

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

Université des Sciences et de la Technologie Sâad Dahleb -Blida.

FACULTE DES SCIENCES

DEPARTEMENT D'INFORMATIQUE

Mémoire du Projet de fin d'études

Pour l'obtention du diplôme
d'Ingénieur d'Etat en Informatique

Option: INTELLIGENCE ARTIFICIELLE

SUJET:

*Simulation et vérification des programmes
d'usinage des surfaces gauches sur des
fraiseuses à commande numérique à 3 axes*

Thème proposé par:

Equipe CFAO

Division Productique et Robotique. (CDTA)

Etudié par :

Mr. CHOURAR Ali.

Mr. NEBBAT Hichem.

Encadré par:

Mr. BEY Mohamed: Attaché de Recherche. CDTA

Mr. BENDIFALLAH Hassène: Ingénieur Principal. CDTA

Soutenu le: 13/12/2004

PROMOTION: DECEMBRE 2003/2004.

إهداء

بحمد الله أنجزت هذا المشروع ووضعت بين أيدي أساتذتي الكرام الذين لم يدخروا جهدا في مد يد المساعدة، وإيا هذا المشروع بقلته بعد خطوات كماه لكل فرد في المجتمع التعليمي عامة وجمعية اللاسري خاصة وفعلا إلى الرقي من خطوة إلى أخرى.

فإلى أئبوي الذين لم يدخروا أي أسلوب سوي لدفعي إلى الأمام والتحصيل في ميادين المعرفة. وإلى أئخي منير الذي تدلني بد العود سواء في دراستي الأولى أو خلال سنوات دراستي المتخرج. وإلى أئخي فخر الدين وهو يقدم خدماته لي متكورا في إنجاز هذا المشروع من تجربته في الميدان. وإلى أئخي وزوجها تمنيا لهما حياة سعيدة ووربا محفوا بالورود يفوح منها أرجاء بيتي إلى أئق. أوسع في الحياة.

وإلى أئخي عصام وحمد اللامين تمنيا لهما مستقبل ورايا ناجحا. وإلى كل المعلمين والأساتذة الكرام منذ إورايجي في أول يوم جلست علي مقعد الدراسة ولم أنقل من فصح إلى آخر ومن طور إلى إجلاله وإلا وجدت عمرة جهدهم أطفها مبنها وهي تعكس ما قدموه لي من تربية وعلم.

إلى كل زملائي في مختلف الأطوار والذين نشأت وسطهم فلم يخلوا علي بالحب والدفء الذي لا يضاهيه إلا الدفء اللاسري.

إلى جميع أصدقائي في مختلف مراحل الصبا والطفولة والتعب سواء بحكم الجوار أو الالتقاء إلى مدينة واحدة أو مختلف أطوار الدراسة.

إلى جميع كل من قد نسبت ذكره وخاصة منهم من تعاملت معه من قريب أو بعيد قليلة أو كثيرة، في حياتي العامة أو الخاصة، في غير إنجاز هذا المشروع أو في إنجازه.

7. Types de fraiseuses.....	18
7.1. Fraiseuses verticales.....	18
7.1. Fraiseuses horizontales.....	18
8. Avantages des fraiseuses à commande numérique.....	19
9. Elaboration d'un programme.....	19
10. Définition et structure d'un programme machine.....	20
10.1. Format d'un mot.....	21
10.2. Format d'un bloc.....	22
11. Classification des fonctions préparatoires G et auxiliaires M.....	23
11.1. Classification des fonctions préparatoires G.....	23
11.2. Classification des fonctions auxiliaires M.....	23
12. conclusion.....	24

Chapitre 3 :

Usinage des surfaces gauches

1. Introduction.....	25
2. Différents paramètres d'usinage.....	25
3. Types d'outils (fraises) utilisés pour l'usinage des surfaces gauches.....	25
4. Position de l'outil par rapport à la surface.....	26
4.1. Calcul d'une position de l'outil tangente à la surface à usiner.....	27
5. Erreurs d'usinage.....	27
5.1. Erreur de flèche.....	27
5.2. Erreur de crête.....	28
5.3. Erreur d'interférence.....	28
5.4. Erreur de collision.....	29
6. Méthodes d'usinage des surfaces gauches des courbes isoparamétriques.....	29
6.1. Aller simple (one way).....	29
6.2. Aller retour (Zig Zag).....	30
6.3. Concentrique.....	30
6.4. Spiral in.....	30
6.5. Spiral out.....	31
6.6. Radiale.....	31
7. Autres méthodes d'usinage.....	32
7.1. Usinage par plans parallèles au plan (x, z).....	32
7.2. Trajectoire obtenue par usinage par courbes de niveau.....	32
8. Qualité d'usinage des surfaces.....	32
9. Conclusion.....	33

Chapitre 4 :

Simulation d'usinage des surfaces gauches

1. Introduction.....	34
2. Définition de simulation.....	34
3. Méthodes de simulation.....	34
	34

3.1. Méthode de représentation solide CGS.....	
3.2. Méthode du Z-Buffer.....	35
3.3. Méthode de visualisation graphique.....	36
3.4. Méthode du point vecteur.....	36
4. Triangulation.....	37
4.1. Diagramme de Voronoï.....	38
4.2. Triangulation de Delaunay.....	39
4.3. Algorithme de Delaunay.....	39
5. Algorithme de subdivision.....	40
6. Intersection.....	42
6.1. Plan dans l'espace.....	43
6.2. Droite dans l'espace.....	44
6.3. Cylindre dans l'espace.....	46
6.4. Sphère dans l'espace.....	47
7. Conclusion.....	47

Chapitre 5 :

Conception de l'application logicielle

1. Introduction.....	48
2. Structuration de l'application logicielle.....	48
3. Triangulation.....	50
4. Calcul des vecteurs normaux.....	50
5. Calcul d'intersection entre les normales et le volume balayé par l'outil.....	51
6. Calcul du volume enlevé.....	53
7. Localisation des différentes zones.....	53
8. Spectre d'erreurs.....	54
9. Elimination des points d'interférence.....	55
10. Conclusion.....	55

Chapitre 6 :

Implémentation et résultats

1. Introduction.....	56
2. Environnement de développement.....	56
2.1. Système d'exploitation.....	56
2.2. Langage de programmation.....	56
2.3. Avantages de la Programmation Orientée Objet.....	56
2.4. Programmation avec la bibliothèque OpenGL.....	57
3. Présentation de l'application.....	57
3.1. Interface utilisateur.....	57
3.2. Menu principal.....	58
4. Résultats et validations.....	63
4.1. Exemple d'usinage avec la méthode exacte.....	64
4.2. Exemple d'usinage avec la méthode simplifiée.....	67
	69

5. Conclusion.....

Conclusion générale 70

Annexe 1 71

Annexe 2 75

Références bibliographiques 78

*I*ntroduction générale

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

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

Les pièces de formes gauches (moules, formes aérodynamiques, carrosseries de voitures, formes esthétiques,... etc.) sont devenues, par l'évolution du style et des techniques d'usinage, des pièces courantes de notre vie quotidienne. Comme toute pièce utilisée en mécanique, les pièces de formes gauches sont conçues dans le but d'assurer des fonctions inscrites dans le cahier des charges, par conséquent, ces pièces doivent répondre à des exigences fonctionnelles et/ou de style. Avec l'évolution des machines outils à commande numérique (MOCN), le niveau de qualité de ces pièces a augmenté d'une façon considérable, ce qui impose une attention particulière dans leurs mises en production.

Le travail que nous présentons dans ce mémoire s'insère dans le cadre de développement d'outils de conception et de fabrication des surfaces de formes irrégulières (surfaces gauches) initié par l'équipe Conception et Fabrication Assistées par Ordinateur (CFAO) de la division productique et robotique du Centre de Développement des Technologies Avancées (CDTA).

Ce projet est une continuité des travaux précédents traitant la modélisation et la conception des courbes et des surfaces B-Splines et NURBS ainsi que la génération de ce type de courbes et de surfaces par interpolation et par approximation à partir d'un nuage de points, et l'usinage des surfaces gauches par les courbes isoparamétriques en utilisant les six stratégies d'usinage (One Way, ZigZag, Concentrique, Spiral In, Spiral Out, Radiale), ainsi que l'usinage par les plans parallèles en utilisant les deux stratégies d'usinage ((One Way, ZigZag).

Les programmes d'usinage générés automatiquement pour l'usinage des surfaces gauches contiennent des milliers de lignes d'instructions et qui peuvent contenir des erreurs. Par conséquent, la vérification manuelle de ces programmes est impossible. Les surfaces usinées contiennent toujours des erreurs causées par les différentes approximations faites lors de la génération des trajets outils. Ces erreurs peuvent être soit l'existence des zones non usinés,

soit l'existence des zones où l'usinage est très profond de ce qu'il est demandé, soit les deux en même temps. La simulation est un outil très important permettant de réduire les temps d'usinage effectifs et de valider les trajets d'usinage en détectant les différentes erreurs et leurs emplacements avant de passer à l'usinage effectif. Un important problème pratique est de trouver une réponse à la question suivante : est-ce que les surfaces usinées sont dans l'intervalle de tolérance exigée dans le cahier des charges ? Une fois les zones critiques détectées, il est nécessaire de corriger les erreurs.

L'objectif de notre travail est le développement d'une application logicielle graphique et interactive sous Windows, qui permet, à partir d'un programme d'usinage, de simuler l'opération d'usinage (enlèvement de la matière) de surfaces gauches, et cela quelque soit la méthode d'usinage utilisée (isoparamétriques, plans parallèles, courbes de niveau, ...etc.), afin de détecter les zones en dehors de la tolérance, puis les corriger.

Ce travail permettra d'améliorer la qualité d'usinage, et garantira le non gaspillage du temps et de la matière à utiliser inutilement.

Le présent mémoire est organisé comme suit:

- Dans le premier chapitre, nous allons décrire les méthodes de conception et de représentation des surfaces.
- Dans le chapitre deux nous présentons l'architecture et la programmation des fraiseuses à commande numérique.
- Dans le chapitre trois nous décrivons l'usinage des surfaces gauches par les méthodes des courbes isoparamétriques, plans parallèles et courbes de niveau, ainsi que les erreurs d'usinage possibles.
- Dans le chapitre quatre nous présentons les différentes méthodes de simulation, en se basant sur la méthode « point vecteur ».
- La conception et l'architecture de l'application que nous avons développée sont présentées au chapitre cinq.
- Le chapitre six décrit notre implémentation informatique en mettant l'action sur les différentes fenêtres et actions de notre application ainsi que la présentation de quelques résultats. Enfin, nous terminons ce mémoire par une conclusion générale.

1. Introduction :

La conception des surfaces est une spécialité très importante dans le domaine de la conception assistée par ordinateur (CAO). Cette spécialité nécessite des bases en informatique, des concepts mathématiques très poussés et de l'ingénierie.

Les surfaces complexes sont très utilisées dans la conception des modèles des pièces moulées, la carrosserie des voitures, les coques des navires et dans l'aviation. Ce large champ d'application des surfaces, nécessite des outils et des méthodes adéquats qui facilitent la conception, l'édition et la modification des surfaces d'une manière interactive. Pour cela, les modèles des courbes et des surfaces doivent avoir des interprétations géométriques et permettent de connaître intuitivement la forme finale de la courbe ou de la surface.

Dans ce chapitre, nous allons décrire en premier lieu les deux grandes familles de représentation des surfaces qui sont les surfaces paramétriques et les surfaces non paramétriques, puis nous allons présenter les différentes méthodes de conception des surfaces en CAO, en détaillant les surfaces B-Spline et les surfaces NURBS.

2. Méthodes de représentation des surfaces en CAO :

Les méthodes de représentation des surfaces sont classées en deux grandes classes :

- Les surfaces non paramétriques.
- Les surfaces paramétriques.

2.1. Surfaces non paramétriques [6] :

Elles sont représentées sous deux formes différentes :

- **Forme explicite** : qui est donnée par l'équation algébrique suivante:

$$z = f(x, y) \quad (1.1)$$

Où à chaque x, y correspond une seule valeur de z . Puisque des valeurs multiples ne sont pas permises, cette surface ne peut pas être fermée. De plus il est impossible de représenter toutes les surfaces.

- **Forme implicite** : qui est donnée par l'équation suivante :

$$f(x, y, z) = 0 \quad (1.2)$$

La forme implicite n'a pas la limitation de la représentation explicite, mais la manipulation interactive des surfaces est difficile.

2.2. Surfaces paramétriques :

2.2.1. Définition des surfaces paramétriques [3]:

Une surface paramétrique est représentée par un ensemble de trois coordonnées, chaque coordonnée est une fonction de deux paramètres u et v (figure1), cette surface est donnée par :

$$P(u,v) = (x(u,v), y(u,v), z(u,v)) \quad (1.3)$$

Les paramètres u et v appartiennent en général à l'intervalle $[0,1]$.

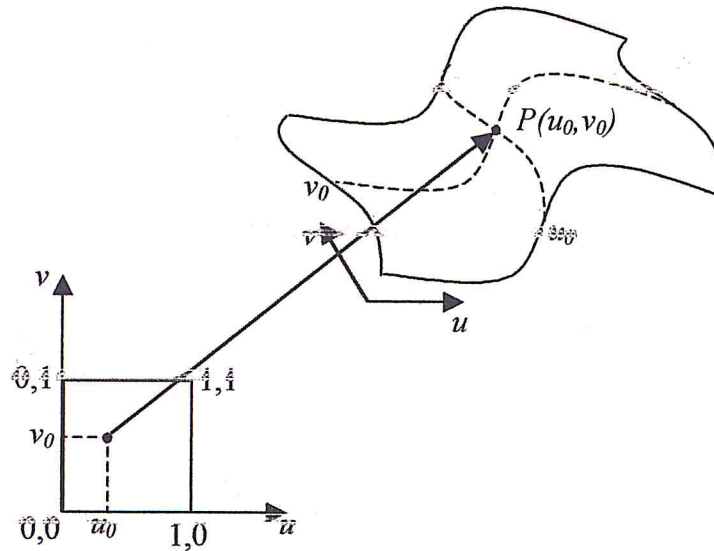


Figure 1 . Détermination d'un point sur une surface paramétrique.

Le point (u_0, v_0) est un point du carré limité par les sommets $(0,0)$, $(0,1)$, $(1,0)$ et $(1,1)$ dans le plan des coordonnées uv . $P(u_0, v_0)$ est le point sur la surface paramétrique qui correspond au point (u_0, v_0) .

2.2.2. Propriétés des surfaces paramétriques :

- Vecteurs tangents et vecteur normal en un point [2]:

Les vecteurs tangents à la surface paramétrique au point $P(u,v)$ sont donnés par [1] :

$$T_u = \frac{\partial P}{\partial u} = \left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u} \right) \quad (1.4)$$

$$T_v = \frac{\partial P}{\partial v} = \left(\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v} \right) \quad (1.5)$$

Le premier vecteur correspond au vecteur tangent dans la direction u , le second vecteur correspond au vecteur tangent dans la direction v . Ces deux vecteurs définissent le plan tangent à la surface au point $P(u,v)$.

Le vecteur normal unitaire $n(u,v)$ à la surface au point $P(u,v)$ est le produit vectoriel des deux vecteurs tangents (figure 2). Ce vecteur est donné par :

$$n = \frac{\frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v}}{\left| \frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v} \right|} \quad (1.6)$$

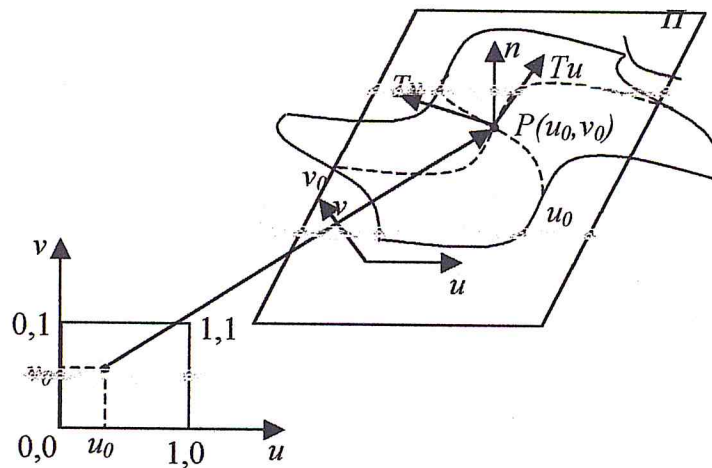


Figure 2. Vecteurs tangents et vecteur normal en un point sur une surface paramétrique.

- **Courbes isoparamétriques [5,6] :**

Une surface paramétrique peut être considérée comme l'union d'un nombre infini de courbes appelées courbes isoparamétriques.

Pour la surface paramétrique $P(u,v)$, si u est fixé à la valeur u_0 et en laissant v varier, cela génère une courbe sur la surface. C'est la courbe isoparamétrique dans la direction v avec $u = u_0$. Similairement, en fixant v à la valeur v_0 et en laissant u varier, on obtient une courbe isoparamétrique dans la direction u avec $v = v_0$ (figure 3).

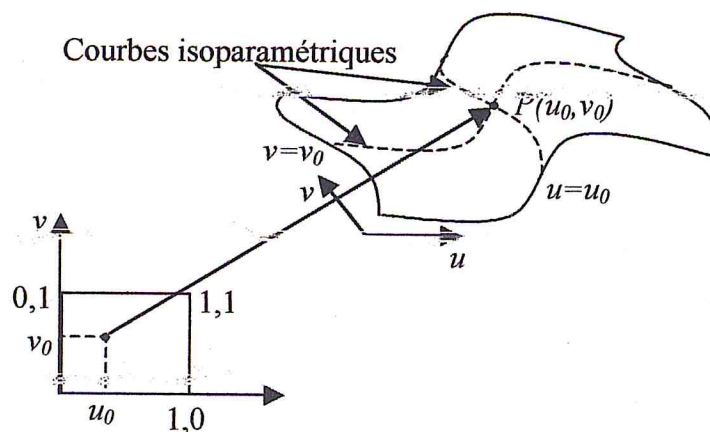


Figure 3. Courbes isoparamétriques d'une surface paramétrique.

• **Courbure d'une courbe tracée sur une surface [5]:**

Une surface paramétrique peut être définie sous deux formes qui sont les formes fondamentales:

La première forme fondamentale est définie comme suit :

$$ds^2 = E du^2 + 2F du dv + G dv^2 \quad (1.7)$$

C'est la formule de la première forme fondamentale qui est appliquée au point $P(u, v)$ de la surface paramétrique S , qui a comme paramètres (du, dv) , où E , F et G sont définies comme suit :

$$E = \frac{\partial P}{\partial u} \times \frac{\partial P}{\partial u} \quad (1.8)$$

$$F = \frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v} \quad (1.9)$$

$$G = \frac{\partial f}{\partial v} \times \frac{\partial f}{\partial v} \quad (1.10)$$

La formule de la deuxième forme fondamentale d'une surface paramétrique S qui a les paramètres (du, dv) est donnée par :

$$\Phi(du, dv) = L du^2 + 2M du dv + N dv^2 \quad (1.11)$$

Où L , M et N sont définies comme suit :

$$L = n \times \frac{\partial^2 P}{\partial u^2} \quad (1.12)$$

$$M = n \times \frac{\partial^2 P}{\partial u \partial v} \quad (1.13)$$

$$N = n \times \frac{\partial^2 P}{\partial v^2} \quad (1.14)$$

La valeur calculée $M^2 - LN$ nous permet d'étudier quelques propriétés de la surface :

- Si $M^2 - LN < 0$: la surface est convexe au point $P(u, v)$;
- Si $M^2 - LN = 0$: la surface possède un point parabolique $P(u, v)$;
- Si $M^2 - LN > 0$: la surface est à courbures opposées au point $P(u, v)$.

Le calcul d'un type donné de courbure d'une surface (une surface à plusieurs types de courbures) permet la détection des anomalies extérieures possibles de cette surface. Ainsi, c'est souvent nécessaire de calculer plusieurs genres de courbures pour obtenir une idée précise de la forme de la surface.

Les courbures principales sont les valeurs (K_1, K_2) , qui sont les deux valeurs extrêmes min et max d'une surface en un point. Ces rayons de courbures principaux sont les solutions de l'équation de second degré suivante :

$$(LN-M^2)R^2 + (2MF-GL-EN)R + EG-F^2 = 0 \quad (1.15)$$

Où L, E, N, G, M et F sont les paramètres utilisés pour le calcul des formes fondamentales.

Si on suppose que les solutions de cette équation sont R_1 et R_2 , alors les courbures principales seront :

$$K_1 = \frac{1}{R_1} \quad (1.16)$$

$$K_2 = \frac{1}{R_2} \quad (1.17)$$

La courbure moyenne H d'une surface S en un point sera calculée comme suit :

$$H = \frac{K_1 + K_2}{2} \quad (1.18)$$

La courbure gaussienne K d'une surface en un point est donnée par :

$$K = K_1 \times K_2 \quad (1.19)$$

Suivant les deux valeurs précédentes de courbures, la surface aura les propriétés suivantes :

- Si $K > 0$ alors x est un point elliptique ;
- Si $K < 0$ alors x est un point hyperbolique ;
- Si $K = 0$ alors x est un point parabolique ;
- Si $K = 0$ et $H = 0$ alors x est un point plat.

On définit un autre type de courbures qui est nommé courbure absolue par :

$$A = |K_1| + |K_2| \quad (1.20)$$

En plus des deux courbures principales, nous pouvons définir une autre paire de courbures. Nous définissons la courbure dans la direction U (pour un v constant), qui correspond à la courbure de la courbe isoparamétrique dans cette direction, pour être courbure

primaire et l'autre dans la direction V (pour u constant), qui correspond à la courbure de la courbe isoparamétrique dans la direction V que nous l'appelons la courbure secondaire.

3. Méthodes de conception des surfaces [7]:

Les méthodes de conception des surfaces sont classées en deux grandes classes:

• ~~Classe des méthodes basées sur les points: dans ces méthodes l'information~~ élémentaire est le point. Les types de surfaces de cette classe sont:

- Interpolation et approximation d'un nuage de points;
- Surfaces de Bézier;
- Surfaces de Bézier Rationnelle;
- Surfaces B-Spline;
- Surfaces NURBS.

• Classe des méthodes basées sur les courbes: dans ces méthodes, la reconnaissance de quelque courbes clés permet la détermination de la surface. Les types de surfaces de cette classe sont:

- Surface réglée;
- Surface de révolution;
- Surface balayée;
- Surface oscillante;
- Surface lissée;
- Surface de Gordon;
- Surface de Coons;
- Surface extrudée.

Le choix de la méthode de représentation d'une surface repose sur plusieurs critères:

- La représentation de toutes les surfaces de la même façon;
- La représentation des surfaces fermées;
- La modification de la surface en temps réel et d'une manière intuitive;
- La translation, la rotation, le changement d'échelle, le changement de repère et toutes autres transformations géométriques linéaires n'influent pas sur la méthode de ~~représentation de la surface;~~
- Peut satisfaire au maximum les contraintes de design et certaines contraintes de continuités;
- La moins coûteuse en source informatique et la plus fidèle à la représentation de l'objet;
- La conception d'une surface ou la modification d'une surface existante est intuitive et flexible.

Les différentes méthodes de conception des surfaces en CAO sont représentées par la figure 4:

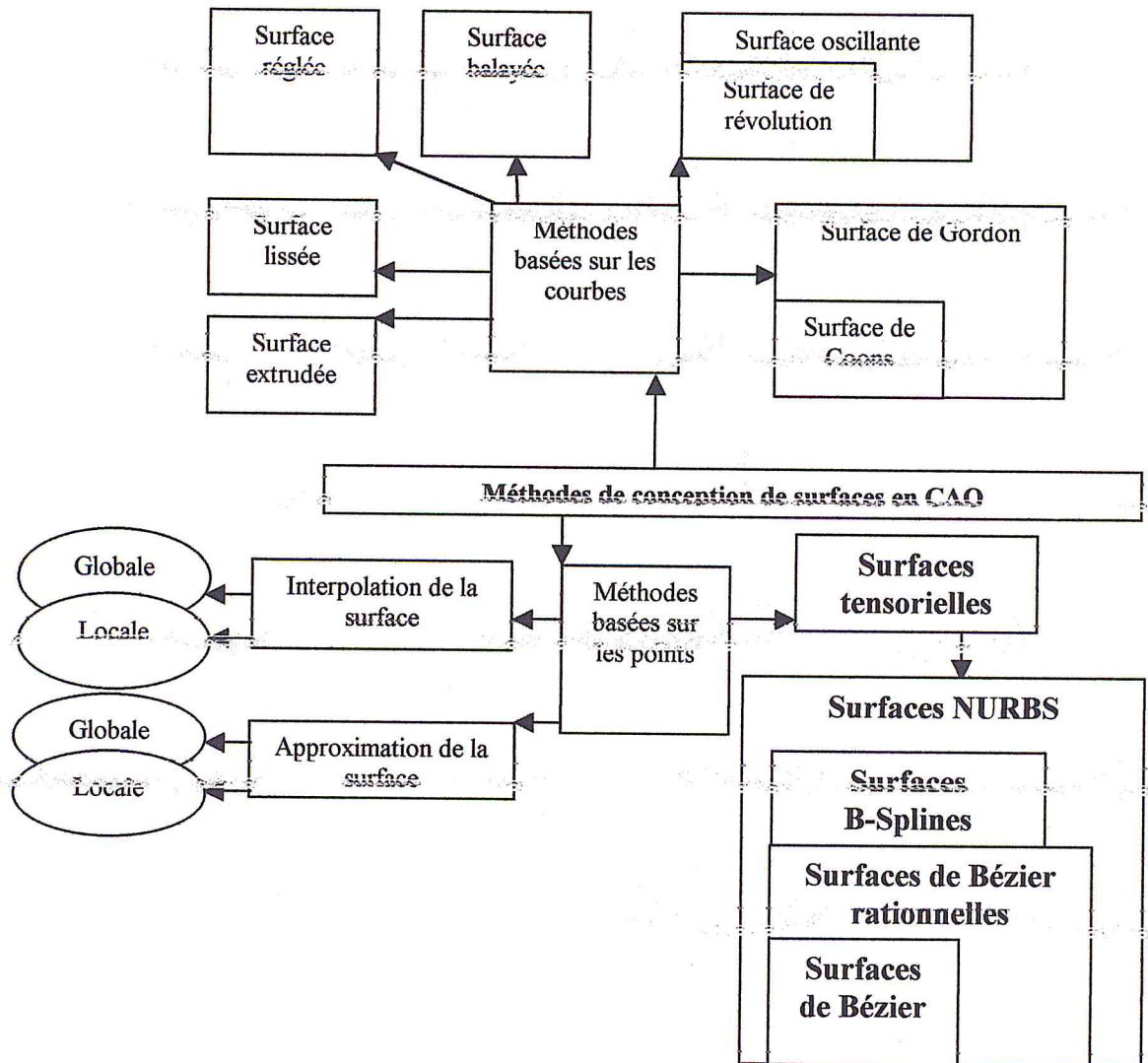


Figure 4 . Les différentes méthodes de conception d'une surface en CAO

4. Surfaces B-Spline et NURBS [2,3]:

Une surface B-Spline ou NURBS est définie par :

- Un réseau de $(m+1) \times (n+1)$ points de contrôle P_{ij} , où $0 \leq i \leq m$ et $0 \leq j \leq n$;
- Un vecteur des nœuds de $h+1$ nœuds dans la direction u , $U = \{0 = u_0, u_1, \dots, u_h = 1\}$;
- Un vecteur des nœuds de $k+1$ nœuds dans la direction v , $V = \{0 = v_0, v_1, \dots, v_k = 1\}$;
- Le degré p de la surface dans la direction u ;
- Le degré q de la surface dans la direction v .

Mais la surface NURBS a de plus :

- Pour chaque point de contrôle est associé un poids $w_{ij} \geq 0$,

La surface B-Spline est définie par la formule suivante :

$$p(u, v) = \sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) P_{i,j} \quad (1.21)$$

La figure 5 montre les différents paramètres d'une surface B-Spline :

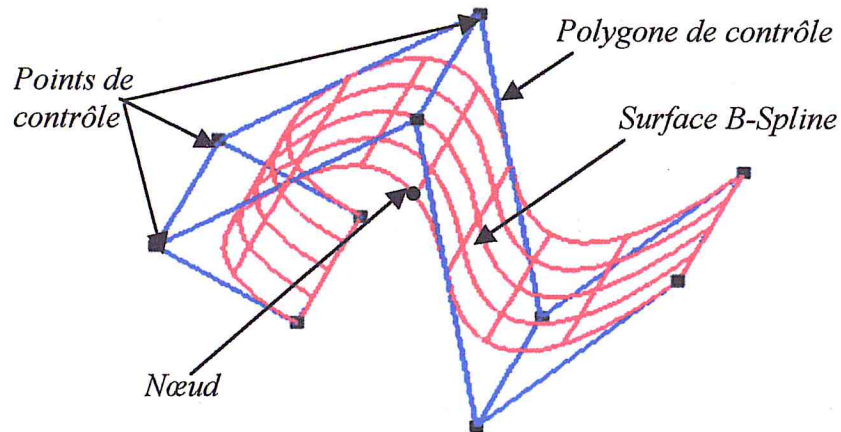


Figure 5. Surface B-spline .

La surface NURBS est donnée par la formule suivante :

$$p(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) w_{i,j}} \quad (1.22)$$

Les différents paramètres d'une surface NURBS sont montrés dans la figure 6:

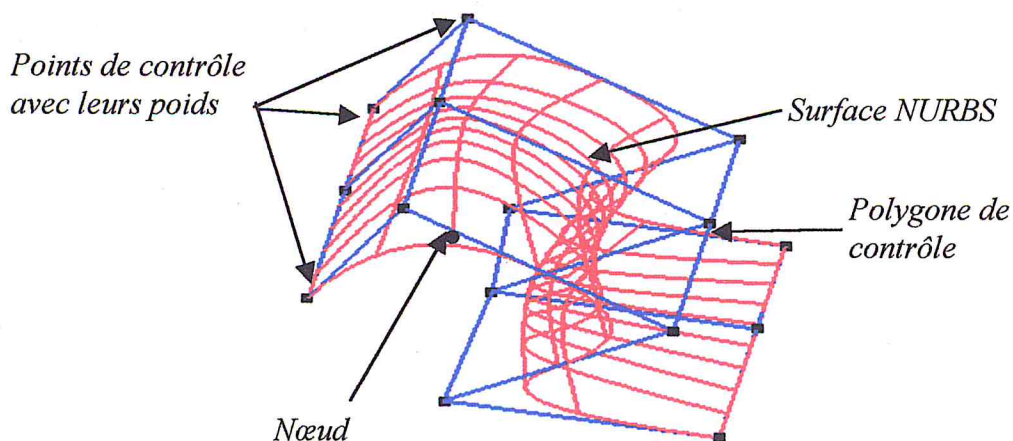


Figure 6. Surface NURBS .

Où $N_{i,p}(u)$ et $N_{j,q}(v)$ sont les fonctions base B-Spline de degré p et q respectivement. Ces fonctions sont données par :

$$N_{i,0}(u) = \begin{cases} 1 & \text{si } u_i \leq u < u_{i+1} \\ 0 & \text{sinon} \end{cases} \quad (1.23)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (1.24)$$

$$N_{j,0}(v) = \begin{cases} 1 & \text{si } v_j \leq v < v_{j+1} \\ 0 & \text{sinon} \end{cases} \quad (1.25)$$

$$N_{j,q}(v) = \frac{v - v_j}{v_{j+p} - v_j} N_{j,q-1}(v) + \frac{v_{j+q+1} - v}{v_{j+q+1} - v_{j+1}} N_{j+1,q-1}(v) \quad (1.26)$$

- **Surface pincée, fermée et ouverte [5,6]:**

On peut avoir trois formes possibles de surface B-Spline ou NURBS dans chaque direction (pincée, ouverte ou fermée).

- Si le vecteur nodal dans une direction n'a aucune structure particulière, la surface produite est ouverte dans cette direction.
- La surface est pincée dans la direction U (respectivement dans la direction V) si le premier et le dernier nœud est répété $p+1$ (respectivement $q+1$) fois.
- En répétant quelques nœuds et points de contrôle, la surface produite peut être fermée, et dans ce cas les premiers et derniers points de contrôle (dans la direction où la surface est fermée) sont égaux.

- **Propriétés des surfaces B-Splines et NURBS [2]:**

Les propriétés d'une surface B-Spline peuvent être vues comme une extension des fonctions base B-Spline. Les propriétés sont les suivantes:

- La surface B-Spline est une surface polynomiale de degré (p,q) car $N_{i,p}(u)$ et $N_{j,q}(v)$ sont des fonctions polynomiales de degré p et de degré q .
- $P(u, v)$ a une continuité C^{p-s} (resp. C^{q-t}) dans la direction u (resp. la direction v) si u (resp. v) est un nœud de multiplicité s (resp. t).
- La partition d'unité : la somme de toutes les fonctions $N_{i,p}(u)$ et $N_{j,q}(v)$ est égale à 1, c'est-à-dire:

$$\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) \cdot N_{j,q}(v) = 1 \quad (1.27)$$

- La non négativité pour tout i, p, q, u et v , la fonction $N_{i,p}(u)$ et $N_{j,q}(v)$ est non négative.
- Le schéma de modification locale : $N_{i,p}(u), N_{j,q}(v)$ est non nulle si (u,v) est dans l'intervalle $[u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}]$ et nulle ailleurs.
- L'invariance affine : pour appliquer une transformation affine à une surface B-Spline, on peut appliquer cette transformation à tous les points de contrôle et la

surface définie par les points transformés est la même que celle obtenue en faisant la même transformation pour tous les points de la surface d'origine.

• **Propriété de l'enveloppe convexe** : si (u, v) est dans l'intervalle $[u_i, u_{i+1}] \times [v_j, v_{j+1}]$, alors $P(u, v)$ est contenue dans l'enveloppe convexe définie par les points de contrôle $P_{s,t}$ tel que $i-p \leq s \leq i$ et $j-q \leq t \leq j$.

• Si $m-p, n-q$ et les premières $p+1$ (resp. $q+1$) valeurs nodales sont égales à 0 et les dernières $p+1$ (resp. $q+1$) valeurs nodales sont égales à 1, alors la surface B-Spline devient une surface de Bézier.

• Toute courbe isoparamétrique sur la surface B-Spline est une courbe B-Spline définie par un ensemble de points de contrôle.

• Les courbes frontières pour $u=0, u=1, v=0$ et $v=1$ sont définies par les points de contrôle suivants :

Pour la courbe $P(0, v) : P_{0,0}, P_{1,0}, \dots, P_{m,0}$;

Pour la courbe $P(1, v) : P_{0,n}, P_{1,n}, \dots, P_{m,n}$;

Pour la courbe $P(u, 0) : P_{0,0}, P_{0,1}, \dots, P_{0,n}$;

Pour la courbe $P(u, 1) : P_{m,0}, P_{m,1}, \dots, P_{m,n}$.

• Les deux identités fondamentales doivent être satisfaites :

$$\begin{cases} \hat{n} = m + p + 1 \\ et \\ k = n + q + 1 \end{cases} \quad (1.28)$$

• La surface est indépendante du système de coordonnées.

• Pour une surface B-Spline pincée, la surface passe par les quatre points: $p_{0,0}, p_{m,0}, p_{0,n}$ et $p_{m,n}$ et elle est tangente aux huit segments du réseau de contrôle.

En plus des propriétés des surfaces B-Spline les surfaces NURBS possèdent les propriétés suivantes:

• A chaque fois que la valeur du poids W_{ij} du point de contrôle P_{ij} augmente (resp. diminue) une portion de la surface est approchée (resp. éloignée) de ce point.

• Les surfaces B-Spline et les surfaces de Béziérs sont des cas particuliers des surfaces NURBS. Si tous les poids sont égaux, la surface NURBS devient une surface B-Spline, et si de plus $m=p$ et $n=q, u=\{0,0,\dots, 1,1,\dots,1\}$ et $v=\{0,0,\dots, 1,1,\dots,1\}$, alors une surface NURBS devient une surface de Bézier.

• Toute courbe isoparamétrique sur la surface NURBS est une courbe NURBS définie par un ensemble de points de contrôle.

• Une surface NURBS est une surface rationnelle, donc, elle permet de représenter toutes les coniques.

• Invariance projective : si une transformation projective est appliquée à une surface NURBS, alors la surface projetée est construite à partir de la projection des points de contrôle.

5. Conclusion :

D'après l'étude précédente on trouve que les surfaces les plus puissantes qui répondent plus aux besoins de flexibilité, modification locale, représentation des surfaces qui ne peuvent pas être représentées par d'autres types de surfaces, et d'autres avantages, sont les surfaces NURBS qui englobent tout l'ensemble des surfaces étudiées dans ce chapitre. Dans ce qui suit on travail avec ce type de surfaces.

1. Introduction :

La machine-outil à commande numérique (MOCN) représente le moyen de production le plus important des pièces mécaniques. Elle assure la réalisation automatisée des pièces en spécifiant les mouvements nécessaires dans un programme.

La machine-outil a pour but de réaliser physiquement les mouvements nécessaires à l'obtention d'une surface par enlèvement de matière. Elle réalise le mouvement de coupe et le mouvement d'avance de l'outil par rapport à la pièce. De plus, elle doit permettre l'obtention de pièces en respectant les spécifications fonctionnelles.

Ces mouvements sont indiqués par le G-code ou le programme d'usinage qui suit une syntaxe bien définie. Le programme est un ensemble d'instructions, et chaque instruction est composée d'un ensemble de mots.

Parmi les machines-outils à commande numérique, nous pouvons citer les perceuses, les tours, les fraiseuses...

Les surfaces gauches sont usinées sur des fraiseuses à commande numérique, et ces dernières machines-outils sont présentés dans ce chapitre.

2. Composants d'une fraiseuse à commande numérique [10] :

La fraiseuse à commande numérique est composée des éléments suivants :

- La machine outil ;
- Le directeur de commande numérique (DCN) ;
- L'armoire électronique ;
- Le pupitre de commande.

2.1. Machine outil :

La machine outil contient la structure mécanique de la machine. Les mouvements sont commandés par des moteurs et les déplacements sont contrôlés avec des capteurs de mesure.

2.2. Directeur de commande numérique (DCN):

Un ensemble composé d'éléments électroniques construit un formalisme appelé le directeur de commande numérique (DCN). Son rôle principal est de traduire et d'exécuter le programme pièce écrit dans un format de programmation.

Le directeur de commande numérique assure les fonctions suivantes :

- Commande les déplacements du chariot en sens et en vitesse de déplacement programmée ;
- Indication du type de déplacement (rectiligne, circulaire) ;
- Indication des coordonnées du point à atteindre ;
- Connaissance de la position actuelle de la fraise par rapport à la pièce ;

- Calcul des points intermédiaires pour atteindre le point défini dans le bloc instruction ;
- Commande des axes de la machine ;
- Gestion des programmes ;
- Visualisation.

2.3. Armoire électronique :

L'armoire électronique sert de relais entre la machine et le DCN. Elle contient des câbles, des amplificateurs et des fusibles... etc.

2.4. Pupitre de commande :

Le pupitre de commande sert à dialoguer avec le DCN et envoi des ordres de commande codés, il possède des touches sensibles, ainsi qu'un écran graphique qui sert à visualiser par exemple le programme, ou le profil fini de la pièce et la trajectoire des outils.

3. Caractéristiques des fraiseuses à commande numérique [10] :

Les fraiseuses à commande numérique sont caractérisées par :

- Le mode de fonctionnement de la fraiseuse ;
- Le nombre d'axes de la fraiseuse ;
- Le mode de fonctionnement du système de mesure ;
- Le mode d'entrée des informations.

4. Axes de déplacements [8]:

Un système de coordonnées permet de repérer les positions et les déplacements d'un objet par rapport à un point origine. Un système de coordonnées cartésiennes rectangulaire est un trièdre de sens direct constitué de trois axes linéaires X , Y et Z auxquels sont associés trois axes rotatifs A , B et C (figure1).

La règle des trois doigts permet de retrouver facilement l'orientation des axes X , Y et Z .

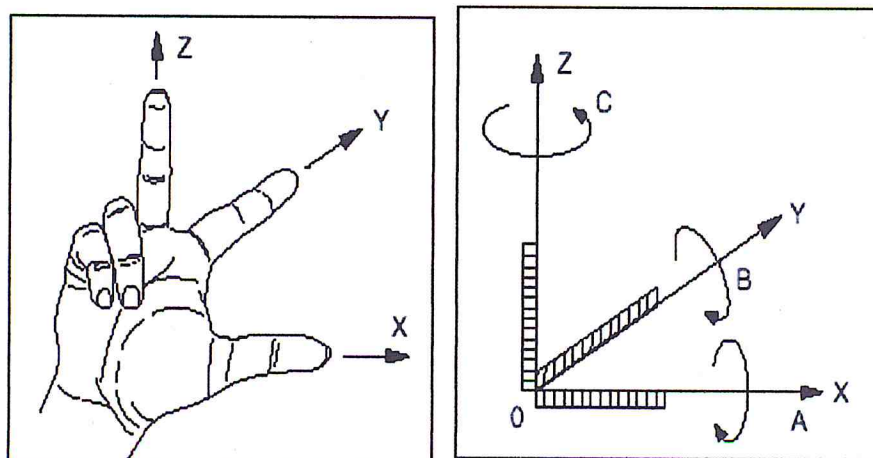


Figure 1. Orientation des axes.

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

- L'axe Z : correspond à l'axe de la broche. Un déplacement dans le sens Z positif accroît la distance entre la pièce et l'outil. Le sens positif de rotation de la broche est celui des aiguilles d'une montre.
- L'axe X : correspond à l'axe ayant le plus grand déplacement, qui est en général horizontal, de sens positif est celui qui correspond à l'accroissement de la distance entre l'outil et l'axe de rotation, X est perpendiculaire à l'axe Z .
- L'axe Y : forme avec les axes X et Z un trièdre de sens direct.

5. Machine-outil à trois axes et à cinq axes [10] :

La combinaison des différents axes donne plusieurs types de fraiseuses. Par exemple, les fraiseuses à 3 axes ou à 5 axes. Chacune de ces fraiseuses a des opérations d'usinage supplémentaires. Chaque axe représente un degré de liberté (figure2).

Une fraiseuse 3 axes possède trois déplacements linéaires suivant les trois axes x , y et z .

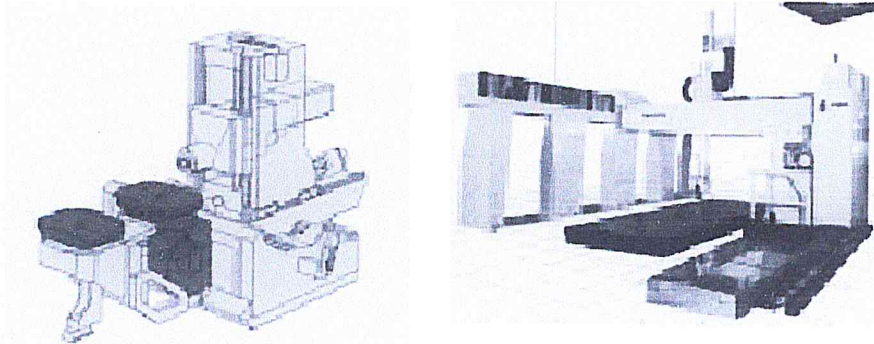


Figure 2. Structure des machines à 3 et à 5 axes.

Par définition, une fraiseuse à cinq axes offre deux degrés de liberté supplémentaires par rapport à une fraiseuse à trois axes permettant d'orienter l'axe de l'outil.

En usinage à trois axes, il existe une unique position de l'outil tangent à la surface en une position donnée, et en usinage à cinq axes, il existe une infinité de positions de l'outil tangent à la surface au point de contact (figure3).

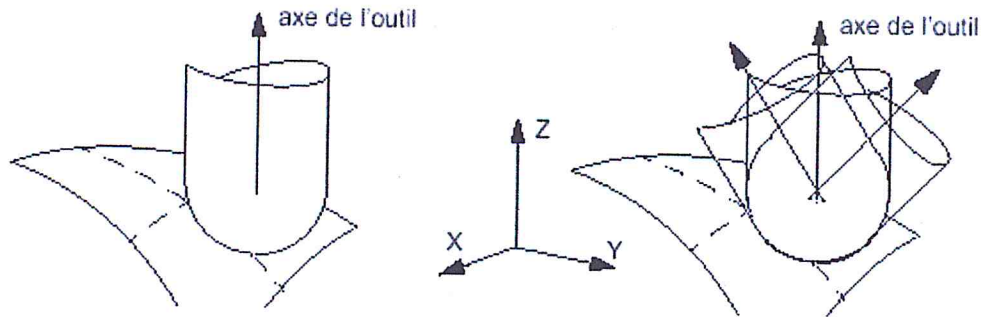


Figure 3 . Comparaison entre l'usinage à trois axes et l'usinage à cinq axes.

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

- Dans le cadre de l'usinage en bout, la surface est réalisée par balayage de l'outil.
- Dans le cadre de l'usinage avec le flanc de l'outil, le contact entre l'outil et la surface est une ligne le long de l'outil. Ici toute la partie coupante est en contact avec la surface. Cette technique ne peut être utilisée que pour des types de surfaces particulières. Les surfaces les plus simples à usiner avec des flancs d'outil rectilignes sont les surfaces développables. Les autres surfaces peuvent être usinées par cette technique, si on autorise une certaine erreur au niveau de contact entre la surface et l'outil.

Les libertés d'orientation de l'outil permettent :

- d'usiner des surfaces en contre-dépouille, par orientation de l'axe de l'outil de façon à éliminer les collisions entre l'outil et la surface.
- d'optimiser la vitesse de coupe dans le cas de l'utilisation d'un outil hémisphérique; par rotation de l'axe de l'outil.
- de changer la géométrie d'outil. Cette technique autorise l'usage d'outils toriques, sans risque de talonnement du fond de l'outil sur la surface.

Par contre, l'ajout de deux angles d'orientation pose un problème à la construction de la trajectoire d'usinage. La recherche de l'orientation de l'outil est un problème d'optimisation difficile à mettre en oeuvre pour des surfaces ayant de fortes variations de courbure. Les risques de collision sont importants. Il est nécessaire de tester le risque de collision sur l'ensemble de l'outil, partie coupante et corps d'outil.

6. Définition des dimensions d'outils (jauges) [10]:

La jauge d'outil est une distance de l'arête coupante de l'outil par rapport au point de référence broche. Pour une fraise, les valeurs à connaître sont la longueur, le rayon de la fraise et le rayon de la partie torique pour une fraise torique (figure4).

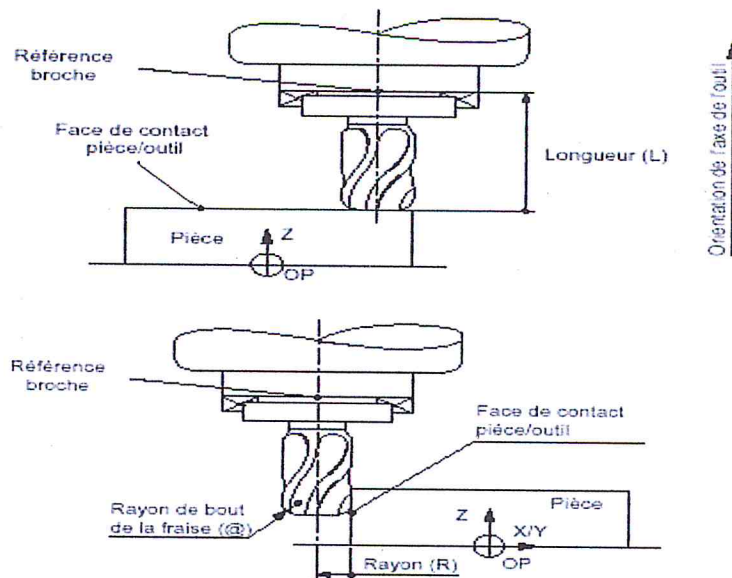


Figure 4 . Les jauges d'outil

7. Types de fraiseuses [6]:

7.1. Fraiseuses verticales :

Les fraiseuses verticales sont caractérisées par (Voir figure 5) :

- L'axe Z est parallèle à celui de la broche et il est vertical ;
- L'axe X est horizontal et il correspond au déplacement longitudinal de la table ;
- L'axe Y est perpendiculaire aux autres axes et il correspond au déplacement transversal ;

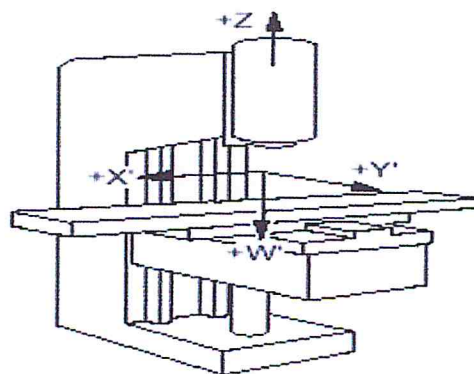


Figure 5. Une fraiseuse verticale.

7.2. Fraiseuses horizontales :

Le système d'axes est différent pour une fraiseuse horizontale, toute fois, l'axe Z est toujours celui de la broche, seulement que la broche est parallèle à la table (figure 6).

Figure 6 . Une fraiseuse horizontale.

8. Avantages des fraiseuses à commande numérique [11]:

Les fraiseuses à commande numérique ont des possibilités impressionnantes d'usinage par rapport aux fraiseuses conventionnelles grâce à :

- La rapidité de lecture et de traitement des informations ;
- La rapidité de réponse des divers organes mobiles ;
- La conception et les vitesses de déplacement ;
- La précision que confère son architecture ;
- La possibilité de réalisation d'un profil de pièce ;
- La souplesse de changement de type de pièce et à la polyvalence des travaux permis ;
- Les cycles programmés ;
- Les modifications des dimensions, des paramètres de coupe sont possible en cours d'usinage.

9. Elaboration d'un programme [10] :

Le programme pièce peut être créé par programmation traditionnelle ou par l'intermédiaire d'un système CFAO(figure 7).

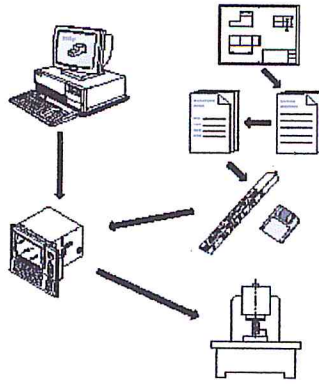


Figure 7. *Elaboration d'un programme.*

Les étapes à suivre pour l'usinage d'une pièce sur une fraiseuse numérique sont les suivantes :

- Etudier les dessins de définition;
- Analyser le travail à effectuer ;
- Choisir les outillages de coupe (fraises) et de contrôle;
- Définir le montage de la pièce ;
- ≡ Déterminer l'origine du programme (OP);
- Numérotter et coter les points du profil;
- Imaginer la trajectoire de la fraise (ordre chronologique);
- Ecrire le programme;
- Mettre la machine sous tension;
- ≡ Charger le programme dans le DCN;
- Actionner la puissance;
- Introduire les dimensions des outils (jauges) dans le DCN;
- Tester le programme mode rapide sans outil;
- Activer le bouton départ du cycle;
- ≡ Usiner la pièce;
- Contrôler la pièce usinée.

10. Définition et structure d'un programme machine [8,9] :

Un programme est une liste d'instructions et de données à transmettre au système. La création d'un programme composé de blocs et de mots est soumise à des règles de structure, de syntaxe ou de format.

Les programmes de production des pièces sont rangés dans des blocs d'informations codées. Ce code est interprété par le contrôleur de la machine outil pour le contrôle de la table, le mouvement des axes et toutes les fonctions auxiliaires.

Ces programmes sont introduits dans les machines à travers le standard RS-232 de la liaison en série et sont enregistrés dans la mémoire du contrôleur.

10.1. Format d'un mot [9] :

Un mot définit une instruction ou une donnée à transmettre au système de commande. Il existe deux types de mot qui sont :

- Mots définissant des dimensions;
- Mots définissant des fonctions.

Le format de mot définit les caractéristiques particulières de chaque mot codé employé en programmation.

Le mot a la forme suivante (Voir figure 8):

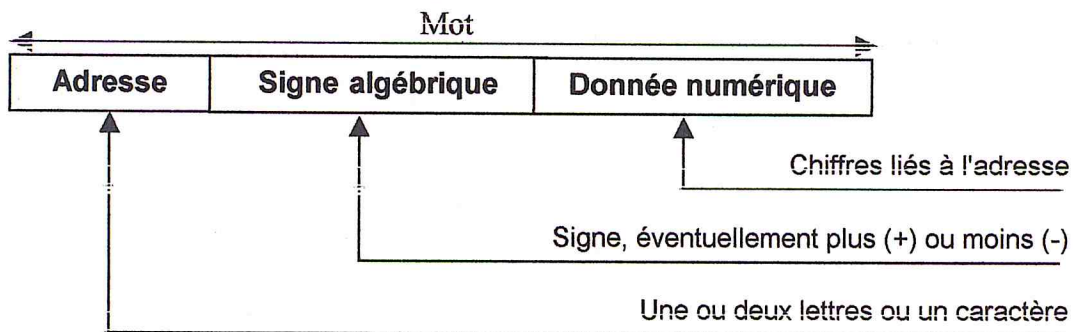


Figure 8. Format d'un mot.

Pour les mots définissant une dimension, le point décimal est généralement explicite, il sépare l'unité de la partie décimale de l'unité.

Le format des mots de dimensions est conditionné par le choix des unités internes du système définies à l'intégration de la CN (commande numérique) par le constructeur de la machine. Les unités internes du système sont définies pour :

- les axes linéaires;
- les axes rotatifs.

Les unités internes affectent directement les courses machine, ainsi que les formats d'acquisition et de visualisation des cotes sur les axes linéaires et rotatifs.

La figure 9 montre un exemple d'un mot :

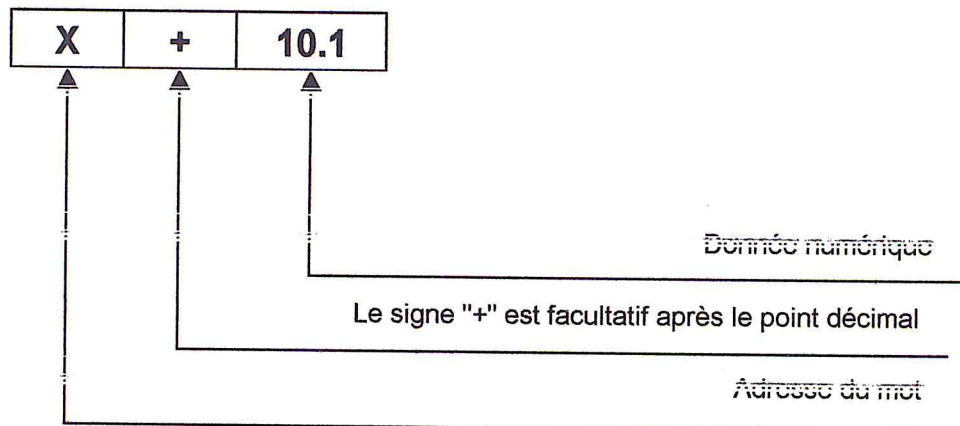


Figure 9 . Exemple d'un mot.

10.2. Format d'un bloc [8] :

Un bloc ou une séquence définit une ligne d'instructions composée de mots codés à transmettre au système de commande.

Le format de bloc définit la syntaxe des mots de fonction et de dimension composant chaque bloc de programmation.

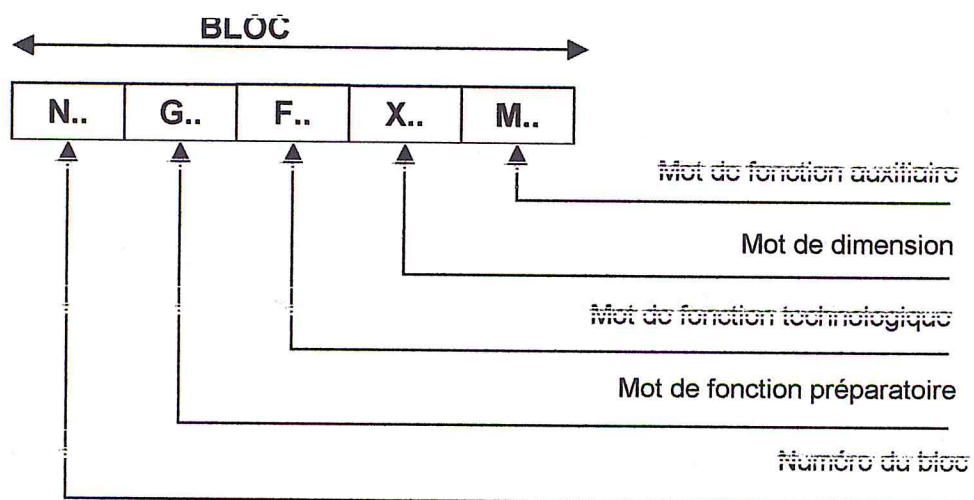


Figure 10 . Format d'un bloc.

Voici un exemple de bloc qui est montré dans la figure suivante :

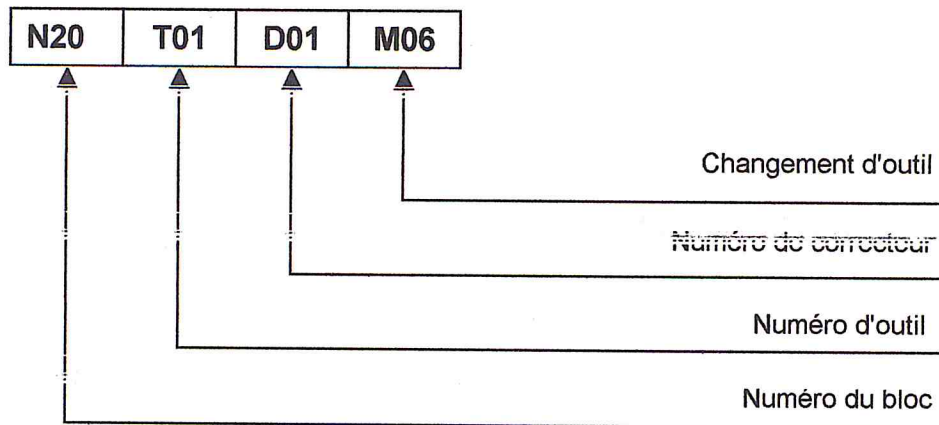


Figure 11 . Exemple d'un bloc.

11. Classification des fonctions préparatoires G et auxiliaires M [10]:

11.1. Classification des fonctions préparatoires G :

Les types existants des fonctions G sont :

- Fonctions G modales;
- Fonctions G non modale.

Certaines fonctions G doivent être programmées avec leurs arguments associés, mais la programmation de certaines fonctions G peut être incompatible avec l'état du programme en cours.

- **fonctions G modales** : elles appartiennent à une famille de fonctions G se révoquant mutuellement. Certaines familles de fonctions G comportent une fonction initialisée à la mise sous tension du système. la validité de ces fonctions est maintenue jusqu'à ce qu'une fonction de même famille révoque leur validité.
- **Fonctions G non modales** : fonctions seulement valides dans le bloc où elles sont programmés.

11.2. Classification des fonctions auxiliaires M :

Types de fonctions M :

- Fonctions M modales ;
- Fonctions M non modales.

Les fonctions M peuvent être :

- Des fonctions « avant » ou « après » ;
- Des fonctions codées ou décodées.

Les différentes fonctions sont décrites dans les paragraphes suivants :

- **Fonctions M modales** : elles appartiennent à une famille de fonctions M se révoquant mutuellement, certaines familles de fonctions M comportent une fonction initialisée à la mise sous tension du système. La validité de ces fonctions est maintenue jusqu'à ce qu'une fonction de même famille révoque leur validité.

- **Fonctions M non modales** : ce sont des fonctions uniquement valides dans le bloc où elles sont programmées.
- **Fonctions M « avant »** : ce sont des fonctions exécutées avant déplacements sur les axes programmés dans le bloc.
- **Fonctions M « après »** : ce sont des fonctions exécutées après déplacements sur les axes programmés dans le bloc.
- **Fonctions M codées** : elles sont définies par le constructeur machine et sont spécifiques à la machine.
- **Fonctions M décodées** : elles sont des fonctions de base du système et dont la signification est connue.

L'annexe I donne les fonctions les plus utilisées.

12. conclusion :

Nous avons présenté dans ce chapitre les différentes parties d'une fraiseuse à commande numérique, les différents types, les importants avantages, ainsi que la structure et la génération d'un programme machine à travers les différentes fonctions préparatoires et auxiliaires utilisées dans la génération du programme d'usinage «G-code».

1. Introduction :

L'usinage est l'étape finale de l'opération de création d'une pièce mécanique pour cela, elle doit occuper une place semblable aux phases de digitalisation, de modélisation géométrique, de dimensionnement pour concrétiser l'ensemble du travail de conception.

Dans les milieux industriels de fabrication, plusieurs procédés d'usinage sont actuellement utilisés, mais le procédé le plus répandu est le fraisage et qui est parmi les premiers procédés ayant employé des techniques numériques.

Actuellement, les moyens de production font un appel intensif aux technologies de commande numérique pour le découpe au laser, la dépose des matériaux composites hautes performances, le soudage,...

Dans ce chapitre, nous allons revoir l'usinage des surfaces sur des fraiseuses à commande numérique.

2. Différents paramètres d'usinage [7]:

Les différents paramètres à prendre en compte pour l'usinage d'une surface sont les suivants:

- Le choix de la forme et des dimensions de l'outil ;
- La détermination des trajectoires de l'outil sur la surface ;
- Le positionnement relatif de l'outil par rapport à la surface ;
- La vérification de la compatibilité des dimensions de l'outil par rapport à la forme de l'objet ;
- Matériaux des outils de coupe et de la pièce à usiner.

3. Types d'outils (fraises) utilisés pour l'usinage des surfaces gauches [10] :

Pour la génération des surfaces de formes libres (surfaces gauches), on utilise des fraises à bout sphérique appelés "fraises boules", des fraises toriques ou des fraises cylindriques (figure1). Dans le cas de l'usinage sur des fraiseuses à 3 axes, les fraises à bouts sphériques sont les plus utilisées.

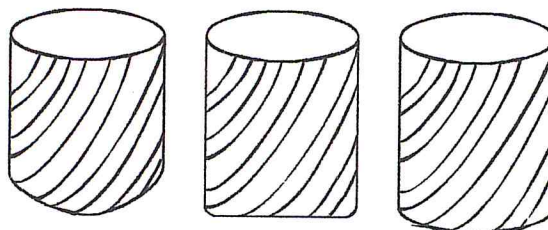


Figure 1 . Une fraise boule, une fraise torique et une fraise cylindrique.

4. Position de l'outil par rapport à la surface [7]:

L'usinage des surfaces gauches, nécessite le positionnement de l'outil par rapport à la surface à usiner tout le long du trajet d'usinage. Le calcul des différentes positions de l'outil s'appuie sur le modèle géométrique de la surface, de la géométrie nominale de l'outil (forme d'outil utilisée).

Le positionnement d'une fraise hémisphérique et d'une fraise torique tangentes à la surface est donné par la figure 2.

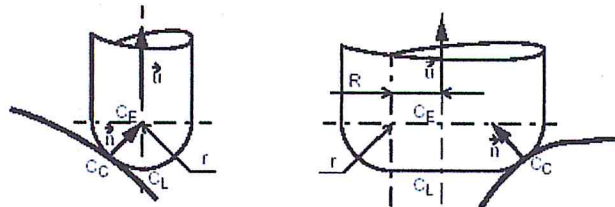


Figure 2 . Représentation des outils tangents à la surface.

On appelle :

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

Dans le cas d'un outil hémisphérique, les positions des points de l'outil sont données par les relations suivantes :

$$\begin{cases} \overline{OC_E} = \overline{OC_C} + r\vec{n} \\ \overline{OC_L} = \overline{OC_E} - r\vec{u} = \overline{OC_C} + r\vec{n} - r\vec{u} \end{cases} \quad (3.1)$$

Pour un outil torique, les positions sont données par les relations suivantes :

$$\vec{\kappa} = \frac{\vec{u} \wedge \vec{n}}{\|\vec{u} \wedge \vec{n}\|} \quad (3.2)$$

$$\begin{cases} \overline{OC_E} = \overline{OC_C} + r\vec{n} + R \cdot \frac{\vec{\kappa} \wedge \vec{u}}{\|\vec{v} \wedge \vec{u}\|} \\ \overline{OC_L} = \overline{OC_E} - r\vec{u} = \overline{OC_C} + r\vec{n} - r\vec{u} + R \cdot \frac{\vec{\kappa} \wedge \vec{u}}{\|\vec{v} \wedge \vec{u}\|} \end{cases} \quad (3.3)$$

Les équations précédentes permettent de calculer soit les coordonnées du centre de l'outil à partir du point de contact, soit le point de contact à partir des coordonnées du centre. Cela

définit le type de posage de l'outil, qui met en relation la stratégie d'usinage et la surface à usiner.

- = Posage par le point de contact de l'outil: on appelle posage par le point de contact, une opération de calcul de la position du centre de l'outil à partir d'une position de contact donnée;
- Posage par le centre de l'outil: on appelle posage par le centre de l'outil, une opération de calcul direct d'une position du centre de l'outil sans la donnée initiale de la position du point de contact.

4.1. Calcul d'une position de l'outil tangente à la surface à usiner:

Le calcul d'une position tangente de l'outil est une méthode de posage par le point de contact, implicite, directe ou indirecte.

Ce calcul est basé sur les équations (3.1), (3.2) et (3.3), et suit la démarche suivante. A partir d'un point de la surface à usiner :

- calcul de la normale à la surface au point considéré ;
- calcul de la position du centre de l'outil ;
- calcul de la position de l'extrémité de l'outil.

5. Erreurs d'usinage [14,15] :

Pour l'usinage des surfaces gauches nous devons prendre en comptes les erreurs suivantes :

5.1. Erreur de flèche :

Les courbes de l'espace que suit l'outil sont approchées par interpolation linéaire et l'outil usine une ligne brisée, car les directeurs de commande numérique fonctionnent par interpolation linéaire. Cependant des interpolations par des B-Spline commencent à apparaître. Aussi le pas longitudinal d'usinage est calculé en général par le respect de la **tolérance d'usinage** qui est la valeur maximale de l'erreur de flèche entre la courbe et chacun des segments de la ligne brisée (figure3).

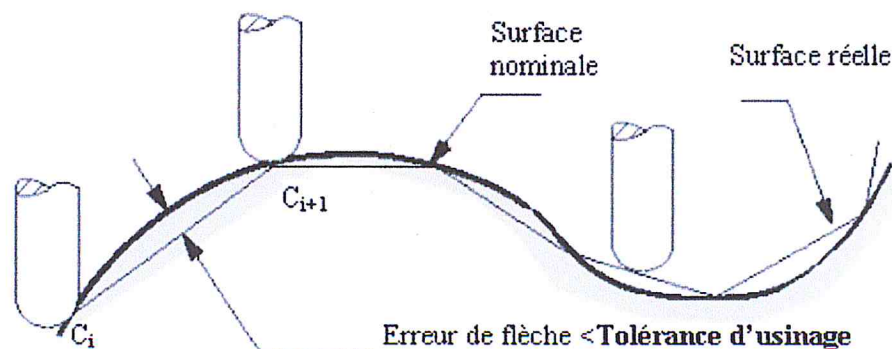


Figure 3 . Erreur de flèche.

5.2. Erreur de crête :

Le pas d'usinage transversal peut être calculé en respectant soit un critère de **distance maximale entre passes**, soit un critère de **hauteur de crête**. Ces deux paramètres sont équivalents, et sont liés par le rayon de l'outil. De plus la distance maximale entre passes peut être établie entre les surfaces de guidage ou sur la surface nominale (figure 4).

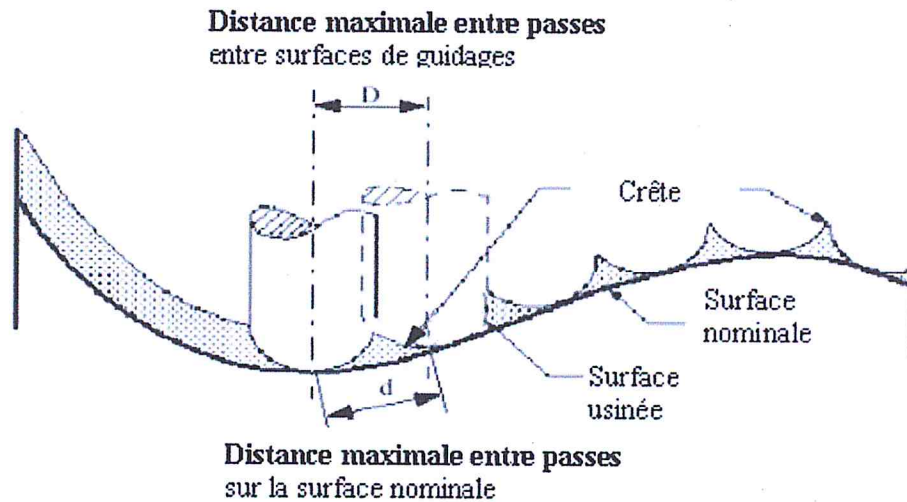


Figure 4. Erreur de crête.

5.3. Erreur d'interférence :

On dit qu'il y'a interférence entre l'outil et la surface s'il y'a une partie concave dont le rayon de courbure est inférieur à celui de l'outil (interférence A)(figure 5), ou dans le cas de raccordement en tangente entre les courbes de contact (interférence B)(figure 6).

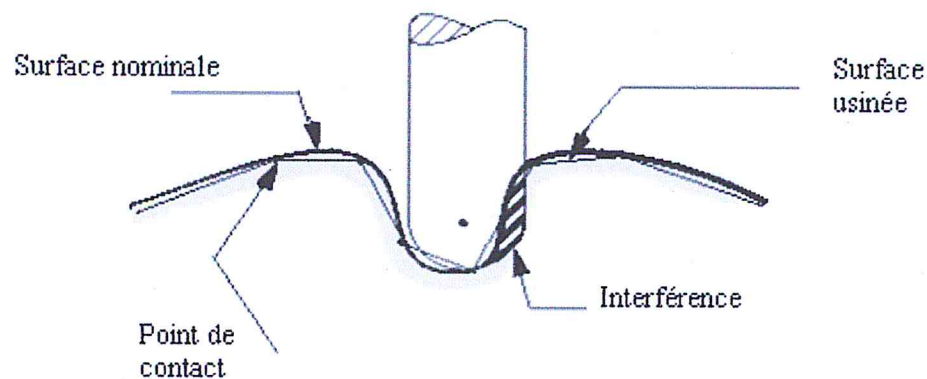


Figure 5. Interférence A.

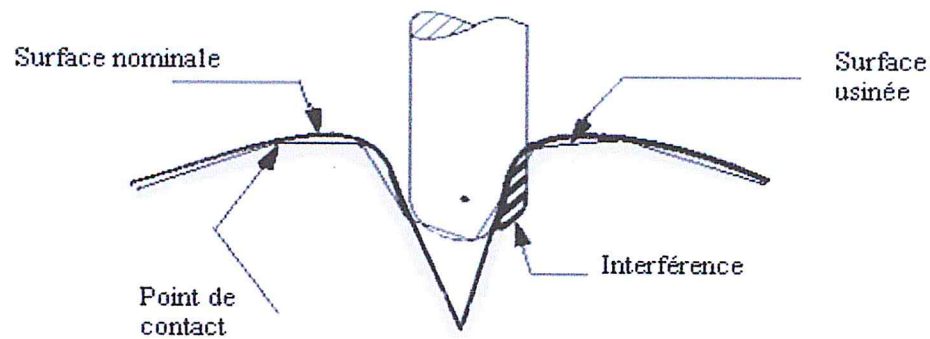


Figure 6 - Interférence R.

5.4. Erreur de collision :

On dit qu'il y'a collision entre l'outil et la surface dans les parties où l'angle formé par la normale et l'axe de la boucle est supérieur à 90° (figure 7).

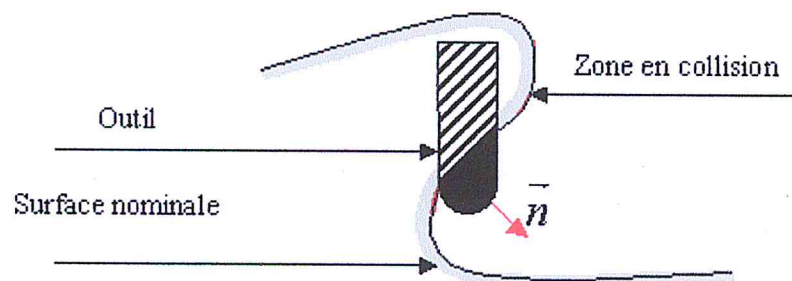


Figure 7 . L'outil en collision.

6. Méthodes d'usinage des surfaces gauches par des courbes isoparamétriques[7]:

6.1. Aller simple (one way) :

Cette méthode consiste à usiner la surface selon une isoparamétrique (u ou v) jusqu'à sa fin et en suite faire déplacer l'outil en quittant la surface dans l'autre isoparamétrique et refaire la même étape jusqu'à la fin de la surface (figure 8).

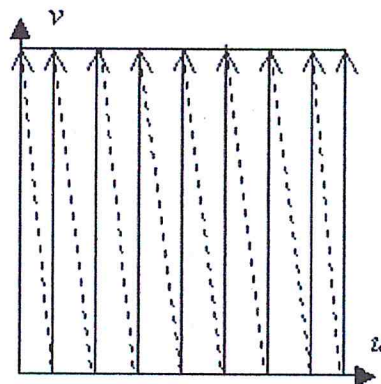


Figure 8 . La méthode d'usinage one way.

6.2. Aller retour (Zig Zag) :

Cette méthode consiste à usiner la surface selon une isoparamétrique (u ou v) jusqu'à sa fin, et en suite faire déplacer l'outil sans quitter la surface et usiner la surface dans l'autre sens et refaire la même étape jusqu'à la fin de la surface (figure9).

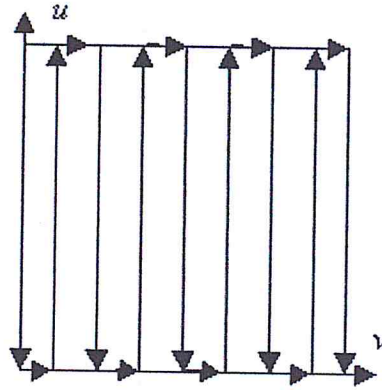


Figure 9 . La méthode d'usinage Zig Zag.

6.3. Concentrique :

Cette méthode consiste à usiner la surface selon des courbes parallèles aux frontières du domaine de définition de la surface jusqu'à la fin de la surface (figure10).

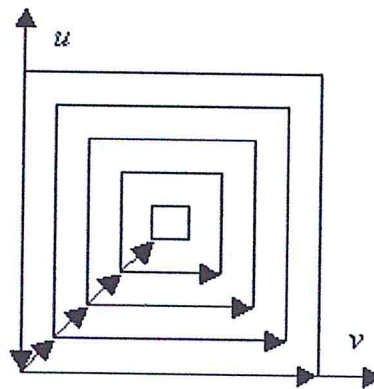


Figure 10 . La méthode d'usinage concentrique.

6.4. Spiral in :

Dans cette méthode l'outil d'usinage suit une trajectoire en forme de spirale jusqu'à usinage de toute la surface en partant de l'extérieure de la surface (figure11).

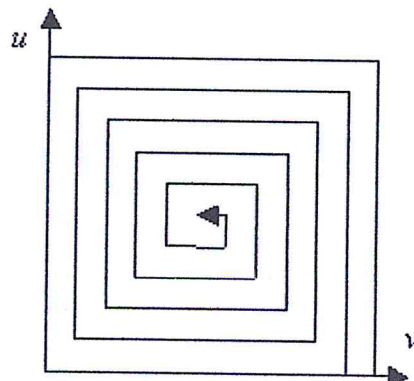


Figure 11 . La méthode d'usinage Spiral in.

6.5. Spiral out :

Dans cette méthode l'outil d'usinage suit une trajectoire en forme de spirale jusqu'à usinage de toute la surface en partant de l'intérieur de la surface (figure 12).

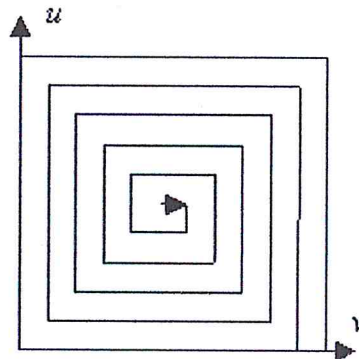


Figure 12 . La méthode d'usinage Spiral out.

6.6. Radiale :

Cette méthode consiste à usiner la surface en commençant du centre de la surface jusqu'à la frontière de la surface et refaire la même étape jusqu'à la fin de la surface en déplaçant chaque fois le point de l'extrémité (figure 13).

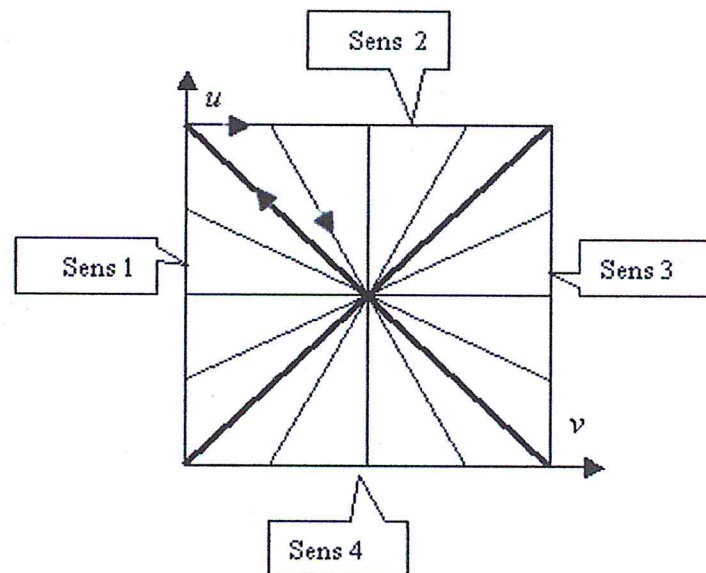


Figure 13 . La méthode d'usinage Radiale.

7. Autres méthodes d'usinage [6,7]:

En plus de l'usinage par les méthodes isoparamétriques, il existe d'autres méthodes d'usinage :

7.1. Usinage par plans parallèles au plan (x, z) :

Ce type de trajectoire est obtenu par la construction de courbes obtenues par intersection de la surface à usiner avec des plans parallèles au plan (XZ). De même, cette intersection peut être faite en n'importe quel plan vertical avec la surface à usiner (figure14).

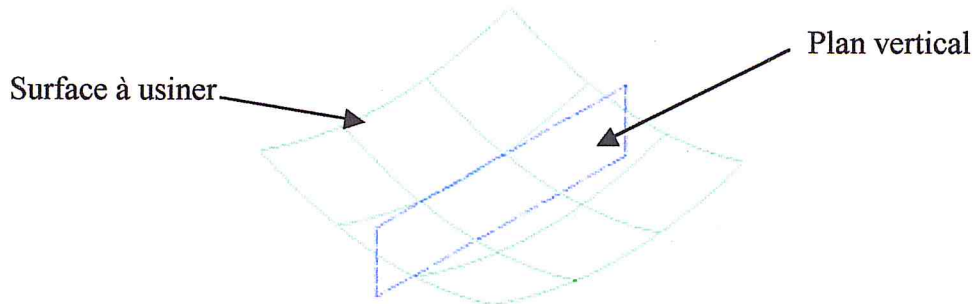


Figure 14 . Usinage par plans parallèle au plan (x, z).

7.2. Trajectoire obtenue par usinage par courbes de niveau :

Ce type de trajectoires est obtenu par intersection de la surface à usiner avec des plans horizontaux avec différentes valeurs de z. Les courbes obtenues sont appelées courbes de niveau (figure15).

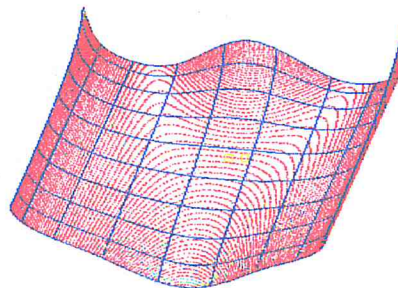


Figure 15 . Usinage par courbes de niveau.

8. Qualité d'usinage des surfaces [12]:

La qualité de l'usinage des surfaces est caractérisée par :

- La précision des dimensions : c'est à dire la correspondance des dimensions réelles de la pièce à la cotation du dessin ;
- Les écarts admissibles de la forme géométrique ne doivent pas dépasser les limites imposées par la tolérance sur l'imprécision d'usinage (tolérances de formes) ;
- Les écarts de positions relative des surface de la pièce (défaut de parallélisme, de perpendicularité, et l'écart d'angle imposé entre les plans,..., etc.) ;

9. Conclusion :

Nous avons vu dans ce chapitre l'usinage et ses différents paramètres, les différents types d'outils utilisés pour l'usinage des surfaces gauches, ainsi que les différentes erreurs qui peuvent être produites lors de l'usinage. Les différentes méthodes d'usinage des surfaces gauches par des courbes isoparamétriques, des plans parallèles, et des courbes de niveau, sont décrites à la fin de ce chapitre.

1. Introduction :

Après l'usinage, les surfaces usinées contiennent toujours des erreurs causées par les différentes approximations faites lors de la génération des trajets outils. Ces erreurs peuvent être soit l'existence des zones non usinées, soit l'existence des zones où l'usinage est très profond de ce qu'il est demandé (zones d'interférence), soit les deux en même temps. Pour cela une des solutions qui existe est la simulation. C'est un outil très important permettant de réduire les temps d'usinage effectifs et de valider les trajets d'usinage en détectant les différentes erreurs et leurs emplacements.

Un important problème pratique est de trouver une réponse à la question suivante : est-ce que les surfaces usinées sont dans l'intervalle de la tolérance exigée dans le cahier des charges ? Une fois les zones critiques détectées, il est nécessaire de corriger les erreurs.

Les fonctions nécessaires à la simulation sont présentées dans ce chapitre.

2. Définition de simulation [15] :

Le mot « simulation » vient du verbe simuler, qui veut dire faire apparaître comme réel, la simulation du fraisage est de faire toutes les actions de fraisage : le mouvement de l'outil, l'enlèvement de la matière et détection des erreurs qui peuvent être produites lors de l'usinage.

Puisque l'opération d'usinage prend beaucoup de temps, surtout pour les grandes pièces, et même pour ne pas abuser de la matière, qui est très coûteuse, on simule l'opération d'usinage pour détecter les erreurs puis les corriger, avant de passer à l'usinage effectif.

Pour la simulation d'une pièce, il y a plusieurs méthodes de détection, de vérification et de correction des erreurs qui sont décrites dans ce chapitre.

3. Méthodes de simulation [15, 18, 19] :

3.1. Méthode de représentation solide CSG :

La méthode de simulation basée sur la représentation solide CSG (Constructive Solid geometry) s'appuie sur la modélisation solide de l'enveloppe du trajet de l'outil (voir figure1). La simulation est effectuée en créant les volumes générés par les mouvements élémentaires de l'outil sous forme CSG, puis contrôler s'il y a intersection entre le modèle de la surface nominale et chaque volume balayé par l'outil.

Figure 1 . Représentation solide du volume balayé par l'outil.

Cette méthode est précise mais dans le cas où le nombre de déplacements est grand, cette solution devient très coûteuse en temps de calcul, car si le nombre de déplacements est N alors le nombre d'opérations est évalué à N^4 .

3.2. Méthode du Z-Buffer :

La technique du Z-buffer est une méthode générale qui s'applique aussi aux méthodes de contrôle basées sur la modélisation CSG et la visualisation de trajet. Cette méthode consiste à représenter la surface sous forme d'une matrice ou d'une grille dont les lignes et les colonnes représentent un quadrillage géométrique dans le plan xy, puis associer un segment vertical d'une altitude z initiale (voir figure 2).

Lors de l'usinage, on calcule l'intersection des segments avec le trajet d'usinage. Pour chaque segment, on retient l'altitude d'intersection la plus basse. On obtient une représentation Z-buffer du trajet d'usinage. Les erreurs de simulation proviennent du pas de la grille et de la modélisation du trajet.

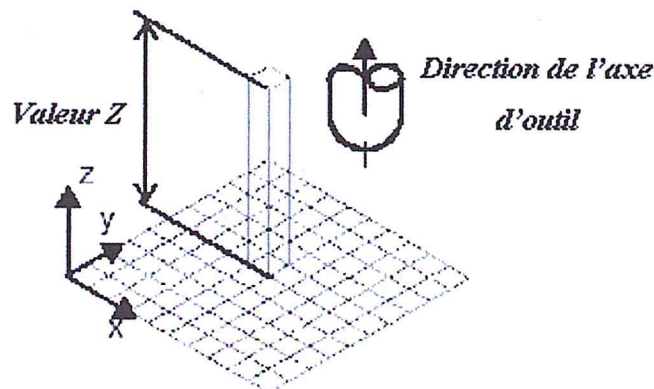


Figure 2 . Concept de la méthode du Z-Buffer.

Dans cette méthode de simulation et de vérification, il suffit de générer le Z-Buffer de la pièce nominale, le brut et la pièce usinée, puis pour détecter les interférences, les collisions et les zones non usinées il suffit de faire les soustractions nécessaires des Z-Buffer entre eux.

La figure 3 représente le Z-Buffer d'une pièce donnée.

Figure 3 . Représentation d'une pièce par le Z-Buffer.

La finesse de la grille du Z-Buffer induit la précision de la vérification, c'est-à-dire plus les carreaux de la grille sont plus petits on obtient des résultats plus exacts, donc l'opération devienne coûteuse en temps à cause des opérations de calcul.

3.3. Méthode de visualisation graphique :

Cette méthode se déduit de la méthode du Z-buffer (ou le contraire). Cette fois-ci le plan de projection est le plan de l'écran associé au point de vue de l'utilisateur. La technique est la même que celle du Z-buffer. Pour chaque point de la grille on stocke la position et la longueur du segment de matière associée, et on opère la soustraction avec le mouvement de l'outil.

3.4. Méthode du point vecteur :

Cette méthode consiste à découper la surface selon des méthodes spécifiques pour avoir un ensemble de points. Pour chaque point on associe un vecteur qui a comme limites la surface nominale et le brut.

Il existe deux façons pour l'orientation de ces vecteurs (voir figure4) :

- 1- Suivant la normale à la surface (exacte) : dans ce cas, chaque vecteur est indépendant linéairement des autres vecteurs.
- 2- Suivant l'axe Z de l'outil (simplifiée) : dans ce cas, tous les vecteurs sont parallèles à l'axe Z.

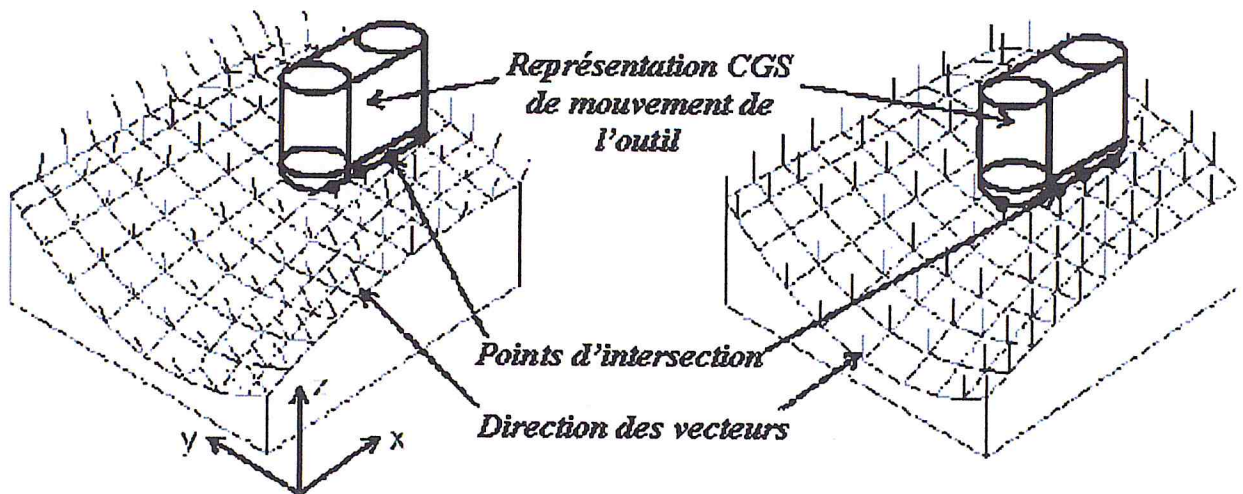


Figure 4 . Orientation des vecteurs suivant la normale. Et l'axe de l'outil.

Pour simuler l'usinage on calcule l'intersection des vecteurs avec la représentation CSG du mouvement de l'outil, au fur et à mesure des déplacements de l'outil, la longueur et le sens des vecteurs sont modifiées, alors pour détecter les erreurs comme les collisions, les interférences et les zones non usinées il suffit de voir la direction des vecteurs (positive : zone non usinée, négative : usinage plus que voulu) et leurs longueurs (s'ils ne sont pas dans les tolérances d'usinage, il faut passer à la correction).

Pour le choix de l'orientation des vecteurs, si on prend l'orientation suivant l'axe de l'outil ça facilite les calculs mais la longueur du vecteur restante ne représente pas réellement le volume de matière restante ou usinée, contrairement à l'orientation suivant la normale, malgré la difficulté des calculs par rapport à la première orientation, les résultats finaux donnent l'idée exacte sur la matière enlevée et la matière restante.

4. Triangulation [20]:

La triangulation est un problème de la Géométrie Informatique. L'objectif de la triangulation est de discrétiser un objet (en 3 D) et de l'approximer par des triangles.

Les sommets des triangles résultants de la triangulation forment l'ensemble de points nécessaires pour l'opération de simulation.

La figure 5 représente une surface semis bien triangulé.

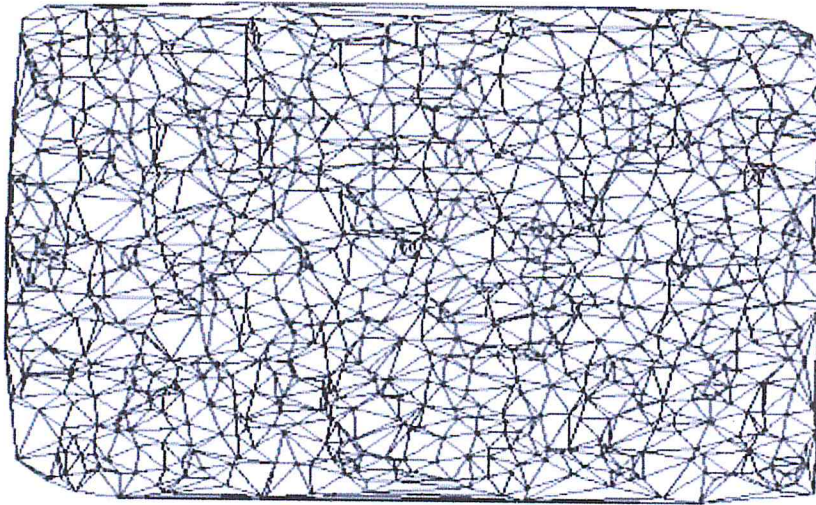


Figure 5 . Semis bien triangulé (triangles compacts).

Le problème qui se pose est de trouver la meilleure triangulation possible. Intuitivement, il s'agit d'avoir les triangles "les moins allongés" possibles.

La triangulation de Delaunay, permet de caractériser la meilleure triangulation possible. Avant de la définir, nous allons introduire la définition des diagrammes de Voronoï.

4.1. Diagramme de Voronoï :

On se donne une famille de n points $\{P_i\}$ du plan. A chaque couple ordonné de points (P_i, P_j) , on associe le demi-plan, bordé par la médiatrice des deux points, et contenant P_i . A chaque point P_i est ainsi associé $n-1$ demi-plans, on note V_i l'intersection de ces demi-plans. L'ensemble formé par ces demi-plans (qu'on limite par l'enveloppe convexe du semis de points) s'appelle le diagramme de Voronoï associé au point P_i (voir figure 6).

Ce diagramme a la propriété suivante : soit O un point quelconque de l'enveloppe convexe du semis, si O est compris dans V_i , alors le point du semis le plus proche de O est P_i .

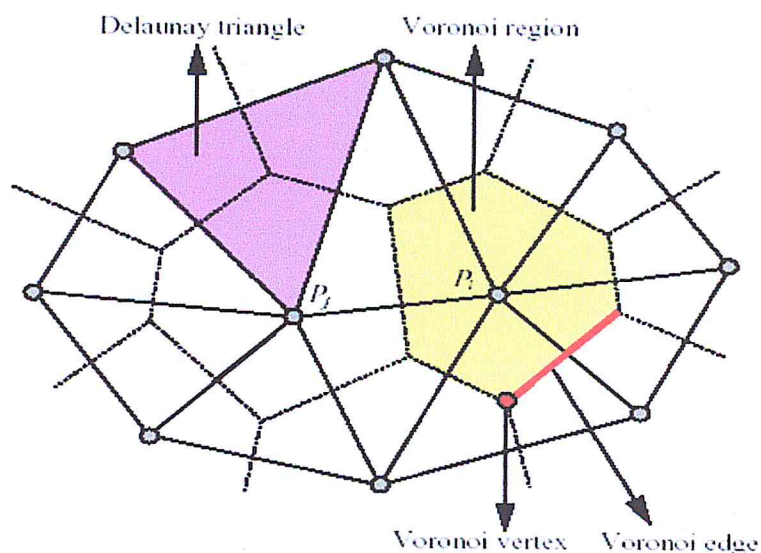


Figure 6 . Diagramme de Voronoï

4.2. Triangulation de Delaunay:

On appelle triangulation de Delaunay de l'enveloppe convexe du semis de points $\{P_i\}$, l'ensemble des triangles s'appuyant sur les $\{P_i\}$, et obtenu par dualité à partir du diagramme de Voronoï:

- Le segment $P_i P_j$ appartient à la triangulation si V_i et V_j ont une intersection non nulle ;
- Le triangle $P_i P_j P_k$ appartient à la triangulation si V_i , V_j , et V_k ont une intersection non nulle.

Intuitivement on relie chaque point à son voisin, le diagramme de Voronoï permettant de définir plus précisément la notion de voisin.

4.3. Algorithme de Delaunay:

Cet algorithme est incrémental. On part d'une triangulation partielle du semis (trois points quelconques du semis suffisent), et on construit la triangulation en y intégrant un point à chaque étape. Il existe deux cas; le point à rajouter est à l'extérieur de la triangulation existante, ou à l'intérieur. Nous ne détaillerons que le deuxième cas. Dans la pratique, il suffira à partir de quatre points fictifs englobant les semis à trianguler.

Ajout d'un point au semis déjà triangulé (voir figure 7). Soit les n points $\{P_i\}$ déjà triangulés et le point O à trianguler, contenu dans l'enveloppe convexe de $\{P_n\}$. On cherche l'ensemble des triangles existants dont le cercle circonscrit contient O . On sépare alors les arêtes des triangles ainsi trouvés en deux:

- l'ensemble $\{a_j\}$ de celles qui appartiennent à deux de ces triangles.
- l'ensemble $\{b_j\}$ de celles qui n'appartiennent qu'à un seul de ces triangles.

La nouvelle triangulation consiste à casser les triangles trouvés, en supprimant les arêtes a_j . On forme les nouveaux triangles de façon qu'il s'appuie sur une arête b_j et sur O .

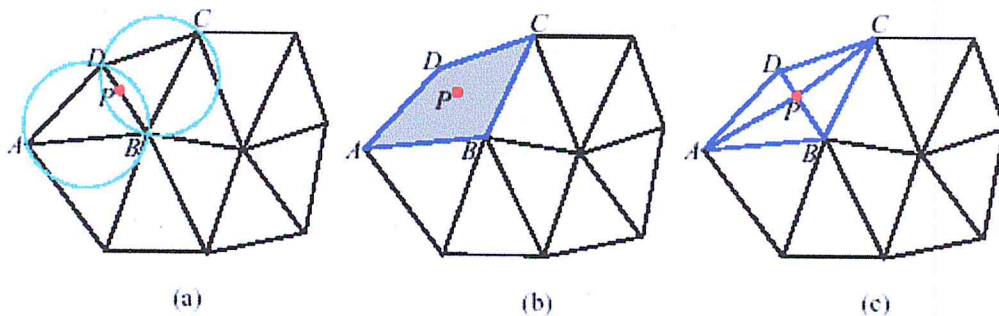


Figure 7. Algorithme de Delaunay.

5. Algorithme de subdivision [16, 17]:

Cette section décrit comment la surface est subdivisée pour produire du SPS (Surface Points Set) de sorte que l'interférence non détectée soit moins qu'une erreur définie par l'utilisateur.

Puisque le temps de simulation est proportionnel au nombre des points du SPS, on cherche une triangulation des surfaces à usiner ayant le minimum de nombre de points. La profondeur possible maximum d'une interférence non détectée peut être analysée en termes d'erreur de surface e_s et e_p d'erreur de pénétration. Dans le plus mauvais cas, ces deux erreurs s'ajoutent, ainsi notre triangulation doit assurer que $e_s + e_p < e$, où e est l'erreur définie par l'utilisateur. Cependant, si le triangle se trouve entièrement en dehors de la surface cas des zones convexes les erreurs sont dans la même direction et on doit s'assurer que les max (e_s, e_p) $< e$. Pour les zones convexes, les erreurs e_s et e_p s'ajoutent et on doit s'assurer que $e_s + e_p < e$ (voir figure 8).

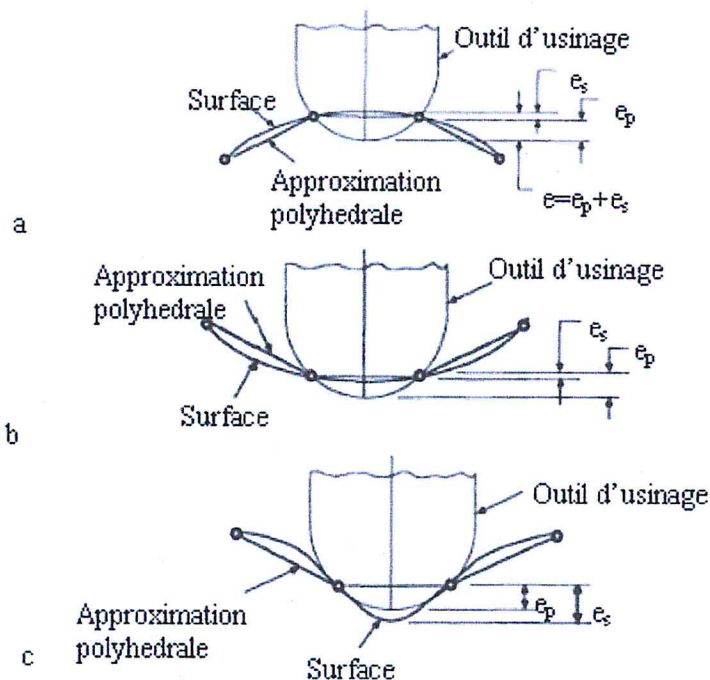


Figure 8. L'erreur de simulation de découpage est provoquée par écart entre la surface réelle et l'approximation polyhedral (e_s) et la crête de l'outil entre les points surface (e_p). Quand l'approximation polyhedral se trouve à l'intérieur de la surface, comme en (a), l'erreur de simulation est la somme de e_s et du e_p . Quand l'approximation polyhedral se trouve en dehors de la surface, comme en (b) et (c), l'erreur de simulation est le maximum de e , ou e_p .

Nous avons la distance maximum entre les points d'intersection de n'importe quel triangle est d , la saillie non détectée maximum d'outil, e_p , est donnée par:

$$e_p = r - (r^2 - d^2/3)^{0.5} \quad (4.1)$$

Où r est le rayon de l'outil hémisphérique. Ainsi la crête non détectée maximum d'outil dépend de la taille du triangle et du rayon de l'outil utilisé. L'erreur de crête d'outil peut être

rendue aussi petite comme désirée en diminuant la taille du triangle. Cette formule peut être employée pour déterminer r donné par d et e . Résolvant pour d , nous obtenons

$$d = (6re_p - 3e_p^2)^{0.5} \quad (4.2)$$

Quand e_p est petit par rapport à r (ce qui est habituellement le cas), d peut être approché par :

$$d \sim (6re_p)^{0.5} \quad (4.3)$$

La distance entre les points est proportionnelle à la racine carrée du rayon de l'outil et de l'erreur de crête d'outil. Puisque le nombre de points est inversement proportionnel à d^2 , nous concluons que dans les zones de la faible courbure le nombre de points croît linéairement avec l'inverse du rayon d'outil et l'inverse de l'exactitude désirée de crête d'outil. Notre but est la triangulation de chaque surface avec des triangles presque équilatéraux. Une méthode employée pour garder le nombre de points près du minimum. Nous commençons par diviser chaque surface en bandes. Chaque bande est au commencement triangulée de sorte que les triangles soient presque équilatéraux, avec des côtés moins de 95 pour cent de d (Voir figure 9). Dans les zones de faible courbure, e_s sont proches de zéro et la triangulation est complète, parce que $e_s + e_p \leq e$.

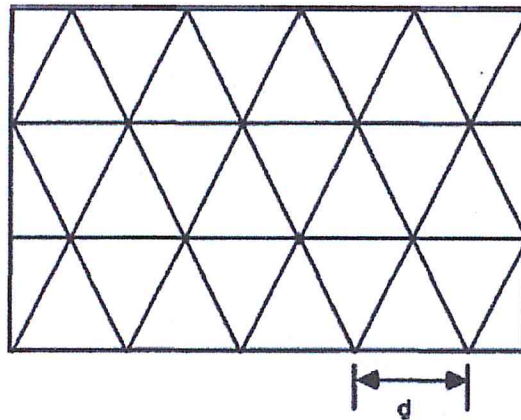


Figure 9. La surface est triangulée en la divisant en triangles.

La subdivision est exécutée plus efficacement dans l'espace des paramètres que dans l'espace cartésien. Nous mesurons la largeur et la longueur de la surface dans l'espace cartésien et nous employons cette information pour décider combien de bandes sont nécessaires et puis combien de triangles peuvent entrer dans chaque bande.

Pour chaque triangle nous calculons ensuite e_s et le e_p réels pour voir si le triangle vérifiera l'erreur de simulation e . Sinon, le triangle est récursivement subdivisé. Quand un côté est trop long, le triangle est subdivisé.

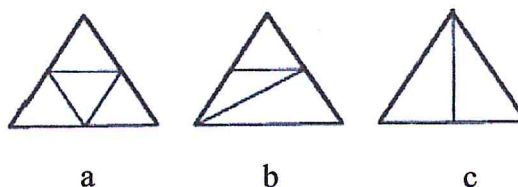


Figure 10. Subdivision des triangles.

Chacune des arêtes de triangle est testée, et si le test échoue pour chacun des trois côtés, le triangle est subdivisé en quatre triangles en utilisant la subdivision. Si le test échoue pour un ou deux côtés, alors le triangle est subdivisé en deux ou trois triangles, respectivement (Voir figure 10).

Le même procédé est périodiquement appliqué à tous les nouveaux triangles résultant de la subdivision et quitte quand ϵ est moins que l'exactitude indiquée.

6. Intersection [7]:

L'intersection se fait entre la forme CSG de déplacement de l'outil avec les vecteurs sortants de la surface à usiner, mais le calcul direct est impossible alors on découpe la forme CSG en des formes simples pour qu'on puisse calculer l'intersection.

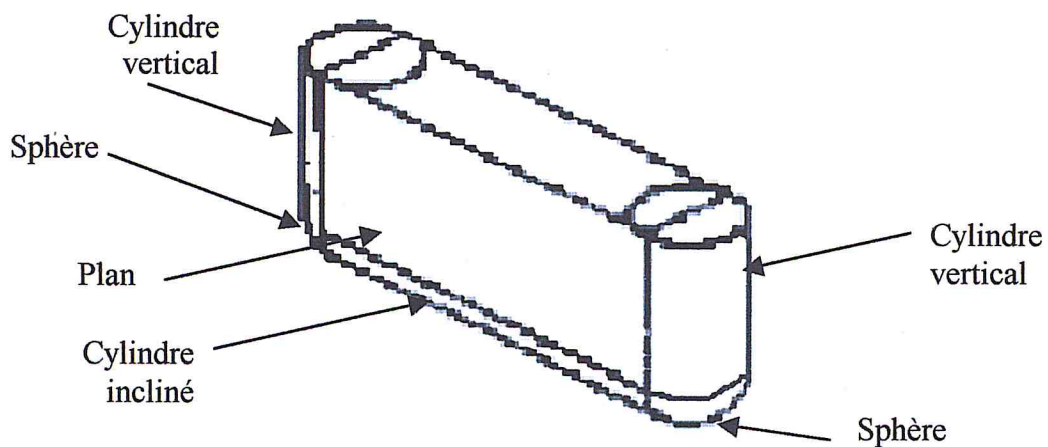


Figure 11. Composition de la représentation CSG.

D'après cette figure on trouve qu'une représentation CSG du trajet de l'outil entre deux positions successives est composée de (Voir figure 11) :

- Deux parties sphériques ;
- Trois cylindres ;
- Deux plans.

De cette manière, le calcul de l'intersection devient simple. Il suffit de calculer l'intersection du vecteur avec chaque partie puis choisir la valeur valide comme résultat final de l'intersection.

Pour avoir le point d'intersection avec chaque partie de la représentation il faut avoir les équations de la droite (représentant le vecteur sortant de la surface), le plan, le cylindre et la sphère ; puis pour avoir le point d'intersection il suffit de résoudre les systèmes : droite-plan, droite-sphère et droite-cylindre, et choisir les points d'intersection valides.

6.1. Plan dans l'espace :

Parmi les plans parallèles à l'axe OZ, il n'y a qu'un seul plan qui passe par les deux points $P_0(x_0, y_0, z_0)$ et $P_1(x_1, y_1, z_1)$ (Voir figure 12).

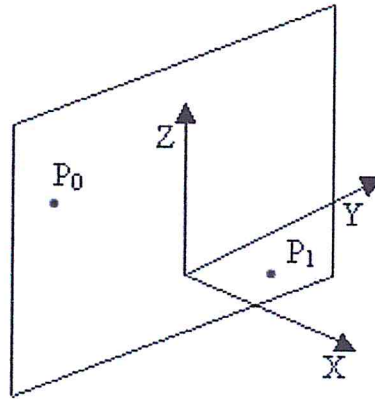


Figure 12 . Plan parallèle à l'axe OZ et passe par deux points.

L'équation d'un plan dans l'espace est comme suit :

$$Ax + By + Cz + D = 0 \quad (4.4)$$

Puisque le plan est parallèle à l'axe OZ, alors $C = 0$, donc l'équation (4.4) devient :

$$Ax + By + D = 0 \quad (4.5)$$

La projection de ce plan sur le plan XY donne une droite qui est définie par l'équation du plan. On peut déterminer l'équation de cette droite par la projection des deux points P_0, P_1 sur le plan XY (Voir figure 13).

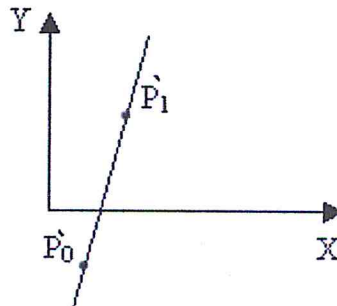


Figure 13 . Projection du plan parallèle à l'axe OZ et des deux points sur le plan XY.

Le vecteur directeur de cette droite est : $\overrightarrow{P_0P_1} \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix} = \begin{pmatrix} -B \\ A \end{pmatrix}$

Alors
$$\begin{cases} B = -(x_1 - x_0) \\ A = y_1 - y_0 \end{cases}$$

Pour déterminer D on remplace les coordonnées du point P_0 (ou P_1) dans l'équation (4.5), on trouve :

$$D = -((y_1 - y_0) x_0 - (x_1 - x_0) y_0)$$

Alors l'équation du plan parallèle à l'axe OZ et passe par deux points $P_0(x_0, y_0, z_0)$ et $P_1(x_1, y_1, z_1)$ est comme suit :

$$(y_1 - y_0) x - (x_1 - x_0) y - ((y_1 - y_0) x_0 - (x_1 - x_0) y_0) = 0 \quad (4.6)$$

6.2. Droite dans l'espace :

Par deux points P_0, P_1 dans l'espace on ne peut faire passer qu'une seule droite. Soit P_0, P_1 deux points de l'espace \mathbb{R}^3 . La droite qui passe par P_0 et P_1 est définie comme l'ensemble des points \bar{M} qui vérifient la relation $\overrightarrow{P_0M} \parallel \overrightarrow{P_0P_1}$.

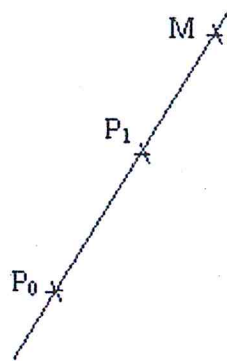


Figure 14. Droite dans l'espace.

Soient $P_0(x_0, y_0, z_0), P_1(x_1, y_1, z_1)$ deux points de l'espace, soit $M(x, y, z)$ un point de la droite (Δ_M) qui passe par les deux points P_0, P_1 (Voir figure 14).

$$M \in (\Delta_M) \Leftrightarrow \overrightarrow{P_0M} \parallel \overrightarrow{P_0P_1}.$$

$$\Leftrightarrow \exists \alpha \in \mathbb{R} / \overrightarrow{P_0M} = \alpha \cdot \overrightarrow{P_0P_1}.$$

$$\Leftrightarrow \exists \alpha \in \mathbb{R} / \begin{vmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{vmatrix} = \alpha \cdot \begin{vmatrix} x_1 - x_0 \\ y_1 - y_0 \\ z_1 - z_0 \end{vmatrix} \quad (4.7)$$

Donc

$$\begin{cases} x - x_0 = \alpha(x_1 - x_0) \\ y - y_0 = \alpha(y_1 - y_0) \\ z - z_0 = \alpha(z_1 - z_0) \end{cases} \Leftrightarrow \begin{cases} \alpha = \frac{x - x_0}{x_1 - x_0} & \text{si } x_1 \neq x_0 \\ \alpha = \frac{y - y_0}{y_1 - y_0} & \text{si } y_1 \neq y_0 \\ \alpha = \frac{z - z_0}{z_1 - z_0} & \text{si } z_1 \neq z_0 \end{cases}$$

$$\text{D'où on peut écrire } \frac{x-x_0}{x_1-x_0} = \frac{y-y_0}{y_1-y_0} = \frac{z-z_0}{z_1-z_0} \quad (4.8)$$

De la relation (4.8) on peut définir un système d'équations composé de trois équations :

$$(I) \begin{cases} (x-x_0)(y_1-y_0) = (x_1-x_0)(y-y_0) \dots (1') \\ (y-y_0)(z_1-z_0) = (y_1-y_0)(z-z_0) \dots (2') \\ (x-x_0)(z_1-z_0) = (x_1-x_0)(z-z_0) \dots (3') \end{cases}$$

Dans ce système d'équations on peut remarquer que chaque équation peut être obtenue par les deux autres équations. Donc on peut éliminer une des équations, cette élimination se fait selon les sept cas suivants.

1^{er} cas: si $x_1 \neq x_0$, $y_1 \neq y_0$, $z_1 \neq z_0$.

Dans ce cas on peut éliminer n'importe quelle équation. En éliminant l'équation (2') on obtient le système :

$$\begin{cases} (y_1-y_0)x - (x_1-x_0)y - (y_1-y_0)x_0 + (x_1-x_0)y_0 = 0 \\ (z_1-z_0)x - (x_1-x_0)z - (z_1-z_0)x_0 + (x_1-x_0)z_0 = 0 \end{cases} \quad (4.9)$$

2^{eme} cas: si $x_1 \neq x_0$, $y_1 \neq y_0$, $z_1 = z_0$.

Remplaçons dans le système (z_1-z_0) par 0 on obtient:

$$\begin{cases} (x-x_0).(y_1-y_0) = (x_1-x_0).(y-y_0) \\ (y-y_0).0 = (y_1-y_0).(z-z_0) \\ (x-x_0).0 = (x_1-x_0).(z-z_0) \end{cases} \Leftrightarrow \begin{cases} (x-x_0).(y_1-y_0) = (x_1-x_0).(y-y_0) \\ (y_1-y_0).(z-z_0) = 0 \\ (x_1-x_0).(z-z_0) = 0 \end{cases} \quad (4.10)$$

On sait que $x_1-x_0 \neq 0$ et $y_1-y_0 \neq 0$ d'où le système devient :

$$\begin{cases} (x-x_0).(y_1-y_0) = (x_1-x_0).(y-y_0) \\ z-z_0 = 0 \\ z-z_0 = 0 \end{cases} \quad (4.11)$$

On remarque que la troisième équation n'est rien d'autre que la deuxième équation, donc on élimine la troisième équation, le système devient :

$$\begin{cases} x(y_1-y_0) - y(x_1-x_0) - x_0(y_1-y_0) + y_0(x_1-x_0) = 0 \\ z-z_0 = 0 \end{cases} \quad (4.12)$$

3^{eme} cas: si $x_1 \neq x_0$, $z_1 \neq z_0$, $y_1 = y_0$.

Par les mêmes étapes on peut trouver le système suivant:

$$\begin{cases} x(z_1-z_0) - z(x_1-x_0) - x_0(z_1-z_0) + z_0(x_1-x_0) = 0 \\ y-y_0 = 0 \end{cases} \quad (4.13)$$

4^{eme} cas: si $x_1 \neq x_0$, $y_1 = y_0$, $z_1 = z_0$.

$$\begin{cases} y-y_0 = 0 \\ z-z_0 = 0 \end{cases} \text{ Cette droite est parallèle à l'axe (ox)} \quad (4.14)$$

5^{eme} cas: si $x_1 = x_0, y_1 \neq y_0, z_1 \neq z_0$.

$$\begin{cases} y(z_1 - z_0) - z(y_1 - y_0) - y_0(z_1 - z_0) + z_0(y_1 - y_0) = 0 \\ x - x_0 = 0 \end{cases} \quad (4.15)$$

6^{eme} cas: si $x_1 = x_0, y_1 \neq y_0, z_1 = z_0$.

$$\begin{cases} x - x_0 = 0 \\ z - z_0 = 0 \end{cases} \quad \text{Cette droite est parallèle à l'axe (oy)} \quad (4.16)$$

7^{eme} cas: si $x_1 = x_0, y_1 = y_0, z_1 \neq z_0$.

$$\begin{cases} x - x_0 = 0 \\ y - y_0 = 0 \end{cases} \quad \text{Cette droite est parallèle à l'axe (oz)} \quad (4.17)$$

6.3. Cylindre dans l'espace :

Le cylindre qui est défini par l'axe qui passe par les deux points $P_0(x_0, y_0, z_0)$, $P_1(x_1, y_1, z_1)$ et le rayon R (voir figure 15) est l'ensemble des points $M(x, y, z)$ qui ont une distance R de l'axe, il est donné par la formule suivante :

$$R = \frac{\|\overrightarrow{P_0P_1} \wedge \overrightarrow{P_0M}\|}{\|\overrightarrow{P_0P_1}\|} \quad (4.18)$$

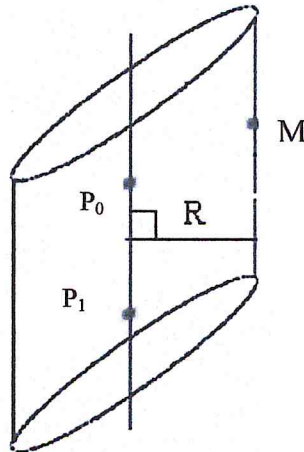


Figure 15. Cylindre dans l'espace.

En multipliant la formule (4.18) par $\|\overrightarrow{P_0P_1}\|$

$$\text{On trouve : } \|\overrightarrow{P_0P_1} \wedge \overrightarrow{P_0M}\| = R \times \|\overrightarrow{P_0P_1}\| \quad (4.19)$$

Le produit vectoriel $\|\overrightarrow{P_0P_1} \wedge \overrightarrow{P_0M}\|$ nous donne le vecteur suivant :

$$\overrightarrow{P_0P_1} \wedge \overrightarrow{P_0M} = \begin{pmatrix} (y_1 - y_0) \times (z - z_0) - (z_1 - z_0) \times (y - y_0) \\ -(x_1 - x_0) \times (z - z_0) + (z_1 - z_0) \times (x - x_0) \\ (x_1 - x_0) \times (y - y_0) - (y_1 - y_0) \times (x - x_0) \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (4.20)$$

Alors la norme de ce vecteur est calculée comme suit :

$$\|\overrightarrow{P_0P_1} \wedge \overrightarrow{P_0M}\| = \sqrt{X^2 + Y^2 + Z^2} \quad (4.21)$$

$$\|\overrightarrow{P_0P_1}\| = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2} \quad (4.22)$$

En remplaçant les formules (4.21) et (4.22) dans la formule (4.19), on trouve l'équation du cylindre qui a comme inconnu (x,y,z).

6.4. Sphère dans l'espace :

La sphère de centre $P_0(x_0, y_0, z_0)$, et de rayon R est définie par l'équation suivante :

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2 \quad (4.23)$$

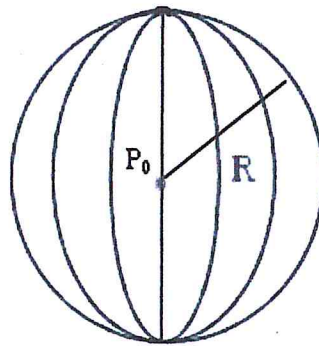


Figure 16 . Sphère dans l'espace.

7. Conclusion :

Nous avons décrit dans ce chapitre les différentes méthodes de simulation d'usinage et d'enlèvement de matière, ainsi que les avantages et les inconvénients de chaque méthode.

Après l'étude de toutes ces méthodes nous trouvons que la meilleure méthode est celle du point-vecteur, cette méthode s'applique sur l'ensemble de points qui représente la surface, Pour avoir ces points, Nous avons étudié une des méthodes de triangulation et de subdivision de la surface.

Pour avoir les points d'intersections qui résultent de cette méthode de simulation nous avons besoin de connaître les intersections des vecteurs sortants de la surface avec le trajet de l'outil.

1. Introduction:

En informatique la résolution d'un problème exige sa décomposition en une suite d'étapes successives pour arriver à donner une solution adéquate.

L'objectif de ce travail est la conception d'une application logicielle permettant la simulation d'enlèvement de matière lors de l'usinage des surfaces gauches sur des fraiseuses à commande numérique à 3 axes, et la détection des différentes zones usinées. Dans ce qui suit, on présente les différents algorithmes de l'application.

2. Structuration de l'application logicielle :

La démarche logique de résolution d'un problème donné, est la décomposition de celui-ci en des parties simples à résoudre. Dans le cas de notre travail, la résolution du problème de simulation passe par les étapes suivantes :

- Modèle mathématique de la surface à usiner ,
- Introduction des paramètres d'usinage et d'outil ;
- Calcul de la position d'outil ,
- Introduction des données de triangulation (distance maximale entre deux points dans un triangle, distance maximale entre un point sur le triangle et sa projection sur la surface) ;
- Calcul de la normale pour chaque point, et fixer sa longueur ,
- Introduction de la tolérance d'usinage (erreur tolérée) ;
- Triangulation adaptative de la surface ,
- Simulation d'enlèvement de matière, et calcul du volume enlevé ;
- Localisation des différentes zones ,
- Dessin du spectre d'erreurs ;
- Élimination des points d'interférence.

Une décomposition logique de notre application nous ramène vers l'organigramme suivant (Figure1) :

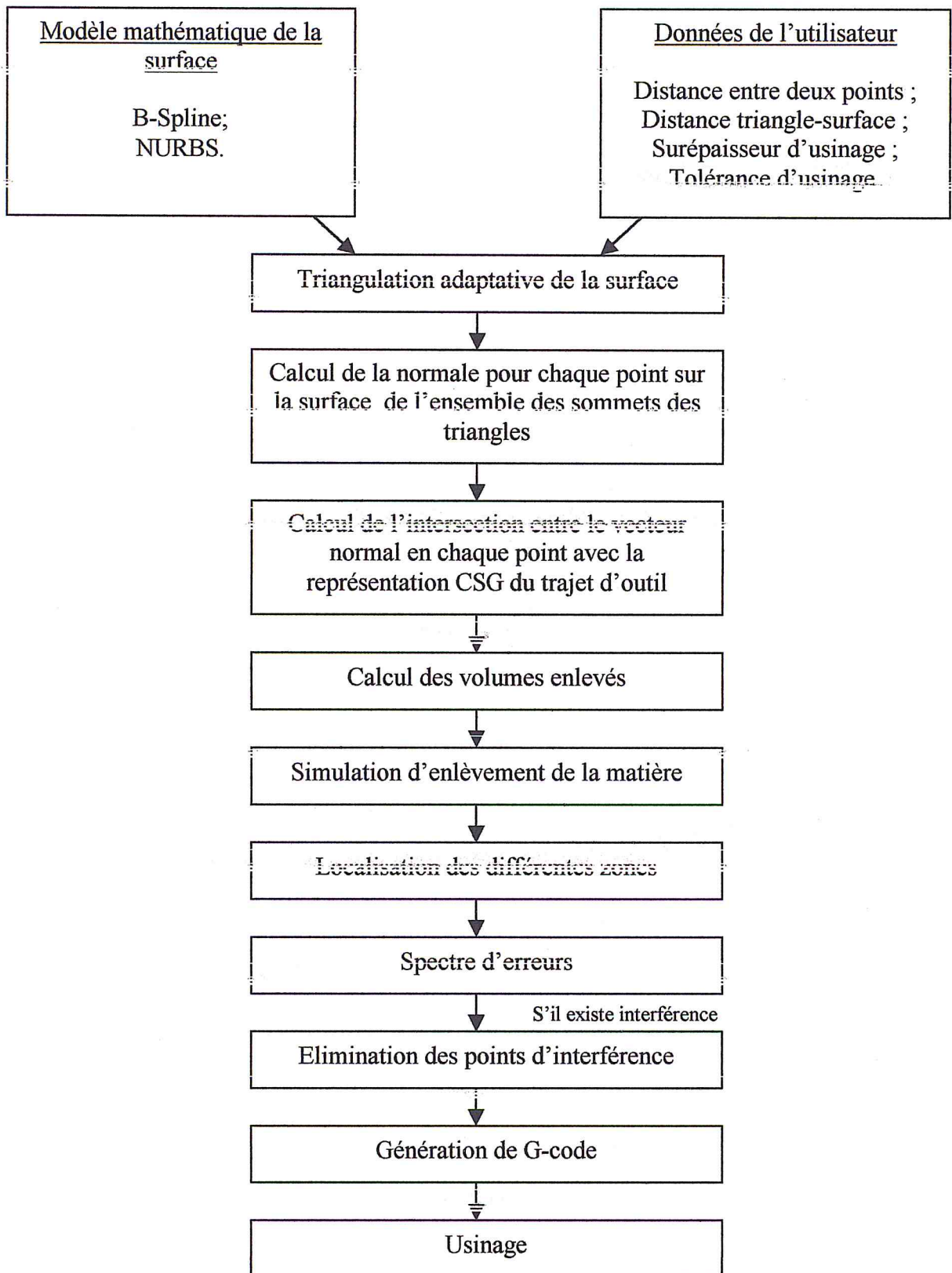


Figure 1. Vue générale de l'application.

3. Triangulation :

La première étape dans le processus de simulation est la triangulation des différentes surfaces. La triangulation adaptative est une triangulation plus efficace que la triangulation uniforme. Dans ce mode de triangulation, le nombre de triangles est créé d'une manière dynamique et adaptative à partir des critères de précision fixés par l'utilisateur. L'algorithme de la triangulation adaptative est le suivant :

Début

$V = \text{constante } (v_0)$

Diviser l'intervalle U en des sous intervalles $[u_0, u_1] \dots [u_{n-1}, u_n]$ telle que la distance $P(u_i, v_0) P(u_{i+1}, v_0)$ est constante.

$U = \text{constante } (u_0)$

Diviser l'intervalle V en des sous intervalles $[v_0, v_1] \dots [v_{m-1}, v_m]$ telle que la distance $P(u_0, v_j) P(u_0, v_{j+1})$ est constante.

calculer l'ensemble des points

$p(u_0, v_0) p(u_0, v_1) \dots p(u_0, v_m)$

$p(u_1, v_0) p(u_1, v_1) \dots p(u_1, v_m)$

.....

$p(u_n, v_0) p(u_n, v_1) \dots p(u_n, v_m)$

Former l'ensemble des triangles initiaux

Pour chaque triangle

Si un triangle ne vérifie pas l'un des critères de précision (distance et taille)

alors Diviser le triangle

Fsi

Fin pour

Fin

Pour chaque triangle on fait sept tests pour la longueur des cotés (les trois cotés ensemble, chaque deux cotés ensemble et chaque coté seul), sept tests pour la distance entre le centre de coté et sa projection sur la surface et un test pour le centre de coté.

Le résultat de la triangulation est un ensemble de triangles qui approximent la surface théorique et qui seront utilisés dans les étapes suivantes.

4. Calcul des vecteurs normaux :

Pour chaque sommet du triangle, on calcule le vecteur normal. Le début du vecteur sortant de la surface est le point sur la surface, mais la fin de ce vecteur dépend de l'épaisseur fixée par l'utilisateur, alors la longueur de ce vecteur n'est pas celle du vecteur normal mais la direction est la même.

Le calcul des normales en chaque sommet du triangle est donné par l'algorithme suivant :

Début

L'ensemble des triangles.

Pour chaque triangle

Calcule la normale $(P(u_0, v_0))$; // premier point du triangle.

Calcule la normale $(P(u_1, v_1))$; // deuxième point du triangle.

Calcule la normale $(P(u_2, v_2))$; // troisième point du triangle.

Fin pour

Fin.

5. Calcul d'intersection entre les normales et le volume balayé par l'outil :

Cette partie de l'application est la partie la plus importante. L'intersection entre la droite porteuse du vecteur sortant de la surface et le volume balayé par l'outil entre deux positions du trajet de l'outil peut nous donner plusieurs points mais qui nous intéresse, c'est le point le plus profond en direction de la surface.

Le volume balayé par l'outil est formé de deux plans, deux sphères, et trois cylindres. Les points d'intersections entre la normale et les différentes formes sont obtenus par la résolution des systèmes suivants :

- Droite-plan : (Voir figure 2) peut donner :
 - Ensemble vide ;
 - Un point ;
 - Une droite.

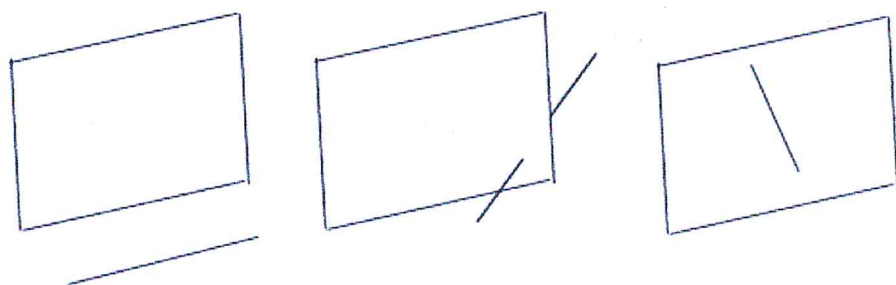


Figure 2. Intersection entre droite et plan.

- Droite-sphère : (Voir figure 3) peut donner :
 - Ensemble vide ;
 - Un point ;
 - Deux points.

Figure 3 . Intersection entre droite et sphère.

- Droite-cylindre : (Voir figure 4) peut donner :
 - Ensemble vide ;
 - Un point ;
 - Deux points.

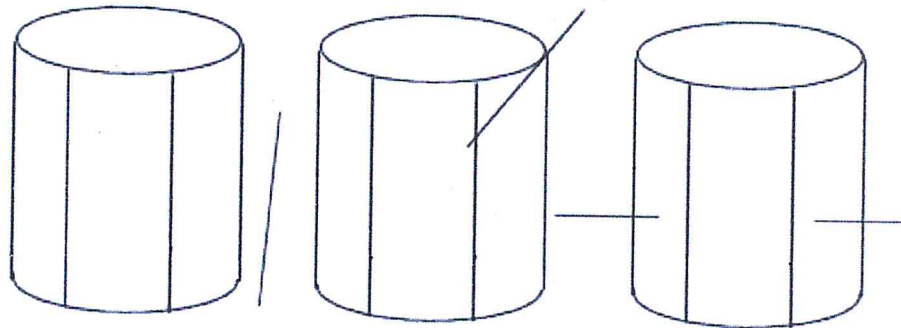


Figure 3 . Intersection entre droite et cylindre.

Pour avoir le point d'intersection le plus profond on doit suivre l'algorithme suivant :

Début

Pour (chaque deux points du trajet)

 Pour (tout point sur les surfaces)

 Intersection (les droites, les trois cylindres) ;

 Intersection (les droites, les deux plans) ;

 Intersection (les droites, les deux sphères) ;

 Fin pour

Fin pour

Fin.

6. Calcul du volume enlevé :

Le calcul est fait pour chaque deux positions de l'outil et pour tout point de trajet (volume total).

L'algorithme de calcul du volume enlevé est le suivant :

Début

Volume_totale= 0;

V0 = volume initial ; // Volume de la surface avant l'usinage.

Pour (chaque deux positions de l'outil)

Faire

V1 ;// Volume la surface après l'usinage.

V2 = V0 – V1 ; // Volume enlevé pour chaque deux postions.

Volume_totale= Volume_totale + V2 ; // Calcule le volume total enlevé.

Fait

Fin.

7. Localisation des différentes zones :

Une fois que la simulation est terminée, nous devons localiser les différentes régions qui sont :

- Zones en dessous de la tolérance : ce sont des points d'interférence ;
- Zones dans l'intervalle de tolérance ;
- Zones en dessus de l'intervalle de tolérance (excès de matière).

La détection de ces différentes zones est faite pour l'organigramme suivant :

Début

Pour (chaque point de surface)

\vec{P} = produit scalaire (vecteur (point sur surface, point résultat d'usinage), vecteur normal) ;
D = distance entre (point sur surface, point résultat d'usinage)

Si (D <= tolérance)

Alors

Point appartient à zone dans l'intervalle de tolérance ;

Si non.

Si (P > 0)

Alors

Point appartient à zone d'excès de matière ;

Si non

Point appartient à zone d'interférence ;

Fin Si ;

Fin Si.

Fin pour ;

Fin.

8. Spectre d'erreurs :

Le spectre d'erreurs est le résultat final de l'application. Après la localisation des différentes zones, le spectre d'erreurs nous indique, après la simulation, les zones qui sont bien usinées (c'est à dire dans l'intervalle de la tolérance d'usinage qui est définie par l'utilisateur) et les zones qui sont en dehors de l'intervalle de tolérance (usinage plus ou moins que voulu).

Le spectre donne les résultats suivants :

- = Zones dans l'intervalle de tolérance colorée en vert ;
- Zones en dessus de l'intervalle (excès de matière) de tolérance colorée en bleu ;
- Zones en dessous de l'intervalle (zone d'interférence) de tolérance colorée en rouge.

L'algorithme de calcul du spectre d'erreurs est le suivant :

Début

Pour (chaque point de surface)

Si (point appartient à zone dans l'intervalle de tolérance)

Alors

Couleur = vert ;

Fin Si ;

Si (point appartient à zone d'excès de matière)

Alors

Couleur = bleu ;

Fin Si ;

Si (point appartient à zone d'interférence)

Alors

Couleur = rouge ;

Fin Si ;

Fin pour ;

Fin.

9. Elimination des points d'interférence :

Cette partie est la dernière partie de notre application. On va corriger l'usinage par laisser les zones en rouge sans usinage, c'est-à-dire, quand l'outil arrive à une zone rouge, il monte en haut jusqu'au dessus de brute, puis il passe sans usinage, puis il redescend pour continuer l'usinage.

L'algorithme de correction d'usinage est le suivant :

Début

j = la position entre deux positions de l'outil ou il y a les zones en rouge.

Pour (i = chaque deux positions de l'outil)

Faire

Si ($i = j$) alors en laisser ces zones sans usinage.

Sinon

Pour (tous points sur les surfaces)

Faire

Intersection (les droits, les trois cylindres) ;

Intersection (les droits, les deux plans) ;

Intersection (les droits, les deux sphères) ;

Fait

Fsi

Fait

Fin.

10. Conclusion :

Nous avons décrit dans ce chapitre l'architecture générale de notre application logicielle relative à la simulation d'enlèvement de matière pour l'usinage des surfaces gauches.

Nous avons aussi présenté les différents algorithmes de la simulation, afin de permettre à l'utilisateur de détecter les différentes zones et leurs types ainsi que la correction du trajet par élimination des zones d'interférence.

1. Introduction:

Dans le présent chapitre, nous allons présenter le langage de programmation utilisé pour développer notre application à savoir le langage C++ Builder, il s'ensuit d'une autre présentation concernant la bibliothèque graphique OpenGL la plus utilisée pour l'implémentation des applications graphiques. Nous terminons ce chapitre par la présentation des différentes fenêtres de notre application et les résultats de génération des programmes de simulation d'usinage pour quelques surfaces obtenus à partir de notre application.

2. Environnement de développement:

2.1. Système d'exploitation:

Il existe plusieurs systèmes d'exploitation mais la majorité des utilisateurs développent les applications (logiciels) sous Windows, sachant que Windows offre de puissants outils d'interface que l'on peut exploiter. Nous avons implémenté notre application sous Windows qui sera supportée par les différentes versions de Windows 95, 98, Windows millenium, Windows NT ,Windows XP ou Windows 2003 .

2.2. Langage de programmation :

Tout développeur passe généralement par une étape qui est le choix du langage de programmation. Pour ce qui est de notre application, nous avons opté pour le langage C++ Builder avec son concept d'orientée objet.

Le C++ Builder est le nouveau produit de développement rapide d'application (RAD) de Borland C++ pour l'écriture d'applications. Les fonctionnalités les plus importantes de C++ Builder sont le fait qu'il travaille avec le langage C++, successeur du langage C utilisé par l'interface de programmation des applications Windows (API), et sa simplicité de programmation offerte aux programmeurs qui minimise considérablement le temps de programmation en offrant une bibliothèque riche de classes.

La POO (Programmation Orienté Objet) est une méthode d'implémentation dans laquelle, les programmes et les données sont organisés comme des ensembles coopératifs d'objets, où chacun représente une instance d'un certain ensemble de données et des fonctions, nommées classes.

2.3. Avantages de la Programmation Orientée Objet:

La programmation orientée objet (POO) avec C++ Builder offre à l'utilisateur un outil de programmation souple en l'aidant à concevoir des programmes de taille minimale et en lui garantissant la modularité, l'extensibilité et la réutilisation des composants du programme beaucoup plus aisée que dans les modes de programmation classique.

2.4. Programmation avec la bibliothèque OpenGL :

Toute application graphique requiert, à son plus bas niveau, l'interfaçage avec le matériel graphique utilisé. C'est le rôle d'une bibliothèque graphique.

OpenGL (Open Graphics Library) est une bibliothèque graphique standard qui regroupe un ensemble de primitives utilisées lors de l'implémentation des applications graphiques afin d'éviter tout interfaçage direct avec le hardware. Elle regroupe environ 200 commandes différentes qui permettent de spécifier un objet ou de développer des applications interactives en 3D.

D'une manière générale, les fonctions élémentaires de l'OpenGL se résument aux tâches suivantes:

- Création de formes géométriques à partir des primitives géométriques (OpenGL considère les points, les lignes et d'autres images comme étant primitives),
- Conversion de l'information concernant la description mathématique de l'objet et sa couleur en un ensemble de pixels sur l'écran,
- Rendu réaliste des scènes par définition des textures,
- Manipulation des scènes complexes,
- Représentation et manipulation des objets 3D.

3. Présentation de l'application :

Notre application nous permet la simulation d'enlèvement de la matière pour l'usinage des surfaces gauches sur des fraiseuses à commande numérique à 3 axes est une application graphique interactive développée en utilisant le C++ Builder et la bibliothèque graphique OpenGL. De ce fait, elle est composée d'interface qui permet à l'utilisateur de consulter et d'exécuter les différentes fonctions de la simulation d'enlèvement de la matière.

3.1. Interface utilisateur:

Notre interface utilisateur (figure 1) a pour rôle de guider l'utilisateur vers les différentes fonctions de l'application et d'apprécier en même temps le résultat de son travail.

Cette interface est composée de deux fenêtres:

- Une fenêtre principale qui contient le menu principal, les boutons de traitements et la barre d'état ;
- Une deuxième fenêtre dite «fenêtre de visualisation» sur laquelle sont affichées les différentes formes générées.

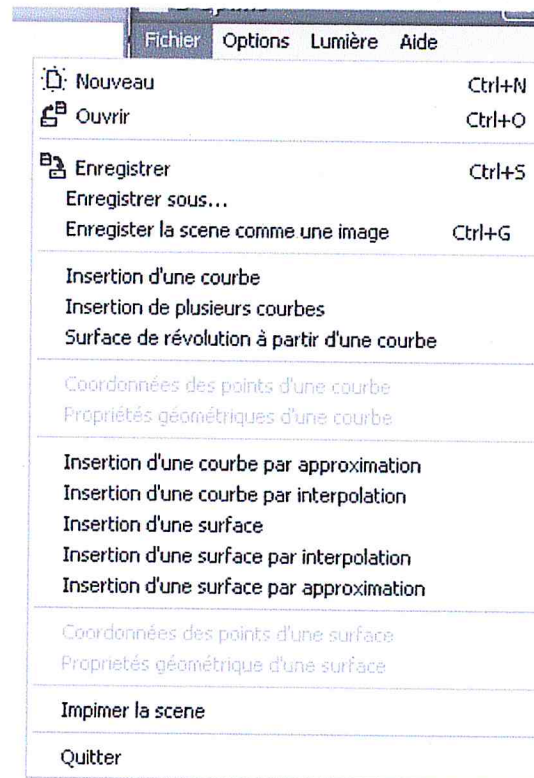


Figure 2 . Rubrique fichier.

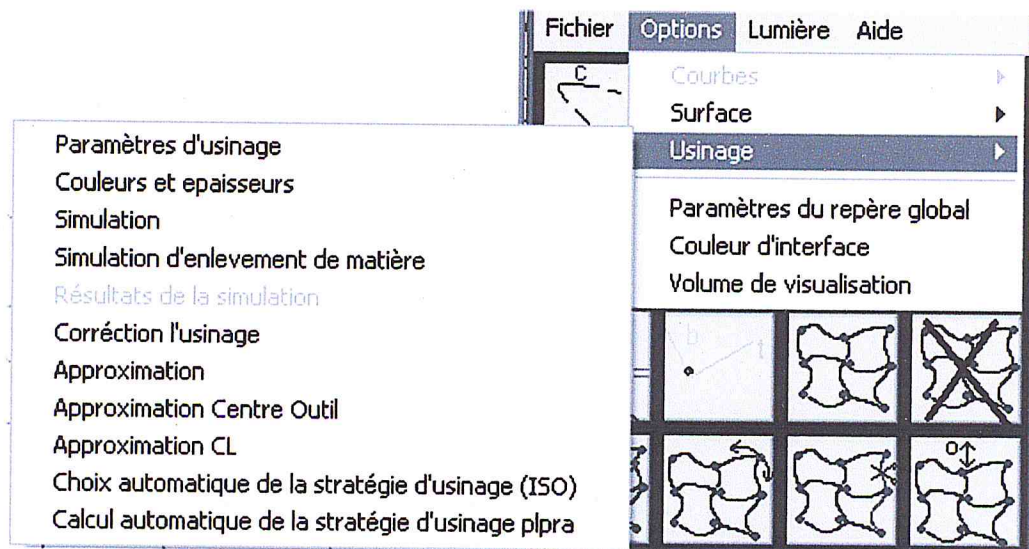


Figure 3 . Rubrique options.

Dans la rubrique " Simulation d'enlèvement de matière", l'utilisateur est invité à introduire les données (distance d'un segment, erreur de triangulation, épaisseur initiale, tolérance d'usinage) pour faire la triangulation adaptative et la simulation d'enlèvement de matière. L'utilisateur choisi la méthode manuelle pour introduire les données de triangulation, ou choisi la méthode automatique qui permet de calculer la longueur d'un segment et l'erreur de triangulation.

Nous avons donné la possibilité à l'utilisateur de choisir une méthode de calcul parmi les deux méthodes (exacte et simplifiée), type d'affichage (rendu ou filaire), et de visualiser chacune de ces parties : la surface nominale triangulée, la surface décalée triangulée, les normales, les spectres des erreurs, la première et la deuxième position d'outil, et les surfaces nominales à trianguler (figure 4). De même dans cette fenêtre, les coordonnées de la première et deuxième position d'outil sont affichées, ainsi que le calcul de volume enlevé et de choisir le temps de simulation. Dans la méthode de calcul « exacte » la surépaisseur d'usinage est calculée suivant la normale en chaque sommet d'un triangle, tandis que pour la méthode de calcul « simplifiée » la surépaisseur d'usinage est calculée suivant l'axe de Z en ignorant les composants x et y de la normale en chaque sommet du triangle.

La rubrique " spectre des erreurs " nous donne comme résultat trois zones, chaque zone est colorée par une couleur spécifiée, la zone en dessous de l'intervalle de tolérance est colorée en rouge, la zone en dessus de l'intervalle de tolérance est colorée en bleu et la zone dans l'intervalle de tolérance est colorée en vert, avec cette fonction on peut détecter les erreurs, tels que les zones colorées en rouge, ou les zones non usinées (en bleu).

Il est possible de modifier les données à n'importe quel moment même en cours d'exécution de la simulation.

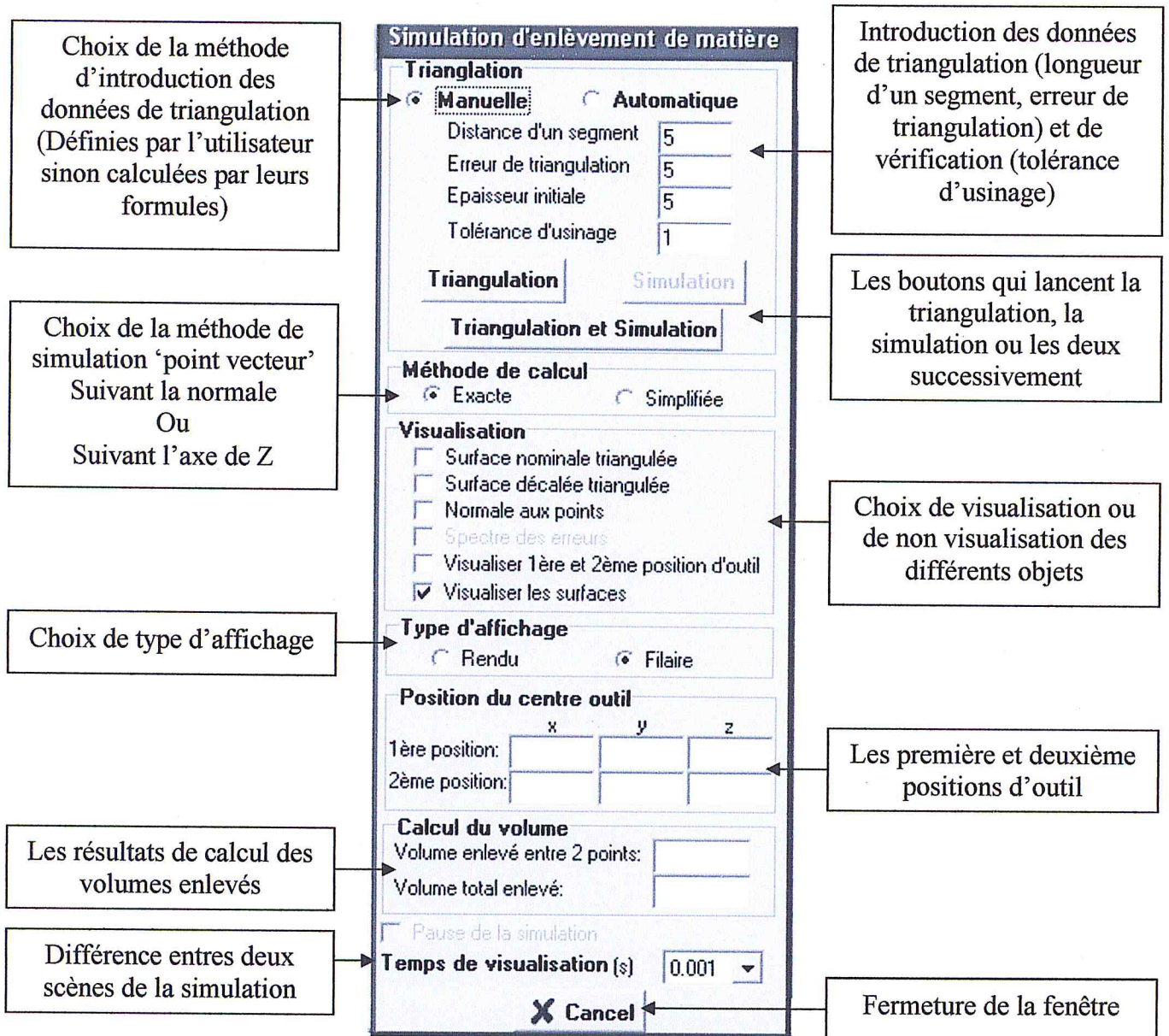


Figure 4 . Fenêtre de simulation d'enlèvement de matière.

Une fois la simulation d'enlèvement de matière est faite, la rubrique " résultat de la simulation " du menu options est activée, lorsqu'on clique sur cette dernière la fenêtre " résultat de la simulation " apparaît (voir figure 5). Dans cette fenêtre, nous pouvons visualiser les points de la zone en dessus de l'intervalle de tolérance, les points de la zone en dessous de l'intervalle de tolérance et les points dans l'intervalle de tolérance.

Pour les zones en dessous de l'intervalle de tolérance ou les zones d'interférences, on peut détecter toutes les positions d'outil qui causent ces interférences. À partir de cette donnée, on peut corriger éliminer toutes les positions d'outils qui engendrent les interférences.

Dans la cette fenêtre, pour chaque point sont affichées les erreurs correspondantes, ainsi que le numéro du trajet d'usinage.

Résultats de la simulation

Visualisation des points

- Points dans l'intervalle de tolérance
- Points en dessus de l'intervalle de tolérance
- Points en dessous de l'intervalle de tolérance
- Tout les points

Visualiser 1ère et 2ème position d'outil

X	Y	Z	Erreur	Position d'ou
53,53603	0,00050	3,53503	5,00000	0
51,40574	-3,98980	1,40533	1,98773	1

X	Y	Z	Erreur	Position d'ou
49,99997	-7,98000	-0,00004	-0,00005	2
49,99997	-11,97000	-0,00004	-0,00005	2
49,99997	-15,96000	-0,00004	-0,00005	2

X	Y	Z	Erreur	Position d'ou
44,17139	0,00000	5,45139	-0,02809	12
41,12718	0,00000	8,03906	-0,03590	14
37,99334	-99,75000	10,50974	-0,04404	19

Epaisseur du point: 1.5

Cancel

Donne les informations concernant les points de la zone bleue

Donne les informations concernant les points de la zone verte

Donne les informations concernant les points de la zone rouge

Visualisation n'importe quel type de points dans la fenêtre de visualisation

Taille de point à visualiser

Figure 5 . Fenêtre de résulta de la simulation.

Une fois le résultat de simulation est consulté, on passe à la correction de l'usinage (voir figure 6). Dans cette fenêtre, nous pouvons corriger l'usinage, tel que ne laisser que les zones bleues et vertes et éliminer les zones rouges.

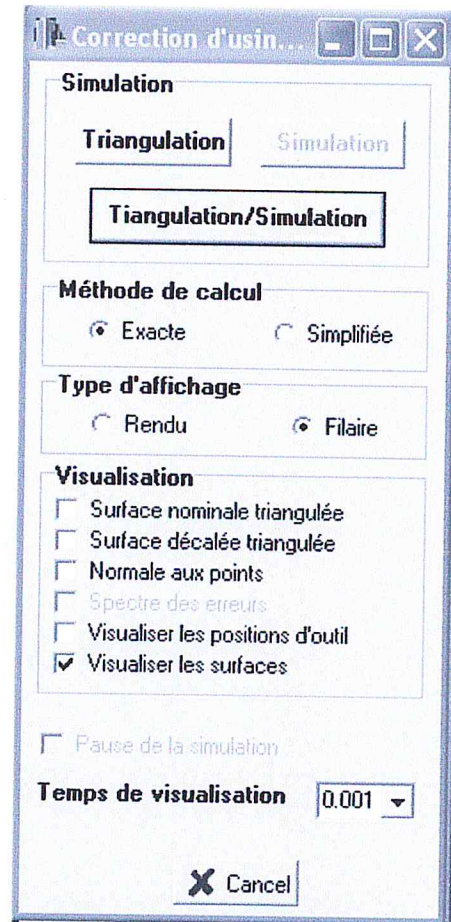


Figure 6 . Fenêtre correction d'usinage

4. Résultats et validations :

Dans cette partie, nous allons présenter deux exemples de la simulation, la première en utilisant la méthode exacte et la deuxième en utilisant la méthode simplifiée.

Nous avons choisis dans notre application la surface du demi vase (voir figure 7) avec les paramètres suivants:

- Le rayon d'outil égal à 15 millimètres ;
- La longueur maximum d'un segment de triangle égale à 5 millimètres ;
- L'erreur de triangulation égale à 5 millimètres ;
- L'épaisseur initiale égale à 5 millimètres ;
- La tolérance d'usinage égale à 0.5 millimètres.

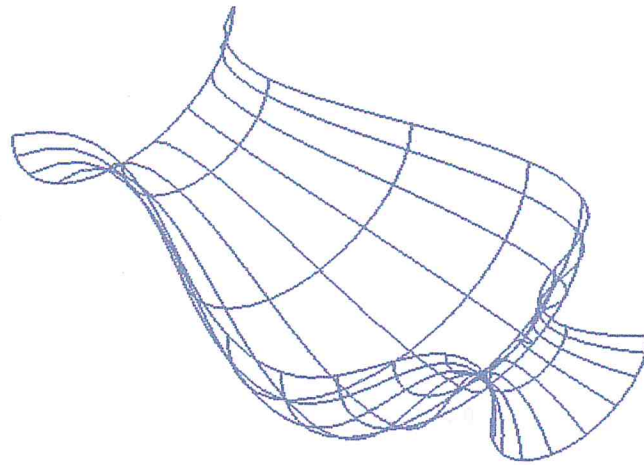


Figure 7 . Surface à usiner.

4.1. Exemple d'usinage avec la méthode exacte :

Les figures 8 montre la surface triangulée par la méthode exacte avec deux types d'affichage, la surface nominale est en bleu, les normales en rouge, la surface décalée en violet.

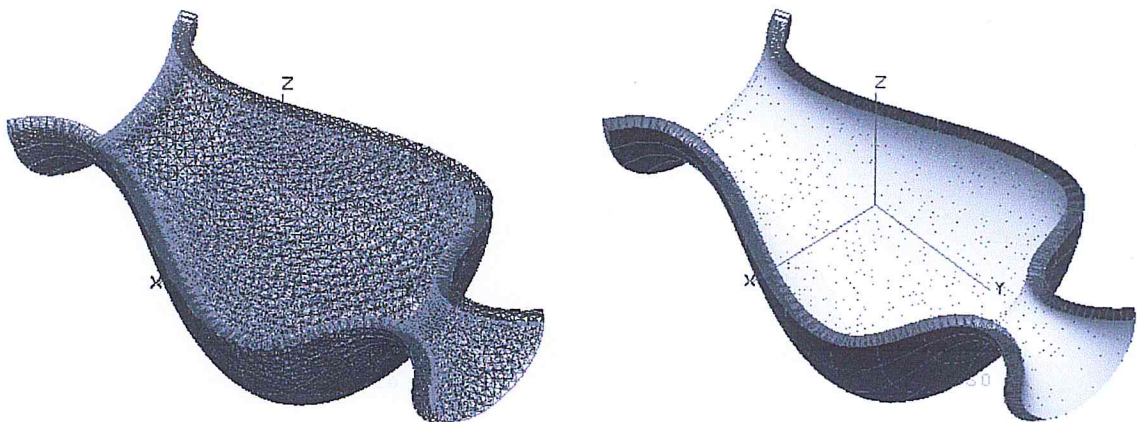


Figure 8 . Exemple d'une surface triangulée avec les types d'affichage filaire et rendu.

La figure 9 montre la surface en cours d'usinage. Elle montre aussi la première et deuxième position d'outil.

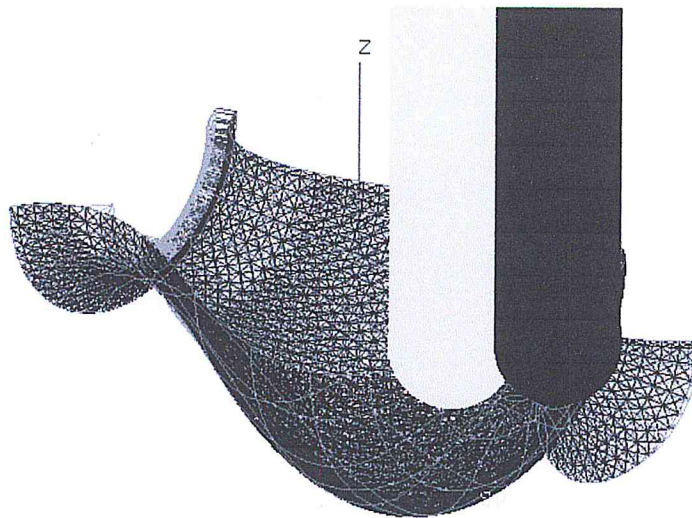


Figure 9 . Exemple d'une surface en cours de simulation d'usinage.

La figure 10 montre la surface après l'usinage.

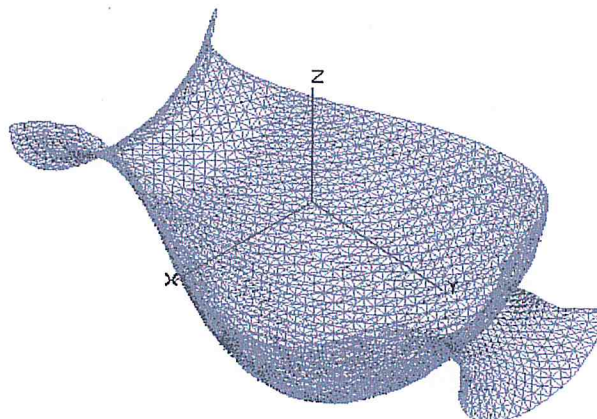


Figure 10 . Exemple d'une surface usinée avec type d'affichage filaire.

La figure 11 montre le spectre d'erreurs. On remarque que la zone de goulot (cou) de vase est en rouge (interférence).

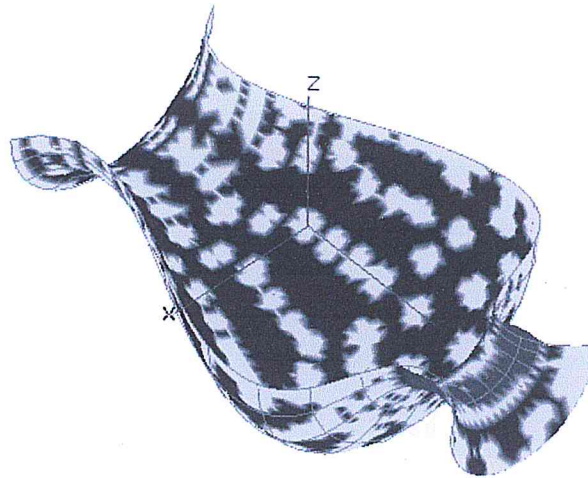


Figure 11 . Exemple d'une surface avec le spectre d'erreurs.

La figure 12 montre la correction d'usinage, en laissant la partie sans usinage afin de remplacer l'outil par un autre qui a un rayon plus petit.

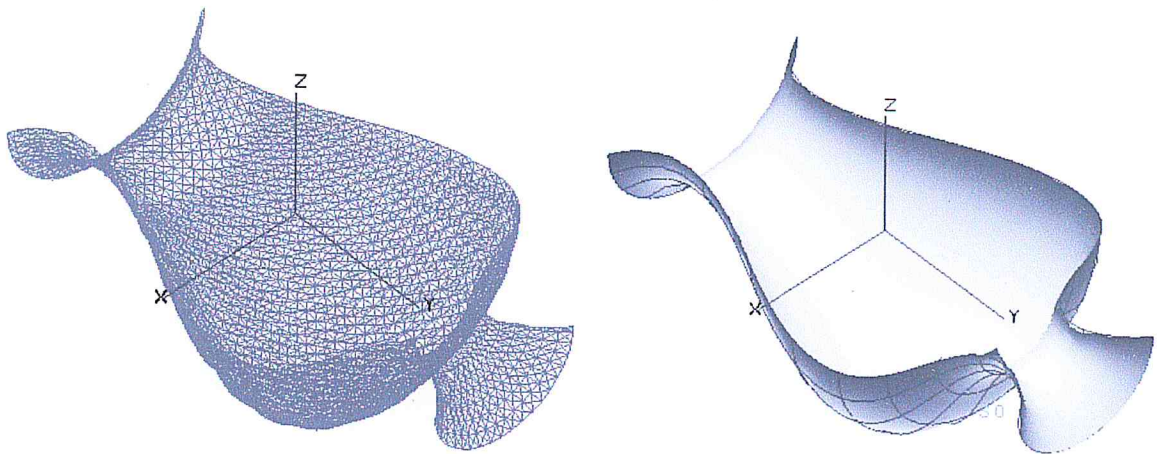


Figure 12 . Exemple d'une surface corrigée avec les type d'affichage filaire et rendu.

La figure 13 montre le spectre d'erreurs après la correction où nous constatons l'absence des zones rouges (interférence).

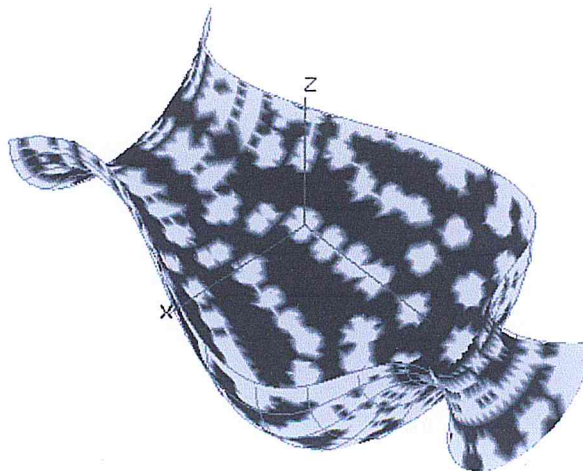


Figure 13 . Exemple d'une surface avec le spectre d'erreurs après la correction.

4.2. Exemple d'usinage avec la méthode simplifiée :

La figure 14 montre une surface triangulée par la méthode simplifiée.

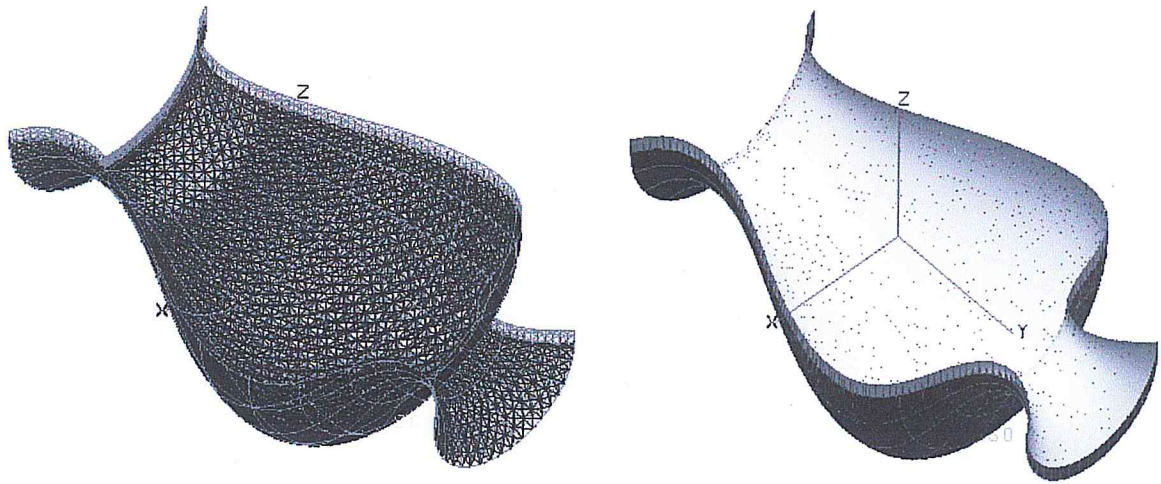


Figure 14 . Exemple d'une surface triangulée avec les types d'affichage filaire et rendu.

La figure 15 montre la surface après usinage.

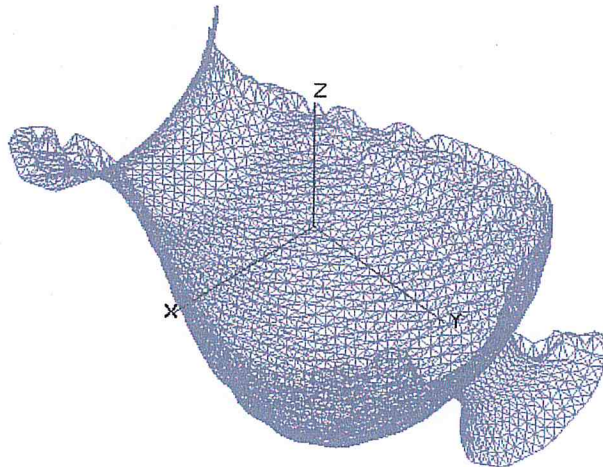


Figure 15 . Exemple d'une surface usinée avec type d'affichage filaire.

La figure 16 montre le spectre d'erreurs.

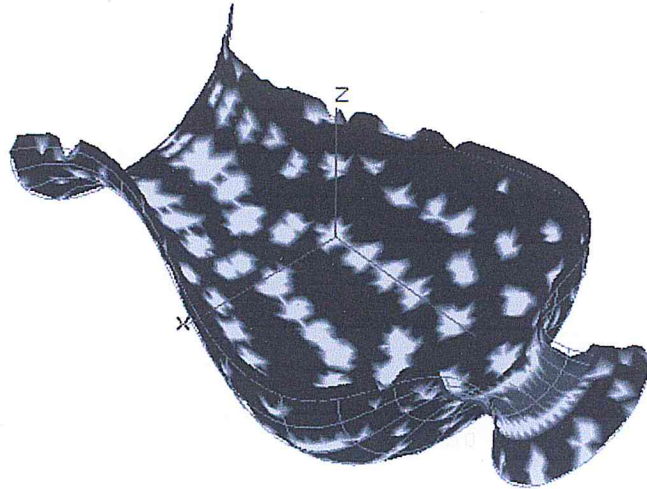


Figure 16 . Exemple d'une surface avec le spectre d'erreurs.

La figure 17 montre la correction des erreurs.

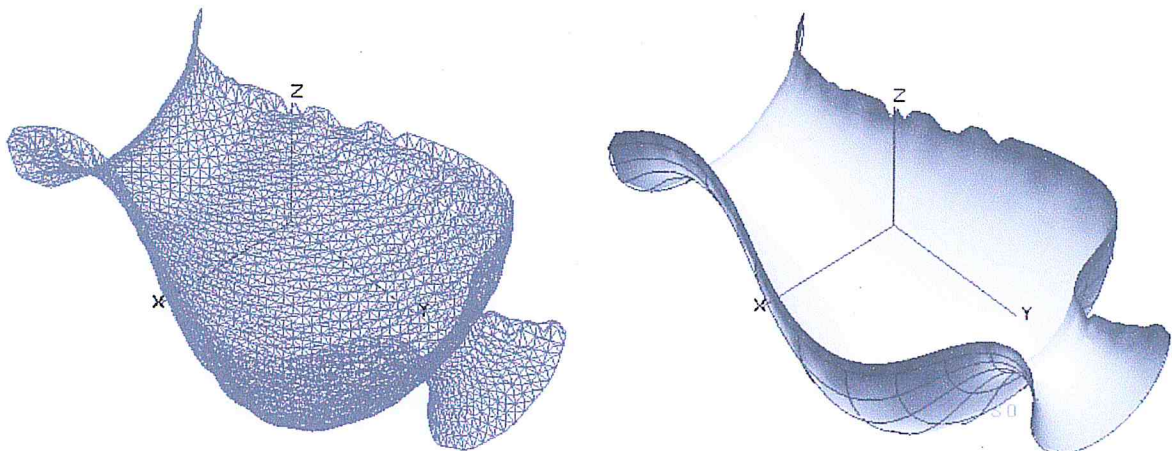


Figure 17 . Exemple d'une surface corrigée avec les types d'affichage filaire et rendu.

La figure 18 montre le spectre d'erreurs après la correction.

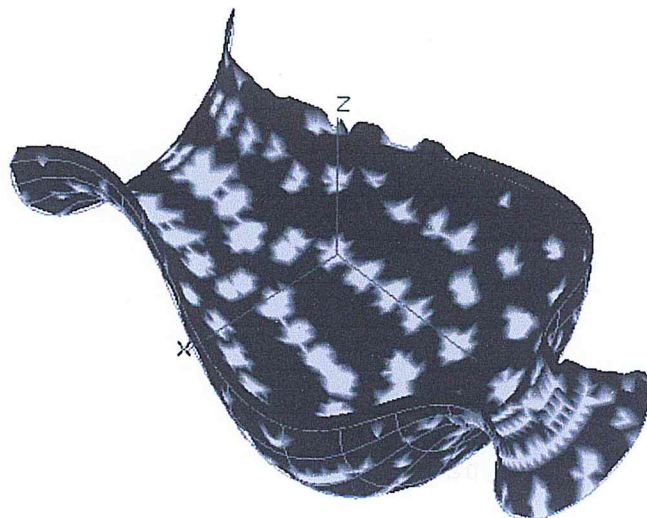


Figure 18 . Exemple d'une surface corrigée avec type d'affichage rendu.

L'annexe 2 montre un deuxième exemple de simulation d'usinage d'une autre surface avec les deux méthodes (exacte, simplifiée).

5. Conclusion :

Dans ce chapitre, nous avons présenté les différents outils informatiques utilisés pour l'implémentation de notre application. Après la présentation du système d'exploitation sous lequel nous avons travaillé, nous avons présenté le langage de programmation utilisé avec une brève définition de la bibliothèque graphique OpenGL. Les différentes fenêtres et actions de notre application, sont présentées avec des exemples d'application.

*C*onclusion générale

Le travail que nous avons présenté dans ce mémoire traite la simulation et la vérification des programmes d'usinage (G-Codes) des surfaces gauches sur des fraiseuses à commande numérique à 3 axes suivant la méthode point-vecteur.

Dans un premier temps, nous avons étudié le processus de conception des formes gauches, et nous avons analysé les différentes caractéristiques et propriétés de ces surfaces en appuyant notre analyse sur les surfaces B-Spline et NURBS. Puis nous avons étudié les principales parties qui forment la machine sur laquelle nous réalisons l'usinage. Pour cela nous avons étudié l'architecture des fraiseuses à commande numérique, ainsi que le corps et les principales fonctions du langage de programmation de ces machines.

Le résultat de notre travail est le développement d'une application logicielle graphique interactive sous Windows qui permet, à partir d'un fichier d'usinage G-code, de simuler et de vérifier virtuellement l'usinage des surfaces et de détecter automatiquement les zones de la tolérance d'usinage exigé, et éliminer tous les points d'interférence avant de passer à l'usinage effectif sur machine.

L'application développée permet de simuler l'opération d'enlèvement de matière lors de l'usinage de surfaces gauches et ce quelque soit la méthode d'usinage utilisée (isoparamétriques, plans parallèles, courbes de niveau ... etc.

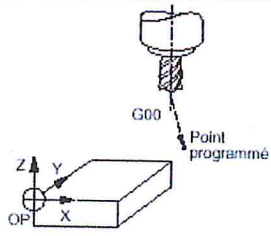
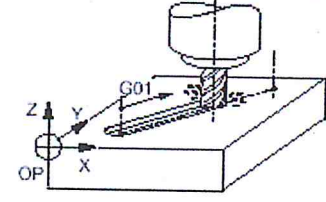
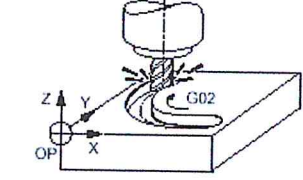
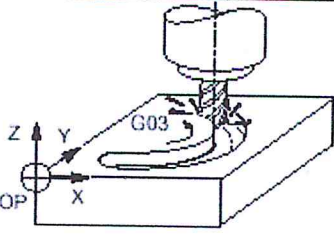

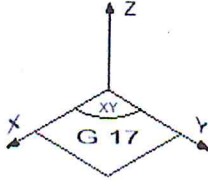
Ce travail nous ramène à conclure que l'informatique résout beaucoup de problèmes et fait bénéficier les autres domaines de l'automatisation des diverses fonctions. Son apport est considérable en terme du gain de temps comme dans notre situation la CFAO a permis aux mécaniciens d'éviter des dégâts matériels et des heures de travail devant la machine.

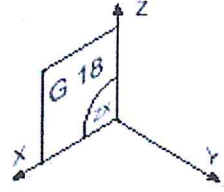
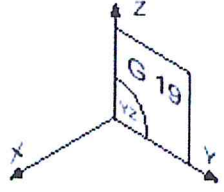
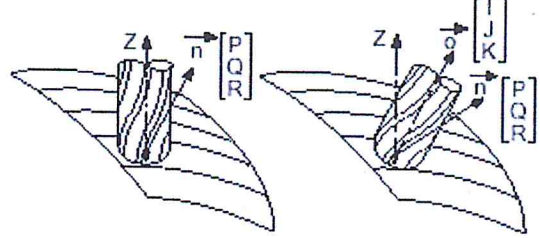
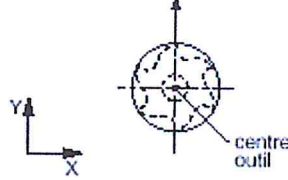
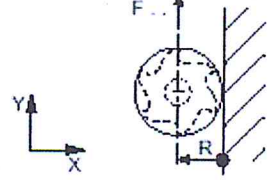
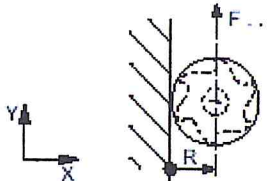

En perspective de notre travail, nous recommandons le traitement des points suivants:

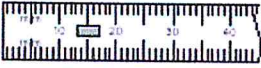
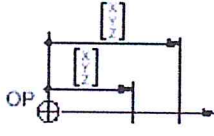
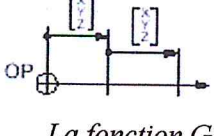
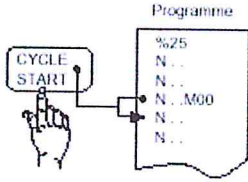
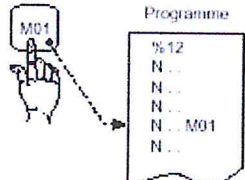
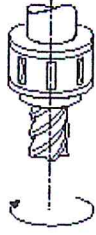
- Développement et implémentation des méthodes d'ébauchage des surfaces gauches ;
- Finition des surfaces gauches par les courbes de niveau ;
- Simulation virtuelle des mouvements d'une fraiseuse à commande numérique à 3 axes ;
- Traiter l'usinage des surfaces gauches sur des fraiseuses à commande numérique à 5 axes ;
- Choix automatiques des dimensions des outils.

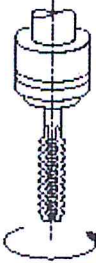
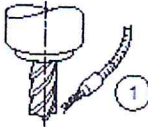
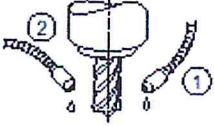
Tableau récapitulatif des fonctions du G-code

Le tableau suivant résume les principales fonctions d'un programme G-code:

Fonctions	Figure
<p>G00 : Interpolation linéaire à vitesse rapide. Fonction modale Syntaxe : N.. [G90/G91] G00 [□±] X.. Y.. Z..</p>	<p style="text-align: center;">Figure</p>  <p style="text-align: center;"><i>La fonction G00.</i></p>
<p>G01 : Interpolation linéaire à vitesse d'avance. Programmée Fonction modale Syntaxe : N.. [G90/G91] G01 [□±] X.. Y.. Z.. [F..]</p>	 <p style="text-align: center;"><i>La fonction G01.</i></p>
<p>G02 : Interpolation circulaire sens antitrigonométrique à vitesse d'avance programmée. Fonction modale Syntaxe (plan XY) : N.. [G17] [G90/G91] G02 X.. Y.. I.. J.. ou R.. [F..]</p>	 <p style="text-align: center;"><i>La fonction G02.</i></p>
<p>G03 : Interpolation circulaire sens trigonométrique à vitesse d'avance programmée. Fonction modale Syntaxe (plan XY) : N.. [G17] [G90/G91] G03 X.. Y.. I..J.. ou R.. [F..]</p>	 <p style="text-align: center;"><i>La fonction G03.</i></p>
<p>G04 : Temporisation programmable. Fonction modale Syntaxe : N.. G04 F..</p>	 <p style="text-align: center;"><i>La fonction G04.</i></p>
<p>G17 : Choix du plan XY Fonction modale Syntaxe : N.. G17</p>	 <p style="text-align: center;"><i>La fonction G17.</i></p>

<p>G18 : Choix du plan ZX Fonction modale Syntaxe : N.. G18</p>	 <p>La fonction G18.</p>
<p>G19 : Choix du plan YZ Fonction modale Syntaxe : N.. G19</p>	 <p>La fonction G19.</p>
<p>G29 : Correction d'outil dans l'espace 3 ou 5 axes. Fonction modale Syntaxe : N.. [D..] [G01] G29 X.. Y.. Z.. P.. Q.. R.. [I.. J.. K..] [A.. / B.. / C..]</p>	 <p>La fonction G29.</p>
<p>G40 : Annulation de correction de rayon Fonction modale Syntaxe : N.. [G00/G01] G40 X.. Y.. Z..</p>	 <p>La fonction G40.</p>
<p>G41 : Correction de rayon à gauche du profil à usiner Fonction modale Syntaxe (plan XY) : N.. [G17] [D..] [G00/G01/G02/G03] G41 X.. Y..</p>	 <p>La fonction G41.</p>
<p>G42 : Correction de rayon à droite du profil à usiner Fonction modale Syntaxe (plan XY) : N.. [G17] [D..] [G00/G01/G02/G03] G42 X.. Y..</p>	 <p>La fonction G42.</p>
<p>G70 : Programmation en pouce Fonction modale Syntaxe : N.. G70</p>	 <p>La fonction G70.</p>

<p>G71 : Programmation en métrique Syntaxe : N.. G71</p>	 <p><i>La fonction G71.</i></p>
<p>G90 : Programmation absolue par rapport à l'origine programme Fonction modale Syntaxe : N.. G90 X.. Y.. Z.. A.. B.. C..</p>	 <p><i>La fonction G90.</i></p>
<p>G91 : Programmation relative par rapport au point de départ du bloc Fonction modale Syntaxe : N.. G91 X.. Y.. Z.. A.. B.. C..</p>	 <p><i>La fonction G91.</i></p>
<p>M00 : Arrêt programmé . Fonction non modale Syntaxe : N.. [G40] M00 [\$0 ...]</p>	 <p><i>La fonction M01.</i></p>
<p>M01 : Arrêt programmé optionnel. Fonction modale Syntaxe : N.. [G40] M01 [\$0 ...]</p>	 <p><i>La fonction M02.</i></p>
<p>M02 : Fin de programme. Fonction modale Syntaxe : N.. M02</p>	
<p>M03 : Rotation de broche sens antitrigonométrique Fonction modale Syntaxe : N.. M03</p>	 <p><i>La fonction M03.</i></p>

<p>M04 : Rotation de broche sens trigonométrique Fonction modale Syntaxe : N.. M04</p>	 <p style="text-align: center;"><i>La fonction M04.</i></p>
<p>M05 : Arrêt de broche Fonction modale Syntaxe : N.. M05</p>	
<p>M08 : Arrosage numéro (1). Fonction modale avant Syntaxe : N.. M08</p>	 <p style="text-align: center;"><i>La fonction M08.</i></p>
<p>M09 : Arrêt des arrosages (1) et (2). Fonction non modale après Syntaxe : N.. M09</p>	 <p style="text-align: center;"><i>La fonction M09.</i></p>
<p>F : Avance, temporisation, nombre de filets Syntaxe : N.. G93 F.. (Avance en V/L). N.. G94 F.. (Avance en mm/min, degrés/min, pouce/min) N.. G95 F.. (Avance en mm/t, pouce/tour) N.. G04 F.. (Temporisation en secondes) N.. G31 F.. (Nombre de filets)</p>	
<p>S : Nombre de tours/minute, nombre de répétitions de sous programme Syntaxe : N.. G97 S.. (Vitesse de broche en tours/min. N.. G77 [H.] [N.. N..] S.. (Appel et répétitions de sous programme.</p>	
<p>T : Numéro d'outil Syntaxe : N.. T.. M00 (Appel de l'outil)</p>	

Exemple de simulation d'une pièce

Nous allons présenter dans cette annexe un exemple de simulation d'usinage d'une surface gauche avec les deux méthodes (exacte, simplifiée).

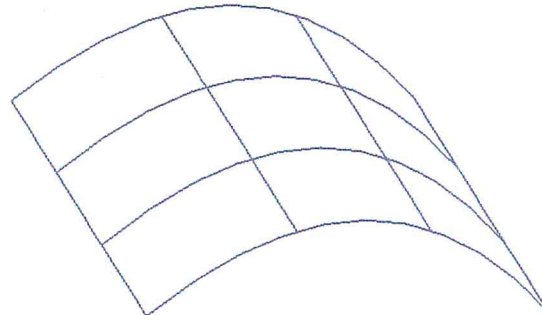


Figure 1 . Surface à usiner.

L'usinage et la simulation de cette pièce (demi cylindre) (voir figure 1) a les paramètres suivants:

- Le rayon d'outil égal à 10 millimètres ;
- La longueur maximum d'un segment de triangle égale à 5 millimètres ;
- L'erreur de triangulation égale à 5 millimètres ;
- L'épaisseur initiale égale à 5 millimètres ;
- La tolérance d'usinage égale à 1 millimètres.

1. Méthode d'usinage exacte :

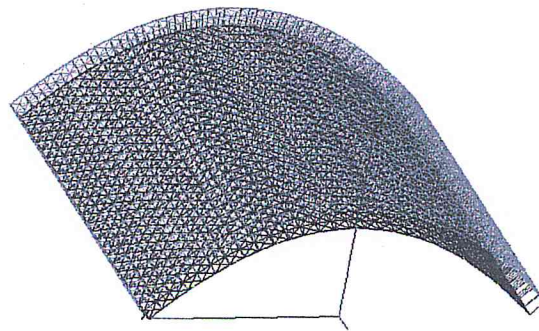


Figure 2 . Exemple d'une surface triangulée avec le type d'affichage filaire.

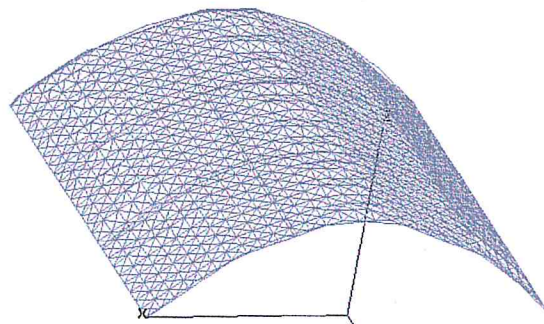


Figure 3 . Exemple d'une surface usinée avec type d'affichage filaire.

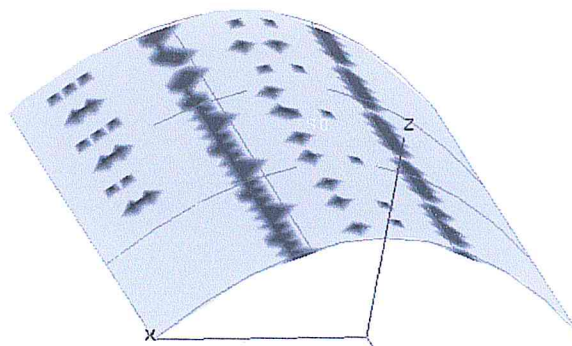


Figure 4 . Exemple d'une surface avec le spectre d'erreurs.

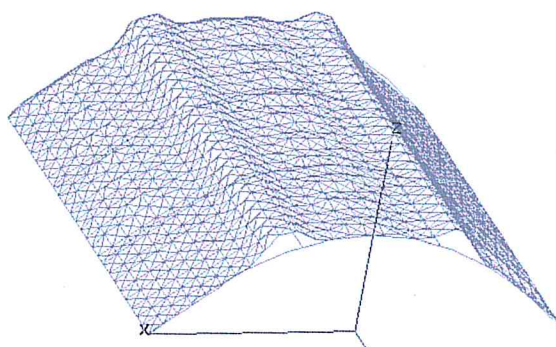


Figure 5 . Exemple d'une surface corrigée avec le type d'affichage filaire.

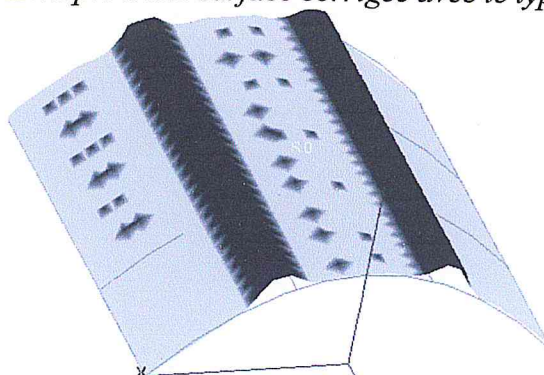


Figure 6 . Exemple d'une surface avec le spectre d'erreurs après la correction.

1. Méthode d'usinage simplifiée :

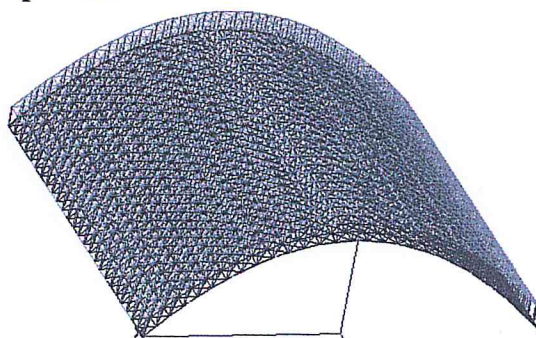


Figure 7 . Exemple d'une surface triangulée avec le type d'affichage filaire.

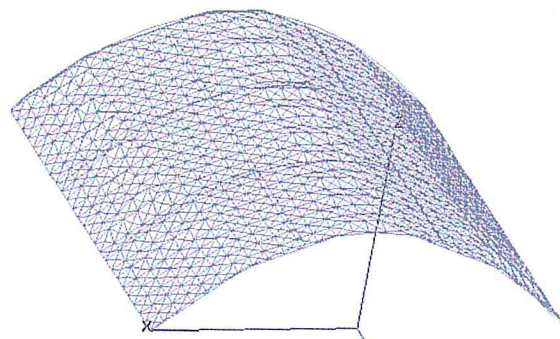


Figure 8 . Exemple d'une surface usinée avec type d'affichage filaire.

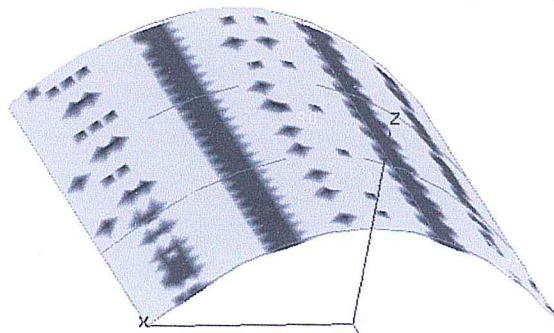


Figure 9 . Exemple d'une surface avec le spectre d'erreurs.

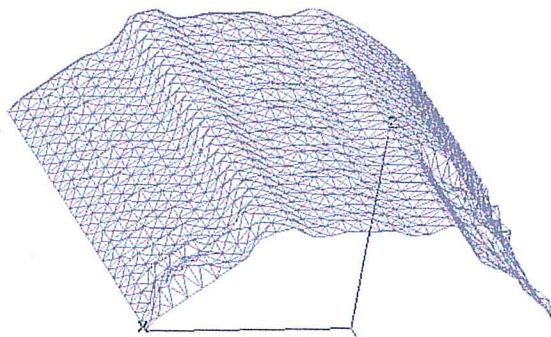


Figure 10 . Exemple d'une surface corrigée avec le type d'affichage.

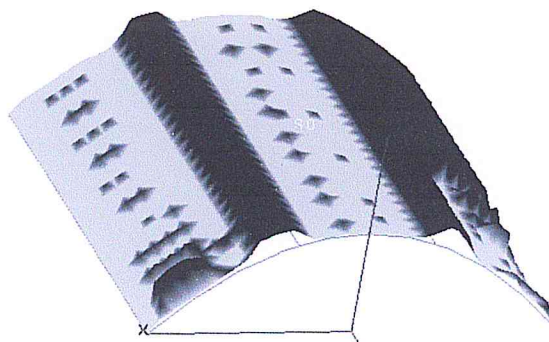


Figure 11 . Exemple d'une surface avec le spectre d'erreurs après la correction.

Références bibliographiques:

- [1] T.LIEBLING, H.RÖTHLISBERGER “*Infographie et applications* ” Editions MASSON.
- [2] Bey Mohamed “ *Modélisation des courbes et des surfaces* ” Rapport de Recherche 2000. CDTA .
- [3] Arezki LADJADI, Abdel Hamid REZZAZ “ *Conception et réalisation d’un outil de modélisation des courbes B-Splines et NURBS*” Mémoire du Projet de fin d’études 2001. Université des Sciences et de la Technologie Houari BOUMEDIENE.
- [4] C.K. Shene “ *Introduction to Computing with Geometry Notes*” Michigan Technological University.
- [5] Sofiane HOUBANE, Moussa YALAOUI “*Reconstruction des courbes et des surfaces B-Spline à partir d’un nuage de points*” Mémoire du Projet de fin d’études 2002. Université des Sciences et de la Technologie Houari BOUMEDIENE.
- [6] Mohamed GUARET, Fouad BOUABDELLAH “*Conception et Réalisation d’un Outil de Modélisation des Surfaces B-Spline et NURBS*” Mémoire du Projet de fin d’études 2002. Université des Sciences et de la Technologie Houari BOUMEDIENE.
- [7] BOUDJOUAD Salim, TAFAT-BOUZID Nazim “*Méthode des plans parallèles pour l’usinage des surfaces gauches sur des fraiseuses à commande numérique à 3 axes*” Mémoire du Projet de fin d’études 2004. Université des Sciences et de la Technologie Houari BOUMEDIENE.
- [8] “ NUM 1020/1040/1060 M, *Manuel de programmation –volume I*”. Société NUM 1996.
- [9] G.W.Vickers, M.H.Ly, R.G.Oetter “ *Numerically controlled machine tools* ” Ellis horwood.
- [10] Yabia Sahmi Azzedine Sarirette “ *Usinage surfaces gauches* ” Mémoire du Projet de fin d’études 2003. Université des Sciences et de la Technologie Houari BOUMEDIENE.
- [11] Abed Mohamed Benyoucef Nasr-eddine “ *Etude et conception d’une tête à aléser adaptée à une fraiseuse à commande numérique*” Mémoire du Projet de fin d’études 1995. Institut de mécanique Option : construction mécanique Université de Sâad Dahleb Blida.
- [12] F.Barachov “ *Le fraisage* ” Editions Mir Moscou 1975.
- [13] “ *Cours de fabrication mécanique de 3^{ème} année mécanique* ” 2003-2004 institut de mécanique Université de Sâad Dahleb Blida.
- [14] Emmanuel DUC. “ *Evaluation des modes de génération des trajets outil pour usinage de formes gauches* ”. Mémoire de DEA à l’école normale supérieur de Cachan, France, juillet 1996.
- [15] Emmanuel DUC. “ *Usinage de formes gauches; génération de trajets outil sans collision*”. Mémoire de DEA à l’école normale supérieur de Cachan, France, juillet 1996.

- [16] Robert B. Jerard, Robert L. Drysdale, Kenneth Hauck, and Barry Schaudt. “ *Discrete Simulation of NC Machining.*” University of New. Hampshire, Dartmouth College.
- [17] Robert B. Jerard, Robert L. Drysdale, Kenneth Hauck, and Barry Schaudt, John Magewick “*Sculptured Surfaces. Methods for Detecting Errors in Numerically Controlled Machining of Sculptured Surfaces.*” University of New. Hampshire, Dartmouth College, Ford Motor Company.
- [18] Robert B. Jerard, Robert L. Drysdale, Barry Schaudt. “ *Approximate methods for simulation and verification of numerically controlled machining programs.*” University of New. Hampshire, Dartmouth College.
- [19] Robert B. Jerard, Robert L. Drysdale, Kenneth Hauck, and Barry Schaudt, Peter Su “ *The Use of Surface Points Sets for Generation, Simulation, Verification and Automatic Correction of NC Machining Programs.*” University of New. Hampshire, Dartmouth College.
- [20] Risler J.-J. “ *Méthodes mathématiques pour la triangulation*”MASSON. coll. recherches en mathématiques appliquées. Paris, 1991.