui tend à améliorer l'efficacité de l'usinage. Cette démarche peut être généralisée à un ensemble de caractéristiques géométriques utilisées comme critère pour trouver des régions composées de points similaires, chaque région est ensuite usinée indépendamment des autres.

La stratégie espace de configurations (*Configuration space*, ou *C-space*) cherche à trouver une succession de degrés de liberté optimaux de l'outil, afin de générer une trajectoire d'usinage de longueur minimale, tout en satisfaisant les contraintes de hauteur de crête et de non-interférence [38].

Pour ce qui concerne les surfaces non-paramétriques, les méthodes à base de nuages de points ou de modèles polyédraux peuvent être utilisées [68]. Il est aussi possible de procéder par usinage de surface composée, c'est-à-dire utiliser des stratégies d'usinages différentes sur des régions différentes de la surface.

Contrairement à l'usinage 3 axes, où l'outil n'a que trois mouvements de translation, l'usinage 5 axes permet deux axes de rotation supplémentaires. Traditionnellement, pour identifier l'orientation de l'outil (définie à chaque point par les deux angles de rotations), l'angle d'inclinaison est fixé, ou bien des méthodes comme *principal axis* et *multi-point machining* sont utilisées, quoiqu'elles ne sont pas très connues à cause de leur complexité mathématique et la possibilité de ne pas converger vers une solution [77]. D'autres méthodes telles que : *C-space*, *rolling ball* et *arc-intersect* [33] peuvent également être utilisées pour identifier l'orientation de l'outil.

Afin d'éviter des changements conséquents de l'orientation de l'outil, ce qui est contre-productif pour l'usinage, des méthodes pour trouver des orientations de l'outil qui varient régulièrement ont été développées [38, 48].

L'usinage 3+2 axes est souvent préféré à l'usinage 5 axes. En effet, la dynamique des axes de rotation entraîne souvent des ralentissements lorsqu'ils sont utilisés en continu. Dans ce cas, il peut être plus judicieux d'utiliser les axes de rotation uniquement pour positionner la pièce dans l'espace de travail de la machine. Une fois la pièce correctement positionnée et orientée, pendant le processus de fraisage proprement dit, seuls les axes de translation sont utilisés. C'est ce que l'on appelle l'usinage 3+2 axes. Pour les surfaces gauches, l'usinage 3+2 axes est plus simple à manipuler, et donne souvent de meilleurs résultats que l'usinage 5 axes en continu [67], surtout lorsque les surfaces sont partitionnées efficacement.

## 1.4 Éléments de statistique

Nous rappelons ici quelques éléments fondamentaux de la statistique qui nous seront utiles dans la suite.

### 1.4.1 Moyenne et médiane

Plusieurs indicateurs existent pour mesurer la tendance centrale d'un échantillon statistique. Soit  $n \geq 1$  un entier, et  $X = (X_i)_{1 \leq i \leq n} \in \mathbb{R}^n$  un échantillon statistique. On définit la moyenne (arithmétique) comme la somme des observations  $X_i$  divisée par leur nombre :

$$E(X) = \bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

La moyenne sert à résumer la tendance centrale de l'échantillon, il existe d'autres valeurs comme la médiane, le mode et les quantiles.

La moyenne est linéaire. En effet, soit  $X = (X_i)_{1 \leq i \leq n} \in \mathbb{R}^n$  et  $Y = (Y_i)_{1 \leq i \leq n} \in \mathbb{R}^n$  deux échantillons statistiques,  $\lambda, \mu \in \mathbb{R}$  deux scalaires, on a :

$$E(\lambda X + \mu Y) = \lambda E(X) + \mu E(Y)$$

La médiane est définie comme la valeur qui permet de couper les observations en deux parties de tailles égales ;  $\text{card}\{X_i \mid X_i \leq \text{Med}(X)\} = \frac{n}{2}$ . La médiane est homogène mais pas additive :

$$\text{Med}(\lambda X) = \lambda \text{Med}(X) \text{ mais } \text{Med}(X + Y) \neq \text{Med}(X) + \text{Med}(Y)$$

La moyenne,  $E(X)$ , minimise les écarts au carré (équation 1.6), tandis que la médiane,  $\text{Med}(X)$ , minimise les écarts en valeur absolue 1.7. Il est important de noter que la médiane est plus robuste par rapport aux valeurs aberrantes.

$$\forall t \in \mathbb{R}, \sum_{i=1}^n (X_i - E(X))^2 \leq \sum_{i=1}^n (X_i - t)^2 \quad (1.6)$$

$$\forall t \in \mathbb{R}, \sum_{i=1}^n |X_i - \text{Med}(X)| \leq \sum_{i=1}^n |X_i - t| \quad (1.7)$$

### 1.4.2 Variance et écart-type

Plusieurs indicateurs existent pour mesurer la dispersion d'un échantillon statistique, à savoir :

— la variance, définie comme la moyenne des écarts à la moyenne au carré,

$$\text{Var}(X) = E((X - E(X))^2) = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$$

— l'écart type, défini comme la racine carré de la variance, s'exprime dans les mêmes unités que la moyenne. Ainsi ces quantités peuvent être additionnées pour considérer la dispersion à l'aide d'intervalles.

$$\sigma(X) = \sqrt{\text{Var}(X)}$$

— l'étendue, définie par la différence entre la valeur maximale et minimale de l'échantillon :

$$\max_{1 \leq i \leq n} X_i - \min_{1 \leq i \leq n} X_i$$

La variance peut être utilisée comme une mesure de la capacité à résumer l'information contenue dans l'échantillon par la moyenne. Autrement dit, la variance quantifie l'information non expliquée par la moyenne.

La variance est quadratique et invariante par translation :

$$\forall a \in \mathbb{R}, \text{Var}(aX) = a^2 \text{Var}(X) \text{ et } \forall b \in \mathbb{R}, \text{Var}(X + b) = \text{Var}(X)$$

La variance est nulle si et seulement si l'échantillon est constant. En effet :

$$\text{Var}(X) = 0 \iff \forall 1 \leq i \leq n, (X_i - \text{E}(X))^2 = 0 \iff \forall 1 \leq i \leq n, X_i = \text{E}(X).$$

Il est possible d'exprimer les données d'un échantillon dans une échelle neutre en les adimensionnant de leurs unités physiques. Ceci est fait en centrant l'échantillon autour de sa moyenne en divisant par l'écart-type. Une fois centrées et réduites, les observations s'expriment comme un nombre d'écart-types par rapport à la moyenne. L'équation 1.8 donne l'échantillon centré-réduit  $Z$  en fonction de l'échantillon  $X$ , ce qui implique bien :  $\text{E}(Z) = 0$  et  $\text{Var}(Z) = 1$ .

$$\forall 1 \leq i \leq n, Z_i = \frac{X_i - \bar{X}}{\sigma(X)} \quad (1.8)$$

### 1.4.3 Covariance et corrélation

Pour deux échantillons  $X$  et  $Y$  on définit la covariance, notée  $\text{Cov}(X, Y)$ , comme la moyenne du produit des écarts à la moyenne de chaque échantillon :

$$\text{Cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (X_i - \text{E}(X))(Y_i - \text{E}(Y)).$$

La covariance est positive si les  $X_i$  et les  $Y_i$  ont tendance à être simultanément du même côté de leurs moyennes respectives. Inversement, elle est négative si les  $X_i$  et les  $Y_i$  ont tendance à être simultanément du côté opposé de leurs moyennes respectives.

La covariance est symétrique :

$$\text{Cov}(X, Y) = \text{Cov}(Y, X)$$

La covariance est bilinéaire :

$$\forall \lambda, \mu \in \mathbb{R} \quad \text{Cov}(\lambda X + \mu Z, Y) = \lambda \text{Cov}(X, Y) + \mu \text{Cov}(Z, Y)$$

La covariance est définie positive :

$$\text{Cov}(X, X) = \text{Var}(X) = \sigma(X)^2 \geq 0 \text{ et } \text{Cov}(X, X) = 0$$

On en déduit que la covariance est un produit scalaire et l'écart type est sa norme associée. L'inégalité de Cauchy-Schwarz s'écrit :

$$|\text{Cov}(X, Y)| \leq \sigma(X) \sigma(Y)$$

Le coefficient de corrélation linéaire de Pearson entre les échantillons  $X$  et  $Y$  est défini par :

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma(X) \sigma(Y)} \in [-1, 1]$$

Par linéarité de la covariance,  $\rho(X, Y)$  est également la covariance des versions centrées-réduites de  $X$  et  $Y$ . Le signe de  $\rho(X, Y)$  s'interprète comme celui de  $\text{Cov}(X, Y)$ . Si  $|\rho(X, Y)| = 1$ , alors il y a égalité dans la relation de Cauchy-Schwarz et donc colinéarité, c'est à dire que les observations sont distribuées sur une droite (figure 1.12).

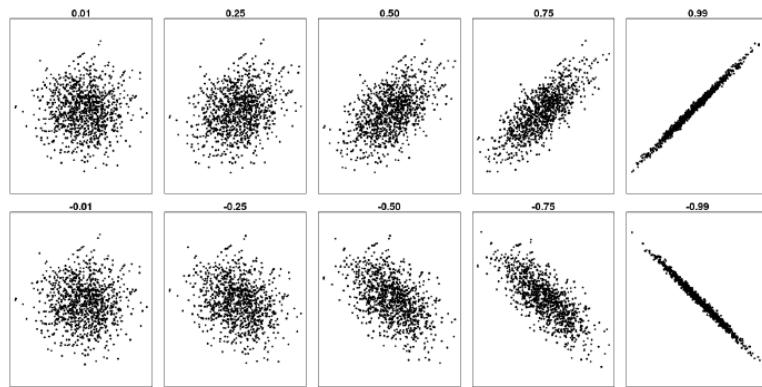


Figure 1.12 – Exemples de coefficients de corrélation linéaire

#### 1.4.4 Analyse en composantes principales

On considère cette fois un échantillon  $X = (X^i)_{1 \leq i \leq n} \in \mathbb{R}^{p \times n}$  de vecteurs de dimension  $p$ . Une façon simple de généraliser la variance dans un cadre multidimensionnel consiste à définir l'inertie (standard), notée  $I(X)$ , comme la somme des variances :

$$I(X) = \sum_{l=1}^p \text{Var} X_l = \frac{1}{n} \sum_{l=1}^p \sum_{i=1}^n (X_l^i - \mathbb{E}(X_l))^2 = \frac{1}{n} \sum_{i=1}^n |X^i - \mathbb{E}(X)|^2.$$

Soit  $v \in \mathbb{R}^p$  de norme unitaire  $|v|^2 = 1$ , et  $i \in \{1, \dots, n\}$ , la projection orthogonale du vecteur observé  $X^i - \mathbb{E}(X)$  sur la droite engendrée par  $v$  est donnée par :

$$\langle X^i - \mathbb{E}(X), v \rangle v = v^T X^i - \mathbb{E}(X) v$$

Autrement dit, l'échantillon (scalaire) obtenu en projetant l'échantillon  $X$  sur la droite engendrée par  $v$  s'écrit :  $\langle X^1 - \mathbb{E}(X) \rangle, \dots, \langle X^n - \mathbb{E}(X) \rangle$ . Les données projetées sont centrées par construction :

$$\mathbb{E}(\langle X - \mathbb{E}(X) \rangle) = \frac{1}{n} \sum_{i=1}^n \langle X^i - \mathbb{E}(X), v \rangle = \left\langle \frac{1}{n} \sum_{i=1}^n X^i - \mathbb{E}(X), v \right\rangle = \langle \mathbb{E}(X) - \mathbb{E}(X), v \rangle = 0$$

La variance des données projetées s'écrit :

$$\begin{aligned}\text{Var}(\langle X - \mathbb{E}(X), v \rangle) &= \frac{1}{n} \sum_{i=1}^n \langle X^i - \mathbb{E}(X), v \rangle^2 = \\ \frac{1}{n} \sum_{i=1}^n v^T (X^i - \mathbb{E}(X)) (X^i - \mathbb{E}(X))^T v &= v^T \left( \frac{1}{n} \sum_{i=1}^n (X^i - \mathbb{E}(X)) (X^i - \mathbb{E}(X))^T \right) v = v^T \Sigma v\end{aligned}$$

La matrice  $\Sigma$  de taille  $p \times p$ , qui apparaît dans le calcul de la variance,

$$\Sigma = \frac{1}{n} \sum_{i=1}^n (X^i - \mathbb{E}(X)) (X^i - \mathbb{E}(X))^T$$

s'appelle la matrice de covariance car,

$$\forall l_1, l_2 \in \{1, \dots, p\} \Sigma_{l_1 l_2} = \frac{1}{n} \sum_{i=1}^n (X_{l_1}^i - \mathbb{E}(X_{l_1})) (X_{l_2}^i - \mathbb{E}(X_{l_2})) = \text{Cov}(X_{l_1}, X_{l_2})$$

Cette matrice est symétrique car,

$$\Sigma_{l_1 l_2} = \text{Cov}(X_{l_1}, X_{l_2}) = \text{Cov}(X_{l_2}, X_{l_1}) = \Sigma_{l_2 l_1}.$$

Elle est aussi semi-définie positive puisque,

$$\forall u \in \mathbb{R}^p, \quad u^T \Sigma u = \text{Var}(u^T X) \geq 0,$$

elle est donc diagonalisable. Le vecteur propre associé à la plus grande valeur propre de  $\Sigma$  est le vecteur unitaire  $u$  qui maximise le terme  $u^T \Sigma u$ . Il détermine donc la droite sur laquelle la projection de l'échantillon  $X$  conserve le plus d'information au sens de la variabilité.

La matrice  $\Sigma$  possède  $p$  vecteurs propres  $v^1, \dots, v^p$  associés aux valeurs propres respectives  $\lambda_1 \geq \dots \geq \lambda_p \geq 0$ . Les vecteurs propres forment une base orthonormale de  $\mathbb{R}^p$ .

Le vecteur propre  $v^1$  est la première direction principale et l'inertie selon la droite qu'il génère,  $v^1 \Sigma v^1 = \lambda_1$ . Plus généralement, l'espace de dimension  $d$  qui conserve le plus d'information est l'espace engendré par les  $d$  premiers vecteurs propres  $v^1, \dots, v^d$  et l'inertie des données projetées dans cet espace vaut :

$$\sum_{l=1}^d v^l \Sigma v^l = \sum_{l=1}^d \lambda_l \left( \leq \sum_{l=1}^p \lambda_l = I(X) \right)$$

Dans la base des directions principales, l'échantillon  $X$  peut s'écrire  $C = X V$ . La matrice  $C$  est la matrice des composantes principales.

## 1.5 Optimisation de boite noire et sans dérivées

Le problème d’usinage est rarement (si ce n’est jamais en pratique) décrit par des fonctions analytiques ; de plus, des quantités comme la longueur de trajectoire d’outil, ou la durée d’usinage ne peuvent être calculées que numériquement. Il est donc nécessaire d’avoir recours à des algorithmes d’optimisation sans dérivées (ou de boites noires). Le livre de Charles Audet et Warren Hare [8] présente une introduction aux différents algorithmes d’optimisation sans dérivées. Dans ce cadre, on ne dispose pas d’informations concernant le gradient ou la hessienne de la fonction-objectif et des contraintes. De fait, l’évaluation des fonctions se fait à travers une routine (un code) ou une expérimentation physique, décrite comme une boite noire, et donc aucune formule analytique pour les fonctions objectif et les contraintes n’est définie. Voici quelques algorithmes d’optimisation sans dérivée, cités dans ce livre, et couramment employés :

— Algorithme de recherche par coordonnées :

Ou algorithme *coordinate search* (CS). Dans sa version la plus simple, à chaque itération un ensemble de sonde ou *poll set* est créé en utilisant la base positive maximale :  $P = \{e_i, -e_i \text{ pour } i \in \llbracket 1; n \rrbracket\}$  et un pas  $\delta$ . Si le *poll set* contient un point meilleur que le point courant  $x_k$ , celui-ci est alors mis à jour. Sinon le pas est réduit :  $\delta_{k+1} = \frac{\delta_k}{2}$  par exemple.

— Algorithmes génétiques :

Cette famille d’algorithmes tente une optimisation globale, en évoluant un ensemble de point (population) à chaque itération à travers des opérateurs de sélection, croisement et mutations. Cependant, l’utilisation des ces algorithmes s’avère très coûteuse en termes de nombre d’évaluations des fonctions. De plus, il n’y a pas de preuve de convergence dans le cas général.

— Algorithme de Nelder-Mead :

Cette méthode, basée sur la notion de simplexes, est connue pour donner d’assez bon résultats. Par contre, l’algorithme de Nelder-Mead est un optimiseur local (sans preuve de convergence en dimension supérieure à 1). Il reste néanmoins moins coûteux que les algorithmes génétiques par exemple. Dans  $\mathbb{R}^2$ , un simplexe est un triangle défini par les points  $\mathbf{y}^0, \mathbf{y}^1$  et  $\mathbf{y}^2$  tels que  $f(\mathbf{y}^0) \leq f(\mathbf{y}^1) \leq f(\mathbf{y}^2)$ . Selon l’orientation du triangle définie dans  $\mathbb{R}^3$  par les points  $\begin{pmatrix} \mathbf{y}^0 \\ f(\mathbf{y}^0) \end{pmatrix}, \begin{pmatrix} \mathbf{y}^1 \\ f(\mathbf{y}^1) \end{pmatrix}$  et  $\begin{pmatrix} \mathbf{y}^2 \\ f(\mathbf{y}^2) \end{pmatrix}$ , les sommets du simplexe sont mis à jour via une réflexion, expansion, contraction intérieure ou extérieure. Ceci conduit à un nouveau simplexe dont le volume tend vers 0 à convergence, définissant la solution proposée.

— Algorithme *Generalized Pattern Search* (GPS) :

Cette méthode repose sur un maillage de l’espace de recherche et peut être vue comme une généralisation de l’algorithme de recherche directionnelle. À chaque itération  $k$ , une étape de recherche est d’abord effectuée afin de tenter de quitter un optimum local. Cette étape de recherche est flexible, et reste à définir par l’utilisateur. Si l’étape de recherche ne parvient pas à trouver un meilleur point, alors l’étape de sonde est effectuée : une famille positivement génératrice  $D^k$  est sélectionnée à partir des colonnes de  $D = GZ$ , où  $G$  est une matrice inversible de  $\mathbb{R}^{n \times n}$  et  $Z \in \mathbb{Z}^{n \times p}$  est tel que les vecteurs colonnes de  $Z$

forment une famille positivement génératrice de  $\mathbb{R}^n$ . Si le *poll set*  $P^k = \{x_k + \delta_k d, d \in D^k\}$  contient un point qui améliore la solution courante  $x_k$  alors cette solution est mise à jour et le pas  $\delta$  est augmenté. Sinon,  $x_k$  est un optimum local pour ce maillage, le maillage est donc raffiné en diminuant le pas  $\delta$ , et ainsi de suite jusqu'à convergence.

— Algorithme *Mesh Adaptive Direct Search* (MADS) :

L'algorithme MADS est très similaire à l'algorithme GPS. La différence est que pendant le *poll step*, l'algorithme MADS utilise la notion de *frame of extent*  $\Delta_k$  (ou cadre de sonde  $\Delta_k$ ) qui peut être vu comme une extension de la notion de maillage, car elle autorise plus de directions. Par ailleurs, la fonction barrière progressive (*progressive barrier function*) est évaluée, au lieu de la fonction-objectif, afin de pouvoir traiter les problèmes d'optimisation avec contraintes.

— modèles linéaire et quadratique :

L'équation 1.9 définit un modèle d'interpolation linéaire :

$$L_Y(\mathbf{x}) = \alpha_0 + \alpha^T \mathbf{x} \text{ avec } [1 \ Y^T] \begin{bmatrix} \alpha_0 \\ \alpha \end{bmatrix} = f(Y) \text{ et } Y = [\mathbf{y}^0 \ \mathbf{y}^1 \dots \mathbf{y}^n] \quad (1.9)$$

L'équation 1.10 définit à son tour un modèle de régression linéaire :

$$L_Y(\mathbf{x}) = \alpha_0 + \alpha^T \mathbf{x} \text{ avec } (\alpha_0, \alpha) \text{ la solution unique du} \\ \text{problème de moindres carrés : } \min_{\alpha_0, \alpha} \sum_{i=0}^m (\alpha_0 + \alpha^T \mathbf{y}^i - f(\mathbf{y}^i))^2 \quad (1.10)$$

L'équation 1.11 définit le modèle d'interpolation quadratique :

$$Q_Y(\mathbf{x}) = \alpha_0 + \alpha^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T H \mathbf{x} \text{ avec } \alpha_0, \alpha, H = H^T \text{ la solution unique de :} \\ \alpha_0 + \alpha^T \mathbf{y}^i + \frac{1}{2} \mathbf{y}^{iT} H \mathbf{y}^i = f(\mathbf{y}^i), \text{ pour } 0 \leq i \leq \frac{(n+1)(n+2)}{2} - 1 \quad (1.11)$$

Les méthodes basées sur de tels modèles (par exemple la méthode DFO présentée dans le livre [22]) minimisent le modèle afin de proposer un nouveau point d'évaluation (possiblement en conjonction avec l'utilisation d'une région de confiance).

## 1.6 Calcul des temps et longueur d'usinage

Dans cette section, nous expliquons en détails le calcul de la trajectoire d'outil, en utilisant la stratégie plans parallèles, et du temps d'usinage correspondant.

### 1.6.1 Longueur de trajectoire

La trajectoire d'outil est calculée itérativement. Dans le cas de la stratégie par plans parallèles, le parcours d'outil total est constitué de portions sans angles aigus (appelés trajectoires) qui sont calculées successivement. La distance entre deux trajectoires adjacentes (dite pas transversal) est calculée en respectant la contrainte de hauteur de crête. Chaque trajectoire est définie par des points d'interpolation, qui sont eux aussi calculés de façon itérative. Le premier point de la trajectoire étant sur le bord de la surface, le prochain point d'interpolation  $(u_{k+1}, v_{k+1})$  est calculé à partir du point courant  $(u_k, v_k)$ , en avançant d'un vecteur  $\delta$  selon la direction d'avance puis en le projetant orthogonalement sur la surface, ce qui conduit à la relation 1.12 ci-dessous, où  $\mathbf{n}$  est le vecteur normal. Ainsi, pour calculer chaque point d'interpolation, il faut résoudre un système non-linéaire de 3 équations à 3 inconnues, à savoir  $u_{k+1}$ ,  $v_{k+1}$  et  $\varepsilon$ . Ce problème est résolu par la méthode de Newton-Raphson.

$$\mathbf{S}(u_{k+1}, v_{k+1}) = \mathbf{S}(u_k, v_k) + \delta + \varepsilon \mathbf{n}(u_{k+1}, v_{k+1}) \quad (1.12)$$

La résolution du système non-linéaire de taille 3 par la méthode de Newton-Raphson est assez rapide en soi, toutefois, ce système est résolu à chaque point d'interpolation, ce qui conduit à un temps de calcul assez conséquent. Encore une fois, le temps de simulation pour une seule évaluation du temps d'usinage reste acceptable, néanmoins, pour un processus d'optimisation (par exemple), où on est amené à faire un grand nombre d'évaluations du temps d'usinage, le temps de calcul risque d'être inacceptable. Il est donc nécessaire de trouver une seconde approche pour calculer la trajectoire de l'outil, qui soit plus rapide que la première. Pour cela, on utilise le réseau isoparamétriques qui définit le maillage de la surface. La longueur de la trajectoire de l'outil est approximée par la longueur de la courbe linéaire par morceaux définie à partir des points d'intersections entre le plan d'usinage courant et le réseau des isoparamétriques. On rappelle que la stratégie d'usinage adoptée est la stratégie plans parallèles, cette approximation est donc fiable à condition que le maillage soit assez fin. La figure 1.13 illustre ce concept où les points rouges sont les points d'intersections des plans d'usinage avec les courbes isoparamétriques.

### 1.6.2 Temps d'usinage

Le coût de production d'une surface est directement lié au temps d'usinage. De nombreux auteurs, par exemple : [26, 34, 44, 50, 55, 56, 63], supposent que le temps d'usinage est proportionnel à la longueur de trajectoire, et se concentrent donc sur la minimisation de la longueur de trajectoire, plutôt que sur le temps d'usinage. Cette approximation n'est pas exacte, sauf si la vitesse de l'outil est constante, ce qui est idéal mais peu réaliste. En pratique, l'outil doit décélérer lorsqu'il arrive à un angle aigu (c'est à dire une trajectoire linéaire avec discontinuité de tangence). De même, il doit accélérer une fois cet angle passé (changement de direction). Il est évident que la fraise ne peut pas passer par une discontinuité de tangence sans réduire sa vitesse à zéro. Pour éviter une vitesse complètement nulle, les commandes

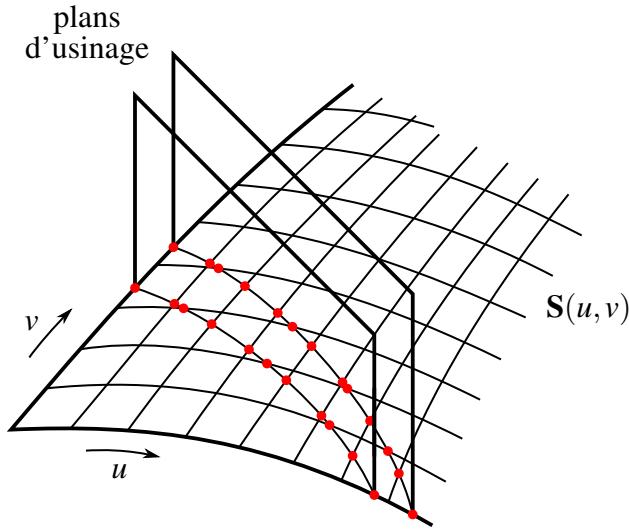


Figure 1.13 – Approximation de la longueur d’usinage en utilisant l’intersection entre les plans d’usinage et les isoparamétriques

numériques (CN) actuelles permettent une trajectoire légèrement arrondie avec une tolérance donnée, mais cette tolérance est maintenue suffisamment faible pour ne pas avoir d’impact visible sur la pièce finie. Cependant, la vitesse d’avance doit toujours diminuer à l’approche des angles les plus aigus, puis augmenter à nouveau une fois cet angle passé. De plus amples explications (et une optimisation) de ce problème peuvent être trouvées dans [59].

Par conséquent, les capacités cinématiques de la machine CNC doivent être prises en compte, car les caractéristiques de déplacement de l’outil doivent être maintenues dans ces limites (voir [58] et [57] pour une modélisation complète de la cinématique des machines à 3 axes).

Afin de calculer le temps nécessaire pour parcourir une portion de trajectoire linéaire entre deux angles aigus, certaines valeurs doivent être connues :

- la vitesse d’avance nominale, notée  $V$  (m/s)
- l’accélération maximale, notée  $A_{max}$  (m/s<sup>2</sup>)
- le gradient maximal d’accélération, également appelé jerk, noté  $J_{max}$  (m/s<sup>3</sup>)

En utilisant ces notations, trois cas peuvent se présenter :

1. la vitesse d’avance nominale  $V$  et l’accélération maximale  $A_{max}$  sont atteintes
2. l’avance nominale  $V$  est atteinte, mais pas l’accélération maximale  $A_{max}$
3. l’avance nominale  $V$  et l’accélération maximale  $A_{max}$  ne sont pas atteintes.

Pour un segment de droite de longueur  $2l$ , en fonction des valeurs de  $V$ ,  $A_{max}$  et  $J_{max}$ , on définit également les temps caractéristiques suivants :

- le temps pour atteindre l’accélération maximale dans le cas 1 :  $t_A = \frac{A_{max}}{J_{max}}$ .
- le temps pour atteindre la vitesse d’avance nominale dans le cas 2 :  $t_V = \sqrt{\frac{V}{J_{max}}}$

- le temps pour parcourir la demi-longueur de la distance  $l$  dans le cas 3 :  $t_l = \sqrt[3]{\frac{l}{J_{max}}}$

Ces temps caractéristiques permettent de distinguer les cas précédemment énoncés. La caractérisation de chaque cas et le temps d’usinage  $t$  correspondant sont alors les suivants :

- **cas 1** : si  $t_A < t_V$  et  $t_A < t_l$ ,

$$t = 2 \frac{l}{V} + \frac{A_{max}}{J_{max}} + \frac{V}{A_{max}} \quad (1.13)$$

- **cas 2** : si  $t_V < t_A$  et  $t_V < t_l$ ,

$$t = 2 \frac{l}{V} + 2 \sqrt{\frac{V}{J_{max}}} \quad (1.14)$$

- **cas 3** : si  $t_l < t_A$  et  $t_l < t_V$ ,

$$t = 4 \sqrt[3]{\frac{l}{J_{max}}} \quad (1.15)$$

En particulier, lorsque  $t_V < t_A$ , qui sera le cas le plus courant pour les valeurs des paramètres cinématiques utilisés dans ce travail, le cas 1 est écarté et le cas 2 (respectivement le cas 3) s’applique si et seulement si  $l > \sqrt{\frac{V^3}{J_{max}}}$  (respectivement  $l < \sqrt{\frac{V^3}{J_{max}}}$ ).

Par conséquent, le temps d’usinage associé à un parcours d’outil augmente lorsque le nombre de trajectoires augmente, même si la longueur du parcours est constant.



# Chapitre 2

## Approximation des surfaces gauches

Une nouvelle méthodologie d'approximation des surfaces gauches par des plans, à l'aide de l'Analyse en Composantes Principales (ACP) est introduite dans ce chapitre. Cette méthodologie fournit une approximation, certes sommaire, mais présentant l'avantage d'être complètement définie par quelques valeurs numériques.

Diverses utilisations de cette approximation seront utilisées tout au long de nos travaux. Dans ce chapitre, sont présentées les premières applications développées. Par exemple, en s'appuyant sur cette approximation, la durée d'usinage et la longueur de parcours d'outil peuvent être estimées analytiquement. Dans le même ordre d'idées, l'utilisation de cette approximation permet de définir une méthodologie d'aide au choix du type d'outil et de la direction d'usinage.

Une nouvelle notion de direction principale d'une surface est introduite et définie dans ce chapitre. Cette notion ne doit pas être confondue avec la direction de courbure principale bien connue en géométrie différentielle. Les deux directions sont qualifiées de principales, car résultent de la décomposition propre, mais de deux matrices différentes.

### 2.1 Introduction et motivation

Les surfaces gauches contiennent en général une grande quantité d'information géométrique difficile à résumer pour obtenir une vision globale, ce qui rend leur traitement par ordinateur assez délicat. En effet, se basant principalement sur le calcul des dérivées, la géométrie différentielle permet amplement de décrire les propriétés locales des surfaces gauches. De fait, une vision locale et quantifiable des surfaces gauches est accessible, grâce aux propriétés géométriques locales (telles que le vecteur normal, la courbure normale, la courbure gaussienne, etc). Cependant, aucune information quantitative globale n'est fournie par la géométrie différentielle. En conséquent, sur le plan global, seule une vision qualitative des surfaces gauches est disponible.

Par exemple, un usineur peut qualifier une surface gauche de "compliquée", d'"alambiquée", ou même de "chahutée", pour utiliser un terme plus fleuri. Cette affirmation se base sur l'analyse em-

pirique et purement qualitative des variations de la coordonnée  $S_z$  en fonction des coordonnées  $S_x$  et  $S_y$ . Toutefois, il n'a aucun moyen à sa disposition pour quantifier son observation. De la même façon, il peut énoncer que la surface a plutôt tendance à s'étaler selon une direction donnée, sans avoir aucune valeur précise ni moyen numérique pour quantifier cet "étalement".

L'objectif de ce chapitre est d'introduire l'approximation de surfaces gauches par des plans. Ce qui permet d'appréhender de manière quantitative une surface gauche et de valider les observations de l'usineur par des valeurs numériques. En outre, l'analyse statistique s'avère plus appropriée pour capturer les tendances globales des surfaces et résumer l'information quantitative. En particulier, l'ACP est utilisée pour trouver le plan qui approxime le mieux la surface. Un plan étant défini par deux vecteurs unitaires et deux dimensions (largeur et hauteur), une vision globale des surfaces gauches est quantifiable grâce à ces informations.

Pour mettre tout de suite en évidence l'intérêt d'approximer une surface gauche par un plan, la section suivante 2.2 est consacrée à l'étude de l'usinage des plans dans l'espace 3D : Il y est montré comment le temps et la longueur d'usinage peuvent être calculés analytiquement, et une méthodologie simple d'aide au choix de l'outil et de la direction d'usinage y est développée. Ensuite, l'approximation des surfaces gauches par des plans à l'aide de l'ACP est présentée dans la section 2.3. Dans la section 2.4, la procédure de choix d'outil est généralisée aux surfaces gauches. Elle est ensuite appliquée sur des cas tests. L'efficacité et les limites de cette approche sont discutées dans la section 2.5. Une conclusion du chapitre est énoncée dans la section 2.6 ainsi que des futurs axes de recherche. D'autres applications de cette approximation basée sur l'ACP sont présentées dans les chapitres suivants.

## 2.2 Étude des surfaces planes

Dans cette section, l'usinage de surfaces planes inclinées est analysé afin de mettre en évidence l'intérêt d'utiliser de telles surfaces pour approximer les surfaces gauches. De plus, la problématique du choix de l'outil, qui constitue une application simple de cette approximation, est introduite pour les surfaces planes rectangulaires.

L'efficacité de l'outil torique et de l'outil sphérique est étudiée sur une surface plane rectangulaire, en fonction de la plus forte pente  $s$  et du facteur de forme largeur/longueur  $\mu$ . Afin de calculer le temps d'usinage, le modèle cinématique de la machine CNC présenté dans la section 1.6 est utilisé.

La stratégie d'usinage utilisée ici est celle des plans parallèles "zigzag", de sorte qu'une trajectoire d'outil est composée d'une séquence de segments linéaires reliés bout à bout (code CN : G01). Dans la suite, les paramètres cinématiques sont fixés à :  $J_{max} = 40 \text{ m/s}^3$ ,  $A_{max} = 6 \text{ m/s}^2$ ,  $V = 5 \text{ m/min}$ , et les outils sont définis par le rayon extérieur  $R = 5 \text{ mm}$ , et le rayon du tore  $r = 2 \text{ mm}$  pour la fraise torique. La contrainte de hauteur de crête est  $sh = 0.01 \text{ mm}$ .

### 2.2.1 Problématique de choix d'outil

Il est connu que l'utilisation de l'outil torique selon la direction de plus grande pente maximise le rayon effectif à travers toute la surface, ce qui entraîne des pas transversaux plus importants entre les trajectoires adjacentes, réduisant ainsi la longueur totale du chemin parcouru. Cependant, si la direction de la plus grande pente est proche de la perpendiculaire à la direction de la plus grande dimension de la surface, le parcours généré sera composé de trajectoires plus courtes. Dans ce cas, l'utilisation d'un outil sphérique dans la direction de la plus grande dimension de la surface, appelée ci-après direction principale, peut donner de meilleurs résultats en termes de temps d'usinage, malgré un chemin parcouru potentiellement plus long. En effet, le comportement cinématique de l'outil, qui doit décélérer à l'approche de chaque angle aigu du parcours puis accélérer à nouveau une fois l'angle dépassé, est beaucoup plus pénalisant dans le premier cas que dans le second.

Pour illustrer ce constat, considérons une surface plane rectangulaire de largeur  $w = 56$  mm et de hauteur  $h = 28$  mm ; l'angle de sa pente est de  $60^\circ$  et la direction de la plus forte pente est perpendiculaire à la direction principale (figure 2.1). Cette surface plane est d'abord usinée à l'aide d'un outil torique de rayons  $R = 5$  mm et  $r = 2$  mm selon la direction de la pente (située dans le plan  $(\mathbf{X}, \mathbf{Z})$ ). Ensuite, la même surface est usinée à l'aide d'un outil à bout sphérique de rayon  $R = 5$  mm dans la direction principale (selon l'axe  $\mathbf{Y}$ ). La hauteur de crête maximale autorisée est de  $sh = 0,01$  mm, ce qui est une valeur standard dans les applications industrielles.

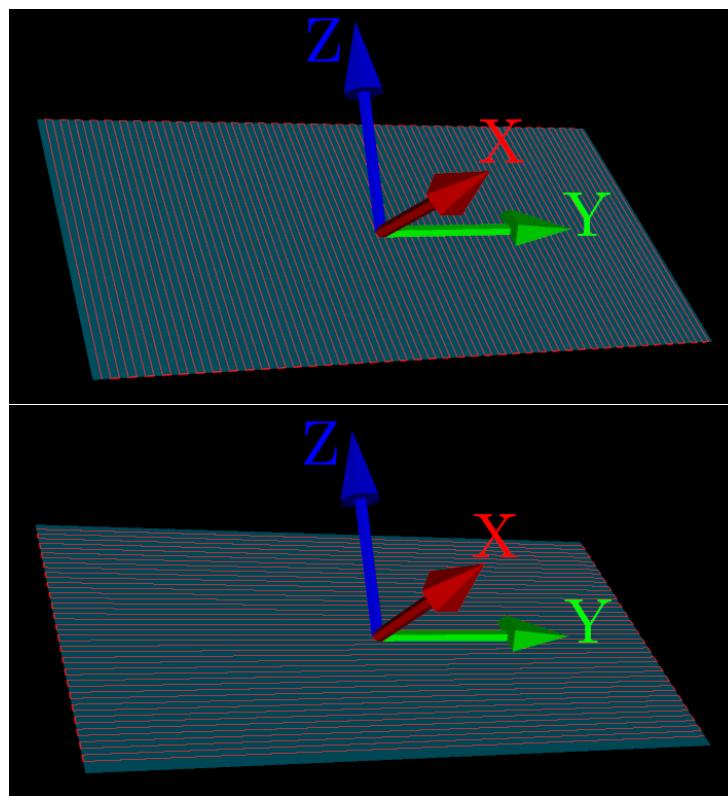


Figure 2.1 – Usinage d'une surface plane rectangulaire : selon la direction de plus grande pente avec un outil torique (en haut), selon la direction principale avec un outil sphérique (en bas)

Les résultats en termes de rayon effectif, de temps d’usinage et de longueur du parcours d’outil sont indiqués dans le tableau 2.1. La longueur du parcours est la même, mais il est intéressant de noter que même si le rayon effectif est 9,2% plus grand dans le premier cas (usinage avec un outil torique dans la direction de la pente la plus forte), le gain obtenu en usinant dans la direction principale avec un outil sphérique d’est environ 16% en termes de temps d’usinage.

direction d’usinage	rayon effectif	temps d’usinage	longueur parcourue
plus forte pente	5.46 mm	43.8 s	2460 mm
principale	5 mm	36.8 s	2459 mm

TABLE 2.1 – Résultats de la surface plane usinée à l’aide d’une fraise torique selon la plus grande pente et d’une fraise sphérique selon la direction principale

Pour approfondir les tests, un facteur de forme  $\mu$  est défini pour le plan tel que  $\mu = \frac{w}{h}$ , où  $w$  est la largeur de la surface et  $h$  ( $\leq w$ ) est la hauteur. En fonction de cela, une taille caractéristique  $\chi$  peut être définie de telle sorte que :

$$\chi = wh = \frac{w^2}{\mu} = h^2 \mu \quad (2.1)$$

Différents tests sont effectués pour différentes valeurs de la pente  $s$  et ce facteur de forme  $\mu$  pour une taille caractéristique constante. Comme prévu, l’outil torique a tendance à présenter de meilleurs temps d’usinage lorsque la pente diminue vers  $0^\circ$  et le facteur de forme s’approche de 1. À l’inverse, l’outil sphérique tend à être plus efficace lorsque la pente augmente vers  $90^\circ$  et que le facteur de forme est élevé.

En fait, le paramètre  $\mu$  représente l’allongement de la surface, tandis que  $\chi$  indique sa taille et sa superficie. Comme indiqué précédemment, la pente  $s$  et le facteur de forme  $\mu$  sont les paramètres les plus influents sur le temps d’usinage. Par conséquent, afin de se concentrer sur ces paramètres, la taille caractéristique  $\chi$  est considérée comme constante dans ce qui suit.

Sur la base de ces observations, une question bien fondée se pose : est-il possible de définir des valeurs critiques pour les paramètres  $s$  et  $\mu$  qui correspondent à un temps d’usinage égal pour les deux processus d’usinage ? Si oui, ces valeurs critiques correspondraient au cas limite où les deux outils sont également efficaces. Par conséquent, pour un cas pratique donné, la comparaison des valeurs réelles de ces paramètres pour la surface avec ces valeurs critiques permettrait de prévoir quel outil sera le plus efficace. Dans la section suivante, cette question est étudiée pour une surface plane.

## 2.2.2 Développement analytique

Dans cette section, le temps d’usinage est analytiquement calculé pour les surfaces planes rectangulaires dans chacun des deux cas de figure : usinage selon la direction de plus grande pente avec outil torique, et selon la direction principale avec outil sphérique. Les paramètres critiques, qui permettent de prédire quel cas de figure présente le meilleur temps d’usinage, sont introduits et calculés.

### 2.2.2.1 Outil torique selon la direction de plus grande pente

Dans cette partie, le temps d'usinage avec un outil torique dans la direction de plus grande pente (image supérieure de la figure 2.1) est calculé en fonction de la géométrie de la fraise, des paramètres cinématiques et des paramètres de la surface. Pour tout point de la surface, le rayon effectif de l'outil torique peut être calculé à l'aide de l'équation 1.5 avec  $\alpha = 0$  :

$$R_{eff} = \frac{R - r}{\sin s} + r. \quad (2.2)$$

Le pas transversal est donné par :

$$sod_1 = 2\sqrt{2shR_{eff} - sh^2} \quad (2.3)$$

Par la suite, le nombre de trajectoires  $n_1$  est obtenu par :

$$n_1 = \frac{w}{sod_1} = \frac{\sqrt{\mu\chi}}{sod_1}.$$

Le parcours d'outil est égal à la somme des trajectoires plus les pas transversaux pour passer d'une trajectoire à la suivante :

$$\begin{aligned} L_1 &= n_1(h + sod_1) = n_1 \sqrt{\frac{\chi}{\mu}} + n_1 sod_1 \\ &= \frac{\sqrt{\mu\chi}}{sod_1} \sqrt{\frac{\chi}{\mu}} + \frac{\sqrt{\mu\chi}}{sod_1} sod_1 \\ &= \frac{\chi}{\sqrt{2sh\frac{R-r}{\sin s} + 2shr - sh^2}} + \sqrt{\mu\chi} \end{aligned}$$

Afin de calculer le temps d'usinage, le cas cinématique réel doit être trouvé (voir section 1.6). Ceci dépend de la valeur de  $\chi$ , puisque  $t_A$  et  $t_V$  sont connus :

$$t_A = \frac{A_{max}}{J_{max}} = \frac{6}{40} = 0.15 \text{ s}$$

et

$$t_V = \sqrt{\frac{V}{J_{max}}} = \sqrt{\frac{5}{60 \times 40}} = 0.045 \text{ s.}$$

Vu que  $t_V < t_A$ , le cas 1 est rejeté, et le cas 2 s'applique si  $t_V < t_l$ . En considérant  $t_l = \sqrt[3]{\frac{l}{J_{max}}}$  avec  $l = \frac{h}{2} = \frac{1}{2} \sqrt{\frac{\chi}{\mu}}$ , cette dernière condition est équivalente à :

$$\frac{V^3}{J_{max}^3} < \frac{\chi}{4\mu J_{max}^2},$$

d'où :

$$h = \sqrt{\frac{\chi}{\mu}} > 2 \sqrt{\frac{V^3}{J_{max}}} = 7.6 \text{ mm.}$$

De la même façon, puisque  $t_V < t_A$ , le cas 3 est équivalent à  $t_l < t_V$ , donc  $h < 7.6$  mm. Cette valeur peut être considérée comme trop faible pour les surfaces usinables. C'est pourquoi, dans ce qui suit, seul le cas 2 sera pris en compte. Le temps d'usinage pour une seule trajectoire est donc (voir équation 1.14) :

$$t_p = \frac{h}{V} + 2 \sqrt{\frac{V}{J_{max}}} = \frac{\sqrt{\chi}}{\sqrt{\mu} V} + 2 \sqrt{\frac{V}{J_{max}}}.$$

Le temps d'usinage nécessaire pour passer d'une trajectoire à la suivante doit également être calculé. Il dépend du pas transversal. La cinématique de la machine correspond au cas 3 si  $sod_1 < 7.6$  mm, ce qui est toujours vrai puisque dans le pire des cas ( $s = 0^\circ$ ) on avait  $sod_1 = 2(R - r) + 2\sqrt{r^2 - (r - sh)^2} = 6.4$  mm. On suppose donc le cas 3 pour le calcul du temps d'usinage entre deux trajectoires adjacentes (voir équation 1.15) :

$$t_{ip} = 4 \sqrt[3]{\frac{sod_1}{2J_{max}}}.$$

Finalement, le temps d'usinage total s'écrit :

$$T_1 = n_1 (t_p + t_{ip}) = \frac{\sqrt{\mu \chi}}{sod_1} \left( \frac{\sqrt{\chi}}{\sqrt{\mu} V} + 2 \sqrt{\frac{V}{J_{max}}} + 4 \sqrt[3]{\frac{sod_1}{2J_{max}}} \right). \quad (2.4)$$

### 2.2.2.2 Outil sphérique selon la direction principale

Dans cette section, la surface est usinée avec une fraise à bout sphérique dans la direction principale (axe **Y**, dans la figure 2.1), qui est orthogonale à la direction de la plus grande pente (axe **X**, dans la figure 2.1), et le temps d'usinage est calculé. Dans ce cas, le rayon effectif est de  $R_{eff} = R$ . Il est à noter que dans ce cas, la fraise torique présentera un rayon effectif minimal de  $r$  selon cette direction. Ainsi, dans ce cas, la fraise sphérique est beaucoup plus efficace que la fraise torique. Le pas transversal s'écrit :

$$sod_2 = 2\sqrt{2shR - sh^2}.$$

Le nombre de trajectoires dans ce cas est de :

$$n_2 = \frac{h}{sod_2} = \frac{\sqrt{\chi}}{\sqrt{\mu} \times sod_2} = \frac{\sqrt{\chi}}{2\sqrt{\mu} \sqrt{2shR - sh^2}}.$$

Là encore, la longueur du parcours est égale à la somme des trajectoires plus les pas transversaux pour passer d'une trajectoire à la suivante :

$$L_2 = n_2 \left( \sqrt{\frac{\chi}{\mu}} + 2\sqrt{2shR - sh^2} \right).$$

De façon identique à la section précédente, la cinématique de la machine est représentée par le cas 2. Ainsi, le temps d'usinage correspondant à une trajectoire s'écrit :

$$t_p = \frac{\sqrt{\chi \mu}}{V} + 2 \sqrt{\frac{V}{J_{max}}}.$$

Et la durée nécessaire pour passer d'une trajectoire à la suivante, modélisée avec le cas 3 de la cinématique, s'écrit :

$$t_{ip} = 4 \sqrt[3]{\frac{sod_2}{2J_{max}}}.$$

Finalement, la durée total d'usinage est :

$$T_2 = n_2 (t_p + t_{ip}) = \frac{\sqrt{\chi}}{\sqrt{\mu} \times sod_2} \left( \frac{\sqrt{\chi \mu}}{V} + 2 \sqrt{\frac{V}{J_{max}}} + 4 \sqrt[3]{\frac{sod_2}{2J_{max}}} \right). \quad (2.5)$$

### 2.2.2.3 Valeurs critiques des paramètres facteur de forme et pente

Dans les sections précédentes (2.2.2.1 et 2.2.2.2), les longueurs des parcours d'outils ( $L_i$ ) et les temps d'usinage ( $T_i$ ) sont calculés pour l'usinage avec outil torique dans la direction de plus forte pente ( $L_1$  et  $T_1$ ), et l'usinage avec outil sphérique dans la direction principale ( $L_2$  et  $T_2$ ). Cette section se concentre sur la comparaison des temps d'usinage  $T_1$  et  $T_2$ . Cette comparaison est essentiellement effectuée en étudiant le cas limite  $T_1 = T_2$ . Remarquez que  $T_1 = T_2$  est équivalent à :

$$\frac{\sqrt{\mu \chi}}{sod_1} \left( \frac{\sqrt{\chi}}{\sqrt{\mu} V} + 2 \sqrt{\frac{V}{J_{max}}} + 4 \sqrt[3]{\frac{sod_1}{2J_{max}}} \right) = \frac{\sqrt{\chi}}{\sqrt{\mu} sod_2} \left( \frac{\sqrt{\chi \mu}}{V} + 2 \sqrt{\frac{V}{J_{max}}} + 4 \sqrt[3]{\frac{sod_2}{2J_{max}}} \right)$$

Dans l'équation ci-dessus, les paramètres cinématiques  $J_{max}$ ,  $A_{max}$  et  $V$ , ainsi que les rayons  $R$  et  $r$ , sont connus ; le pas transversal  $sod_2$  est calculée à partir du rayon  $R$  et de la hauteur maximale de crête  $sh$ , donc  $sod_2$  est également connu ; et la taille caractéristique  $\chi$  est considérée comme constante (voir la section 2.2.1). Par conséquent, les seuls paramètres influents sont le facteur de forme  $\mu$  et la pente  $s$  qui apparaît dans l'expression du rayon effectif  $R_{eff}$  utilisé pour le calcul de  $sod_1$ .

Lorsque le facteur de forme  $\mu$  augmente, à valeur de  $\chi$  constante, le nombre de trajectoires obtenues avec un outil torique augmente en conséquence (les trajectoires sont plus courtes), tandis que son homologue avec un outil sphérique diminue (et les trajectoires deviennent plus longues), puisque "l'allongement" se fait dans la direction principale, qui est orthogonale à la direction de la pente la plus forte. Ainsi, l'augmentation de  $\mu$  tend à favoriser, du point de vue cinématique, l'usinage avec un outil à bout sphérique dans la direction principale.

D'autre part, lorsque la pente tend vers 0, le rayon effectif de l'outil torique tend vers l'infini, ce qui maximise le pas transversal et favorise l'usinage avec un outil torique dans la direction de la plus grande pente.

Afin de déterminer quel type d'outil et quelle direction d'usinage sont les meilleurs, une analyse de ces paramètres d'influence doit être effectuée. Cela peut se faire soit en exprimant la pente  $s$  en fonction du facteur de forme  $\mu$ , soit en exprimant le facteur de forme  $\mu$  en fonction de la pente  $s$ . D'un point de vue pratique, il peut être plus utile d'exprimer la pente  $s$  en fonction du facteur de forme  $\mu$ . En effet, pour une surface donnée, le facteur de forme est déjà défini, tandis que la pente  $s$  peut encore être modifiée, en définissant l'orientation de la pièce dans l'espace de travail de la machine-outil. Par conséquent, l'expression de la pente  $s$  en fonction du facteur de forme  $\mu$  est présentée ci-après, tandis que l'expression du facteur de forme  $\mu$  en fonction de la pente  $s$  est présentée dans l'annexe A.1.

Le travail sur l'expression de la pente  $s$  en fonction du facteur de forme  $\mu$ , vise à faire apparaître une valeur critique de  $s$ , ci-après appelée  $s_c$ , qui différencie le cas où l'usinage avec un outil torique conduit à de meilleurs résultats que son homologue sphérique. Pour un facteur de forme donné  $\mu$ , la pente critique  $s_c$  est définie de manière à ce que les temps d'usinage  $T_1$  et  $T_2$  soient égaux. Là encore,  $T_1 = T_2$  équivaut à :

$$\frac{\sqrt{\mu \chi}}{sod_1} \left( \frac{\sqrt{\chi}}{\sqrt{\mu} V} + 2 \sqrt{\frac{V}{J_{max}}} + 4 \sqrt[3]{\frac{sod_1}{2J_{max}}} \right) = T_2,$$

ou encore  $\frac{T_2}{\sqrt{\mu \chi}} sod_1 - \frac{4}{\sqrt[3]{2J_{max}}} \sqrt[3]{sod_1} - \frac{\sqrt{\chi}}{\sqrt{\mu} V} - 2 \sqrt{\frac{V}{J_{max}}} = 0$ .

Cette dernière équation est un polynôme de troisième degré en  $z = \sqrt[3]{sod_1}$  de la forme :

$$z^3 - pz - q = 0, \quad (2.6)$$

où :

$$p = \frac{4 \sqrt{\mu \chi}}{T_2 \sqrt[3]{2J_{max}}}, \quad (2.7)$$

$$\text{et } q = \frac{\sqrt{\mu \chi}}{T_2} \left( \frac{\sqrt{\chi}}{\sqrt{\mu} V} + 2 \sqrt{\frac{V}{J_{max}}} \right) \quad (2.8)$$

Il est facile de voir que  $p$  et  $q$  sont positifs. La formule de Cardan donne la solution générale de l'équation (2.6). Le discriminant de cette équation est :

$$\begin{aligned} \Delta &= 4p^3 - 27q^2 \\ &= \frac{\mu \chi}{T_2^2} \left( 128 \frac{\sqrt{\mu \chi}}{J_{max} T_2} - 27 \left( \frac{\sqrt{\chi}}{\sqrt{\mu} V} + 2 \sqrt{\frac{V}{J_{max}}} \right)^2 \right). \end{aligned}$$

En utilisant les valeurs numériques de  $V$  et  $J_{max}$ , en limitant le facteur de forme à 9<sup>1</sup> (c'est à dire

1. Au delà de  $\mu = 9$ , les surfaces sont tellement allongées que l'usinage dans la direction principale est sans doute la

$\sqrt{\mu} < 3$ ), et en supposant les surfaces assez grandes de telle sorte que l'ordre de grandeur du temps d'usinage soit supérieur à une seconde ( $T_2 > 1$  s), il vient que :

$$\Delta = \frac{\mu \chi}{T_2^2} \left( 3.2 \frac{\sqrt{\mu \chi}}{T_2} - 27 \left( 12 \frac{\sqrt{\chi}}{\sqrt{\mu}} + 0.1 \right)^2 \right).$$

ainsi,

$$\Delta < \frac{\mu \chi}{T_2^2} \left( 9.6 \sqrt{\chi} - 27 (4 \sqrt{\chi} + 0.1)^2 \right)$$

La fonction  $\chi \mapsto 9.6 \sqrt{\chi} - 27 (4 \sqrt{\chi} + 0.1)^2$  est négative pour  $\sqrt{\chi} > 0.01$ . Par conséquent,  $\Delta < 0$  pour les surfaces de d'aire supérieure à  $100 \text{ mm}^2$ ; cette hypothèse est largement acceptable pour les surfaces couramment usinées dans l'industrie. Ainsi, l'équation (2.6) comporte une solution réelle (et deux solutions complexes conjuguées) :

$$z = \sqrt[3]{\frac{-q + \sqrt{\frac{-\Delta}{27}}}{2}} + \sqrt[3]{\frac{-q - \sqrt{\frac{-\Delta}{27}}}{2}}. \quad (2.9)$$

En utilisant les équations (2.2) et (2.3), l'expression de la pente critique  $s_c(\mu)$  pour un facteur de forme donnée  $\mu$  peut être obtenue à partir de :

$$\begin{aligned} sod_1^2 = z^6 &\Leftrightarrow 4 (2shR_{eff} - sh^2) = z^6 \\ &\Leftrightarrow R_{eff} = \frac{sh}{2} - \frac{z^6}{8sh} = \frac{R - r}{\sin s_c} + r, \end{aligned}$$

et enfin :

$$s_c(\mu) = \arcsin \left( \frac{8sh(R - r)}{z^6 + 4sh^2 - 8shr} \right) \quad (2.10)$$

où  $\mu$  apparaît dans les termes  $p$  et  $q$  (équations (2.7) et (2.8)) de la solution  $z$  établie dans l'équation (2.9).

Si  $s < s_c$  alors  $\sqrt[3]{sod_1}$  est supérieur à la solution de l'équation (2.6) et  $z^3 - pz - q > 0$ , ce qui signifie que  $T_2 > T_1$ . Dans ce cas, il est préférable d'utiliser un outil torique dans le sens de la plus forte pente. Inversement, si  $s > s_c$ , alors  $\sqrt[3]{sod_1}$  est inférieur à la solution de l'équation (2.6) et  $z^3 - pz - q < 0$ , ce qui signifie que  $T_2 < T_1$ . Dans ce cas, l'usinage à l'aide d'un outil sphérique dans la direction principale est plus efficace. Ce résultat est cohérent avec le fait que le rayon effectif (et le pas transversal  $sod_1$ ) de l'outil torique diminue à mesure que la pente  $s$  augmente.

### 2.2.3 Validation numérique

L'expression analytique du paramètre critique  $s_c$  est donnée par l'équation (2.10). Cette section est consacrée à la validation de cette expression à l'aide de simulations numériques.

meilleure solution.

### 2.2.3.1 Considérations préliminaires

Le processus de simulation d'usinage utilisé pour calculer la trajectoire d'outil pour une surface paramétrique donnée sous la forme d'une séquence de points d'interpolation est basé sur l'intersection de courbes isoparamétriques et de plans parallèles verticaux. Ce processus est largement testé et efficace.

Tout d'abord, pour revenir à l'exemple donné dans la section 2.2.1, la pente est de  $s = 60^\circ$  alors que le facteur de forme est  $\mu = \frac{56}{28} = 2$ . En utilisant l'équation (2.10), la pente critique peut être évaluée :  $s_c(\mu) = 33,8^\circ$ . Ainsi,  $s_c < s$ , ce qui est cohérent avec le fait que le temps d'usinage à l'aide d'une fraise à bout sphérique le long de la direction principale est plus court (table 2.1). Comme le montre cet exemple, la procédure proposée est en mesure de fournir un bon indicateur pour le choix de l'outil à utiliser, afin d'obtenir les meilleurs résultats. En fait, grâce à la procédure basée sur les paramètres critiques, il est possible de prévoir le type d'outil, ce qui permet de réduire le temps d'usinage sans avoir à effectuer de simulations d'usinage.

Pour une meilleure compréhension de l'influence de chaque paramètre, l'évolution des temps d'usinage par rapport au facteur de forme  $\mu$ , pour des valeurs fixes de  $s$ , est affichée dans la figure 2.2.

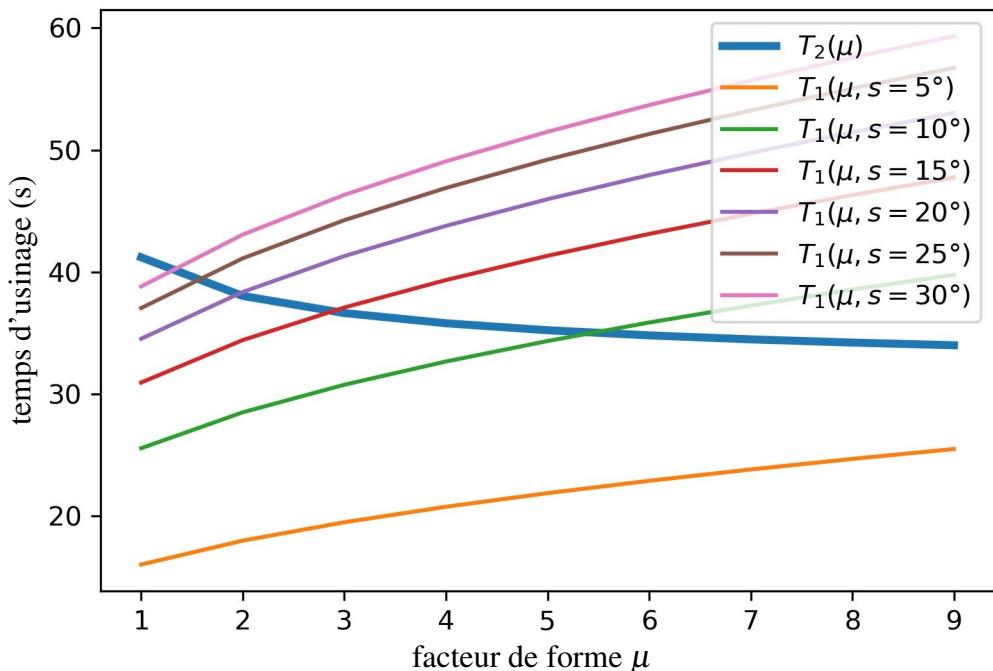


Figure 2.2 – Évolution des temps d'usinage par rapport au facteur de forme  $\mu$  pour différentes valeurs de la pente  $s$

Comme l'usinage avec un outil sphérique ne dépend pas de la pente  $s$ , une seule courbe est tracée pour le temps d'usinage correspondant ( $T_2$ ). Comme prévu, le temps d'usinage  $T_2$  diminue avec le facteur de forme  $\mu$ , tandis que le temps d'usinage  $T_1$  augmente avec le facteur de forme  $\mu$ . Plus la pente  $s$  est élevée, plus le temps  $T_1$  est élevé.

### 2.2.3.2 Validation de l'approximation analytique du temps d'usinage

Pour analyser plus en profondeur la fiabilité de l'indication donnée par la procédure des paramètres critiques, il convient d'abord d'analyser l'erreur dans les calculs analytiques du temps d'usinage ( $T_1$  et  $T_2$ ). Pour ce faire, plusieurs simulations ont été réalisées, pour différentes paires de valeurs  $s$  et  $\mu$ , constituant un plan d'expérience factoriel tel que  $s \in [0^\circ, 90^\circ]$  et  $\mu \in [1, 10]$ . Pour chaque paire de valeurs, les temps d'usinage de surfaces planes inclinées sont évalués tant analytiquement ( $T_1$  et  $T_2$ ) que numériquement ( $T_1^*$  et  $T_2^*$ , calculés par simulation d'usinage). L'erreur relative (équation 2.11) est tracée dans la figure 2.3.

$$\varepsilon_i = \frac{|T_i^* - T_i|}{\max(T_i^*, T_i)} \quad \text{pour } i = 1, 2 \quad (2.11)$$

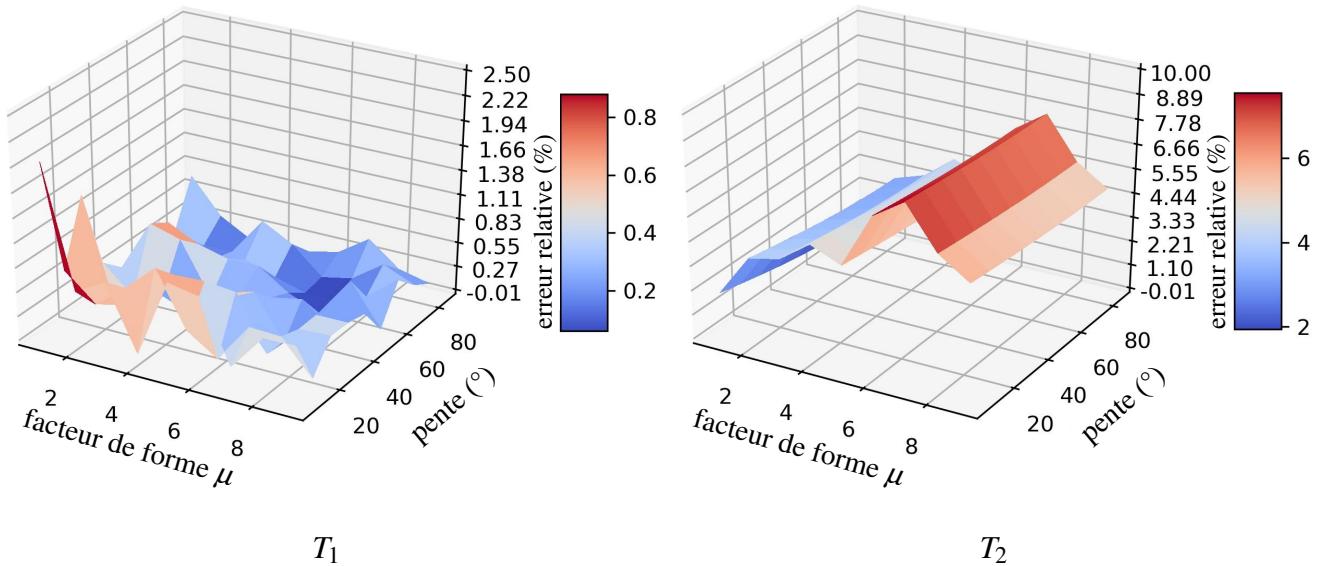


Figure 2.3 – Erreurs relatives des temps d'usinage  $T_1$  et  $T_2$  en fonction des paramètres  $s$  et  $\mu$

L'erreur relative moyenne pour le temps d'usinage  $T_1$  est de  $\bar{\varepsilon}_1 = 0,36\%$ , tandis que pour  $T_2$  elle est de  $\bar{\varepsilon}_2 = 4,75\%$ . Comme prévu, l'erreur  $\varepsilon_2$  ne dépend pas de la pente  $s$ , puisque l'usinage d'une surface plane avec un outil sphérique n'est pas affecté par l'angle d'inclinaison de la surface. Cette erreur est principalement due à l'approximation du nombre de trajectoires par la largeur (ou la hauteur, selon la direction d'usinage) divisée par le pas transversal, ce qui ne donne pas un nombre entier. En outre, le processus de simulation numérique de l'usinage utilise de petites marges par rapport aux limites initiales et finales pour définir les premiers et derniers trajets. Par conséquent, le temps d'usinage analytique sur-estime son homologue numérique de presque le temps correspondant à une trajectoire. Cela expliquerait également le fait que l'erreur  $\varepsilon_2$  est supérieure à  $\varepsilon_1$  puisque la longueur d'une seule trajectoire est plus importante dans le second cas. Enfin, le calcul analytique des temps d'usinage  $T_1$  et  $T_2$  est considéré comme suffisamment précis pour donner une bonne estimation du temps d'usinage réel pour les outils et les directions.

### 2.2.3.3 Validation des paramètres critiques

Sur la base des considérations précédentes, le paramètre critique  $s_c$  peut être analysé numériquement. Par ailleurs, de la même manière que le paramètre critique réciproque  $\mu_c$  a été défini dans l'appendice A.1, l'analyse réciproque concernant ce paramètre est présentée dans l'appendice A.2.

Pour effectuer une analyse du paramètre  $s_c$ , un échantillon de valeurs du facteur de forme  $\mu$  est considéré et la valeur critique  $s_c$  est calculée pour chaque  $\mu$  de cet échantillon. Pour chaque couple  $(\mu, s_c(\mu))$ , les simulations d'usinage sont ensuite effectuées pour fournir les temps d'usinage  $T_1^*$  et  $T_2^*$ . Compte tenu de cette méthodologie,  $T_1^*$  et  $T_2^*$  devraient alors être égaux pour chaque couple  $(\mu, s_c(\mu))$ . La figure 2.4 montre l'évolution de  $s_c(\mu)$  et l'erreur relative de la différence entre les temps  $T_1^*(\mu)$  et  $T_2^*(\mu)$ .

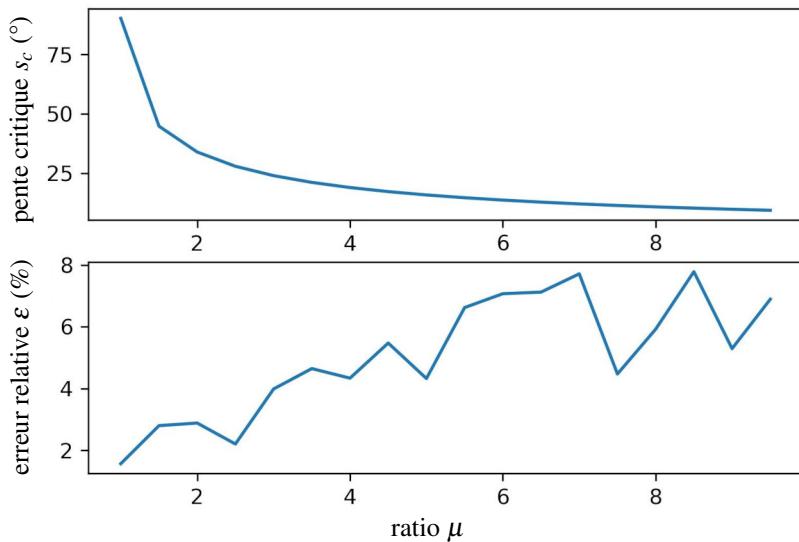


Figure 2.4 – Évolution de la pente critique  $s_c(\mu)$  et de l'erreur relative du temps d'usinage  $\varepsilon(\mu)$  par rapport au facteur de forme  $\mu$

L'erreur est d'environ 5%, ce qui confirme la validité des expressions des paramètres critiques. Il convient de noter que l'erreur est plus élevée pour les facteurs de forme plus importants. En effet, dans ce cas, les trajectoires dans la direction principale sont plus longues, et l'approximation sur le nombre de chemins a un impact plus important.

En résumé, les temps d'usinage calculés analytiquement et les paramètres critiques ont été vérifiés numériquement. Ainsi, l'approche proposée permet de sélectionner, pour une surface plane donnée, le type d'outil et la direction d'usinage les plus efficaces sans calculer les temps d'usinage, mais en utilisant uniquement la pente et le facteur de forme, ce qui est simple pour les surfaces rectangulaires et planes. Pour ce faire, étant donné un plan, sa pente  $s$  et son facteur de forme  $\mu$ , la pente  $s$  peut être comparée à la pente critique  $s_c(\mu)$ . Ensuite, l'outil torique doit être privilégié si  $s < s_c(\mu)$ . Sinon, il faut choisir l'outil sphérique.

## 2.3 Approximation des surfaces gauches

L'objectif de cette section est d'approximer les surfaces gauches par des plans à l'aide de l'ACP. Cette approximation permet d'étendre la procédure de choix d'outil définie précédemment sur les surfaces planes rectangulaires aux surfaces gauches. Dans un premier temps, on se limitera à approximer une surface gauche par un plan, mais il est tout à fait possible d'envisager des approximations plus précises (et plus complexes).

En montrant comment il est possible dans le cas de plans inclinés, d'offrir une aide au choix d'outil à partir de simples développements analytiques, cette étude permet d'envisager une démarche similaire pour les surfaces gauches. En effet, en approximant une surface gauche par un plan, il sera possible d'étendre cette méthodologie à de telles surfaces (section 2.4). Or, autant cette démarche ne présente que peu d'intérêt pratique pour les plans qui ne sont, pour ainsi dire, jamais usinés dans telles conditions, autant elle présente un intérêt certain pour les surfaces gauches. C'est là une première application pratique de l'approximation des surfaces gauches par des plans.

### 2.3.1 Considérations préliminaires

Pour mettre en œuvre une procédure d'analyse en composantes principales sur une surface gauche, on considère un maillage uniforme en  $(u, v)$  défini par des courbes isoparamétriques. De cette façon, on obtient un ensemble de mailles (quads de surface élémentaires). Ensuite, pour chaque maille, un point échantillon est défini comme étant le centre de la maille. Les directions d'usinage sont définies à l'aide de ces points échantillons désignés par  $\mathbf{S}_i$ , pour  $i \in \{0, \dots, n\}$ .

Dans le cas d'un plan, la direction de plus forte pente est la même quel que soit le point de la surface considérée. Dans le cas d'une surface gauche, cette direction peut varier pour tout point pris en considération. Dans un premier temps, la direction de plus forte pente d'une surface gauche est estimée par la moyenne des directions de plus forte pente calculées en chaque point de l'échantillon (section 2.3.4).

Pour une surface rectangulaire et plane, la direction principale, le long de laquelle la surface est la plus étendue, est parallèle à sa plus grande frontière. Elle peut également être considérée, pour toute surface, comme la direction de la ligne qui minimise la somme des carrés des distances entre les points de la surface et leur projection orthogonale sur cette ligne. En utilisant le maillage de la surface précédemment définie, cette somme peut être calculée de manière discrète, en s'appuyant sur les points de l'échantillon  $\mathbf{S}_i$ .

D'après ces considérations, la direction principale correspond à la droite qui minimise la somme des carrés des distances  $d_i^2$  entre les points d'échantillonnage  $\mathbf{S}_i$  et leur projection orthogonale sur cette ligne. Ce principe est illustré sur la figure 2.5 pour quelques points d'échantillonnage. Ainsi, trouver la direction principale est équivalent à la célèbre Analyse en Composantes Principales (ACP) [1] qui fournit la "meilleure régression" pour un nuage de points.

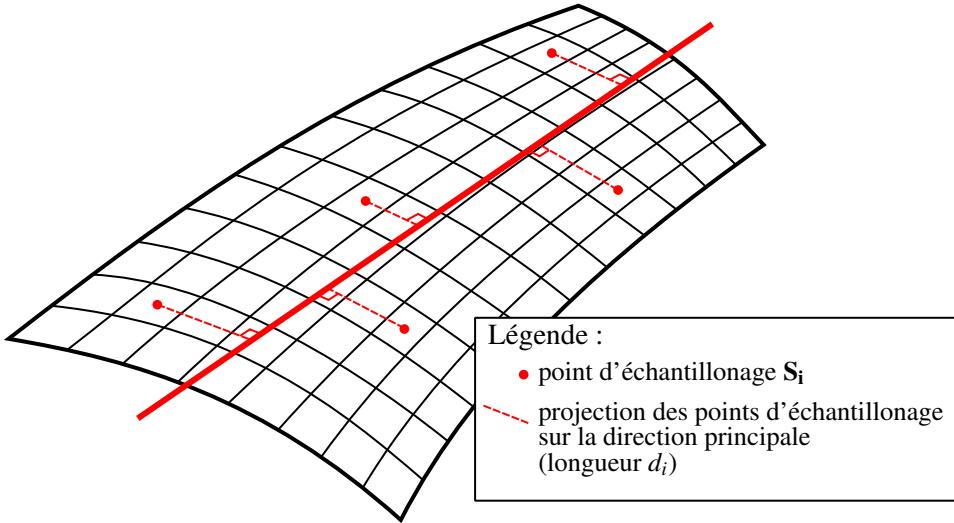


Figure 2.5 – Évaluation de la direction principale pour une surface gauche : Principe

### 2.3.2 Utilisation de la matrice de covariance pour définir la direction principale

Dans notre contexte, pour une surface donnée  $\mathcal{S}$ , les variables sont les coordonnées des points de surface dans l'espace euclidien à trois dimensions :  $(S_x, S_y, S_z)$ . L'ACP est effectuée sur les points  $S_i$  échantillons par décomposition propre de sa matrice de covariance, ce qui donne trois vecteurs propres orthogonaux  $S_I$ ,  $S_{II}$  et  $S_{III}$  correspondant aux valeurs propres  $\lambda_I \geq \lambda_{II} \geq \lambda_{III}$ . Notez que  $S_I$  est le vecteur qui correspond le mieux aux points  $S_i$  de l'échantillon. En effet, il minimise la somme des carrés des distances entre les points d'échantillonnage et la projection orthogonale sur lui-même. Ainsi,  $S_I$  est considéré comme la direction principale de la surface gauche  $\mathcal{S}$ . La matrice de covariance  $C$  des points  $S_i$  de l'échantillon s'écrit :

$$C = \begin{pmatrix} \text{Var}(S_x) & \text{Cov}(S_x S_y) & \text{Cov}(S_x S_z) \\ \text{Cov}(S_x S_y) & \text{Var}(S_y) & \text{Cov}(S_y S_z) \\ \text{Cov}(S_x S_z) & \text{Cov}(S_y S_z) & \text{Var}(S_z) \end{pmatrix}, \quad (2.12)$$

avec  $\begin{cases} \text{Var}(S_x) = E(S_x^2) - E(S_x)^2 \\ \text{Var}(S_y) = E(S_y^2) - E(S_y)^2 \\ \text{Var}(S_z) = E(S_z^2) - E(S_z)^2 \end{cases}$  et  $\begin{cases} \text{Cov}(S_x S_y) = E(S_x S_y) - E(S_x) E(S_y) \\ \text{Cov}(S_x S_z) = E(S_x S_z) - E(S_x) E(S_z) \\ \text{Cov}(S_y S_z) = E(S_y S_z) - E(S_y) E(S_z) \end{cases}$

Dans ces expressions,  $E(X)$  est la moyenne de la quantité  $X$  sur les  $n$  points d'échantillonnage définis précédemment. Les vecteurs propres sont orthogonaux car la matrice de covariance est symétrique. Comme l'outil doit rester tangent à la surface pendant l'usinage, et que les surfaces sont des variétés géométriques de dimension 2, le vecteur propre  $S_{II}$  représente la direction de la surface la moins étendue, ce qui explique que les meilleurs résultats (en termes de prédiction du type d'outil à l'aide des paramètres critiques) seront obtenus lorsque la direction de plus forte pente est proche de la perpendiculaire à la

direction principale  $\mathbf{S}_I$  (d'autres considérations sur ce point sont abordées dans la section 2.5.3.1). Le vecteur propre  $\mathbf{S}_{III}$  n'est pas pris en compte dans un premier temps. En général, plus  $\lambda_{III}$  est petit, plus la surface est plane.

### 2.3.3 Utilisation des valeurs propres pour définir un facteur de forme approximatif d'une surface gauche

Il convient de noter que les vecteurs propres  $\mathbf{S}_I$  et  $\mathbf{S}_{II}$  génèrent le plan qui correspond le mieux à la surface  $\mathcal{S}$  (cela doit être compris en termes de minimisation de la somme des carrés des distances entre les points d'échantillonnage  $\mathbf{S}_i$  et leur projection orthogonale sur ce plan). Les valeurs propres  $\lambda_I$  et  $\lambda_{II}$  représentent la variance des points projetés sur  $\mathbf{S}_I$  et  $\mathbf{S}_{II}$ , respectivement. On sait que la projection d'une surface continue sur un vecteur est un segment continu dont la longueur est proportionnelle à l'écart-type des points projetés. Par conséquent, le rapport largeur/hauteur d'une surface gauche peut être approximé par la racine carrée de la fraction des valeurs propres et la taille caractéristique est proportionnelle à la racine carrée du produit des valeurs propres.

Ainsi, un facteur de forme approximé  $\tilde{\mu}$  d'une surface gauche peut être défini par :

$$\tilde{\mu} = \sqrt{\frac{\lambda_I}{\lambda_{II}}}. \quad (2.13)$$

De la même façon, une taille caractéristique approximée  $\tilde{\chi}$  d'une surface gauche peut être définie par :

$$\tilde{\chi} = 12 \sqrt{\lambda_I \lambda_{II}}. \quad (2.14)$$

Le mot "approximé" doit être compris dans le sens où ces définitions dépendent de la finesse du maillage. Cependant, il a été empiriquement observé que la finesse du maillage ne change pas de manière significative ces valeurs dès lors qu'il est choisi suffisamment grand. En outre, il a été prouvé (voir l'annexe B) que pour les surfaces rectangulaires planes,  $\tilde{\mu}$  (respectivement  $\tilde{\chi}$ ) coïncide avec  $\mu = \frac{w}{h}$  (respectivement avec  $\chi = wh$ ). Par conséquent, le plan qui approxime la surface est défini par les deux vecteurs propres  $\mathbf{S}_I$  et  $\mathbf{S}_{II}$ , de largeur  $w = \sqrt{12 \lambda_I}$  (selon  $\mathbf{S}_I$ ), et de hauteur  $h = \sqrt{12 \lambda_{II}}$  (selon  $\mathbf{S}_{II}$ ).

### 2.3.4 Pente approximée d'une surface gauche

Contrairement aux surfaces planes, la plus forte pente des surfaces gauches n'est pas constante. Comme proposé dans la section 2.3.1, une première approximation pourrait être la moyenne des pentes des points d'échantillonnage sur toute la surface. Dans ce cas, la pente approximative est donnée par :

$$\tilde{s} = E(s_i), \quad (2.15)$$

où  $s_i$  est la pente au point d'échantillonnage  $\mathbf{S}_i$ .

Une autre approximation, s'appuyant sur l'ACP, consiste à utiliser la pente du plan généré par les vecteurs propres  $\mathbf{S}_I$  et  $\mathbf{S}_{II}$ , puisque c'est le plan qui approxime le mieux la surface :

$$\tilde{s} = \frac{\pi}{2} - \arccos \left( \frac{\mathbf{S}_I \times \mathbf{S}_{II}}{\|\mathbf{S}_I \times \mathbf{S}_{II}\|} \cdot \mathbf{Z} \right) \quad (2.16)$$

ce qui donne l'expression de la pente approximée en fonction des vecteurs propres. Bien que donnant des résultats légèrement différents pour des surfaces gauches, les deux approximations sont exactes pour les surfaces planes.

En d'autres termes, la plus forte pente d'une surface gauche  $\mathcal{S}$  peut être approchée soit par la moyenne des pentes les plus fortes de la surface, soit par la pente la plus forte du plan généré par  $\mathbf{S}_I$  et  $\mathbf{S}_{II}$ . L'influence de ce choix est examinée dans la section 2.5.3.2, mais d'un point de vue purement calculatoire, la deuxième méthode évite le calcul de la pente à chaque point d'échantillonnage (et donc le calcul du vecteur normal  $\mathbf{n}_i$  à chaque point d'échantillonnage  $\mathbf{S}_i$ ).

De même, la direction de plus forte pente peut être calculée soit comme la moyenne de toutes les directions de plus forte pente calculée à chaque point d'échantillonnage  $\mathbf{S}_i$ , soit comme la projection de  $\mathbf{S}_I \times \mathbf{S}_{II}$  sur le plan horizontal ( $X, Y$ ).

En mettant en oeuvre une analyse en composantes principales, il est donc possible d'approximer une surface gauche par un plan.

## 2.4 Application au choix de l'outil

Cette section est consacrée à l'application pratique de la méthode que nous proposons sur la base des résultats déjà démontrés. Il est à noter qu'aucune simulation d'usinage n'est nécessaire pour appliquer cette méthode.

### 2.4.1 Description de la procédure du choix d'outil

Pour mettre en place une telle méthode, peu de données sont nécessaires. Il est évident que les caractéristiques de la machine  $A_{max}$  et  $J_{max}$ , et les données de l'opération d'usinage  $V$ ,  $R$ ,  $r$  et  $sh$  sont indispensables. En outre, un ensemble de points d'échantillonnage de la surface à usiner est également nécessaire. Ces points peuvent être extraits pour une représentation CAO de la surface, mesurés sur une pièce prototype, ou obtenus par toute autre méthode. La méthodologie d'aide au choix d'outil pour l'usinage de surfaces gauches se déroule comme suit :

1. Définir un ensemble de points d'échantillonnage  $\mathbf{S}_i$ , sur toute la surface  $\mathcal{S}$ . Seules les coordonnées  $(S_x, S_y, S_z)$  de ces points sont nécessaires pour les calculs qui suivent.
2. Calculer la matrice de covariance  $C$  en utilisant l'équation (2.12)

3. Effectuer une décomposition en éléments propres de  $C$  pour trouver  $\lambda_I, \lambda_{II}, \mathbf{S_I}, \mathbf{S_{II}}$
4. Calculer la plus forte pente  $\tilde{s}$  en utilisant l'équation (2.16) et la direction de la plus forte pente
5. Calculer le facteur de forme  $\tilde{\mu}$  et la taille caractéristique  $\tilde{\chi}$  à l'aide des équations (2.13) et (2.14)
6. Calculer le paramètre critique  $s_c(\tilde{\mu})$  en utilisant l'équation (2.10)
7. Comparer  $\tilde{s}$  avec le paramètre critique : si  $s_c(\tilde{\mu}) < \tilde{s}$ , alors choisir l'outil sphérique le long de la direction principale. Sinon, choisir l'outil torique selon la direction de la plus forte pente.

Remarquez que la plupart des étapes de cette procédure sont des applications directes de formules analytiques. Cela signifie que ces étapes sont très rapides à réaliser. La pièce maîtresse du calcul est la décomposition en éléments propres de la matrice de covariance  $C$ . De nombreux logiciels sont très efficaces pour effectuer cette tâche, en particulier sur les petites matrices ( $3 \times 3$ ).

## 2.4.2 Application à des cas tests

Afin de tester l'approche susmentionnée, des tests avec deux surfaces gauches sont effectués. Les deux surfaces sont choisies de manière à ce que la direction principale soit presque orthogonale à la direction de plus grande pente moyenne, et discrétisées à l'aide d'un maillage isoparamétrique régulier de  $80 \times 80$ . Les temps d'usinage, correspondant à l'outil torique dans la direction de plus forte pente, et à un outil sphérique dans la direction principale, sont comparés au résultat prévu par les paramètres critiques  $\mu_c(\tilde{s})$  et  $s_c(\tilde{\mu})$ .

### 2.4.2.1 Cas test 1

Le premier cas test est une surface de Bézier de taille  $3 \times 3$  dont les points de contrôle sont donnés dans le tableau 2.2. Cette surface a été utilisée dans [73] et est illustrée dans la figure 2.6 où les axes **X**, **Y** et **Z** respectivement, sont représentés par les vecteurs rouge, vert et bleu respectivement.

point (0,0)	point (0,1)	point (0,2)
0.0	0.0	0.0
-15.32	-6.42	15.32
-12.85	7.66	12.85
point (1,0)	point (1,1)	point (1,2)
40.0	40.0	40.0
-18.53	-9.64	12.10
-9.02	11.49	16.68
point (2,0)	point (2,1)	point (2,2)
80.0	80.0	80.0
-28.17	-22.49	2.46
2.46	26.81	28.17

TABLE 2.2 – Surface test 1 : coordonnées cartésiennes des points de contrôle

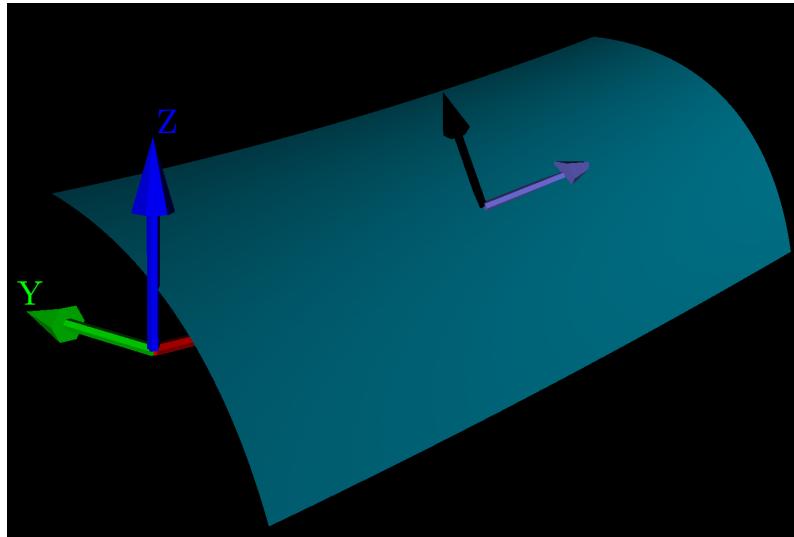


Figure 2.6 – Surface test 1 avec sa direction principale (violet) et sa direction de plus grande pente (noir)

	temps d'usinage	longueur de parcours
outil torique selon la direction de plus forte pente	91.3 s	5606 mm
outil sphérique selon la direction principale	79.7 s	5687 mm

TABLE 2.3 – Temps d'usinage et longueur de parcours pour la surface test 1

Sur cette figure, le vecteur noir indique la direction de plus grande pente tandis que le vecteur violet indique la direction principale. La pente approximée  $\tilde{s}$ , calculée en utilisant l'équation (2.16), est égale à  $0.739 \text{ rad} = 42.34^\circ$ . La direction de grande pente (projétée sur le plan ( $\mathbf{X}, \mathbf{Y}$ )) forme un angle avec l'axe  $\mathbf{X}$  de  $1.169 \text{ rad} = 66.97^\circ$ . Appliquer l'ACP sur les coordonnées de la surface permet de calculer les valeurs propres :  $\lambda_I = 572.62$  et  $\lambda_{II} = 133.31$ , ce qui conduit à un facteur de forme :  $\tilde{\mu} = 2.07$ , et à une direction principale, qui définit un angle de  $-0.17 \text{ rad} = -9.74^\circ$  avec l'axe  $\mathbf{X}$ . Les résultats de l'usinage selon les deux directions sont rapportés sur le tableau 2.3. La simulation numérique de l'usinage dure environ 3 s. Les trajectoires d'usinage dans les deux cas sont affichées sur la figure 2.7.

A partir des résultats obtenus, on remarque l'importance de prendre en compte la cinématique de la machine CNC. En effet, même si la longueur du parcours est légèrement plus courte pour l'outil torique, l'outil à bout sphérique présente un temps d'usinage plus court.

Sur cette surface, la procédure proposée se déroule comme suit : la taille caractéristique  $\tilde{\chi}$  du plan généré par les vecteurs propres est déduite des valeurs propres :

$$\tilde{\chi} = 12 \sqrt{\lambda_I \lambda_{II}} = 3315.47 \text{ mm}^2$$

Sachant que  $\tilde{\mu} = 2.07$ , la pente critique est calculée en utilisant l'équation (2.10) :  $s_c(\tilde{\mu}) = 38.9^\circ$ . Puisque

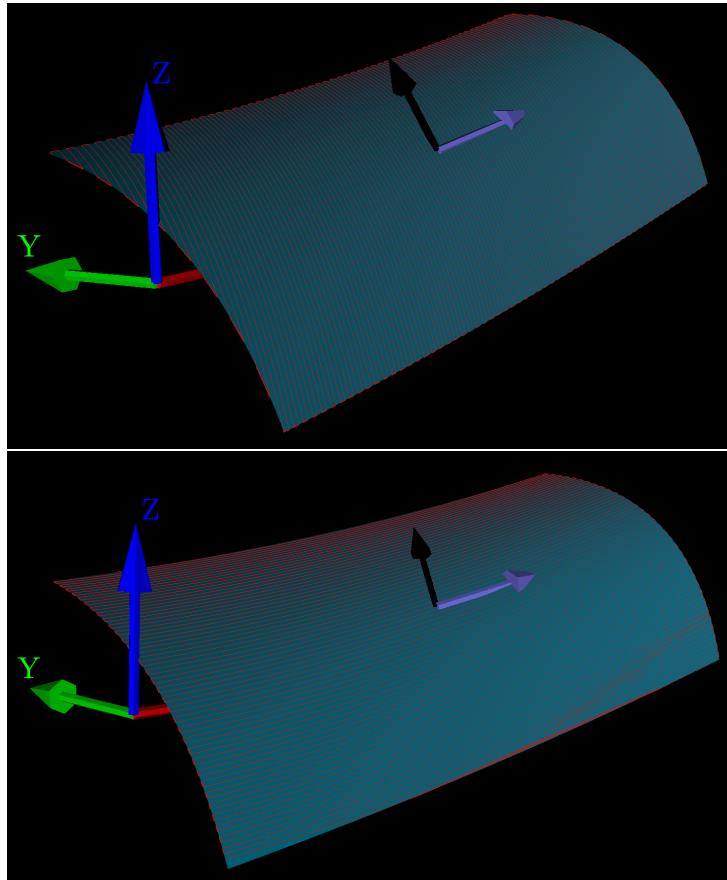


Figure 2.7 – Usinage de la surface test 1 : selon la direction de plus grande pente avec un outil torique (haut), selon la direction principale avec un outil sphérique (bas)

$s_c < \tilde{s}$ , notre approche prédit que l'outil sphérique donnera un temps d'usinage plus court, ce qui est en accord avec les résultats numériques. Par conséquent, la prédiction est correcte sur ce premier cas test.

#### 2.4.2.2 Cas test 2

Le second cas test est une surface de Bézier de  $4 \times 4$  points de contrôle qui sont donnés dans le tableau 2.4. Cette surface était auparavant utilisée par [21] pour des tests d'usinage de surfaces gauches. Elle est illustrée dans la figure 2.8, avec le même code couleur que pour le cas de test précédent pour les axes **X**, **Y** et **Z**.

Là encore, les directions principale et de plus forte pente sont presque orthogonales. La pente approximative  $\tilde{s}$  est égale à  $0,299 \text{ rad} = 17,13^\circ$ , et sa direction est alignée avec l'axe **X**. Les valeurs propres sont  $\lambda_I = 312,07$  et  $\lambda_{II} = 219,02$ , qui sont utilisées pour calculer le facteur de forme :  $\tilde{\mu} = 1,19$ , et la direction principale, définissant un angle de  $1,57 \text{ rad} = 90,0^\circ$  avec l'axe **X**. Les résultats de l'usinage dans les deux directions sont résumés dans le tableau 2.5. La simulation numérique a duré environ 3 s. La figure 2.9 montre les trajectoires d'usinage dans les deux cas.

Ici, la longueur de parcours est légèrement plus courte pour l'outil torique, et cet outil est également celui qui présente le temps d'usinage le plus court.

point (0,0)	point (0,1)	point (0,2)	point (0,3)
7.92	6.33	6.33	7.92
0.0	20.4	40.8	61.2
37.26	29.81	29.81	37.26
point (1,0)	point (1,1)	point (1,2)	point (1,3)
23.72	22.14	22.14	23.72
0.0	20.4	40.8	61.2
26.11	18.66	18.66	26.11
point (2,0)	point (2,1)	point (2,2)	point (2,3)
42.70	41.12	41.12	42.70
0.0	20.4	40.8	61.2
29.87	22.42	22.42	29.87
point (3,0)	point (3,1)	point (3,2)	point (3,3)
56.02	54.44	54.44	56.02
0.0	20.4	40.8	61.2
19.25	11.79	11.79	19.25

TABLE 2.4 – Surface test 2 : coordonnées cartésiennes des points de contrôle

	temps d'usinage	longueur de parcours
outil torique selon la direction de plus grande pente	71.8 s	4725 mm
outil sphérique selon la direction principal	76.8 s	5184 mm

TABLE 2.5 – Temps d'usinage et longueur de parcours d'outil pour la surface test 2

Sur cette surface, l'application de la procédure proposée se déroule comme suit : La taille caractéristique du plan généré par les vecteurs propres est égale à  $\tilde{\chi} = 3137 \text{ mm}^2$ ; notez que cette valeur est proche de l'aire de la surface. Sachant que  $\tilde{\mu} = 1,19$ , la pente critique peut être calculée à l'aide de l'équation (2.10) et est égale à  $s_c(\tilde{\mu}) = 62,6^\circ$ . Puisque  $\tilde{s} < s_c$ , notre approche prédit que l'outil torique présentera un temps d'usinage plus court, ce qui est en accord avec les résultats numériques. Par conséquent, la prédition est également correcte sur ce deuxième cas de test. Afin d'étudier la fiabilité de la méthode proposée sur ces cas concrets, des tests sont effectués pour déterminer numériquement la pente critique  $s_c^*$ . Pour ce faire, les deux temps d'usinage  $T_1^*$  et  $T_2^*$  sont estimés par simulation numérique pour différentes valeurs de la pente de surface  $\tilde{s}$ . La réalisation de ces expériences révèle des temps d'usinage égaux pour  $s_c^* = 54,7^\circ$ . Par rapport à la valeur  $s_c = 62,6^\circ$  fournie par la méthode proposée, la prédition présente une erreur d'environ 10%. Cette valeur doit être considérée avec prudence, car elle dépend fortement de la géométrie de la surface à usiner.

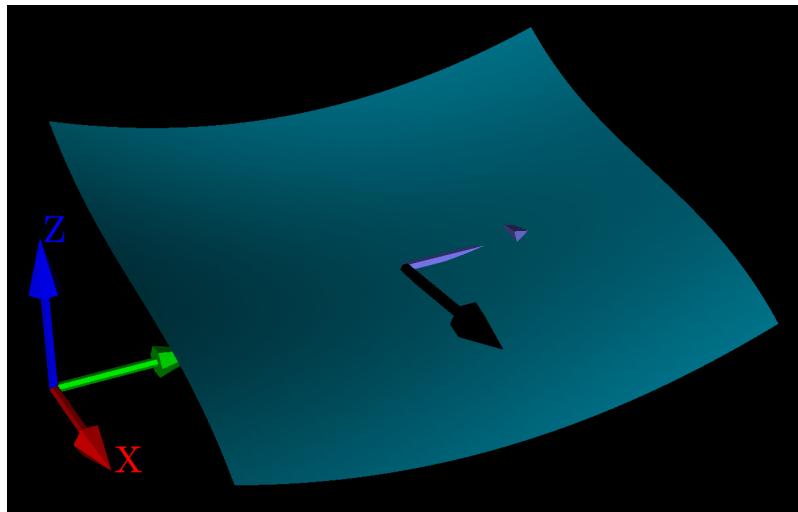


Figure 2.8 – Surface test 1 avec sa direction principale (violet) et sa direction de plus grande pente (noir)

## 2.5 Discussion

La validité et les limitations des développements proposés dans ce chapitre sont discutés dans cette section.

### 2.5.1 Sur l'étude analytique des surfaces planes

Pour mener à bien cette étude sur des surfaces planes rectangulaires, quelques hypothèses ont été formulées :

- Premièrement, le rapport  $\frac{\chi}{\mu}$  est considéré comme suffisamment important pour garantir que le temps d'usinage des trajectoires est calculé de telle sorte que la vitesse de consigne soit atteinte avant l'accélération maximale. Pour les valeurs des paramètres cinématiques utilisées dans ce document, la limite inférieure est  $\frac{\chi}{\mu} > 7.6$  mm. Cette condition exclut les très petites surfaces, ce qui est acceptable dans la pratique.
- Deuxièmement, il est supposé que les pas transversaux (entre deux trajectoires adjacentes) ne sont pas suffisamment longs pour atteindre la vitesse de consigne. Ceci est vérifié pour les deux outils dès que la largeur de la bande d'usinage est inférieure à 7.6 mm. Cette condition dépend des valeurs des paramètres cinématiques mais aussi des dimensions de l'outil, et doit être vérifiée lorsque ces valeurs changent. En pratique, les dimensions des outils utilisés pour la finition sont suffisamment petites pour considérer cette condition comme vérifiée.
- La dernière hypothèse concerne le calcul de la pente critique  $s_c$ . Il a été supposé que  $\chi > 100$  mm<sup>2</sup> afin de s'assurer que l'équation (2.6) n'a qu'une seule solution réelle. En pratique, les cas où la surface est inférieure à 100 mm<sup>2</sup> sont très rares.

En outre, l'approche proposée repose fortement sur les paramètres cinématiques de la machine-outil à CN :  $J_{max}$ ,  $A_{max}$  et la vitesse de consigne  $V$ . Par conséquent, les hypothèses concernant le calcul du temps

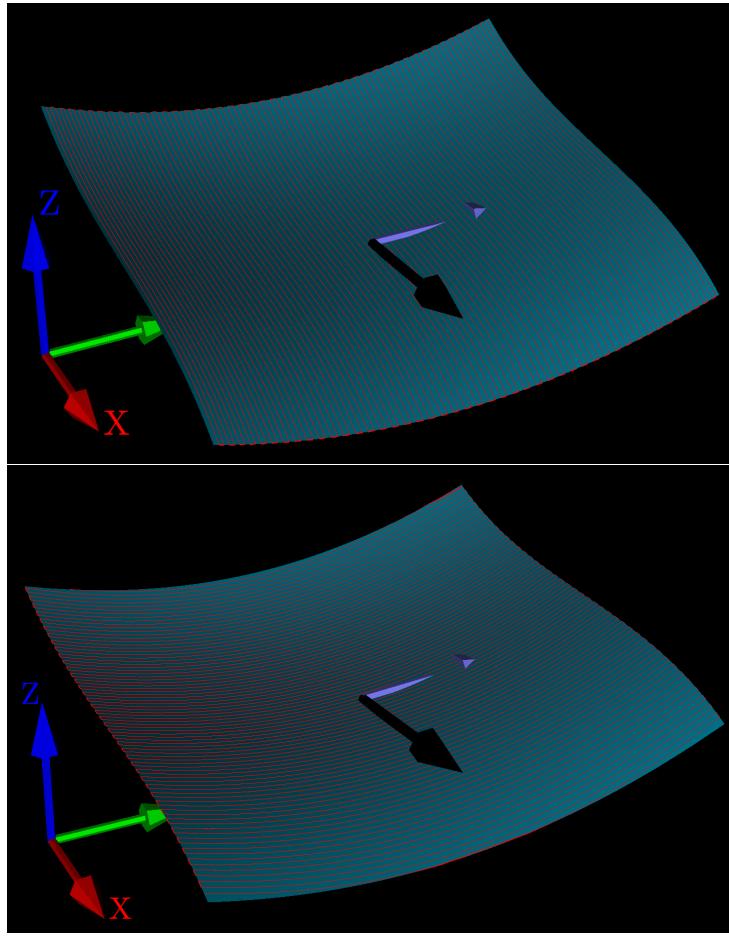


Figure 2.9 – Usinage de la surface test 2 : selon la direction de plus forte pente avec un outil torique (en haut), selon la direction principale avec un outil sphérique (en bas)

d’usinage (section 1.6) doivent être vérifiées pour différentes valeurs de paramètres cinématiques, mais la procédure reste la même.

### 2.5.2 Sur l’approximation des surfaces gauches par l’ACP

Dans la section 2.3, une approximation des surfaces gauches basée sur l’ACP est développée. Le but premier de cette approximation est de permettre d’appréhender une surface gauche à travers un ensemble de paramètres numériques faciles à manipuler par les algorithmes. Grâce à la décomposition en vecteurs propres de la matrice de covariance des coordonnées d’un ensemble de points pris sur la surface, le plan rectangulaire approximant au mieux la surface peut être défini à partir de ces vecteurs propres et des valeurs propres associées. De plus, le troisième vecteur propre,  $\lambda_{III}$ , donne une bonne idée de la normale moyenne de cette surface. Une fois rassemblées, toutes ces informations constituent un ensemble de données numériques permettant d’avoir une vision globale et quantifiable de la surface.

Néanmoins, cette méthodologie d’approximation demeure une approximation et il est dès lors indispensable de s’interroger sur la précision de cette approximation. Malheureusement, étant donnée la variété infinie des surfaces gauches qui peuvent être définies, il n’est pas possible d’estimer cette pré-

cision autrement que qualitativement. De manière évidente, plus la surface sera plane, meilleure sera l'approximation. Dans le même ordre d'idées, plus la projection de la surface dans un plan normal à  $\lambda_{III}$  sera proche d'un rectangle, meilleure sera l'approximation. Cependant, au-delà de ces considérations manifestes, il est difficile de conjecturer sur la précision de cette approximation. Seule l'expérimentation et la simulation numérique permettent *in fine* de lever le doute à son sujet. Néanmoins, tout au long de ces travaux, elle s'est avérée suffisante pour satisfaire aux objectifs poursuivis.

Par ailleurs, si, pour un cas de figure donnée, elle s'avère insuffisante, il est très aisément d'améliorer en modifiant la manière dont la surface est approximée à partir des valeurs propres et vecteurs propres calculés par l'ACP. Par exemple, il est possible de considérer une surface plane elliptique en lieu et place d'une surface plane rectangulaire. En effet, il est tout à fait envisageable de considérer que les vecteurs  $\sqrt{12\lambda_I}\mathbf{S}_I$  et  $\sqrt{12\lambda_{II}}\mathbf{S}_{II}$  définissant le rectangle, définissent plutôt, respectivement, le grand et le petit axe de cette ellipse. De plus, pour les surfaces présentant un profil à forte concavité, il est possible d'utiliser la troisième composante de la décomposition en valeurs propres pour tenir compte de cette concavité. En effet, à partir de trois vecteurs orthogonaux dans l'espace, il est aisément de définir un ellipsoïde, ou plutôt, en l'occurrence, un demi-ellipsoïde qui dans le cas de surfaces à forte concavité, permettra sans aucun doute de mieux tenir compte de cette concavité dans l'approximation de la surface.

En résumé, même si, dans le cas général, il est difficile d'évaluer l'erreur commise par l'approximation réalisée, il est tout à fait possible d'améliorer cette approximation pour certains cas particuliers délicats. Ceci étant, un tel cas ne s'est jamais présenté au cours de ces travaux et l'approximation des surfaces gauches par des plans rectangulaires s'est toujours avérée suffisante étant donné les objectifs à atteindre.

### 2.5.3 Sur la méthodologie d'aide au choix d'outil

#### 2.5.3.1 De l'efficacité de la méthode proposée

La procédure de choix d'outil proposée est pleinement efficace lorsque les directions d'usinage (c'est-à-dire la direction de la plus forte pente, et la direction selon laquelle la surface est le plus étendue, appelée direction principale) sont presque perpendiculaires, sinon le calcul des paramètres critiques perd sa précision. Ceci peut être illustré par les tests menés sur la surface du premier cas de test (section 2.4.2.1) afin de déterminer numériquement la pente critique  $s_c^*$ , tout comme le deuxième cas de test (section 2.4.2.2). La réalisation de ces tests nécessite une rotation de la surface pour définir différentes valeurs de la pente  $\tilde{s}$ , mais la rotation de la surface entraîne un changement significatif de la direction de plus forte pente, qui n'est plus orthogonale à la direction principale. Par conséquent, la préiction est correcte pour l'ensemble des pentes  $\tilde{s}$  de sorte que les deux directions sont presque orthogonales, mais elle devient rapidement erronée lorsque les deux directions sont trop éloignées de la perpendiculalité. Ainsi, plus l'angle entre les deux directions est proche de 90 degrés, plus le calcul des paramètres critiques est précis. Cela peut sembler être un facteur limitant, mais en fait, plus la direction de plus forte

pente et la direction principale sont proches, plus l'outil torique doit être privilégié, car il permet d'obtenir des pas transversaux plus importants sans augmenter le nombre de trajectoires. Par conséquent, la procédure proposée est vraiment utile et précise pour les surfaces problématiques, et moins intéressante pour les surfaces pour lesquelles la direction d'usinage et le type d'outil sont évidents à choisir.

### 2.5.3.2 Sur la mise en œuvre de la méthode

Il est évident que la méthodologie proposée est plus précise pour les surfaces dont le vecteur normal varie peu, et donc pour les directions de plus forte pente qui varient peu. *A contrario*, l'approche est moins précise lorsque la direction de la pente varie beaucoup, en particulier lorsque la stratégie des plans parallèles est utilisée. En effet, la stratégie des plans parallèles implique que le pas transversal est déterminé par le point d'interpolation le plus défavorable, contrairement à la stratégie isocrète, et la gamme des pas transversaux calculés est plus large lorsque le vecteur normal à la surface varie beaucoup.

Pour les surfaces gauches, il existe deux façons, comme mentionné précédemment (section 2.3.4), de calculer la pente approximative  $\tilde{s}$  qui peuvent donner des valeurs différentes. Ici encore, la différence est plus importante lorsque le vecteur normal varie beaucoup. Cependant, au cours de tous les tests effectués pendant les travaux sur cette question, le choix d'une méthode plutôt qu'une autre ne conduit jamais à une différence suffisamment importante pour faire une prévision erronée.

Remarquez que si l'utilisateur ne peut utiliser que l'outil sphérique, la direction principale est une direction d'usinage de choix, puisqu'elle réduit le nombre de trajectoires du parcours, par rapport aux autres directions à une distance à pas transversal constant. En outre, l'identification de la direction principale peut être intéressante dans l'usinage 5 axes et 3+2 axes avec un outil torique, ce qui fait l'objet d'études dans les chapitres suivants.

Un autre point qui mérite d'être noté est que la tolérance de hauteur de crête  $sh$  ne semble pas avoir une influence notable sur les paramètres critiques. En effet, la même courbe d'évolution du paramètre critique  $s_c(\mu)$  (voir figure 2.4) se retrouve pour  $sh = 0.1$  mm (10 fois plus grand).

L'impact du maillage est également étudié car les deux directions d'usinage définies précédemment (plus forte pente et principale) sont calculées à partir des points échantillonnes  $S_i$  sur le maillage, et dépendent donc du maillage. De nombreux tests pour différents maillages (de  $20 \times 20$  à  $200 \times 200$ ) ont été effectués, et il a été constaté que la modification des temps d'usinage n'est pas significative, en particulier pour les maillages supérieurs à  $60 \times 60$ . Les paramètres  $\tilde{s}$  et  $\tilde{\mu}$  dépendent également de la taille du maillage, et des observations similaires ont été trouvées. Par conséquent, la taille du maillage peut être choisie suffisamment grande pour que les résultats n'y soient pas sensibles.

### 2.5.3.3 De l'influence des paramètres auxiliaires

Tout au long de cette étude, la taille caractéristique  $\chi$  est considérée comme constante. En effet, ce paramètre, représentant l'aire de la surface, a beaucoup moins d'influence que l'autre paramètre,  $\mu$ , représentant l'allongement de la surface. Néanmoins, son influence est examinée ici, malgré son faible

rôle. L'usinage dans la direction principale avec un outil sphérique est plus efficace que son homologue (outil torique dans la direction de plus grande pente), lorsque les trajectoires générées dans le premier cas sont telles que la vitesse de consigne est atteinte sur une proportion nettement plus importante des trajectoires, par rapport au second cas. Ainsi, pour une vitesse de consigne donnée  $V$ , plus la surface est grande, plus grande est la proportion de trajectoires pour lesquelles la vitesse de consigne est atteinte pour les deux cas d'usinage, favorisant l'outil torique dans la direction de plus forte pente. En conséquence, le paramètre critique  $s_c$  devrait augmenter au fur et à mesure que  $\chi$  augmente. Cette affirmation est prouvée dans ce qui suit.

On montre d'abord que  $\mu_c(s)$  est une fonction croissante de  $\chi$ , puis ensuite on utilise le fait que  $\mu_c(s)$  et  $s_c(\mu)$  sont des fonctions réciproques.

Pour montrer que  $\mu_c(s)$  est croissante en fonction de  $\chi$ , la dérivée de  $\mu_c$  par rapport à  $\chi$  (notée  $\frac{\partial \mu_c}{\partial \chi}$ ) doit d'abord être calculée, tout autre paramètre étant fixe. L'équation (2.17), donne l'expression de la dérivée  $\frac{\partial \mu_c}{\partial \chi}$  calculée à partir de l'expression analytique de l'équation (8) :

$$\frac{\partial \mu_c}{\partial \chi} = \frac{\mu_c}{\beta_1} \frac{\beta_2}{2\sqrt{\chi}} + \frac{\beta_2^2}{2\sqrt{\chi\beta_2^2 + 4\beta_1\beta_3}}, \quad (2.17)$$

$$\begin{aligned} \text{avec } \beta_1 &= 2 \frac{sod_2}{sod_1} \left( \sqrt{\frac{V}{J_{max}}} + 2 \sqrt[3]{\frac{sod_1}{2J_{max}}} \right) \geq 0 \\ \beta_2 &= \frac{\left(1 - \frac{sod_2}{sod_1}\right)}{V} \geq 0 \\ \beta_3 &= 2 \left( \sqrt{\frac{V}{J_{max}}} + 2 \sqrt[3]{\frac{sod_2}{2J_{max}}} \right) \geq 0. \end{aligned}$$

Il est évident à partir de cette expression que la dérivée est positive. Ainsi  $\mu_c$  est croissante en fonction de  $\chi$ .

Soit  $s \in [0, \pi/2]$  et  $\mu \geq 1$ . D'après la définition des paramètres critiques, pour n'importe quelle valeur de  $\chi$  :  $\mu_c(s)$  est la valeur que le facteur de forme doit prendre si la pente est égale à  $s$ , pour que les temps d'usinage dans les deux cas soient égaux. Par conséquent, si  $\mu = \mu_c(s)$ , alors  $s = s_c(\mu)$ . Autrement dit,  $\mu_c$  et  $s_c$  sont deux fonctions réciproques l'une de l'autre :

$$\forall s \in \left[0, \frac{\pi}{2}\right] \quad s_c(\mu_c(s)) = s \text{ et } \forall \mu \geq 1 \quad \mu_c(s_c(\mu)) = \mu.$$

Ceci peut être écrit en incluant la dépendance à  $\chi$  :

$$\forall s \in \left[0, \frac{\pi}{2}\right], \forall \chi \geq 0, \quad s_c(\mu_c(s, \chi), \chi) = s.$$

Dériver l'équation précédente par rapport à  $\chi$  pour  $s$  constant, en utilisant la formule de dérivée totale donne :

$$\frac{\partial s_c}{\partial \chi}(\mu_c(s, \chi), \chi) = -\frac{\partial \mu_c}{\partial \chi}(s, \chi) \frac{\partial s_c}{\partial \mu}(\mu_c(s, \chi), \chi)$$

Or, on vient de démontrer que  $\frac{\partial \mu_c}{\partial \chi}$  est positive. De plus, on a vu que  $s_c$  est une fonction décroissante de  $\mu$ . Ainsi,  $\frac{\partial s_c}{\partial \mu}$  est négative. Puisque  $\mu_c$  est une bijection, alors pour tout  $\mu \geq 1$  il existe (un unique)  $s \in [0, \pi/2]$  tel que  $\mu = \mu_c(s, \chi)$ . Par conséquent, le terme  $\frac{\partial s_c}{\partial \chi}(\mu, \chi)$  est positif pour n'importe quelle valeur de  $\mu$ , et  $s_c$  est une fonction croissante de  $\chi$ .

De même, la vitesse de consigne  $V$  a été considérée comme fixe. Pourtant, on peut montrer qu'une vitesse de consigne élevée favorise l'usinage dans la direction principale, tandis qu'une vitesse de consigne faible favorise la direction de la pente. Plus précisément, la vitesse de consigne est obtenue sur des portions plus importantes des trajectoires car sa valeur est plus faible, et l'intérêt de la direction principale est perdu dans ce cas. Inversement, une vitesse de consigne élevée pénaliserait les parcours avec plus de changements de direction (ou de zig-zag), favorisant ainsi l'usinage dans la direction principale. Cette affirmation est confirmée par des simulations numériques montrant que les paramètres critiques  $\mu_c$  et  $s_c$  sont des fonctions décroissantes de  $V$ .

La méthode proposée effectue une ACP sur les coordonnées des points de la surface à usiner afin de trouver la direction principale et d'estimer les paramètres  $\tilde{\mu}$  et  $\tilde{\chi}$  en utilisant les valeurs propres  $\lambda_I$ ,  $\lambda_{II}$  et leurs vecteurs propres. Bien que le troisième vecteur propre  $\lambda_{III}$  ne soit pas utilisé dans notre méthode, il peut donner une idée globale de la planéité d'une surface donnée et donc de l'applicabilité de la méthode proposée à cette surface. Des critères quantitatifs ne sont pas disponibles à ce stade, mais certains paramètres locaux, tels que la courbure gaussienne, doivent être pris en compte pour les construire.

## 2.6 Conclusion

Dans ce chapitre, une méthodologie d'approximation de surfaces gauches dans l'espace 3D a été présentée. En se basant sur une analyse en composantes principales, cette méthodologie permet de caractériser une surface gauche par un ensemble de paramètres numériques. Elle fournit donc une vision globale et quantifiable d'une surface gauche qui pourra par la suite être utilisée par divers algorithmes numériques.

Une première application de cette approximation a également été présentée dans ce chapitre. Il s'agit d'une procédure d'aide au choix d'outil pour le fraisage en bout des surfaces gauches en 3-axes. Cette application présente l'avantage de ne nécessiter aucune simulation numérique de l'usinage pour pouvoir prédire lequel de l'outil torique usinant dans la direction de plus grande pente ou de l'outil sphérique usinant dans la direction principale permettra d'usiner la surface dans le temps le plus court (tous les autres paramètres étant égaux par ailleurs, bien sûr). Par ailleurs, cette aide au choix d'outil s'avère la

plus efficace pour les surfaces les plus problématiques, c'est à dire pour les surfaces dont la direction de plus grande pente et la direction principale sont perpendiculaires.

Ce travail n'est qu'une première approche de l'approximation des surfaces gauches par des caractéristiques numériques quantifiables. Pour l'instant seule l'approximation par un plan a été développée, mais il est tout à fait possible d'envisager d'autres entités géométriques. Par exemple, en utilisant la troisième direction propre, on pourra approximer une surface gauche avec des entités géométriques plus complexes que des plans, telles que des paraboloïdes elliptiques.



# Chapitre 3

## Partitionnement de surfaces gauches pour l’usinage

Ce chapitre traite du partitionnement de surfaces à l’aide d’algorithmes de *clustering* (classification non-supervisée). Une nouvelle métrique est introduite permettant de favoriser les partitions avec des zones les plus allongées possible, ce qui améliore le comportement cinématique de la machine, tout en gardant un compromis avec la direction de plus grande pente qui maximise les rayons effectifs, et par la suite la longueur d’usinage.

### 3.1 Motivation et intérêt

Soit  $S$  une surface de représentation paramétrique  $\mathbf{S}(u, v)$ , une partition  $(z_j)_{j=1, \dots, n}$  de la surface  $S$  est définie comme une famille finie de sous-ensembles de  $S$  deux-à-deux disjoints et dont la réunion est égale à  $S$  :

$$z_i \cap z_j = \emptyset, \forall (i, j) \in [1; n]^2, i \neq j \text{ et } \cup_{j=1}^n z_j = S$$

Le partitionnement de surfaces en zones est une approche très intéressante qui est de plus en plus étudiée et utilisée pour le fraisage des surfaces gauches. En effet, elle permet d’usiner chaque zone avec des paramètres d’usinage (qui incluent, entre autres, la direction d’usinage) *ad-hoc*. Ce qui peut réduire le temps d’usinage de chaque zone sans affecter les autres.

En outre, le partitionnement est directement lié au choix du type d’outil. En effet, il est bien établi qu’un outil torique peut donner de meilleurs résultats que son homologue sphérique [12]. En fait, un outil torique donne de meilleurs résultats qu’un outil sphérique lorsqu’il est utilisé dans la direction de plus forte pente ; par contre s’il est utilisé perpendiculairement à cette direction, ses performances sont moins bonnes que celles de l’outil sphérique. Comme la direction de plus grande pente peut varier considérablement sur une surface gauche, le partitionnement de cette surface en plusieurs zones (chacune d’entre elles sera ensuite usinée dans une direction appropriée) est considérée comme la meilleure approche pour améliorer l’efficacité de l’outil torique.

## 3.2 État de l'art

La définition des zones pour l'usinage en bout des surfaces gauches est un problème compliqué qui a été abordé par plusieurs auteurs. La plupart des études se concentrent sur l'usinage 3 axes. L'une d'entre elles est présentée dans [42]. En utilisant un outil sphérique, cette méthode est basée sur le champ de directions d'usinage préférées, qui maximisent la largeur de la bande d'usinage (ou le pas transversal) en respectant la contrainte de hauteur de crête. Ce champ a les propriétés d'un tenseur symétrique d'ordre 2, ce qui permet d'identifier les points critiques de ce tenseur correspondant à des points dégénérés, où toutes les directions sont équivalentes. Ensuite, selon le type de chaque point critique, des limites entre zones sont identifiées autour de ce point, ce qui définit une partition de la surface. Enfin, chaque zone est usinée avec la stratégie isocrête. Une méthode très similaire est proposée dans [47] : une approche analytique est présentée, où la direction d'usinage qui maximise la largeur de la bande est identifiée.

Des méthodes d'optimisation heuristiques inspirées du problème de tournées de véhicules sont proposées dans [26]. L'objectif est de diviser la surface en zones avec de petites variations de la direction de plus forte pente afin d'augmenter le rayon effectif de l'outil torique. Cela devrait permettre de maximiser les pas transversaux et de minimiser la longueur du parcours de l'outil. Les meilleurs résultats sont obtenus par l'algorithme de Clarke et Wright (décrit dans [4]), la version parallélisée de cet algorithme étant plus rapide.

Le même objectif, la recherche d'une partition optimale, est poursuivi dans l'approche présentée dans [25]. Cette approche est basée sur les systèmes multi-agents adaptatifs (AMAS) [14]. L'idée est de partitionner la surface en zones selon un critère basé sur la direction optimale d'usinage qui est, pour chaque point, la direction de plus forte pente. La surface est maillée, et la direction optimale est calculée sur chaque maille unitaire. Le problème est ensuite résolu (en utilisant l'approche AMAS) en considérant un agent sur chaque maille. Les zones sont caractérisées par deux critères contradictoires : le nombre d'éléments (agents) dans la zone, et la différence maximale entre deux directions de la même zone. Le processus d'optimisation nécessite cependant de nombreuses itérations pour converger vers un état stable dans lequel aucun agent ne prend une nouvelle décision.

Dans [73], les auteurs proposent une approche de partitionnement pour l'usinage 3-axes avec un outil torique garantissant l'efficacité de cet outil par rapport à l'outil sphérique. La première zone contient tous les points qui pourraient être usinés dans la direction de plus forte pente du point où l'angle de pente est maximal, tout en satisfaisant au moins une amélioration minimale du rayon effectif par rapport à l'outil sphérique de même rayon. La deuxième zone est construite de la même manière en tenant compte des points restants. Ce processus est répété jusqu'à ce qu'il ne reste plus aucun point.

D'autres études portent sur l'usinage 3+2-axes, qui est le même que l'usinage 3-axes, en ajoutant deux axes de rotation utilisés pour positionner la pièce entre deux phases d'usinage. Parmi elles, la méthode proposée dans [18] discrétise la surface avec un maillage isoparamétrique régulier, et calcule la courbure gaussienne et la courbure moyenne pour chaque nœud du maillage. Ensuite, un type de

courbure : convexe, concave ou en forme de selle, est attribué à chaque point. Les points sont alors divisés en points sans interférence et en points non-accessibles. Cette partition est finalement affinée à l'aide de l'algorithme *fuzzy-Cmeans*, et le nombre de zones est décidé par la méthode de "subtractive clustering" [54].

L'un des travaux les plus remarquables dans les méthodes de partitionnement pour l'usinage 3+2-axes est la méthode de partitionnement de surfaces gauches proposée dans [63]. Les auteurs étudient d'abord les paramètres à utiliser pour le vecteur de caractéristiques "*feature vector*" afin d'obtenir une meilleure partition de la surface en utilisant l'algorithme *fuzzy-Cmeans*. En appelant  $\mathbf{S}(u, v)$  la fonction paramétrique définissant une surface gauche à usiner, et  $\mathbf{n}(u, v)$  le vecteur normal à cette surface au point  $(u, v)$ ; cette étude, réalisée sur deux surfaces d'essai, montre que la combinaison de paramètres  $(u, v, \mathbf{n}(u, v))$  est efficace. L'orientation de l'outil, désignée par le vecteur  $\mathbf{T}$ , est calculée de telle sorte que : l'axe de l'outil est coplanaire avec la direction d'avance,  $\mathbf{F}$ , et l'axe,  $\mathbf{Z_m}$ , de la machine. L'angle d'inclinaison est défini par l'utilisateur. Pour les points situés dans des régions où la surface est convexe, l'angle d'inclinaison peut être nul, et il peut être le plus faible possible aux points concaves, de façon à éviter les interférences locales. La stratégie d'usinage en plan parallèle est adoptée pour un cas test qui montre que généralement la longueur du parcours de l'outil diminue à mesure que le nombre de zones augmente. En fait, le temps d'usinage diminue jusqu'à ce qu'un nombre optimal de zones soit atteint ; il augmente ensuite en raison du temps nécessaire à l'outil pour se retirer et se déplacer de zone en zone.

Dans [46], l'algorithme *K-means* est utilisé pour partitionner une surface gauche pour l'usinage en 3+2-axes. Dans cette étude, le vecteur caractéristique est composé des coordonnées 3D de  $\mathbf{S}(u, v)$  et  $\mathbf{n}(u, v)$ , ce qui donne un vecteur caractéristique à six paramètres  $(S_x, S_y, S_z, n_x, n_y, n_z)$ . Le choix des coordonnées 3D peut cependant poser des problèmes (voir la section 3.3.2).

Le partitionnement de surfaces pour l'usinage 5-axes a également été abordées dans la littérature. Une méthode basée sur l'algorithme de coupe normalisée "*Normalized-Cut*" est proposée dans [50]. Cet algorithme est utilisé dans la segmentation des images et la classification des champs vectoriels [17]. Tout d'abord, un champ de directions optimales est projeté sur une grille régulière, qui définit les points d'interpolation du parcours de l'outil dans l'espace paramétrique. Ensuite, la surface est partitionnée en utilisant l'algorithme de coupe normalisée, pour lequel la mesure de dissimilarité est une combinaison convexe de paramètres de proximité (les coordonnées paramétriques  $u$  et  $v$ ), et la direction optimale d'usinage. Chaque zone est usinée en utilisant soit une stratégie d'iso-scallop "en zigzag", soit une stratégie d'iso-contour, selon la nature des points critiques de la zone.

En résumé, l'objectif de ces approches est de trouver un partitionnement approprié de la surface pour améliorer l'usinage. Certaines approches considèrent le partitionnement comme un problème d'optimisation où la fonction objectif est un critère d'usinage (durée d'usinage ou longueur de trajectoire). Bien qu'elles soient coûteuses en termes de calcul, ce type d'approches présente l'avantage de lier directement le partitionnement à l'usinage. D'autres approches considèrent le partitionnement comme un problème de classification non-supervisée dans le but de trouver une partition de la surface avec des

zones homogènes.

Le point commun entre la plupart de ces approches est qu’elles sont basées sur des critères et des propriétés locales : géométrie locale de la surface, rayon effectif et pas transversal au point de contact, etc. Compte tenu de la complexité des surfaces gauches, une vision plus globale du contexte peut conduire à des partitions plus efficaces (en termes de critères d’usinage). Une analyse des surfaces gauches basée sur l’ACP a été présentée dans le chapitre précédent (et publiée dans [36]) pour l’ensemble de la surface. Elle peut être utilisée de la même manière pour chaque zone individuelle afin de construire une approximation rapide de la forme et de l’orientation de la zone. Le processus de partitionnement peut ensuite s’appuyer sur cette approximation pour exploiter des résultats bien connus sur les processus d’usinage. Par exemple, il est facile de comprendre que plus le parcours de l’outil dans une direction donnée est long, plus l’usinage sera efficace, car les ralentissements dus aux limites cinématiques aux deux extrémités du parcours sont moins influents. Il convient de mentionner que les approches exclusivement locales ne peuvent pas intégrer ce type de résultats.

### 3.3 Discréétisation et création du data-set

Cette section traite de la création de l’ensemble de points de données à partitionner, appelé *data-set*. Cet ensemble de points, appelés *sample-points*, est créé en discréétisant et échantillonnant la surface.

#### 3.3.1 Maillage et *sample-points*

Dans le contexte d’usinage de surfaces gauches, les zones sont définies par un ensemble de points échantillonnés sur la surface. En théorie, n’importe quel nuage de sample-points peut être utilisé, pourvu que les points soient répartis sur toute la surface et la représentent suffisamment bien. En pratique, pour obtenir ce résultat, un maillage régulier de courbes isoparamétriques est défini sur la surface. De cette façon, un ensemble de mailles élémentaires sont également définies (figure 3.1). Pour chaque maille, un sample-point situé en son centre est défini. Le *data-set* est créé à partir de ces points et les algorithmes de clustering opèrent sur ces *sample-points* pour définir des zones.

#### 3.3.2 *Feature-vector* et *data-set*

Quel que soit l’algorithme de classification non-supervisée utilisé, il est nécessaire de définir la notion de distance ou similitude, entre deux points (pour les familles d’algorithmes de type *K-means* et competitive learning) ou entre deux *clusters* (pour la classification hiérarchique). De plus, cette distance doit être calculée en utilisant les paramètres les plus pertinents. Ce choix a un grand impact sur le résultat du processus de partitionnement. Enfin, les paramètres choisis constituent le vecteur de caractéristiques appelé *feature-vector*.

En usinage de surfaces gauches, il est nécessaire de considérer deux différents types de paramètres :

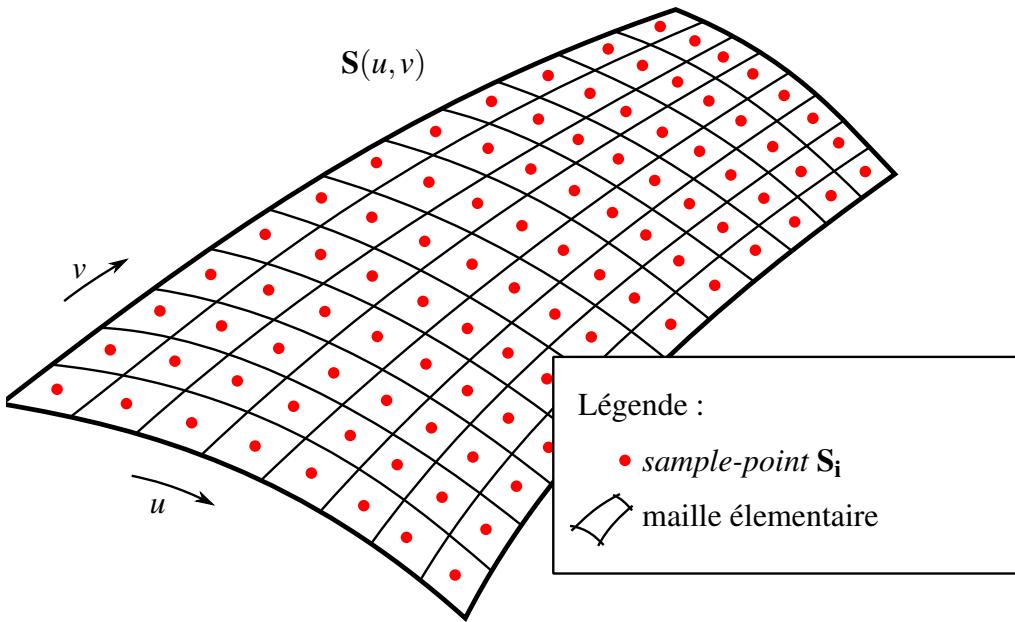


Figure 3.1 – Définition des sample-points

- les paramètres de proximité spatiale : tel que les coordonnées réelles 3D des *sample-points* ou leurs coordonnées paramétriques  $(u, v)$ .
- les paramètres liés à l'usinage : le vecteur normal  $\mathbf{n}(u, v)$ , les courbures gaussiennes, moyennes ou principales.

Une analyse comparative de *feature-vector* avec différents paramètres est présentée dans [63]. Dans cette étude, les auteurs ont choisi un *feature-vector* dont les paramètres sont les coordonnées paramétriques  $u, v$  et le vecteur normal  $\mathbf{n}(u, v)$ . Dans [46], les coordonnées spatiales 3D ( $S_x, S_y, S_z$ ) et le vecteur normal  $\mathbf{n}(u, v)$  ont été choisis.

Dans notre cas, le *feature-vector* choisi est légèrement différent. D'abord, l'utilisation des coordonnées réelles ( $S_x, S_y, S_z$ ) n'est pas complètement adaptée pour les surfaces gauches. En effet, comme le montre la figure 3.2, deux points peuvent être éloignés l'un de l'autre sur une surface tout en ayant une petite distance euclidienne. Dans ce cas, idéalement, la distance géodésique devrait être considérée au lieu de la distance euclidienne. Cependant, la distance géodésique est très coûteuse à calculer et ne peut pas être prise en compte pour une application pratique. Ainsi, la meilleure alternative est de considérer les coordonnées paramétriques  $(u, v)$ . Une telle alternative est préférée car très rapide d'utilisation et compatible avec la métrique euclidienne.

Par ailleurs, au lieu des coordonnées du vecteur normale  $\mathbf{n}(u, v)$ , on choisit des paramètres qui interviennent directement dans le processus d'usinage. Ce choix se base sur deux propriétés bien connues de l'outil torique : premièrement, la direction de plus grande pente est celle qui maximise l'efficacité de l'outil torique. Deuxièmement, plus la valeur de la pente est petite, plus l'outil torique est efficace. Étant donné que le calcul analytique du rayon effectif  $R_{eff}$  pour l'outil torique fait intervenir directement la plus grande pente  $s(u, v)$  est sa direction  $\theta(u, v)$  comme démontré par [62], il est donc plus judicieux de

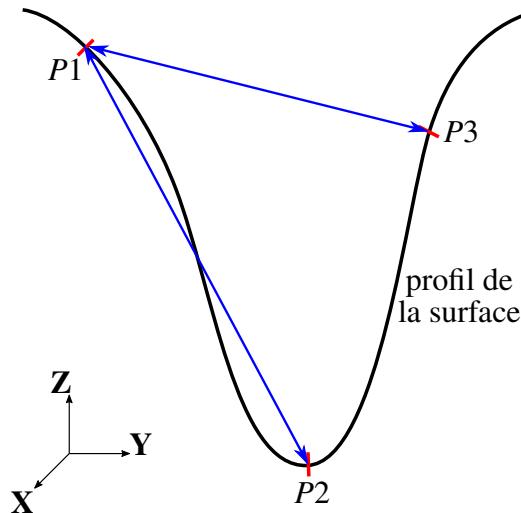


Figure 3.2 – Un exemple montrant 3 points  $P_1$ ,  $P_2$  et  $P_3$  tel que  $P_3$  est plus proche de  $P_1$  que de  $P_2$  en termes de distance euclidienne, alors que  $P_2$  est plus proche selon la distance géodésique

les utiliser comme paramètres du *feature-vector* plutôt que  $\mathbf{n}$ . La figure 3.3, illustre ces angles, définis comme suit :

- $s(u, v)$  : la valeur de la plus grande pente,  $s \in [0, \frac{\pi}{2}]$ , défini comme l'angle entre  $\mathbf{n}(u, v)$  et sa projection sur le plan  $\mathbf{XY}$ .
- $\theta(u, v)$  : l'angle entre l'axe  $\mathbf{X}$  et la projection de  $\mathbf{n}(u, v)$  sur le plan  $\mathbf{XY}$  ramené à l'intervalle  $[-\frac{\pi}{2}, \frac{\pi}{2}]$ .

Pour résumer, les coordonnées paramétriques  $u$  et  $v$  sont utilisées pour exprimer la proximité spatiale entre les sample-points, tandis que les angles  $s(u, v)$  et  $\theta(u, v)$  sont choisis comme paramètres reflétant la performance de l'usinage (figure 3.3) :

Ces choix ont été faits afin de réduire la taille du *feature-vector* et d'être plus précis sur les paramètres qui influencent le processus d'usinage. Le *data-set* est ainsi constitué de l'ensemble de *feature-vector* des sample-points.

## 3.4 Algorithmes de *clustering*

Une fois le *data-set* construit à partir de la surface, les algorithmes de *clustering* vont opérer sur celui-ci pour définir des classes, ou *clusters* du *data-set*, ce qui partitionne la surface en zones différentes. Les algorithmes de *clustering* sont certainement les méthodes les plus adaptées pour partitionner les surfaces gauches. En effet, les algorithmes de classification supervisée requièrent un ensemble d'apprentissage (*learning-set*), or, il est très difficile, voire impossible, de construire un *learning-set* convenable pour toutes les surfaces gauches ni de labelliser les points d'une surface donnée de façon à produire une partition efficace en terme d'usinage. Par conséquent, seuls les algorithmes de classification non-supervisée sont considérés dans cette étude. Par ailleurs, plusieurs algorithmes et variantes ont été développés dans la littérature. On se concentrera donc sur quelques algorithmes représentant une large

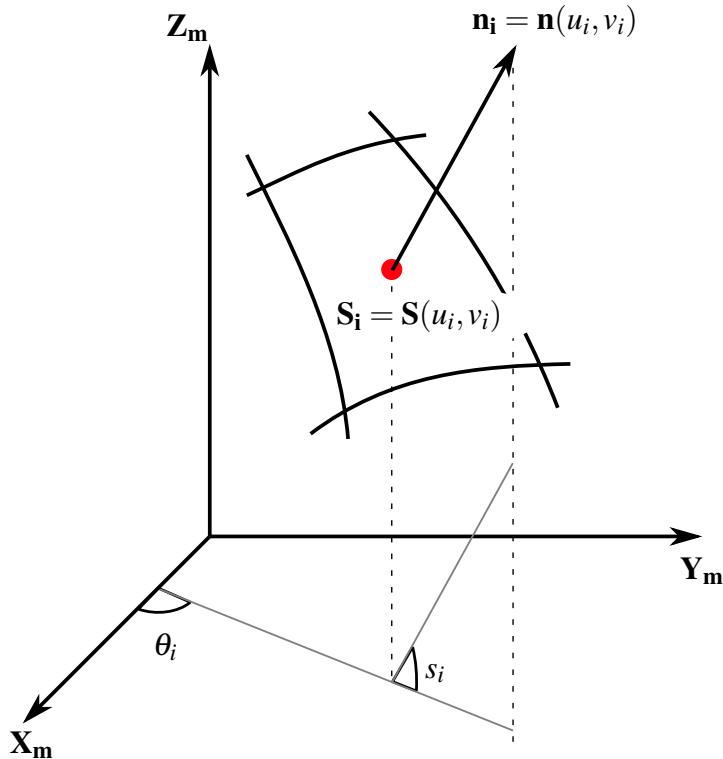


Figure 3.3 – Les quatres composantes ( $u_i$ ,  $v_i$ ,  $s_i$  et  $\theta_i$ ) du *feature-vector*

gamme d’algorithmes de *clustering*. En fait, les méthodes non-supervisées peuvent être divisées en trois grandes familles :

- déterministes basés sur le calcul de centroïdes
- aléatoires de type *competitive-learning*
- classification hiérarchique.

Pour représenter les 3 familles, on choisit les algorithmes suivant : *K-means*, basé sur le calcul des distances par rapport aux centroïdes des *clusters*, et qui représente les algorithmes déterministes tels que *K-medoids* et le *fuzzy C-means*. Par ailleurs, l’algorithme *Rival Penalized Competitive Learning* (RPCL), qui est l’un des algorithmes d’apprentissage compétitif les plus courants, est également choisi. Cet algorithme inclut une composante aléatoire et nécessite moins de calculs de centroïdes. Le dernier algorithme inclu est la classification hiérarchique ascendante (noté HAC pour *Hierarchical ascending clustering*) qui, comme tout algorithme de classification hiérarchique, se libère de la notion de centroïdes de *clusters* en définissant directement les distances entre les *clusters*.

### 3.4.1 L’algorithme *K-means*

*K-means* (Algorithme 1) est l’un des algorithmes de clustering les plus courants. Il a été introduit par [49]. Son cadre théorique et sa preuve de convergence sont présentés dans [28]. Diverses versions

parallélisées de cet algorithme ont également été proposées, par exemple dans [79].

---

**Algorithme 1 : *K-means***

---

**Data :** *Data-set*  $D$ , nombre de *clusters* (classes)  $K$

**Initialisation :**

Définir la classe de chaque point  $\forall X \in D, \text{class}(X) \in \{1, \dots, K\}$

**Répéter jusqu’à convergence :**

**Étape 1 :** calcul des centroïdes :

$$\forall j \in [1; K], \text{center}_j = \frac{1}{\text{card}\{X | \text{class}(X) = j\}} \sum_{X \in D} X \times \mathbf{1}(\text{class}(X) = j)$$

**Étape 2 :** affectation des classes :

$$\forall X \in D, \text{class}(X) = \arg \min_{j \in [1; K]} (\|\text{center}_j - X\|)$$

**return** classe de chaque point :  $\text{class}(X)$

---

### 3.4.2 L’algorithme RPCL

L’algorithme *Rival Penalized Competitive Learning* (RPCL, Algorithme 2), introduit par [78], est un algorithme de classification non-déterministe. Initialement, un certain nombre de centroïdes sont définis. À chaque itération, un point du *data-set* est choisi au hasard et la distance entre chaque centre et ce point est calculée. Le centre gagnant (le plus proche) est rapproché du point en question, tandis que son rival (le deuxième centre le plus proche) est éloigné (c’est-à-dire pénalisé). Contrairement au *K-means*, cet algorithme permet, en écartant les *clusters* additionnels (en trop), d’obtenir des partitions avec moins de *clusters* que le nombre initial. De plus, l’algorithme RPCL prend en compte l’historique du centre gagnant, c’est-à-dire le nombre de fois qu’il a gagné, afin de réduire la sensibilité de l’algorithme par rapport à l’initialisation des centroïdes.

---

**Algorithme 2 : *Rival Penalized Competitive Learning* (RPCL)**

---

**Data :** *Data-set*  $D$ , nombre de centroïdes  $k$ ,  $\alpha_c$  taux d’apprentissage du gagnant,  $\alpha_r$  taux d’apprentissage du rival

**Initialisation :**

Choisir aléatoirement les  $k$  centroïdes  $(\omega_i)_{1 \leq i \leq k}$

**Répéter jusqu’à convergence :**

**Étape 1 :** Choisir aléatoirement un point  $x \in D$  et calculer pour  $i \in [1; k]$  :

$$u_i = \begin{cases} 1, \text{ si } i = c = \arg \min_j \gamma_j \|\vec{x} - \vec{\omega}_j\|^2 \\ -1, \text{ si } i = r = \arg \min_{j \neq c} \gamma_j \|\vec{x} - \vec{\omega}_j\|^2 \\ 0, \text{ sinon} \end{cases}$$

Où  $\forall 1 \leq j \leq k$ ,  $\gamma_j = \frac{n_j}{\sum_{i=1}^k n_i}$  avec  $n_j$  le nombre cumulé d’occurrences de  $u_j = 1$

**Étape 2 :** Mise à jour des centroïdes :  $\Delta \vec{\omega}_i = \alpha_i u_i (\vec{x} - \vec{\omega}_i)$

**return** les centroïdes  $(\omega_i)_{1 \leq i \leq k}$

---

Le RPCL est connu pour être plus rapide que les autres algorithmes, mais sa nature non-déterministe

est susceptible de générer différentes solutions pour la même surface. Cela signifie, d'un point de vue pratique, que le fait de relancer le même algorithme sur la même surface peut conduire à un résultat meilleur (ou pire). Ce n'est pas très pratique à gérer dans le contexte d'un processus industriel.

### 3.4.3 La classification hiérarchique ascendante

La classification hiérarchique ascendante (HAC), détaillée dans l'algorithme 3, est aussi largement utilisée dans la classification non-supervisée. Au départ, chaque individu forme une classe, c'est-à-dire qu'on a  $n$  classes. Le nombre de classes est réduit itérativement à  $n_c < n$ . À chaque étape, les deux classes les plus proches (les plus similaires) sont fusionnées. Pour plus de détails, on peut se référer à [70]. Plusieurs mesures de dissimilarité existent dans la littérature, la plus connue est la distance de variance minimale de Ward (voir section 3.5.4).

---

#### Algorithme 3 : Classification Hiérarchique Ascendante

---

**Data :** Data-set  $D = \{x_i, i = 1 \dots n\}$

**Initialisation :**

$clusters : \forall 1 \leq i \leq n, c_i = \{x_i\}$

$distances : \forall 1 \leq i < j \leq n, d_{ij} = \|x_i - x_j\|$

**Répéter jusqu'à convergence :**

**Étape 1 :**  $(i^*, j^*) = \arg \min_{i < j} (d_{ij})$   
 $c_{i^*} = c_{i^*} \cup c_{j^*}$  et  $c_{j^*} = \emptyset$

**Étape 2 :**  $\forall k$  mettre à jour  $d_{i^*k}$  (ou  $d_{ki^*}$  si  $k < i^*$ ) en utilisant la distance de Ward

$\forall k, d_{j^*k} = d_{kj^*} = +\infty$

**return** clusters  $c_i$

---

L'avantage du HAC, par rapport au *K-means* et au RPCL, est qu'il n'est pas nécessaire de définir le nombre de classes. En effet, dans la littérature, le nombre de classes communément retenu correspond à l'itération avec le saut de distance maximum. Cependant, la classification hiérarchique prend plus de temps de calcul que les autres algorithmes, notamment parce que le maillage de la surface devant être assez fin pour des raisons de précision, cet algorithme opère sur un très grand nombre de classes au départ. De plus, la classification hiérarchique ne permet aucune possibilité de retour en arrière, la fusion de deux classes étant en effet irréversible.

## 3.5 Métriques classiques

Une fois l'algorithme de clustering et les paramètres du *feature-vector* pertinents sélectionnés, une métrique (ou plus généralement, une mesure de dissimilarité) doit être définie. La métrique définit comment la distance, ou plus précisément la similarité, entre deux points (ou *clusters*) est calculée. En fait, le résultat du processus de partitionnement dépend fortement du choix de la métrique.

Dans ce qui suit, les mesures de dissimilarité entre deux points sont appelées métriques par raccourci. Une telle définition n’inclut pas nécessairement toutes les propriétés d’une métrique au sens mathématique du terme, mais elle est suffisante pour notre objectif. C’est pourquoi cette dénomination a été retenue dans un souci de simplicité.

Différentes métriques sont testées ci-dessous avec différents algorithmes de clustering. Dans la section 3.6, une nouvelle métrique, spécifiquement dédiée à l’usinage de surfaces de gauches, est introduite.

La surface  $S$  est discrétisée avec un maillage régulier de courbes isoparamétriques. Soit  $n$  le nombre de mailles. Pour chaque maille  $i \in \{1, \dots, n\}$ , le *feature-vector* est défini au point central de la maille :  $x_i = (u_i, v_i, s_i, \theta_i)^T$ . Conformément à ce qui est présenté en section 3.4, ces points représentent notre ensemble de données  $D$ . Dans ce qui suit, la maille numéro  $i$  est désignée par  $m_i$ , et le *data-point* qui lui est attaché est  $x_i$ . De plus, étant donné une partition de  $S$  constituée de  $K$  zones, le centre de la zone  $z_k$ ,  $k \in \{1, \dots, K\}$ , est désigné par  $c_k = (\bar{u}_k, \bar{v}_k, \bar{s}_k, \bar{\theta}_k)$ . Les définitions ci-dessous utilisent ces paramètres.

### 3.5.1 Distance euclidienne

Dans l’espace quadridimensionnel des paramètres  $(u, v, s, \theta)$ , la distance euclidienne  $d$  entre une maille  $m_i$  et le centre d’une zone  $z_k$  est définie par :

$$d^2(x_i, c_k) = (u_i - \bar{u}_k)^2 + (v_i - \bar{v}_k)^2 + (s_i - \bar{s}_k)^2 + (\theta_i - \bar{\theta}_k)^2$$

Cette distance donne la même importance à toutes les variables, ce qui n’est pas nécessairement optimal. En outre, plusieurs termes qui ne sont pas homogènes (unités et ordres de grandeur différents) sont additionnés, ce qui manque de signification physique. L’avantage de la métrique euclidienne réside dans sa simplicité de mise en œuvre.

### 3.5.2 Distance de Mahalanobis

Cette distance est largement utilisée en statistique et en traitement de signal [53]. Elle diffère de la distance euclidienne, car elle prend en compte la variance et la corrélation de l’ensemble des données. Ainsi, contrairement à la distance euclidienne où toutes les composantes des vecteurs sont traitées indépendamment et de la même manière, cette métrique donne un poids plus faible aux variables les plus dispersées et prend en compte la corrélation entre les variables.

Soit  $\Sigma$  la matrice de covariance des *feature-vectors* sur le maillage de toute la surface :

$$\Sigma = \begin{pmatrix} \text{Var}(u) & \text{Cov}(u, v) & \text{Cov}(u, s) & \text{Cov}(u, \theta) \\ \text{Cov}(u, v) & \text{Var}(v) & \text{Cov}(v, s) & \text{Cov}(v, \theta) \\ \text{Cov}(u, s) & \text{Cov}(v, s) & \text{Var}(s) & \text{Cov}(s, \theta) \\ \text{Cov}(u, \theta) & \text{Cov}(v, \theta) & \text{Cov}(s, \theta) & \text{Var}(\theta) \end{pmatrix}$$

La distance  $d$  entre une maille  $m_i$  et le centre de la zone  $z_k$  est définie par :

$$d^2(x_i, c_k) = (x_i - c_k)^T \Sigma^{-1} (x_i - c_k)$$

### 3.5.3 Distance basée sur les écarts-types

Cette distance est également utilisée en statistique pour le traitement de données hétérogènes ; avec différents ordres de grandeur et variances. Chaque terme est divisé par l'écart-type correspondant. Ainsi, la distance entre une maille  $m_i$  et le centre d'une zone  $z_k$  est définie par :

$$d^2(x_i, c_k) = \frac{(u_i - \bar{u}_k)^2}{\sigma_u} + \frac{(v_i - \bar{v}_k)^2}{\sigma_v} + \frac{(s_i - \bar{s}_k)^2}{\sigma_s} + \frac{(\theta_i - \bar{\theta}_k)^2}{\sigma_\theta}$$

où  $\sigma_u = \sqrt{\text{Var}(u)}$  est l'écart type de la variable  $u$  sur toute la surface  $S$ . En employant une définition similaire pour les autres variables, la distance peut être également exprimée comme :

$$\begin{aligned} d^2(x_i, c_k) &= (x_i - c_k)^T \begin{pmatrix} \sigma_u & 0 & 0 & 0 \\ 0 & \sigma_v & 0 & 0 \\ 0 & 0 & \sigma_s & 0 \\ 0 & 0 & 0 & \sigma_\theta \end{pmatrix}^{-1} (x_i - c_k) \\ &= (x_i - c_k)^T D_\sigma^{-1} (x_i - c_k) \end{aligned}$$

Cette distance pourrait être considérée comme un cas particulier de la distance de Mahalanobis, où la covariance des différentes paires de variables n'est pas prise en compte. En fait, la distance basée sur l'écart-type ne tient pas compte de la corrélation entre les variables. Pourtant, elle permet la sommation de différents termes puisque le poids affecté à chaque terme est proportionnel à sa variance, ce qui n'est pas le cas de la distance euclidienne.

Il convient de mentionner que ces trois distances sont dérivées de la forme générale de distance  $d$  suivante sur  $\mathbb{R}^4$ ,  $d^2(x, y) = (x - y)^T M (x - y)$  où  $M$  est une matrice définie positive symétrique.

### 3.5.4 Distance de Ward

Contrairement aux distances précédentes, la distance de Ward est destinée à mesurer la dissimilarité entre les *clusters* plutôt que la dissimilarité entre les data-points ; cela signifie que, parmi les algorithmes présentés précédemment, la distance de Ward ne peut être utilisée qu'avec l'algorithme HAC.

Pour tout cluster  $C_k$  contenant  $n_k$  data-points, soit  $c_k$  son centroïde :

$$c_k = \frac{1}{n_k} \sum_{x \in C_k} x$$

et  $\zeta(C_k)$  la somme des distances euclidiennes au carré calculées des points de  $C_k$  par rapport à son centre  $c_k$  :

$$\zeta(C_k) = \sum_{x \in C_k} \|x - c_k\|^2$$

Pour toute paire de *clusters* disjoints  $C_i$  et  $C_j$  (de taille respective  $n_i$  et  $n_j$ ), la distance de Ward indique de combien cette somme de carrés augmentera lorsque les deux *clusters* sont fusionnées :

$$d(C_i, C_j) = \zeta(C_i \cup C_j) - \zeta(C_i) - \zeta(C_j) = \frac{n_i n_j}{n_i + n_j} \|c_i - c_j\|^2$$

Lorsque deux *clusters*  $C_i, C_j$  sont fusionnés, la distance entre  $C_i \cup C_j$  et tout autre cluster  $C_k$  (de taille  $n_k$ ) peut être mise à jour en utilisant la relation suivante :

$$d(C_i \cup C_j, C_k) = \frac{1}{n_i + n_j + n_k} ((n_i + n_k) d(C_i, C_k) + (n_j + n_k) d(C_j, C_k) - n_k d(C_i, C_j)) \quad (3.1)$$

## 3.6 Métrique basée sur l’ACP

Une nouvelle métrique basée sur l’ACP est proposée dans cette section.

### 3.6.1 Considérations préliminaires

L’Analyse en Composantes Principales (ACP) est une procédure statistique multivariée pour l’analyse des données. Elle est principalement utilisée pour la réduction des dimensions : Elle vise à condenser les informations contenues dans un grand nombre de variables originales en un ensemble plus restreint de nouvelles variables orthogonales au moyen de combinaisons linéaires, avec un minimum de perte d’informations. Mathématiquement, l’ACP repose sur la décomposition en éléments propres des matrices semi-définies positives.

Dans notre contexte, l’ACP permet de trouver la direction le long de laquelle un cluster  $k$  est le plus étendu. Dans ce qui suit, cette direction est appelée direction principale. En effet, cette direction est particulièrement intéressante puisqu’elle peut être directement liée à la longueur des trajectoires de l’outil. Elle peut ensuite être utilisée pour s’assurer que la vitesse de consigne est atteinte sur une plus grande proportion des trajectoires.

En pratique, la direction principale d’un cluster  $k$  est trouvée en effectuant une décomposition en éléments propres de  $\Gamma_k$ , la matrice de covariance des coordonnées réelles ( $S_x, S_y, S_z$ ) des sample-points  $\mathbf{S}_i$  appartenant au cluster  $k$ . Ainsi, la direction principale est définie par le vecteur propre  $\mathbf{X}_I$  correspondant à la plus grande valeur propre  $\lambda_I$  de  $\Gamma_k$ . Cette procédure est décrite dans le chapitre 2.

### 3.6.2 Définition de la métrique

L'objectif de la nouvelle métrique que nous présentons est de fournir une vision globale de la zone, en s'appuyant sur des considérations pratiques d'usinage, afin d'améliorer le processus de partitionnement. Ceci est possible en introduisant la direction principale des zones, afin de promouvoir certaines formes de zones. Plus explicitement, si la direction principale d'une zone coïncide avec sa direction de plus forte pente, un gain drastique est observé en termes de temps d'usinage pour la zone concernée. Il convient donc de promouvoir ce cas de figure par rapport à d'autres.

Ainsi, pour évaluer la distance entre une maille  $i$  et le centre d'une zone  $k$ , deux termes sont ajoutés à la distance euclidienne (voir figure 3.4) : Le premier terme est déterminé par l'angle entre la direction principale et la direction définie par  $m_i$  et  $c_k$ . Ce terme est utile pour pénaliser les mailles qui sont éloignées de la direction principale actuelle de la zone. Le second terme est déterminé par l'angle entre la direction principale et la direction de la pente la plus forte. Ce terme garantit que la direction principale résultante ne s'éloigne pas trop de la direction de la plus grande pente.

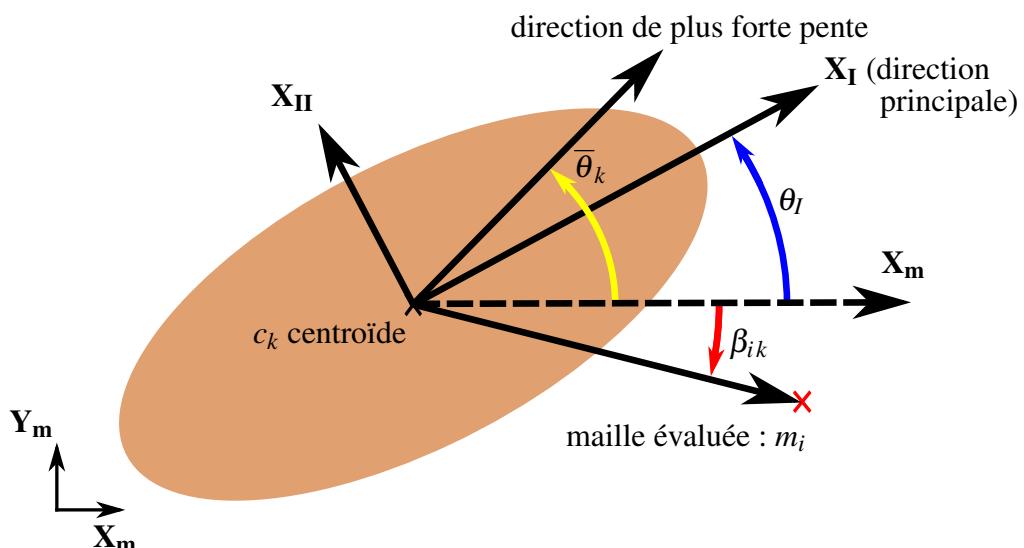


Figure 3.4 – Les différents angles impliqués dans l'évaluation de la distance basée sur l'ACP

Une expression de la distance  $d$  basée sur l'ACP, entre  $m_i$  et  $c_k$ , est proposée dans l'équation 3.2 ci-dessous. Cette expression n'est pas la seule et probablement pas la meilleure façon de prendre en compte la direction principale, mais cette question pourra faire l'objet d'investigations à part.

$$d^2(x_i, c_k) = (u_i - \bar{u}_k)^2 + (v_i - \bar{v}_k)^2 + (s_i - \bar{s}_k)^2 + (\theta_i - \bar{\theta}_k)^2 + f \left( \frac{\lambda_{II}}{\lambda_I} \right) \left[ \sin^2(\theta_I - \beta_{ik}) + \sin^2(\bar{\theta}_k - \theta'_I) \right] \quad (3.2)$$

où :

- $\lambda_I$  et  $\lambda_{II}$  sont la première et deuxième valeur propre de  $\Gamma_k$ ,

- $f\left(\frac{\lambda_{II}}{\lambda_I}\right)$  est une fonction qui détermine comment évolue le compromis entre la direction de plus grande pente et la direction principale en fonction du rapport  $\frac{\lambda_{II}}{\lambda_I}$ .
- $\theta_I$  est l’angle orienté entre l’axe  $\mathbf{X}_m$  du repère machine et la direction principale de la zone  $k$ ,
- $\bar{\theta}_k$  est l’angle orienté entre l’axe  $\mathbf{X}_m$  de la machine et la direction de plus grande pente de la zone  $k$ ,
- $\beta_{ik}$  est l’angle orienté entre l’axe  $\mathbf{X}_m$  et le vecteur  $\mathbf{S}(u_i, v_i) - \mathbf{S}(u_k, v_k)$ ,
- $\theta'_I$  définie la nouvelle direction principale de la zone  $k$  dans le cas où la maille  $m_i$  lui est affectée.

### 3.6.3 Quelques remarques

Comme mentionné précédemment, l’expression proposée de la métrique basée sur l’ACP n’est pas unique. Cependant, l’expression 3.2 garantit que les points qui sont plus proches de la direction principale d’une zone ont plus de chances de rejoindre cette zone, ce qui favorise l’apparition de zones avec des directions principales distinguées. En outre, elle garantit le compromis avec la direction de plus forte pente, de sorte que les pas transversaux ne se détériorent pas trop.

L’expression de  $f\left(\frac{\lambda_{II}}{\lambda_I}\right)$  a également un impact sur les partitions résultantes. Dans ce document, nous pouvons proposer deux expressions (équation 3.3). Chaque expression correspond à un point de vue différent : le rapport  $\frac{\lambda_{II}}{\lambda_I}$  est compris entre 0 et 1. Il tend vers 0 lorsque la direction principale est distinguée et vers 1 dans le cas contraire. Par conséquent, le choix de l’expression de  $f$  a un impact sur la convergence et le résultat obtenu. L’effet du choix de l’expression peut être étudié dans un travail séparé.

$$f\left(\frac{\lambda_{II}}{\lambda_I}\right) = \left(\frac{\lambda_{II}}{\lambda_I}\right)^2 \text{ ou } f\left(\frac{\lambda_{II}}{\lambda_I}\right) = 1 - \left(\frac{\lambda_{II}}{\lambda_I}\right)^2 \quad (3.3)$$

Le point commun entre les métriques présentées dans la section 3.5 est qu’elles n’utilisent toutes que des informations locales, sur un data-point donné et le centroïde d’un cluster, pour décider de l’affection. Contrairement aux métriques précédentes, la métrique basée sur l’ACP calcule la distance entre un data-point donné et le centroïde d’un cluster, tout en impliquant des informations globales qui sont directement liées à la forme du cluster. Ainsi de la connaissance-métier peut être incluse dans cette formulation. Les *clusters* obtenus à l’aide de cette métrique devraient être usinés dans des durées plus courtes. Bien entendu, l’efficacité de cette métrique dépend des propriétés géométriques de la surface, comme nous le verrons plus loin.

À titre exploratoire, la métrique basée sur l’ACP et la métrique euclidienne considérée comme référence ont été testées sur un ensemble de surfaces générées aléatoirement. Les temps d’usinage obtenus par les deux métriques ont été comparés et les résultats ont montré que la métrique basée sur l’ACP peut réduire le temps d’usinage avec un gain de 10% en moyenne par rapport à la métrique euclidienne, ce qui révèle le potentiel de la métrique proposée.

## 3.7 Détails d'implémentation

Dans cette section sont expliqués quelques détails sur l'implémentation de la procédure de partitionnement. Par exemple, le choix du point initial et critère d'arrêt, la mise à jour incrémentale et récursive de la matrice de covariance, qui permet un gain drastique de temps de calcul, et enfin la nécessité de la recherche de composantes connexes.

### 3.7.1 Partition initiale

Tout d'abord, la plupart des algorithmes de classification nécessitent un point de départ. Le résultat dépend plus ou moins de la partition initiale, et la sensibilité d'un algorithme donné au choix du point de départ peut être étudiée. Dans le cas du *K-means* ou du RPCL, le point initial est défini en fixant les positions initiales des centroïdes. Les centroïdes peuvent être soit choisis au hasard dans l'espace paramétrique, soit répartis régulièrement sur l'espace paramétrique. En pratique, il s'avère que le fait de placer les points initiaux uniformément sur un cercle dont le centre coïncide avec le centre de l'espace paramétrique et dont le diamètre est égal à la moitié de la plus petite limite de l'espace paramétrique donne de bons résultats. Par exemple, l'espace paramétrique des surfaces Bézier est défini par  $(u, v) \in [0, 1]^2$ . En considérant les  $K$  centroïdes, les coordonnées paramétriques du centre  $k \in [1, K]$  sont données par les équations 3.4.

$$\begin{cases} \bar{u}_k = \frac{1}{2} + \frac{1}{4} \cos \frac{2k\pi}{K} \\ \bar{v}_k = \frac{1}{2} + \frac{1}{4} \sin \frac{2k\pi}{K} \end{cases} \quad (3.4)$$

Les deux paramètres restants sont pris en ce point. En d'autres termes, la valeur initiale de  $\bar{s}_k$  est  $s(\bar{u}_k, \bar{v}_k)$ , et la valeur initiale de  $\bar{\theta}_k$  est  $\theta(\bar{u}_k, \bar{v}_k)$ .

### 3.7.2 Calcul incrémental de la matrice de covariance

L'utilisation d'un algorithme de clustering avec certaines métriques telles que la distance de Mahalanobis, la distance basée sur les écarts-types, ou encore la distance basée sur l'ACP, requiert la matrice de covariance pour chaque zone et à chaque itération. Sachant que les zones évoluent constamment, l'approche de base (naïve) consiste à calculer à chaque fois la matrice de covariance, ce qui s'avère être très coûteux en termes de temps de calcul et en espace mémoire. Ceci limite la finesse du maillage, et par la suite la précision du partitionnement et de l'usinage.

Il est donc intéressant de pouvoir mettre à jour les matrices de covariance des zones concernées, à chaque fois qu'une maille change de zone. Dans la plupart des algorithmes, lorsqu'une zone est modifiée, il n'y a que deux possibilités : soit une nouvelle maille lui est attribuée, soit une maille en est supprimée. Il est ainsi proposé de calculer la nouvelle matrice de covariance à partir de l'ancienne, en utilisant une formule de récurrence dans chaque cas.

Ces formules peuvent être établies pour toute taille de cluster et toute dimension des *feature-vector* : soit  $n \in \mathbb{N}$  le nombre d'éléments d'un cluster donné,  $\mathbf{X}_i \in \mathbb{R}^p$  le *feature-vector* de la maille  $i$ , où  $p$  est la dimension de l'espace des *features*. Le vecteur  $\mathbf{E}_n \in \mathbb{R}^p$  désigne le *feature-vector* moyen de la zone, tandis que  $\Sigma_n \in \mathbb{R}^{p \times p}$  désigne sa matrice de covariance.

- **Cas d'ajout d'une maille.** En considérant une nouvelle maille de vecteur  $\mathbf{X}_{n+1}$  ajoutée à la zone donnée, le nombre de mailles est maintenant de  $n + 1$ . Le vecteur moyen de la nouvelle zone est  $\mathbf{E}_{n+1}$  et sa matrice de covariance est  $\Sigma_{n+1}$ . L'équation 3.5 permet le calcul de  $\Sigma_{n+1}$  à partir de  $\Sigma_n$ ,  $\mathbf{E}_n$  et  $\mathbf{X}_{n+1}$ . La preuve de cette formule est donnée en annexe D.1.

$$\Sigma_{n+1} = \frac{n}{n+1} (\Sigma_n + \mathbf{E}_n \mathbf{E}_n^T) + \frac{\mathbf{X}_{n+1} \mathbf{X}_{n+1}^T}{n+1} - \left( \frac{n}{n+1} \mathbf{E}_n + \frac{\mathbf{X}_{n+1}}{n+1} \right) \left( \frac{n}{n+1} \mathbf{E}_n + \frac{\mathbf{X}_{n+1}}{n+1} \right)^T \quad (3.5)$$

- **Cas de retrait d'une maille.** Si l'on considère maintenant qu'une maille  $r \in [1, n]$  a été retirée de la zone, le nombre de mailles est maintenant de  $n - 1$ . Le vecteur moyen de la nouvelle zone est  $\mathbf{E}_{n-1}$  et sa matrice de covariance est  $\Sigma_{n-1}$ . L'équation 3.6 permet le calcul de  $\Sigma_{n-1}$  à partir de  $\Sigma_n$ ,  $\mathbf{E}_n$  et  $\mathbf{X}_r$ . La preuve de cette formule est donnée en annexe D.2.

$$\Sigma_{n-1} = \frac{n}{n-1} (\Sigma_n + \mathbf{E}_n \mathbf{E}_n^T) - \frac{\mathbf{X}_r \mathbf{X}_r^T}{n-1} - \left( \frac{n}{n-1} \mathbf{E}_n - \frac{\mathbf{X}_r}{n-1} \right) \left( \frac{n}{n-1} \mathbf{E}_n - \frac{\mathbf{X}_r}{n-1} \right)^T \quad (3.6)$$

En pratique, une maille peut changer de zone à chaque itération au cours de l'étape d'affectation, ainsi l'équation 3.6 s'applique à la zone d'origine et l'équation 3.5 s'applique à la zone de destination. Si une maille ne change pas de zone, ce qui arrive plus souvent lorsque le partitionnement converge, il n'est pas nécessaire de mettre à jour la matrice de covariance. Cela permet de gagner beaucoup de temps de calcul, notamment lors des dernières itérations.

### 3.7.3 Critères d'arrêt

Pour certains algorithmes de clustering, le choix des critères d'arrêt est crucial. En effet, la partition finale peut être fortement dépendante des critères d'arrêt. En outre, ces critères peuvent différer selon l'algorithme de classification. Ainsi, il n'est pas possible de choisir le même critère pour les trois algorithmes proposés dans cette étude.

Pour l'algorithme *K-means*, le choix est simple : l'algorithme converge lorsque plus aucun centroïde ne bouge, ce qui signifie que la partition ne change plus. La convergence du *K-means* peut être prouvée si la métrique utilisée est une norme euclidienne. Dans le cas contraire, la convergence n'est pas garantie, un second critère d'arrêt sur le nombre maximum d'itérations est donc recommandé.

Le choix des critères d'arrêt pour l'algorithme RPCL est beaucoup plus complexe compte tenu du caractère aléatoire de l'algorithme. En pratique, les taux d'apprentissage (gagnant et rival) sont contraints de tendre vers 0 lorsque le nombre d'itérations tend vers l'infini. Ainsi, un premier critère d'arrêt consiste à arrêter l'algorithme lorsque ces deux paramètres sont inférieurs à une tolérance pré-définie par l'uti-

lisateur. Ce critère s'avère insuffisant, car le partitionnement ainsi obtenu dépend de la façon dont les taux d'apprentissage tendent vers 0. Un second critère d'arrêt peut être défini en effectuant l'étape d'affectation à chaque itération (étape 3 dans la boucle) dans l'algorithme 2, où chaque maille est affectée à son centre le plus proche. De cette façon, à une itération donnée, le nombre de mailles qui ont changé de zone peut être calculé. Le deuxième critère d'arrêt consiste donc à vérifier, à chaque itération, si le rapport du nombre de mailles modifiées sur le nombre total de mailles est inférieur à un seuil pré-définie par l'utilisateur (1% par exemple) en dessous duquel on considère qu'il y a convergence.

Concernant l'algorithme HAC, le nombre maximum d'itérations est égal au nombre de mailles moins une, puisque, à chaque itération, les deux classes les plus proches sont fusionnées. Pour une itération donnée, le saut de distance est défini comme étant la distance entre les deux classes fusionnées. La partition qui correspond à l'itération avec le saut de distance maximum est conservée. En pratique, l'algorithme est exécuté jusqu'à ce qu'il ne reste plus que deux zones. Ensuite, la partition qui précède immédiatement l'étape de fusion avec le saut de distance maximale est retenue comme résultat. Ce critère est couramment utilisé dans la littérature.

### 3.7.4 Recherche de composantes connexes

Étant donné le *feature-vector* choisi, les algorithmes de clustering ne garantissent pas un partitionnement où tous les *clusters* sont connexes. En effet, si le *feature vector* ne contient que des paramètres de proximité tels que :  $u$ ,  $v$  ou  $\mathbf{S}(u, v)$ , les *clusters* résultants seront plutôt connectés. Cependant, puisque deux points géométriquement éloignés peuvent avoir des paramètres d'usinage similaires (pente  $s$  et direction de plus grande pente  $\theta$ ), une classification utilisant un *feature-vector* contenant de tels paramètres peut conduire à des *clusters* non-connexes. Bien entendu, les zones non-connexes ne peuvent pas être usinées en une seule fois et doivent être divisées en autant de zones connexes que nécessaire.

Il faut donc vérifier, avant l'usinage, que toutes les zones sont connexes et de pouvoir identifier, dans les zones non-connexes, les composants connexes et les considérer comme des zones indépendantes. Ceci peut être réalisé en utilisant des algorithmes issus de la théorie des graphes.

En fait, ce problème pourrait être considéré comme une recherche de composantes connexes (CCS) bien connue [66] dans un graphe non orienté. Pour appliquer un algorithme CCS, un graphe est associé à une partition donnée, les sommets du graphe correspondant aux mailles de la surface. Deux sommets sont liés s'ils représentent deux mailles adjacentes appartenant à la même zone. Par conséquent, trouver les composantes connexes du graphe équivaut à trouver les régions connexes parmi chaque zone. Le problème est résolu en utilisant un algorithme de recherche de composantes connexes basé sur l'algorithme Breadth-First Search (BFS).

Cependant, le temps de calcul nécessaire pour identifier les composantes connexes peut être non négligeable lorsqu'un algorithme de CCS est utilisé, surtout si le maillage est fin, ce qui correspond à un grand nombre de sommets dans le graphe. Dans ce cas, le BFS peut prendre beaucoup de temps.

Un algorithme plus rapide, inspiré des algorithmes du Connected Components Labeling (CCL) [35]

a été développé. Aussi appelés extraction de blob ou étiquetage de régions, les algorithmes CCL sont à l’origine des techniques de traitement et d’analyse d’images, visant à regrouper les pixels d’une image en composants à l’aide d’une heuristique donnée.

L’algorithme que nous avons développé travaille sur le résultat brut du processus de clustering, c’est-à-dire un tableau bidimensionnel  $zID$ , dont les tailles  $tessU$  et  $tessV$  sont les nombres de mailles dans chaque dimension paramétrique. Dans ce tableau, sont stockés les identifiants de zone uniques de chaque maille. À ce stade, les zones non-connexes n’ont pas encore été détectées, donc les régions non-connexes d’une même zone ont le même numéro. Pour stocker le résultat de la procédure, un tableau bidimensionnel de même taille, appelé  $labels$ , est également créé, et toutes ses valeurs sont initialisées à  $null$ . Le principe de l’algorithme développé est décrit en pseudo-code dans l’algorithme 4.

---

**Algorithme 4 :** Principe d’algorithme de type CCL pour la recherche de composantes connexes

---

**Data :** array  $zID$ , array  $labels$ ,  $tessU$ ,  $tessV$

**Function**  $label(i, j, l)$  :

```

 $labels[i][j] = l$ 
if  $zID[i - 1][j] = zID[i][j]$  and  $labels[i - 1][j] = null$ 
|  $label(i - 1, j, l)$ 
if  $zID[i][j - 1] = zID[i][j]$  and  $labels[i][j - 1] = null$ 
|  $label(i, j - 1, l)$ 
if  $zID[i + 1][j] = zID[i][j]$  and  $labels[i + 1][j] = null$ 
|  $label(i + 1, j, l)$ 
if  $zID[i][j + 1] = zID[i][j]$  and  $labels[i][j + 1] = null$ 
|  $label(i, j + 1, l)$ 

```

**Function**  $main$  :

```

 $l = 0$ 
 $label(0, 0, l)$ 
for  $i = 0$  to  $tessU - 1$ 
| for  $j = 0$  to  $tessV - 1$ 
| | if  $labels[i][j] = null$ 
| | |  $l = l + 1$ 
| | |  $label(i, j, l)$ 

```

---

La principale différence entre le CCS et l’algorithme de type CCL que nous avons développé est que l’algorithme BFS doit parcourir un large graphe pour trouver chaque maille appartenant au même cluster alors que cela est fait par une fonction récursive dans le cas du CCL. Par conséquent, l’algorithme CCL est beaucoup plus rapide que le CCS. En effet, plusieurs tests ont été effectués et ont montré que le temps de calcul est, en moyenne, 35 fois plus rapide que le CCS.

Dans ce qui suit, le processus de recherche de composantes connexes est considéré comme étant effectué et, sans perte de généralité, les zones non-connexes sont considérées comme des zones différentes.

## 3.8 Comparaison des résultats

Dans cette section, les résultats des algorithmes *K-means*, RPCL et HAC sont présentés et comparés. Il convient de noter que le but de ces tests n'est pas de minimiser le temps d'usinage pour une surface donnée, mais de comparer les résultats obtenus pour différents cas, combinant la direction d'usinage, l'algorithme de clustering et la métrique utilisés. En ce qui concerne le *feature-vector*, le vecteur  $(u, v, s, \theta)$  est utilisé dans les tests. Ce choix est justifié par l'analyse fournie dans la section 3.3.2.

### 3.8.1 Protocole de tests

À des fins de comparaison, le nombre de *clusters* est fixé à  $K = 3$  et le nombre maximal d'itérations est limité à 300 pour *K-means* et RPCL. Les algorithmes sont testés sur deux surfaces différentes (figure 3.5), discrétisées à l'aide d'un maillage isoparamétrique régulier de taille  $80 \times 80$ . La première surface est utilisée dans [21] et la seconde dans [73]. Pour cette raison, dans la suite de ce chapitre, elles seront appelées surface de Choi et surface de Rubio.

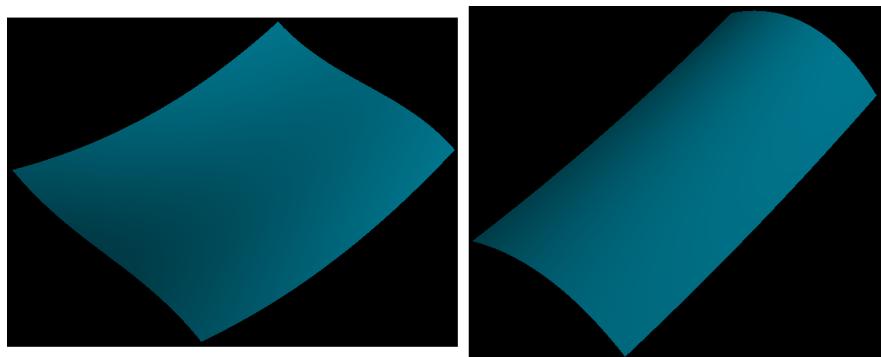


Figure 3.5 – Surfaces test : surface de Choi (à gauche) et surface de Rubio (à droite)

Pour chaque zone, deux directions d'usinage sont considérées :

- la direction de plus forte pente de la zone.
- la direction principale de la zone, obtenue par ACP.

Ces deux directions ont été prises en compte, car elles sont typiques des processus d'usinage. En effet, il est bien connu que la fraise torique donne ses meilleurs résultats lorsqu'elle est utilisée dans la direction de plus forte pente. Mais il a également été souligné dans [36] que, dans certains cas, l'usinage avec une fraise sphérique dans la direction principale peut conduire à de meilleurs résultats que l'usinage avec la fraise torique dans la direction de plus forte pente. Ce cas se présente surtout lorsque ces deux directions sont presque perpendiculaires. Pour être sûr de ne négliger aucun cas, les deux directions typiques ont été incluses dans cette étude.

L'outil torique de rayons  $R$  et  $r$  est utilisé lors de l'usinage dans la direction de plus forte pente. Pour la direction principale, l'outil torique et l'outil sphérique (rayon  $R$ ) ont été testés, bien que seul le meilleur résultat parmi ces deux soit présenté dans les résultats suivants.

### 3.8.2 Analyse des résultats

Les résultats de chaque algorithme sont comparés en termes de partition finale, de longueur du parcours d’outil, de durée d’usinage ainsi que de temps nécessaire pour effectuer la classification. Le tableau 3.1, respectivement 3.2, résume les durées et longueurs d’usinage correspondant à la surface de Choi, respectivement de Rubio, pour chaque algorithme de classification, métrique et direction d’usinage.

TABLE 3.1 – Résultats des tests sur la surface de Choi (les durées sont exprimées en secondes et les longueurs en millimètres)

direction	métrique	<i>K-means</i>		RPCL		HAC (métrique de Ward)	
		durée	longueur	durée	longueur	durée	longueur
plus forte pente	euclidienne	104.7	4763	98.7	4861		
	ACP	96.2	4919	95.8	5495	117.1	4454
	Mahalanobis	125.2	6179	105.9	5777		
	écart-type	100.1	4800	101.2	4780		
principale	euclidienne	107.8	6230	102.2	6333		
	ACP	106.5	6190	110.9	6356	136.8	6302
	Mahalanobis	108.0	6593	104.1	6652		
	écart-type	111.1	6426	110.7	6447		

TABLE 3.2 – Résultats des tests sur la surface de Rubio (les durées sont exprimées en secondes et les longueurs en millimètres)

direction	métrique	<i>K-means</i>		RPCL		HAC (métrique de Ward)	
		durée	longueur	durée	longueur	durée	longueur
plus forte pente	euclidienne	91.6	4634	97.0	4726		
	ACP	91.5	4626	91.7	4593	96.2	4630
	Mahalanobis	88.7	4935	99.8	6277		
	écart-type	91.5	4612	94.3	4966		
principale	euclidienne	84.3	5881	90.7	5940		
	ACP	84.3	5889	84.1	5869	85.3	5850
	Mahalanobis	89.0	5519	85.7	5874		
	écart-type	83.0	5808	90.1	6009		

Les algorithmes RPCL et *K-means* donnent tous deux des résultats proches en termes de temps et longueurs d’usinage, même si RPCL peut parfois conduire à des partitions avec un nombre différent de zones. Par exemple, les deux algorithmes donnent 3 zones pour la surface de Rubio, et 4 zones (en raison de la procédure de recherche de composantes connexes) pour la surface de Choi ; pourtant, RPCL peut parfois conduire à moins de zones.

Les meilleurs résultats ont été obtenus la plupart du temps par la métrique basée sur l’ACP. En effet, la métrique basée sur l’ACP est intéressante, car elle tente d’aligner la direction de la pente la plus forte

avec la direction principale. Ceci est illustré par les figures 3.6 et 3.7, qui montrent le partitionnement et l’usinage de la surface de Choi dans les deux cas suivants : métrique euclidienne et métrique basée sur l’ACP. Dans ces figures, le vecteur violet indique la direction principale tandis que le vecteur noir indique la direction de la plus forte pente (qui est aussi la direction d’usinage). À noter que sur ces images, le sens des vecteurs n’a pas d’importance, seul leurs direction compte (les orientations  $\pm\pi$  ont la même signification).

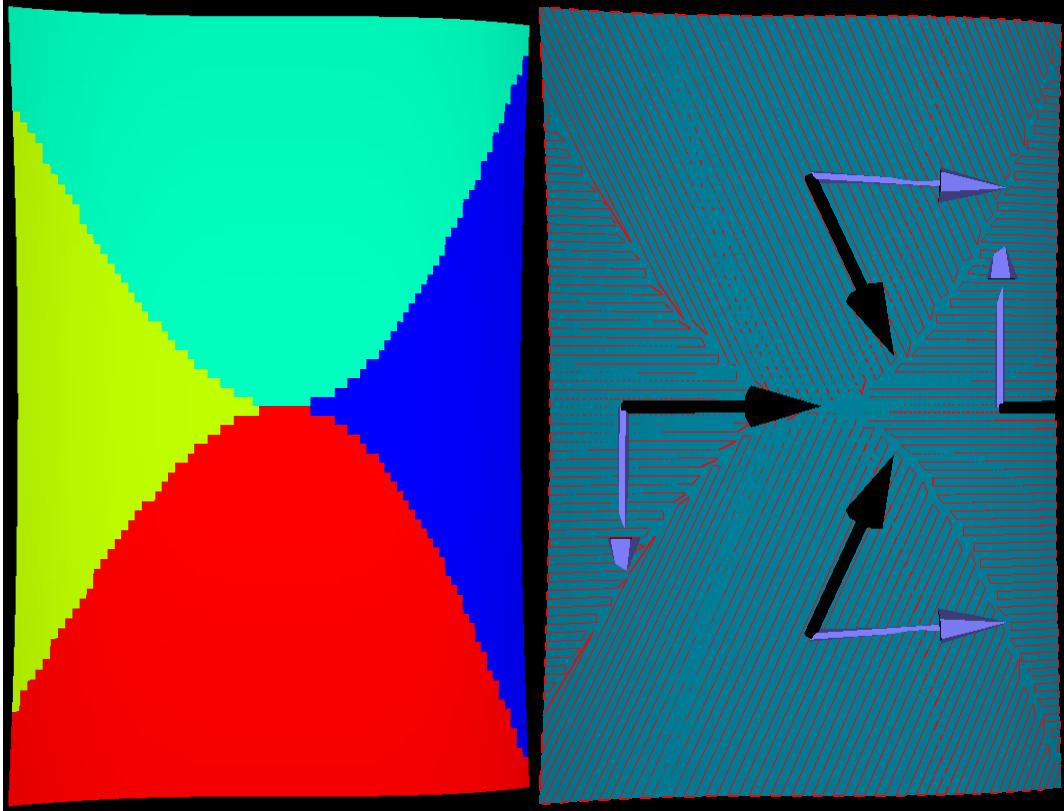


Figure 3.6 – Partitionnement et usinage de la surface de Choi avec l’algorithme *K-means* et la métrique euclidienne

Afin de mettre encore plus en évidence cette propriété, l’angle représentant la différence entre les deux directions est mesuré pour chaque zone et comparé à son homologue à l’aide de la métrique euclidienne. Les résultats du tableau 3.3 montrent que la différence est effectivement plus faible lorsque la métrique basée sur l’ACP est utilisée.

TABLE 3.3 – Angle entre la direction principale et la direction de plus forte pente sur chaque zone de la surface de Choi en utilisant les métriques euclidienne et basée sur l’ACP

zone	1	2	3	4
métrique euclidienne	62.0°	90.3°	62.3°	89.9°
métrique basée sur l’ACP	60.5°	73.1°	18.5°	81.3°

En outre, la métrique basée sur l’ACP tend à produire des zones plus allongées, c’est à dire des zones présentant des rapports longueur/largeur plus importants. Ce dernier point peut être vérifié en

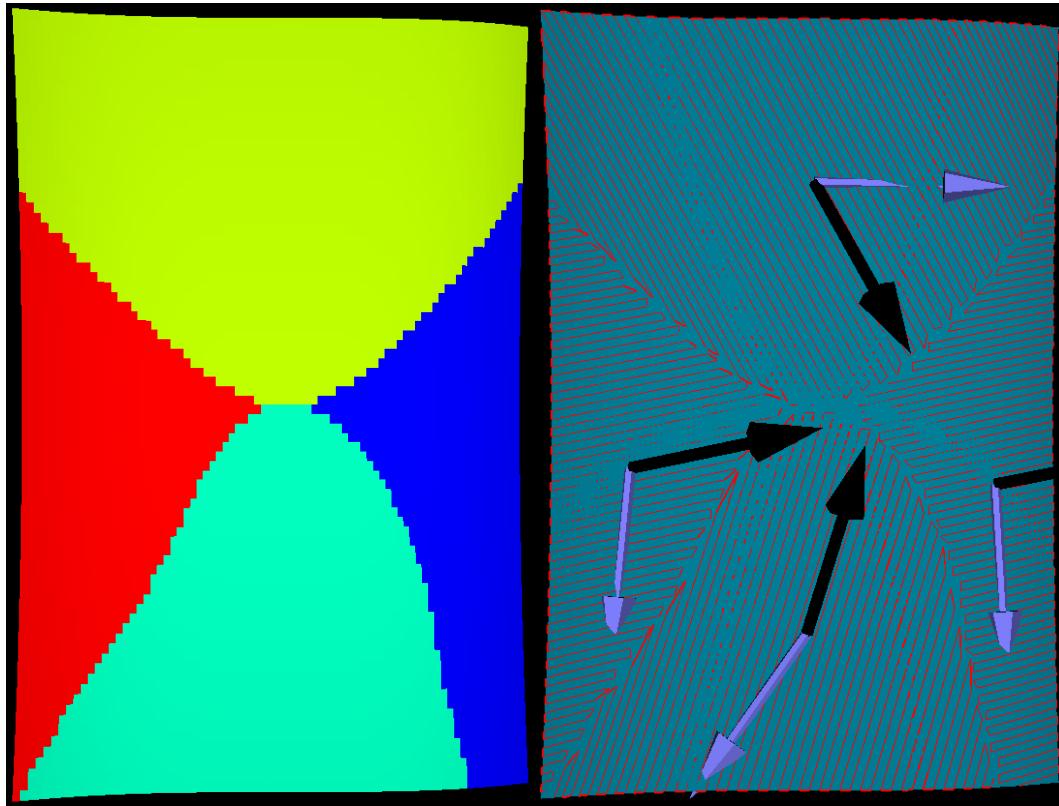


Figure 3.7 – Partitionnement et usinage de la surface de Choi avec l'algorithme *K-means* et la métrique basée sur l'ACP

comparant les rapports des valeurs propres des zones, sur surface de Choi, pour la métrique euclidienne et la métrique basée sur l'ACP. Le tableau 3.4 montre qu'en effet les rapports de valeurs propres augmentent globalement lorsqu'on utilise la métrique basée sur l'ACP.

TABLE 3.4 – Comparison du rapport des valeurs propres  $\frac{\lambda_1}{\lambda_H}$  pour chaque zone de la surface de Choi en utilisant les métriques euclidienne et basée sur l'ACP

zone	1	2	3	4
métrique euclidienne	1.44	4.49	1.45	8.75
métrique basée sur l'ACP	1.80	5.00	1.38	10.09

L'usinage dans la direction principale donne des temps plus courts pour la surface de Rubio, tandis que c'est la direction de la plus forte pente qui donne de meilleurs temps d'usinage pour la surface de Choi. Les résultats de différents algorithmes de clustering avec différentes métriques confirment assez bien cette affirmation. Une tendance inverse est observée pour la longueur du chemin d'outil, elle diminue lorsque le temps d'usinage augmente et vice-versa. Ce comportement est cohérent avec le modèle cinématique de la machine qui ne considère pas la vitesse d'avance constante et pénalise le temps d'usinage des courtes distances. Comme la direction principale est, en général, différente de la direction de la plus forte pente, cette dernière entraîne un nombre plus élevé de trajectoires et donc de courtes distances pour passer d'une trajectoire à l'autre, ce qui conduit à une augmentation du temps d'usinage.

Cette observation est confirmée par les résultats obtenus sur la surface de Rubio. Cependant, cet effet est contrebalancé par le fait que la direction de la plus forte pente permet des pas transversaux plus élevés (surtout pour les petits angles de pente), que la direction principale lorsqu'une fraise torique est utilisée. Ainsi, le temps d'usinage peut être plus long pour la direction principale si l'angle de pente est suffisamment petit et/ou si la direction principale est suffisamment proche de la direction de plus forte pente. Ceci est bien illustré par les résultats obtenus sur la surface de Choi.

En effet, le partitionnement de la surface de Rubio présente des zones avec des rapports de valeurs propres élevés  $\frac{\lambda_1}{\lambda_2}$  comme le montre le tableau 3.5 par rapport à leurs homologues de la première surface. D'autre part, les angles de pente moyens sont plus petits pour la surface de Choi, ce qui permet un gain, à la fois en temps et en distance d'usinage, lors de l'utilisation de la fraise torique dans la direction de la plus forte pente.

TABLE 3.5 – Comparaison des rapports des valeurs propres  $\frac{\lambda_1}{\lambda_2}$  des zones obtenues après partitionnement des surfaces tests à l'aide de l'algorithme *K-means* et de la métrique euclidienne

zone	1	2	3	4
surface de Choi	1.44	4.49	1.45	8.75
surface de Rubio	26.25	60.39	26.10	

Pour la surface de Choi, l'algorithme de classification hiérarchique génère plus de zones que les algorithmes RPCL et *K-means*. Par conséquent, les performances de la fraise torique dans la direction de plus forte pente augmentent, car la direction de la pente est plus homogène dans chaque zone ; et la longueur d'usinage est la plupart du temps plus courte que pour les autres algorithmes et métriques. Cependant, le temps d'usinage augmente car le nombre élevé de zones conduit à passer beaucoup de temps à se déplacer d'une zone à l'autre. Quant à la surface de Rubio, l'algorithme de classification hiérarchique aboutit à une partition contenant 3 zones, similaires à celles obtenues par *K-means* et RPCL, donc le temps et la distance d'usinage sont presque les mêmes. Cependant, l'algorithme HAC nécessite généralement un temps de calcul largement supérieur aux deux autres.

## 3.9 Discussion

Dans cette section, les avantages et les limites des algorithmes et métriques étudiés sont discutés. De plus, nos résultats sont comparés à ceux obtenus avec les approches proposées par d'autres auteurs sur les mêmes surfaces de test.

### 3.9.1 Avantages et limites

L'algorithme *K-means* permet généralement d'obtenir des partitions avec des frontières lisses. Le nombre de *clusters* est fixe et spécifié par l'utilisateur ; il peut augmenter après la convergence en raison

de l’existence de *clusters* non-connexes. Généralement, la distance euclidienne est utilisée comme métrique, mais une métrique définie par l’utilisateur peut également être utilisée. Il est simple de définir un critère d’arrêt (point fixe sans changement) pour l’algorithme *K-means* et la convergence est assurée pour toute métrique euclidienne dérivant d’un produit scalaire. En effet, l’algorithme *K-means* peut diverger pour des métriques non-euclidiennes telles que la métrique basée sur l’ACP, et il est important pour la convergence que les centroïdes des *clusters* soient exactement les points moyens (et non le centre de la maille la plus proche de la moyenne).

En raison de son caractère aléatoire, l’algorithme RPCL permet d’explorer efficacement différentes solutions. Cependant, il nécessite plus de réglages de paramètres ; ces paramètres contrôlent la vitesse de convergence de l’algorithme. Une limite supérieure du nombre de *clusters* doit être fournie. Par rapport à l’algorithme *K-means*, des critères d’arrêt supplémentaires doivent être définis, sinon cela peut prendre beaucoup de temps jusqu’à ce que la convergence (partition fixe, sans changements) soit atteinte. La version de base de l’algorithme RPCL ne permet pas l’utilisation de métriques qui contiennent des informations globales sur les *clusters* (comme la métrique basée sur l’ACP). Ainsi, une étape d’affectation est ajoutée au RPCL, permettant l’utilisation de telles métriques. Le temps de calcul a légèrement augmenté, mais reste inférieur à celui du *K-means*.

Lorsque l’algorithme de classification hiérarchique ascendante est utilisé, aucune connaissance préalable du nombre de *clusters* n’est nécessaire. En fait, le nombre de *clusters* diminue de 1 après chaque itération et le nombre de *clusters* est trouvé en fonction des critères d’arrêt, qui sont définis par l’utilisateur. Les frontières entre les zones obtenues sont moins lisses par rapport au *K-means* et RPCL, et le nombre de zones a tendance à augmenter lorsque le maillage devient plus fin. En fait, en raison de ses mauvais résultats en termes de partitions, l’algorithme HAC n’est pas vraiment adapté au processus de clustering final ; mais, pour une surface donnée, il peut fournir une indication sur le nombre de *clusters* à utiliser dans un autre algorithme de classification, surtout si l’utilisateur a besoin d’utiliser des métriques personnalisées. En effet, les métriques utilisées pour *K-means* et RPCL ne peuvent pas être utilisées pour l’algorithme HAC, puisque la distance est évaluée entre les *clusters* et non pas entre les points de données. Le temps de calcul associé à la classification hiérarchique augmente avec la taille du maillage plus rapidement que RPCL et *K-means*. Pour la taille de maille utilisée dans cette étude ( $80 \times 80$ ), le temps de calcul associé au HAC était entre 1 et 2 minutes alors que *K-means* et RPCL prenaient moins de 3 secondes.

Plusieurs métriques ont été proposées dans cette étude, la plus basique étant la distance euclidienne dans l’espace des caractéristiques (construit à partir des paramètres de proximité  $(u, v)$  et des paramètres liés à l’usinage  $(s, \theta)$ ). Cette métrique est étendue en utilisant l’ACP pour prendre en compte l’allongement des zones en utilisant l’information globale des matrices de covariance. Ceci est fait dans le but d’adapter le clustering à la réponse cinématique de la machine, et ainsi réduire le temps d’usinage. D’autres métriques largement utilisées en statistique ont été testées : la distance de Mahalanobis qui est utile pour traiter les données corrélées en statistiques et la distance basée sur l’écart-type qui est

équivalente à la distance euclidienne où chaque terme est normalisé par son écart-type.

Le temps et la longueur d’usinage dépendent du choix de la métrique et, pour une surface donnée, le choix n’est pas évident. Pourtant, dans la plupart des cas, la métrique basée sur l’ACP fournit de meilleurs résultats que les autres métriques.

### 3.9.2 Comparaison avec d’autres travaux

Dans cette section, les résultats obtenus avec notre approche (en particulier la métrique basée sur l’ACP) sont comparés aux résultats d’autres articles sur les mêmes surfaces test. Pour la validité de la comparaison, nous nous assurons que les différents paramètres sont les mêmes. Les études précédentes citées ci-dessous ne fournissent pas le temps d’usinage ; dans ce cas, seule la longueur du parcours de l’outil peut être comparée. D’ailleurs, ces études sont les seules à fournir des données suffisantes pour rendre les comparaisons possibles.

La surface test de Choi a été présentée dans [21] où la surface a été usinée à l’aide d’une fraise sphérique de rayon  $R = 3,175$  mm (0.125 in) en respectant une contrainte de hauteur de crête de  $s_h = 0,254$  mm (0.01 in). La longueur d’usinage trouvée dans [21] est 1712 mm. En utilisant la même fraise sphérique, les résultats fournis par notre approche sont très similaires, tant en termes d’aspect de la partition que de longueur du parcours d’outil. Mais en utilisant une fraise à bout sphérique, il est très difficile de décider entre les méthodes de partition, car le rayon effectif est le même quelle que soit la direction d’usinage. Si deux méthodes conduisent au même nombre de *clusters*, la longueur du chemin d’outil sera sensiblement la même, surtout pour les surfaces peu inclinées, comme celle de Choi. Pour aller un peu plus loin, une autre simulation, utilisant une fraise torique avec un rayon de tore  $r = 1$  mm a été réalisée, tous les autres paramètres (notamment le rayon extérieur de la fraise) restant les mêmes. Cette nouvelle simulation conduit à une longueur de 1472 mm, soit un gain de 14% environ.

La surface de Rubio apparaît dans [26] et [25]. Dans le premier article cité, un algorithme de Clarke et Wright (entre autres) est utilisé pour le partitionnement, tandis que dans le second, une approche de type AMAS (Adaptive Multi-Agent Systems) est adoptée. Dans les deux articles, la stratégie du plan parallèle est appliquée. Seuls les meilleurs résultats fournis par ces deux articles sont inclus dans la présente comparaison.

Dans l’article [26], une fraise torique  $R = 5$  mm,  $r = 2$  mm est utilisée et la hauteur maximale de crête est fixée à 0,01 mm. Le meilleur résultat, en termes de longueur de parcours d’outil, fourni par cette étude est égal à 4988 mm, tandis que le parcours d’outil le plus court que nous avons trouvé, dans des conditions d’usinage identiques, est égal à 4562 mm. Le temps de calcul associé aux approches de [26] varie entre 60 s et 80 s alors que notre approche n’a pris que 3 s.

Dans le cas de [25], une fraise torique de  $R = 3.175$  mm,  $r = 1$  mm est utilisée et la hauteur maximale de crête est fixée à 0.254 mm. Avec ce contexte d’usinage, la longueur du parcours d’outil est de 1183 mm. Cependant, la longueur du chemin d’outil a été calculée sans tenir compte des connexions entre les trajectoires adjacentes. Nous trouvons une longueur de parcours d’outil similaire (1180 mm)

avec la même partition, en négligeant la longueur des connexions. Le temps de calcul correspondant à notre approche est inférieur à 2 s alors que l’approche AMAS a pris 160 s.

Par ailleurs, la surface de Rubio est également apparue dans une étude plus récente [73] où une approche de partitionnement basée sur le concept d’intervalles d’usinage efficaces est proposée. Cette approche conduit à une partition en trois zones et à une longueur totale de parcours d’outil égale à 4614 mm. L’approche proposée dans notre étude conduit à des longueurs de parcours d’outil similaires (voir tableau 3.2). La surface a été usinée à l’aide d’un outil torique avec les mêmes rayons que ceux de notre étude. Deux vitesses d’avance ont été utilisées : une faible vitesse d’avance de 1 m/min (pour les matériaux durs et/ou les outils de grande taille) et une vitesse d’avance élevée égale à 10 m/min (pour les alliages légers et/ou les petits outils). La faible avance conduit à un temps d’usinage égal à 311 s et la forte avance à 81 s, tandis que notre approche aboutit à 295 s pour la faible avance et 72 s pour la forte avance.

L’algorithme *fuzzy C-means* est utilisé dans [63], ce qui donne lieu à de grandes zones connexes, chaque cluster (zone) contient des points avec une plus grande variance dans les valeurs de  $s$  et  $\theta$ , par rapport au partitionnement net fourni par les algorithmes de clustering durs tels que *K-means*. On s’attend à ce que cela génère des plans parallèles avec des valeurs de *sod* plus petites dans le cas du *fuzzy C-means*, ce qui augmente le temps d’usinage. Cependant, le nombre de zones connexes est plus faible, ce qui peut être avantageux. Le cas idéal serait d’obtenir un faible nombre de zones connexes et une faible variance intra-cluster, afin de maximiser l’efficacité de l’usinage. Cependant, cela peut ne pas être possible pour les surfaces gauches complexes. Il faut donc trouver un compromis entre une variance intra-cluster élevée et un faible nombre de zones connexes. Une façon de contrôler le caractère flou du C-means est d’ajuster le paramètre  $m$  dit *fuzzifier*, qui intervient dans le calcul des centroïdes des *clusters* et de la somme pondérée des distances au carré au sein du cluster. En outre, un vecteur de caractéristiques à cinq paramètres  $(u, v, \mathbf{n})$  est utilisé dans cette étude. Comme indiqué précédemment (voir la section 3.3.2), un vecteur de caractéristiques à quatre composantes  $(u, v, s, \theta)$  est plus adapté à l’outil torique même si les mêmes informations sont contenues. Notre approche est testée sur la même surface, avec la même fraise torique de rayons  $R = 6.7$  mm,  $r = 0.7$  mm, une contrainte de hauteur de crête  $sh = 0.0254$  mm, une vitesse d’avance nominale de  $V = 2$  mm/min et un nombre de classes  $K = 4$ . En [63], l’usinage de cette surface en 3+2 axes donne un temps d’usinage de 634 s, alors que notre approche (utilisant l’algorithme *K-means*) en 3 axes donne 611 s en utilisant la métrique ACP. Le temps d’usinage peut être encore réduit à 601 s dans le cas d’un usinage 3+2-axes.

L’approche de [63] a également été testée sur les deux surfaces présentées dans notre étude pour un usinage 3 axes. Pour ce faire, l’algorithme *fuzzy C-means* et la métrique euclidienne basée sur le vecteur  $(u, v, \mathbf{n})$  ont été mis en œuvre. Pour la surface de Choi : le temps d’usinage était de 139,3 s en utilisant une fraise torique dans la direction de plus forte pente, et de 109,5 s en utilisant une fraise sphérique dans la direction principale. La surface de Rubio, quant à elle, a donné lieu à un temps d’usinage de 132,9 s en utilisant une fraise torique et de 93,6 s en utilisant une fraise sphérique. Ces résultats sont très

intéressants, car non seulement ils confirment que les paramètres  $s, \theta$  sont plus appropriés que  $n_x, n_y, n_z$ , mais ils montrent également le potentiel de l’usinage dans la direction principale. En effet, la fraise sphérique a également été testée dans la direction de la pente la plus forte et cela conduit à une 119,4 s pour la surface de Choi et une 101,2 s pour la surface de Rubio. Un gain de 8,3% et 7,5% a été obtenu pour chaque surface uniquement en usinant le long de la direction principale.

L’approche décrite dans [46] a également été testée sur nos deux surfaces de test en usinage 3 axes. Pour la surface de Choi : le temps d’usinage était de 137,7 s en utilisant une fraise torique dans la direction de la pente la plus raide, et de 101,7 s en utilisant une fraise sphérique dans la direction principale. Alors que le test de la surface de Rubio a donné lieu à un temps d’usinage de 124,1 s en utilisant une fraise torique et 88,6 s en utilisant une fraise sphérique. Ces résultats montrent la même tendance que la comparaison précédente, et une fois de plus, la métrique basée sur l’ACP conduit à des temps d’usinage plus courts : 96,4 s pour la surface de Choi et 83,3 s pour la surface de Rubio.

## 3.10 Conclusions et perspectives

Ce chapitre présente une approche originale du partitionnement de surfaces gauches pour l’usinage. Trois algorithmes de clustering ont été testés et leurs avantages et inconvénients ont été mis en évidence. En outre, une nouvelle métrique, basée sur l’analyse en composantes principales, est introduite. Le principal intérêt de cette métrique est qu’elle fournit une vue globale des zones sur lesquelles les algorithmes travaillent, alors que les méthodes existantes ne se basent que sur des informations locales. Grâce à ce point de vue plus global, dans la plupart des cas testés, la métrique basée sur l’ACP fournit de meilleurs résultats que les autres métriques. En outre, afin de réduire le temps de calcul des simulations de clustering, une mise à jour efficace de la matrice de covariance, utilisant une formule itérative, a été établie. De plus, une nouvelle procédure, permettant une identification très rapide des composantes connexes, a été mise en œuvre.

Ce travail n’est qu’une première approche pour améliorer le temps d’usinage dans le cas de l’usinage par zone en utilisant des algorithmes de clustering. À court terme, la métrique basée sur l’ACP devrait être améliorée. Sa capacité fondamentale à inclure une vue globale de la zone sur laquelle les algorithmes de clustering travaillent est un atout essentiel pour définir des métriques plus efficaces. Par ailleurs, le choix du point de départ doit être étudié pour évaluer son influence sur le résultat final de l’usinage. De même, l’influence de la finesse du maillage doit également être analysée. Le chapitre qui suit est lié à l’optimisation des paramètres d’usinage afin d’obtenir une amélioration supplémentaire du temps d’usinage. Ces futures pistes de recherche peuvent également inclure l’optimisation de l’étape de clustering, même si le lien entre clustering et temps d’usinage n’est pas facile à modéliser. Par ailleurs, l’approche présentée dans ce chapitre est remarquablement rapide, ce qui est particulièrement intéressant pour l’optimisation. Réduire encore plus le temps de simulation en utilisant l’optimisation pourrait également faire l’objet de nouvelles recherches.



# Chapitre 4

## Optimisation de l'usinage de surfaces gauches

Dans ce chapitre, l'usinage de surfaces gauches est formulé comme un problème d'optimisation où l'on cherche à minimiser le temps d'usinage sous contrainte de hauteur de crête. Après une revue de littérature des travaux abordant le problème sous l'angle de l'optimisation mathématique, une formulation originale intégrant les étapes de partitionnement et d'usinage est proposée. Cette formulation implique des contraintes et une fonction-objectif de type boîte noire dont l'évaluation peut être coûteuse.

Pour réduire ce coût, plusieurs fonctions de substitution (ou *surrogates*) sont développées. Une approche heuristique combinant ACP et NOMAD est également proposée. Ces deux approches sont appliquées pour usiner des surfaces tests et comparées à l'état de l'art.

### 4.1 Introduction

De nombreux auteurs ont modélisé différents aspects du problème d'usinage sous forme d'un problème d'optimisation. Ainsi, [3] a utilisé un algorithme génétique pour trouver une trajectoire maîtresse (courbe initiale) de la forme  $v = A_0 + A_1u + A_2u^2 + A_3u^3$ , qui minimise la longueur de trajectoire en utilisant la stratégie isocrête en usinage 3 axes, les coefficients  $A_0, \dots, A_3$  étant les variables d'optimisation. Dans [44], le problème de génération de trajectoires optimales a été modélisé sous la forme d'un problème de voyageur de commerce et d'arbre couvrant minimal, avec pour objectif de minimiser les forces de coupe. Aussi, [16] s'appuie sur le problème du voyageur de commerce avec des contraintes de précédence pour minimiser le temps non-productif ("airtime") en connectant de manière "optimale" différents segments de la trajectoire d'outil. La cinématique des machines-outils à commande numérique est également un domaine d'intérêt : [27] introduit le jerk (dérivée de l'accélération) autorisé comme contrainte dans un modèle d'optimisation de la vitesse d'avance sous des contraintes de vitesse et d'accélération pour générer des trajectoires de temps "minimum".

L'optimisation de l'usinage 5 axes a fait également l'objet de recherches approfondies. Trois approches différentes sont présentées dans [50] : La première définit la trajectoire d'outil en se basant sur une approche de génération de grille curviligne, la deuxième s'appuie sur des "*space filling curves*",

tandis que la troisième crée des *clusters* basés sur un champ vectoriel de directions d’usinage localement optimales. Des approximations au second ordre de la surface outil, de la surface enveloppe et de la surface de la pièce sont utilisées par [32] pour dériver un problème d’optimisation analytique du positionnement local de l’outil en usinage 5 axes. Dans la même optique, [38] propose l’optimisation et le lissage du positionnement de l’outil pour l’usinage 5 axes. Une méthode de recherche de limite (“*boundary search*”) est d’abord utilisée pour trouver un ensemble d’orientations d’outil réalisables (sans interférences locales ou globales) dans l’espace de configuration, ensuite les orientations d’outil sont optimisées localement pour minimiser la hauteur de crête, avant de lisser la séquence d’orientations afin d’éviter les changements brusques. D’autres travaux sur l’optimisation de l’usinage 5 axes sont disponibles dans [41, 31, 5].

Trouver une partition de surface pour l’usinage 3 axes a fait l’objet de plusieurs recherches, par exemple : Des méthodes heuristiques inspirées du problème de tournées de véhicules (recherche tabou, heuristique de Clarke et Wright, ...) sont utilisées par [26]. Dans le même but, une approche basée sur les systèmes multi-agents adaptatifs est présentée dans [25]. Une autre approche basée sur le concept d’intervalles d’efficacité (ensemble de directions où un outil torique est meilleur qu’un outil sphérique) est proposée par [74, 73]. L’objectif de ces approches est de trouver des zones présentant une faible variation de la direction de plus forte pente afin d’augmenter le rayon effectif, ce qui permet d’obtenir des grands pas transversaux, et par la suite de minimiser la longueur du parcours de l’outil.

Comme on peut le voir, la plupart des algorithmes d’optimisation utilisés sont des heuristiques ou des procédures qui impliquent peu d’itérations et donc *a priori* n’explore pas efficacement l’espace de recherche. La raison principale est le coût élevé des simulations d’usinage, en particulier le fait que l’évaluation du temps d’usinage (ou d’autres fonctions impliquées dans l’optimisation) se fait inévitablement de manière numérique, puisque toutes les stratégies d’usinage sont numériques et itératives, sans alternative analytique pour les surfaces gauches. Cela correspond donc à un problème d’optimisation de boîte noire qui nécessite *a priori* de nombreuses évaluations compte tenu de la complexité des surfaces gauches.

## 4.2 Formulation du problème d’optimisation

L’approche d’optimisation présentée ici est basée sur deux étapes principales : le partitionnement et l’usinage. Le partitionnement des surfaces en zones est effectué à l’aide d’algorithmes de classification non supervisés appliqués sur un ensemble de points construits à partir d’un maillage isoparamétrique uniforme (figure 3.1). La procédure est la même que celle décrite au chapitre précédent.

Les étapes de partitionnement et d’usinage peuvent toutes les deux comporter des variables d’optimisation. Certains algorithmes de *clustering*, tels que K-means, nécessitent que l’utilisateur spécifie le nombre de *clusters*, désigné par  $K \geq 1$ , qui pourrait ainsi être considéré comme une première variable d’optimisation. Une limite supérieure  $K_{max}$  de  $K$  peut être définie car le temps d’usinage augmente continuellement après avoir dépassé un certain nombre de clusters, en raison de la perte de temps lorsque

l'outil termine une zone et passe à la suivante. Il est à noter que cette pénalité doit être prise en compte, sinon le problème devient trivial et la solution est obtenue en considérant chaque maille comme un cluster ( $K = n$ ,  $n$  étant le nombre total de *sample-points*). En outre, les algorithmes de *clustering* nécessitent en général un *feature-vector* et une mesure de dissimilarité, appelée communément la métrique. Dans ce qui suit, le *feature-vector* est  $(u, v, s, \theta)$  et la dissimilarité entre deux points  $S_i = (u_i, v_i, s_i, \theta_i)$  et  $S_j = (u_j, v_j, s_j, \theta_j)$  est mesurée par la pseudo-métrique suivante :

$$d^2(S_i, S_j) = \omega_1 (u_i - u_j)^2 + \omega_2 (v_i - v_j)^2 + \omega_3 (s_i - s_j)^2 + \omega_4 (\theta_i - \theta_j)^2,$$

où les paramètres de pondération  $0 \leq \omega_1, \omega_2, \omega_3, \omega_4 \leq 1$  sont à déterminer. Ces choix ont été présentés, évalués et discutés au chapitre précédent.

Une fois la partition obtenue, l'étape d'usinage peut commencer et chaque zone est traitée indépendamment. Dans ce qui suit, la stratégie des plans parallèles est choisie. La distance entre deux plans adjacents (appelée pas transversal, notée *sod*) est définie comme la distance maximale telle que la contrainte de hauteur de crête soit respectée pour l'ensemble de la trajectoire. Ce choix facilite la gestion de la contrainte de qualité. En effet, à un plan donné, la position du plan adjacent est facile à déterminer : il suffit de considérer le point le plus défavorable, en termes de hauteur de crête, de la trajectoire définie par le plan donné. Cela suffit à garantir que la contrainte de qualité sera satisfaite pour l'ensemble de la trajectoire. Du point de vue de l'optimisation, cette stratégie est très utile car les contraintes sont ainsi satisfaites par construction. En utilisant une méthode rapide de calcul de la hauteur de crête, comme dans [62], le processus de planification de la trajectoire décrit ci-dessus devient plus rapide, ce qui permet plus d'évaluations de la boîte noire. Ainsi, dans le cas de l'usinage 3-axes, pour chaque cluster  $k$ , la planification de la trajectoire dépend uniquement de la direction de l'usinage, et l'angle définissant cette direction par rapport à l'axe **X**, noté  $\Theta_k$ , est considéré comme une variable d'optimisation.

En outre, l'usinage 3+2 axes peut être envisagé, auquel cas deux angles d'orientation doivent être déterminés pour chaque zone. Par conséquent, pour chaque cluster  $k$ , on définit l'angle entre l'axe de l'outil et l'axe **Z**, désigné  $\Phi_k$ , et l'angle entre la projection de l'axe de l'outil sur le plan **XY** et l'axe **X**, désigné  $\Psi_k$ . La figure 4.1 illustre les angles d'orientation. Les angles d'orientation sont censés varier au cours du processus d'optimisation. Par conséquent, l'accessibilité de l'outil à tous les points de la surface à usiner doit être vérifiée chaque fois que les orientations des zones changent. En d'autres termes, les angles  $\Phi_k$  et  $\Psi_k$  doivent satisfaire la contrainte d'accessibilité. Un point donné de la surface, de coordonnées paramétriques  $(u, v)$ , est considéré comme accessible par l'outil si l'angle entre le vecteur normal  $\mathbf{n}(u, v)$  et l'axe **Z** est compris entre  $-\frac{\pi}{2}$  et  $\frac{\pi}{2}$ . Une nouvelle contrainte (équation 4.1) est ainsi obtenue. En pratique, cette contrainte n'est évaluée que sur les  $N_k$  *sample-points* du *data-set* appartenant à la  $k$ -ième zone.

$$|\arccos(\mathbf{n}_{\Phi_k, \Psi_k}(u_i, v_i) \cdot \mathbf{Z})| \leq \frac{\pi}{2}, \quad i = 1, \dots, N_k, \quad k = 1, \dots, K \quad (4.1)$$

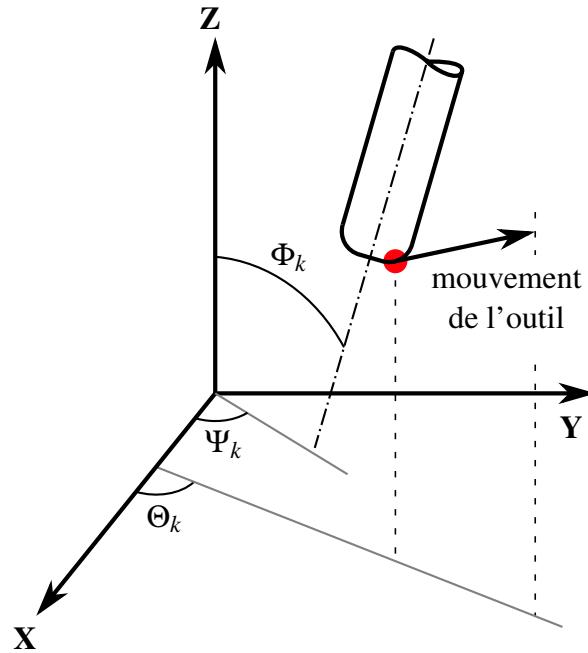


Figure 4.1 – Angles d'orientation utilisés pour l'optimisation en 3+2-axes

En résumé, l'approche proposée minimise le temps d'usinage en résolvant un problème d'optimisation avec contraintes où la fonction-objectif (temps d'usinage) est le résultat d'une simulation numérique en deux étapes (boîte noire), et la contrainte d'accessibilité est aussi évaluée numériquement. En outre, le nombre de variables et de contraintes dépend directement de la valeur de  $K$  (qui est donc une variable structurelle) et indirectement des  $\omega_i$  (en raison de la recherche de composants connexes, voir le chapitre précédent, section 3.7.4).

La fonction-objectif dépend de variables d'optimisation qui peuvent être divisées en deux classes : Les variables de *clustering*, telles que  $K \in \mathbb{N}^*$ ,  $\omega \in [0, 1]^4$ , et les variables d'usinage, qui sont  $\Theta \in [0, \pi]^K$ ,  $\Phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]^K$  et  $\Psi \in [0, \pi]^K$ . La figure 4.2 présente un organigramme récapitulatif du modèle. Notez que le problème combine des variables entières et continues. Il s'agit donc d'un problème d'optimisation à variables mixtes.

Une première formulation du problème d'optimisation est donnée par l'équation 4.2. Si  $K_{max}$  n'est pas trop élevé, la procédure la plus efficace consiste à fixer  $K$  et à résoudre le problème d'optimisation

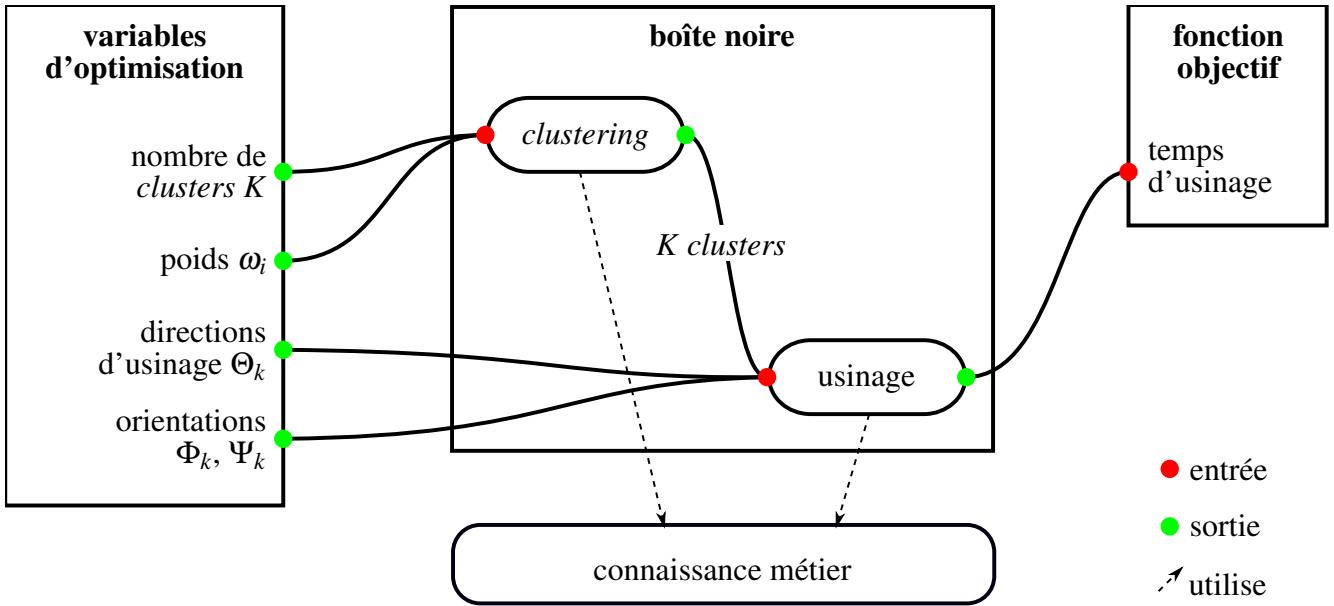


Figure 4.2 – Organigramme du modèle d'optimisation en deux étapes

pour toutes les valeurs de  $K$  et à retenir la solution ayant le plus petit temps d'usinage.

$$\begin{aligned}
 \min_{K, \omega, \Theta, \Phi, \Psi} \quad & \text{machining\_time}(K, \omega, \Theta, \Phi, \Psi) \\
 \text{s.c.} \quad & 1 \leq K \leq K_{max} \\
 & 0 \leq \omega_i \leq 1, i = 1, \dots, 4 \\
 & 0 \leq \Theta_k \leq \pi, k = 1, \dots, K \\
 & -\frac{\pi}{2} \leq \Phi_k \leq \frac{\pi}{2}, k = 1, \dots, K \\
 & 0 \leq \Psi_k \leq \pi, k = 1, \dots, K \\
 & \left| \arccos(\mathbf{n}_{\Phi_k, \Psi_k}(u_i, v_i) \cdot \mathbf{Z}) \right| \leq \frac{\pi}{2}, i = 1, \dots, N_k, k = 1, \dots, K,
 \end{aligned} \tag{4.2}$$

Où  $N_k$  est le nombre de maille appartenant à la zone  $k$ , au moment de l'évaluation de la contrainte. Il est à noter que la valeur de  $N_k$  varie durant l'optimisation et qu'elle dépend des  $\omega_i$ .

## 4.3 Fonctions surrogates

Une façon de palier au coût élevé des évaluations de la boîte noire est de développer des fonctions de substitution (*surrogates*). Une *surrogate* est une fonction (boîte noire ou analytique) moins chère à évaluer et qui est utilisée au moins partiellement pour piloter l'optimisation [7]. Il s'agit soit d'un modèle physique simplifié de la fonction de boîte noire, soit d'un modèle d'approximation obtenu en évaluant la fonction de boîte noire en un nombre donné de points puis en interpolant ou en lissant les valeurs. Il existe deux types de *surrogates* : statique et dynamique. Les *surrogates* statiques sont fournis par l'utilisateur et ne sont pas mises à jour pendant l'optimisation. Les *surrogates* dynamiques sont en revanche construites

progressivement à partir des évaluations disponibles en cours d’exécution de l’algorithme d’optimisation. En utilisant NOMAD, une bibliothèque de *surrogates* dynamiques est fournie. Pour plus de détails, [15] présente une étude sur l’optimisation des *surrogates*.

Dans cette section, nous nous concentrons sur les *surrogates* statiques. L’objectif est de construire des *surrogates* les moins coûteuses possible, afin de réduire considérablement le temps de calcul. En fait, une *surrogate* efficace n’a pas besoin d’être une bonne approximation des valeurs exactes de la boîte noire. Elle doit plutôt reproduire aux mieux ses variations et approcher l’emplacement des points optimaux. Cela permet une certaine flexibilité dans la construction des *surrogates*.

Dans notre contexte, le temps d’usinage est le résultat d’une simulation basée sur le calcul numérique des intersections de la surface à usiner et de plans parallèles. Les courbes d’intersection (trajectoires d’usinage) reposent sur un maillage isoparamétrique de la surface. La précision des courbes et le temps nécessaire à leur calcul augmentent avec la finesse du maillage. Par conséquent, la finesse du maillage est un paramètre qui permet de simplifier le modèle physique (évaluations moins coûteuses mais moins précises) et peut être utilisé pour créer des *surrogates*.

Un deuxième paramètre qui pourrait être utilisé pour réduire le temps de simulation est la tolérance de hauteur de crête. En effet, l’augmentation de cette valeur entraîne une augmentation du pas transversal, ce qui réduit le nombre de plans parallèles et, par conséquent, le nombre d’intersections avec la surface de la pièce. Bien sûr, cela réduira les valeurs de la fonction-objectif (temps d’usinage). Néanmoins, il est concevable que les variations et les points optimaux soient préservés, ce qui est examiné plus en détail dans cette section.

Jusqu’à présent, les simplifications du modèle ont porté sur l’étape d’usinage plutôt que sur le *clustering*. Une façon de simplifier la phase de *clustering* est d’augmenter la taille du maillage de *clustering*. Cela permet de réduire le temps de calcul nécessaire au partitionnement puisque l’algorithme K-means (et tout autre algorithme de *clustering*) nécessite moins de temps pour fonctionner avec moins de data-points. Cependant, la précision des zones de la surface (les *clusters*) est alors détériorée ; les limites des zones sont plus approximatives, ce qui affecte les valeurs des temps d’usinage. La réduction du temps de simulation est plus ou moins efficace selon l’algorithme utilisé. Dans notre cas, l’algorithme K-means est assez rapide et un gain énorme n’est donc pas attendu.

Une autre alternative pour simplifier la boîte noire serait d’approximer les zones de la surface à usiner par des plans rectangulaires, de sorte que les temps d’usinage puissent être calculés analytiquement, ce qui peut réduire considérablement le temps de calcul des évaluations de la boîte noire. L’approximation par plans doit être précise, même si certaines zones ne sont pas adaptées à une telle approximation. L’analyse en composantes principales (ACP) [1] est utilisée pour calculer les plans qui s’adaptent le mieux aux zones de surface. Cette procédure est la même que celle de [36]. Le temps d’usinage est ensuite calculé selon l’approche analytique détaillée en annexe E.

À partir de ce qui précède, nous proposons quatre fonctions *surrogates*. La première est obtenue à partir de la fonction boîte noire en augmentant la taille du pas de maillage de 0.5 mm à 5 mm, et la

seconde est obtenue en l'augmentant seulement à 2 mm. L'approximation des zones est modifiée pour les deux *surrogates* afin que les frontières soient mieux approximées, même pour les maillages grossiers. La troisième *surrogate* approxime les zones de la surface avec des plans en utilisant l'ACP, de sorte que les temps d'usinage soient calculés analytiquement. Enfin, la quatrième *surrogate* est obtenue à partir de la première *surrogate* en augmentant aussi la hauteur de crête acceptable de 0.01 mm à 0.1 mm.

Afin d'étudier l'efficacité des quatre *surrogates*, nous traçons leurs variations en fonction de la direction d'usinage sur plusieurs surfaces et nous les comparons à la variation de la boîte noire. Les temps de calcul des différentes *surrogates* et leurs corrélations avec la boîte noire sont comparés.

Pour calculer le coefficient de corrélation, les fonctions sont d'abord échantillonnées de manière uniforme. Soit  $N$  le nombre de points d'échantillonnage, le coefficient de corrélation  $\rho$  entre une fonction  $f$  (la boîte noire par exemple) et une fonction  $g$  (une *surrogate*) est défini comme la covariance de  $f$  et  $g$  divisée par le produit des écarts types de  $f$  et  $g$ . En utilisant les échantillons  $\mathbf{F} = (f_j)_{1 \leq j \leq N}$  et  $\mathbf{G} = (g_j)_{1 \leq j \leq N}$ , la corrélation  $\rho$  est calculée en utilisant l'équation 4.3. Remarquez que  $|\rho| = 1$  lorsque les deux fonctions sont proportionnelles, autrement dit  $\mathbf{F}$  et  $\mathbf{G}$  sont colinéaires :  $\mathbf{F} = \lambda \mathbf{G}$ , avec  $\lambda \neq 0$ , le signe de  $\rho$  est le même que  $\lambda$ . En fait, le coefficient de corrélation mesure la linéarité entre deux fonctions  $f$  et  $g$  ; de plus, il peut être interprété géométriquement comme le cosinus de l'angle (dans l'espace  $\mathbb{R}^N$ ) entre les vecteurs de leurs points d'échantillonnage  $\mathbf{F}$  et  $\mathbf{G}$ .

$$\rho = \frac{\text{cov}(\mathbf{F}, \mathbf{G})}{\sigma_{\mathbf{F}} \sigma_{\mathbf{G}}} \quad (4.3)$$

avec

$$\begin{aligned} \text{cov}(\mathbf{F}, \mathbf{G}) &= \frac{1}{N} \sum_{j=1}^N (f_j - \bar{f})(g_j - \bar{g}) \\ \sigma_{\mathbf{F}} &= \sqrt{\frac{1}{N} \sum_{j=1}^N (f_j - \bar{f})^2} \\ \sigma_{\mathbf{G}} &= \sqrt{\frac{1}{N} \sum_{j=1}^N (g_j - \bar{g})^2} \\ \bar{f} &= \frac{1}{N} \sum_{j=1}^N f_j \text{ et } \bar{g} = \frac{1}{N} \sum_{j=1}^N g_j \end{aligned}$$

Les surfaces à tester sont obtenues en partitionnant les deux surfaces tests illustrées par la figure 4.3 à l'aide de l'algorithme K-means et de la métrique euclidienne pour  $K = 3$ , ce qui donne sept zones au total comme le montre la figure 4.3. Les figures 4.4, respectivement 4.5, montrent l'évolution du temps d'usinage calculé par la fonction boîte noire et les différentes *surrogates* par rapport à la direction d'usinage pour chaque zone de la surface 1, respectivement 2. Les tableaux 4.1 et 4.2 donnent les pourcentages des temps de calcul de chaque *surrogate* par rapport à la fonction boîte noire pour les différentes zones ainsi que la corrélation avec la boîte noire.

Seule la variation du temps d'usinage par rapport à l'angle définissant la direction d'usinage est

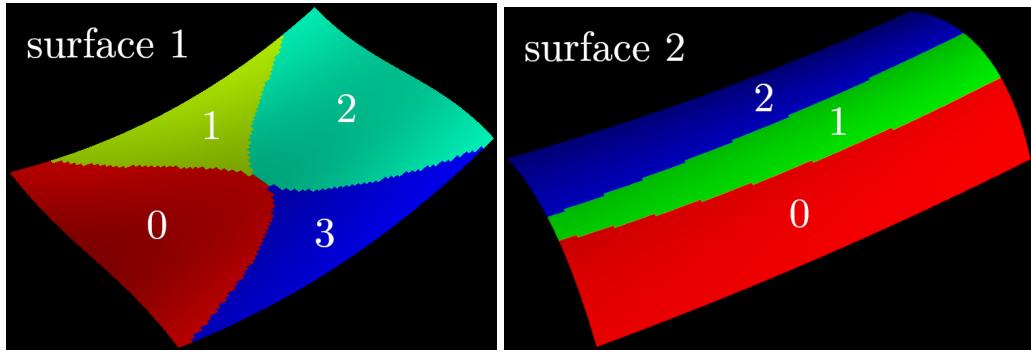


Figure 4.3 – Partitionnement des deux surfaces (la numérotation des zones apparaît en blanc)

 TABLE 4.1 – Ratio de temps de calcul moyen entre les *surrogates* 1 à 4 et la boîte noire pour chaque zone

surface	zone	surrogate 1	surrogate 2	surrogate 3	surrogate 4
1	zone 0	26.2%	62.5%	5.4%	14.9%
	zone 1	25.2%	66.7%	3.1%	17.4%
	zone 2	23.9%	82%	3%	18.8%
	zone 3	18.1%	59.4%	1.6%	14%
2	zone 0	18.8%	84%	3%	8%
	zone 1	19.8%	106%	1.3%	11.3%
	zone 2	13.6%	103%	1.2%	9.4%

considérée. Ceci est justifié par le fait que toutes les autres variables (à part  $\Theta$ ) modifient la zone à usiner, soit en modifiant sa forme (paramètres de *clustering*) soit en changeant son orientation (angles  $\Phi$  et  $\Psi$ ). Seule la variable  $\Theta$  agit sur le temps d’usinage sans modifier la zone. Ainsi, l’étude de la performance des *surrogates* peut se contenter de la variable  $\Theta$ , ce qui permet entre autres de tracer les graphes des figures 4.4 et 4.5.

On remarque que les trois zones de la surface 2 ont des formes plus rectangulaires que celles provenant de la surface 1, ce qui explique la corrélation plus forte dans le premier cas pour la *surrogate 3* qui est basée sur l’approximation planaire résultante de l’ACP. Comme prévu, le temps de calcul de la *surrogate 1* est plus court et la corrélation est plus faible que pour la *surrogate 2*, puisque cette dernière

 TABLE 4.2 – Coefficient de corrélation entre les *surrogates* 1 à 4 et la boîte noire pour chaque zone

surface	zone	surrogate 1	surrogate 2	surrogate 3	surrogate 4
1	zone 0	0.962	0.979	0.854	0.966
	zone 1	0.925	0.915	0.824	0.925
	zone 2	0.973	0.982	0.850	0.980
	zone 3	0.920	0.932	0.650	0.915
2	zone 0	0.813	0.992	0.931	0.820
	zone 1	0.990	0.999	0.936	0.991
	zone 2	0.810	0.981	0.931	0.815

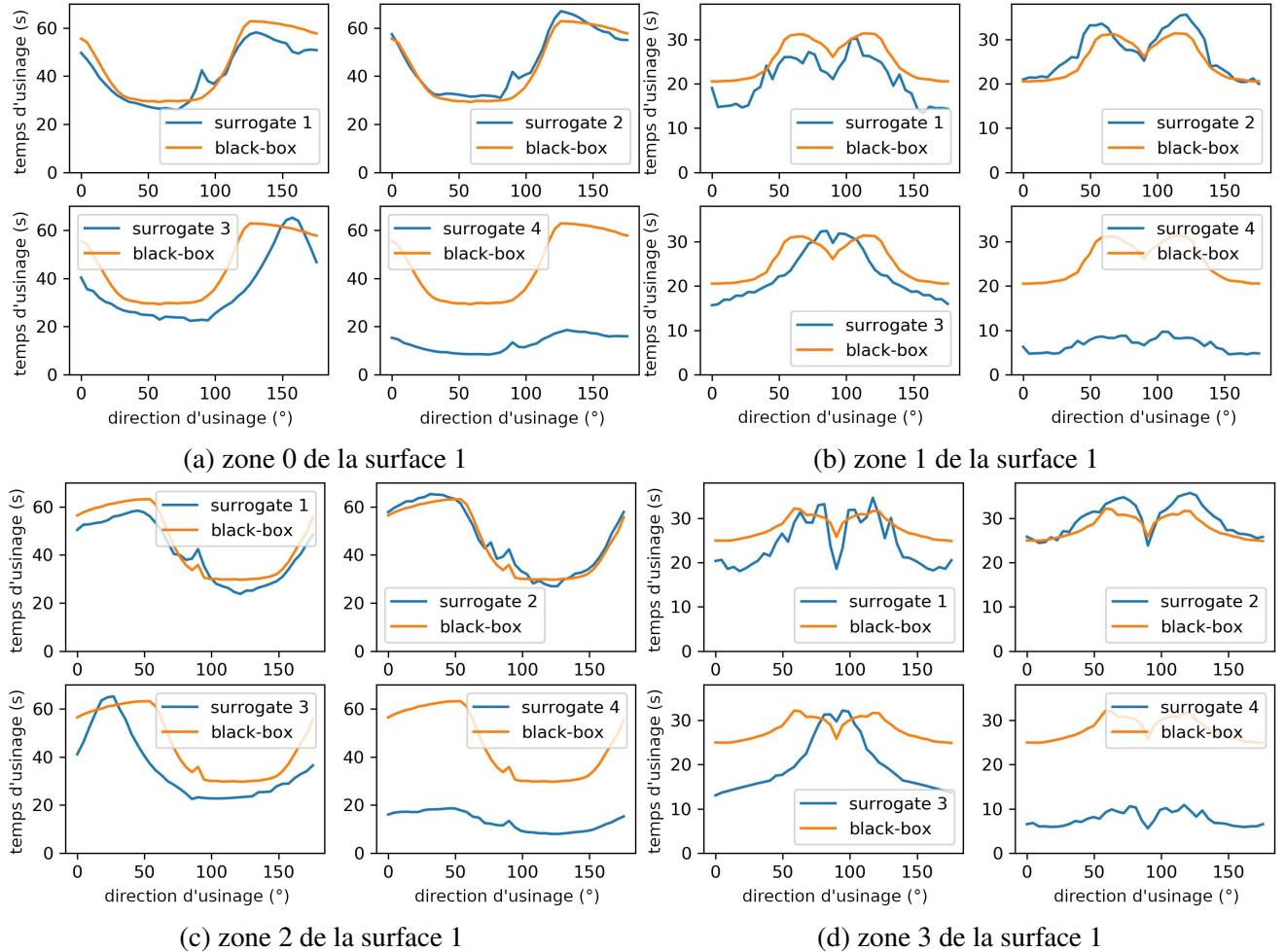


Figure 4.4 – Évolutions des temps d’usinage obtenu par les quatres *surrogates* comparés au temps d’usinage de la boîte noire pour les zones de la surface 1

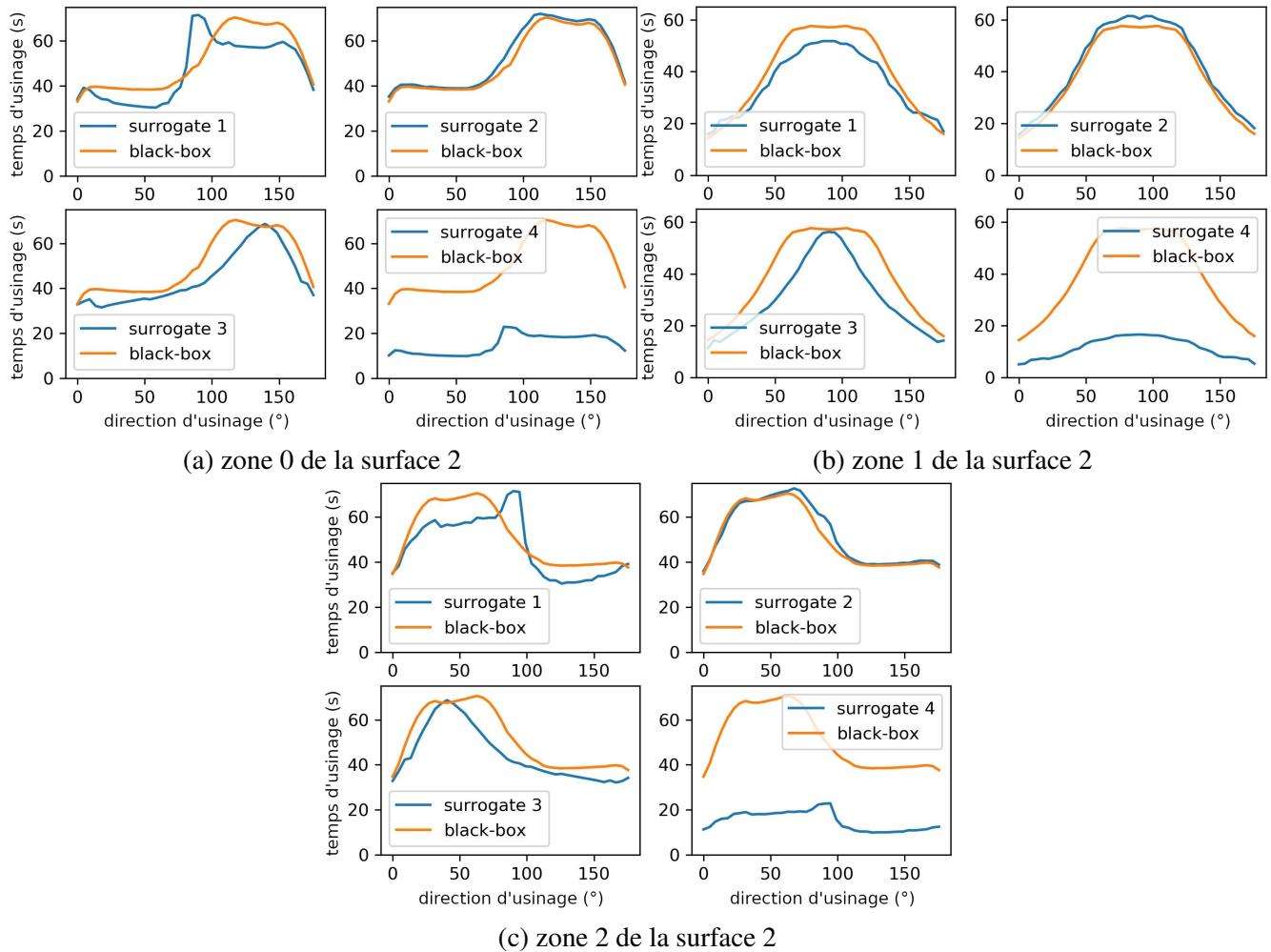


Figure 4.5 – Évolutions des temps d'usinage obtenu par les quatre *surrogates* comparés au temps d'usinage de la boîte noire pour les zones de la surface 2

utilise un maillage plus fin. Le rapport des temps de calcul de la *surrogate* 2 et de la boîte noire dépasse 100% en raison du temps supplémentaire qu'exige le traitement des frontières de zones. On rappelle que ce traitement est nécessaire pour les *surrogates* 1, 2 et 4. Les résultats de la *surrogate* 4 montrent que l'augmentation de la tolérance de la hauteur de crête (multipliée par 10) permet de réduire le temps de calcul (d'environ 2 fois) sans réduire la corrélation : en fait la fonction est juste "tassée" sans changements significatifs des variations. Le temps de calcul de la *surrogate* 3 est beaucoup plus court que celui des autres *surrogates* puisqu'elle est calculée analytiquement.

En résumé, plusieurs façons de créer des fonctions *surrogates* de la boîte noire (temps d'usinage) ont été présentées. Quatre *surrogates* différentes ont été données à titre d'exemple. La comparaison de leur efficacité en termes de temps de calcul et de corrélation avec la boîte noire montre que la *surrogate* 2 est très lente et ne peut pas être utilisée. La première *surrogate* est plus rapide que la seconde mais reste lente par rapport aux *surrogates* 3 et 4. La *surrogate* 3 est la plus rapide et donc très adaptée à l'optimisation de la boîte noire, même si sa corrélation diminue légèrement lorsque la forme des zones est "loin d'être rectangulaire", en raison de l'approximation basée sur l'ACP. Dans la suite, seules les *surrogates* 3 et 4

seront utilisées dans les tests.

## 4.4 Résultats

Dans cette section, nous présentons les résultats du modèle d'optimisation (équation 4.2) dans les deux cas d'usinage 3 axes et 3+2 axes, appliqués aux deux surfaces de test utilisées dans la section 4.3.

La limite supérieure du nombre de clusters est fixée à  $K_{max} = 10$ , le maillage du *clustering* est une grille isoparamétrique uniforme de  $80 \times 80$  pour les évaluations en boîte noire, et il est réduit à  $40 \times 40$  pour les évaluations des *surrogates*. L'algorithme K-means est utilisé pour partitionner les surfaces en utilisant la distance euclidienne comme métrique.

L'usinage est effectué avec un outil torique dont les rayons sont  $R = 5$  mm et  $r = 2$  mm. On considère une stratégie de plans parallèles. La tolérance de hauteur de crête est fixée à  $sh = 0.01$  mm. Les paramètres cinématiques de la machine à commande numérique sont un jerk constant de  $J = 40$  m/s<sup>3</sup>, une accélération maximale  $A_{max} = 6$  m/s<sup>2</sup> et une avance nominale  $V_{max} = 0.083$  m/s (5 m/min).

Le paramètre de type de direction du maillage de l'algorithme MADS est fixé à ORTHO 2N, ce qui signifie que les directions de sondage (*poll directions*) sont déterministes et générées en utilisant la base positive maximale produite à partir d'une base orthogonale de l'espace de recherche, et son opposé. Pendant l'étape de recherche, un modèle quadratique est utilisé par défaut comme *surrogate* dynamique, le modèle est alimenté par les évaluations de la boîte noire disponibles. Il est possible d'utiliser d'autres modèles pour l'étape de recherche, tels que le processus gaussien et les fonctions à base radiale. Seules les *surrogates* 3 et 4 sont testées car la *surrogate* 2 est trop lente et la *surrogate* 1 est dominée par la *surrogate* 4 (en termes de temps et de corrélation). Les *surrogates* statiques sont utilisées pour réorganiser les points de la *poll step*, avant qu'ils ne soient évalués par la boîte noire, car la stratégie dite opportuniste est utilisée. En fait, ce traitement économise considérablement le nombre d'évaluations de la boîte noire et permet à l'optimisation d'aller plus loin, d'explorer davantage l'espace des variables et, souvent, de trouver un meilleur optimum que celui trouvé sans *surrogate*.

Dans un souci de simplicité et d'efficacité, le problème d'optimisation a été résolu plusieurs fois pour  $K$  fixé à des valeurs comprises entre 2 et  $K_{max}$ . Le nombre maximal d'évaluations est fixé à 1000 évaluations par valeur de  $K$ . Pour chaque valeur de  $K$ , le point initial est défini avec  $\omega_i = 1$ ,  $i = 1, 2, 3, 4$ ,  $\Theta_k$  égale à la direction moyenne de plus forte pente du cluster  $k$ , et  $\Phi_k = \Psi_k = 0$  pour chaque cluster  $k$ . Pour chaque optimisation, le gain  $\eta$  est défini par rapport à ce point initial :

$$\eta = \frac{f_0 - f^*}{f_0} \quad (4.4)$$

où  $f_0$  est la valeur de la fonction-objectif au point initial, et  $f^*$ , la valeur finale de la fonction-objectif, donnée par NOMAD. Le gain moyen  $\bar{\eta}$  sur toutes les valeurs de  $K$  est utilisé pour comparer globalement les résultats obtenus avec ou sans utilisation de *surrogates*.

Par ailleurs, l’évaluation des *surrogates* est comptabilisée dans le temps de calcul au prorata de leur coût en temps par rapport à celui de la boîte noire. Pour des raisons de comparaison, le processus d’optimisation est arrêté après un temps équivalent à 1000 évaluations de la boîte noire.

#### 4.4.1 Modèle 3-axes

Pour l’usinage 3-axes, tous les angles  $\Phi_k$  et  $\Psi_k$  sont fixés à zéro. Ainsi, le problème d’optimisation à résoudre est donné par l’équation 4.5. Pour chaque surface, trois optimisations ont été exécutées : Une sans fonction *surrogate*, l’autre avec la *surrogate 3* et la dernière avec la *surrogate 4*. Le tableau 4.3 résume les résultats de l’optimisation pour chaque cas sur la surface test 1.

$$\begin{aligned}
 \min_{K, \omega, \Theta} \quad & \text{machining\_time}(K, \omega, \Theta) \\
 \text{s.t.} \quad & 1 \leq K \leq K_{max} \\
 & 0 \leq \omega_i \leq 1, i = 1, \dots, 4 \\
 & 0 \leq \Theta_k \leq \pi, k = 1, \dots, K
 \end{aligned} \tag{4.5}$$

$K$	valeur initiale $f_0$	$f^*$ , sans <i>surrogate</i>	$f^*$ , <i>surrogate 3</i>	$f^*$ , <i>surrogate 4</i>	$\eta$ , sans <i>surrogate</i>	$\eta$ , <i>surrogate 3</i>	$\eta$ , <i>surrogate 4</i>
2	96.19	92.01	92.01	92.01	4.34%	4.34%	4.34%
3	104.77	93.92	92.92	92.52	10.35%	11.31%	11.69%
4	147,68	98.5	95.92	95.06	33.3%	35.04%	35.63%
5	101.90	96.41	96.23	100.32	5.38%	5.56%	1.55%
6	100.1	95.96	98.41	97.3	4.13%	1.68%	2.79%
7	103.78	102.02	100.06	102.13	1.69%	3.58%	1.58%
8	102.4	95.48	91.93	92.87	6.75%	10.22%	9.3%
9	153,5	102.7	95.29	95.88	33.09%	37.92%	37.53%
10	143,9	97.25	97.91	97.21	32.41%	31.96%	32.44%

TABLE 4.3 – Résultats de l’optimisation de l’usinage 3-axes pour la surface 1

Pour la surface 1, l’initialisation fournit une valeur minimale de  $f_0 = 96.19$  s pour  $K = 2$ , et une valeur maximale de  $f_0 = 153.5$  s pour  $K = 9$ . Lorsqu’aucune *surrogate* n’est utilisée, le point optimal est atteint avec  $K = 2$  et le temps d’usinage minimal  $f^*$  est 92.01 s. Le même point optimal est trouvé en utilisant la *surrogate 4*, tandis que la *surrogate 3* a donné un temps d’usinage optimal de 91.93 s pour  $K = 8$ . Le gain moyen  $\bar{\eta}$  est égal à 14,6% dans le cas où aucune *surrogate* n’est utilisée, 15,73% si la *surrogate 3* est utilisée et 15,2% avec la *surrogate 4*.

Les trois optimisations ont pris presque le même temps CPU, à savoir 35 min. Les graphiques de

l'historique de l'optimisation, présentés dans la figure 4.6, montrent que les fonctions *surrogate*, en particulier la *surrogate 3*, accélèrent la convergence en nécessitant moins d'évaluations de la boîte noire (pour arriver à une même valeur de la fonction objectif), surtout lorsque le nombre de variables est élevé.

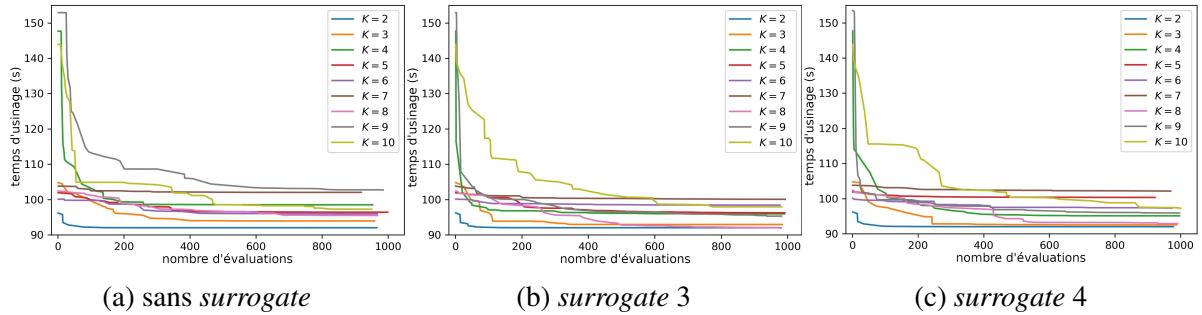


Figure 4.6 – Historique d'optimisation en 3-axes pour la surface 1

Le tableau 4.4 résume les résultats de l'optimisation pour la surface 2. Pour cette surface, l'initialisation fournit une valeur minimale de  $f_0 = 91.54$  s pour  $K = 2$ , et une valeur maximale de  $f_0 = 116.11$  s pour  $K = 10$ . Lorsqu'aucune *surrogate* n'est utilisée, le point optimal est atteint avec  $K = 4$  et le temps d'usinage minimal  $f^*$  est 78.86 s. Le même point optimal est trouvé en utilisant la *surrogate 3*, tandis que la *surrogate 4* a donné un temps d'usinage optimal de 87.28 s pour  $K = 6$ . Le gain moyen  $\bar{\eta}$  est égal à 13.64% dans le cas où aucune *surrogate* n'est utilisée, 14.18% si la *surrogate 3* est utilisée et 12.95% avec la *surrogate 4*.

$K$	valeur initiale $f_0$	$f^*$ , sans surrogate	$f^*$ , surrogate 3	$f^*$ , surrogate 4	$\eta$ , sans surrogate	$\eta$ , surrogate 3	$\eta$ , surrogate 4
2	91.54	90.91	90.82	91.01	0.68%	0.78%	0.57%
3	91.62	86.83	90.77	90.63	5.22%	0.92%	1.08%
4	103.53	78.86	78.86	87.55	23.82%	23.82%	15.43%
5	96.75	90.06	87.26	89.49	6.91%	9.8%	7.5%
6	102.8	88.49	90.03	87.28	13.92%	12.42%	14.52%
7	106.1	89.05	86.89	87.87	16.06%	18.11%	17.18%
8	108.55	93.61	86.52	90.32	13.76%	20.3%	16.79%
9	108.23	88.39	86.75	87.94	18.33%	19.85%	18.75%
10	116.11	88.11	90.97	87.38	24.11%	21.65%	24.74%

TABLE 4.4 – Résultats de l'optimisation de l'usinage 3-axes pour la surface 2

Les graphiques de l'historique d'optimisation présentés dans la figure 4.7 montrent encore que les fonctions *surrogate*, en particulier la *surrogate 3*, rendent la convergence plus rapide, surtout lorsque le

nombre de variables est élevé. La durée d'exécution de chacune des trois optimisations a été d'environ 33 min.

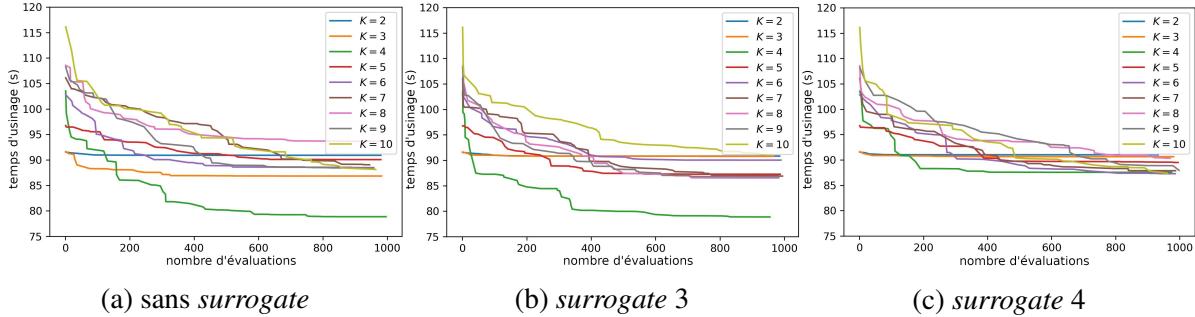


Figure 4.7 – Historique d'optimisation en 3-axes pour la surface 2

#### 4.4.2 Modèle 3+2-axes

En raison du nombre élevé de variables d'optimisation impliquées dans l'usinage 3+2 axes, notamment pour les valeurs élevées de  $K$ , l'étape de recherche "search step" est ignorée. En fait, la construction de modèles de recherche peut prendre beaucoup de temps dans ce cas. De plus, la *surrogate 3* est celle qui fournit les meilleurs résultats, ainsi, dans ce qui suit, seuls ses résultats sont comparés au cas sans *surrogate*.

Le tableau 4.5 résume les résultats de l'optimisation pour la surface 1. Pour cette surface, l'initiali-

$K$	valeur initiale $f_0$	$f^*$ , sans <i>surrogate</i>	$f^*$ , <i>surrogate 3</i>	$\eta$ , sans <i>surrogate</i>	$\eta$ , <i>surrogate 3</i>
2	96.19	85.46	85.68	11.15%	10.92%
3	104.77	97.24	83.59	7.18%	20.21%
4	147,68	100.11	100.14	32.21%	32.19%
5	101.90	98.42	96.2	3.41%	5.59%
6	100.1	97.19	97.41	2.9%	2.69%
7	103.78	96.00	97.8	7.49%	5.76%
8	102.4	93.48	94.36	8.71%	7.85%
9	153,5	108.54	91.93	29.28%	40.11%
10	143,9	99.70	96.09	30.71%	33.22%

TABLE 4.5 – Résultats de l'optimisation de l'usinage 3+2-axes pour la surface 1

sation fournit une valeur minimale de  $f_0 = 96.19$  s pour  $K = 2$ , et une valeur maximale de  $f_0 = 153.5$  s pour  $K = 9$ . Lorsqu'aucune *surrogate* n'est utilisée, le point optimal est atteint avec  $K = 2$  et le temps d'usinage minimal  $f^*$  est de 85.46 s, tandis que la *surrogate 3* donne un temps d'usinage optimal de

83.59 s pour  $K = 3$ . Le gain moyen  $\bar{\eta}$  est égal à 14.78% dans le cas où aucune *surrogate* n'est utilisée, 15.64% si la *surrogate* 3 est utilisée. En comparant les meilleurs résultats fournis par l'optimisation sans *surrogate* et l'optimisation avec la *surrogate* 3, il apparaît qu'un gain de 2.18% sur le temps d'usinage optimal est obtenu grâce à l'utilisation de la *surrogate* 3.

Encore une fois, comme le montrent les graphiques d'historique d'optimisation présentés dans la figure 4.8, le *surrogate* 3 a rendu la convergence plus rapide, nécessitant moins d'évaluations de la boîte noire, surtout lorsque le nombre de variables est élevé. Les deux optimisations ont pris presque le même temps CPU, environ 1 heure.

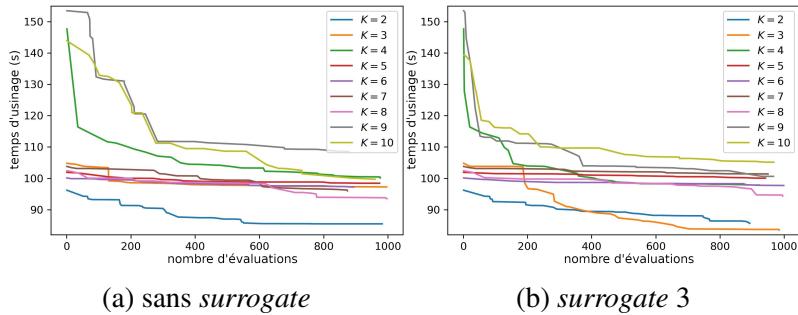


Figure 4.8 – Historique d'optimisation en 3+2-axes pour la surface 1

Le tableau 4.6 résume les résultats de l'optimisation pour la surface 2.

$K$	valeur initiale $f_0$	$f^*$ , sans <i>surrogate</i>	$f^*$ , <i>surrogate</i> 3	$\eta$ , sans <i>surrogate</i>	$\eta$ , <i>surrogate</i> 3
2	91.54	75.67	74.96	17.33%	18.11%
3	91.62	77.23	73.84	15.70%	19.4%
4	103.53	79.94	71.95	22.78%	30.5%
5	96.75	83.67	82.01	13.51%	15.23%
6	102.8	86.46	85.43	15.89%	16.89%
7	106.1	91.29	82.23	13.95%	22.5%
8	108.55	93.65	84.03	13.72%	22.58%
9	108.23	91.28	93.91	15.66%	13.23%
10	116.11	98.76	92.04	14.94%	20.73%

TABLE 4.6 – Résultats de l'optimisation de l'usinage 3+2-axes pour la surface 2

Pour cette surface, l'initialisation fournit une valeur minimale de  $f_0 = 91.54$  s pour  $K = 2$ , et une valeur maximale de  $f_0 = 116.11$  s pour  $K = 10$ . Lorsqu'aucune *surrogate* n'est utilisée, le point optimal est atteint pour  $K = 2$  et le temps d'usinage minimal  $f^*$  est de 75.67 s, tandis que la *surrogate* 3 a donné un temps d'usinage optimal de 71.95 s pour  $K = 4$ . Le gain moyen  $\bar{\eta}$  est égal à 15.94% dans le cas

où aucune *surrogate* n'est utilisée, et 16.75% si la *surrogate* 3 est utilisée. En comparant les meilleurs résultats fournis par l'optimisation sans *surrogate* et l'optimisation avec la *surrogate* 3, il apparaît qu'un gain de 4.91% sur le temps d'usinage optimal est obtenu grâce à l'utilisation de la *surrogate* 3.

Comme prévu, l'historique d'optimisation présenté dans la figure 4.9, montre que la *surrogate* 3 a rendu la convergence plus rapide, nécessitant moins d'évaluations que la boîte noire, surtout lorsque le nombre de variables est élevé. Les deux optimisations ont pris presque le même temps CPU, environ 1 heure.

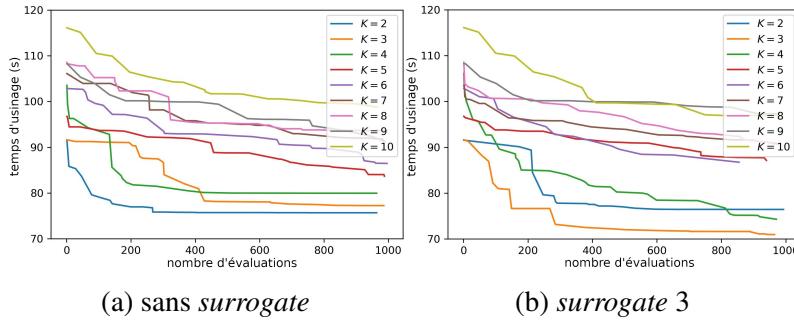


Figure 4.9 – Historique d'optimisation en 3+2-axes pour la surface 2

## 4.5 Heuristique 3+2-axes

Le problème d'optimisation associé à l'usinage 3+2 axes comporte beaucoup de variables ce qui rend les évaluations coûteuses. Dans cette section on propose donc plutôt une approche heuristique. Les variables de *clustering* sont fixées dès le départ et le choix des variables d'usinage est guidé par l'ACP et leur nombre est réduit.

### 4.5.1 Considérations générales

En optimisation et en science de données, une heuristique est une méthode spécialisée conçue pour résoudre une classe spécifique de problèmes de manière plus rapide et moins coûteuse que les méthodes conventionnelles, au détriment de la garantie d'optimalité. De telles méthodes, pour être efficaces, sont souvent basées sur des connaissances métier liées au domaine d'application ou exploitent des structures mathématiques particulières du problème d'optimisation considéré. Les heuristiques sont par exemple largement utilisées pour résoudre les problèmes de décision de classe NP-difficile [23]. Pour ce type de problèmes, aucun algorithme permettant la résolution exacte en un temps polynomial n'est connu, bien qu'il soit possible de vérifier en un temps polynomial si une alternative (un point donné) est solution ou pas.

Les méthodes heuristiques peuvent, soit générer directement des solutions, soit être utilisées pour fournir une bonne base de référence et être complétées *a posteriori* par d'autres algorithmes d'optimisa-

tion. Les méthodes heuristiques sont le plus souvent utilisées lorsque les solutions approximatives sont suffisantes et que les solutions exactes sont nécessairement coûteuses en termes de calcul.

Parmi ces méthodes, on compte les méta-heuristiques (une méta-heuristique est une heuristique générique qu'il faut adapter à chaque problème) dont les plus courantes d'utilisation incluent :

- la famille des algorithmes "*swarm intelligence*" tels que le "*particle swarm optimization*", "*ant colony optimization*" ou encore le "*artificial bee colony optimization*". Ces algorithmes sont inspirés par le comportement naturel et auto-organisé de certains animaux, et utilisent un certain nombre d'agents qui peuvent interagir entre eux.
- le "*tabu search algorithm*", ou la recherche tabou. Cette méthode permet de contourner le problème des minima locaux, rencontrés souvent avec les méthodes d'optimisation locales, à savoir être bloqué au voisinage d'un minimum local. Ceci est fait en considérant un ensemble de règles définies dynamiquement qui empêchent la recherche autour d'une même région de façon redondante.
- Le recuit simulé (*simulated annealing*), est une méthode stochastique inspirée de la métallurgie, qui accepte avec une probabilité (et surtout au début de l'optimisation) des pas de recherche conduisant à une détérioration de la fonction objectif, ce qui tend à se détacher des minima locaux. C'est une méthode de référence lorsqu'il s'agit d'optimisation globale.
- Les algorithmes génétiques qui sont une sous-classe des algorithmes évolutionnaires : à chaque itération une nouvelle génération de solutions est créée à partir de l'ancienne par des opérateurs d'élitisme, de croisements et de mutations.

Quelques exemples de problèmes où l'utilisation des méthodes heuristiques est commune sont :

- le célèbre problème du voyageur de commerce. Une méthode heuristique basique utilisée pour résoudre approximativement mais rapidement le problème est le voisin plus proche "*nearest neighbor*", une méthode glouton qui consiste tout simplement à partir d'une ville donnée puis choisir la ville la plus proche et ainsi de suite pour construire un circuit.
- le problème du sac à dos est aussi un problème d'optimisation très classique. Une heuristique utilisée pour résoudre le problème consiste à ordonner les objets selon leur rapport de valeur par masse (ou volume ou autre dimension limitant la capacité du sac), puis mettre les objets dans l'ordre dans le sac jusqu'à ce qu'il n'y ait plus de place.

Dans les deux exemples ci-dessous les méthodes heuristiques utilisées produisent directement une solution réalisable du problème.

### 4.5.2 Définition de l'heuristique basée sur l'ACP

Dans le contexte de l'usinage, on dispose de plusieurs variables qui représentent des choix de natures différentes (directions d'usinages, orientation de la pièce en usinage 3+2 axes, paramètres de partitionnement, etc), ce qui rend le problème d'optimisation assez difficile, et exige souvent un grand nombre d'évaluations de la boîte noire (temps d'usinage), sans garanti sur l'optimalité de la solution trouvée.

Ainsi, l'heuristique qu'on propose ici, a pour objectif de réduire considérablement le nombre de variables, et donc le temps d'optimisation par la suite, tout en proposant une solution construite par définition pour minimiser le temps d'usinage en 3+2 axes. L'optimalité (globale) de la solution proposée pourra être démontrée dans le cas de surfaces planes rectangulaires.

Cette heuristique repose avant tout sur la connaissance métier, elle est donc applicable exclusivement au problème d'usinage. Le constat de départ est que lorsque la stratégie d'usinage en plans parallèles est utilisée, minimiser la longueur de la trajectoire d'usinage revient à maximiser les pas transversaux (distances entre trajectoires adjacentes), du moins pour les surfaces planes, ce qui revient à usiner selon la direction de plus grande pente.

En second lieu, pour une longueur de trajectoire totale donnée, la durée d'usinage peut varier selon le nombre et la longueur des trajectoires constituant le parcours d'outil. Ceci est dû aux limitations cinématiques de la machine-outil qui fait en sorte que la vitesse d'avance n'est pas constante et que la vitesse de consigne n'est pas atteinte sur toute la trajectoire. Ainsi, le temps d'usinage est aussi influencé par la cinématique de l'outil, qui est le plus efficace lorsque la vitesse de consigne est atteinte sur la plus grande partie possible de la trajectoire. Ceci est réalisé lorsque les trajectoires générées sont les plus longues possibles, ce qui implique d'usiner selon la direction principale.

Pour récapituler, on dispose de deux critères pour améliorer le temps d'usinage : le premier exige d'usiner selon la direction de plus grande pente pour minimiser la longueur de la trajectoire d'usinage. Le second exige d'usiner selon la direction principale de la surface afin que la vitesse de consigne soit atteinte sur la plus grande partie de la trajectoire. Ces deux directions étant généralement différentes, il n'est donc pas possible de satisfaire ces deux critères simultanément en usinage 3-axes. Cependant, en 3+2-axes il est possible de faire coïncider ces deux directions à l'aide de rotations de la surface, c'est ce que fait l'heuristique proposée ici.

Considérons une surface (ou zone de surface) donnée définie dans le repère machine ( $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ ), ayant une direction de plus grande pente moyenne définie par le vecteur  $\mathbf{D}$  (figure 4.10), où  $\alpha$  désigne l'angle que fait la projection de ce vecteur dans le plan ( $\mathbf{X}, \mathbf{Y}$ ) avec l'axe  $\mathbf{X}$ . De plus, la direction principale de cette surface (ou zone de surface), appelée  $\mathbf{X}_I$ , est calculée par ACP.

En notant  $\beta$  l'angle entre l'axe  $\mathbf{X}_{III}$  et  $\mathbf{Z}$ , on opère une première rotation autour de l'axe  $\mathbf{X}_{III} \wedge \mathbf{Z}$  (produit vectoriel) d'angle  $\beta$  si  $\mathbf{X}_{III} \cdot \mathbf{Z} > 0$  ou  $\beta - \pi$  si  $\mathbf{X}_{III} \cdot \mathbf{Z} < 0$ . La nouvelle pente moyenne (normalement proche de 0) et la nouvelle direction de plus grande pente peuvent être calculées à ce stade. Enfin, une seconde rotation autour de  $\mathbf{X}_{II}$ , notée  $\zeta$ , est appliquée afin de créer de la pente selon  $\mathbf{X}_I$ , ce qui a pour objectif de faire coïncider la nouvelle direction de plus grande pente avec  $\mathbf{X}_I$  (c'est à dire pour que  $\mathbf{D}$  soit contenu dans le plan ( $\mathbf{X}_I, \mathbf{Z}$ )). L'angle de rotation  $\zeta$  est choisi tel que les deux directions projetées dans ( $\mathbf{X}, \mathbf{Y}$ ) deviennent confondues et que la pente moyenne de la surface demeure la plus petite possible. Il convient cependant de noter que cette rotation n'est pas immédiatement connue étant donné que le vecteur  $\mathbf{D}$  de la moyenne des pentes que l'on cherche à faire coïncider avec la direction principale  $\mathbf{X}_I$  dépend lui-même de l'orientation de la surface. En première approche cependant, il est possible de

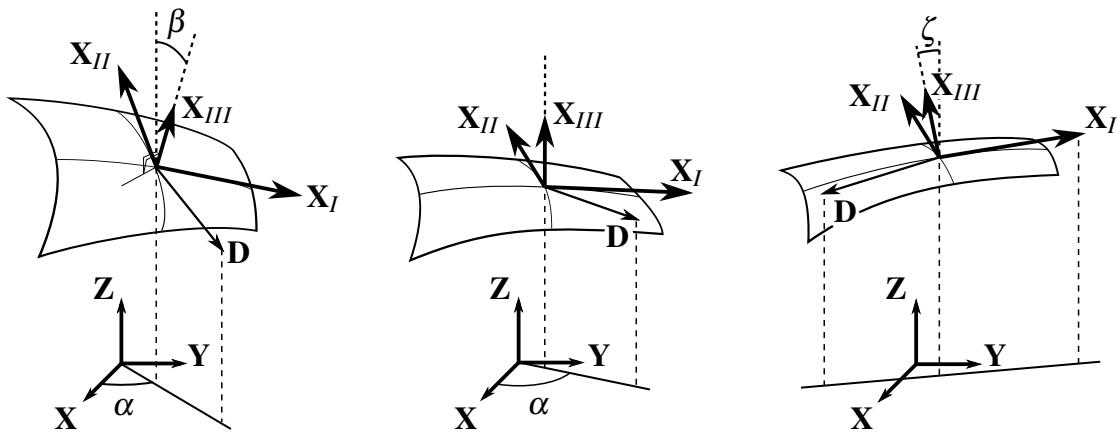


Figure 4.10 – Définition de l’heuristique, la première rotation autour de  $\mathbf{X}_{III} \times \mathbf{Z}$  aligne  $\mathbf{X}_{III}$  avec  $\mathbf{Z}$ , la deuxième rotation autour de  $\mathbf{X}_{II}$  aligne  $\mathbf{X}_I$  et  $\mathbf{D}$

déterminer la valeur de cet angle  $\zeta$  par itérations successives en l’augmentant petit à petit, à partir de  $0^\circ$ , et jusqu’à ce que les deux directions soient suffisamment proches. Une autre approche, plus générique, est de considérer la détermination de cet angle comme un problème d’optimisation, la valeur de  $\zeta$  devenant alors une variable d’optimisation de ce problème. Ainsi formulé, le problème est plus souple, car, comme on le verra par la suite, il est alors possible d’ajouter d’autres variables de manière à élargir un peu l’espace de recherche autour de la solution attendue, qui, étant donné la formulation du problème, est une direction de plus grande pente coïncidant avec la direction principale.

Quelle que soit la méthode retenue pour déterminer la rotation  $\zeta$ , une fois la surface orientée, la plus grande pente moyenne coïncide normalement avec la direction principale, ce qui constitue une direction d’usinage préférentielle.

La procédure de l’heuristique se résume donc par les étapes suivantes :

1. Calculer les directions principales  $\mathbf{X}_I, \mathbf{X}_{II}, \mathbf{X}_{III}$  par ACP, la direction de plus grande pente moyenne  $\mathbf{D}$  dont la projection forme un angle  $\alpha$  avec  $\mathbf{X}$  dans le plan  $(\mathbf{X}, \mathbf{Y})$ , et le centre de la surface (ou zone)  $\mathbf{S}(u_0, v_0)$ .
2. Effectuer une première rotation autour de l’axe  $\mathbf{X}_{III} \wedge \mathbf{Z}$  d’angle  $\beta$  si  $\mathbf{X}_{III} \cdot \mathbf{Z} > 0$  ou  $\beta - \pi$  si  $\mathbf{X}_{III} \cdot \mathbf{Z} < 0$ .
3. Calculer le nouveau vecteur  $\mathbf{D}$  et la nouvelle direction de plus grande pente  $\alpha$ .
4. Effectuer une deuxième rotation autour de l’axe  $\mathbf{X}_{II}$ , l’angle  $\zeta$  de cette rotation étant calculé de sorte que  $\mathbf{D}$  coïncide avec  $\mathbf{X}_I$  (c’est à dire pour que  $\mathbf{D}$  soit contenu dans le plan  $(\mathbf{X}_I, \mathbf{Z})$ ).
5. Usiner selon la direction principale (qui aussi la direction de plus grande pente).

Comme on l’a annoncé précédemment, il est possible de déterminer la valeur de la rotation  $\zeta$  en définissant un problème d’optimisation. Dans sa forme la plus simple, ce problème consiste à déterminer la valeur de  $\zeta$  permettant de réduire au minimum le temps d’usinage. À cela, s’ajoute la possibilité d’introduire d’autres variables d’optimisation telles que le nombre de zones  $K$  ou la direction d’usinage

Θ. Ainsi formulé, le problème d'optimisation s'écrit :

$$\begin{aligned}
 \min_{K, \Theta, \zeta} \quad & \text{machining\_time}(K, \Theta, \zeta) \\
 \text{s.c.} \quad & 2 \leq K \leq 10 \\
 & 0 \leq \Theta_k \leq \pi, k = 1, \dots, K \\
 & -\frac{\pi}{2} \leq \zeta_k \leq \frac{\pi}{2}, k = 1, \dots, K \\
 & |\arccos(\mathbf{n}_{\zeta_k}(u_i, v_i) \cdot \mathbf{Z})| \leq \frac{\pi}{2}, i = 1, \dots, N_k, k = 1, \dots, K
 \end{aligned} \tag{4.6}$$

Cette formulation du problème est celle qui a été utilisée pour fournir les résultats présentés en section 4.5.4.

Ainsi, l'heuristique proposée peut être utilisée pour résoudre le problème d'optimisation avec NOMAD de la façon suivante : d'abord la surface est partitionnée en un nombre  $K$  de zone en utilisant l'algorithme *K-means* et la métrique euclidienne, ensuite NOMAD est lancé avec les variables d'optimisation introduite par l'heuristique pour chaque zone. À chaque évaluation de la boîte noire, le temps d'usinage est calculé en orientant la zone grâce à l'heuristique. NOMAD renvoie les valeurs des variables d'optimisation qui minimisent le temps d'usinage ainsi calculé.

### 4.5.3 Remarques sur l'optimalité de l'heuristique

Il est à noter que la solution fournie par cette heuristique est, en règle générale, optimale pour les surfaces planes. En effet, en utilisant cette heuristique pour ces surfaces, la direction principale (qui optimise la cinématique de la machine) coïncide avec la direction de plus grande pente partout (sur toute la zone), ce qui optimise à la fois la réponse cinématique de la machine et la longueur du parcours. Bien que l'on n'ait pas réussi à démontrer cela, il semble qu'une condition suffisante pour que ce résultat optimal soit atteint est que la surface plane en question ait un pourtour convexe (ce qui est très fréquemment le cas pour les surfaces à usiner), ou ne présente pas de forte non-convexité. Cependant, il est à noter que les surfaces planes dont le pourtour présente de fortes concavités ne sont jamais usinées en pratique d'un seul coup, mais découpées par les usineurs en plusieurs zones (généralement sans utiliser d'algorithmes de *clustering* ni d'optimisation). Il est d'ailleurs assez facile de trouver un contre-exemple pour lequel l'heuristique ne fournit pas la valeur optimale pour une surface plane. Un tel exemple est donné par la figure 4.11 où on usine une surface plane (dans le plan horizontal  $\mathbf{X}, \mathbf{Y}$ ) en forme de croix. Pour simplifier les calculs, cette surface est usinée avec un outil sphérique de rayon  $R = 5$  mm, selon l'axe  $\mathbf{X}$  dans un premier cas, puis selon l'axe  $\mathbf{Y}$ , mais la même démonstration peut être réalisée avec un outil torique. Les dimensions de la surface  $w, h, p$  et  $q$  seront choisies de telle sorte que la direction principale de la surface coïncide avec l'axe  $\mathbf{X}$  mais que le temps d'usinage soit meilleur selon l'axe  $\mathbf{Y}$ .

Les paramètres cinématiques de la machine sont  $J = 40$  m/s<sup>3</sup>,  $A = 6$  m/s<sup>2</sup> et  $V = 5$  m/min. La hauteur de crête est  $sh = 0.01$  mm. Ici, le pas transversal *sod* est constant sur toute la surface et donné

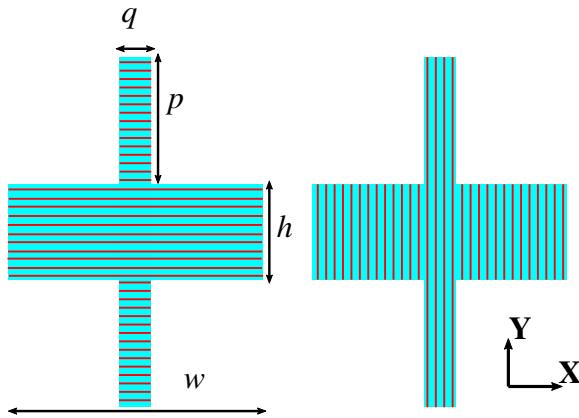


Figure 4.11 – Exemple montrant que la direction principale n'est pas toujours celle qui optimise la cinématique de l'outil

par :

$$sod = 2 \sqrt{2shR - sh^2}$$

Pour le premier cas (usinage selon l'axe **X**), la longueur d'usinage est :

$$L = 2L_1 + L_2 \text{ avec } L_1 = \frac{p}{sod} (q + sod) \text{ et } L_2 = \frac{h}{sod} (w + sod)$$

On fixe les valeurs numériques des dimensions de la surface à  $p = 40$  mm,  $q = 10$  mm,  $w = 80$  mm et  $h = 30$  mm, la durée d'usinage correspondante est donc :

$$T = 2T_1 + T_2 \text{ avec } T_1 = \frac{p}{sod} \left( \frac{q}{V} + 2\sqrt{\frac{V}{J}} + 4\sqrt[3]{\frac{sod}{2J}} \right) \text{ et } T_2 = \frac{h}{sod} \left( \frac{q}{V} + 2\sqrt{\frac{V}{J}} + 4\sqrt[3]{\frac{sod}{2J}} \right)$$

De la même façon, la longueur de trajectoire d'usinage pour le deuxième cas (usinage selon l'axe **Y**) s'écrit :

$$L = 2L_1 + L_2 \text{ avec } L_1 = \frac{w-q}{sod} (h + sod) \text{ et } L_2 = \frac{q}{sod} (h + 2p + sod)$$

Le temps d'usinage est :

$$T = 2T_1 + T_2 \text{ avec } T_1 = \frac{w-q}{2sod} \left( \frac{h}{V} + 2\sqrt{\frac{V}{J}} + 4\sqrt[3]{\frac{sod}{2J}} \right) \text{ et } T_2 = \frac{q}{sod} \left( \frac{h+2p}{V} + 2\sqrt{\frac{V}{J}} + 4\sqrt[3]{\frac{sod}{2J}} \right)$$

Pour s'assurer que la direction principale est selon l'axe **X**, on calcule la matrice de covariance des coordonnées de la surface. Vu que  $S_z = 0$  sur toute la surface, on peut considérer la matrice de taille  $2 \times 2$  avec seulement les coordonnées  $S_x$  et  $S_y$  :

$$C = \begin{pmatrix} \text{Var}(S_x) & \text{Cov}(S_x, S_y) \\ \text{Cov}(S_x, S_y) & \text{Var}(S_y) \end{pmatrix}$$

avec  $\text{Var}(S_x) = \text{E}(S_x^2) - \text{E}(S_x)^2$ , la moyenne  $\text{E}(S_x)$  est égale à 0 puisque la surface est symétrique et centrée à l’origine du repère ; en effet :

$$\text{E}(S_x) = \frac{1}{|S|} \int_S S_x dS = \frac{1}{|S|} \left[ 2 \int_{-\frac{q}{2}}^{\frac{q}{2}} x dx p + \int_{-\frac{w}{2}}^{\frac{w}{2}} x dx h \right] = 0$$

où  $|S| = wh + 2pq$ . Par ailleurs :

$$\begin{aligned} \text{E}(S_x^2) &= \frac{1}{|S|} \int_S S_x^2 dS = \frac{1}{|S|} \left[ 2 \int_{-\frac{q}{2}}^{\frac{q}{2}} x^2 dx p + \int_{-\frac{w}{2}}^{\frac{w}{2}} x^2 dx h \right] \\ &= \frac{1}{|S|} \left( 2p \left[ \frac{x^3}{3} \right]_{-\frac{q}{2}}^{\frac{q}{2}} + h \left[ \frac{x^3}{3} \right]_{-\frac{w}{2}}^{\frac{w}{2}} \right) = \frac{2pq^3 + hw^3}{12(wh + 2pq)} \end{aligned}$$

Ainsi, on trouve  $\text{Var}(S_x) = \frac{2pq^3 + hw^3}{12(wh + 2pq)}$ , et de la même façon  $\text{Var}(S_y) = \frac{(w-q)h^3 + q(2p+h)^3}{12[q(2p+h) + (w-q)h]}$ . La matrice de covariance s’écrit :

$$\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \text{ avec } \lambda_1 = \text{Var}(S_x) \text{ et } \lambda_2 = \text{Var}(S_y)$$

Avec les valeurs numériques de  $p, q, w$  et  $h$ , on trouve  $\lambda_1 = 402 \text{ mm}^2 > \lambda_2 = 395 \text{ mm}^2$ . Ce qui signifie que la direction principale  $\mathbf{X}_I$  est bien selon l’axe  $\mathbf{X}$ . Par contre, le temps d’usinage selon l’axe  $\mathbf{X}_I = \mathbf{X}$  est de 90.4 s alors que le temps d’usinage selon l’axe  $\mathbf{X}_{II} = \mathbf{Y}$  est de 82.3 s. Pour cette surface, le temps d’usinage selon  $\mathbf{X}_I$  est donc supérieur au temps d’usinage selon  $\mathbf{X}_{II}$ , ce qui prouve que cette surface est un contre-exemple. La longueur de la trajectoire d’usinage est presque la même pour les deux cas (5172 mm et 5142 mm), ce qui n’est pas surprenant vu qu’on usine une surface plane avec un outil sphérique.

Encore une fois, en pratique, cette surface ne serait probablement pas usinée en une seule fois. En pratique, un usineur reconnaîtra sans difficultés 3 zones à usiner, à savoir la section horizontale au centre et les deux sections verticales en haut et en bas. Dans ce cas, chaque section sera usinée selon sa direction principale (section centrale selon l’axe  $\mathbf{X}$  et sections verticales selon  $\mathbf{Y}$ ), ce qui conduit à une durée d’usinage de 74.2 s et une longueur de 5112 mm. Le temps d’usinage est bien inférieur au deux cas précédents, ce qui tend à montrer que les surfaces à forte non-convexité, qui risquent d’avoir une direction principale qui n’est pas celle qui optimise la cinématique de la machine, sont généralement des surfaces pour lesquelles on pense directement à usiner en plusieurs zones, et qu’en usinant chaque zone selon sa direction principale, la cinématique de la machine est bien mieux exploitée.

La métrique euclidienne a été utilisée plutôt que celle basée sur l’ACP (introduite au chapitre précédent) avec l’heuristique bien que celle-ci soit aussi basée sur l’ACP. La raison derrière ce choix est que la métrique ACP cherche un compromis entre la direction principale et la direction de plus grande pente, alors qu’en usinage 3+2 axes il est possible d’aligner ces directions on orientant la surface correctement

(et c'est justement ce que fait l'heuristique).

#### 4.5.4 Résultats de l'heuristique

L'heuristique proposée est appliquée sur les deux surfaces extraites de la bibliographie et introduites précédemment (figure 4.3) ainsi que sur une surface industrielle (une pale — figure 4.14). Les résultats de l'optimisation sont présentés ci dessous, pour différentes valeurs de  $K$ . Seule la *surrogate 3* est utilisée. Le modèle d'optimisation correspondant est donné par l'équation 4.6. Les variables d'optimisation sont le nombre de zones  $K$ , et pour chaque zone  $k \in \{1, \dots, K\}$ , la direction d'usinage  $\Theta_k$  et l'angle de rotation  $\zeta_k$  par rapport à la direction principale  $\mathbf{X}_{II}^k$  de la zone. La contrainte d'accessibilité de l'outil à la surface doit être toujours vérifiée (pour éviter les contre-dépouilles). Le point initial, pour chaque valeur de  $K$ , correspond à la direction d'usinage selon la plus grande pente de chaque zone avec une rotation  $\zeta_k = 0$ .

Le tableau 4.7 résume les résultats de l'optimisation pour la surface 1. Pour cette surface, le meilleur

$K$	2	3	4	5	6	7	8	9	10
$f_0$	121.24	117.51	93.49	94.32	91.5	95.75	93.64	99.47	98.47
$f^*$	77.71	83.15	87.68	84.98	83.0	86.57	86.0	92.61	91.98
$\eta$	35.9%	29.2%	6.2%	9.9%	9.2%	9.6%	8.1%	6.9%	6.6%

TABLE 4.7 – Résultats de l'optimisation avec l'heuristique basée sur l'ACP pour l'usinage en 3+2-axes de la surface 1

point de départ était pour  $K = 6$  avec une valeur de fonction-objectif  $f_0$  de 91,5 s et le pire point de départ donne 121.24 s pour  $K = 2$ . Le point optimal est atteint pour  $K = 2$  et le temps d'usinage minimal  $f^*$  est de 77.71 s. Le gain moyen  $\bar{\eta}$  est égal à 13.5%. L'historique de l'optimisation est affiché dans la figure 4.12.

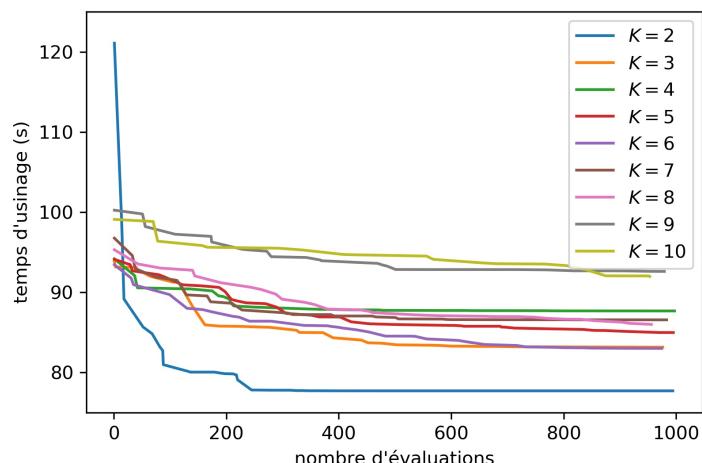


Figure 4.12 – Historique d'optimisation avec l'heuristique basée sur l'ACP pour l'usinage en 3+2-axes de la surface 1

$K$	2	3	4	5	6	7	8	9	10
$f_0$	69.96	65.38	107.82	70.83	86.70	100.74	84.86	101.86	112.98
$f^*$	65.78	62.65	66.11	63.39	64.24	62.64	60.7	59.58	61.34
$\eta$	6%	4%	38.6%	10.5%	25.9%	37.8%	28.4%	41.5%	45.7%

TABLE 4.8 – Résultats de l'optimisation avec l'heuristique basée sur l'ACP pour l'usinage en 3+2-axes de la surface 2

Le tableau 4.8 résume les résultats de l'optimisation pour la surface 2. Pour cette surface, le meilleur point de départ était pour  $K = 3$  avec une valeur de fonction objectif  $f_0$  de 65.38 s et le pire point de départ donne 112,98 s pour  $K = 10$ . Le point optimal est atteint pour  $K = 9$  et le temps d'usinage minimal  $f^*$  est de 59.58 s. Le gain moyen  $\bar{\eta}$  est égal à 26.4%. L'historique de l'optimisation est affiché dans la figure 4.13.

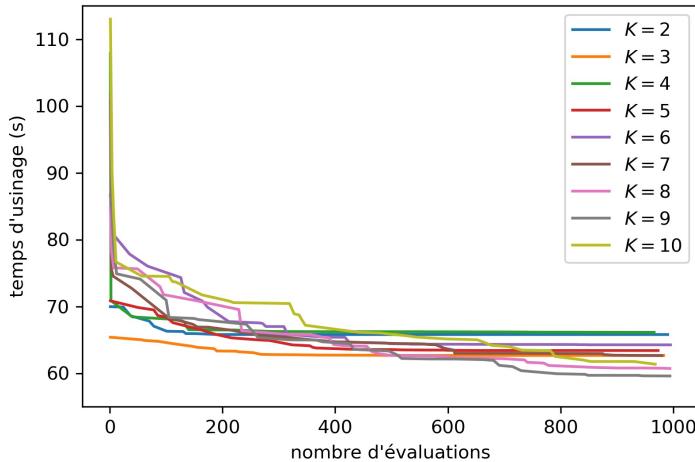


Figure 4.13 – Historique d'optimisation avec l'heuristique basée sur l'ACP pour l'usinage en 3+2-axes de la surface 2

Pour la surface de la pale le nombre de zone est fixé à  $K = 4$ , la figure 4.14 montre le partitionnement de la surface en 4 zones, le temps d'usinage associé au point initial est de  $f_0 = 491.43$  s tandis que le temps d'usinage optimal est de  $f^* = 455.69$  s ce qui donne un gain relatif de  $\eta = 7.3\%$ . L'historique de l'optimisation est affiché dans la figure 4.15.

## 4.6 Discussion et comparaisons

Dans cette section, les résultats obtenus par les modèles d'optimisation en 3-axes, 3+2-axes et l'heuristique basée sur l'ACP en 3+2-axes sont discutés et comparés avec les résultats disponibles dans la littérature sur les mêmes surfaces.

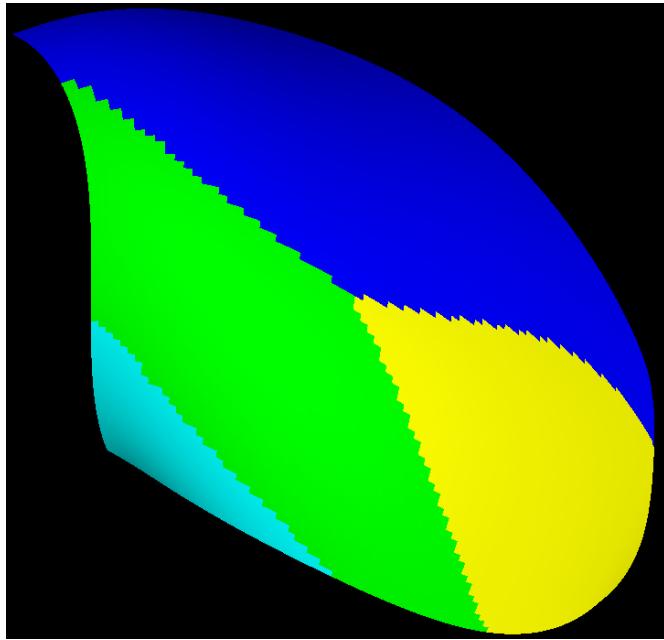


Figure 4.14 – Partitionnement de la surface pale en  $K = 4$  zones en utilisant l’algorithme K-means avec la métrique euclidienne

### 4.6.1 Discussion des résultats obtenus

Pour l’usinage 3 axes et 3+2 axes, le processus d’optimisation a permis d’améliorer les temps d’usinage, sur les deux surfaces de test, de près de 15% par rapport aux valeurs de référence. Ces valeurs de référence sont obtenues en utilisant l’approche de *clustering* avec le même poids ( $\omega_i = 1$ ) pour chaque paramètre du vecteur caractéristique et sans optimisation. Ces valeurs de référence sont assez représentatives des résultats de l’état de l’art. Les gains de temps d’usinage sont encore plus importants pour l’usinage 3+2 axes, ce qui peut s’expliquer par le grand nombre de degrés de liberté.

La *surrogate* 3 est non-seulement la plus rapide, mais aussi, en raison de sa formulation analytique, elle est celle qui donne les meilleurs résultats. Le gain obtenu par la *surrogate* 3 par rapport aux autres cas (y compris le cas sans *surrogate*) est encore plus important pour l’usinage 3+2 axes. En effet, les *surrogates* sont en général plus efficaces pour les problèmes avec un nombre élevé de variables. De plus, les graphiques de l’historique d’optimisation montrent une diminution plus rapide des temps d’usinage en cas d’utilisation de la *surrogate* 3. Ce qui peut être utile lorsque le nombre d’évaluations possible de la boîte noire est assez limité.

Pour ce qui est des résultats de l’heuristique basée sur l’ACP, ils peuvent être comparés avec ceux du modèle d’optimisation en 3+2-axes (sans heuristique). Premièrement, on remarque que les points de départ obtenus par l’heuristique sont nettement meilleurs que ceux du modèle d’optimisation sans heuristique. En effet, les points de départ du modèle sans heuristique correspondent à l’usinage de la surface en 3-axes selon la direction de plus grande pente de chaque zone, tandis qu’avec l’heuristique, la surface est orientée dès le début (et à chaque évaluation) pour que cette direction soit alignée avec la direction principale. Ceci explique les meilleurs points de départ obtenus avec l’heuristique. Ces points

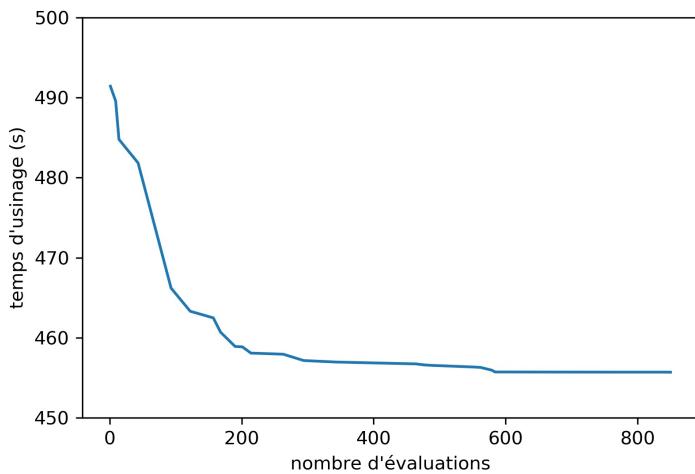


Figure 4.15 – Historique d’optimisation avec l’heuristique basée sur l’ACP pour l’usinage en 3+2-axes de la surface pale

de départs sont encore meilleurs dans le cas de la surface 2, vu qu’elle est partitionnée en zones ayant des directions principales plus marquées (les rapports  $\frac{\lambda_I}{\lambda_{II}}$  sont plus élevés). De même, ils sont meilleurs lorsque le nombre de zones  $K$  est grand, car les zones deviennent de plus en plus homogènes (en terme de direction de plus grande pente).

En second lieu, les temps d’usinage obtenus pour chaque valeur de  $K$ , sont globalement meilleurs lorsque l’heuristique est utilisée. De plus, le temps d’usinage est amélioré par l’heuristique, et ce pour les deux surfaces. Pour la surface 1, on passe de 83.59 s à 77.71 s, soit un gain de 7%. Pour la surface 2, on passe de 71.95 s à 59.58 s ce qui représente un gain de 17.2%. Cette comparaison montre clairement que l’heuristique basée sur l’ACP améliore considérablement le temps d’usinage.

#### 4.6.2 Comparaison avec d’autres articles

Dans cette section, les résultats obtenus avec notre approche en 3-axes et 3+2-axes (sans heuristique) sont comparés aux résultats de la littérature sur les mêmes surfaces de test. Pour la validité de la comparaison, nous nous assurons que les valeurs des différents paramètres sont les mêmes. Les études anciennes citées ci-dessous ne fournissent pas toutes le temps d’usinage ; dans ce cas, seule la longueur du parcours peut être comparée. D’ailleurs, à notre connaissance, ces études sont les seules à fournir suffisamment de données pour rendre les comparaisons possibles.

La surface d’essai de Rubio (surface 2) apparaît dans [26] et [25]. Dans le premier article cité, un algorithme de Clarke and Wright (entre autres) est utilisé pour le partitionnement, tandis que dans le second, une approche AMAS (*Adaptive Multi-Agents Systems*) est adoptée. Dans les deux articles, la stratégie des plans parallèles est appliquée pour l’usinage en 3 axes. Seuls les meilleurs résultats fournis par ces deux articles sont inclus dans la présente comparaison.

Dans l’article [26], une fraise torique  $R = 5$  mm,  $r = 2$  mm est utilisée et la hauteur maximale de crête

est fixée à 0.01 mm. Le meilleur résultat, en termes de longueur de parcours, fourni par cette étude est égal à 4988 mm, tandis que le parcours le plus court que nous avons trouvé, dans des conditions d’usinage identiques, avec 200 évaluations de la boîte noire par valeur de  $k$ ,  $2 \leq k \leq 10$ , est égal à 4018 mm. Le temps de calcul associé aux approches de [26] varie entre 60 s et 80 s alors que notre approche n’a pris que 34 s par valeur de  $k$ .

Dans l’étude [25], une fraise torique de rayons  $R = 3.175$  mm, et  $r = 1$  mm est utilisée et la hauteur maximale de crête est fixée à 0.254 mm. Sous ces conditions, la longueur du parcours d’outil est de 1183 mm. Cependant, la longueur du parcours a été calculée sans tenir compte des connexions entre les parcours adjacents. Notre approche conduit à une longueur de parcours optimale de 995 mm. Le temps de calcul correspondant à notre approche est d’environ 32 s par valeur de  $k$ , alors que l’approche AMAS a pris 160 s.

Par ailleurs, la surface de Rubio est également apparue dans une étude plus récente [73] où une approche de partitionnement basée sur le concept d’intervalles d’usinage efficaces est proposée. La surface a été usinée en 3 axes en utilisant un outil torique avec les mêmes rayons que notre étude. Deux vitesses d’avance ont été utilisées : une faible vitesse d’avance de 1 m/min (pour les matériaux durs et/ou les outils de grande taille) et une vitesse d’avance élevée égale à 10 m/min (pour les alliages légers et/ou les petits outils). La faible avance conduit à un temps d’usinage égal à 311 s et la forte avance à 81 s, tandis que notre approche aboutit à 291 s pour la faible avance et 48 s pour la forte avance.

L’algorithme fuzzy C-means est utilisé dans [63] avec un vecteur de caractéristiques à cinq paramètres ( $u, v, \mathbf{n}$ ) pour partitionner une surface de test pour l’usinage 3+2 axes. Notre approche est testée sur la même surface (celle de [63]), avec la même fraise torique de rayons  $R = 6.7$  mm,  $r = 0.7$  mm, une contrainte de hauteur de crête  $sh = 0.0254$  mm, une vitesse d’avance nominale de  $V = 2$  m/min et un nombre de classes  $K = 4$ . Dans [63], l’usinage de cette surface en 3+2 axes aboutit à un temps d’usinage de 634 s, alors que notre approche conduit à 601 s.

## 4.7 Conclusion et perspectives

Ce chapitre présente une approche intégrale d’optimisation de boîte noire et sans dérivée pour le partitionnement et l’usinage de surfaces gauches. L’étape de partitionnement est pilotée par le nombre de zones et les coefficients de pondération pour les paramètres du *feature-vector*, tandis que l’usinage est géré via les directions d’usinage et également les angles d’orientation dans le cas d’un usinage 3+2 axes.

Quatre fonctions *surrogates* ont été proposées, et leur efficacité a été étudiée. La *surrogate* la plus efficace a été celle formulée analytiquement en approximant chaque zone par une surface plane, résultant de la décomposition de la matrice de covariance de la zone (ACP). L’approche proposée a été évaluée avec deux surfaces test issues de la littérature, et a montré des gains significatifs par rapport à une solution de référence. De plus, une heuristique pour l’usinage 3+2-axes basée sur l’ACP a été introduite.

Les résultats sur les surfaces test montrent le potentiel de cette heuristique à améliorer davantage le temps d’usinage.

Ce travail est encore en cours de développement et plusieurs pistes peuvent être explorées. Tout d’abord, l’étape de partitionnement peut être améliorée en introduisant de nouvelles variables qui peuvent mieux décrire le partitionnement, élargir l’espace de recherche et avoir plus d’impact sur le temps d’usinage. Pour l’étape d’usinage aussi, d’autres stratégies d’usinage peuvent être incluses pour couvrir un plus grand nombre de parcours possibles. Afin d’améliorer le processus d’optimisation, l’utilisation de substituts hybrides peut être envisagée. En outre, une analyse de sensibilité peut s’avérer très utile pour évaluer l’impact des différentes variables sur la fonction-objectif, ce qui peut guider le processus d’optimisation pour se concentrer davantage sur les variables influentes. En outre, plusieurs algorithmes de *clustering*, *feature-vector*, métriques, stratégies d’usinage et types de fraises sont disponibles. Leur choix pourrait être introduit comme variables d’optimisation de catégorie au lieu de considérer d’emblée une seule configuration donnée.

# Conclusion générale et perspectives

Ce travail traite de l'usinage en bout des surfaces gauches en 3-axes et 3+2-axes avec la stratégie plans parallèles. Dans un premier temps, les prérequis à connaître sur le problème d'usinage de surfaces gauches ont été présentés, ainsi que les notions de bases nécessaires à la compréhension de la suite du travail.

Une méthodologie d'approximation de surfaces gauches dans l'espace 3D a été introduite au deuxième chapitre. En se basant sur une analyse en composantes principales, cette méthodologie permet de caractériser une surface gauche (ou une zone de surface gauche) par un ensemble de paramètres numériques. Elle fournit donc une vision globale et quantifiable de cette surface gauche qui pourra par la suite être utilisée par divers algorithmes numériques. Une première application de cette approximation a également été présentée dans ce chapitre. Il s'agit d'une procédure d'aide au choix d'outil pour l'usinage en 3-axes. Cette application présente l'avantage de ne nécessiter aucune simulation numérique de l'usinage pour pouvoir prédire lequel de l'outil torique usinant dans la direction de plus grande pente ou de l'outil sphérique usinant dans la direction principale permettra d'usiner la surface dans le temps le plus court. Par ailleurs, cette aide au choix d'outil s'avère la plus efficace dans les configurations les plus problématiques, c'est à dire pour les surfaces dont la direction de plus grande pente et la direction principale sont perpendiculaires (ou proches de l'être).

Le troisième chapitre traite du partitionnement des surfaces gauches pour l'usinage en 3-axes. Trois algorithmes de *clustering* ont été testés et leurs avantages et inconvénients ont été mis en évidence. En outre, une nouvelle métrique, basée sur l'analyse en composantes principales, est introduite. Le principal intérêt de cette métrique est qu'elle utilise une vue globale des zones sur lesquelles les algorithmes travaillent, alors que les méthodes existantes ne se basent que sur des informations locales. Grâce à ce point de vue plus global, dans la plupart des cas testés, la métrique basée sur l'ACP fournit de meilleurs résultats que les autres métriques. En outre, afin de réduire le temps de calcul des simulations de *clustering*, une mise à jour efficace de la matrice de covariance, utilisant une formule itérative, a été établie. Une nouvelle procédure, permettant une identification très rapide des composantes connexes, a également été développée.

Dans une dernière partie, une approche intégrale d'optimisation boîte noire et sans dérivée pour le partitionnement et l'usinage de surfaces gauches est présentée. L'étape de partitionnement est pilotée par le nombre de zones et les coefficients de pondération pour les paramètres du *feature-vector*, tandis que

## Conclusion générale et perspectives

---

l’usinage est géré via les directions d’usinage ainsi que les angles d’orientation dans le cas d’un usinage 3+2 axes. Quatre fonctions *surrogates* ont été proposées, et leur efficacité a été étudiée. La surrogate la plus efficace a été celle formulée analytiquement en approximant chaque zone par une surface plane, résultant de la décomposition de la matrice de covariance de la zone (ACP). L’approche proposée a été évaluée avec deux surfaces test issues de la littérature, et conduit à des gains significatifs par rapport à une solution de référence. De plus, une heuristique pour l’usinage 3+2-axes basée sur l’ACP a été introduite. Les résultats sur les surfaces test montrent le potentiel de cette heuristique à améliorer davantage le temps d’usinage.

Les prolongements possibles à nos travaux se situent à trois niveaux. Pour l’instant seule l’approximation des surfaces gauches par des plans a été développée, mais il est tout à fait possible d’envisager d’autres entités géométriques. Par exemple, en utilisant la troisième direction propre, on pourra approximer une surface gauche avec des entités géométriques plus complexes que des plans, telles que des paraboloïdes elliptiques.

En ce qui concerne le partitionnement, la métrique basée sur l’ACP pourrait être améliorée. Sa capacité fondamentale à inclure une vue globale de la zone sur laquelle les algorithmes de *clustering* travaillent est un atout essentiel pour définir des métriques plus efficaces. En outre, le choix du point de départ devrait être étudié pour évaluer son influence sur le résultat final de l’usinage. Par ailleurs, seule la stratégie plans parallèles a été utilisée, des techniques de classification supervisée pourraient être utilisées afin de partitionner les surfaces gauches et usiner chaque zone avec la stratégie la plus appropriée.

Pour l’optimisation, plusieurs pistes peuvent être explorées. Tout d’abord, l’étape de partitionnement peut être améliorée en introduisant de nouvelles variables qui peuvent mieux décrire le partitionnement, élargir l’espace de recherche et avoir plus d’impact sur le temps d’usinage. Pour l’étape d’usinage aussi, d’autres stratégies d’usinage peuvent être incluses en tant que variables de catégorie pour couvrir un plus grand nombre de parcours d’outil possibles. Afin d’améliorer le processus d’optimisation, l’utilisation de *surrogates* hybrides peut être envisagée. En outre, une analyse de sensibilité peut s’avérer très utile pour évaluer l’impact de différentes variables sur la fonction-objectif, ce qui peut guider le processus d’optimisation pour se concentrer davantage sur les variables influentes. Enfin, plusieurs algorithmes de *clustering*, *features-vector*, métriques, stratégies d’usinage et types d’outils sont disponibles. Leur choix pourrait être introduit comme variables d’optimisation de catégorie au lieu de considérer une seule configuration.

# Bibliographie

- [1] H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews : Computational Statistics*, 2(4) :433–459, 2010.
- [2] M. Abramson, C. Audet, G. Couture, J. Dennis, Jr., S. Le Digabel, and C. Tribes. *The NOMAD project*. Published : Software available at <https://www.gerad.ca/nomad/>.
- [3] R. K. Agrawal, D. K. Pratihar, and A. Roy Choudhury. Optimization of CNC isoscallop free form surface machining using a genetic algorithm. *International Journal of Machine Tools and Manufacture*, 46(7) :811–819, 2006.
- [4] I. K. Altinel and T. Oncan. A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem. *Journal of the Operational Research Society*, 56(8) :954–961, 2005.
- [5] W. Anotaipaiboon, S. Makhanov, and E. Bohez. Optimal setup for five-axis machining. *International Journal of Machine Tools and Manufacture*, 46(9) :964–977, 2006.
- [6] W. Anotaipaiboon and S. S. Makhanov. Curvilinear space-filling curves for five-axis machining. *Computer-Aided Design*, 40(3) :350–367, Mar. 2008.
- [7] C. Audet, S. L. Digabel, and C. Tribes. NOMAD user guide. Technical Report G-2009-37, Les cahiers du GERAD, 2009.
- [8] C. Audet and W. Hare. *Derivative-Free and Blackbox Optimization*. Springer Series in Operations Research and Financial Engineering. Springer International Publishing, 2017.
- [9] J. Balic and M. Korosec. Intelligent tool path generation for milling of free surfaces using neural networks. *International Journal of Machine Tools and Manufacture*, 42(10) :1171–1179, Aug. 2002.
- [10] R. E. Barnhill, G. Farin, M. Jordan, and B. R. Piper. Surface/surface intersection. *Computer Aided Geometric Design*, 4(1) :3–16, July 1987.
- [11] R. E. Barnhill and S. N. Kersey. A marching method for parametric surface/surface intersection. *Computer Aided Geometric Design*, 7(1) :257–280, June 1990.
- [12] S. Bedi, F. Ismail, M. Mahjoob, and Y. Chen. Toroidal versus ball nose and flat bottom end mills. *The International Journal of Advanced Manufacturing Technology*, 13 :326–332, 1997.

## Bibliographie

---

- [13] M. Berger. *150 ans de Géométrie Riemannienne*. Géométrie au XXe siècle, 1930-2000 : histoire et horizons. Presses inter Polytechnique, 2005.
- [14] C. Bernon, M.-P. Gleizes, S. Peyruqueou, and G. Picard. ADELFE : A Methodology for Adaptive Multi-agent Systems Engineering. In P. Petta, R. Tolksdorf, and F. Zambonelli, editors, *Engineering Societies in the Agents World III*, pages 156–169. Springer, 2003.
- [15] A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17(1) :1–13, 1999.
- [16] K. Castelino, R. D’Souza, and P. K. Wright. Toolpath optimization for minimizing airtime during machining. *Journal of Manufacturing Systems*, 22(3) :173–180, 2003.
- [17] J.-L. Chen, Z. Bai, B. Hamann, and T. J. Ligocki. Normalized-cut algorithm for hierarchical vector field data segmentation. In *Visualization and Data Analysis 2003*, volume 5009, pages 79–90. International Society for Optics and Photonics, June 2003.
- [18] Z. C. Chen, Z. Dong, and G. W. Vickers. Automated surface subdivision and tool path generation for 31212-axis CNC machining of sculptured parts. *Computers in Industry*, 50(3) :319–331, 2003.
- [19] C.-J. Chiou and Y.-S. Lee. A machining potential field approach to tool path generation for multi-axis sculptured surface machining. *Computer-Aided Design*, 34(5) :357–371, Apr. 2002.
- [20] H. Cho, Y. Jun, and M. Yang. Five-axis CNC milling for effective machining of sculptured surfaces. *International Journal of Production Research*, 31(11) :2559–2573, 1993.
- [21] Y.-K. Choi and A. Banerjee. Tool path generation and 3D tolerance analysis for free-form surfaces. *International Journal of Machine Tools and Manufacture*, 47(3-4) :689–696, 2007.
- [22] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.
- [23] S. A. Cook. An overview of computational complexity. In *ACM Turing award lectures*, page 1982. Association for Computing Machinery, New York, NY, USA, Jan. 2007.
- [24] J. J. Cox, Y. Takezaki, H. R. P. Ferguson, K. E. Kohonen, and E. L. Mulkay. Space-filling curves in tool-path applications. *Computer-Aided Design*, 26(3) :215–224, Mar. 1994.
- [25] S. Djebali, A. Perles, S. Lemouzy, S. Segonds, W. Rubio, and J.-M. Redonnet. Milling plan optimization with an emergent problem solving approach. *Computers & Industrial Engineering*, 87 :506–517, 2015.
- [26] S. Djebali, S. Segonds, J. M. Redonnet, and W. Rubio. Using the global optimisation methods to minimise the machining path length of the free-form surfaces in three-axis milling. *International Journal of Production Research*, 53(17) :5296–5309, 2015.
- [27] J. Dong, P. Ferreira, and J. Stori. Feed-rate optimization with jerk constraints for generating minimum-time trajectories. *International Journal of Machine Tools and Manufacture*, 47(12-13) :1941–1955, 2007.

- [28] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2012.
- [29] G. Elber and E. Cohen. Toolpath generation for freeform surface models. *Computer-Aided Design*, 26(6) :490–496, June 1994.
- [30] J.-C. Fiorot and P. Jeannin. *Courbes et surfaces rationnelles*. Masson, Issy-les-Moulineaux (Hauts-de-Seine), France, 1989.
- [31] N. A. Fountas, N. M. Vaxevanidis, C. I. Stergiou, and R. Benhadj-Djilali. Globally optimal tool paths for sculptured surfaces with emphasis to machining error and cutting posture smoothness. *International Journal of Production Research*, 57(17) :5478–5498, 2019.
- [32] H. Gong, F. Z. Fang, X. T. Hu, L.-X. Cao, and J. Liu. Optimization of tool positions locally based on the BCELTP for 5-axis machining of free-form surfaces. *Computer-Aided Design*, 42(6) :558–570, 2010.
- [33] P. J. Gray, F. Ismail, and S. Bedi. Arc-intersect method for 31212-axis tool paths on a 5-axis machine. *International Journal of Machine Tools and Manufacture*, 47(1) :182–190, Jan. 2007.
- [34] J. G. Griffiths. Toolpath based on Hilbert’s curve. *Computer-Aided Design*, 26(11) :839–844, Nov. 1994.
- [35] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao. The connected-component labeling problem : A review of state-of-the-art algorithms. *Pattern Recognition*, 70 :25–43, 2017.
- [36] M. Herraz, J.-M. Redonnet, M. Mongeau, and M. Sbihi. A new method for choosing between ball-end cutter and toroidal cutter when machining free-form surfaces. *The International Journal of Advanced Manufacturing Technology*, 111(5) :1425–1443, 2020.
- [37] C. G. Jensen, W. E. Red, and J. Pi. Tool selection for five-axis curvature matched machining. *Computer-Aided Design*, 34(3) :251–266, Mar. 2002.
- [38] C.-S. Jun, K. Cha, and Y.-S. Lee. Optimizing tool orientations for 5-axis machining by configuration-space search method. *Computer-Aided Design*, 35(6) :549–566, May 2003.
- [39] B. Kim and C. Chu. Effect of cutter mark on surface roughness and scallop height in sculptured surface machining. *Computer-Aided Design*, 26(3) :179–188, Mar. 1994.
- [40] B. H. Kim and B. K. Choi. Guide surface based tool path generation in 3-axis milling : An extension of the guide plane method. *Computer-Aided Design*, 32(3) :191–199, Mar. 2000.
- [41] J.-P. Kruth and P. Klewais. Optimization and Dynamic Adaptation of the Cutter Inclination during Five-Axis Milling of Sculptured Surfaces. *CIRP Annals*, 43(1) :443–448, 1994.
- [42] G. H. Kumazawa, H.-Y. Feng, and M. J. Barakchi Fard. Preferred feed direction field : A new tool path generation method for efficient sculptured surface machining. *Computer-Aided Design*, 67-68 :1–12, 2015.
- [43] A. Lasemi, D. Xue, and P. Gu. Recent development in CNC machining of freeform surfaces : A state-of-the-art review. *Computer-Aided Design*, 42(7) :641–654, July 2010.

## Bibliographie

---

- [44] I. Lazoglu, C. Manav, and Y. Murtezaoglu. Tool path optimization for free form surface machining. *CIRP Annals*, 58(1) :101–104, 2009.
- [45] R.-S. Lin and Y. Koren. Efficient Tool-Path Planning for Machining Free-Form Surfaces. *Journal of Engineering for Industry*, 118(1) :20–28, Feb. 1996.
- [46] X. Liu, Y. Li, and Q. Li. A region-based 3+2-axis machining toolpath generation method for free-form surface. *The International Journal of Advanced Manufacturing Technology*, 97(1-4) :1149–1163, 2018.
- [47] X. Liu, Y. Li, S. Ma, and C.-h. Lee. A tool path generation method for freeform surface machining by introducing the tensor property of machining strip width. *Computer-Aided Design*, 66 :1–13, 2015.
- [48] J. Lu, R. Cheatham, C. G. Jensen, Y. Chen, and B. Bowman. A three-dimensional configuration-space method for 5-axis tessellated surface machining. *International Journal of Computer Integrated Manufacturing*, 21(5) :550–568, July 2008.
- [49] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1 : Statistics*, pages 281–297, Berkeley, CA, USA, 1967. University of California Press.
- [50] S. Makhanov. Optimization and correction of the tool path of the five-axis milling machine : Part 1. Spatial optimization. *Mathematics and Computers in Simulation*, 75(5) :210–230, 2007.
- [51] K. Marciniak. *Geometric Modelling For Numerically Controlled Machining pdf buddy market free download Krzysztof Marciniak*. Oxford University Press Oxford, 1991.
- [52] S. Marshall and J. G. Griffiths. A new cutter-path topology for milling machines. *Computer-Aided Design*, 26(3) :204–214, Mar. 1994.
- [53] G. J. McLachlan. Mahalanobis distance. *Resonance*, 4(6) :20–26, 1999.
- [54] N. R. Pal and D. Chakraborty. Mountain and subtractive clustering method : Improvements and generalizations. *International Journal of Intelligent Systems*, 15(4) :329–341, 2000.
- [55] S. C. Park. Tool-path generation for Z-constant contour machining. *Computer-Aided Design*, 35(1) :27–36, Jan. 2003.
- [56] S. C. Park and B. K. Choi. Tool-path planning for direction-parallel area milling. *Computer-Aided Design*, 32(1) :17–25, Jan. 2000.
- [57] X. Pessoles. *Aide au choix du posage en usinage 5 axes continus par la modélisation du comportement cinématique des machines-outils*. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2010.
- [58] X. Pessoles, Y. Landon, and W. Rubio. Kinematic modelling of a 3-axis NC machine tool in linear and circular interpolation. *The International Journal of Advanced Manufacturing Technology*, 47(5-8, SI) :639–655, Mar. 2010.

- [59] X. Pessoles, J.-M. Redonnet, S. Segonds, and M. Mousseigne. Modelling and optimising the passage of tangency discontinuities in NC linear paths. *International Journal of Advanced Manufacturing Technology*, 58(5-8) :631–642, 2012.
- [60] S. P. Radzevich. Conditions of proper sculptured surface machining. *Computer-Aided Design*, 34(10) :727–740, Sept. 2002.
- [61] J.-M. Redonnet. *Etude globale du positionnement d'un outil pour l'usinage de surfaces gauches sur machines cinq axes et génération de trajectoires*. Ph.d thesis, Université Paul Sabatier, Toulouse 3, France, Jan. 1999.
- [62] J.-M. Redonnet, S. Djebali, S. Segonds, J. Senatore, and W. Rubio. Study of the effective cutter radius for end milling of free-form surfaces using a torus milling cutter. *Computer-Aided Design*, 45(6) :951–962, 2013.
- [63] A. Roman, S. Bedi, and F. Ismail. Three-half and half-axis patch-by-patch NC machining of sculptured surfaces. *The International Journal of Advanced Manufacturing Technology*, 29(5) :524–531, 2006.
- [64] W. Rubio. *Génération de trajectoires du centre de l'outil pour l'usinage de surfaces complexes sur machines à trois et cinq axes*. thesis, Toulouse 3, Jan. 1993.
- [65] J. Senatore. *Analyse qualitative des paramètres influents pour la planification des trajectoires sur surfaces gauches*. PhD thesis, Université Paul Sabatier (Toulouse 3), 2007.
- [66] S. S. Skiena. *The Algorithm Design Manual*. Springer-Verlag, 1998.
- [67] S.-H. Suh and J.-J. Lee. Five-Axis Part Machining With Three-Axis CNC Machine and Indexing Table. *Journal of Manufacturing Science and Engineering*, 120(1) :120–128, Feb. 1998.
- [68] W. Sun, C. Bradley, Y. F. Zhang, and H. T. Loh. Cloud data modelling employing a unified, non-redundant triangular mesh. *Computer-Aided Design*, 33(2) :183–193, Feb. 2001.
- [69] K. Suresh and D. C. H. Yang. Constant Scallop-height Machining of Free-form Surfaces. *Journal of Engineering for Industry*, 116(2) :253–259, May 1994.
- [70] G. J. Szekely and M. L. Rizzo. Hierarchical Clustering via Joint Between-Within Distances : Extending Ward's Minimum Variance Method. *Journal of Classification*, 22(2) :151–183, 2005.
- [71] C. Tournier. *Contribution à la conception des formes complexes : La surface d'usinage en fraisage 5 axes isocristale*. PhD thesis, Cachan, Ecole normale supérieure, 2001.
- [72] G. W. Vickers and K. W. Quan. Ball-Mills Versus End-Mills for Curved Surface Machining. *Journal of Engineering for Industry*, 111(1) :22–26, Feb. 1989.
- [73] D. Vu, F. Monies, S. Segonds, and W. Rubio. Automatic minimal partitioning method guaranteeing machining efficiency of free-form surfaces using a toroidal tool. *The International Journal of Advanced Manufacturing Technology*, 107 :4239–4254, 2020.

## Bibliographie

---

- [74] D. D. Vu. *A new partitioning method for machining free-form surfaces : application to parallel plane strategy in 3 axes and 3+ 2 axes*. PhD thesis, Toulouse 3, 2019.
- [75] H. Wang, H. Chang, R. A. Wysk, and A. Chandawarkar. On the Efficiency of NC Tool Path Planning for Face Milling Operations. *Journal of Engineering for Industry*, 109(4) :370–376, Nov. 1987.
- [76] A. Warkentin, S. Bedi, and F. Ismail. Five-axis milling of spherical surfaces. *International Journal of Machine Tools and Manufacture*, 36(2) :229–243, Feb. 1996.
- [77] A. Warkentin, F. Ismail, and S. Bedi. Multi-point tool positioning strategy for 5-axis machining of sculptured surfaces. *Computer Aided Geometric Design*, 17(1) :83–100, Jan. 2000.
- [78] L. Xu, A. Krzyzak, and E. Oja. Rival penalized competitive learning for clustering analysis, RBF net, and curve detection. *IEEE Transactions on Neural Networks*, 4(4) :636–649, 1993.
- [79] W. Zhao, H. Ma, and Q. He. Parallel K-Means Clustering Based on MapReduce. In M. G. Jaatun, G. Zhao, and C. Rong, editors, *Cloud Computing*, pages 674–679. Springer, 2009.

## **Annexes**



## A Étude du paramètre critique $\mu_c = f(s)$

L'expression du paramètre critique  $s_c = f(\mu)$  a été développée dans la section 2.2.2.3. La fonction réciproque  $\mu_c = f(s)$  est beaucoup moins intéressante d'un point de vue pratique, cependant certains cas d'utilisation peuvent être imaginés, notamment au stade de la conception de la pièce.

### A.1 Expression analytique de $\mu_c$

Pour une pente  $s$  donnée, le facteur de forme critique  $\mu_c$  est défini de telle sorte que les temps d'usinage  $T_1$  et  $T_2$  associés aux deux fraises et directions soient égaux :

$$\begin{aligned}
T_1 &= T_2 \\
\Leftrightarrow & \frac{\sqrt{\mu} \chi}{sod_1} \left( \frac{\sqrt{\chi}}{\sqrt{\mu} V} + 2 \sqrt{\frac{V}{J_{max}}} + 4 \sqrt[3]{\frac{sod_1}{2J_{max}}} \right) \\
&= \frac{\sqrt{\chi}}{\sqrt{\mu} sod_2} \left( \frac{\sqrt{\chi} \mu}{V} + 2 \sqrt{\frac{V}{J_{max}}} + 4 \sqrt[3]{\frac{sod_2}{2J_{max}}} \right) \\
\Leftrightarrow & \frac{\sqrt{\chi} \mu}{V} + 2 \sqrt{\frac{V}{J_{max}}} + 4 \sqrt[3]{\frac{sod_2}{2J_{max}}} \\
&= \mu \frac{sod_2}{sod_1} \left( 2 \sqrt{\frac{V}{J_{max}}} + 4 \sqrt[3]{\frac{sod_1}{2J_{max}}} \right) + \mu \frac{sod_2}{sod_1} \frac{\sqrt{\chi}}{\sqrt{\mu} V}
\end{aligned}$$

Ainsi,  $T_1 = T_2$  est équivalent à

$$\frac{sod_2}{sod_1} \left( 2 \sqrt{\frac{V}{J_{max}}} + 4 \sqrt[3]{\frac{sod_1}{2J_{max}}} \right) \mu + \frac{\sqrt{\chi}}{V} \left( \frac{sod_2}{sod_1} - 1 \right) \sqrt{\mu} - 2 \sqrt{\frac{V}{J_{max}}} - 4 \sqrt[3]{\frac{sod_2}{2J_{max}}} = 0$$

La dernière équation est un polynôme du second degré en  $\sqrt{\mu}$  qui s'écrit :

$$a\mu + b\sqrt{\mu} + c = 0, \tag{7}$$

avec :

$$\begin{aligned}
a &= \frac{sod_2}{sod_1} \left( 2 \sqrt{\frac{V}{J_{max}}} + 4 \sqrt[3]{\frac{sod_1}{2J_{max}}} \right) \\
b &= -\frac{\sqrt{\chi}}{V} \left( 1 - \frac{sod_2}{sod_1} \right)
\end{aligned}$$

$$c = - \left( 2 \sqrt{\frac{V}{J_{max}}} + 4 \sqrt[3]{\frac{sod_2}{2J_{max}}} \right)$$

Il est trivial de vérifier que  $a$  est positif et que  $c$  est négatif. De plus, puisque  $0 < s < \pi/2$ , alors  $0 < \sin s < 1$  et

$$\frac{R-r}{\sin s} + r > R \quad \text{donc} \quad sod_1 > sod_2.$$

Il en résulte que  $b < 0$ . Le discriminant de l'équation (7)  $\Delta = b^2 - 4ac > b^2$  est positif. L'équation (7) admet donc deux solutions dont l'une est négative. En effet :

$$\frac{-b - \sqrt{\Delta}}{2a} = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

est négatif, puisque  $4ac < 0$  et  $2a > 0$ .

Par conséquent, l'expression du facteur de forme critique  $\mu_c(s)$  dépend de  $s$  via  $sod_1$  et est donnée par l'autre solution de l'équation (7). Pour une valeur donnée de la pente  $s$  :

$$\mu_c = \left( \frac{\frac{\sqrt{\chi}}{V} \left( 1 - \frac{sod_2}{sod_1} \right) + \sqrt{\frac{\chi}{V^2} \left( 1 - \frac{sod_2}{sod_1} \right)^2 + 16 \frac{sod_2}{sod_1} \left( \sqrt{\frac{V}{J_{max}}} + 2 \sqrt[3]{\frac{sod_1}{2J_{max}}} \right) \left( \sqrt{\frac{V}{J_{max}}} + 2 \sqrt[3]{\frac{sod_2}{2J_{max}}} \right)}}{4 \frac{sod_2}{sod_1} \left( \sqrt{\frac{V}{J_{max}}} + 2 \sqrt[3]{\frac{sod_1}{2J_{max}}} \right)} \right)^2 \quad (8)$$

Si  $\mu > \mu_c$ , alors  $a\mu + b\sqrt{\mu} + c > 0$ , ce qui signifie que  $T_1 > T_2$ , et donc que l'usinage selon la direction principale avec un outil sphérique est plus efficace. À l'inverse, si  $\mu < \mu_c$ , alors  $a\mu + b\sqrt{\mu} + c < 0$  et  $T_1 < T_2$ . L'usinage selon la direction de la pente avec un outil torique doit alors être privilégié.

## A.2 Analyse numérique du paramètre critique $\mu_c$

Le facteur de forme critique  $\mu_c$  peut être analysé numériquement comme sa réciproque  $s_c$  (section 2.2.3.3). En utilisant la même méthodologie que celle présentée dans la section mentionnée précédemment, on obtient les résultats suivants : La figure A.1 montre l'évolution de  $\mu_c(s)$  et l'erreur relative de la différence entre  $T_1^*(s)$  et  $T_2^*(s)$  calculés numériquement.

Comme dans le cas réciproque, l'erreur est d'environ 5%, ce qui confirme à nouveau la validité des expressions des paramètres critiques. Remarquez que l'erreur est plus élevée pour les petites pentes, pour les mêmes raisons que celles exposées dans la Section 2.2.3.3.

Par conséquent, la procédure proposée peut également être exécutée en utilisant le paramètre de facteur de forme critique au lieu du paramètre de pente critique. Pour ce faire, étant donné un plan et sa pente  $s$ , son facteur de forme  $\mu$  est comparé au facteur de forme critique  $\mu(s)$ . Si  $\mu < \mu_c(s)$ , alors on choisit l'outil torique suivant la direction de la pente. Sinon, l'outil sphérique selon la direction principale est recommandé.

Remarquez que pour les deux cas test présentés dans les sections 2.4.2.1 et 2.4.2.2, la même prédition est fournie, que la procédure basée sur le facteur de forme critique ou celle sur la pente critique soit utilisée.

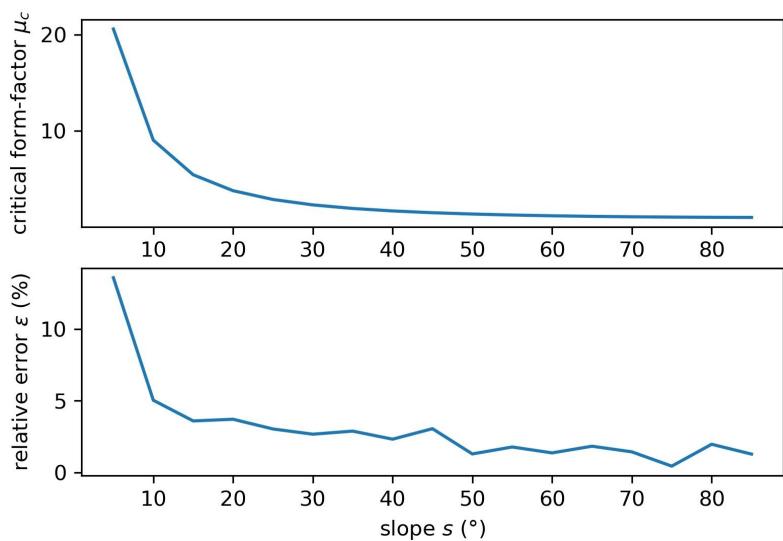


Figure A.1 – Évolution du facteur de forme critique  $\mu_c(s)$  et de l'erreur relative des temps d'usinage  $\varepsilon_c(s)$  en fonction de la pente  $s$



## B Preuve que l'approximation du facteur de forme par les valeurs propres est exacte pour un plan

Soit  $\mathbf{S}$  une surface rectangulaire de largeur  $w$  et de hauteur  $h \leq w$ . Le système de coordonnées est défini de telle sorte que l'origine du repère est le centre de la surface rectangulaire, l'axe  $\mathbf{X}$  est dans le sens de la largeur, et l'axe  $\mathbf{Y}$  dans le sens de la hauteur. Alors, les coordonnées d'un point paramétrique  $(u, v)$  peuvent être écrites comme suit :

$$\mathbf{S}(u, v) = \begin{pmatrix} w(u - \frac{1}{2}) \\ h(v - \frac{1}{2}) \\ 0 \end{pmatrix}$$

L'aire de la surface dans l'espace paramétrique est égale à :

$$|\mathbf{S}| = \int_{u=0}^{u=1} \int_{v=0}^{v=1} dudv = 1.$$

La moyenne de la coordonnée  $S_x$  est ensuite calculée :

$$E(S_x) = \frac{1}{|\mathbf{S}|} \int_{u=0}^{u=1} \int_{v=0}^{v=1} w \left( u - \frac{1}{2} \right) dudv = 0.$$

De la même façon,  $E(S_y) = E(S_z) = 0$ . Les variances peuvent s'écrire :

$$\begin{aligned} \text{Var}(S_x) &= E(S_x^2) - E(S_x)^2 = \int_0^1 \int_0^1 w^2 \left( u - \frac{1}{2} \right)^2 dudv \\ &= \frac{w^2}{3} \left[ \left( u - \frac{1}{2} \right)^3 \right]_0^1 = \frac{w^2}{12}. \end{aligned}$$

et

$$\text{Var}(S_y) = \frac{h^2}{12} \text{ et } \text{Var}(S_z) = 0$$

Les covariances sont égales à 0, puisque  $\text{Cov}(S_x S_z) = \text{Cov}(S_y S_z) = 0$  et :

$$\begin{aligned} \text{Cov}(S_x S_y) &= E(S_x S_y) - E(S_x) E(S_y) \\ &= \int_0^1 \int_0^1 w h \left( u - \frac{1}{2} \right) \left( v - \frac{1}{2} \right) dudv \\ &= \int_0^1 w \left( u - \frac{1}{2} \right) du \int_0^1 h \left( v - \frac{1}{2} \right) dv = 0. \end{aligned}$$

Ainsi, la matrice de covariance est égale à :

$$C = \begin{pmatrix} \frac{w^2}{12} & 0 & 0 \\ 0 & \frac{h^2}{12} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Vu que la matrice  $C$  est diagonale, ses valeurs propres sont :

$$\lambda_I = \frac{w^2}{12} \text{ et } \lambda_{II} = \frac{h^2}{12}$$

## C Preuve de la décroissance des fonctions critiques

Dans ce qui suit, on cherche à prouver que  $s \mapsto \mu_c(s, \chi)$  et  $\mu \mapsto s_c(\mu, \chi)$  sont des fonctions décroissantes, quelque soit la valeur du paramètre  $\chi$ . Par souci de simplicité, des notations plus courtes sont introduites :  $sod_1$ , respectivement  $sod_2$ , est noté  $d_1$ , respectivement  $d_2$ , et  $J_{max}$  est noté  $J$ . En premier lieu, la dérivée du pas transversal  $d_1$  par rapport à la pente  $s$  est calculée :

$$\frac{\partial d_1}{\partial s} = -2 \frac{(R-r) sh \cos s}{\sin^2(s) \sqrt{2 sh \left(\frac{R-r}{\sin s} + r\right) - sh^2}} \leq 0$$

Ensuite, l'équation (8) de l'annexe peut-être réécrite comme :

$$4d_2 \left( \sqrt{\frac{V}{J}} + 2 \sqrt[3]{\frac{d_1}{2J}} \right) \sqrt{\mu_c} = \frac{\sqrt{\chi}}{V} (d_1 - d_2) + \sqrt{\frac{\chi(d_1 - d_2)^2}{V^2} + 16d_2 d_1 \left( \sqrt{\frac{V}{J}} + 2 \sqrt[3]{\frac{d_1}{2J}} \right) \left( \sqrt{\frac{V}{J}} + 2 \sqrt[3]{\frac{d_2}{2J}} \right)}$$

Cette dernière équation est dérivée par rapport à  $d_1$  pour obtenir :

$$4d_2 \left( \sqrt{\frac{V}{J}} + 2 \sqrt[3]{\frac{d_1}{2J}} \right) \frac{\partial \sqrt{\mu_c}}{\partial d_1} + 4d_2 \left( \sqrt{\frac{V}{J}} + \frac{2}{3} \sqrt[3]{\frac{1}{2Jd_1^2}} \right) \sqrt{\mu_c} = \frac{\sqrt{\chi}}{V} + \frac{\frac{\chi(d_1 - d_2)}{V^2} + 8d_2 \left( \sqrt{\frac{V}{J}} + 2 \sqrt[3]{\frac{d_2}{2J}} \right) \left( \sqrt{\frac{V}{J}} + \frac{8}{3} \sqrt[3]{\frac{d_1}{2J}} \right)}{\sqrt{\frac{\chi(d_1 - d_2)^2}{V^2} + 16d_1 d_2 \left( \sqrt{\frac{V}{J}} + 2 \sqrt[3]{\frac{d_1}{2J}} \right) \left( \sqrt{\frac{V}{J}} + 2 \sqrt[3]{\frac{d_2}{2J}} \right)}}$$

Le deuxième terme du membre de droite (la grande fraction) est positif. Par conséquent, on obtient facilement :

$$\frac{4d_2}{\sqrt{\mu_c}} \left( \sqrt{\frac{V}{J}} + 2 \sqrt[3]{\frac{d_1}{2J}} \right) \frac{\partial \sqrt{\mu_c}}{\partial d_1} \geq \frac{\sqrt{\chi}}{V \sqrt{\mu_c}} - 4d_2 \left( \sqrt{\frac{V}{J}} + \frac{2}{3} \sqrt[3]{\frac{1}{2Jd_1^2}} \right)$$

Soit  $w_c$ , respectivement  $h_c$ , la largeur et la hauteur critique, correspondant au facteur de forme critique  $\mu_c(s, \chi)$ , ainsi :

$$\sqrt{\frac{\chi}{\mu_c}} = \sqrt{w_c h_c \frac{h_c}{w_c}} = h_c$$

Par ailleurs :

$$\frac{4d_2}{\sqrt{\mu_c}} \left( \sqrt{\frac{V}{J}} + 2 \sqrt[3]{\frac{d_1}{2J}} \right) \geq 0$$

Il en résulte que le terme  $\frac{\partial \sqrt{\mu_c}}{\partial d_1}$  est positif si :

$$h_c \geq 4d_2 V \left( \sqrt{\frac{V}{J}} + \frac{2}{3} \sqrt[3]{\frac{1}{2Jd_1^2}} \right)$$

La valeur de  $d_1$  dépend de la pente  $s$ , et est minimale lorsque  $d_1 = d_2$ , dans le pire des cas (au sens de l'inégalité précédente). Ainsi, avec les valeurs numériques utilisées pour les paramètres  $R, r, J, V$  et  $sh$ , on obtient que  $h_c \geq 5,3$  mm est une condition suffisante pour que la dérivée de  $\sqrt{\mu_c}$  par rapport à  $d_1$  soit positive. La variation des paramètres d'outil et d'usinage dans des plages de valeurs réalistes conduit à une borne inférieure de  $h_c$  variant entre 3 mm et 10 mm. Cependant, de telles valeurs sont très faibles pour les dimensions des surfaces à usiner en pratique, donc :

$$\frac{\partial \mu_c}{\partial s} = \frac{\partial \mu_c}{\partial \sqrt{\mu_c}} \frac{\partial \sqrt{\mu_c}}{\partial d_1} \frac{\partial d_1}{\partial s} = 2 \sqrt{\mu_c} \frac{\partial \sqrt{\mu_c}}{\partial d_1} \frac{\partial d_1}{\partial s} \leq 0.$$

En conclusion,  $s \mapsto \mu_c(s, \chi)$  est une fonction décroissante. La fonction  $\mu \mapsto s_c(\mu, \chi)$  est aussi décroissante puisqu'elle est la réciproque de  $s \mapsto \mu_c(s, \chi)$ .

## D Preuve des formules utilisées dans le calcul incrémental de la matrice de covariance

La preuve des formules établies dans les équations 3.5 et 3.6 sont données ici.

### D.1 Cas d'ajout d'une maille

Soit  $n \in \mathbb{N}$ , soit  $(x_i)_{1 \leq i \leq n} \in \mathbb{R}^n$  un échantillon statistique de moyenne  $E_n$  et de variance  $\text{Var}_n$ . Soit  $x_{n+1} \in \mathbb{R}$ , on cherche à calculer la moyenne  $E_{n+1}$  et la variance  $\text{Var}_{n+1}$  de l'échantillon augmenté  $(x_i)_{1 \leq i \leq n+1}$  à partir de  $E_n$  et  $\text{Var}_n$  :

$$\begin{aligned} E_{n+1} &= \frac{1}{n+1} \sum_{i=1}^{n+1} x_i = \frac{1}{n+1} \left( \sum_{i=1}^n x_i + x_{n+1} \right) \\ &= \frac{n}{n+1} E_n + \frac{x_{n+1}}{n+1} \end{aligned} \quad (9)$$

Quant à la variance :

$$\text{Var}_n = \frac{1}{n} \sum_{i=1}^n x_i^2 - E_n^2$$

En utilisant la dernière équation, il est évident que :

$$\begin{aligned} \text{Var}_{n+1} &= \frac{1}{n+1} \sum_{i=1}^{n+1} x_i^2 - E_{n+1}^2 = \frac{n}{n+1} \sum_{i=1}^n \frac{x_i^2}{n} + \frac{x_{n+1}^2}{n+1} - E_{n+1}^2 \\ &= \frac{n}{n+1} (\text{Var}_n + E_n^2) + \frac{x_{n+1}^2}{n+1} - \left( \frac{n}{n+1} E_n + \frac{x_{n+1}}{n+1} \right)^2 \end{aligned} \quad (10)$$

Une relation similaire peut être trouvée pour la covariance de deux variables : Soit  $(x_i)_{1 \leq i \leq n} \in \mathbb{R}^n$  et  $(y_i)_{1 \leq i \leq n} \in \mathbb{R}^n$  deux échantillons réels de moyennes  $E_{x,n}$  et  $E_{y,n}$  respectivement, on note  $\text{Cov}_n$  leur covariance :

$$\text{Cov}_n = \frac{1}{n} \sum_{i=1}^n x_i y_i - E_{x,n} E_{y,n}$$

Ainsi, la covariance  $\text{Cov}_{n+1}$  des échantillons augmentés  $(x_i)_{1 \leq i \leq n+1}$  et  $(y_i)_{1 \leq i \leq n+1}$  s'écrit :

$$\begin{aligned} \text{Cov}_{n+1} &= \frac{1}{n+1} \sum_{i=1}^{n+1} x_i y_i - E_{x,n+1} E_{y,n+1} \\ &= \frac{n}{n+1} \sum_{i=1}^n \frac{x_i y_i}{n} + \frac{x_{n+1} y_{n+1}}{n+1} - E_{x,n+1} E_{y,n+1} \\ &= \frac{n}{n+1} (\text{Cov}_n + E_{x,n} E_{y,n}) + \frac{x_{n+1} y_{n+1}}{n+1} \\ &\quad - \left( \frac{n}{n+1} E_{x,n} + \frac{x_{n+1}}{n+1} \right) \left( \frac{n}{n+1} E_{y,n} + \frac{y_{n+1}}{n+1} \right) \end{aligned} \quad (11)$$

D'après les équations scalaires 9, 10 et 11 on peut déduire l'équation matricielle qui permet de mettre à jour la matrice de covariance pour un échantillon vectoriel. Soit  $p \in \mathbb{N}^*$  la dimension des vecteurs,  $n \in \mathbb{N}$  leur nombre,  $(\mathbf{X}_i)_{1 \leq i \leq n} \in (\mathbb{R}^p)^n$  un échantillon de vecteurs de  $\mathbb{R}^p$ . La moyenne de l'échantillon est notée  $\mathbf{E}_n$  tandis que  $\Sigma_n$  désigne sa matrice de covariance :

$$\forall (k, l) \in \{1, \dots, p\}^2, (\Sigma_n)_{kl} = \text{Cov}((\mathbf{X}_i)_k, (\mathbf{X}_i)_l)$$

où  $(\mathbf{X}_i)_k$  désigne l'échantillon scalaire défini par la  $k$ -ième composante des vecteurs  $\mathbf{X}_i$ .

Soit  $\mathbf{X}_{n+1} \in \mathbb{R}^p$  un nouveau vecteur à rajouter à l'échantillon, la matrice de covariance de l'échantillon augmenté  $(\mathbf{X}_i)_{1 \leq i \leq n+1}$  s'écrit :

$$\Sigma_{n+1} = \frac{n}{n+1} (\Sigma_n + \mathbf{E}_n \mathbf{E}_n^T) + \frac{\mathbf{X}_{n+1} \mathbf{X}_{n+1}^T}{n+1} - \left( \frac{n}{n+1} \mathbf{E}_n + \frac{\mathbf{X}_{n+1}}{n+1} \right) \left( \frac{n}{n+1} \mathbf{E}_n + \frac{\mathbf{X}_{n+1}}{n+1} \right)^T \quad (12)$$

L'équation 12 permet donc de mettre à jour la matrice de covariance  $\Sigma_{n+1}$  d'une zone (cluster) qui contient initialement  $n$  mailles dont les feature-vector sont  $(\mathbf{X}_i)_{1 \leq i \leq n}$  (et matrice de covariance  $\Sigma_n$ ), après lui avoir ajouter une nouvelle maille de feature-vector  $\mathbf{X}_{n+1}$ .

## D.2 Cas de retrait d'une maille

Dans ce cas, les zones vides ne sont pas traitées puisqu'il n'y a rien à enlever. De même, les zones contenant une seule maille ne sont pas traitées car celle-ci deviennent vides après retrait et la matrice de covariance est nulle. Dans ce qui suit, nous supposons donc  $n \geq 2$  :

Soit  $(x_i)_{1 \leq i \leq n} \in \mathbb{R}^n$  un échantillon scalaire de moyenne  $\mathbf{E}_n$  et de variance  $\text{Var}_n$ . On cherche à calculer la moyenne  $\mathbf{E}_{n-1}$  et la variance  $\text{Var}_{n-1}$  de l'échantillon réduit  $(x_i)_{1 \leq i \leq n-1}$  à partir de  $\mathbf{E}_n$  et  $\text{Var}_n$  :

$$\begin{aligned} \mathbf{E}_{n-1} &= \frac{1}{n-1} \sum_{i=1}^{n-1} x_i = \frac{1}{n-1} \left( \sum_{i=1}^n x_i - x_n \right) \\ &= \frac{n}{n-1} \mathbf{E}_n - \frac{x_n}{n-1} \end{aligned} \quad (13)$$

Quant à la variance :

$$\begin{aligned} \text{Var}_{n-1} &= \frac{1}{n-1} \sum_{i=1}^{n-1} x_i^2 - \mathbf{E}_{n-1}^2 = \frac{n}{n-1} \sum_{i=1}^n x_i^2 - \frac{x_n^2}{n-1} - \mathbf{E}_{n-1}^2 \\ &= \frac{n}{n-1} (\text{Var}_n + \mathbf{E}_n^2) - \frac{x_n^2}{n-1} - \left( \frac{n}{n-1} \mathbf{E}_n - \frac{x_n}{n-1} \right)^2 \end{aligned} \quad (14)$$

Une relation similaire peut être trouvée pour la covariance de deux variables : Soit  $(x_i)_{1 \leq i \leq n} \in \mathbb{R}^n$  et  $(y_i)_{1 \leq i \leq n} \in \mathbb{R}^n$  deux échantillons scalaires de moyennes  $\mathbf{E}_{x,n}$  et  $\mathbf{E}_{y,n}$  respectivement,  $\text{Cov}_n$  désigne leur

covariance :

$$\text{Cov}_n = \frac{1}{n} \sum_{i=1}^n x_i y_i - \mathbf{E}_{x,n} \mathbf{E}_{y,n}$$

Ainsi, la covariance  $\text{Cov}_{n-1}$  des variables  $(x_i)_{1 \leq i \leq n-1}$  et  $(y_i)_{1 \leq i \leq n-1}$  s'écrit :

$$\begin{aligned} \text{Cov}_{n-1} &= \frac{1}{n-1} \sum_{i=1}^{n-1} x_i y_i - \mathbf{E}_{x,n-1} \mathbf{E}_{y,n-1} \\ &= \frac{n}{n-1} \sum_{i=1}^n \frac{x_i y_i}{n} - \frac{x_n y_n}{n-1} - \mathbf{E}_{x,n-1} \mathbf{E}_{y,n-1} \\ &= \frac{n}{n-1} (\text{Cov}_n + \mathbf{E}_{x,n} \mathbf{E}_{y,n}) - \frac{x_n y_n}{n-1} \\ &\quad - \left( \frac{n}{n-1} \mathbf{E}_{x,n} - \frac{x_n}{n-1} \right) \left( \frac{n}{n-1} \mathbf{E}_{y,n} - \frac{x_n}{n-1} \right) \end{aligned} \quad (15)$$

En utilisant les équations scalaires 13, 14 et 15, on peut trouver l'équation matricielle qui permet de mettre à jour la matrice de covariance pour un échantillon vectoriel. Soit  $p \in \mathbb{N}^*$  la dimension des vecteurs,  $n \geq 2$  la taille de l'échantillon et  $(\mathbf{X}_i)_{1 \leq i \leq n} \in (\mathbb{R}^p)^n$  un échantillon de vecteurs de  $\mathbb{R}^p$ .  $\mathbf{E}_n$  désigne le vecteur moyen de l'échantillon et  $\Sigma_n$  sa matrice de covariance. Ainsi, la matrice de covariance de l'échantillon réduit  $(\mathbf{X}_i)_{1 \leq i \leq n-1}$  s'écrit :

$$\Sigma_{n-1} = \frac{n}{n-1} (\Sigma_n + \mathbf{E}_n \mathbf{E}_n^T) - \frac{\mathbf{X}_n \mathbf{X}_n^T}{n-1} - \left( \frac{n}{n-1} \mathbf{E}_n - \frac{\mathbf{X}_n}{n-1} \right) \left( \frac{n}{n-1} \mathbf{E}_n - \frac{\mathbf{X}_n}{n-1} \right)^T \quad (16)$$

L'équation 16 permet donc de mettre à jour la matrice de covariance  $\Sigma_{n-1}$  d'une zone (cluster) qui contient initialement  $n$  mailles dont les feature-vector sont  $(\mathbf{X}_i)_{1 \leq i \leq n}$  (et matrice de covariance  $\Sigma_n$ ), après lui avoir retirer la  $n$ -ième maille de feature-vector  $\mathbf{X}_n$ .



## E Calcul analytique du temps d'usinage pour la surrogate 3

Pour une zone de surface donnée, soit  $\mathbf{X}_I$  et  $\mathbf{X}_{II}$  la première et seconde directions principales résultantes de la décomposition en valeurs propres de la matrice de covariance des coordonnées des points de la zone,  $\lambda_I$  et  $\lambda_{II}$  leur valeurs propres.

Le plan rectangulaire qui approxime le mieux la zone est généré par  $\mathbf{X}_I$  et  $\mathbf{X}_{II}$ . Sa largeur est selon  $\mathbf{X}_I$  et est noté  $w = \sqrt{12\lambda_I}$ , tandis que sa hauteur est selon  $\mathbf{X}_{II}$  et est noté  $h = \sqrt{12\lambda_{II}}$ .

Soit  $\mathbf{F}_p$  le vecteur d'avance définissant la direction d'usinage, il appartient au plan ( $\mathbf{X}_I\mathbf{X}_{II}$ ), et  $\alpha$  l'angle entre  $\mathbf{F}_p$  et  $\mathbf{X}_I$ . Soit  $\mathbf{F}$  le vecteur d'avance projeté sur le plan ( $\mathbf{XY}$ ), et  $\theta$  l'angle orienté entre  $\mathbf{X}$  et  $\mathbf{F}$ . On a :

$$\mathbf{F} = (\cos \theta, \sin \theta, 0) \text{ et } \mathbf{F}_p = (\cos \theta, \sin \theta, z) \text{ avec } z \text{ inconnu}$$

L'objectif est de calculer le temps d'usinage du plan rectangulaire  $\mathbf{X}_I, \mathbf{X}_{II}$  en fonction de la direction d'usinage  $\theta$ . Mais d'abord, il faut exprimer  $\alpha$  en fonction de  $\theta$  : On sait que  $\mathbf{F}_p \cdot \mathbf{X}_{III} = 0$  où  $\mathbf{X}_{III} = \mathbf{X}_I \wedge \mathbf{X}_{II}$ , ainsi :

$$z = -\frac{\mathbf{X}_{IIIx} \cos \theta + \mathbf{X}_{IIy} \sin \theta}{\mathbf{X}_{IIIz}} \text{ et } \alpha = \arccos(\mathbf{F}_p \cdot \mathbf{X}_I)$$

A présent, le temps d'usinage peut être calculé en fonction de  $\alpha$ . Soit  $p$  le pas transversal entre deux passes adjacentes,  $p$  est constant pour les surfaces planes. Deux cas sont distingués selon la valeur de  $\alpha$ . Pour la cinématique de la machine, nous supposons que la vitesse de consigne  $V$  est atteinte sur chaque trajectoire mais pas sur les connexions entre passes adjacentes. L'accélération maximale  $A$  n'est jamais atteinte.

### — Cas 1 : $\tan \alpha \geq \frac{h}{w}$

La trajectoire d'outil s'étale sur une région au centre à longueur de passes constante et deux régions avec les passes restantes sur les côtés, toutes les passes de la région centrale ont une longueur constante  $l$  qui vérifie :

$$\sin \alpha = \frac{h}{l} \implies l = \frac{h}{\sin \alpha}$$

le nombre de ces passes, noté  $n$ , vérifie :

$$n \delta = w - a$$

où  $\delta$  est la projection de  $p$  sur l'axe  $X_I$  :

$$\cos\left(\frac{\pi}{2} - \alpha\right) = \frac{p}{\delta} \implies \delta = \frac{p}{\sin \alpha}$$

et  $a$  la taille des régions restantes sur les côtés, exprimé comme :

$$\tan \alpha = \frac{h}{a} \implies a = \frac{h}{\tan \alpha}$$

par conséquent :

$$n = \frac{\sin \alpha}{p} \left( w - \frac{h}{\tan \alpha} \right)$$

et le temps d'usinage associé à cette région est :

$$t_1 = n \left( \frac{l}{V} + 2 \sqrt{\frac{V}{J}} + 4 \sqrt[3]{\frac{\delta}{2J}} \right)$$

Pour les régions sur les côtés, il s'agit de calculer des passes avec différentes longueur, le nombre de ces passes est égal à  $\frac{a}{\delta} = \frac{h \cos \alpha}{p}$ , la longueur de la  $i$ -ème passe est  $\frac{ip}{\cos \alpha \sin \alpha}$ . En prenant en compte les connections entre passes, le temps d'usinage associé à chaque régions sur le côté s'écrit :

$$t_2 = \sum_{i=1}^{\frac{a}{\delta}} \frac{ip}{\cos \alpha \sin \alpha V} + 2 \sqrt{\frac{V}{J}} + 2 \sqrt[3]{\frac{\delta}{2J}} + 2 \sqrt[3]{\frac{p}{2J \cos \alpha}}$$

ainsi, le temps d'usinage total est  $t = t_1 + 2t_2$ .

— **Cas 2 :**  $\tan \alpha \leq \frac{h}{w}$

De manière similaire au premier cas, on peut distinguer la région où les passes ont une longueur constante  $l$ , et les deux régions sur les côtés où ce n'est pas le cas. La longueur  $l$  vérifie :

$$\cos \alpha = \frac{w}{l} \implies l = \frac{w}{\cos \alpha}$$

Le nombre des passes, noté  $n$ , vérifie :

$$n \delta = h - b$$

où  $\delta$  est la projection de  $p$  sur l'axe  $X_{II}$  :

$$\cos \alpha = \frac{p}{\delta} \implies \delta = \frac{p}{\cos \alpha}$$

et  $b$  la taille des régions restantes sur les côtés, exprimé comme :

$$\tan \alpha = \frac{b}{w} \implies b = w \tan \alpha$$

Par conséquent :

$$n = \frac{\cos \alpha}{p} (h - w \tan \alpha)$$

et le temps d'usinage associé à cette région est :

$$t_1 = n \left( \frac{l}{V} + 2 \sqrt{\frac{V}{J}} + 4 \sqrt[3]{\frac{\delta}{2J}} \right)$$

Pour les régions sur les côtés, le nombre de ces passes est égal à  $\frac{b}{\delta} = \frac{w \sin \alpha}{p}$ , la longueur de la  $i$ -ème passe

est  $\frac{ip}{\cos \alpha \sin \alpha}$ , en prenant en compte les connections entre passes, le temps d'usinage associé à chaque régions sur les côtés s'écrit :

$$t_2 = \sum_{i=1}^{\frac{b}{\delta}} \frac{ip}{\cos \alpha \sin \alpha V} + 2\sqrt{\frac{V}{J}} + 2\sqrt[3]{\frac{\delta}{2J}} + 2\sqrt[3]{\frac{p}{2J \sin \alpha}}$$

ainsi, le temps d'usinage total est  $t = t_1 + 2t_2$



# Table des figures

1.1	Principe du fraisage en bout . . . . .	12
1.2	Fraisage en bout en 3-axes et en 5-axes . . . . .	13
1.3	Exemple de surface réglée . . . . .	14
1.4	Paramétrage géométrique des outils . . . . .	15
1.5	Les principaux outils utilisés pour le fraisage en bout . . . . .	16
1.6	Comparaison du pas transversal des outils sphérique (gauche) et torique (droite) . . . . .	17
1.7	Hauteur de crête . . . . .	18
1.8	Stratégie d'usinage par surface guide . . . . .	18
1.9	Stratégie d'usinage par plans parallèles . . . . .	19
1.10	Courbes isoparamétriques . . . . .	20
1.11	Génération de trajectoire d'usinage avec des courbes de Hilbert . . . . .	20
1.12	Exemples de coefficients de corrélation linéaire . . . . .	25
1.13	Approximation de la longueur d'usinage en utilisant l'intersection entre les plans d'usinage et les isoparamétriques . . . . .	30
2.1	Usinage d'une surface plane rectangulaire : selon la direction de plus grande pente avec un outil torique (en haut), selon la direction principale avec un outil sphérique (en bas) .	35
2.2	Évolution des temps d'usinage par rapport au facteur de forme $\mu$ pour différentes valeurs de la pente $s$ . . . . .	42
2.3	Erreurs relatives des temps d'usinage $T_1$ et $T_2$ en fonction des paramètres $s$ et $\mu$ . . . . .	43
2.4	Évolution de la pente critique $s_c(\mu)$ et de l'erreur relative du temps d'usinage $\varepsilon(\mu)$ par rapport au facteur de forme $\mu$ . . . . .	44
2.5	Évaluation de la direction principale pour une surface gauche : Principe . . . . .	46
2.6	Surface test 1 avec sa direction principale (violet) et sa direction de plus grande pente (noir) . . . . .	50
2.7	Usinage de la surface test 1 : selon la direction de plus grande pente avec un outil torique (haut), selon la direction principale avec un outil sphérique (bas) . . . . .	51
2.8	Surface test 1 avec sa direction principale (violet) et sa direction de plus grande pente (noir) . . . . .	53

2.9	Usinage de la surface test 2 : selon la direction de plus forte pente avec un outil torique (en haut), selon la direction principale avec un outil sphérique (en bas) . . . . .	54
3.1	Définition des sample-points . . . . .	65
3.2	Un exemple montrant 3 points $P1$ , $P2$ et $P3$ tel que $P3$ est plus proche de $P1$ que de $P2$ en termes de distance euclidienne, alors que $P2$ est plus proche selon la distance géodésique	66
3.3	Les quatres composantes ( $u_i$ , $v_i$ , $s_i$ et $\theta_i$ ) du <i>feature-vector</i> . . . . .	67
3.4	Les différents angles impliqués dans l'évaluation de la distance basée sur l'ACP . . . . .	73
3.5	Surfaces test : surface de Choi (à gauche) et surface de Rubio (à droite) . . . . .	79
3.6	Partitionnement et usinage de la surface de Choi avec l'algorithme <i>K-means</i> et la métrique euclidienne . . . . .	81
3.7	Partitionnement et usinage de la surface de Choi avec l'algorithme <i>K-means</i> et la métrique basée sur l'ACP . . . . .	82
4.1	Angles d'orientation utilisés pour l'optimisation en 3+2-axes . . . . .	92
4.2	Organigramme du modèle d'optimisation en deux étapes . . . . .	93
4.3	Partitionnement des deux surfaces (la numérotation des zones apparaît en blanc) . . . . .	96
4.4	Évolutions des temps d'usinage obtenu par les quatres <i>surrogates</i> comparés au temps d'usinage de la boîte noire pour les zones de la surface 1 . . . . .	97
4.5	Évolutions des temps d'usinage obtenu par les quatre <i>surrogates</i> comparés au temps d'usinage de la boîte noire pour les zones de la surface 2 . . . . .	98
4.6	Historique d'optimisation en 3-axes pour la surface 1 . . . . .	101
4.7	Historique d'optimisation en 3-axes pour la surface 2 . . . . .	102
4.8	Historique d'optimisation en 3+2-axes pour la surface 1 . . . . .	103
4.9	Historique d'optimisation en 3+2-axes pour la surface 2 . . . . .	104
4.10	Définition de l'heuristique, la première rotation autour de $\mathbf{X}_{\text{III}} \times \mathbf{Z}$ aligne $\mathbf{X}_{\text{III}}$ avec $\mathbf{Z}$ , la deuxième rotation autour de $\mathbf{X}_{\text{II}}$ aligne $\mathbf{X}_{\text{I}}$ et $\mathbf{D}$ . . . . .	107
4.11	Exemple montrant que la direction principale n'est pas toujours celle qui optimise la cinématique de l'outil . . . . .	109
4.12	Historique d'optimisation avec l'heuristique basée sur l'ACP pour l'usinage en 3+2-axes de la surface 1 . . . . .	111
4.13	Historique d'optimisation avec l'heuristique basée sur l'ACP pour l'usinage en 3+2-axes de la surface 2 . . . . .	112
4.14	Partitionnement de la surface pale en $K = 4$ zones en utilisant l'algorithme K-means avec la métrique euclidienne . . . . .	113
4.15	Historique d'optimisation avec l'heuristique basée sur l'ACP pour l'usinage en 3+2-axes de la surface pale . . . . .	114

A.1 Évolution du facteur de forme critique $\mu_c(s)$ et de l'erreur relative des temps d'usinage $\varepsilon_c(s)$ en fonction de la pente $s$ . . . . .	129
--	-----



Ce document a pu être réalisé grâce au formateur de texte **L<sup>A</sup>T<sub>E</sub>X**.  
Que les nombreux collaborateurs de ce projet en soient ici remerciés.  
**L<sup>A</sup>T<sub>E</sub>X** est un logiciel libre.

## **Résumé :**

Cette thèse a pour objet la définition de méthodes efficaces pour l'optimisation du temps d'usinage en bout et par zones des surfaces gauches sur les machines 3-axes et 3+2-axes, sous contrainte de qualité (hauteur de crête maximale) en utilisant la stratégie plans parallèles.

Le premier chapitre présente les prérequis à savoir sur le problème d'usinage, ainsi que les notions de bases nécessaires à la compréhension de la suite de ce travail.

Le second chapitre est consacré à une nouvelle méthodologie d'approximation des surfaces gauches par des plans, à l'aide de l'Analyse en Composantes Principales (ACP).

Dans le troisième chapitre, le partitionnement des surfaces gauches est traité. Une nouvelle métrique est introduite permettant prendre en compte le comportement cinématique de la machine, de plus différents algorithmes de partitionnement sont testés et comparés.

Dans le dernier chapitre, l'usinage de surfaces gauches est formulé comme un problème d'optimisation où on cherche à minimiser le temps d'usinage sous contrainte de hauteur de crête. Une formulation originale intégrant les étapes de partitionnement et d'usinage est proposée. Le problème d'optimisation ainsi obtenu est de type boîte noire et résolu par le logiciel NOMAD. Une approche heuristique combinant ACP et optimisation est également proposée. Cette approche permet d'obtenir des gains considérables par rapport aux résultats que nous avons obtenus jusqu'ici et ceux de la littérature.

## **Mots-clés :**

surfaces gauches – usinage en bout – outil torique – partitionnement – classification non-supervisée – analyse en composantes principales – optimisation boîte noire sans dérivées – modèles de substitution.

**Abstract :** *The objective of this thesis is to define efficient methods for optimization of free-form surfaces machining time. Specifically, 3-axis and 3+2-axis by zone end-milling, using parallel-planes strategy are considered under quality constraints (maximum scallop height).*

*The first chapter presents the prerequisites to know about the machining problem, as well as the basic notions necessary to understand the rest of this work.*

*The second chapter is devoted to a new methodology of approximation of free-form surfaces by planes, using Principal Component Analysis (PCA).*

*In the third chapter, the clustering of free-form surfaces is treated. A new metric is introduced to take into account the kinematic behavior of the machine, also, different clustering algorithms are tested and compared.*

*In the last chapter, the machining of free-form surfaces is formulated as an optimization problem where we seek to minimize the machining time under the scallop height constraint. An original formulation integrating the clustering and machining steps is proposed. The resulting optimization problem is a black-box problem solved by the NOMAD software. A heuristic approach combining PCA and optimization is also proposed, this approach allows to obtain considerable gains compared to the results we have obtained so far and those from the literature.*

## **Key words :**

free-form surfaces – end-milling – toroidal cutter – clustering – unsupervised classification – principal components analysis – blackbox and derivative-free optimization – surrogate models.