

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

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département de génie informatique.



**Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.**
Option : Intelligence Artificielle (AI)

Sujet :

**Automatisation de l'opération
d'ébauchage des surfaces sur des
fraiseuses à commande
numérique à 3 axes**

Présenté par : Djamel Chabane
Lyes Mouterfi

Promoteur : Mohamed Bey
Encadreur : Hakim Acheli

Organisme d'accueil : CDTA (Centre de Développement des Technologies Avancées).
Division productique et robotique

Soutenue le: date soutenance, devant le jury composé de :

Nom. président

Nom examinateur 1,

Nom examinateur 2,

- 2004/2005 -

MIG-004-93-1



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿اللَّهُ نُورُ السَّمَاوَاتِ وَالْأَرْضِ مِثْلُ نُورِهِ كَمِشْكَاةٍ فِيهَا مِصْبَاحٌ الْمِصْبَاحُ فِي زُجْجَةٍ

الزُّجْجَةُ كَأَنَّهَا كَوْكَبٌ دُرِّيٌّ يُوقَدُ مِنْ شَجَرَةٍ مُبَارَكَةٍ زَيْتُونَةٍ لَا شَرْقِيَّةٍ وَلَا غَرْبِيَّةٍ يَكَادُ

رُئِيهَا بِيضٌ وَلَوْ لَمْ تَمْسَسْهُ نَارٌ نَوْراً عَلَى نُورٍ يَهْدِي اللَّهُ لِنُورِهِ مَنْ يَشَاءُ وَيَضْرِبُ اللَّهُ

سورة النور ﴿35﴾

الأمثال للناس والله بكل شيء عليم ﴿﴾

De par le nom de Dieu Tout-Miséricordieux, Tout-Compatissant

Dieu est la lumière des cieux et de la terre. Sa lumière est semblable à une niche où se trouve une lampe. La lampe est dans un verre pareil à un astre étincelant qui s'allume grâce à un arbre béni: un olivier qui n'est ni de l'orient ni de l'occident et dont l'huile brillerait sans qu'un feu la touche ou peut s'en faut. Lumière sur lumière.

Dieu dirige vers sa lumière qui il veut. Il propose aux hommes des pêches. Dieu connaît parfaitement toute chose.

[Sourat an-Nûr (la lumière) 35]

DEDICACE

Je dédie cette modeste réalisation à :
mes très chers parents que je ne remercierai jamais assez pour toute
l'aide et l'encouragement qu'ils m'ont continuellement apportés ;
Mon frère et mes sœurs qui m'ont également soutenu ;
Sans oublier ma petite TINA ;
A Djamel ainsi que tous mes amis ;
Et enfin à tous les assoiffés de savoir et de connaissance.

Lyes

A ma mère pour son amour et sa générosité...
A mon père pour son aide...
A ma YAYA pour sa générosité et ses encouragements ...
A mes frères.
A toute ma famille.
A Lyes,
A mes amis et frères pour leurs encouragement
sans oublier youssf pour sa bicyclette.
...
Etc...

Djamel

REMERCIEMENT

Nous remercions Dieu le Tout -Puissant qui nous a éclairé et donné la force et le courage de continuer afin que cet ouvrage voit le jour.

Au enseignants ; ceux qui nous ont aidée et encouragé durant nos études ; ainsi que notre promoteur.

Sommaire



INTRODUCTION GENERALE

Chapitre I : Notions sur les surfaces B-Spline & NURBS

| | |
|--|----|
| I. INTRODUCTION | 1 |
| II. CONCEPTION DES SURFACES | 3 |
| II.1. Méthodes basées sur les points..... | 3 |
| II.2. Méthode basée sur les courbes..... | 4 |
| III. REPRESENTATION DES SURFACES | 6 |
| III.1. Surfaces Non paramétrique..... | 6 |
| III.1.1. Forme explicite..... | 6 |
| III.1.2. Forme implicite..... | 6 |
| III.2. Surfaces Paramétrique | 6 |
| IV. PROPRIETES DES SURFACES PARAMETRIQUES | 7 |
| IV.1. Vecteur tangent et vecteur normal..... | 7 |
| IV.2. Les courbes iso-paramétriques..... | 8 |
| IV.3. Notion de Courbure..... | 8 |
| V. SURFACES B-SPLINE, ET SURFACES NURBS | 10 |
| V.1. Les surfaces B-Splines..... | 8 |
| V.1.1. Vecteur nodal | 10 |
| V.1.2. Propriétés des Surfaces de B-Spline..... | 13 |
| V.1.3. Les formes des surfaces B-Spline | 13 |
| V.2. Les surfaces NURBS..... | 14 |
| V.2.1. Propriétés des Surfaces de NURBS..... | 15 |
| VI. CONCLUSION | 15 |

Chapitre II : Architecture & programmation des MOCN

| | |
|--|----|
| I. INTRODUCTION | 16 |
| II. EVOLUTION HISTORIQUE DES MACHINES D'USINAGE | 16 |
| III. DEFINITION D'UNE MACHINE OUTIL | 17 |

| | |
|---|-----------|
| IV. ETUDE DE QUELQUE MACHINES OUTILS..... | 17 |
| V. DEFINITION ET CONSTRUCTION D'UNE MOCN | 19 |
| V.1. Définition d'une MOCN..... | 19 |
| V.2. Construction des fraiseuses..... | 20 |
| VI. LES AXES DE DEPLACEMENTS | 24 |
| VI.1. Mouvements de translations X, Y, et Z | 24 |
| VI.1.1. Axe de mouvement Z | 24 |
| VI.1.2. Axe de mouvement X | 24 |
| VI.1.3. Axe de mouvement Y | 24 |
| VI.2. Mouvements de rotations A, B, C..... | 24 |
| VI.3. Notion d'axe numérique, et demi axe numérique..... | 25 |
| VI.4. Différents types de fraiseuses..... | 25 |
| VI.4.1. Fraiseuses verticales | 25 |
| VI.4.2. Fraiseuses horizontales | 25 |
| VII. LA PROGRAMMATION DES FRAISEUSES | 26 |
| VII.1. Les différents modes de programmation. | 27 |
| VII.2. Programmation en langage ISO..... | 28 |
| VII.2.1. Format de bloc | 30 |
| VII.2.2. Structure générale d'un programme | 30 |
| VIII. L'APPORT DES MOCN | 31 |
| IX. CONCLUSION | 31 |

Chapitre III : Usinage des surfaces gauches

| | |
|---|-----------|
| I. INTRODUCTION | 32 |
| II. PRESENTATION DES PARAMETRES DE FRAISAGE | 32 |
| II.1. Types d'outils (fraises) | 32 |
| II.2. Modes d'usinage | 33 |
| II.2.1. Fraisage de face (en bout)..... | 33 |
| II.2.2. Fraisage de profil (en roulant)..... | 34 |
| II.3. Types de trajectoires..... | 35 |
| II.4. Positionnement de l'outil sur la pièce a usinée..... | 35 |

| | |
|--|-----------|
| II.5. Calcul de la position..... | 37 |
| III. TECHNIQUE D'USINAGE DES SURFACES GAUCHES EN FINITION | 38 |
| III.1. Alle simple (one way)..... | 38 |
| III.2. Aller retour (Zig-Zag)..... | 38 |
| III.3. Concentrique..... | 39 |
| III.4. Spiral in et Spiral out..... | 39 |
| III.5. Radiale..... | 40 |
| IV. AUTRES METHODES D'USINAGE..... | 40 |
| IV.1. Usinage par la méthode des plans parallèles..... | 40 |
| IV.2. Usinage par la méthode Z-Constant..... | 41 |
| V. DEFAULT D'USINAGE..... | 41 |
| V.1. Défaut de flèche | 41 |
| V.2. Défaut dû aux crêtes..... | 42 |
| V.3. Problème d'interférence..... | 42 |
| V.4. Problème de collision | 43 |
| VI. QUALITE D'USINAGE DES SURFACES GAUCHES..... | 43 |
| VII.CONCLUSION | 43 |
| Chapitre IV : Usinage des surfaces gauches en ébauche | |
| I. INTRODUCTION..... | 44 |
| II. APPROCHE ET STRATEGIE D'USINAGE DES SURFACES GAUCHES..... | 44 |
| III. DEFINITION DE L'EBAUCHE..... | 45 |
| IV. PROCEDURE D'EBAUCHE ET LES OUTILS UTILISE..... | 47 |
| IV.1. Sélection et classement de stratégie..... | 47 |
| IV.2. Sélection de l'outil pour l'usinage des cavités..... | 49 |
| V. LA TRIANGULATION | 49 |
| V.1. Les types de triangulation | 50 |
| V.1.1. La triangulation uniforme..... | 50 |
| V.1.2. La triangulation adaptative..... | 51 |
| VI. CALCUL DES INTERSECTIONS | 52 |
| VI.1. Plan dans l'espace..... | 53 |

| | |
|---|----|
| VI.2. Droite dans l'espace..... | 54 |
| VI.2.1. Appartenance d'un point à un segment de droite..... | 56 |
| VII.CONCLUSION | 57 |

Chapitre V : Conception de l'application

| | |
|--|----|
| I. INTRODUCTION..... | 58 |
| II. MODELISATION OBJET..... | 58 |
| II.1. Phase de modélisation objet..... | 58 |
| III. ETUDE D'UNE METHODE DE MODELISATION | 59 |
| IV. REALISATION DE L'APPLICATION | 59 |
| IV.1. Etablissement du cahier de charge | 59 |
| IV.1.1. Présentation du projet en globale..... | 59 |
| IV.1.2. Description du projet | 59 |
| IV.1.3. Définition du problème | 59 |
| IV.1.4. Les objectifs visés | 59 |
| IV.1.5. Description de la plateforme existante | 60 |
| IV.2. Solution de la problématique | 60 |
| IV.3. Modélisation de l'application en UML | 61 |
| IV.3.1. Diagramme de cas d'utilisation. | 62 |
| IV.3.2. Diagramme de classes. | 62 |
| IV.3.3. Diagramme de collaboration | 68 |
| IV.3.4. Diagramme d'activité..... | 69 |
| V.CONCLUSION | 70 |

Chapitre VI : Présentation de l'application

| | |
|--|----|
| I. INTRODUCTION. | 71 |
| II. INTERFACE DE CONCEPTION | 71 |
| II.1. Ecran d'accueil. | 71 |
| II.2. Interface d'usinage..... | 72 |
| II.3. Paramètres d'usinage | 73 |
| III. EXPOSITION DU TRAVAIL REALISER LORS DE CE PROJET..... | 74 |

| | |
|---|----|
| III.1. La fenêtre d'accueil d'ébauchage. | 74 |
| III.2. Lancement des opérations de calculs | 75 |
| III.2.1. Fenêtre de simulation. | 76 |
| III.2.2. Fenêtre programme G-Code..... | 76 |
| III.2.3. Fenêtre optimisation des outils. | 77 |
| III.2.3.1. En cas d'échec. | 77 |
| IV. Exemple d'application. | 78 |
| V. CONCLUSION. | 82 |
| CONCLUSION GENERALE. | 83 |
| Annexe I : Fonctions G-Code. | 84 |
| Annexe II : ETUDE DE LA METHODE DE MODELISATION « UML ». | 87 |
| Références bibliographiques. | 97 |

Chapitre I :

Figure I.1: L'approche globale et l'approche locale.

Figure I.2 : Les différentes méthodes de conception de surface.

Figure I.3 : Les vecteurs tangents, et le vecteur normal en un point de la surface.

Figure I.4 : Une surface composée de 08 courbes isoparamétriques 04 dans chaque direction (u, v).

Figure I.5 : La courbure d'une courbe en un point.

Figure I.6 : Surface B-Spline.

Figure I.7 : Mise en pratique des vecteurs nodaux (u, v).

Figure I.8 : Surface NURBS.

Chapitre II :

Figure II.1 : Machine Tour.

Figure II.2 : Centre d'usinage a 5 axes.

Figure II.3 : Une fraiseuse a 3 axes.

Figure II.4 : Les deux compartiments composant une **MOCN**.

Figure II.5 : Un Centre d'Usinage à commande numérique par ordinateur.

Figure II.6 : Une jauge d'outil.

Figure II.7 : Schéma général d'intercommunication du système de commande.

Figure II.8 : Architecture physique d'une commande numérique.

Figure II.9 : Trièdre formé par les trios axes (x, y, et z).

Figure II.10 : L'orientation positive des axes rotatif; elle est analogue au mouvement de rotation d'une vis dans le sens de vissage.

Figure II.11 : Vue générale d'un centre d'usinage METEOR 5 (Document PCI-SCEMM).

Figure II.12 : Synthèse des différents modes de programmation.

Figure II.13 : Structure d'un programme G-Code.

Figure II.13 : Structure d'un programme G-Code.

Figure II.14 : Structure d'un bloc.

Figure II.15 : Structure générale d'un programme d'usinage.

Chapitre III :

Figure III.1 : Les différent types de fraises à savoir les fraises hémisphériques (1), les fraises cylindriques (2), et les fraises toriques (3).

Figure III.2: Fraisage de face et le résultat donné.

Figure III.3: Fraisage de profil.

Figure III.4: Usinage en avalant.

Figure III.5 : Usinage en opposition.

Figure III.6 : Présentation des différents trajets suivis.

Figure III.7 : Déplacement paraxial.

Figure III.8 : Position de la fraise sur la surface.

Figure III.9 : Enveloppe de la surface a usinée.

Figure III.10 : Schéma de calcul de la trajectoire d'usinage.

- Figure III.11 : Usinage en aller simple.
- Figure III.12 : Usinage en aller retour.
- Figure III.13 : Usinages par la méthode concentrique.
- Figure III.14 : Usinage en Spiral in, et Spiral out.
- Figure III.15 : Usinage avec la méthode Radiale.
- Figure III.16 : Méthode des plans parallèles.
- Figure III.17 : Méthode Z-Constant.
- Figure III.18 : Erreur de flèches.
- Figure III.19 : Hauteurs de Crêtes.
- Figure III.20 : Interférence locale, et Interférence globale.
- Figure III.21 : Collision entre outil et surface

Chapitre IV :

- Figure IV.1: Ebauchage en utilisant un outil cylindrique ou d est profondeur de coupe.
- Figure IV.2: Ebauchage en utilisant un outil hémisphérique.
- Tableau IV.3 : Les différentes procédures d'usinages et les outils utilisés.
- Figure IV.4: Codes de classification des différentes formes possibles, et les outils adéquats à leur usinage.
- Figure IV.5: Triangulation d'une forme quelconque.
- Figure IV.6: Triangulation uniforme d'une surface ondulée.
- Figure IV.7: Triangulation uniforme d'une surface ondulée.
- Figure IV.8: Les différents cas d'anomalies dans la triangulation uniforme.
- Figure IV.9 : Les différents cas d'intersections d'un plan avec un triangle.
- Figure IV.10 : Intersection d'un plan passant par les deux points P_0, P_1 et le plan XY.
- Figure IV.11: Représentation d'une droite dans l'espace.
- Figure IV.12 : Position des différents points par rapport à AB.

Chapitre V:

- Figure V.1 : schéma représentant la modélisation d'objet
- Figure V.2 : extraction des points de passage de l'outil.
- Figure V.3: exposition des deux méthodes d'usinage
- Figure V.4: Diagramme de cas d'utilisation.
- Figure V.5: Diagramme de classes.
- Figure V.6 : Diagramme de collaboration.
- Figure V.7: Diagramme de d'activités.

Chapitre VI

- Figure VI.1: Interface de l'écran d'accueil.
- Figure VI.2: Interface de passage à l'étape d'usinage.
- Figure VI.3: Fenêtre des Paramètres d'usinage.
- Figure VI.4: Fenêtre passage à l'opération d'ébauchage.
- Figure VI.5 : Fenêtre de paramétrage de l'opération d'ébauchage.

- Figure VI.6 : Fenêtre de paramétrage de l'opération d'ébauchage (Suite).
- Figure VI.7 : Fenêtre de simulation de l'opération d'usinage
- Figure VI.8: Fenêtre de génération du programme G-Code.
- Figure VI.9 : Fenêtre de optimisation des outils.
- Figure VI.10 : Message en cas d'échec du calcul d'optimisation.
- Figure VI.11 : fenêtre de réalisation de la fonction débauchage (à gauche).
- Figure VI.12 : Configuration des paramètres d'ébauchage.
- Figure VI.13 : Affichage des données.
- Figure VI.14 : Affichage de la simulation.
- Figure VI.15 : Génération du programme G-Code.
- Figure VI.16 : Fenêtre choix automatique des outils.
- Figure VI.17 : Affichage des rayons optimums pour chaque plan.
- Figure VI.18 : interface base de données d'outils.

Annexe II :

- Figure.1: composant graphique du cas d'utilisation.
- Figure.2 : schéma représentant cas d'utilisation d'un distributeur de billets.
- Figure.3 : schéma représentant une classe.
- Figure.4: Définition des attributs et des méthodes.
- Figure.5: schématisation des liens entre les classes.
- Figure.6 : exemple d'un diagramme de classes.
- Figure.7 : relation entre diagramme de classes et diagramme d'objets.
- Figure.8 : exemple d'un diagramme d'objet cas de boîte de dialogue.
- Figure.9 : Diagramme de séquence (cas d'un DAB).
- Figure.10: Diagramme de collaboration (cas d'un distributeur de boisson).
- Figure.11: composants d'un diagramme d'activité.
- Figure.12 : Diagramme d'activité (cas d'une communication).
- Figure.13 : composants d'un diagramme de transition d'état.
- Figure.14 : Diagramme de transition d'état (cas d'une montre).
- Figure.15 : composants d'un diagramme de composant.
- Figure.16 : exemple d'un diagramme de composant cas décodage d'un fichier JPEG.
- Figure.17 : Diagramme de déploiement avec composants.

Résumé

Le présent mémoire fait partie du protocole de recherche intitulé "*l'usinage des surfaces gauches*" initié par l'équipe Conception et Fabrication Assistées par Ordinateur (CFAO) au niveau de la Division Robotique et Productique du Centre de Développement des Technologies Avancées (CDTA).

Dans notre PFE nous allons détaillé la phase de fabrication des surfaces gauches sur des machines outils a commande numérique, en exploitant la phase de conception de ce genre de surface. Notre travail repose sur le développement d'une application graphique interactive adapté a l'utilisation même par les gents qui ne sont pas du domaine. Cette application nous permet de réaliser: l'opération d'ébauchage des surfaces gauches en se basant sur des méthodes spécifiques (One-Way ou en Zig-Zag), simuler virtuellement les mouvements de l'outil de la machine par rapport à la surface et de calculer les temps d'usinage, choisir automatiquement les outils(en exploitant une base de données à outils) pour usiner chaque plan sans avoir des interférences, et le choix optimisé d'un ensemble d'outils qui ne fait gagner du temps dans l'usinage d'une surface quelconque.

INTRODUCTION GENERALE

Le niveau de vie de la société s'est largement amélioré ces dernières années. En effet, l'homme moderne est à priori en quête de confort, pour satisfaire cette condition sine qua non plusieurs industries ont été contraintes à augmenter la quantité et améliorer la qualité de leurs produits.

De ce fait, les grands industriels du monde n'ont pas hésité à consacrer des budgets colossaux afin de trouver de nouveaux procédés technologiques.

L'introduction de l'outil informatique dans plusieurs domaines a apporté une aide qui est devenue indispensable au fil du temps et qu'on peut observer surtout dans les disciplines suivantes: EAO (enseignement assisté par ordinateur), DAO (dessin assisté par ordinateur), CFAO (conception et fabrication assistées par ordinateur).

L'aérospatial, l'industrie automobile ainsi que l'industrie d'outillage mécanique s'appuient sur la puissance de la CFAO, qui apporte flexibilité et souplesse dans la conception et la fabrication des pièces de complexités diverses. L'avantage majeur de cette discipline est sa capacité de représenter les formes gauches tel que les moules des ailes d'avion, des véhicules, ... etc.

Notre apport se résume à la réalisation de l'opération d'ébauchage des surfaces gauches dans le cadre d'un protocole de recherche intitulé " *Usinage des surfaces gauches* " de l'équipe Conception et Fabrication Assistées par Ordinateur (CFAO) du Centre de Développement des Technologies Avancées (CDTA).

Le présent mémoire est une continuité des travaux précédents traitants 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. En plus de l'usinage des surfaces gauches de formes libres sur des fraiseuses à commande numérique à 3 axes en considérant différentes stratégies d'usinage pour la phase de finition, ensuite la simulation de l'opération d'usinage. Dans notre cas, on va automatiser l'opération d'ébauchage des surfaces gauches sur des machines outils à commande numérique (MOCN) à (03) axes.

L'élaboration du programme d'usinage « G-Code » (langage propre au MOCN) est nécessaire afin de générer une séquence de commandes numériques, pour la fabrication des formes gauches, nécessite plusieurs centaines de lignes de code, ce qui n'est pas très pratique et n'est facile. Donc l'objectif de notre travail est le développement d'une application logicielle graphique et interactive sous Windows capable de réaliser automatiquement l'opération de l'ébauche des surfaces gauches suivant des méthodes en plans parallèles horizontaux avec des passes en Zig-Zag et en One-Way avec la génération automatique du G-Code. En plus de cette étape, nous devons automatiser le choix des outils afin de minimiser les temps d'usinage et par conséquent les coûts de fabrication.

Le présent mémoire comprend (06) chapitres complémentaires présentés comme suit :

Chapitre I : dans ce chapitre nous allons étudier les différentes méthodes de conception et de représentation de ces surfaces, ainsi que leurs propriétés.

Chapitre II: nous allons décrire dans ce chapitre les outils utilisés dans l'usinage des surfaces gauches, plus précisément on va étudier la construction et la programmation des MOCN (Machines Outils à Commande Numérique).

Chapitre III : ce chapitre est consacré aux techniques d'usinage des surfaces gauches en finition.

Chapitre IV : l'opération d'ébauchage des surfaces gauches sera détaillée dans ce chapitre.

Chapitre V : dans ce chapitre nous allons donner une représentation simplifiée de notre système (la phase de modélisation), afin de mieux comprendre son fonctionnement.

Chapitre VI : le but dans ce chapitre est de montrer notre application, ainsi qu'une illustration par un exemple.

Enfin, nous terminons ce mémoire par une conclusion générale et quelques recommandations pour la continuité de ce travail.

Chapitre I

*Notions sur les surfaces
B-Spline & NURBS.*

I. INTRODUCTION :

La CAO (Conception Assistée par Ordinateur / (CAD) Computer Aided Design) est une discipline qui touche de plus en plus de grands domaines d'activités. Elle est utilisée dans la conception des pièces avant leurs usinages.

L'un des aspect les plus fondamentaux de la CAO est l'emploi de techniques multiples pour la représentation des objets solides et des surfaces régulières; surfaces de formes simples, mais son utilisation dans la représentation des surfaces gauches; surfaces de formes libres, se fait chez les grands constructeurs (Automobile, Aéronautique ...) a causé des contraintes de représentation.

Les travaux dans le domaine de l'automobile par Coons (Ford), Bézier (Renault) et de Casteljau (Citroën) et autres sont à la base des systèmes d'aujourd'hui.

II. CONCEPTION DES SURFACES [4,5]:

Comme son nom l'indique c'est une science spécialisée dans la modélisation de surfaces ; n'importe quel concepteur 'Designer' n'est pas censé de connaître les différentes équations mathématiques de la surfaces pour la dessiner, son travail est de concevoir des formes quelconques, c'est pourquoi on est obligé de passer par la maquette qui est une représentation fidèle de l'objet à réaliser dans une échelle réduite pour être scannée (numérisée par un Scanner) pour prélever un ensembles de points ou de courbes.

La nécessité d'appliquer des méthodes de CAO avancées pour l'usinage des surfaces se présente dans la fabrication de moules, de matrices et les formes esthétiques, etc.... La plupart de ce type de surface consiste en une combinaison de surfaces complexes qui sont difficiles à concevoir et coûteuses.

La demande industrielle croissante a motivé le développement des systèmes assistés par ordinateurs utilisés dans la conception et la fabrication des surfaces gauches.

Dans la conception des surfaces, deux classes de méthodes sont utilisées :

- Méthodes basées sur les points;
- Méthodes basées sur les courbes.

II.1. Méthodes basées sur les points :

Cette méthode consiste a traité un ensemble fini de points qui représente le mieux la surface, pour cela deux approches sont possibles; approche globale et approche locale. La (Figure I.1) montre les deux approches.

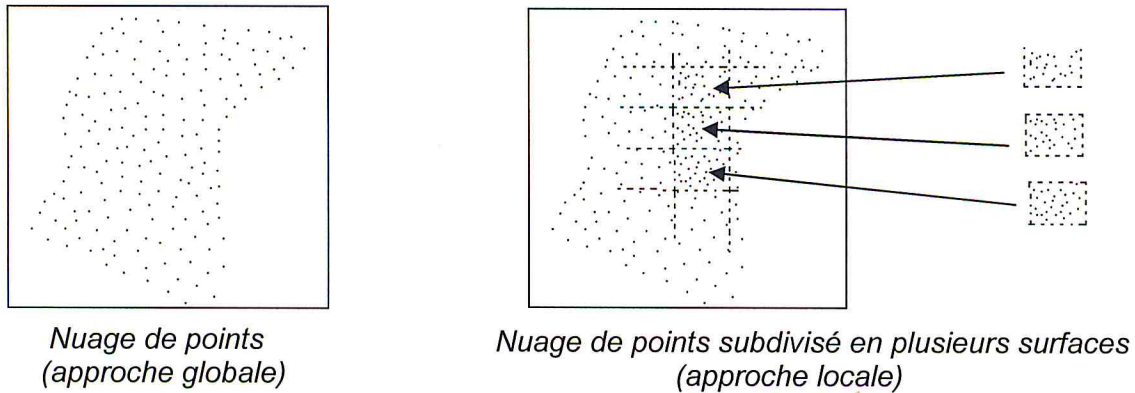


Figure 1.1 : L'approche globale et l'approche locale.

La classe de méthodes basées sur les points s'appuie sur un certain nombre de méthodes :

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

II.2. Méthodes basées sur les courbes :

Cette méthode est différente de la première car les courbes sont l'unité principale dans la construction des formes. Elle consiste à créer quelques courbes primaires ensuite créer toute la surface en se basant sur ces dernières. Elle est aussi nommée méthode de conception de surface par des coupes transversales (Cross Sectional Design).

Ils existent plusieurs méthodes basées sur ce principe et qui sont :

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

Les différentes méthodes de conception des surfaces en CAO sont représentées par la (Figure I.2).

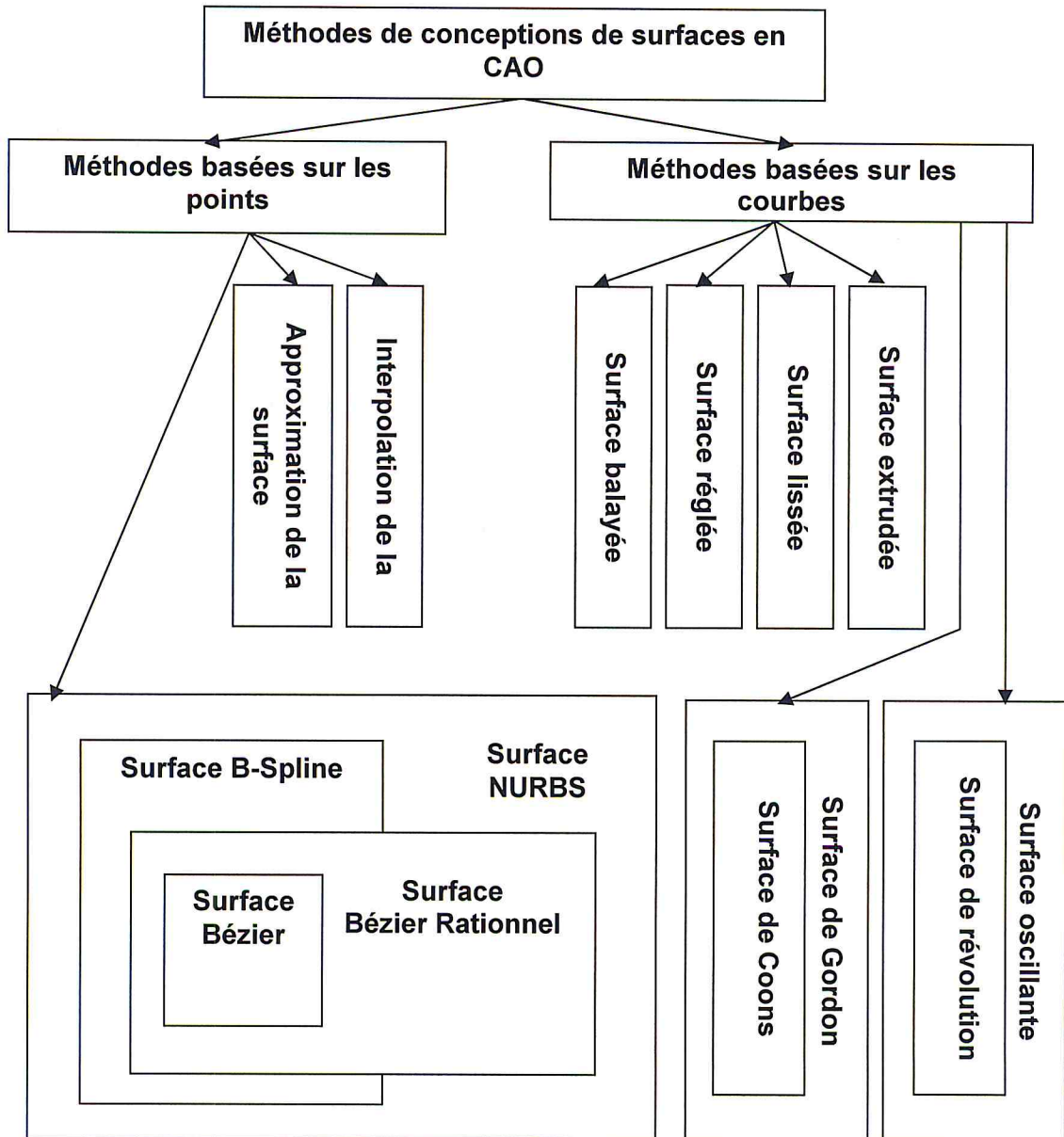


Figure I.2 : Les différentes méthodes de conception de surface.

Le concepteur doit choisir entre l'une des deux classes de conception en se basant sur les critères cités ci-dessous :

- La représentation de la surface d'une manière uniforme, c'est-à-dire de la même façon;

- La représentation des surfaces fermées;
- La modification de la surface à concevoir doit être intuitive et en temps réel;
- Les contraintes géométriques tel que le changement d'échelle, le changement de repère, les mouvements de translations et rotations n'influent pas sur la représentation de la surface;
- La plus rentable (moins coûteuse en source informatique et la plus représentative de l'objet);
- La conception ou la modification d'une surface existante est flexible et intuitive;
- Peut satisfaire au maximum les contraintes de design et certaines contraintes de continuités.

III. REPRESENTATION DES SURFACES [4, 5]:

Il existe deux manières de représenter les surfaces, non paramétrique et paramétrique, mais la plus utilisée en mécanique ainsi que dans notre étude est la seconde :

III.1. Surfaces non paramétriques :

III.1.1. Forme explicite :

Une représentation explicite d'une telle surface est donnée par l'équation :

$$Z = F(x, y) \quad (I.1)$$

Où pour chaque valeur du couple (x, y) correspond une seule valeur de Z ce qui implique l'impossibilité de représentation de surfaces fermées et la difficulté de manipulation.

III.1.2. Forme implicite :

Cette dernière nécessite la résolution d'une équation de la forme :

$$F(x, y, z) = 0 \quad (I.2)$$

Pour obtenir un triplet (x, y, z) cette représentation est plus puissante que la précédente mais elle reste difficile pour une manipulation interactive.

III.2. Surfaces paramétriques :

Cette représentation utilise la notion de paramètres (u, v) et un ensemble de fonctions f, g et h pour déterminer les coordonnées (x, y, z) comme suit :

$$F : \begin{cases} x = f(u, v) \\ y = g(u, v) \\ z = h(u, v) \end{cases} \quad (I.3)$$

- Les paramètres u , et v n'ont pas de significations physiques réelles. Ils prennent leurs valeurs en général dans l'intervalle $[0, 1]$;
- f, g, h sont des fonctions dans \mathbb{R} .

Les surfaces paramétriques n'ont pas d'utilité individuelle, mais leur conjonction (la réunion de plusieurs surfaces paramétriques) offre à l'utilisateur la possibilité de représenter n'importe quelle forme.

La flexibilité de la représentation en surfaces paramétriques est avantagée par rapport aux autres méthodes pour gagner la confiance des concepteurs dans le développement de leur application, aujourd'hui elle couvre pratiquement la représentation de tout type de surfaces.

IV. PROPRIETES DES SURFACES PARAMETRIQUES [6, 7, 8] :

L'utilisation des surfaces paramétriques comme un outil de représentation des formes nécessite de connaître leurs propriétés géométriques.

IV.1. Vecteur tangent et vecteur normal :

Lorsque on dérive la formule de la surface paramétrique $F(u,v)$ par rapport aux deux paramètres u et v on obtient :

$$F'(u) = (df/du, dg/du, dh/du)$$

$$F'(v) = (df/dv, dg/dv, dh/dv)$$

Notons par T_u, T_v les résultats de la dérivé de la surface paramétrique suivant les deux direction u , et v . Ces dérivées donnent les deux vecteurs tangents

$$T_u = (df/du, dg/du, dh/du) \text{ vecteur tangent suivant la direction } u;$$

$$T_v = (df/dv, dg/dv, dh/dv) \text{ vecteur tangent suivant la direction } v. \quad (1.4)$$

Le vecteur normal à la surface en un point $n(u, v)$ est le produit vectoriel des deux vecteurs tangents. Il est donné par la formule suivante :

$$n = \frac{\frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v}}{\left| \frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v} \right|} \quad (1.5)$$

La (Figure I.3) montre la disposition des deux vecteurs tangents et le vecteur normal.

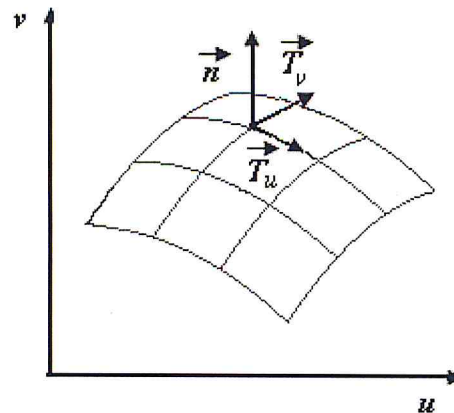


Figure 1.3 : Les vecteurs tangents, et le vecteur normal en un point de la surface.

IV.2. Les courbes isoparamétriques :

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

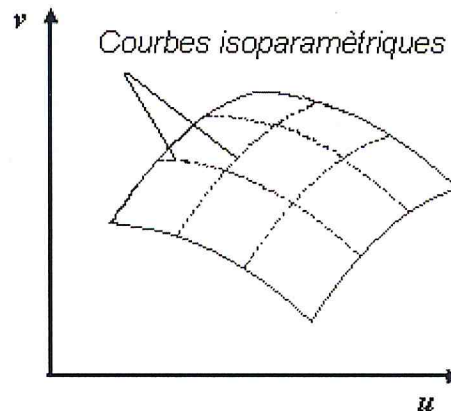


Figure 1.4 : Une surface composée de 08 courbes isoparamétriques, 04 dans chaque direction (u, v).

Les surfaces paramétriques $f(u, v)$ possèdent deux paramètres u, v , si on fixe la valeur u (resp. v) et en faisant varier v (resp. u) on obtient une courbe isoparamétrique dans la direction v (resp. u). La (Figure 1.4) montre les courbes isoparamétriques d'une surface donnée.

IV.3. Notion de Courbure :

L'inverse du rayon du cercle approximant le mieux la courbe en un point précis est appelé la courbure et est donnée par la formule :

$$k = 1 / R \quad (1.6)$$

Où R est le rayon du cercle de courbure, K est la courbure. La (Figure I.5) montre le rayon de courbure d'une courbe en un point donné.

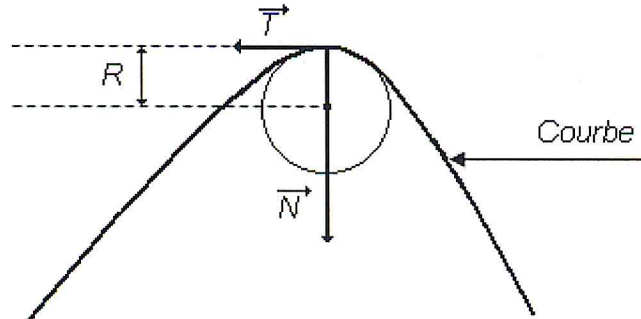


Figure I.5 : La courbure d'une courbe en un point.

Afin d'obtenir une idée bien précise de la surface à représenter, il est nécessaire de calculer plusieurs types de courbures.

Pour chaque point de la surface paramétrique, il existe un nombre indéterminé de courbes qui appartiennent à cette surface et passant par ce point. Par conséquent on aura des nombres indéterminés de courbures (autant de courbes que de courbures).

Mais dans notre étude, on va s'intéresser uniquement aux valeurs extrêmes des courbures pour chaque courbe (courbure minimale et courbure maximale). Ces deux valeurs extrêmes seront appelées les courbures principales.

Les rayons de courbures principaux d'une surface sont calculés par la résolution de l'équation du second degré suivante :

$$(LN - M^2)R^2 + (2MF - GL - EN)R + EG - F^2 = 0 \quad (1.7)$$

Où L , E , N , G , M et F sont les paramètres utilisés pour calculer les deux formes fondamentales et sont donnés par les relations suivantes :

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

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

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

$$L = n \times \frac{\partial^2 P}{\partial u^2} \quad (1.11)$$

$$M = n \times \frac{\partial^2 P}{\partial u \partial v} \quad (I.12)$$

$$N = n \times \frac{\partial^2 P}{\partial v^2} \quad (I.13)$$

Le nombre H défini comme ci-dessous est appelé courbure moyenne d'une surface au point $P(u, v)$.

$$H = \frac{K_1 + K_2}{2} \quad (I.14)$$

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

$$K = K_1 \times K_2 \quad (I.15)$$

En un point x de la surface, et en fonction des valeurs des deux courbures (H , K) 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.

Un autre type de courbure est la courbure absolue qui est calculée par la formule suivante :

$$A = |K_1| + |K_2| \quad (I.16)$$

V. SURFACES B-SPLINE ET SURFACES NURBS [6, 7] :

V.1. Les surfaces B-Splines :

Pour construire une surface B-Spline on a besoin d'un réseaux de points de contrôle $P_{i,j} / 0 \leq i \leq m$ et $0 \leq j \leq n$.

Les surfaces B-Spline sont données par la formule suivante :

$$p(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_{i,p}(u) B_{j,q}(v) p_{i,j} \quad (I.17)$$

Où :

- U et V sont des vecteurs des points nodaux avec $U = \{0=U_0, U_1, \dots, U_m=1\}$ et $V = \{0=V_0, V_1, \dots, V_n=1\}$
- p et q sont les degrés de la surface dans la direction u respectivement v .

- $B_{i,p}$ et $B_{j,q}$ sont des fonctions base B-Spline ayant la forme ci-dessous tout en satisfaisant les égalités suivantes :

$$H = m + p + 1 \text{ et } k = n + q + 1. \quad (I.18)$$

$$B_{i,0}(u) = \begin{cases} 1 & \text{si } u_i \leq u < u_{i+1} \\ 0 & \text{sinon} \end{cases} \quad (I.19)$$

$$B_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} B_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} B_{i+1,p-1}(u) \quad (I.20)$$

$$B_{j,0}(v) = \begin{cases} 1 & \text{si } v_j \leq v < v_{j+1} \\ 0 & \text{sinon} \end{cases} \quad (I.21)$$

$$B_{j,q}(v) = \frac{v - v_j}{v_{j+q} - v_j} B_{j,q-1}(v) + \frac{v_{j+q+1} - v}{v_{j+q+1} - v_{j+1}} B_{j+1,q-1}(v) \quad (I.22)$$

- Le réseau de contrôle et les paramètres u, v sont dans l'intervalle $[0, 1]$

La (Figure I.6) représente les paramètres d'une surface B-Spline.

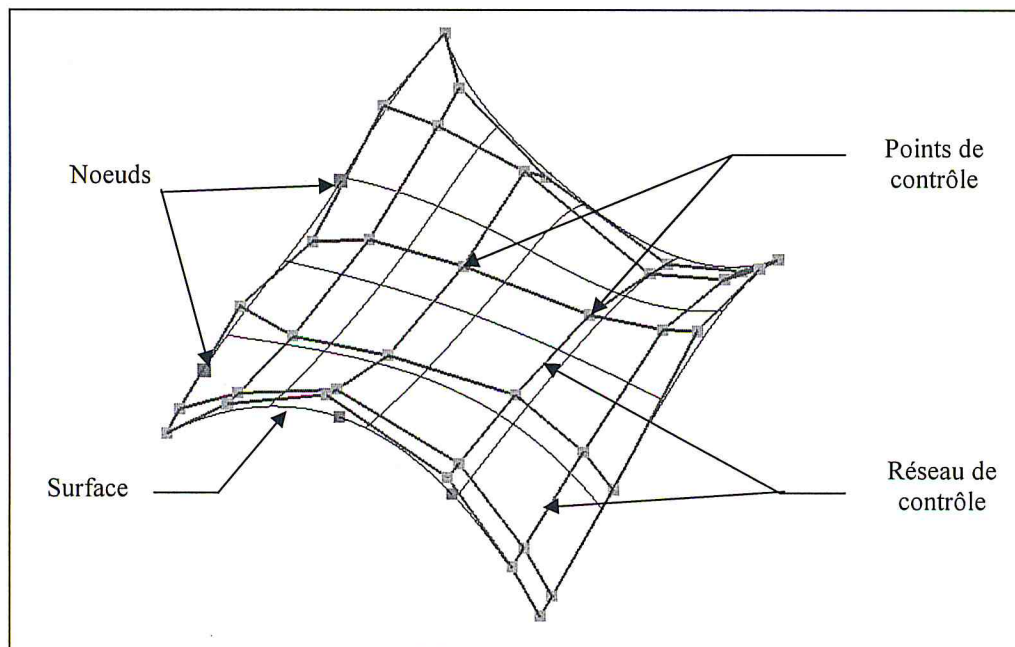


Figure I.6 : Surface B-Spline.

V.1.1. Vecteur nodal :

Soient U et V deux ensembles de $(m+1)$ (respectivement $(n+1)$) nombres réels croissants $u_0 \leq u_1 \leq \dots \leq u_m$, $v_0 \leq v_1 \leq \dots \leq v_n$ qui subdivisent la surface dans la direction u (respectivement v) en petites portions de surfaces. Dans ce qui suit nous allons parler sur une des directions soit par exemple U (tout ce qui va être dit pour U est valable pour V).

Les u_i sont appelés les nœuds et l'ensemble U est le vecteur nodal et l'intervalle $[u_i, u_{i+1}]$ est le $i^{\text{ème}}$ intervalle nodal.

Il est possible que certaines valeurs nodales u_i soient égales ($u_i = u_{i+1} = \dots = u_{i+k-1}$), et dans ce cas ce nœud est dit nœud multiple de multiplicité k , où k est le nombre d'apparition de ce nœud, dans le cas contraire le nœud est dit nœud simple.

Si les nœuds sont à distance égale (c'est-à-dire : $u_{j+1} - u_j$ est une constante pour $0 \leq i \leq m-1$), alors le vecteur nodal est dit uniforme, sinon il est dit non uniforme.

Le vecteur nodal intervenant dans la détermination de la forme de la surface où cette dernière peut être fermée, ouverte ou pincée selon les valeurs des nœuds.

Le concept de vecteur nodal apporte une grande flexibilité aux surfaces B-Splines et NURBS, il permet de gérer indépendamment le degré de la fonction du nombre de points de contrôle.

La (Figure I.7) montre la mise en pratique des vecteurs nodaux (u, v) .

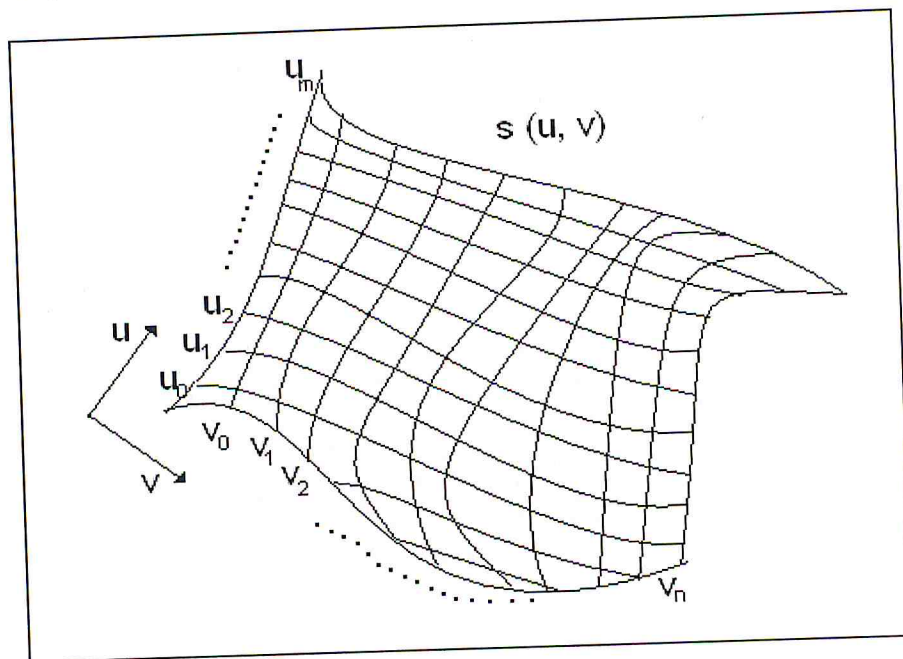


Figure I.7 : Mise en pratique des vecteurs nodaux (u, v) .

V.1.2. Propriétés des Surfaces B-Spline :

Les propriétés des surfaces B-Spline peuvent être résumées dans les points suivants :

- La représentation des formes en surfaces B-Spline est définie par morceaux; ou chaque morceau est une portion de la surface de degré (p, q) ; l'union de toutes ces portions forme la surface B-Spline, par conséquent on peut représenter des surfaces de formes très complexes à base de polynômes de faibles degrés.
- $B_{i,p}(u)$ (respectivement $B_{j,q}(v)$) est strictement positive pour i, p (respectivement (j, q)), avec v et u appartenants à l'intervalle $[0, 1]$.
- La somme de toutes les fonctions base B-Spline est égale à '1'.
- La surface B-Spline satisfait l'égalité fondamentale suivante :

$$H = m + p + 1 \text{ et } k = n + q + 1$$

- La surface B-Spline vérifie la propriété de l'enveloppe convexe; toute la surface est contenue dans son réseau de contrôle.
- La surface B-Spline suit un schéma de modification locale; si un point de contrôle quelconque appartenant au polygone de contrôle change de position cela n'affecte en aucun cas la forme globale de la surface ;
- La surface B-Spline $P(u, v)$ n'est pas aussi complexe que son polygone de contrôle. La propriété de la diminution de variation est donc vérifiée.
- $P(u, v)$ est continue dans chaque direction (v, u) , si u respectivement v est un nœud de multiplicité s respectivement t .
- la surface B-Spline vérifie la propriété de la variance affine ; lorsque une surface B-Spline subit une transformation affine on peut utiliser les images des points de contrôles pour construire la surface.
- l'ensemble des points de contrôle définit une courbe B-Spline l'ensemble des courbes B-Spline définit la surface B-Spline.
- Les courbes $P(0, v)$, $P(1, v)$, $P(0, u)$, $P(1, u)$ sont les courbes limites, la ligne 0 et la ligne n des points de contrôle respectivement (la ligne 0 et la ligne n) définissent les courbes limitées correspondant à $u = 0$ et $u = 1$ respectivement ($v = 0, v = 1$).

V.1.3. Les formes des surfaces B-Spline :

1. Surface ouverte : la surface B-Spline est dite ouverte dans une direction u ou v si le vecteur nodal U ou V n'a aucune structure, elle est dite encore ouverte dans les deux directions si la surface ne passe pas par les points de contrôles $P_{0,0}$, $P_{m,0}$, $P_{0,n}$, et $P_{m,n}$.
2. Surface pincée : elle est dite ainsi si les premiers $p+1$ nœuds sont nuls et les derniers $p+1$ nœuds sont égaux à '1'.

3. Surface fermée : la surface B-Spline est fermée si les premiers points et les derniers points de contrôles sont égaux dans cette direction (v ou u).

V.2. Les surfaces NURBS :

Les surfaces NURBS sont construites en utilisant les mêmes éléments pour construire Les surfaces B-Spline, sauf qu'on a besoin d'affecter pour chaque point de contrôle un poids correspondant. Les surfaces NURBS sont données par la formule suivante :

$$p(u,v) = \frac{\sum_{i=0}^m \sum_{j=0}^n B_{i,p}(u) B_{j,q}(v) w_{i,j} p_{i,j}}{\sum_{i=0}^m \sum_{j=0}^n B_{i,p}(u) B_{j,q}(v) w_{i,j}} \quad (I.23)$$

Où $B_{i,p}$ et $B_{j,q}$ sont des fonctions base B-Spline satisfaisant les égalités suivantes :

$$H = m+p+1 \text{ et } k = n+q+1 \quad (I.24)$$

Donc une surface NURBS représentée en trois (03) dimensions peut être vue comme une projection d'une surface B-Spline en ajoutant la dimension du poids (le plan $W=1$). D'où le calcul fait sur les surfaces NURBS peut être passé par celui des B-Spline, et ensuite on fait une simple projection sur le plan $W=1$.

La (Figure I.8) représente une surface NURBS.

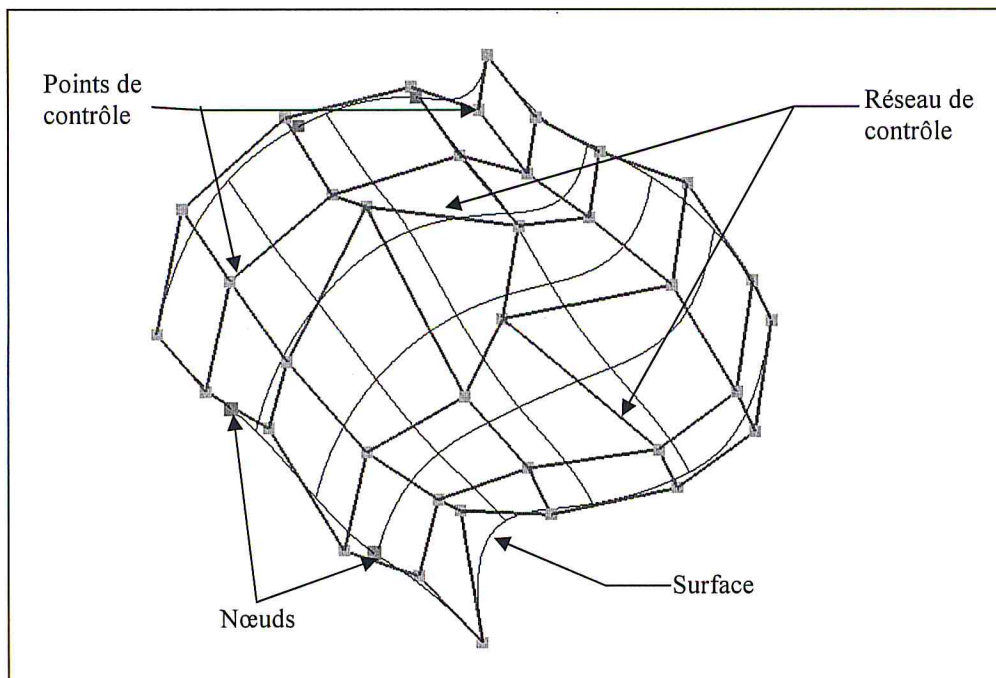


Figure I.8 : Surface NURBS.

V.2.1. Propriétés des Surfaces de NURBS :

On sait bien que les surfaces B-Spline sont des cas particuliers des surfaces NURBS (dans le cas où tout les poids sont égaux, la surface NURBS se transformera en une surface B-Spline). Donc les propriétés des surfaces NURBS sont similaires à celles des surfaces B-Splines, en ajoutant seulement les propriétés ci-dessous :

- Une courbe quelconque sur la surface NURBS est une courbe NURBS, définis à partir d'un polygone de contrôle.
- Une surface NURBS est une surface rationnelle, donc elle permet de représenter toutes les formes coniques.
- L'attraction respectivement (la répulsion) d'une portion de la surface $P(u,v)$ par un point de contrôle dépend de l'augmentation respectivement (la diminution) de la valeur de son poids de contrôle.
- 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.

VI.CONCLUSION :

Nous avons vu dans ce chapitre les différentes méthodes de conceptions et de représentation des surfaces ainsi que leurs propriétés. Nous avons constaté que la conception à base de courbes est plus rentable, car elle possède une interprétation géométrique, et génère des courbes exactes, et le plus important est que cette méthode est flexible. La représentation la plus puissante parmi celles-ci est la représentation en surface NURBS qui englobe toutes les méthodes de conception des surfaces.

Après avoir étudié les surfaces, nous allons décrire les outils utilisés dans l'usinage, les machines de fraisage et leur programmation.

I. INTRODUCTION :

Depuis son apparition il y a seulement quelques dizaines d'années la notion de Commande Numérique CN, s'est imposée au sein de l'industrie avec ses différentes catégories (allant du secteur de petite entreprise (fabrication d'objet simple) tel que les moules, les matrices...etc. arrivant aux domaines les plus compliqués tels que l'industrie aérospatiale (comme c'est le cas des grandes firmes dans ce domaine : **NASA, ESA, BOING, AIRBUS...**).

Sa prolifération (la commande numérique) au sein du monde de l'industrie l'a conduit à une amélioration considérable du point de vue qualité et quantité du produit, elle a bouleversé l'industrie en termes de production, à peine imaginable quelques années auparavant.

La commande numérique a été créée pour piloter des machines par l'intermédiaire d'un programme informatique autonome; sans aucune intervention externe, son premier objectif été d'automatiser des machines-outils traditionnelles : Tours, Fraiseuses, Perceuses et Aléseuses ; La CN est l'origine de plusieurs avancées techniques comme celle de Centre d'usinage.

De nos jours elle est associée étroitement au progrès de la microélectronique, on trouve dans le monde industriel plusieurs types de machines à commande numérique répondant aux besoins des utilisateurs.

II. EVOLUTION HISTORIQUE DES MACHINES D'USINAGE [9]:

L'apparition de **MOCN** (Machine Outil à Commande Numérique), en anglais **NCMT** (Numerically Controlled Machine Tools), n'est pas un fruit du hasard, mais elle est due à l'évolution logique du développement des machines à travers les siècles.

Nous allons présenter en quelques lignes les dates significatives dans l'évolution des machines.

- Vers l'an 1500, Léonard de Vinci propose des solutions à base de tournage à mouvement continu, qui été comme tout le temps en avance sur son époque ;
- 1818, machine à fraiser d'Eli Whitney (USA), l'une des premières machines à fraiser ;
- 1862, fraiseuse universelle Brown et Sharp (USA), exposée à Paris en 1867.
- 1942-1945, premières fraiseuses à commande numérique pour applications spécifiques (Bendix USA) ;
- 1958, machine MilwaukeeMatic de Kearney Trecker (USA), premier centre d'usinage. Le terme **machining center**, traduit en français par **centre d'usinage**, est introduit dans le vocabulaire technique ; 1959, machine MU des Ateliers GSP (France), premier centre d'usinage européen ;
- 1964 Omniline de Sundstrand (USA), premier **FMS (Flexible Machining System)**, traduit en français par **atelier flexible**.
- A la fin des années 70 apparaît les premières MOCN qui révolutionne le monde de l'industrie.

III. DEFINITION D'UNE MACHINE OUTIL :

Voici un extrait de « La Grande Encyclopédie, inventaire raisonné des sciences des lettres et des arts » - H. Lamirault et Cie, éditeurs – environ 1910.

« Une machine-outil est un appareil destiné à faire fonctionner des outils mécaniquement, le mouvement étant d'ailleurs transmis à la machine soit à la main, soit par l'intermédiaire d'un moteur. (...) »

Les machines-outils employées pour le travail des métaux sont très diverses et très répandues, et de nos jours les exigences de la construction ont amené à produire des types d'une grandeur vraiment colossale. (...) Les machines employées dans les ateliers de construction de machines comprennent :

- Les tours ;
- Les machines à percer ;
- Les machines à fraiser ;
- Les machines à aléser ;
- Les machines à raboter ;
- Les limeuses. »

IV. ETUDE DE QUELQUES MACHINES OUTILS [9, 10]:

Le marché industriel comprend plusieurs types de machines outils, dont :

- Le tour : il permet de réaliser des surfaces de révolution. Il est utilisé principalement pour usiner les arbres. La pièce est tenue par un mandrin en mouvement de rotation (mouvement de coupe) transmis par la broche, l'outil peut se déplacer suivant deux directions à travers le chariot, le premier est parallèle à l'axe de la broche et le deuxième l'est perpendiculaire. La (Figure II.1) représente un tour.

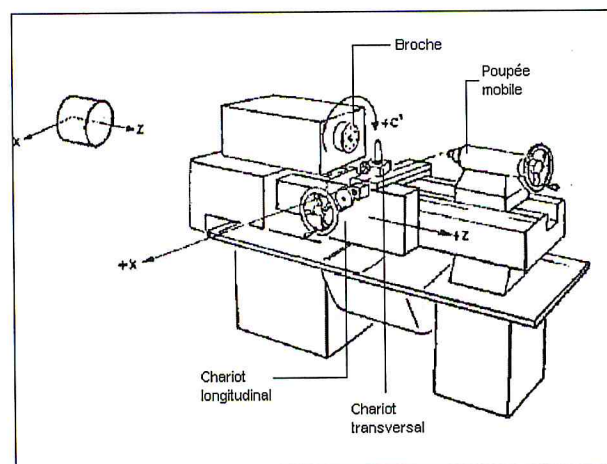


Figure II.1 : Machine Tour.

- Les fraiseuses : par leur activités elles ont supplantées (raboteuses, étaux limeurs), peuvent réaliser des opérations de contournage. L'outil (fraise) est fixé à une broche qui le fait tourner (mouvement de coupe), peut se déplacer en translation par

rapport à la pièce suivant trois directions. Les surfaces gauches peuvent être réalisées avec un centre d'usinage 3 axes en utilisant comme outil une fraise boule. Dans ce cas, les conditions de coupe de la fraise ne sont pas optimales. En effet, lorsque le point générateur de l'outil se trouve sur l'axe de rotation de la fraise, la vitesse de coupe en ce point est nulle. Pour résoudre ce problème il est nécessaire de pouvoir incliner l'axe de rotation de la fraise. Cette inclinaison est obtenue en ajoutant sur le centre d'usinage 3 axes une tête rotative. La tête rotative possède deux axes de rotation motorisés. L'association d'un centre d'usinage 3 axes et d'une tête rotative 2 axes constitue un centre d'usinage 5 axes. La (Figure II.2) et la (Figure II.3) représentent respectivement un centre d'usinage à 5 axes et une fraiseuse à 3 axes.

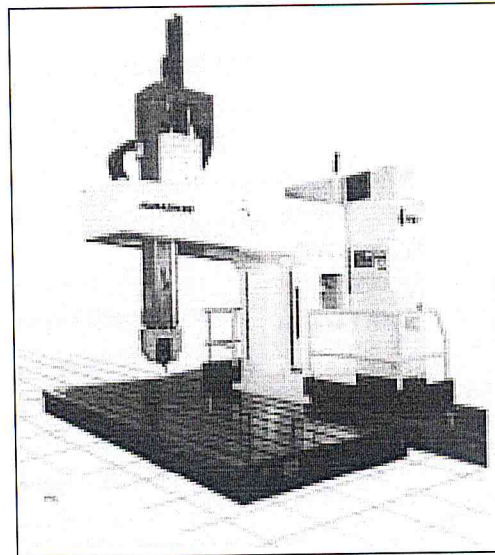


Figure II.2 : Centre d'usinage à 5 axes.

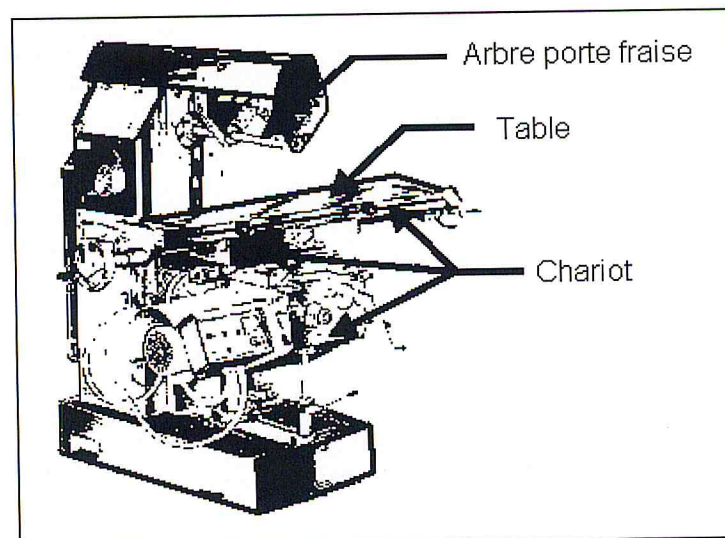


Figure II.3 : Une fraiseuse à 3 axes.

Chapitre II

*Architecture &
programmation des
MOCN.*

V. DEFINITION ET CONSTRUCTION D'UNE MOCN [10, 11] :

V.1. Définition d'une MOCN :

Nous allons voir la définition des **MOCN** (Machine Outil à Commande Numérique) qui est une grande famille (tours, fraiseuse, rectifieuse cylindrique, rectifieuse plane,...etc.). Dans ce qui suit notre étude sera portée uniquement sur les fraiseuses.

La **MOCN** est une machine d'usinage à cycle automatique programmable, le terme de Commande Numérique insigne que la machine est commandée par ordinateur (commande numérique par ordinateur « CNC ») qui contrôle toutes les parties mobiles (position, vitesse,...) ainsi ce type de machine se compose de deux parties complémentaires (Figure II. 4).

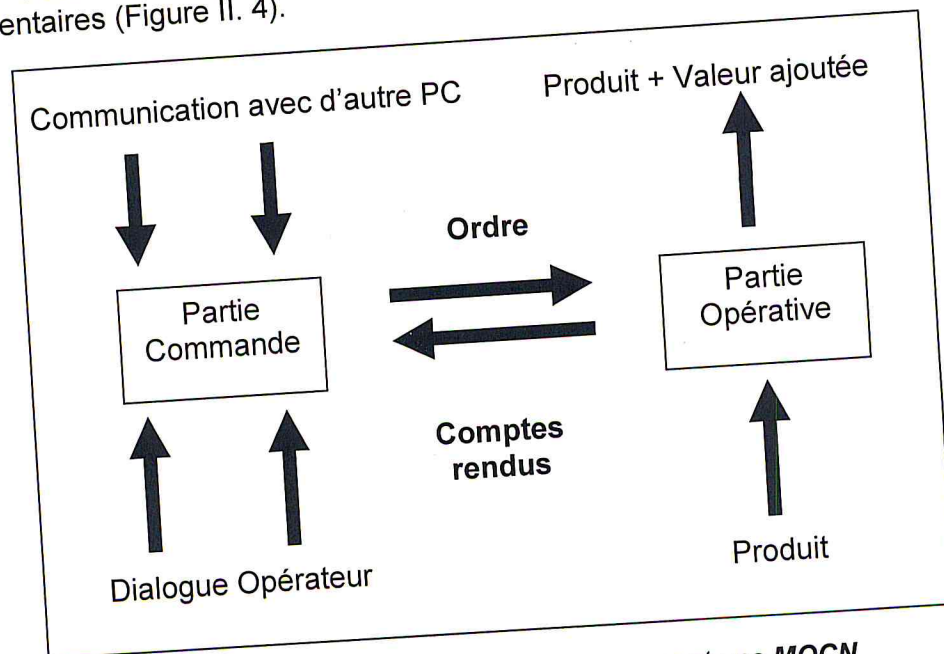


Figure II.4 : Les deux compartiments composant une **MOCN**.

- **Partie opérative** : est composée des organes agissant directement sur le produit final.
- **Partie commande** : contrôle la machine en envoyant des consignes et génère des comptes-rendus du déroulement des opérations.

Une machine outil a pour but de réaliser physiquement les mouvements de coupe 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. Une machine outil à commande numérique assure la réalisation automatisée des pièces et les mouvements nécessaires sont décrits dans un programme.

V.2. Construction des fraiseuses :

Les fraiseuses sont actuellement disponible en grande variété en type et en taille. Un centre d'usinage "CU" typique à commande numérique "CNC" (Computer Numerically Controlled) par ordinateur est présenté dans la (Figure II.5).

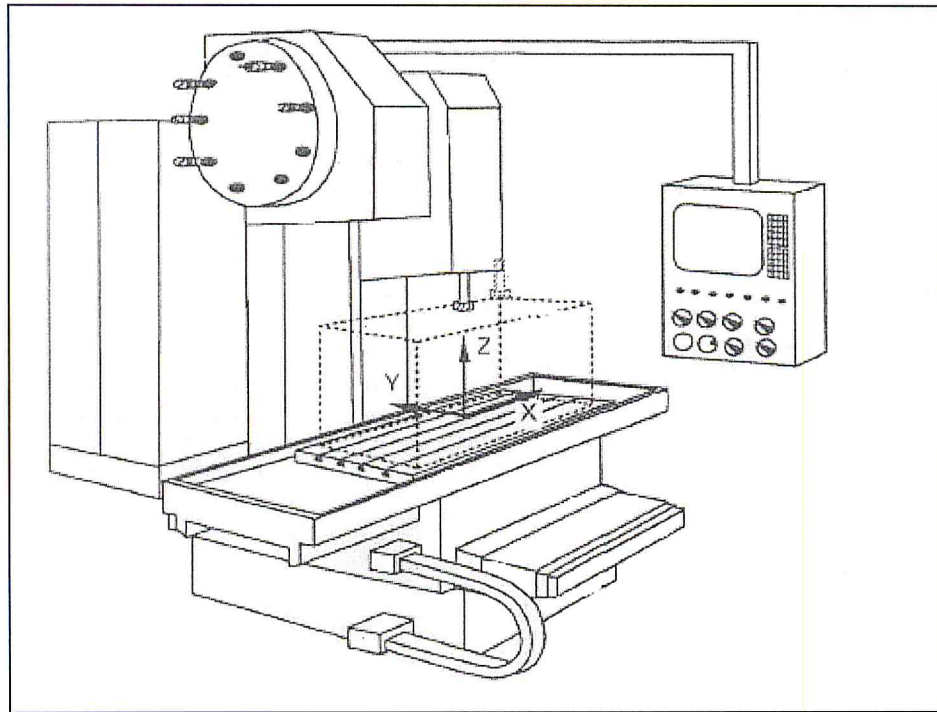


Figure II.5 : Un Centre d'Usinage à commande numérique par ordinateur.

Les machines à commande numérique sont semblables à leurs contreparties manuelles. Les mouvements relatifs primaires de génération de l'outil de coupe et l'objet sont identiques, de même que les techniques et les approches d'usinage. La différence essentielle est que les mouvements de position manuels sont remplacés par des commandes de moteur de sorte que des positions d'outil de coupe soient commandées par des ordinateurs.

Les différentes parties d'une fraiseuse sont les suivantes :

- Le bâti : est la plateforme de la machine, qui est rigide et qui permet de supporter la chaleur et d'assurer une précision accrue lors de l'usinage, il a une liaison importante au sol pour amortir les chocs due aux accélérations élevées des parties mobiles, sa construction est faite de métaux rigide et de béton ainsi qu'une répartition optimisée de sa masse. En plus de ces fonctions essentielles, le bâti remplit aussi des fonctions auxiliaires importantes, comme la gestion des copeaux, et la protection des opérateurs.
- La broche : elle porte l'outil de fraisage et transmet ainsi le mouvement de rotation crée de l'ordre de 25 000 tours/minute à 30 kW (mouvement de coupe) nécessaire à l'opération de fraisage, elle respecte des contraintes fonctionnelles à ces

fréquences. Elle est rigide et stable thermiquement pour assurer une plus grande précision de positionnement de l'outil par rapport à la pièce durant l'usinage.

- Le porte outil : il assure la liaison entre l'outil et la broche, il doit être d'une très grande rigidité pour assurer l'attachement de l'outil à des hautes fréquences.

- L'outil d'usinage : l'outil utilisé dans le fraisage est appelé fraise, qui est un cylindre de métal ou autre type de matière avec une extrémité équipée de dents coupants, la fraise est aussi caractérisée par le sens, la géométrie de coupe, ainsi que le nombre de dents et leurs types.

- la jauge d'outil : la jauge d'outil est une distance qui sépare l'arrête coupante de l'outil et le point de référence broche, dans l'usinage la notion de jauge d'outil est indispensable, pour cela il faut connaître les paramètres suivant : la longueur (L) de l'outil (fraise), la rayon de la fraise (R), le rayon du bout de la fraise (@), et le point référence de la broche. La (Figure II.6) représente les jauges d'outils.

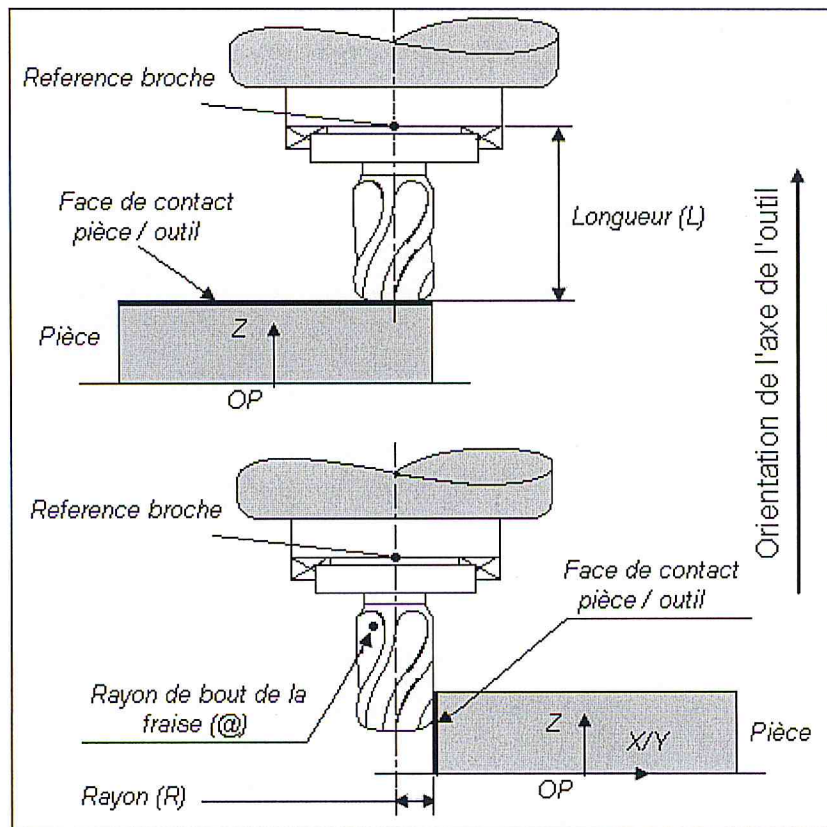


Figure II.6 : Une jauge d'outil. [11].

- Le directeur de commande numérique (DCN) : les machines à commande numérique disposent d'un ordinateur équipé d'une mémoire intégrée et qui permet ainsi d'exécuter les programmes chargés en mémoire d'une façon autonome. La (Figure II.7) montre le fonctionnement d'un DCN.

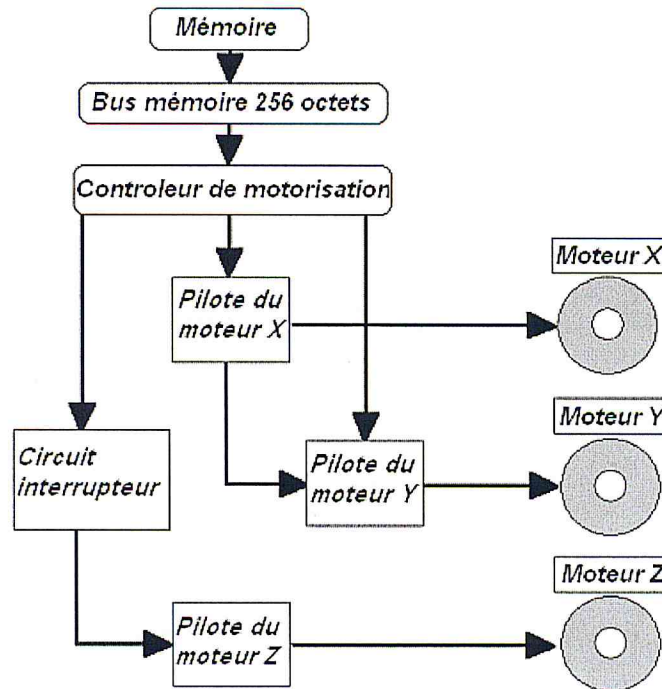


Figure II.7 : Schéma général d'intercommunication du système de commande.

Le DCN assure des fonctions de contrôle et de gestion lors de l'opération d'usinage, il permet l'élaboration et la modification des programmes sur site, ainsi que la gestion de programmes chargés en mémoire et la mémorisation des données de la machine (journal édité lors de l'usinage les erreurs de trajectoire et des opérations de rectification, ...). La CN assure aussi une communication externe par le biais d'une ligne de télécommunication pour échanger des données (programmes, données, messages.) entre le DCN et un équipement informatique externe (serveur, ordinateur, PC, ...) afin d'augmenter infiniment l'espace mémoire ce qui influe sur la capacité d'usinage des pièces qui va être à son tour augmenté (dans le cas d'usinage des surfaces dont la taille de leur programme est plus importante que celle de la mémoire du DCN).

Les fonctions primaires du DCN est le contrôle des organes mobiles en position et en vitesse dans les 3 axes, par le biais d'un commandement en temps réel de la trajectoire prise en fonction des paramètres reçus des capteurs installés sur la machine. Le DCN permet de réaliser les tâches suivantes :

- interprétation du programme d'usinage (G-Code) ;
- détermination des blocs exécutables ;
- gestion des données reçues des capteurs ;
- surveillances des erreurs ;
- correction de trajectoire ;
- commande des actionneurs auxiliaires ;
- distribution d'énergie ;
- traitement des informations de sécurité.

La (Figure II.8) représente une architecture d'une CN.

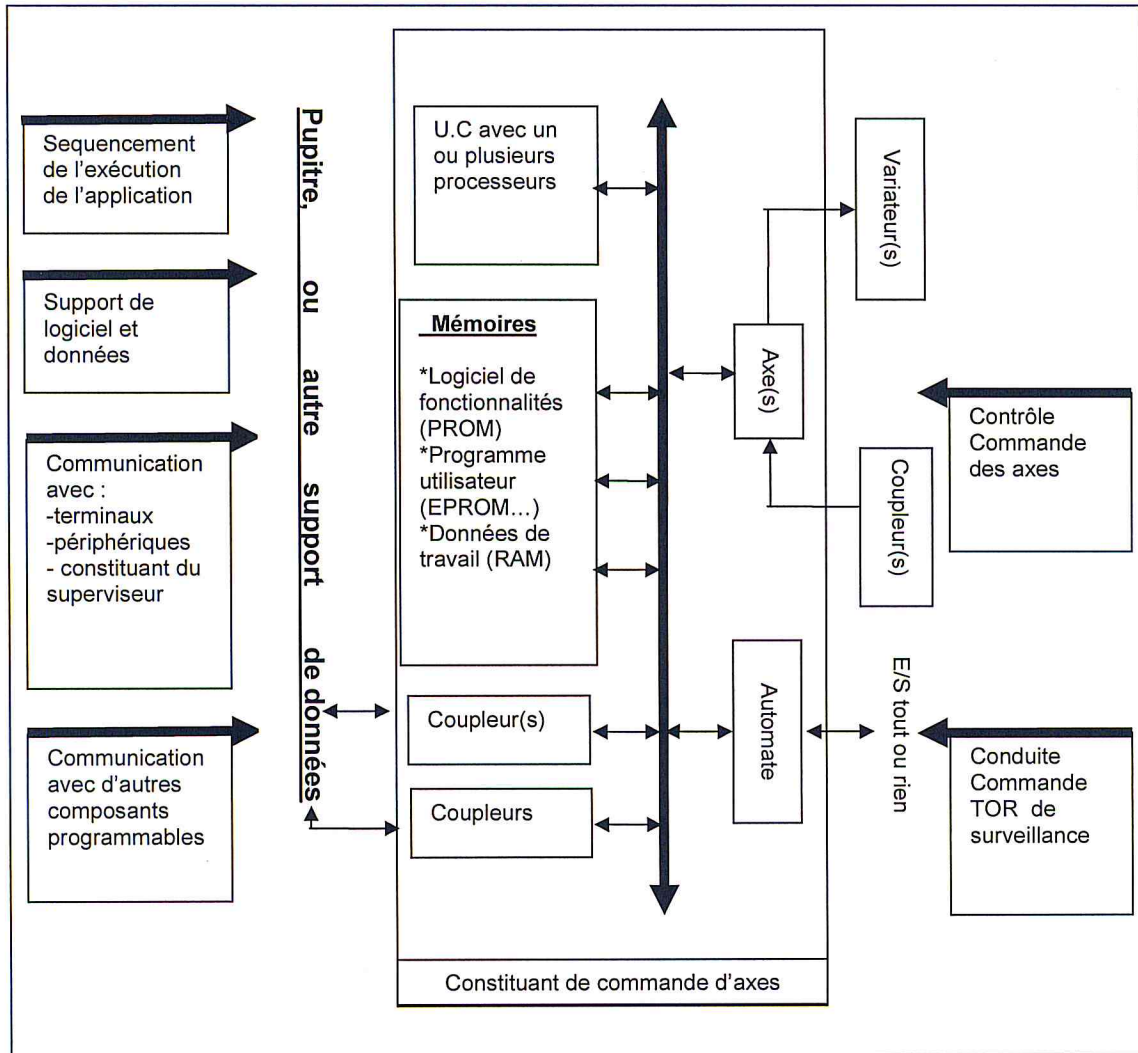


Figure II.8 : Architecture physique d'une commande.

- Pupitre de commande : son rôle primaire est de dialoguer avec le DCN par le biais de messages envoyés appelés Commandes, il possède un écran servant à la visualisation du programme en cours d'usinage ou le profil de la pièce à produire, et un clavier utilisé pour contrôler la machine (arrêt, marche, relancer, ...) ainsi que l'écriture en conversationnel des programmes pendant l'usinage d'une autre pièce dont le programme est déjà écrit.

- L'armoire électronique : interposée entre le DCN et la machine (coté mécanique), elle enferme tous les composants électroniques (transistor, condensateur, fusible, câblage, ...) et des cartes de gestion.

VI. LES AXES DE DEPLACEMENTS [11]:

Les axes mettent en mouvement les parties mobiles de la fraiseuse avec de fortes accélérations. Ces axes sont constitués d'un guidage, d'un système d'entraînement, d'une motorisation et d'un système de mesure.

VI.1. Mouvements de translations X, Y, et Z :

VI.1.1. Axe de mouvement Z : l'axe Z est parallèle à la broche principale de la machine. Lorsque la broche principale est pivotante ou orientable, l'axe Z doit être parallèle à l'axe de la broche quand cette dernière est dans la position zéro. La position zéro de référence est de préférence celle où la broche est perpendiculaire à la surface de bridage de la pièce.

VI.1.2. Axe de mouvement X : quand cela est possible, l'axe X doit être horizontal et parallèle à la surface de bridage de la pièce. Pour les machines avec des pièces en rotation, l'axe X doit être radial et parallèle aux glissières du chariot transversal.

VI.1.3. Axe de mouvement Y : l'axe Y de mouvement forme avec les axes X et Z un trièdre de sens direct. Le trièdre formé par les trios axes (x, y, et z) peut être construit par la règle de la main droite comme il est indiqué dans la (Figure II.9).

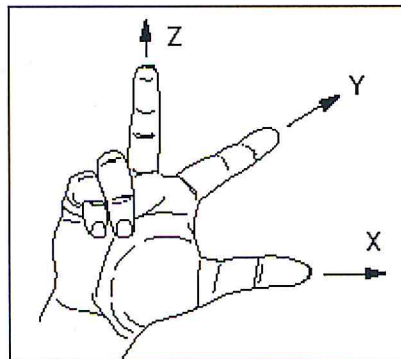


Figure II.9 : Trièdre formé par les trios axes (x, y, et z).

VI.2. Mouvements de rotations A, B, C :

Les angles **A**, **B** et **C** définissent les mouvements de rotation effectués respectivement autour des axes parallèles à **X**, **Y** et **Z**. Le sens positif des axes est défini de la même manière qu'un mouvement dans une direction positive d'axe de translation ou de rotation, augmente les valeurs positives de la position de la pièce par rapport à la machine. La (Figure II.10) montre L'orientation positive d'un axe rotatif correspond à la rotation d'une vis de pas à droite avançant dans le sens positif de l'axe associé (sens du vissage).

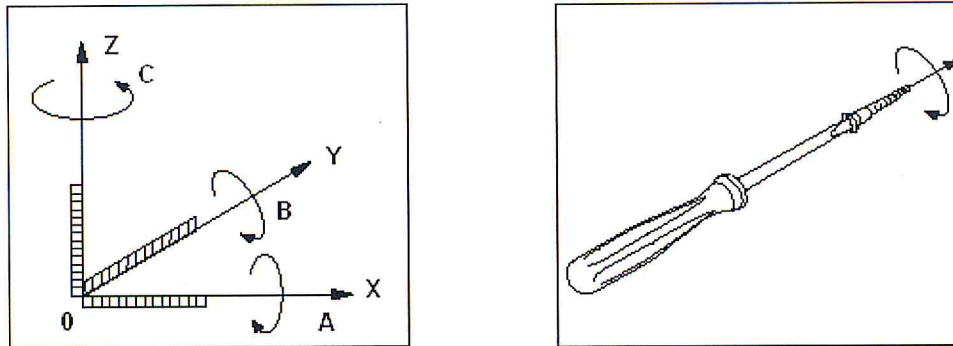


Figure II.10 : L'orientation positive des axes rotatif; elle est analogue au mouvement de rotation d'une vis dans le sens de vissage.

VI.3. Notion d'axe numérique, et demi axe numérique :

Axe numérique : c'est un axe de déplacement pour lequel une infinité de positions peuvent être atteintes à la résolution de positionnement près. L'axe est asservi en position et en vitesse.

Demi Axe numérique : c'est l'axe de déplacement pour lequel un ensemble fini de positions peuvent être atteintes. L'axe est asservi en position ou en vitesse.

VI.4. Différents types de fraiseuses :

On distingue deux types de fraiseuses selon la position des axes

VI.4.1. Fraiseuses verticales :

Elle est appelée ainsi car la broche est verticale, la position des trois axes est comme suit :

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

VI.4.2. Fraiseuses horizontales :

La seule différence qui existe avec la fraiseuse verticale est le fait que l'axe de rotation de la broche devient ici horizontal, se qui engendre un parallélisme avec la table, l'axe des Z suit toujours celui de la broche.

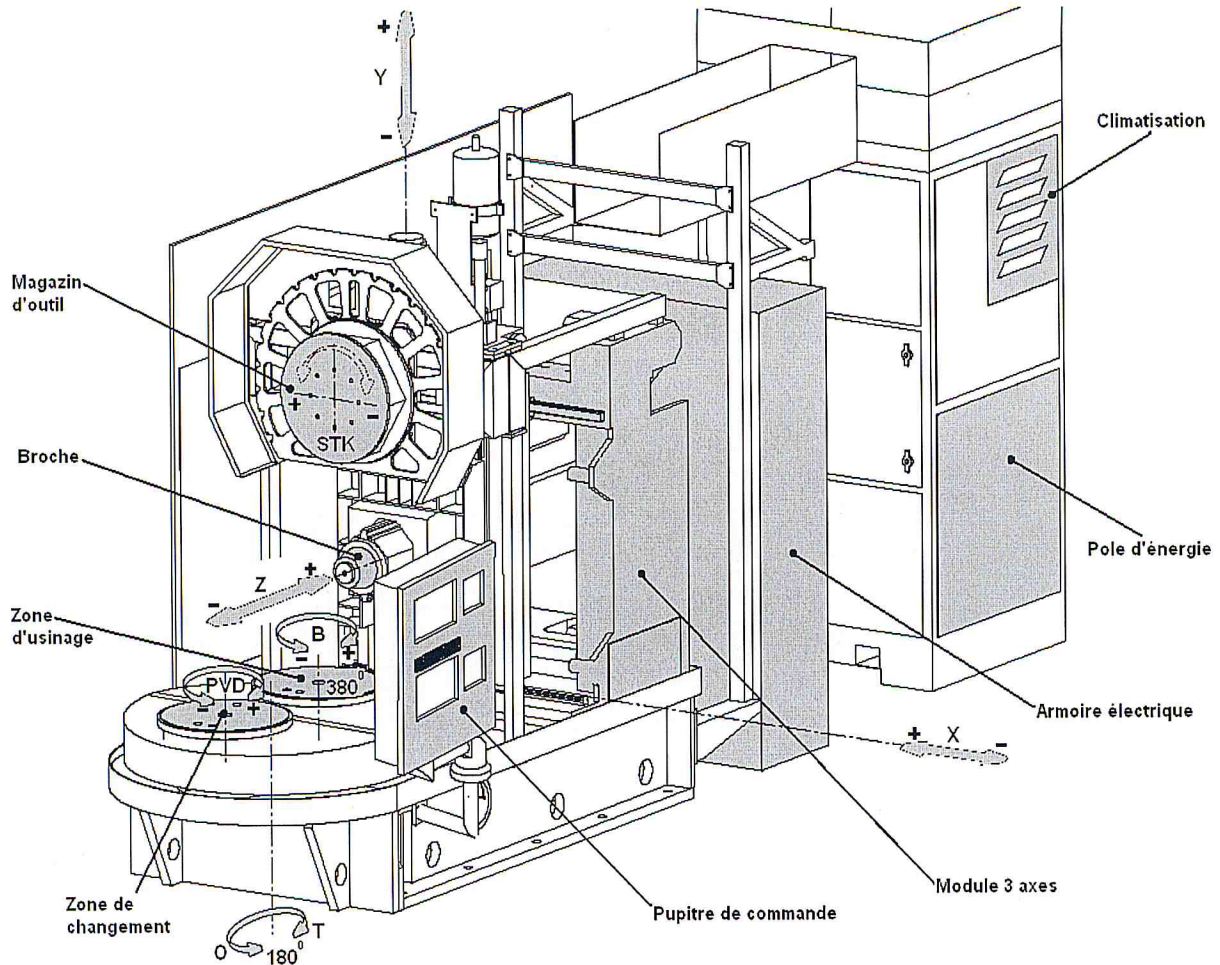


Figure II.11 : Vue générale d'un centre d'usinage METEOR 5 (Document PCI-SCEMM).

VII. LA PROGRAMMATION DES FRAISEUSES [11]:

La programmation des MOCN repose sur des conventions standardisées, définissant les langages de programmation. ISO

Le programme est la description structurée de l'opération d'usinage, il comporte deux types d'informations : ordres de déplacement et ordres auxiliaires.

- L'ordre de déplacement : il est paramétré par un ensemble de données qui sont : point de destination, vitesse de déplacement et le mode d'interpolation. Il existe trois types mode d'interpolations :

- Interpolation linéaire : trajectoire défini par un segment droit.
- Interpolation circulaire : trajectoire décrite par un arc de cercle.
- Interpolation hélicoïdale : trajectoire défini par un hélice.

Il existe un autre type d'interpolation qui est en cours de développement l'interpolation polynomiale dont la trajectoire est une courbe polynomiale c'est pourquoi elle est appelée ainsi, son rendement est meilleur et exempt de erreur.

- Les ordres auxiliaires : ils sont des ordres traités de manière séquentielle permettant l'amélioration de l'opération d'usinage.

VII.1. Les différents modes de programmation

Quelque soit le langage de programmation utilisé pour le développement des programmes pièces, le seul langage compréhensible par la machine est le langage ISO. Le passage d'un langage de haut niveau au langage ISO est possible en utilisant un logiciel de traduction.

- Langage ISO :
- deux techniques permettent d'en programmer comme le montre la (Figure II.12) :
 - Pupitre de CN : cette technique consiste à commander directement la MOCN par le biais du pupitre, ainsi programmer en conversationnel mais cette solution mobilise la machine.
 - Utilisation d'un poste de programmation : la programmation s'effectue séparément de la machine sur poste dédié, mais les tests syntaxiques et la simulation graphique s'effectuent sur la MOCN, ou par le biais d'un logiciel de simulation. Le poste de programmation peut être soit une console spécialisée reproduisant l'interface utilisateur de la CN ou un ordinateur standard utilisant un logiciel spécialisé ou tout simplement un éditeur de texte.
- Langage conversationnel : un logiciel résidant dans la CN permet la création, la modification, la visualisation ou la simulation de programme pièce pendant l'usinage d'une autre pièce.
- Langages de haut niveau (APT-PROMO) : ces langages normalisés facilitent la programmation puisqu'ils assurent une indépendance avec la CN. Un autre logiciel spécifique à chaque CN permet le passage (traduction) en langage ISO. Ils peuvent aussi être utilisée comme interface entre les systèmes FAO et le langage ISO.
- Programmation par FAO :

Les logiciels FAO exploite la géométrie de la pièce générée par la CAO. Ces logiciels exploitent des paramètres tels que le choix de surface à usiner, la méthode d'usinage, paramètres de coupe et l'outil.

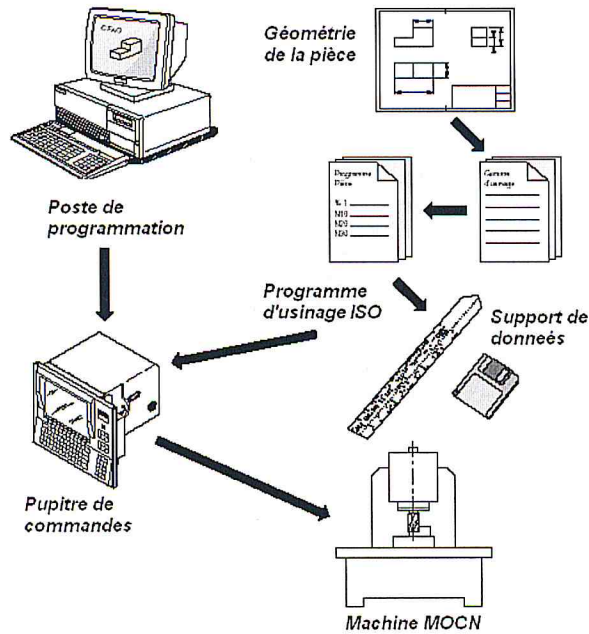


Figure II.12 : Synthèse des différents modes de programmation.[11]

VII.2. Programmation en langage ISO :

Le DNC ne comprend que ce code, il interprète une série de code opération standard en commandes électriques ainsi contrôler les mouvements de la table et des axes et tous autres fonctions auxiliaires.

On appelle le code interprété par le DCN,est le **G-Code**, employé par exemple dans la commande de mouvement rapide (G00), linéaire (G01) et circulaire (G02) et (G03), changement d'outil, définition d'unité de mesure millimètre (G20) ou pouce (G21), compensation de la longueur d'outil (G43), appel de fonctions préprogrammées, incluant d'autres codes, avance rapide (Feedrate) F-Code, de coupure rapide (Speed) S-Code, choix d'outil (Tools) T-Code et d'autres fonction auxiliaires, commandes d'arrêt / marche de la broche M-Code. Les codes cités ci-dessus sont tous inclus sous le nom de code G-Code.

Un programme G-Code est constitué d'un ensemble de lignes appelées '**Bloc**', ce dernier est un assemble d'instructions relatives à une séquence d'usinage (instruction atomique du programme).

Un **Bloc** est constitué d'un ensemble de **Mots** qui est aussi composé d'un ensemble de caractères, **Adresse** (lettre : G, J, F, R, S, T, L, M, N, P, X, Y, Z) suivie d'un **chiffre** constituant une information.

Un mot représente soit une fonction soit un déplacement suivant l'un des axes.

Les fonctions usuelles peuvent être représentées comme suit :

- G..** : Fonctions préparatoires.
- F..** : Fonctions vitesse d'avance (**F**eedrate = avance).
- S..** : Fonctions vitesse de broche (**S**peed = vitesse).
- T..** : Fonctions outils (**T**ools = outils).
- M..** : Fonctions auxiliaires (**M**iscellaneous = varié, divers).
- X..** : Mouvement suivant l'axe **X**
- Y..** : Mouvement suivant l'axe **Y**
- P..** : Fonction paramètre (**P**arameter = paramètre)
- Z..** : Mouvement suivant l'axe **Z**.

La (Figure II.13.B) montre la structure d'un mot et la (Figure II.13.A) montre la structure d'un programme G-Code.

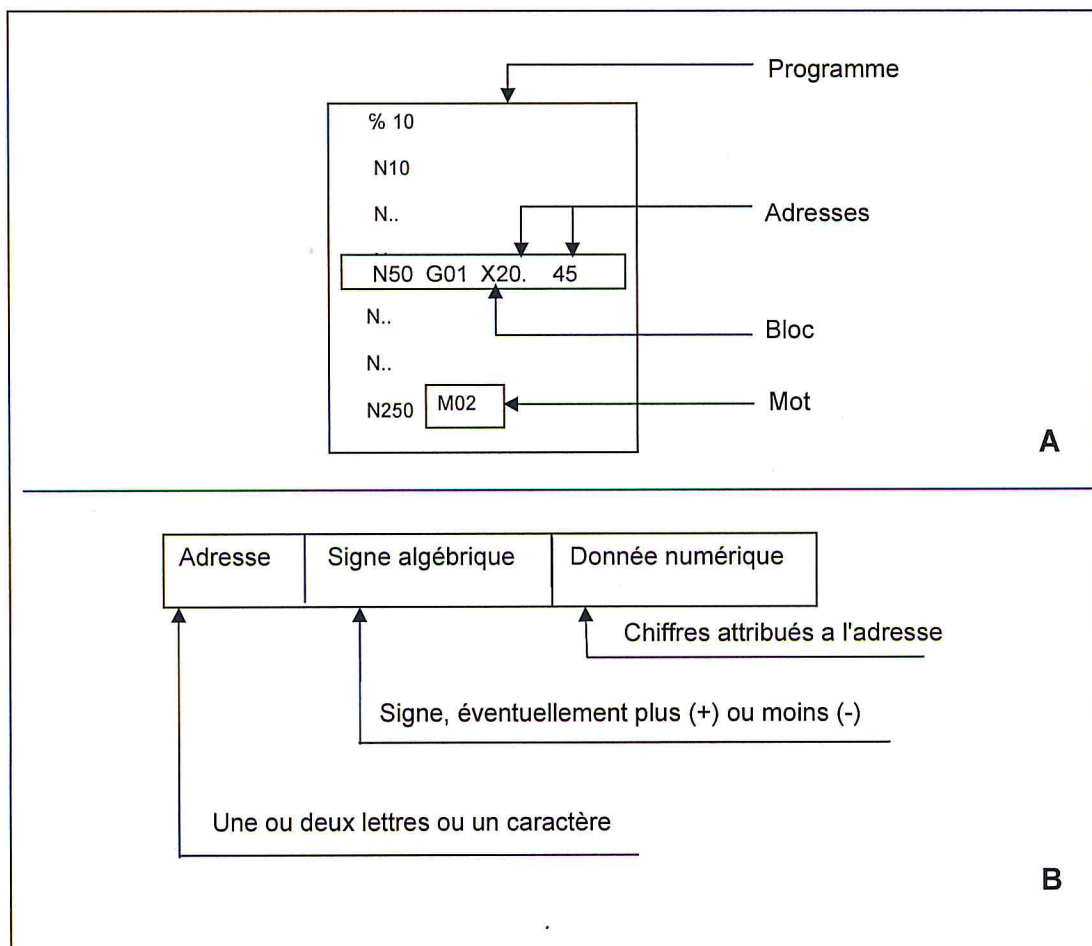


Figure II.13 : Structure d'un programme G-Code.

Soit par exemple un mot définissant une fonction adresse **G**, l'écriture du mot de fonction adresses **G** au format **G02** (format variable).
Le mot **G01** peut s'écrire : **G1**

G : Nombre de chiffres maximum autorisés avec l'adresse,
 0 : Zéros de tête sont facultatifs,
 1 : Adresse du mot.

VII.2.1. Format de bloc :

Un bloc est une séquence d'usinage, composé de plusieurs mots (ordres). Le format de bloc définit la syntaxe des mots de fonction et la dimension composant chaque bloc (Figure II.14).

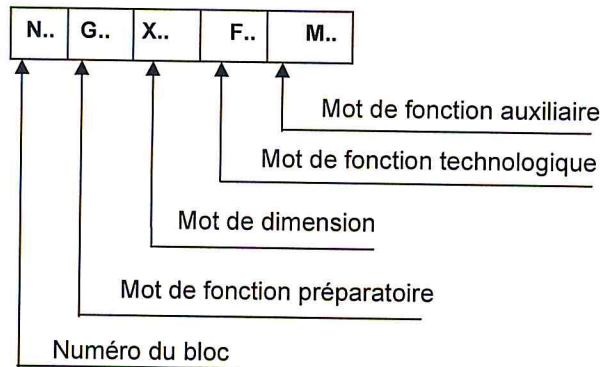


Figure II.14 : Structure d'un bloc.

VII.2.2. Structure générale d'un programme :

Un programme d'usinage est comme tout autre programme informatique, comprend un début caractérisé par une suite de caractères spéciaux de début et une autre suite de fin de programme, l'exécution se fait séquentiellement. Il existe des possibilités d'appel de fonction, on peut même utiliser des variables,...etc. Comme c'est indiqué par la (Figure II.15).

La numérotation des blocs n'intervient pas dans l'ordre de déroulement du programme, mais il est conseillé de numérotter les blocs dans l'ordre d'écriture (de dix en dix par exemple).

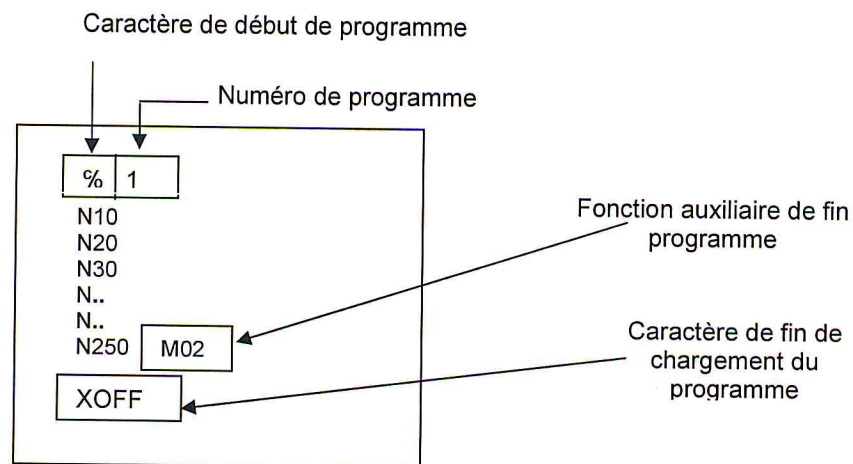


Figure II.15 : Structure générale d'un programme d'usinage.

Un programme commence toujours par le caractère '%' suivi d'un numéro et éventuellement d'un commentaire entre parenthèses distinguant ainsi les programmes, ensuite construit par un ensemble de **blocs** (séquences) et se termine par la fonction auxiliaire '**M02**' qui permet la remise à zéro (RAZ) du système.

VIII. L'APPORT DES MOCN [1, 2]:

Les fraiseuses numériques sont une prouesse technologique, leur invention est un fruit de plusieurs siècles d'expériences, elles ont fait leurs apparitions qu'au années 70 dont elles ont bouleversé l'industrie en apportant :

- Un outil fiable et précis (cloner un objet en plusieurs milliers d'exemplaire);
- Gestion automatique (chargement/déchargement d'outil/pièce à usiner d'une manière automatique);
- Travail continue (laisser la machine travail plusieurs heures jusqu'à l'obtention du résultat);
- Réduction du temps d'usinage par rapport au coût de fabrication des pièces; est cela grâce à la rapidité de lecture et traitement des informations, ainsi que la réponse rapide des divers organes mobiles et sa vitesse de déplacement;
- la souplesse et la polyvalence sont des opportunités offertes par ce type de machines. Elles permettent le changement des dimensions de la pièce en cours d'usinage et la possibilité de réaliser toutes les formes imaginables;
- Elle dispose d'un cycle programmé.

IX.CONCLUSION :

Dans ce chapitre on a vu la définition, la construction, et la programmation des MOCN (machines outils a commande numérique), et plus précisément on a étudié principalement les fraiseuses à commande numérique.

Après avoir étudié les formes à usiner et les machines d'usinage, nous allons voir dans les chapitres suivants les techniques d'usinages des surfaces gauches car sans technique adéquate on ne peut pas aboutir à des résultats satisfaisants même si on a fait une bonne conception des pièces et on a utilisé une machine sophistiquée.

Chapitre III

*Usinage des surfaces
gauches.*

I. INTRODUCTION :

L'usinage est l'étape de fabrication proprement dite. L'usinage concrétise les étapes étudiées précédemment (phase de conception –CAO-), puisque le but final n'est pas la conception en elle même mais plutôt la fabrication, c'est-à-dire l'usinage des formes conçues sur des fraiseuses.

Cette opération se manifeste par le fonctionnement de la fraiseuse en aboutissant à un produit fini représenté par un objet conçu par l'étape CAO et fabriqué par l'étape FAO.

Les ordres de déplacement de l'outil respectent les règles de fabrication et suit un chemin tracé défini en respectant des règles de production.

Dans ce chapitre nous allons voir ces règles et aussi leurs buts et contextes.

II. PRESENTATION DES PARAMETRES DE FRAISAGE [11, 12, 13] :

Toute opération d'usinage exploite un ensemble de paramètres et le fraisage ne fait pas exception à la règle. Le fraisage d'un objet est différent d'un autre par les dimensions, la qualité du produit final, la matière brute (métal, bois, plastic... etc.). Donc lors de l'usinage d'une pièce, il est nécessaire de tenir compte des paramètres suivants :

- type de métal d'outil et dimensions de l'outil (fraise).
- type de matériau de la pièce à usiner.
- forme de l'outil (fraise).
- Modes d'usinage.
- détermination de la trajectoire de l'outil.
- positionnement de l'outil par rapport à la pièce.
- compatibilité de l'outil et de la pièce à usiner.

II.1. Types d'outils (fraises) :

Dans l'usinage des surfaces gauches les outils (fraises) enlèvent la matière uniquement par leur partie active (la partie de la fraise qui est en contact avec la pièce à usiner). Les surfaces gauches sont usinées par les trois types de fraises suivantes :

- Fraise hémisphérique : la partie active prend la forme d'une sphère. La (Figure III.1.1) représente une fraise hémisphérique.
- Fraise cylindrique : la partie active est un cylindre. La (Figure III.1.2) représente une fraise cylindrique.
- Fraise torique (rayonnée) : la partie active prend la forme d'un tore. La (Figure III.1.3) représente une fraise torique.

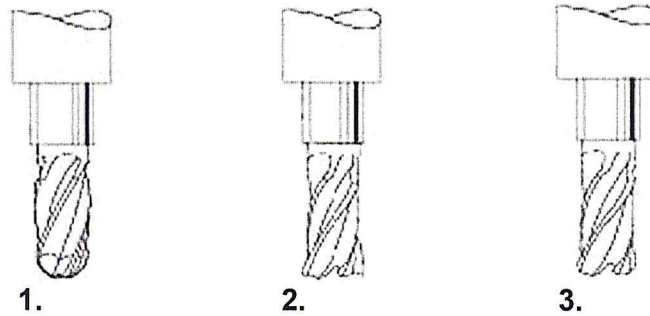


Figure III.1 : Les différents types de fraises.

(1). Fraises hémisphériques. (2). Fraises cylindriques. (3). Fraises toriques.

II.2. Modes d'usinage :

Avant de citer les différents types d'usinage, il faut introduire la notion du mouvement générateur : la génération des formes dans le fraisage s'obtient par les deux mouvements de coupe et d'avance qui peuvent être de même sens ou de sens opposés.

- **Mouvement de coupe** : c'est le mouvement de rotation donné aux fraises, qui travaillent par enveloppe du profil à réaliser.
- **Mouvement d'avance** : un mouvement rectiligne, circulaire, ou quelconque est donné à la pièce où à l'outil, selon la conception et le type des machines. Les différents mouvements d'avance possibles suivant les axes X, Y et Z) sont plus généralement attribués à la pièce par l'intermédiaire de la table porte pièce dans le cas des fraiseuses CN, par contre dans le cas des centres d'usinage; chaque mouvement d'avance est attribué selon le concepteur au porte pièce ou au porte outil.

Le type de fraisage peut être changé en changeant simplement la disposition de l'axe de rotation de l'outil par rapport à la surface à usiner.

II.2.1. Fraisage de face (en bout) :

Dans ce mode l'axe de l'outil est perpendiculaire à la surface à fraiser. La surface résultante présente une série de courbes sécantes (cycloïdes), ces traces sont tracées par les dents de la fraise. La (Figure III.2) représente l'usinage de face et le résultat obtenu par ce mode de fraisage.

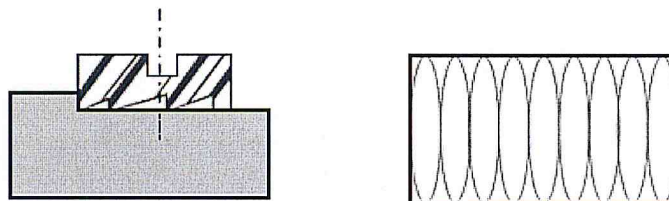


Figure III.2: Fraisage de face et le résultat donné.

II.2.2. Fraisage de profil (en roulant) :

Dans ce cas, l'axe de l'outil est parallèle à la surface à fraiser. La (Figure III.3) montre ce mode d'usinage. Le fraisage de profil laisse une série d'ondulation (crêtes) dont la taille dépend du diamètre de la fraise et de son avancement.

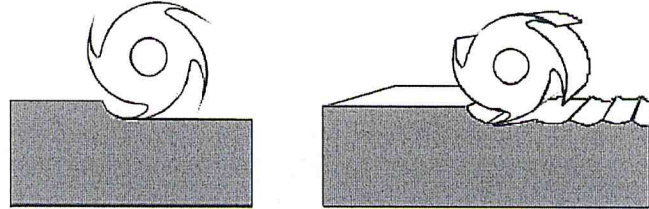


Figure III.3 : Fraisage de profil

Le fraisage de profil comporte lui-même deux types d'usinages :

- Usinage en avalant (concordance) : les deux mouvements de coupe et d'avance convergent dans la même direction. L'outil tourne dans le même sens que le trajet d'usinage. Ce mode d'usinage nécessitant impérativement que le mouvement d'avance soit sans jeu fonctionnel (voir la Figure III.4).

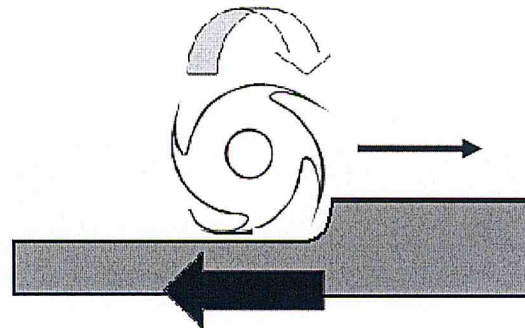


Figure III.4 : Usinage en avalant.

- Usinage en opposition : les deux mouvements de coupe et d'avance s'opposent au point de contact avec la pièce à usiner. C'est le contraire du type précédent, c'est-à-dire que l'outil tourne dans le sens opposé du trajet d'usinage (voir la Figure III.5).

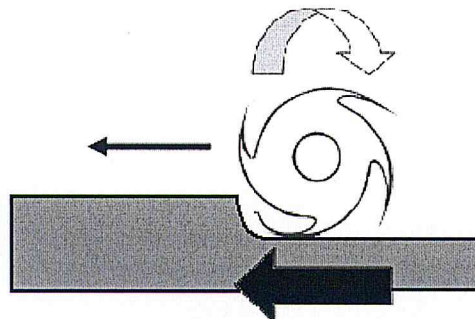


Figure III.5 : Usinage en opposition.

II.3. Types de trajectoires :

Plusieurs types de trajectoires de déplacement de l'outil sont possibles et qui sont :

- Déplacement point à point : dans ce type, le directeur de commande (CNC) reçoit uniquement les coordonnées du point final sans préciser le chemin à suivre. La (Figure III.6) montre plusieurs trajets suivis pour atteindre un même point.

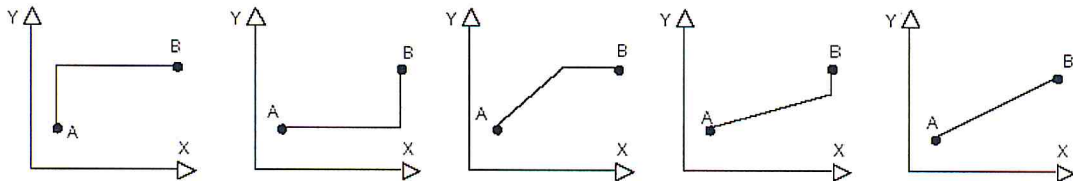


Figure III.6 : Présentation des différents trajets suivis.

- Déplacement paraxial : les trajectoires sont parallèles aux axes de déplacement et la vitesse de déplacement est contrôlée. Ce type de déplacement permet le fraisage précis à vitesse imposée. La (Figure III.7) représente un déplacement paraxial.

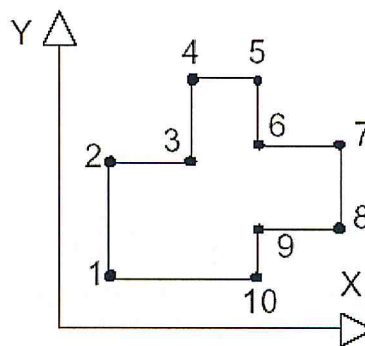


Figure III.7 : Déplacement paraxial.

- Déplacement continue (contournage) : les informations reçues dans ce cas sont plus importantes de celles de point à point puisque cette fois-ci le directeur de commande reçoit les coordonnées du point final avec le trajet tout en effectuant le fraisage. Dans ce mode d'usinage, l'outil se déplace simultanément suivant les axes de la machine. En deux dimensions (2D) c'est le contournage plan tandis qu'en (3D) c'est un contournage dans l'espace. Ce dernier est utilisé dans l'usinage des surfaces gauches.

II.4. Positionnement de l'outil sur la pièce à usiner :

L'outil (fraise) se positionne en un point précis par rapport à la surface à usiner. Pour usiner une surface gauche, il faut que l'outil utilisé soit constamment tangent aux différents points de contact entre l'outil et la surface à usiner. Le repère de l'outil est le

point centre (C_E). Lors du calcul du trajet de l'outil on fait toujours référence à ce point et c'est ainsi qu'on forme une enveloppe au dessus de la surface qui représente la trajectoire de ce point (voir la Figure III.9). Le point de contact de la fraise et de la surface est appelé C_C (cutter contact) qui se trouve à la périphérie de l'outil à une distance r représentant le rayon de la pointe de l'outil pour une fraise hémisphérique et le petit rayon pour une fraise torique. La pointe de la fraise est le C_L (cutter location).

Le rayon de l'outil est le R . \vec{u} représente l'axe de rotation de la fraise, \vec{n} représente la normale de la surface au point de contact (voir la Figure III.8).

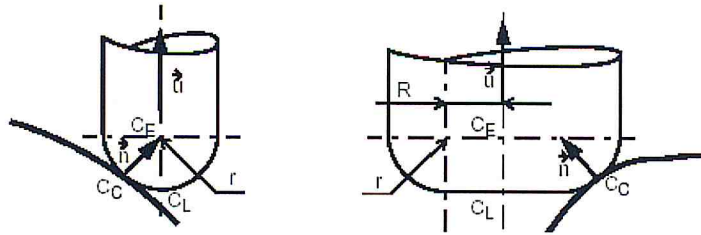


Figure III.8 : Position de la fraise sur la surface.

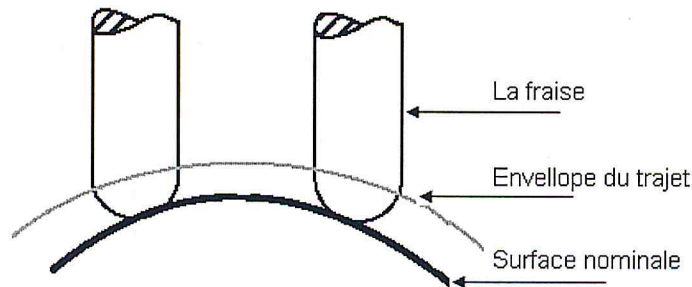


Figure III.9 : Enveloppe de la surface a usinée.

Les positions des points de l'outil hémisphérique sont données par les équations suivantes :

$$\begin{cases} \overrightarrow{OC_E} = \overrightarrow{OC_C} + r\vec{n} \\ \overrightarrow{OC_L} = \overrightarrow{OC_E} - r\vec{u} = \overrightarrow{OC_C} + r\vec{n} - r\vec{u} \end{cases} \quad (III.1)$$

Les positions des points de l'outil torique sont données par les équations suivantes :

$$\vec{k} = \frac{\vec{u} \wedge \vec{n}}{\|\vec{u} \wedge \vec{n}\|} \quad (III.2)$$

$$\begin{cases} \overrightarrow{OC_E} = \overrightarrow{OC_C} + r\vec{n} + R \cdot \frac{\vec{k} \wedge \vec{u}}{\|\vec{v} \wedge \vec{u}\|} \\ \overrightarrow{OC_L} = \overrightarrow{OC_E} - r\vec{u} = \overrightarrow{OC_C} + r\vec{n} - r\vec{u} + R \cdot \frac{\vec{k} \wedge \vec{u}}{\|\vec{v} \wedge \vec{u}\|} \end{cases} \quad (III.3)$$

A partir du point de contact, on peut calculer les coordonnées du centre de l'outil. Le posage de l'outil par rapport à la surface à usiner peut être fait de deux manières différentes :

- 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.

II.5. Calcul de la position d'outil :

La (Figure III.10) décrit le fonctionnement logique du processus de calcul de la trajectoire de l'outil lors de l'usinage.

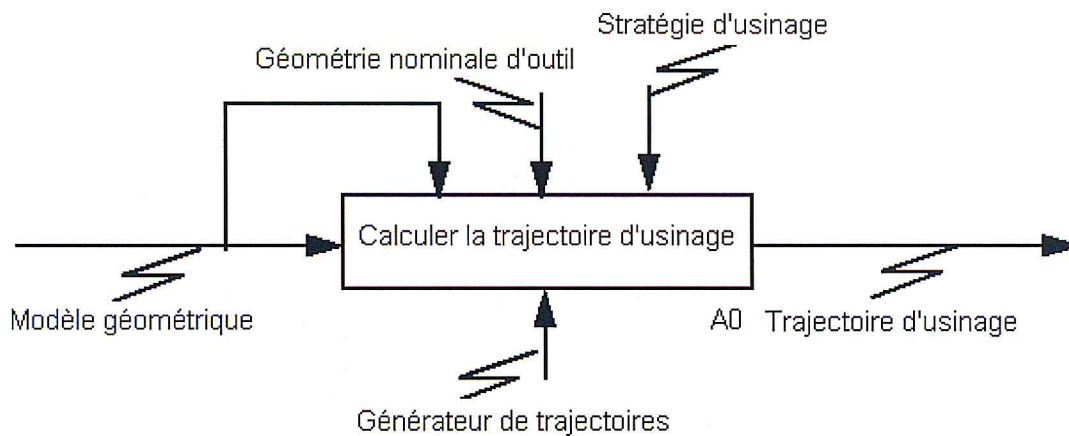


Figure III.10 : Schéma de calcul de la trajectoire d'usinage.

Lors de l'usinage, le calculateur reçoit une série d'informations représentées par les paramètres suivantes :

- la forme géométrique de la surface (pièce à usiner) sous forme d'un modèle surfacique ou solide.
- La géométrie de l'outil et ses dimensions.
- Stratégie d'usinage qui définit la direction d'usinage et les tolérances d'usinage (erreurs d'usinage).

Les trajectoires de l'outil sont obtenues en associant une courbe plane à la surface, et ainsi on obtient une passe d'usinage. En dupliquant la courbe plane et en associant les courbes obtenues à la surface on obtient un ensemble de passes d'usinage qu'il faut raccorder pour construire le trajet. Calculer le trajet est une opération qui vise à exprimer le mouvement d'un point fixe de l'outil (C_E ou C_L) sur l'enveloppe de la surface en prenant en compte les tolérances d'usinage, ce qui nécessite la détermination des points de passage de l'outil qui permettent de respecter les différentes contraintes (précision, tolérances d'erreurs...).

Le calcul du trajet est une identification des courbes théoriques (formant l'enveloppe de la surface) sous un format interprétable par le directeur de commande numérique. Pour le bon déroulement de l'opération de calcul de la trajectoire, le DCN reçoit les informations suivantes :

- Calcul de la position du C_E sur une zone de la forme.
- Gestion des interférences possibles entre l'outil et la surface.
- Calcul des passes d'usinage, en respectant le pas longitudinal, une passe représente une succession de trajets élémentaires reliant deux positions d'outil. Ces successions de trajets doivent respecter les critères de précision ainsi leurs nombres et position sera défini par ces critères.
- Le calcul du trajet se fait par accumulation de passes tout en respectant le pas transversal (précision transversal à la trajectoire).

III. TECHNIQUES D'USINAGE DES SURFACES GAUCHES EN FINITION [5, 13]:

Dans l'usinage des surfaces gauches, plusieurs techniques peuvent être utilisées. Nous allons étudier dans ce chapitre les techniques basées sur l'utilisation des courbes isoparamétriques.

III.1. Aller simple (One-Way):

Dans cette méthode, l'usinage des surfaces suit une isoparamétriques (u ou v) jusqu'à sa fin, après cela l'outil est déplacé en quittant la surface dans l'autre isoparamétrique. Ce processus est répété pour l'usinage de toute la surface. La (Figure III.11) montre le principe de l'usinage en One-Way.

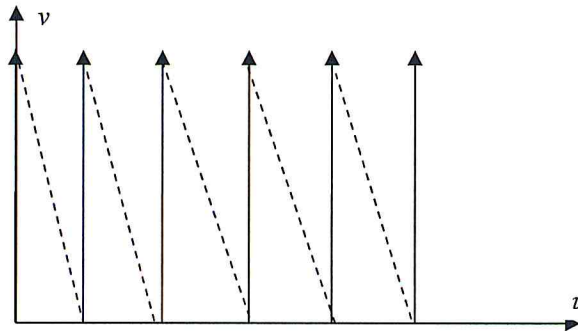


Figure III.11 : Usinage en aller simple.

III.2. Aller retour (Zig-Zag) :

Dans cette méthode, l'usinage des surfaces suit une isoparamétriques (u ou v) jusqu'à sa fin, ensuite on déplace l'outil sans quitter la surface à usiner afin d'usiner la surface dans le sens opposé. Le processus est répété pour l'usinage de toute la surface. La (Figure III.12) montre le principe de l'usinage en Zig-Zag.

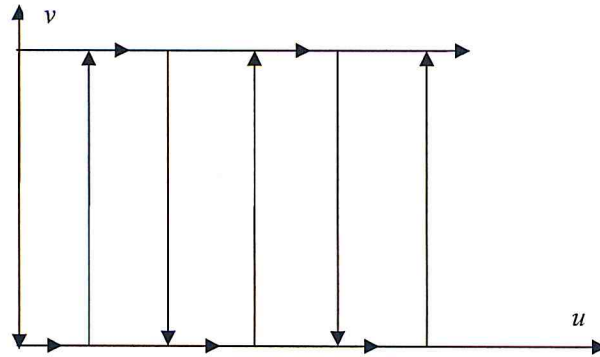


Figure III.12 : Usinage en aller retour.

III.3. Concentrique :

Cette méthode consiste à usiner la surface suivant des courbes parallèles aux limites de celle-ci jusqu'à sa fin. La (Figure III.13) montre le principe de l'usinage en Concentrique.

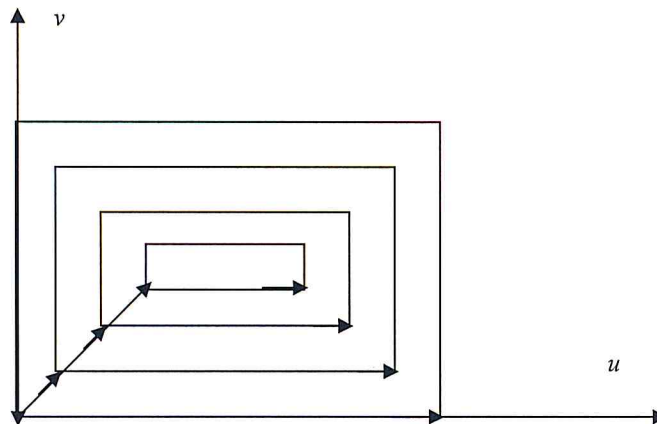


Figure III.13 : Usinages par la méthode concentrique.

III.4. Spiral-In et Spiral-Out:

Dans ces deux méthodes le principe est le même et l'outil d'usinage suit une trajectoire en forme de spirale jusqu'à l'usinage de toute la surface, mais la différence réside dans le point de départ (sens de parcours).

Dans la première méthode (Spiral In); l'outil démarre de l'extérieur de la surface et suit la trajectoire jusqu'à atteindre l'intérieur, tandis que la seconde méthode c'est le contraire et l'outil démarre de l'intérieur pour atteindre l'extérieur. La (Figure III.14) montre le principe de l'usinage en Spiral-In et Spiral-Out et les flèches montrent le sens de parcours.

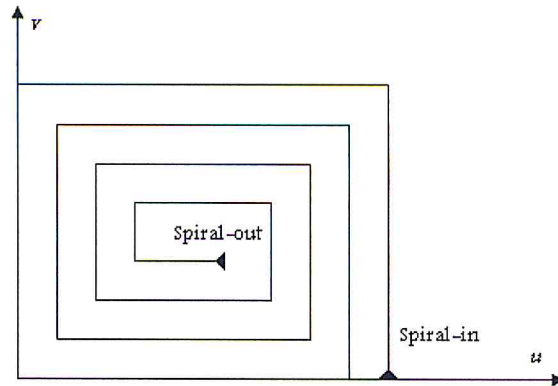


Figure III.14 : Usinage en Spiral in, et Spiral out.

III.5. Radiale :

Cette méthode d'usinage consiste à usiner la surface en commençant par son centre jusqu'à sa limite tout en effectuant un déplacement suivant une trajectoire radiale. Ce processus est répété jusqu'à l'usinage de toute la surface en déplaçant chaque fois le point de l'extrémité. La (Figure III.15) montre le principe de l'usinage Radiale et les lignes radiales sont les lignes suivies par l'outil en cours d'usinage.

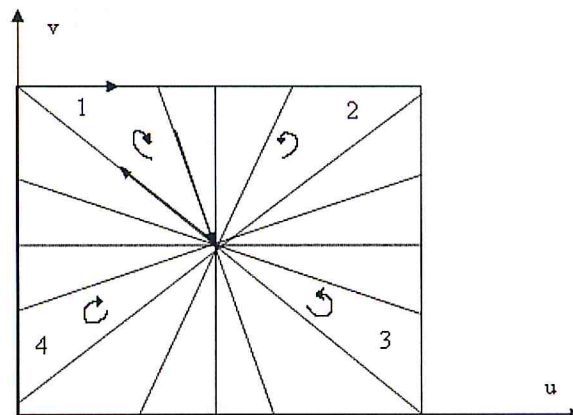


Figure III.15 : Usinage avec la méthode Radiale.

IV. AUTRES METHODES D'USINAGE [5, 14, 15]:

Autres méthodes d'usinage peuvent être utilisées à part les méthodes basées sur les courbes isoparamétriques.

IV.1. Usinage par la méthode des plans parallèles :

Dans cette technique, le trajet d'usinage est obtenu par la construction des courbes qui sont le résultat de l'intersection de la surface à usiner avec des plans verticaux parallèles au plan (XZ) (voir la Figure III.16). Les plans verticaux peuvent avoir n'importe quelle inclinaison.

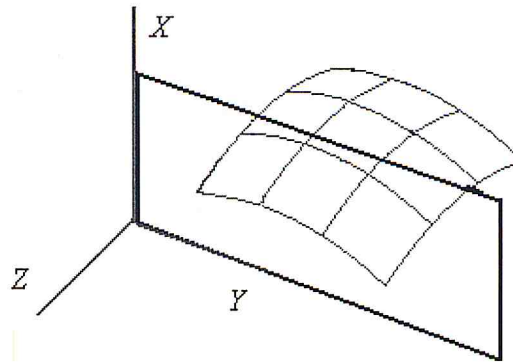


Figure III.16 : Méthode des plans parallèles.

IV.2. Usinage par la méthode Z-Constant :

Ce type de trajectoire est obtenu par intersection de la surface à usiner avec des plans horizontaux avec différentes valeurs de Z. Les courbes obtenues sont appelés « courbes de niveau » (voir la Figure III.17).

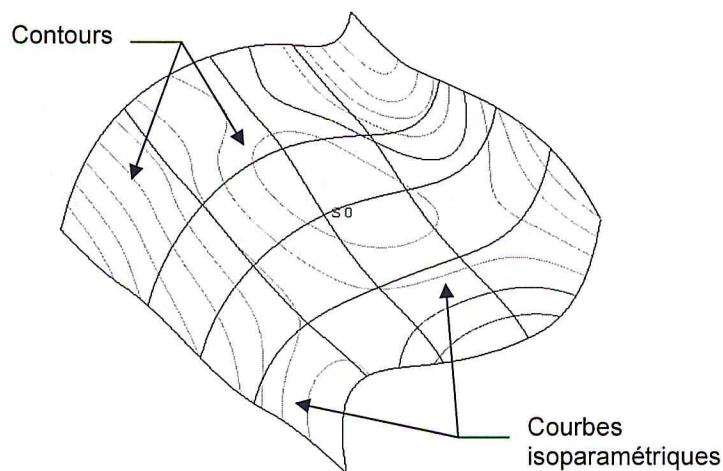


Figure III.17 : Méthode de Z-Constant.

V. DEFANTS D'USINAGE [5, 13-16]:

V.1. Erreur de flèche :

L'outil d'usinage ne peut pas usiner des courbes parfaites car ces dernières sont modélisées par interpolation linéaire, donc le résultat est simplement des segments de droite qui approximent plus ou moins les courbes théoriques en fonction de la précision demandée. Donc, le pas longitudinal d'usinage (un pas le long d'une passe d'usinage dans le sens de la longueur) est calculé en tenant compte de la tolérance d'usinage qui exprime la valeur maximale de l'erreur de flèche entre la courbe théorique et chacun des segments de la ligne brisée qui approximent cette courbe. La (Figure III.18) représente ce type de défaut.

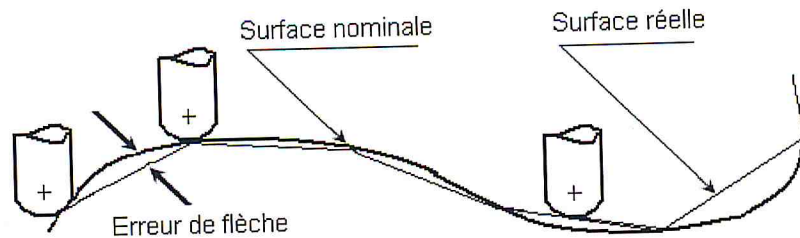


Figure III.18 : Erreur de flèche.

V.2. Hauteur de crêtes :

Pour usiner la totalité de la surface, les passes d'usinage doivent être liées par un pas transversal. 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. La (Figure III.19) représente les crêtes laissées entre les passes juxtaposées.

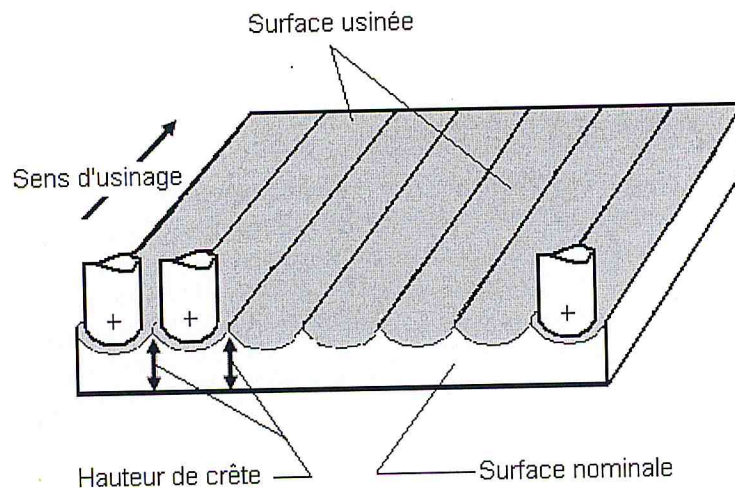


Figure III.19 : Hauteurs de crêtes.

V.3. Problème d'interférence :

Le phénomène d'interférence entre l'outil et la surface se produit dans les cas suivants :

- lorsqu'il existe une partie concave dont le rayon de courbure est inférieur à celui de l'outil, c'est l'interférence locale (voir la Figure III.20).
- dans le cas de raccordement entre les courbes de contact par leurs tangentes, c'est l'interférence globale (voir la Figure III.20).

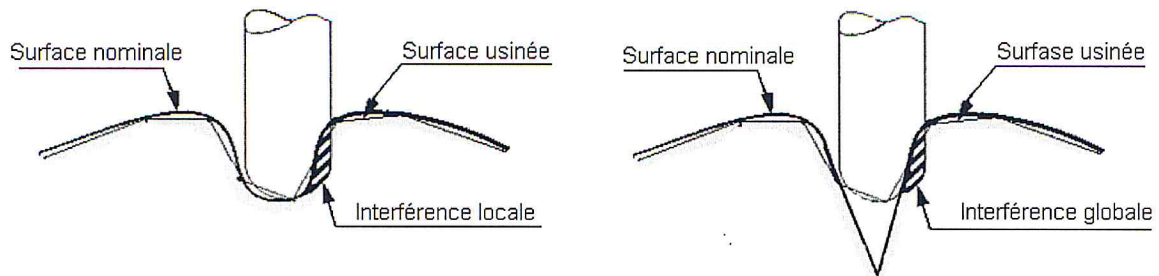


Figure III.20 : Interférence locale, et Interférence globale.

V.4. Problème de collision :

Ce genre de problème est observé dans les parties où l'angle formé par la normale à la surface au point de contact et l'axe de l'outil est supérieur à 90° . Dans ce cas, l'outil et la surface sont en collision (voir la Figure III.21).

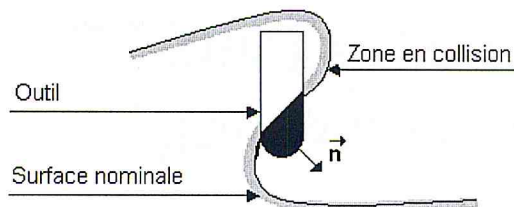


Figure III.21 : Collision entre outil et surface.

VI. QUALITE D'USINAGE DES SURFACES GAUCHES [16]:

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

- La précision des dimensions : lors de l'usinage on fait une correspondance entre les dimensions réelles de la pièce et les cotations issues du dessin lors de la phase de conception;
- Les écarts qu'on peut admettre de la forme géométrique ne doivent pas dépasser certaines limites imposées par les tolérances de formes;
- Les écarts de positions relatives des surfaces de la pièce (défaut de parallélisme, de perpendicularité, et l'écart d'angle imposé entre les plans,... etc.) ;

VII. CONCLUSION :

Dans ce chapitre nous avons présenté les différentes méthodes d'usinage utilisées dans l'opération de finition ainsi que critères à respecter afin d'obtenir une surface usinée qui s'approche plus de la surface théorique, en plus des problèmes d'interférence et de collision.

Avant d'arriver à l'opération de finition, il est nécessaire de passer par les opérations d'ébauchage et de demi finition. Dans le chapitre suivant nous allons voir en détail l'ébauchage des surfaces gauches.

I. INTRODUCTION :

Pour usiner une surface gauche, nous avons besoin de son modèle théorique obtenu dans la phase de conception (CAO). Ce modèle est exploité pour générer un code d'usinage qui sera chargé dans la machine à commande numérique afin qu'elle puisse réaliser l'usinage. Cet usinage n'est autre qu'un mouvement de balayage de l'outil, mais il reste à signaler que le déplacement de l'outil sur la surface n'est pas arbitraire, au contraire l'outil suit un trajet prédéfini par l'opérateur (le concepteur du code d'usinage). Ce trajet n'est autre qu'un ensemble de point issus de l'intersection des plans d'aiguillage avec la surface à usiner. Cette surface est approximée comme par des surfaces simples et régulières.

Le processus d'usinage des surfaces gauches se fait d'une manière organisée est ceci en répartissant le travail sur une série d'opérations dans le but de gagner du temps et d'obtenir une meilleure qualité du produit. Ces opérations doivent maintenir suffisamment de précision afin d'assurer la qualité d'usinage exigée.

II. APPROCHE ET STRATEGIE D'USINAGE DES SURFACES GAUCHES [17] :

La complexité des surfaces qui existent dans le milieu industriel rend leur usinage très difficile. Les difficultés surgissent souvent quand on essaye de déterminer l'opération d'usinage dans la phase de programmation. Pour cela, la fabrication des pièces passe par plusieurs étapes et ceci en prenant en compte des contraintes géométriques des pièces ainsi que le taux d'enlèvement de la matière dans le processus d'ébauche et le mouvement d'outil de coupe avec la précision requise dans le processus de finition.

Le processus d'usinage peut se traduire par la séquence d'opérations nécessaires pour l'enlèvement de la matière dans chaque plan de coupe :

- (1) Dans l'ébauche, utiliser de grandes fraises afin d'enlever le maximum de matière en minimum de temps.
- (2) En semi finition, utiliser de grandes fraises hémisphériques pour supprimer les épaulements laissés sur la surface après la phase d'ébauche et pour avoir une forme qui se rapproche plus de la forme théorique. Une certaine épaisseur d'usinage est laissée pour qu'elle soit traitée en finition (tolérance d'usinage).
- (3) En finition, utiliser de petites fraises hémisphériques pour enlever les parties qui restent afin d'avoir la forme voulue.
- (4) Si après la finition, il y a une quelconque région très courbée et qui ne peut être atteinte par n'importe quel outil disponible, on a recours à d'autres méthodes d'usinage comme l'usinage par électroérosion.
- (5) Si nécessaire, recourir au polissage afin d'égaliser la toute la surface et avoir un bon état de surface.

III. DEFINITION DE L'EBAUCHE [17]:

Puisque notre étude s'intéresse uniquement à l'opération d'ébauche, dans ce qui suit nous allons donner plus de détails sur cette opération très importante dans l'usinage des surfaces gauches.

L'ébauchage (dégrossissage) comme son l'indique c'est l'étape de préparation où l'on ne tient pas compte de la qualité d'usinage ni de la forme et ce qui nous importe le plus est l'enlèvement du maximum de matière en un temps réduit. L'opération d'ébauchage peut être faite par un outil cylindrique ou un outil hémisphérique :

- Ebauchage avec un outil cylindrique : l'ébauche de cavités avec un outil cylindrique ayant un grand rayon est une série d'évidement de poches dans les différents plans de coupe (plan d'usinage) et dont la profondeur de passe (d) est fixée par l'opérateur. Un trajet d'usinage peut être généré en déplaçant l'outil au dessus de la région à usiner. Il est clair que le meilleur moyen possible pour usiner les surfaces serait de le faire dans un mode continu; c'est-à-dire avec le minimum d'interruptions ou transversales rapides pendant l'opération d'usinage en commençant du haut vers le bas (voir la Figure IV.1).
- Ebauchage avec un outil hémisphérique : l'outil hémisphérique de grand diamètre se déplace premièrement vers le bas, ensuite se déplace le long de la surface à usiner avec un dégagement de finition "F" laissé sur la surface, jusqu'à ce qu'il touche le premier plan de coupe. Ensuite l'outil se déplace sur ce plan et par la suite l'outil se déplace le long de ce plan jusqu'à ce qu'il touche une autre partie de la surface. Le processus continu pour le premier plan de coupe jusqu'à la fin (voir la Figure IV.2. (a)). Après avoir achevé le premier plan de coupe l'outil attaque le deuxième plan (voir la Figure IV.2. (b)) et ainsi de suite pour le troisième plan (voir la Figure IV.2. (c)). Ce processus laisse seulement une couche de la tolérance de finition "F" sur la surface. Cette méthode fournit un moyen efficace pour enlever la plupart de la matière à l'intérieur des cavités, car il n'y a pas beaucoup de chevauchement entre les trajets d'usinages.

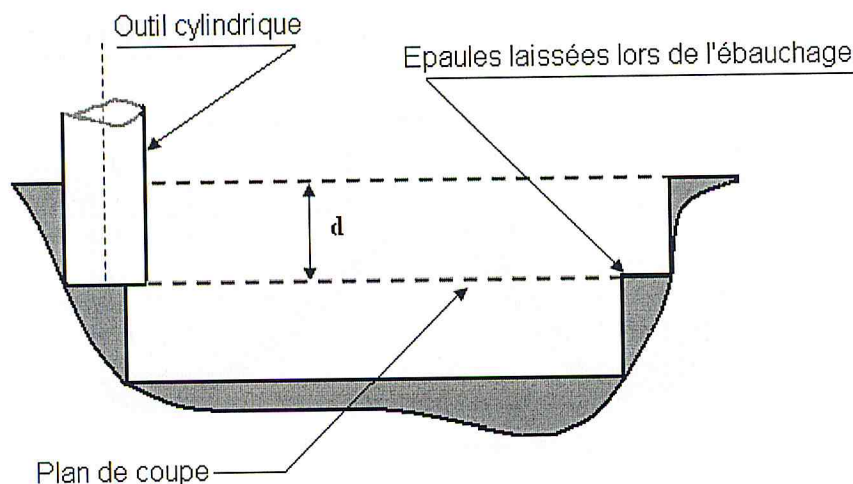
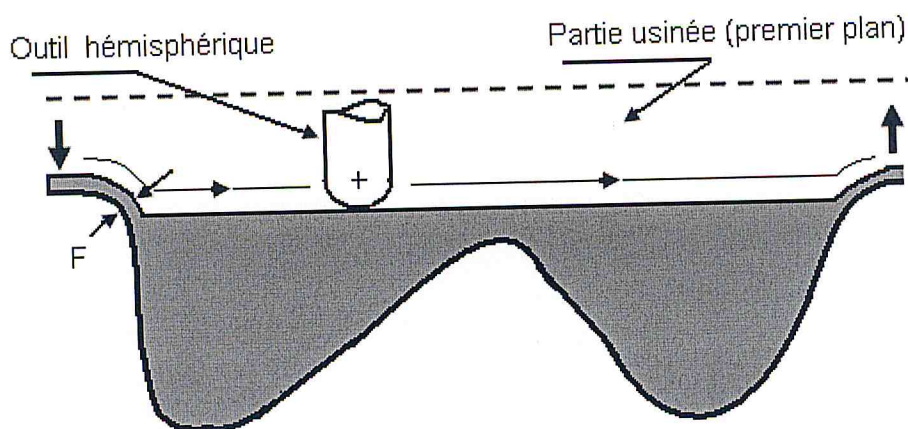


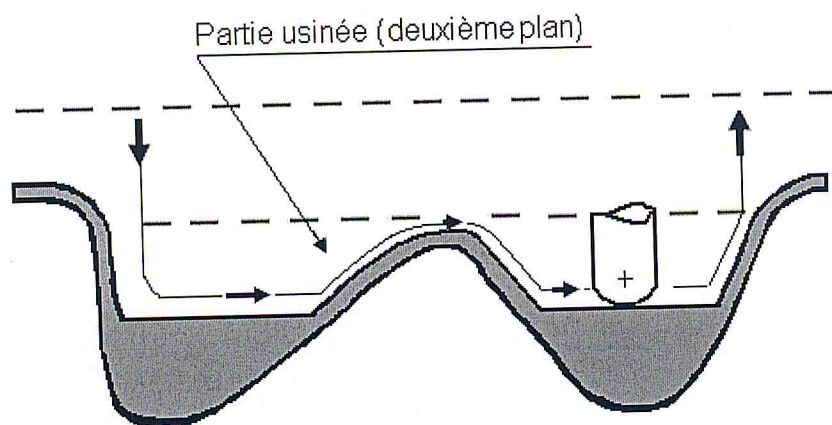
Figure IV.1: Ebauchage avec un outil cylindrique.

Chapitre IV

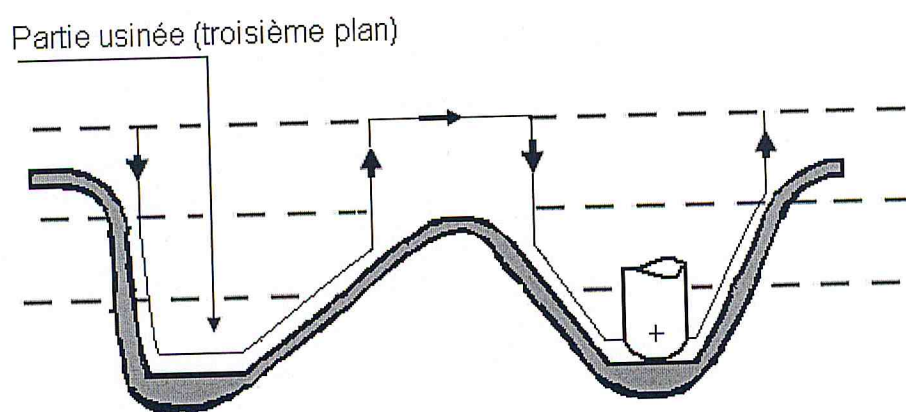
*Usinage des surfaces
gauches en ébauche.*



(a) premier plan usiné.



(b) deuxième plan usiné.



(c) troisième plan usiné.

Figure IV.2: Ebauchage avec un outil hémisphérique.

IV. PROCEDURE D'EBAUCHE ET LES OUTILS UTILISES [17]:

La robustesse de l'usinage de cavité dépend de la complexité géométrique et des contraintes d'usinage de la partie concave de la surface. Le processus d'usinage d'une cavité superficielle (pas trop courbée) est beaucoup plus simple qu'une cavité complexe et profonde. L'usinage d'une cavité large et profonde est différent de l'usinage d'une cavité petite et profonde. Une cavité très courbée nécessite un outil de petit rayon avec beaucoup de déplacement de ce dernier le long de la surface.

Les procédures d'usinage peuvent être déterminés par la complexité et l'aspect d'usinage de la partie concave de la surface. Une cavité peut être classée dans différents types. En considérant cette information d'usinage et le coût, la procédure d'usinage est basée sur le type de cavité.

IV.1. Sélection et classement de stratégie :

Pour le processus de fraisage, le coût d'usinage peut être calculé à partir du temps d'usinage et la durée de vie de l'outil. La sélection de ce dernier et la génération du trajet d'usinage sont décidés sur la base du coût d'usinage. On suppose que le trajet d'usinage le plus économique est celui qui nécessite un minimum de temps d'usinage.

Soient :

R_i : le rayon de l'outil i pour un processus d'usinage donné :

TR_i : le temps d'usinage avec l'outil i ,

TC_i : le temps de changement de l'outil i .

Le temps d'usinage total T_m lors de l'usinage avec n outils est donné par :

$$T_m = (TR_1 + TC_1) + (TR_2 + TC_2) + (TR_3 + TC_3) + \dots + (TR_n + TC_n) \quad (IV.1)$$

Pour un centre d'usinage équipé avec un changeur automatique d'outils, le temps de changement de l'outil est très petit devant le temps d'usinage avec le même outil ($TC_i \ll TR_i$). Donc TR_i est le facteur majeur du coût d'usinage. Cependant, ce n'est pas une bonne pratique d'usinage d'avoir plusieurs changements d'outils pendant l'usinage, car ceci augmente le coût d'usinage et le nombre d'outils à mettre dans le magasin à outils.

Le temps d'usinage peut être évalué par le taux d'enlèvement de la matière; c'est-à-dire plus la quantité de la matière enlevée est grande, plus le temps d'usinage est petit. Par conséquent, la meilleur politique d'usinage est de choisir le plus large outil possible sous des contraintes géométriques.

A partir de l'évaluation des informations d'usinage acquises de la base de données de conception des surfaces formant notre pièce (profondeur et nombre des cavités, le nombre et la hauteur des îlots), on peut faire une classification dont les codes de cette dernière peuvent être utilisés pour indiquer les procédures d'usinage. Par exemple le code de classification d'une cavité avec un îlot large et profond est **001**, et celui d'une cavité sans îlot étroite et superficielle est: **110** (voir la Figure IV.4). Suivant la classe, on peut citer quatre (04) différentes procédures d'usinage (Tableau IV.3) :

Tableau IV.3: Les différentes procédures d'usinage et les outil utilisés.

| Politique | Opération | Outil utilisé | Diamètre de l'outil |
|-----------|-----------------|-------------------|---|
| (a) | Ebauche | Cylindrique -C- | Large |
| | Semi - finition | Hémisphérique -H- | Large |
| | Finition | Hémisphérique -H- | Petit |
| (b) | Ebauche | Cylindrique -C- | Large |
| | Finition | Hémisphérique -H- | Quelconque |
| (c) | Ebauche | Hémisphérique -H- | Large |
| | Finition | Hémisphérique -H- | Petit |
| (d) | Ebauche | Hémisphérique -H- | Même outil pour les deux opérations d'usinage |
| | Finition | Hémisphérique -H- | |

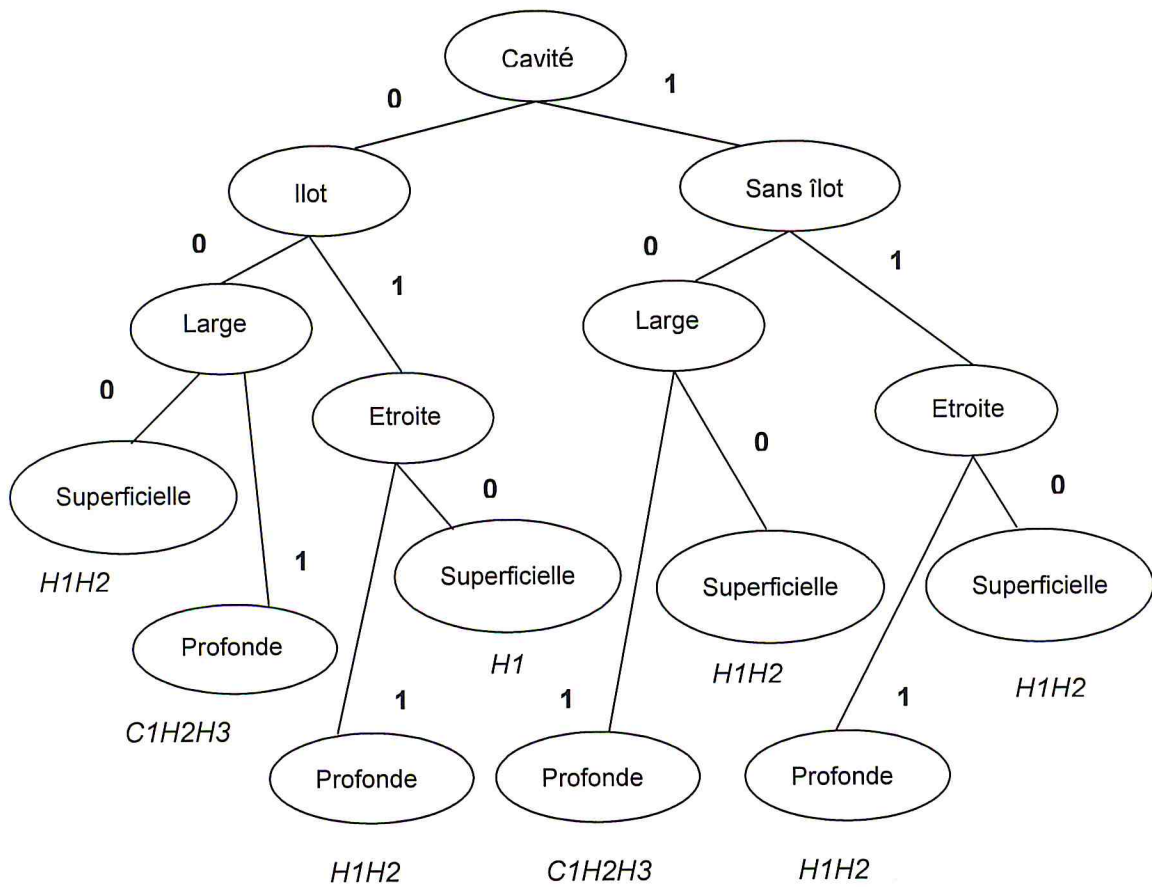


Figure IV.4: Codes de classification des différentes formes possibles et les outils adéquats à leur usinage).

IV.2. Sélection de l'outil pour l'usinage des cavités :

Dans l'usinage d'une cavité le processus d'ébauche a été divisé (comme il a été dit précédemment) en une série de processus d'enlèvement de la matière pour un ensemble de plans, où chaque plan pourrait utiliser un outil de différente taille. Dans la (Figure IV.4), les notations 1, 2, 3, C, et H représentent les différents outils utilisés tel que :

C : Représente un outil cylindrique.

H : Représente un outil hémisphérique.

1 : Un outil de grand diamètre est utilisé en premier.

2 : Un outil de petit diamètre par rapport au premier est utilisé après avoir utilisé le premier, ...etc.

Reprenant les exemples du paragraphe précédent : une cavité profonde avec un îlot large (**001**) sera usinée en commençant d'abord par l'utilisation d'un outil cylindrique de grand diamètre, ensuite un outil hémisphérique de diamètre inférieur au premier, et terminer l'usinage avec un outil hémisphérique de diamètre inférieur aux diamètres des deux outils précédents, ce qui correspond dans la (Figure IV.4) à C1H2H3. La même chose pour une cavité superficielle et étroite et sans îlot (**110**) elle sera usinée par deux outils hémisphériques dont le diamètre du premier est inférieur au diamètre du second H1H2.

Pour l'optimisation de la sélection de l'outil, nous pouvons suivre les étapes suivantes (Bala et Chang (1988)) :

- (1) Trouver les outils appropriés pour chaque plan de coupe (en évaluant les contraintes géométrique de la surface).
- (2) Fusionner les outils choisis pour les plans de coupe afin d'optimiser la sélection d'outil et réduire les changements d'outil non requis.
- (3) Comparer les hauteurs z (coordonnées suivant l'axe z) de chaque plan de coupe avec la profondeur de coupe pour le choix de l'outil et trouver la hauteur z du plan suivant.
- (4) Décider de la taille de l'outil pour chaque plan.

V. LA TRIANGULATION [5, 18]:

On a vu dans la phase d'ébauche qu'on doit spécifier la trajectoire suivie par l'outil du début de l'usinage jusqu'à la fin. Ce trajet peut être défini comme un enchaînement d'intersections entre la surface à usiner avec des plans de guidages avec une profondeur de passe donnée. Cela semble très évident dans le cas où on a affaire à des formes simples, mais ce qui n'est pas le cas dans l'usinage des surfaces gauches. Pour arriver à ébaucher ces surfaces, nous sommes obligés de passer par une étape d'approximation de ces surfaces compliquées par des surfaces plus simples afin de trouver facilement l'ensemble de points constituant le trajet d'outil. La surface élémentaire la plus simple qui vient intuitivement dans l'esprit est un triangle. Donc, la

surface à usiner est vue comme un maillage de triangles (voir la Figure IV.5) où chaque sommet du triangle est un point de la surface à usinée. Cette étape c'est la "triangulation". Donc, la triangulation d'une surface donnée c'est l'approximation de cette dernière par un ensemble de triangles avec une certaine précision fixée au départ.

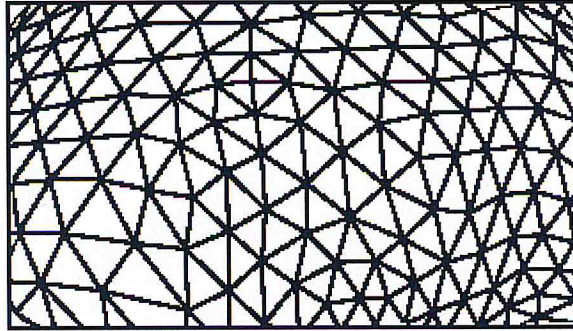


Figure IV.5 : Triangulation d'une forme quelconque.

V.1. Les types de triangulation :

Il existe deux types de triangulation; la triangulation uniforme et la triangulation adaptative.

V.1.1. La triangulation uniforme :

Dans ce type de triangulation, comme son nom l'indique le nombre de triangles est défini uniformément, c'est-à-dire le nombre de triangles suivant chaque direction (u et v) est fixe. La (Figure IV.6.1) représente une surface ondulée théorique tandis que la (Figure IV.6.2) représente la triangulation uniforme de cette dernière où on a utilisé 100 triangles suivant chaque direction u et v . Le nombre de triangles suivant u peut être choisi différent de celui de la direction v .

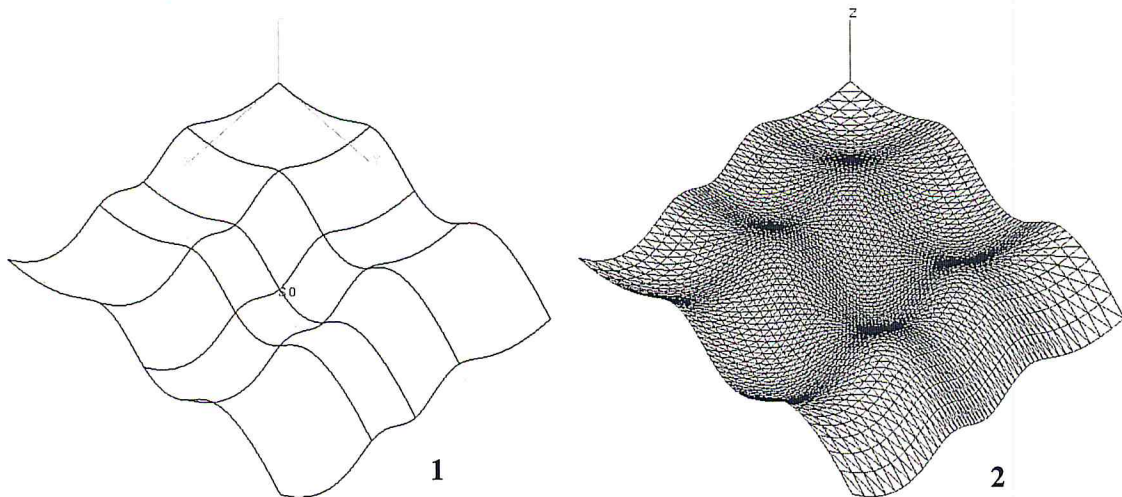


Figure IV.6: Triangulation uniforme d'une surface ondulée.

V.1.2. La triangulation adaptative :

Dans ce type de triangulation, les triangles sont créés d'une manière dynamique et adaptative. Donc, le nombre de triangles suivant la direction u n'est pas forcément égal à celui suivant v . La figure suivante montre la même surface ondulée (voir la Figure IV.6.1) selon la triangulation adaptative en utilisant les paramètres suivants : la longueur de chaque coté du triangle est égale a 5 mm et l'erreur de triangulation est de 5mm (voir la Figure IV.7.1). La (Figure IV.7.2) montre la triangulation adaptative mais cette fois-ci avec les paramètres suivants ; la longueur de chaque coté est égale à 10mm et l'erreur de triangulation est de 8mm. Il reste à signaler que ces paramètres sont définis par l'utilisateur en fonction de la précision avec laquelle il veut approximer la surface théorique.

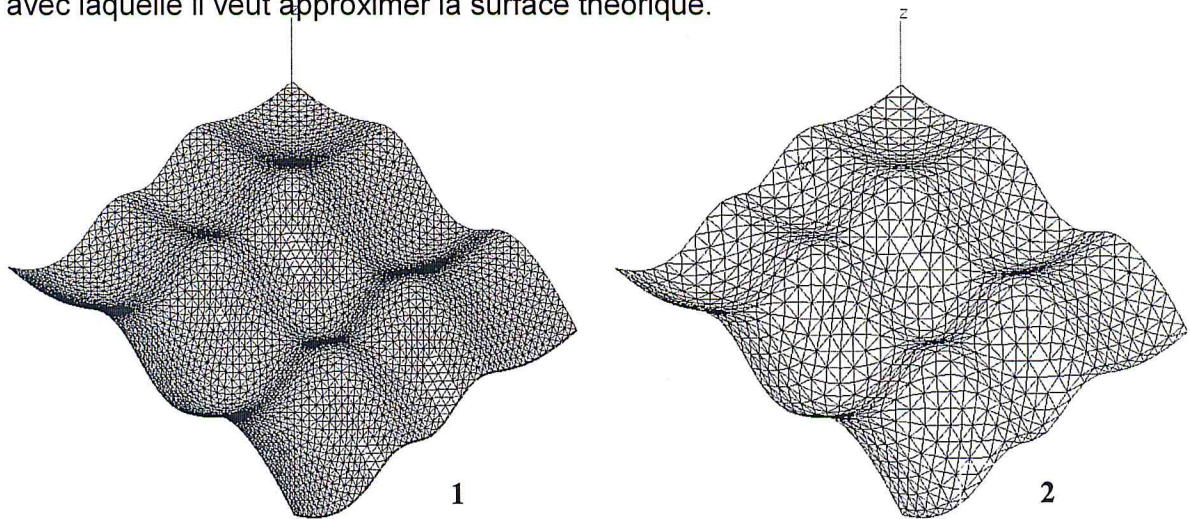


Figure IV.7: Triangulation adaptative d'une surface ondulée.

Pendant chaque triangle obtenu doit répondre à sept (07) critères, sinon il sera triangulé à nouveau.

La taille de chaque coté ne doit pas dépasser une certaine borne imposée par le programmeur ("d" la valeur de cette borne), la distance entre la surface à usiner et le milieu de chaque coté du triangle ainsi que son centre de gravité doivent aussi vérifier une certaine erreur imposée par l'utilisateur (d'). Donc, pour chaque triangle on a :

1. (03) cotés dont la taille est inférieure à d ,
2. (03) cotés dont la distance entre leurs milieux et la surface approximée est inférieure à d' ,
3. son centre de gravité à se trouve à une distance inférieure à d' par rapport à la surface.

Par ailleurs, si le triangle obtenu ne répond pas aux critères cités ci-dessus il est subdivisé en quatre triangles. Si le test échoue encore pour un ou deux côtés, alors le triangle est subdivisé en deux sinon en trois triangles, jusqu'à obtention d'un triangle répondant à tous les critères.

Dans les triangulations de la (Figure IV.8), il existe des côtés qui sont hors tolérance ainsi que le centre de gravité du triangle. En fonction des vérifications à faire, un triangle peut être subdivisé suivant quatre formes :

- (1): Un seul côté est hors tolérance.
- (2): Deux côtés sont hors tolérance.
- (3): Tous les côtés sont hors tolérance.
- (4): Le centre de gravité est hors tolérance.

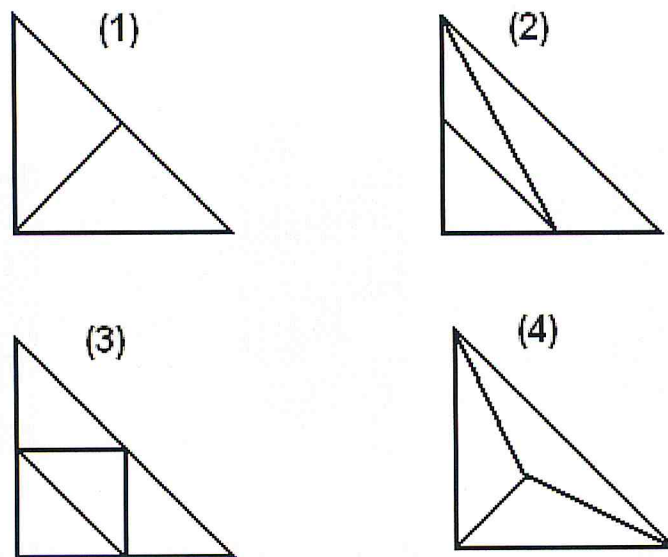
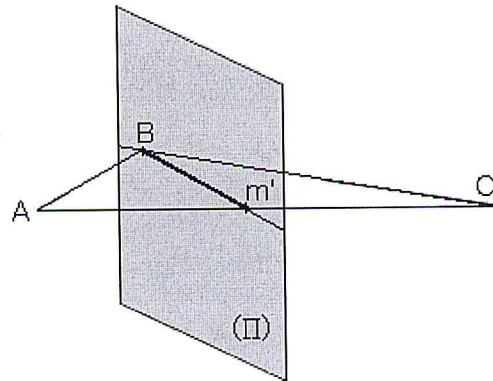
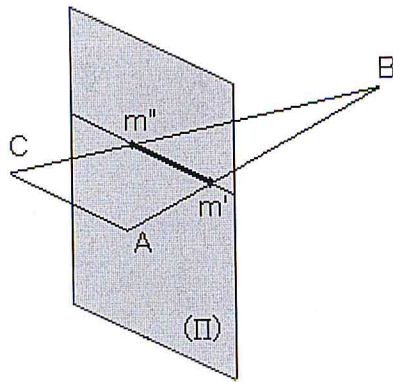


Figure IV.8: Les différents cas d'anomalies dans la triangulation adaptative.

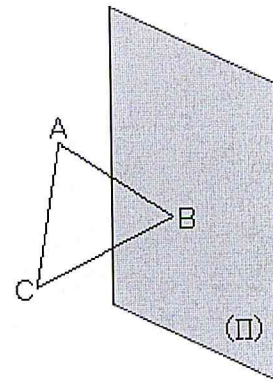
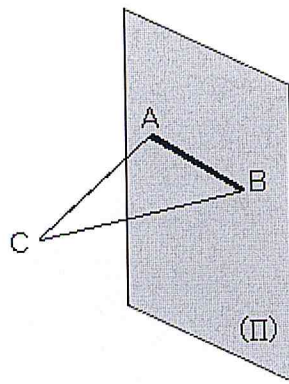
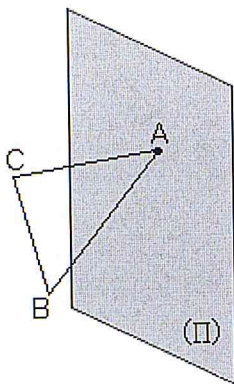
VI. CALCUL DES INTERSECTIONS [5, 19]:

Dans cette partie, nous allons étudier l'intersection entre chaque triangle issu de l'étape de triangulation avec un plan quelconque afin de définir tous les contours de la surface à usiner. Pour qu'un plan (π) intersecte un triangle, il faut et il suffit qu'il intersecte au moins un segment du triangle. Pour que l'intersection entre un plan et un triangle soit un ensemble vide, il faut qu'il existe deux segments de triangle tels que l'intersection entre chacun d'eux et le plan soit un ensemble vide. La (Figure IV.9) illustre les différents résultats de l'intersection entre un plan et un triangle.



- 1) $[CB] \cap (\Pi) = \{m''\}$.
 $[BA] \cap (\Pi) = \{m'\}$.
 $[CA] \cap (\Pi) = \{\}$.
 $(ABC) \cap (\Pi) = [m' m'']$.

- 2) $[AB] \cap (\Pi) = \{B\}$.
 $[BC] \cap (\Pi) = \{B\}$.
 $[AC] \cap (\Pi) = \{m'\}$.
 $(ABC) \cap (\Pi) = [B m'']$.



- 3) $[CA] \cap (\Pi) = \{A\}$.
 $[AB] \cap (\Pi) = \{A\}$.
 $[CB] \cap (\Pi) = \{\}$.
 $(ABC) \cap (\Pi) = \{A\}$.

- 4) $[AB] \cap (\Pi) = [AB]$.
 $[BC] \cap (\Pi) = \{B\}$.
 $[AC] \cap (\Pi) = \{A\}$.
 $(ABC) \cap (\Pi) = [AB]$.

- 5) $[AB] \cap (\Pi) = \{\}$.
 $[AC] \cap (\Pi) = \{\}$.
 $[CB] \cap (\Pi) = \{\}$.
 $(ABC) \cap (\Pi) = \{\}$.

Pour trouver l'intersection entre un plan et un triangle il faut avoir l'équation du segment du triar
 Figure IV.9. Les différents cas d'intersections d'un plan avec un triangle.

VI.1. Plan dans l'espace :

L'équation d'un plan dans l'espace est donnée par la formule suivante :

$$Ax + By + Cz + D = 0. \tag{IV.2}$$

Dans notre cas, le plan est horizontal et est parallèle au plan XY voir (Figure IV.10). Donc, nous avons directement : $A=B=0$, et l'équation du plan devient :

$$CZ + D = 0 \tag{IV.3}$$

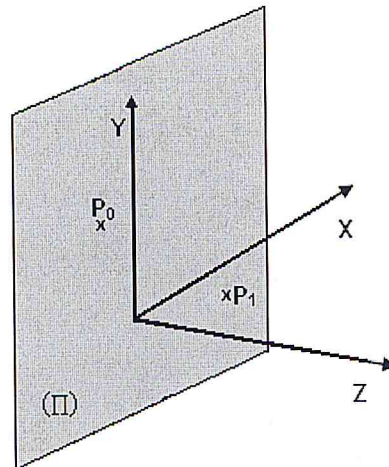


Figure IV.10: Intersection d'un plan passant par les deux points P_0, P_1 et le plan XY.

VI.2. Droite dans l'espace :

Pour avoir une droite dans un espace R^3 , il nous suffit de faire le raccordement de deux points quelconques P_0 et P_1 . L'équation de cette droite est donnée par :

$$\vec{P_0 M} \parallel \vec{P_0 P_1} \tag{IV.4}$$

Où $M(x,y,z)$ est un point appartenant à la droite formée en passant par les deux points $P_0(x_0,y_0,z_0)$, $P_1(x_1,y_1,z_1)$ (voir la Figure IV.11).



Figure IV.11 : Représentation d'une droite dans l'espace

$$M \in (P_0 P_1) \Leftrightarrow P_0 M \parallel P_1 M \Leftrightarrow \exists \alpha \in \mathbb{R} / \begin{vmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{vmatrix} = \alpha \Leftrightarrow \begin{vmatrix} x_1 - x_0 \\ y_1 - y_0 \\ z_1 - z_0 \end{vmatrix} \tag{IV.5}$$

$$\text{Donc: } \begin{cases} x - x_0 = \alpha(x_1 - x_0) \\ y - y_0 = \alpha(y_1 - y_0) \\ z - z_0 = \alpha(z_1 - z_0) \end{cases} \Leftrightarrow \begin{cases} \alpha = \frac{x - x_0}{x_1 - x_0} & \text{si } x_1 \neq x_0 \\ \alpha = \frac{y - y_0}{y_1 - y_0} & \text{si } y_1 \neq y_0 \\ \alpha = \frac{z - z_0}{z_1 - z_0} & \text{si } z_1 \neq z_0 \end{cases} \tag{IV.6}$$

$$\text{D'où : } \frac{x-x_0}{x_1-x_0} = \frac{y-y_0}{y_1-y_0} = \frac{z-z_0}{z_1-z_0} \text{ on a : } \begin{cases} (x-x_0)(y_1-y_0) = (x_1-x_0)(y-y_0)...(1') \\ (y-y_0)(z_1-z_0) = (y_1-y_0)(z-z_0)...(2') \\ (x-x_0)(z_1-z_0) = (x_1-x_0)(z-z_0)...(3') \end{cases} \quad (\text{IV.7})$$

Le système d'équations précédent présente des redondances, on peut donc éliminer l'une d'elle selon les cas suivants :

- 1) Si $x_1 \neq x_0, y_1 = y_0, z_1 = z_0$: la droite trouvée est parallèle à la l'axe (OX)

$$\begin{cases} y - y_0 = 0 \\ z - z_0 = 0 \end{cases} \quad (\text{IV.8})$$

- 2) Si $x_1 = x_0, y_1 \neq y_0, z_1 = z_0$: la droite trouvée est parallèle à la l'axe (OY)

$$\begin{cases} x - x_0 = 0 \\ z - z_0 = 0 \end{cases} \quad (\text{IV.9})$$

- 3) Si $x_1 = x_0, y_1 = y_0, z_1 \neq z_0$: la droite trouvée est parallèle à la l'axe (OZ)

$$\begin{cases} x - x_0 = 0 \\ y - y_0 = 0 \end{cases} \quad (\text{IV.10})$$

- 4) Si $x_1 \neq x_0, y_1 \neq y_0, z_1 = z_0$:

$$\begin{cases} (x-x_0).(y_1-y_0) = (x_1-x_0).(y-y_0) \\ (y-y_0).0 = (y_1-y_0).(z-z_0) \\ (x-x_0).0 = (x_1-x_0).(z-z_0) \end{cases} \Leftrightarrow \begin{cases} (x-x_0).(y_1-y_0) = (x_1-x_0).(y-y_0) \\ (y_1-y_0).(z-z_0) = 0 \\ (x_1-x_0).(z-z_0) = 0 \end{cases} \quad (\text{IV.11})$$

La formule ci-dessus est obtenue en remplaçant : $z_1 = z_0 \Leftrightarrow z_1 - z_0 = 0$. En utilisant $x_1 \neq x_0, y_1 \neq y_0$, la formule devient :

$$\begin{cases} (x-x_0)(y_1-y_0) = (x_1-x_0)(y-y_0) \\ z - z_0 = 0 \\ z - z_0 = 0 \end{cases} \Leftrightarrow \begin{cases} x(y_1-y_0) - y(x_1-x_0) = x_0(y_1-y_0) + y_0(x_1-x_0) \\ z - z_0 = 0 \end{cases} \quad (\text{IV.12})$$

- 5) Si $x_1 \neq x_0, y_1 = y_0, z_1 \neq z_0$: de la même manière, on trouve :

$$\begin{cases} x(z_1-z_0) - z(x_1-x_0) - x_0(z_1-z_0) + z_0(x_1-x_0) = 0 \\ y - y_0 = 0 \end{cases} \quad (\text{IV.13})$$

- 6) Si $x_1 = x_0, y_1 \neq y_0, z_1 \neq z_0$:

$$\begin{cases} y(z_1-z_0) - z(y_1-y_0) - y_0(z_1-z_0) + z_0(y_1-y_0) = 0 \\ x - x_0 = 0 \end{cases} \quad (\text{IV.14})$$

- 7) Si $x_1 \neq x_0, y_1 \neq y_0, z_1 \neq z_0$: on peut éliminer n'importe laquelle, en choisissant de supprimer la ligne (2') de l'équation (IV. 7) on obtient.

$$\begin{cases} (y_1 - y_0)x - (x_1 - x_0)y - (y_1 - y_0)x_0 + (x_1 - x_0)y_0 = 0 \\ (z_1 - z_0)x - (x_1 - x_0)z - (z_1 - z_0)x_0 + (x_1 - x_0)z_0 = 0 \end{cases} \quad (IV.15)$$

Après avoir trouvé l'équation du plan, il est facile maintenant de faire l'intersection entre ce plan avec un triangle, cette intersection n'est d'autre que la résolution d'un système d'équations, l'équation du plan et l'équation d'un segment de triangle.

Soit le segment [P₀P₁] appartenant au triangle P₀P₁P₂, supposons maintenant le système d'équation de la droite passant par [P₀P₁] :

$$\begin{cases} A'x + B'y + C'z + d' = 0 \\ A''x + B''y + C''z + d'' = 0 \end{cases} \quad (IV.16)$$

Soit P un point appartenant au segment [P₀P₁], l'intersection du plan avec cette droite (le point P avec P₀ ou P₁) est résumé dans l'équation ci-dessous :

$$\begin{cases} Ax + By + Cz + d = 0 \\ A'x + B'y + C'z + d' = 0 \\ A''x + B''y + C''z + d'' = 0 \end{cases} \quad (IV.17)$$

Résolution :

- Si le système n'admet pas de solution donc : (P₀P) ∩ (π) = ∅, d'où [P₀P₁] ∩ (π) = ∅.
- Si le système admet une infinité de solution donc : (P₀P) ∩ (π) = (P₀P), d'où [P₀P₁] ∩ (π) = [P₀P₁].
- Si le system admet une solution unique alors (P₀P) ∩ (π) = {m} dont on vérifie si {m} ∈ [P₀P₁]. Si m ∈ [P₀P₁] alors [P₀P₁] ∩ (π) = {m} sinon [P₀P₁] ∩ (π) = ∅.

VI.2.1. Appartenance d'un point à un segment de droite :

Soit [AB] un segment de droite dans l'espace (voir la (Figure IV.12)), on dit que :

$$M \in [AB] \Leftrightarrow \exists \alpha / 0 \leq \alpha \leq 1 \quad \overrightarrow{AM} = \alpha \cdot \overrightarrow{AB} \quad \begin{cases} M_1 \in [AB] & \text{et } 0 \leq \alpha \leq 1 \\ M_2 \notin [AB] & \text{car } \alpha \geq 1 \\ M_3 \notin [AB] & \text{car } \alpha < 0 \end{cases} \quad (IV.18)$$

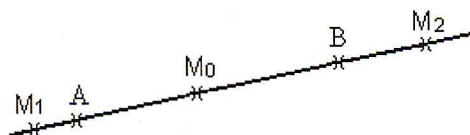


Figure IV.12 : Position des différents points par rapport à AB.

De la relation précédente, on peut déduire les relations suivantes :

$$\overrightarrow{AM} = \alpha \overrightarrow{AB} \quad 0 \leq \alpha \leq 1.$$

On pose A (x_A, y_A, z_A), B (x_B, y_B, z_B) et M(x, y, z) :

$$\text{Donc on a : } \begin{pmatrix} x - x_A \\ y - y_A \\ z - z_A \end{pmatrix} = \alpha \begin{pmatrix} x_B - x_A \\ y_B - y_A \\ z_B - z_A \end{pmatrix} \quad \text{et } 0 \leq \alpha \leq 1 \quad (\text{IV.19})$$

$$\text{Donc } \frac{x - x_A}{x_B - x_A} = \frac{y - y_A}{y_B - y_A} = \frac{z - z_A}{z_B - z_A} = \alpha$$

Donc il suffit que :

$$0 \leq \frac{x - x_A}{x_B - x_A} \leq 1 \quad \text{ou} \quad 0 \leq \frac{y - y_A}{y_B - y_A} \leq 1 \quad \text{ou} \quad 0 \leq \frac{z - z_A}{z_B - z_A} \leq 1.$$

VII. CONCLUSION :

Nous avons vu dans ce chapitre les différents phases de l'usinage, et nous avons détaillé la phase d'ébauche avec une classification des différents types de surfaces et les outils utilisés. Nous avons constaté que la meilleure politique d'usinage et de choisir le plus large outil possible sous les contraintes géométriques. La fin de ce chapitre a traité les étapes de triangulation et d'intersection des triangles avec des plans horizontaux.

Le chapitre suivant sera consacré à l'architecture de toute l'application (les données, les différentes fonctions et l'interaction entre ces dernières) réalisée par cette étude.

Chapitre V

*Conception de
l'application.*

I. INTRODUCTION :

Qu'est-ce qu'un modèle ?

Un modèle est une représentation simplifiée d'un système donné, afin de mieux comprendre son fonctionnement, on modélise un système (parce que qu'on) appelle langage de modélisation. Selon le découpage des informations, un modèle est dit soit fonctionnel ou orienté objet. Une modélisation est dite objet si elle fait ressortir les objets du langage naturel.

Pourquoi créer un modèle ?

Modéliser un système revient à le simplifier afin de mieux le comprendre en faisant abstraction (abstraction : cacher ce qui n'est pas considéré comme important, ou ce qui est du détail par rapport à notre analyse) aux objets non utiles dans notre système (voir figure V.1).

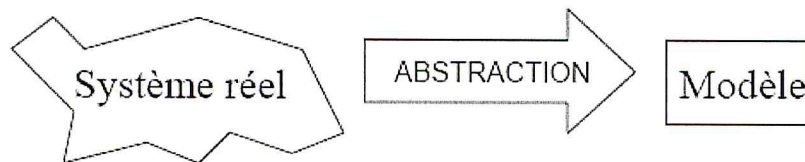


Figure V.1 : schéma représentant la modélisation d'objet

Créer un modèle revient à présenter un résultat compréhensif par les clients, ainsi de spécifier un guide à la construction du système interprétable par toute l'équipe de développement et aussi assurer une maintenance facile et efficace permettant le contrôle de l'avancement du projet.

II. MODELISATION OBJET [20-24] :

Son but est de décrire les objets qui forment le système par leurs interactions. Un objet est une abstraction ayant des limites déterminées et un sens propre dans le système, chaque objet possède une identité unique distinguée des autres.

II.1. PHASES DE MODELISATION OBJET :

Comme toute autre méthode elle passe par un ensemble d'étapes, les étapes clés sont citées ci-dessous :

- Préparer le cahier des charges;
- préparer le dictionnaire de données;
- Identifier les objets et leurs attributs;
- Identifier les associations entre les objets;
- Itérer et affiner le modèle.

Q'apporte la modélisation Objet ?

La modélisation objet réduit l'écart entre le langage utilisateur et le langage conceptuel ainsi qu'une simplification lors de la transformation du niveau conceptuel au niveau physique, et aussi assure une indépendance du modèle par rapport aux fonctionnalités demandées (ajouts, modification, suppression...etc.) sans affecter le modèle en entier.

III. ETUDE D'UNE METHODE DE MODELISATION :

Lors de la mise en œuvre de notre application on a utilisé comme outil de modélisation le langage UML, pour plus de détails sur ce langage consulter plus loin Annexe II.

IV. REALISATION DE L'APPLICATION :

IV.1. Etablissement du cahier de charge [24-27]:

IV.1.1. Présentation du projet en globale :

Le projet en global entre dans le cadre de la Conception et la Fabrication Assistées par Ordinateur (CFAO) chargé par l'équipe **CFAO** du Centre de Développement des Technologies Avancés (CDTA). Son but consiste à automatiser la production mécanique (pièces usinées) par l'intégration des principales fonctions « conception, planification, fabrication » dans le cadre du processus de production.

IV.1.2. Description du projet :

Le processus de fabrication passe par l'étape de conception en traduisant le concept 3D de la pièce à usiner en une suite d'ordres compréhensifs par la machine. Notre but est de réaliser ce passage en étape d'ébauchage, le thème de notre projet s'intitule « Automatisation de l'opération d'ébauchage des surfaces gauches sur des fraiseuses à commande numérique à 3 axes »

c'est pas 2 façon de reperçer

IV.1.3. Définition du problème :

Le problème qui reste à résoudre est comment se fera ce passage ? Ou plus précisément comment on passe du concept 3D en commandes d'usinage G-Code ? Dans l'étape d'ébauchage qui possède ses propres méthodes (usinage en Z-Constant par One-Way ou Zig-Zag...) ces propres contraintes (dégrossissage maximal) et ces propres outils (cylindrique, sphérique en semi finition).

IV.1.4. Les objectifs visés :

Notre but majeur est la génération du programme G-Code, qui peut être divisé en sous objectifs. Un fichier G-Code contient une suite de commandes de passage d'un point à un autre. Les objectifs à atteindre sont les suivants :

1. premier objectif est l'extraction de cet ensemble de points. Chaque point a ses propres paramètres (vitesse d'avance, vitesse d'usinage, la normale, ... etc.).
2. deuxième objectif restitution des paramètres de chaque point (rayon de courbure, tangente, ...)
3. troisième objectif utiliser les méthodes d'usinages en ébauchage et élimination des erreurs d'usinage.
4. quatrième objectif élimination des erreurs d'interférences et de crêtes.
5. cinquième objectif minimisé le temps d'usinage,

Ces principaux objectifs induit l'existence d'autres sous objectifs qui seront traités plus loin.

IV.1.5. Description de la plateforme existante :

La plateforme est un PC compatible doté d'un processeur Intel Pentium 4 de 2.25 GHz et d'un cache de 256 Ko L2, d'une mémoire vive de 512 Mo DDRAM d'un circuit graphique S3 Savage gérant la 3D d'un disque dur de 40 Go d'un écran MAG innovision CRT de 19" (pouce) et d'un onduleur ECM, tournant sous Windows Expert (XP) Professionnel SP2 (2003) de Microsoft. L'environnement de développement est un IDE (integrated Development Environment) de Borland du langage C++ est le Builder C++ dans la sa version 6 (2002).

IV.2. Solution de la problématique :

- Premier Objectif : l'extraction des points se fait par le calcul d'intersection entre un ensemble de droites et les contours décalés de la surface à usiner comme indiqué par la (Figure V.19).

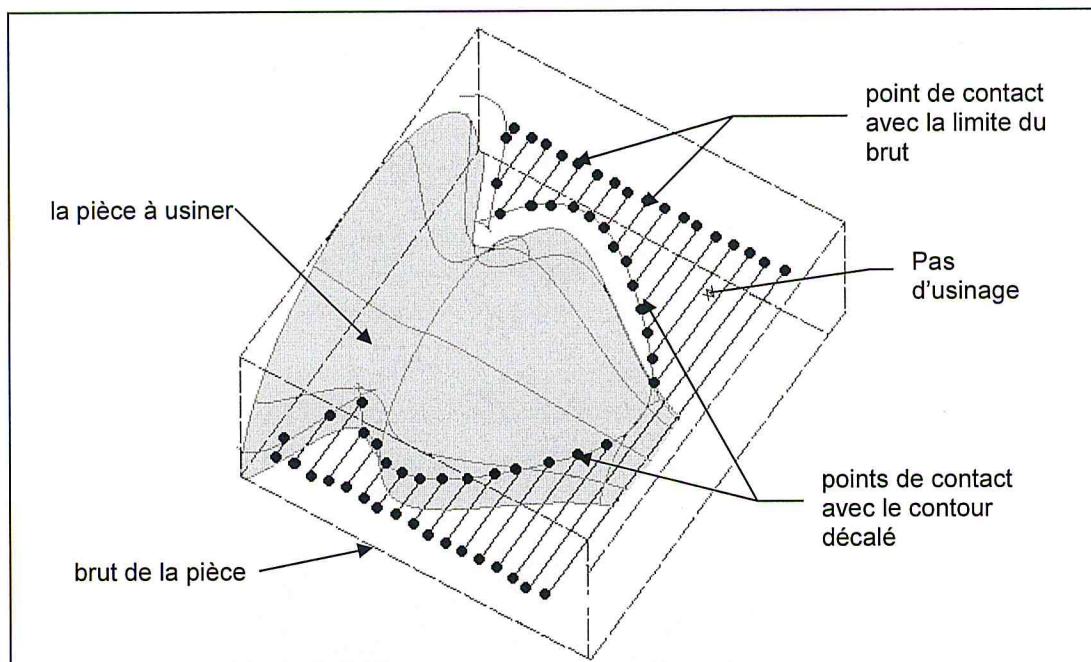


Figure V.19 : Extraction des points de passage de l'outil.

- Deuxième objectif : la récupération des paramètres des points se fait par la projection des paramètres de chaque point du contour sur son semblable sur le contour décalé puisque se qui est appliqué au point du contour décalé influera sur celui du contour.

- Troisième objectif : exploitation des méthodes d'usinage en ébauchage Zig-Zag, One-Way voir (Figure V.20).

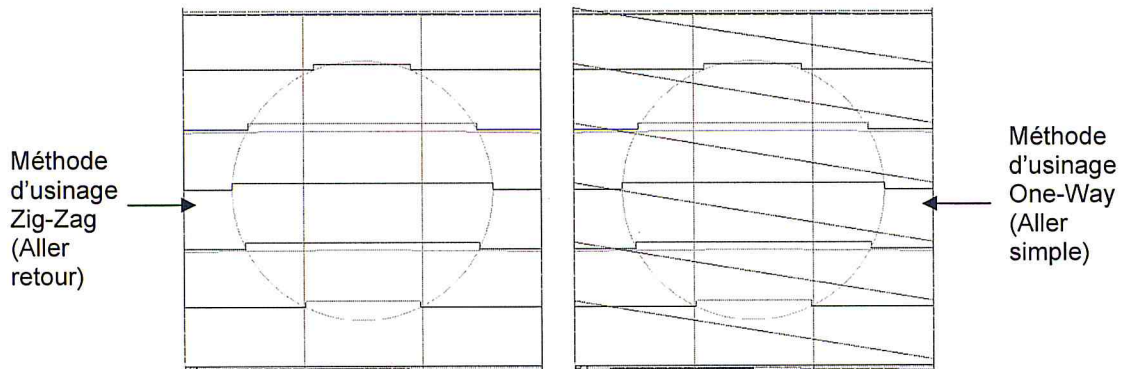


Figure V.20: Exposition des deux méthodes d'usinage

- Quatrième objectif : la suppression des zones non usinées est faite par le choix d'un pas respectant la contrainte suivante : $Pas < 2 * R$, avec Pas définissant le pas d'usinage (distance entre les droites d'usinage voir (Figure V.19) et R le rayon de l'outil. Les erreurs d'interférence sont éliminées par le choix d'un outil de diamètre adéquat, la détermination de l'outil adéquat se fait par le choix d'un outil de rayon adapté à la géométrie de la surface en procédant à l'élection de l'outil dont le rayon est le plus grand assurant l'usinage correcte. L'outil optimum est le plus grand rayon d'outil qui permet d'usiner sans interférences, s'il n'existe aucun rayon garantissant cette contrainte, alors le magasin d'outil (emplacement de rangement des outils) ne possède pas un outil capable d'usiner cette surface sans erreurs.

- Cinquième objectif : le gain de temps d'usinage se fera par la recherche du meilleur outil garantissant l'inexistence d'interférence pour chaque plan d'usinage, cette recherche est réalisée par un algorithme de choix du meilleur outil de chaque plan tout en définissant un nombre maximal d'outils et en essayant de réaliser la combinaison la plus optimale pour l'usinage.

IV.3. Modélisation de l'application en UML :

La modélisation en UML commence par l'établissement d'un scénario qui est représenté par le diagramme de cas d'utilisation (voir figure V.21) (Use Case)

IV.3.1. Diagramme de cas d'utilisation.

Le diagramme de cas d'utilisation exprime le comportement attendu du système. Il est composé de plusieurs objets graphiques.

Le système étudié est représenté par la figure V.21 qui définit les cas d'utilisation à réaliser. Par exemple le cas d'utilisation "Appel opération ébauchage" insiste sur l'existence d'un appel à une activité (ou un ensemble d'activités) réalisant l'opération d'ébauchage, cette dernière est enclenchée par l'intervention d'un acteur externe "Utilisateur" qui lui même s'associe avec l'activité du cas d'utilisation "Configuration des paramètres" ce qui crée un état d'attente de configuration, puisque cette activité fait intervenir un acteur du système "Système de calcul" qui se charge du calcul.

L'acteur "Système de calcul" dans son calcul fait intervenir d'autres activités parmi elles la concrétisation de réalisation de l'opération d'ébauchage ainsi qu'un appel à l'activité de génération du programme G-Code. On tire de ce diagramme les premiers objectifs opérationnels.

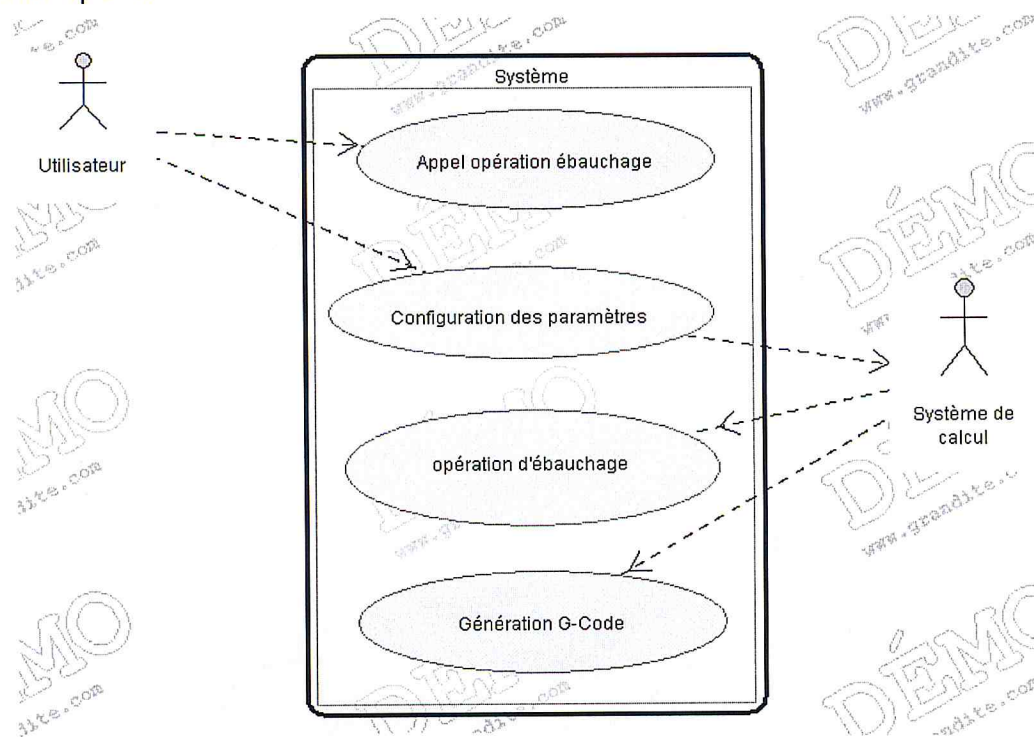


Figure V.21: Diagramme de cas d'utilisation.

IV.3.2. Diagramme de classes :

Le diagramme qui suit (voir Figure V.22) est celui des classes. Il modélise l'ensemble des classes du système ainsi que leurs relations (type et multiplicité), l'ensemble de ces classes par leurs interactions assurent les activités attendues représentées par la (Figure V.21).

La classe principale est la classe TForm800 ce qui explique son volume. Dans cette représentation on a exposé un ensemble réduit de ses attributs et de ses fonctions. Dans la suite de ce paragraphe on fait référence à ces fonctions ainsi que leurs rôles dans le système.

La fonction void TForm800::calcul_contour_decale(int nuplan) :

Cette fonction membre de la classe TForm800 assure l'activité de calcul du contour décalé qui assure l'arrêt du passage d'outil avant la collision avec les contours de la surface à réaliser. Le contour décalé calculé est englobé par un ensemble de points qui sont une projection des points du contour suivant l'orientation de la normale à une distance égale au Rayon (représenté par l'attribut double RayOutil) d'outil+marge de sécurité (représenté par l'attribut double Marge). Les points des contours décalés sont stockés dans le vecteur de points dont le nombre est égal à celui des points du contour tmp_inters_z_constant_contour_decale[indice1].

L'algorithme suivant représente le principe de fonctionnement.

```

debut fonction
  pour tout point contour
    faire
      point_contour_decale_x = point_contour_x + (Marge+RayOutil) *
        suivant_normale_point_x;
      point_contour_decale_y = point_contour_y + (Marge+RayOutil) *
        suivant_normale_point_y;
    fait
  fin

```

La fonction void bool TForm800::calcul_intersection_contour_decale(int nuplan) :

Son rôle est la génération des points de passage de l'outil en assurant l'appartenance de ces points sur soit le brut de la surface ou bien celui du contour décalé. Son principe est le calcul d'intersection des droites d'usinage avec le brut et aussi avec le contour décalé, le résultat de l'opération est stocké dans une matrice cubique cube_droit_inter[indice1][indice2][indice3]

Cette fonction repose sur l'algorithme ci-dessous.

```

debut fonction
  pour toute droite
    faire
      pour tous point_de_contour
        faire
          intersection_droite_brut()
          si(existe_intersection_droite_coutour-decal)
            alors
              recuperation_point_dans_matrice
          fin si
        fait
      fait
    fin
  fin

```

La fonction `bool TForm800::calcul_interference_z_constant(int nuplan)` : la génération des points de passage d'outil peut créer des erreurs d'interférence, cette fonction vérifie s'il existe des interférence avec la surface ou non. Son algorithme est le suivant :

```

debut fonction
  pour tous 2 points du trajet
    faire
      si point1 crée interférence
        alors
          retourner vrai
      fin si
      si point2 crée interférence
        alors
          retourner vrai
      fin si
      si droite point1 a point2 crée interférence
        alors
          retourner vrai
      fin si
    fait
  fin

```

La fonction `void TForm800::generer_programme_G_Code_Ebauche()` :

Le programme G-Code est généré par cette fonction.

```

debut fonction
  pour tous 2 points du trajet
    faire
      si deplacement rapide
        alors
          editer G00
        sinon
          editer G01
      fin si
      si sens rotation broche trigonométrique
        alors
          editer G03
        sinon
          editer G02
      fin si
      editer X point_contour_decale_x
      editer Y point_contour_decale_Y
      editer Z point_contour_decale_Z
      ...
      ...
    fait
  fin

```

La deuxième classe en importance est la TForm804, cette classe s'occupe de la vérification et l'optimisation du trajet d'usinage et son interaction avec la première classe génère le résultat voulu.

Les fonctions de la classe TForm804 font appel à ceux de la classe TForm800. ainsi qu'un ensemble de fonctions définies comme suit :

La fonction void TForm804::Button5Click(TObject *Sender) : calcul le rayon optimal pour chaque plan Z-Constant garantissant l'inexistence d'interférence.

```

debut fonction
  pour chaque plan
    faire
      pour chaque rayon
        faire
          si cree interference
            alors rayon_plus_petit
            sinon outil_trouve
          fin si
        fait
      si outil_non_trouve
        alors afficher message outil non trouve
      fin si
    fait
  fin

```

La fonction void TForm804::Button2Click(TObject *Sender) : calcul la combinaison d'outils optimale pour l'usinage de toute la surface en exploitant les résultats de la précédente fonction et en cherchant le rayon adapté pour chaque plan, son principe de fonctionnement est représenté par l'algorithme suivant :

```

debut fonction
  pour chaque plan
    faire recuperation_outil_utilise
    calcul_frequence_repetition
    fait
  trier_frequence_repetition_outil_ordre_decroissant
  pour chaque plan
    faire pour chaque outil
      faire
        si outil_adapte
          alors recuperer_outil_pour_le_courant_plan
        fin si
      fait
    fait
  fin

```

La classe TForm802 réalise une simulation virtuelle du parcours de l'outil d'usinage lors d'une opération d'usinage réelle, elle possède un ensemble d'outils réalisant cette fonction parmi eux, la variable *visman* est un booléen dont le rôle est d'indiquer le type de simulation, soit manuelle impliquant le dessin en interaction avec l'utilisateur du trajet d'usinage ou automatique le trajet se dessine d'une manière automatiquement en exploitant un *timer* (qui est une fonction du système qui s'active à un intervalle de temps régulier). La variable *Begining* est utilisée seulement dans le cas de simulation automatique marquant ainsi le début et la fin de la simulation. La variable *Temp_Usinage* stock le temps d'usinage de toute la surface qui est calculé par la fonction *Temps_Usinage_Ebauche()* alors que la fonction *Temps_Ebauche_Ecoule(Opr :int)* calcul le temps parcouru d'usinage en exploitant une barre de progression indiquant en parallèle le pourcentage de la progression de l'usinage.

La fonction membre *BitBtn1Click()* active ou désactive le mode de simulation manuel en changeant la donnée de la variable *visman*.

La fonction *M1Click()* lance l'opération de simulation du trajet de l'outil, lors d'une simulation automatique elle est activé par un timer alors qu'en mode manuel elle est activé en interaction avec l'utilisateur.

La fonction *M2Click()* est activé que dans le cas du mode automatique elle a pour fonction de l'arrêt de la simulation qui sera relancée par la fonction *M1Click()*.

L'algorithme de simulation du trajet d'usinage est comme suit :

```

debut Simulation
  calcul_temps_usinage_surface()
  si visualisation automatique
    alors
      si begining vrai
        alors activer_Timer()
      fin si
    sinon si appel dessiner_trajet
      alors dessiner_trajet_outil
    fin si
  fin si

  tant que Timer activf
    faire
      dessiner_trajet_outil()
      calcul_temps_usinage_ecoule()
    fait
  fin

```

La classe TForm803 fait l'interface entre l'application et la base de données des outils, son rôle primordial est la création/destruction de données des outils pour la mise à jour des données outils ainsi que l'accès à ces données

La classe TForm801 est une fenêtre contenant des listes servant à stocker l'ensemble des points du trajet d'usinage ainsi que leurs paramètres, parmi les données stockées, les coordonnées X, Y, Z, le rayon d'outil adéquat ainsi que sa longueur, la vitesse d'avance à ce point ainsi que la vitesse de rotation de la broche et le sens de rotation (trigonométrique/ anti-trigonométrique).

La classe TForm1 englobe tous les outils de dessin. Les données collecter par l'ensemble des objets défini précédemment est visualisés par l'appel à la fonction membre DrawObjet().

Le diagramme qui suit voir (Figure V.22) est celui du diagramme de classes. Il représente les classes de notre système ainsi que leurs interactions.

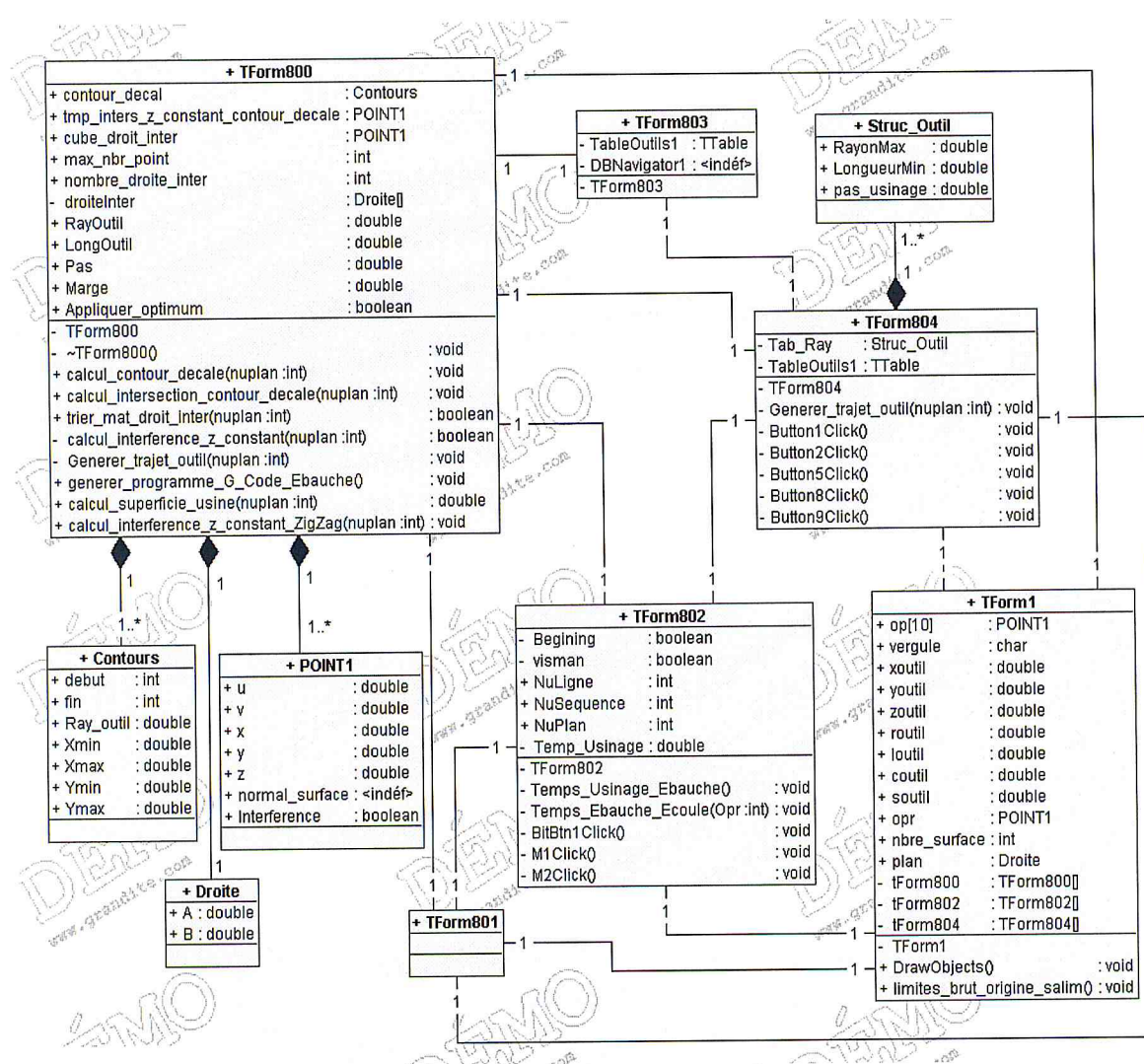


Figure V.22: Diagramme de classes.

IV.3.3. Diagramme de collaboration :

Ce diagramme modélise la collaboration des objets des classes définies précédemment, ainsi que les échanges des données sont schématisés et bien exposés en définissant l'ordre de déroulement des activités.

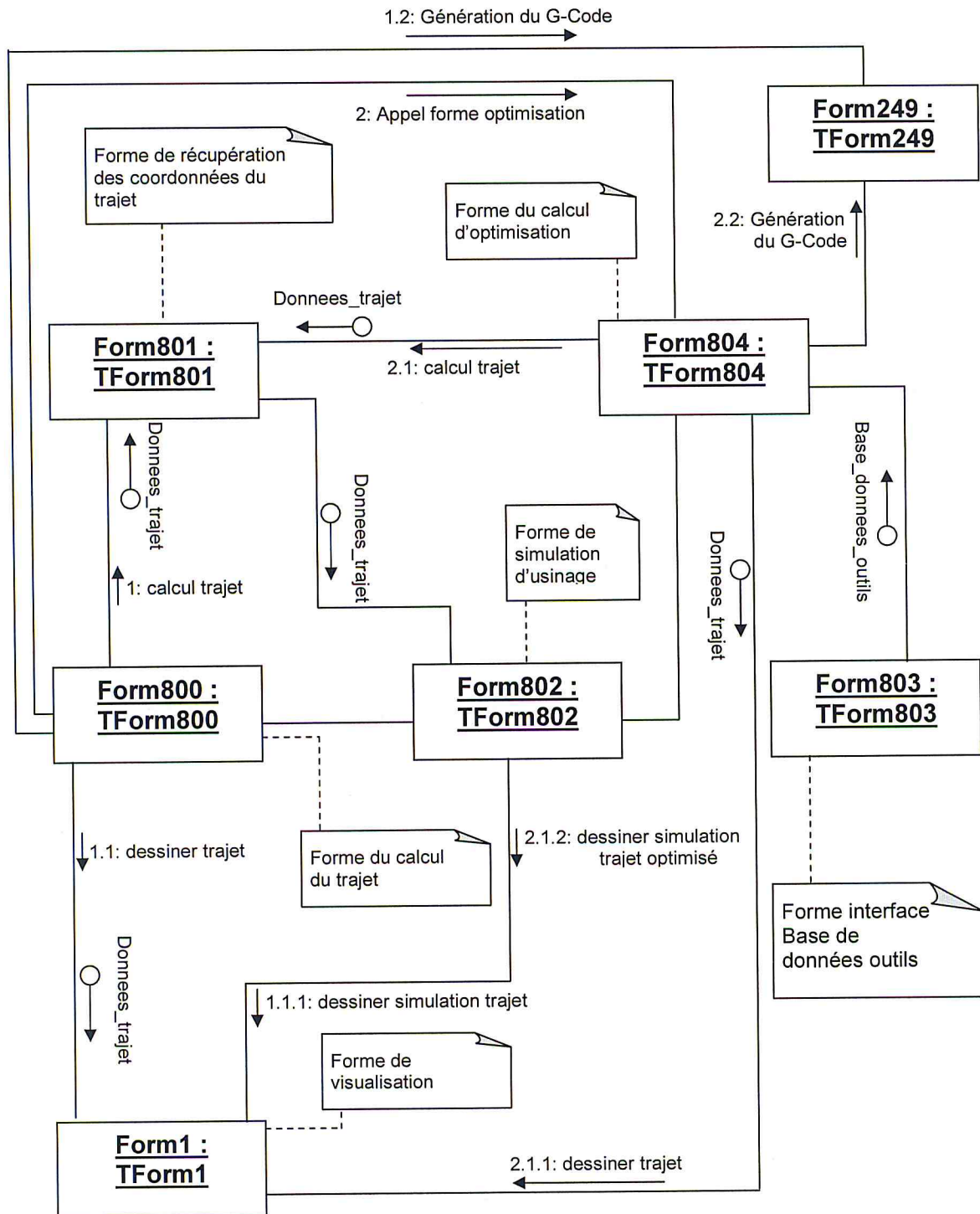


Figure V.23: Diagramme de collaboration.

Le diagramme de la (Figure V.6) fait une description sur le rôle de chaque objet de la classe. En étudiant ce diagramme, on remarque que tout commence par l'activation de l'objet Form800 de la classe TForm800 qui parcourt deux chemins principaux:

- Le premier parcours: lancement de l'activité **1:calcul du trajet** mettant en relation l'objet Form801 qui sert à enregistrer les données générées, puis il y aura l'activité de **1.1: dessiner trajet** par l'objet Form1, l'objet Form249 est activé lors de l'activité de **1.2: Génération du G-Code**, l'activité **1.3: simulation** appelle l'objet Form802 qui englobe les outils permettant la simulation.

- Le deuxième parcours: l'objet Form800 passe le contrôle à Form804 par l'activité du **2: Appel forme optimisation** qui parcourt la même suite d'activités à une exception près est que le calcul se fera avec l'optimisation, **2:calcul du trajet, 2.1: dessiner trajet, 2.2: Génération du G-Code, 2.3: simulation**.

L'objet Form803 sert d'interface avec la base de données des outils

IV.3.4. Diagramme d'activité :

Le diagramme de la (Figure V.7) schématise l'ensemble des activités réalisées par le système dès son exécution. L'activité de **Création des réseaux de contrôle** puis celle de **Construction des courbes** et enfin l'activité de **Construction des surfaces**, toutes ces activités entre dans le domaine de la CAO partie intégrante de la CFAO, ainsi ces activités servent à créer des surfaces en 3D, qui seront usinées par des outils mathématiques de construction des surfaces gauches (B-Splines et NURBS).

L'activité **Calcul des contours** calcul l'intersection entre un plan horizontal (Z constant) et la triangulation approximant le mieux la surface conçue. Par cette activité on récupère un ensemble de points de la surface sur un même plan (même coordonnée sur l'axe des Z), puis l'activité de **Calcul des contours décalés** s'active et en utilisant l'ensemble des points précédents et en suivant la normale calcule un autre ensemble de points possédants les mêmes propriétés que leurs semblables (points du contour) mais à une distance (coordonnées X, Y modifiées) calculée par rapport à la normale des points du contour.

L'activité du **Calcul des intersections entre droites parallèles et contours décalés** comme son nom l'indique fait le calcul des intersections des droites d'un même plan avec les droites des points du contours décalés de ce même plan. L'ensemble des points d'intersection sont ainsi récoltés et sont utilisés dans le calcul du trajet de l'outil. Par la suite, un calcul est lancé afin de déterminer les passages autorisés et les passages non autorisés de l'outil.

L'activité de **Calcul interférence** vérifie l'inexistence des interférences et l'absence de points de collision du trajet défini par l'activité précédente.

Le système par sa fonction, après la réalisation de toutes les activités précédentes entre dans un choix, un chemin (1) menant à l'activité de **simulation** de passage de l'outil lors de l'usinage soit à celle de **Génération du G-Code** qui aboutit au programme G-Code de l'usinage de la surface. Le chemin (2) passant par l'activité d'**optimisation** qui réalise le choix de l'outil optimum pour chaque plan de coupe (Z-Constant) puis le passage aux deux activités **Simulation** et **Génération du G-Code**.

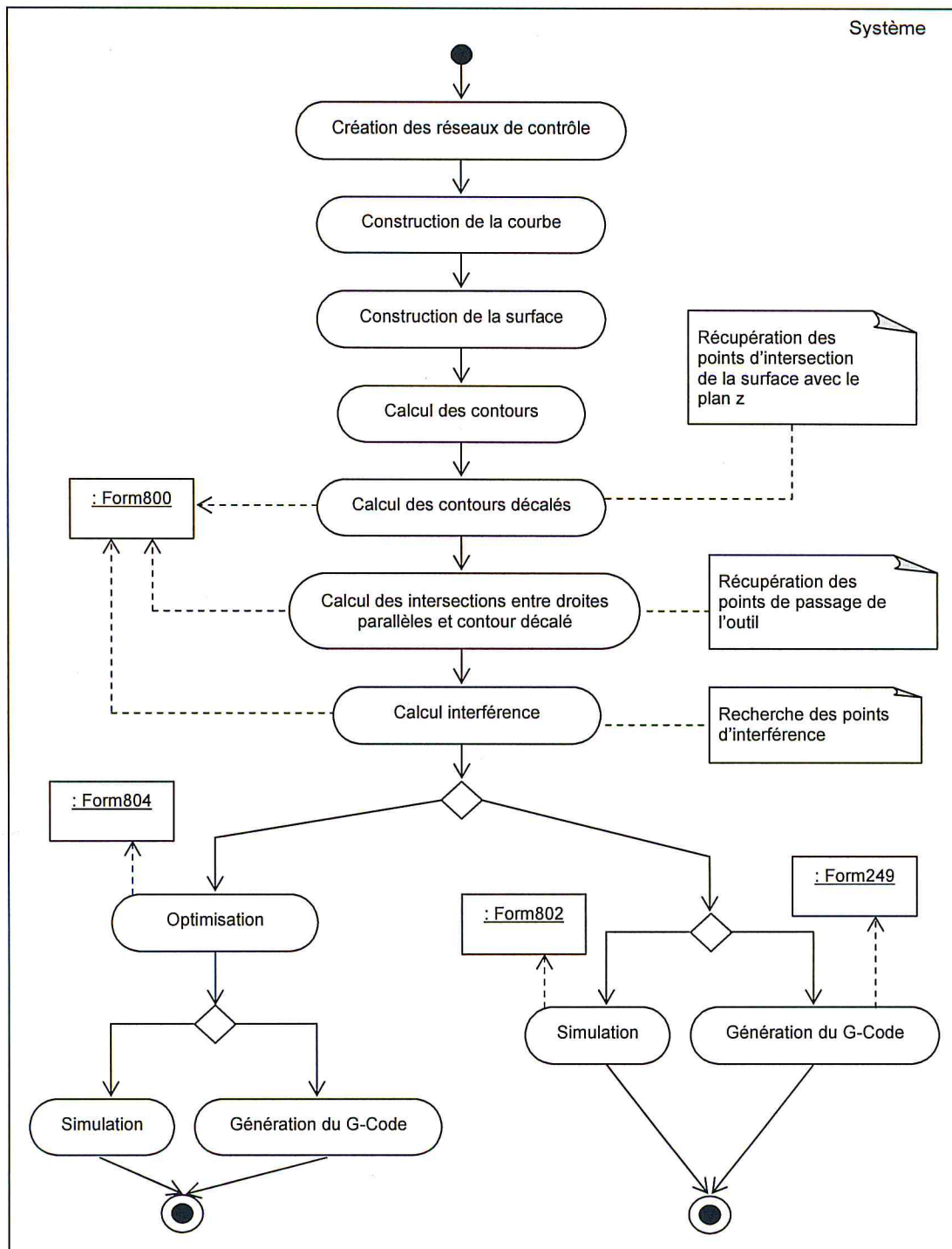


Figure V.24: Diagramme de d'activité.

V. CONCLUSION :

Dans ce chapitre nous avons fait une synthèse sur la réalisation de l'application, les étapes menant aux résultats souhaités ainsi que les différents algorithmes utilisés par les fonctions du système.

Chapitre VI — *Présentation de
L'application.*

I. INTRODUCTION :

Notre application entre dans le domaine de la CFAO (Conception et Fabrication Assistées par Ordinateur), elle réunit deux domaines la CAO (Conception Assistée par Ordinateur) et la FAO (Fabrication Assistée par Ordinateur).

Le premier domaine CAO s'occupe de la conception; la réalisation de modèles géométrique à l'aide d'un ordinateur en exploitant les courbes et les surfaces B-Spline & NURBS, il touche tout ce qui est modélisation, création de courbes, de surfaces et tout autre modèle géométrique.

Dans cette étape, nous réalisons notre pièce, en définissant son aspect, sa perspective (sa vue 3D), ses propriétés et aussi la création du polygone de contrôle, de nombre de points de contrôle et de leurs propriétés ainsi que les propriétés de la surface par la définition de la normale, les tangentes, les rayons de courbures et plusieurs autres propriétés.

Le domaine de la FAO s'attache au côté fabrication, il sert de pont entre la conception de la pièce et son usinage, son rôle primordiale est la traduction du modèle conceptuel en une suite de commande numérique d'usinage afin de réaliser le modèle conçu par l'étape précédente.

Le résultat de cette étape est un fichier contenant du code appelé G-Code compréhensible par les MOCN qui est constitué d'un ensemble fini de coordonnées de points avec des paramètres complémentaires tels que la vitesse d'avance, la vitesse de rotation de la broche, ...etc. Toutes ses données définissent les commandes exécutées par les machines pour réaliser l'opération d'usinage.

Dans notre travail on s'est occupé du côté FAO du projet et plus précisément la phase d'ébauchage (dégrossissage) qui constitue l'étape primaire dans la réalisation de toute pièce en usinage mécanique.

Nous allons voir dans ce qui suit l'architecture de l'interface réalisée par notre étude.

II. INTERFACE DE CONCEPTION :

II.1. Ecran d'accueil:

L'interface d'accueil est constituée de deux fenêtres :

- Fenêtre d'outils de conception : sert à la conception des courbes et des surfaces ainsi que la manipulation des paramètres des surfaces, polygone de contrôle et tout autre manipulation. Cette fenêtre est l'interface CAO du projet.
- Fenêtre de dessin : cette fenêtre est l'espace de visualisation du résultat (visualisation du modèle en 3D), elle exploite toute la puissance de la bibliothèque graphique 3D « OpenGL » dont elle se sert (voir Figure VI.1).

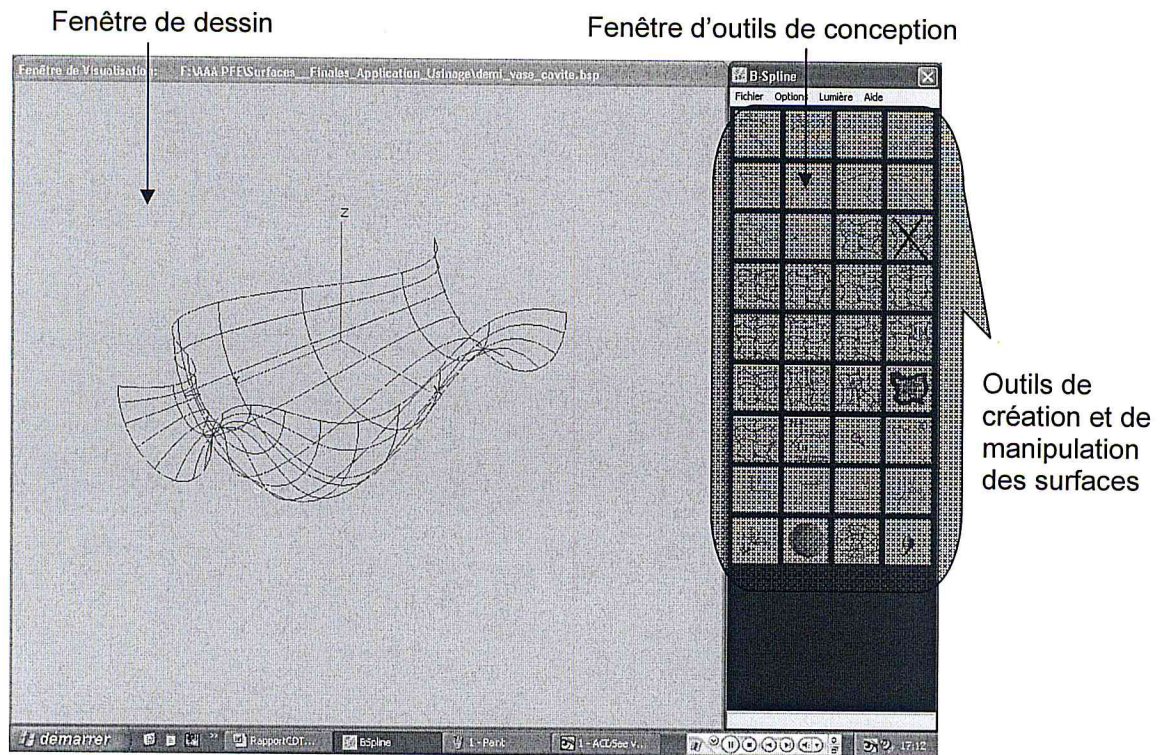
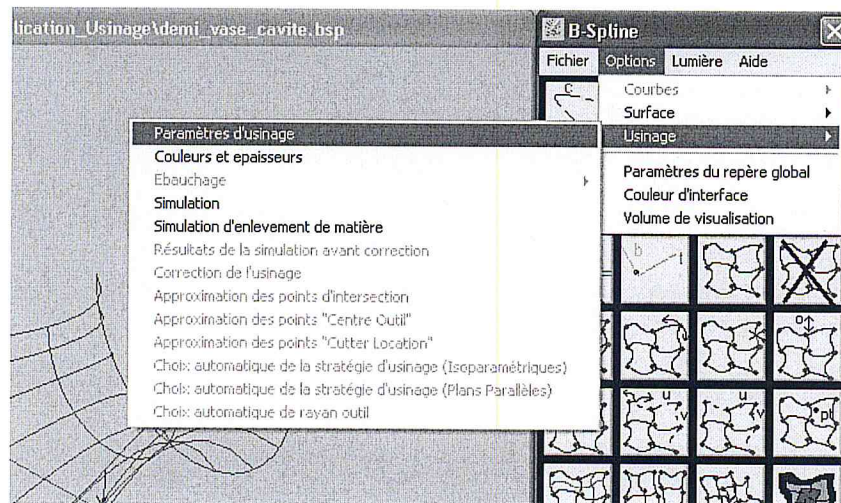


Figure VI.1: Interface de l'écran d'accueil

II.2. Interface d'usinage :

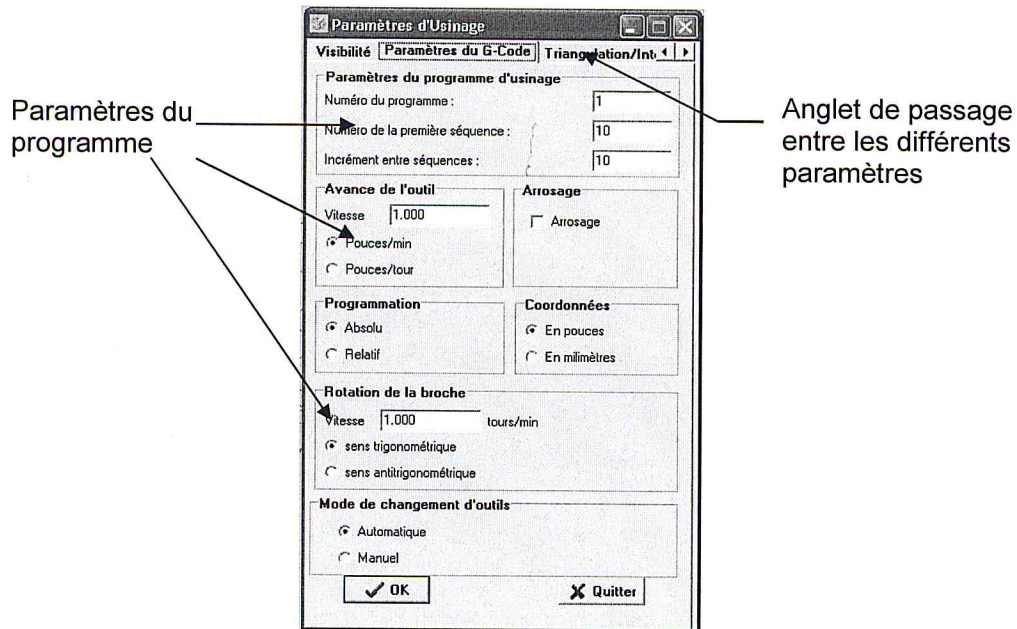
Le passage à l'étape d'usinage se fait par la sélection du menu principal *Option*, puis le sous-menu *Usinage* et enfin *Paramètres d'usinage* (voir FigureVI.2).



FigureVI.2: Interface de passage à l'étape d'usinage.

II.3. Paramètres d'usinage :

Cette fenêtre englobe quelques paramètres d'usinage comme la distance entre deux plans de coupe et des paramètres du G-Code comme la vitesse de rotation de la broche et d'autres paramètres (voir FigureVI.3).



FigureVI.3: Fenêtre des Paramètres d'usinage.

Après la configuration des paramètres d'usinage et le calcul des données à exploiter par l'ébauchage (intersection des plans Z-Constant avec la surface et calcul des contours pour chaque plan). La fenêtre de la (Figure IV.4) fait le lien avec les fenêtres de calcul de l'opération d'ébauchage.

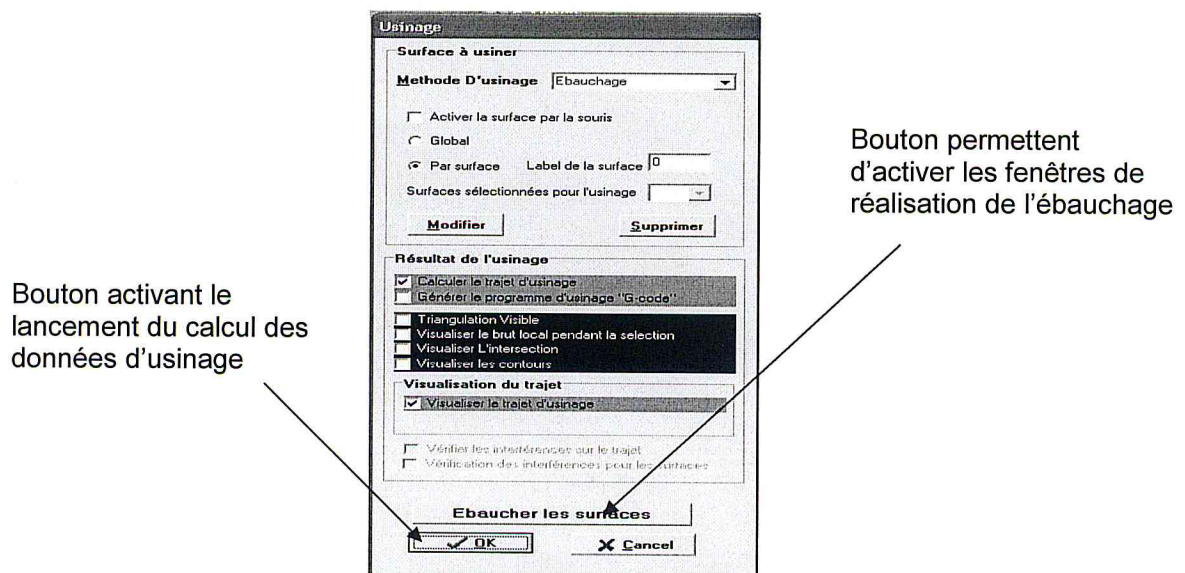


Figure IV.4: Fenêtre du passage à l'opération d'ébauchage.

III. EXPOSITION DU TRAVAIL REALISER LORS DE CE PROJET :

III.1. La fenêtre d'accueil d'ébauchage:

Notre première fenêtre sert à collecter les informations nécessaires pour la réalisation de l'opération d'ébauchage. Cette fenêtre est constituée d'un ensemble de page à anquet (voir figure IV.5).

La fenêtre de gauche (direction d'usinage) permet de choisir l'angle d'inclinaison de la droite d'usinage, tandis que celle de droite (paramètres d'usinage) sert à définir la méthode d'usinage (*One-Way* ou *Zig-Zag*) ainsi que le pas d'usinage, la surépaisseur d'usinage, les distance d'*engagement* et de *dégagement* et aussi une distance de référence.

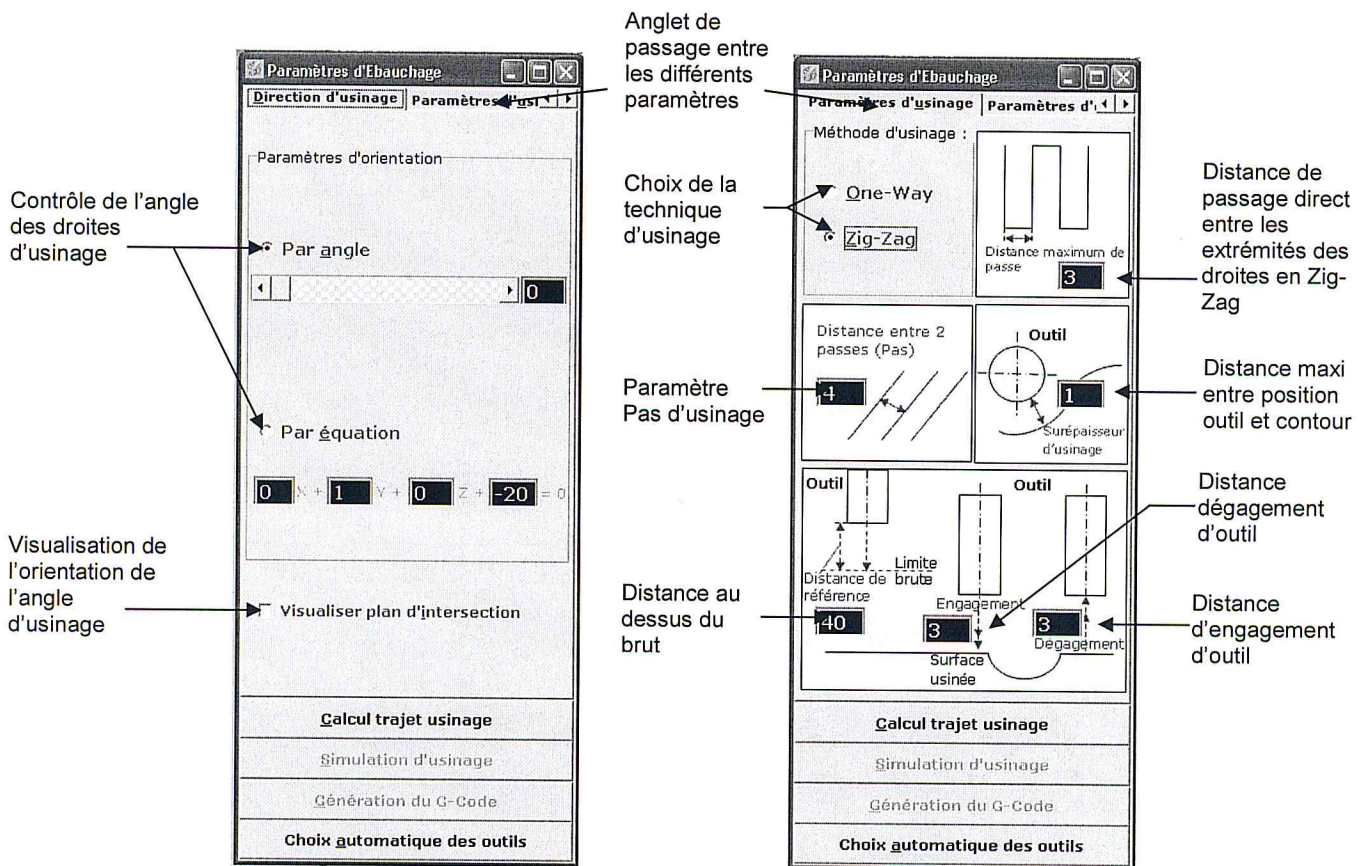


Figure VI.5 : Fenêtre de paramétrage de l'opération d'ébauchage.

La (Figure VI.6) présente l'onglet *paramètre d'outil* qui détermine la vitesse d'avance de l'outil ainsi que la vitesse de rotation de la broche et son sens, le rayon d'outil avec sa longueur peuvent être définis à ce niveau par l'utilisateur.

L'onglet *visualisation* de la même figure permet d'afficher ou de cacher par plan les contours et les contours décalés, les droites d'usinage avec le trajet pris par l'outil ainsi que les positions possibles de l'outil et enfin les points qui entrent en collision avec la surface (interférence) s'ils existent. Le passage d'un plan à un autre se fait par un clic sur l'un des deux boutons (+), (-) avec l'affichage du plan courant.

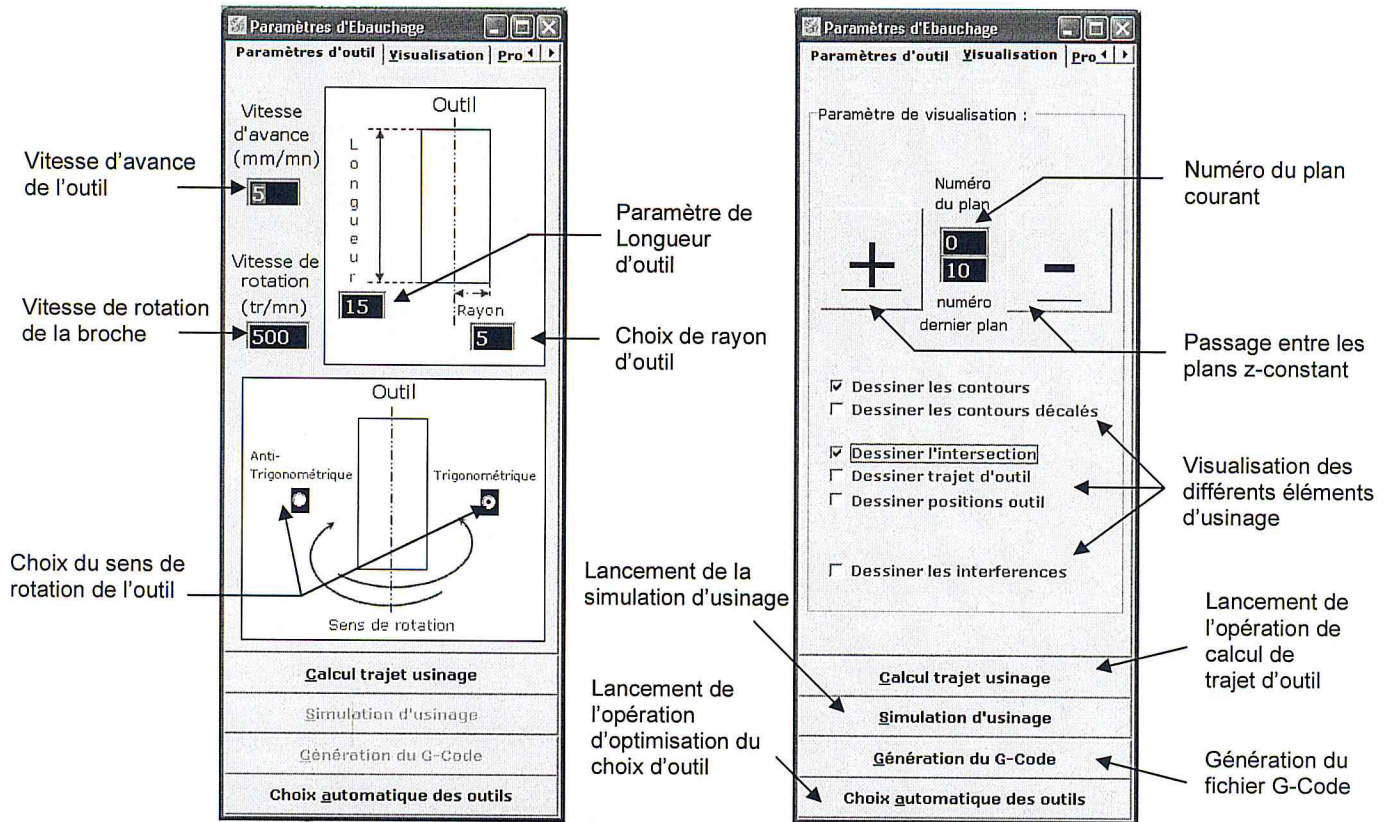


Figure IV.6 : Fenêtre de paramétrage de l'opération d'ébauchage (Suite).

III.2. Lancement des opérations de calculs :

Le bouton **Calcul trajet usinage** active les fonctions de calcul du trajet d'usinage en combinant avec les paramètres définis par l'utilisateur. Cette fonction a pour résultat une succession de points de coordonnées «X, Y, Z» avec une vitesse d'avance «AVANCE», ainsi que la vitesse de rotation de l'outil, et plusieurs autres paramètres.

Le bouton **Simulation** avec le trajet généré précédemment vont réaliser une simulation de comportement de l'outil pendant l'usinage, qui se fait en cliquant sur ce dernier. En plus de la simulation, ce bouton lance l'opération de calcul du temps d'usinage ainsi que les coordonnées «X, Y, Z» de chaque point occupé par l'outil en fonction du repère programmation. (voir FigureVI.7).

Le bouton **Génération du G-Code** appelle une fonction dont la responsabilité est de générer le fichier G-Code en fonction du trajet généré par les fonctions précédentes (voir Figure VI.8).

Le bouton **Choix automatique des outils** affiche une fenêtre (voir Figure VI.9) dont le but est de chercher le rayon d'outil optimal pour chaque plan Z-Constant qui nous évite les interférences lors de l'usinage.

III.2.1. Fenêtre de simulation :

La fenêtre ci-dessous englobe la simulation, elle permet de visualiser le passage de l'outil entre les différents points d'usinage avec le suivi de la progression du travail accompli ainsi que plusieurs autres paramètres.

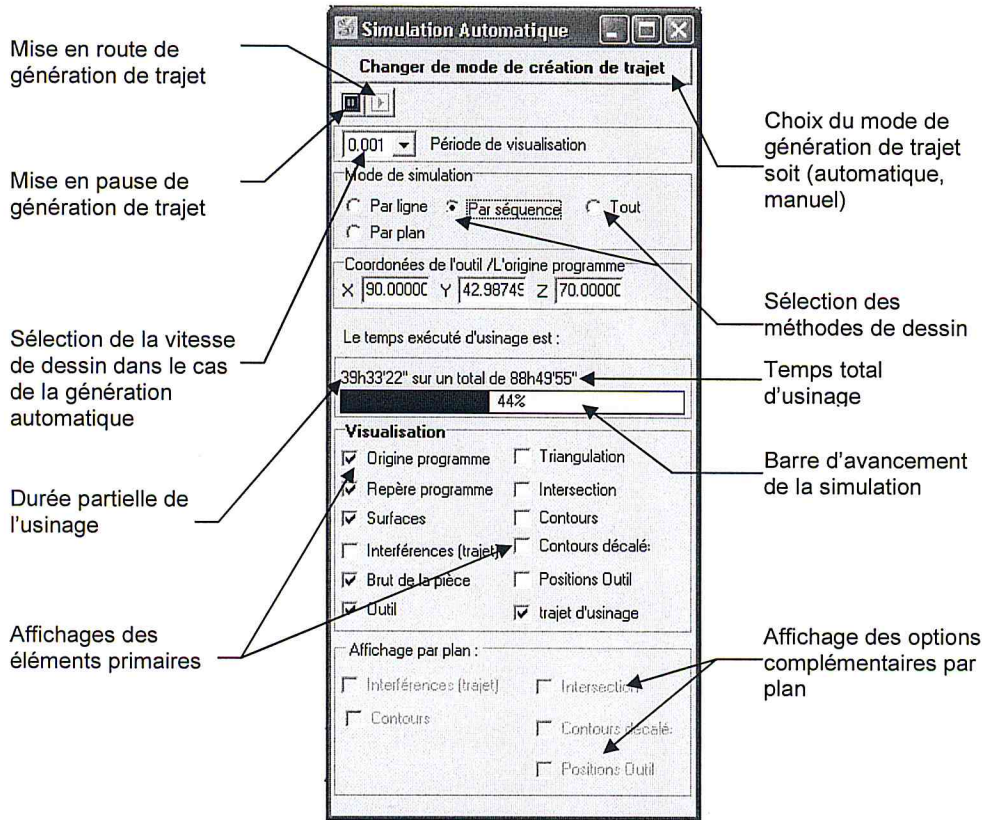


Figure VI.7 : Fenêtre de simulation de l'opération d'usinage

III.2.2. Fenêtre programme G-Code :

Cette fenêtre affiche le programme G-Code généré.

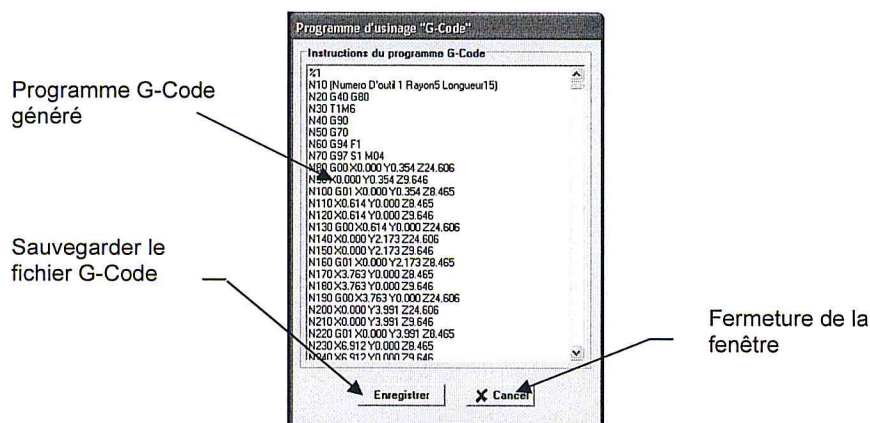


Figure VI.8: Fenêtre de génération du programme G-Code.

III.2.3. Fenêtre optimisation des outils :

Cette fenêtre englobe tous ce qui peut optimiser le trajet d'usinage. Elle possède un ensemble de fonctions calculant le rayon optimal choisi pour chaque plan en fixant un nombre limité d'outils (voir Figure IV.9).

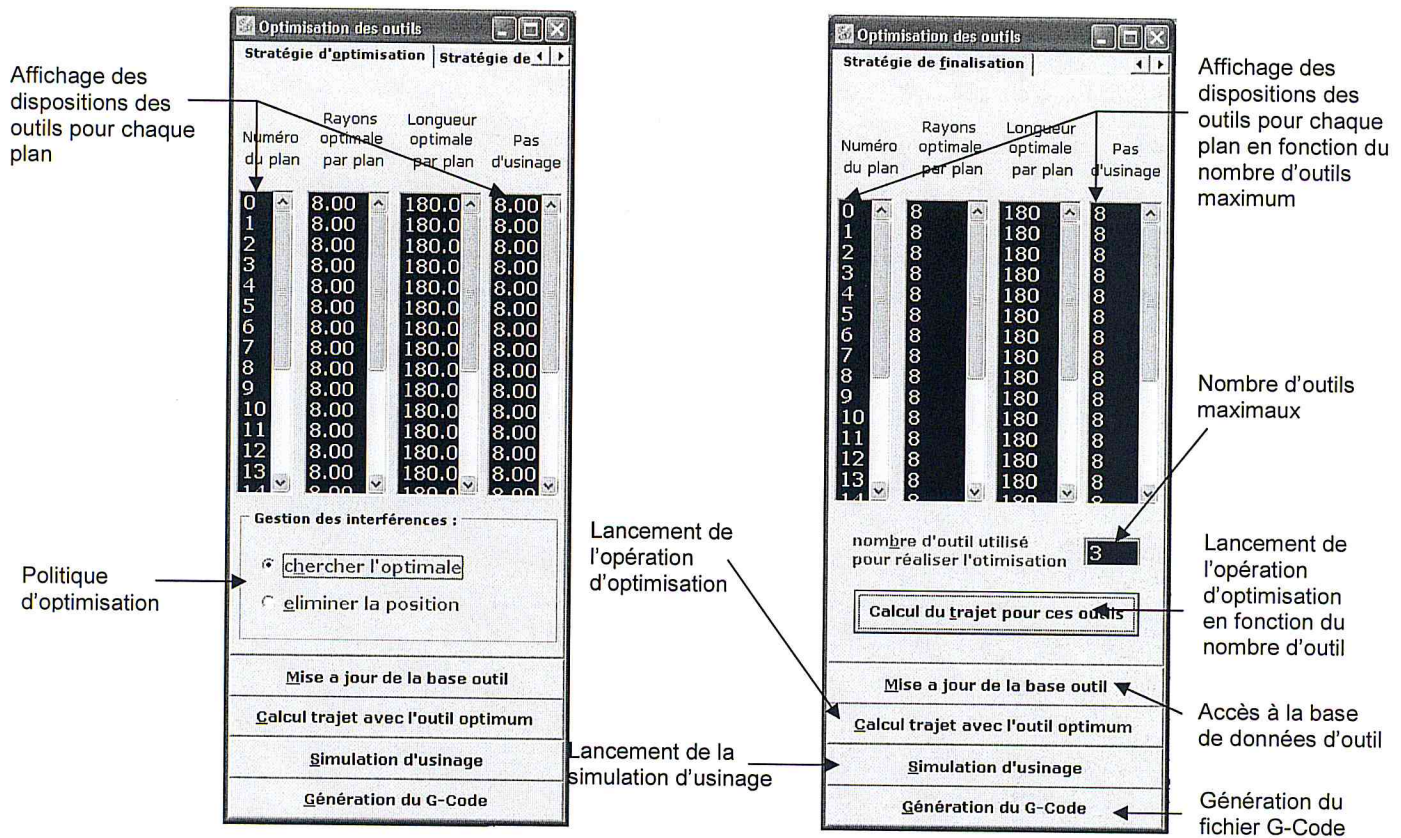


Figure VI.9 : Fenêtre de optimisation des outils.

III.2.3.1. En cas d'échec :

L'échec se réalise lorsque le calcul du meilleur outil garantissant l'absence d'interférence (Cas 1) échoue ou la longueur d'outil engendre la collision de la broche avec la surface usinée (Cas 2).(voir Figure IV.10).

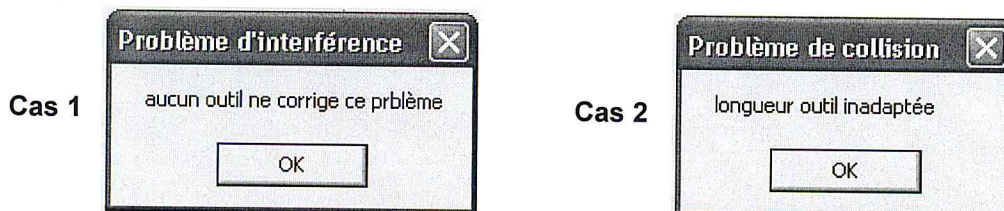


Figure VI.10 : Message en cas d'échec du calcul d'optimisation.

IV. Exemple d'application :

Nous allons voir dans ce qui suit le comportement de l'application avant la génération du fichier G-Code (voir Figure IV.11 et Figure IV.12).

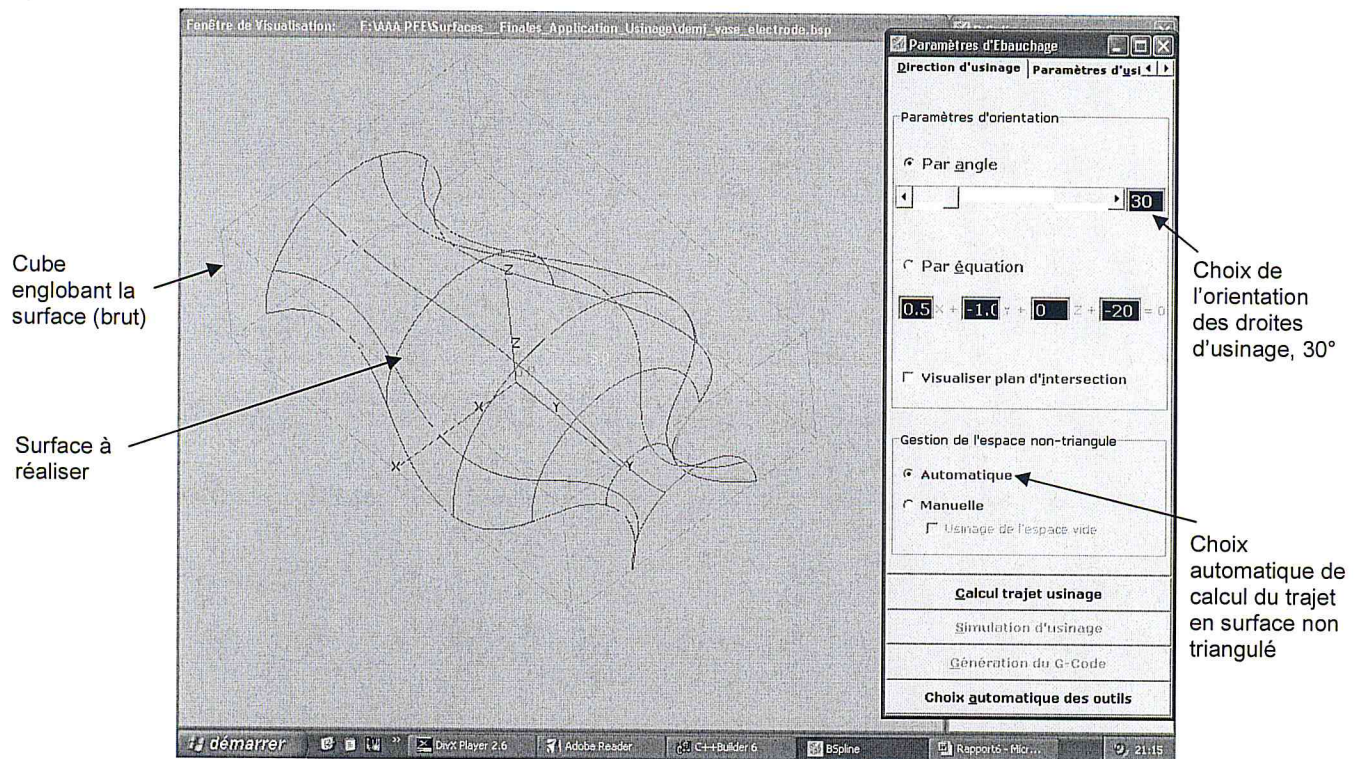


Figure VI.11 : fenêtre de réalisation de la fonction débauchage (à gauche).

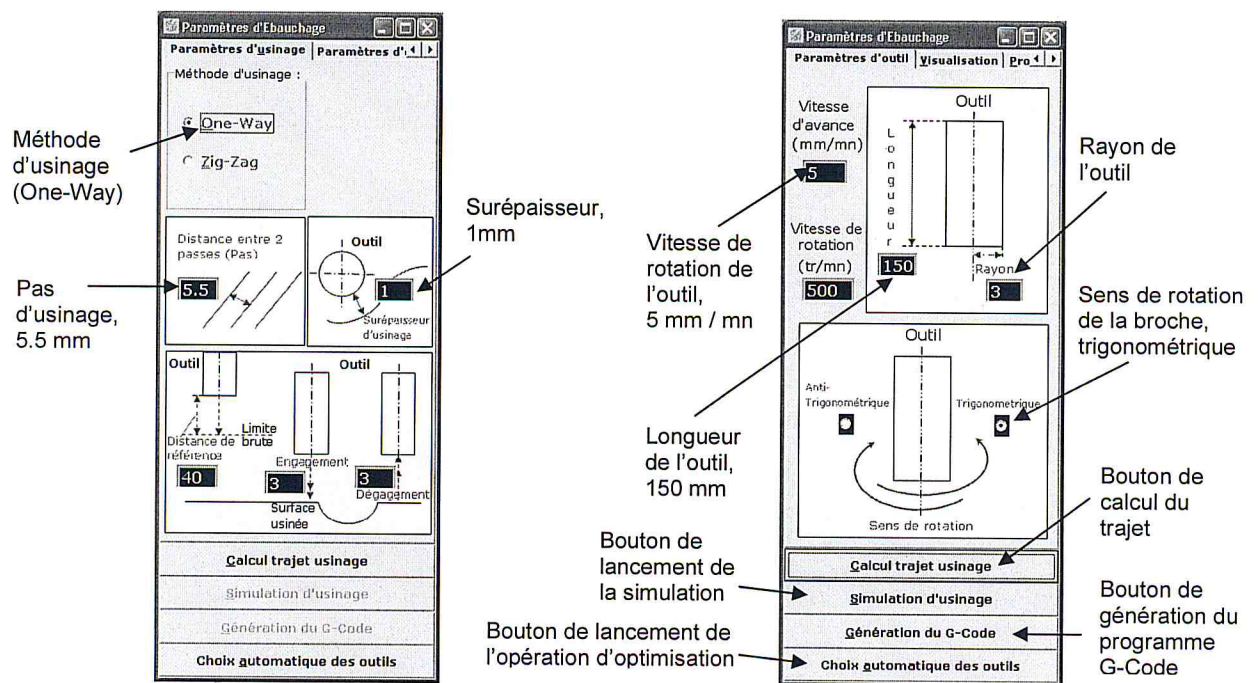


Figure VI.12 : Configuration des paramètres d'ébauchage.

Les deux figures ci-dessous représentent la première fenêtre de réalisation de la fonction d'ébauche, la (Figure VI.11) représente la vue globale de l'application pendant la réalisation de l'ébauchage, et la petite fenêtre à gauche est formée par plusieurs affiches qui sont détaillées dans la (Figure VI.12).

Le lancement de l'opération du calcul de trajet est enclenché par le clic sur le bouton « *Calcul trajet usinage* » qui génère des données et des paramètres stockés dans des structures de données dynamiques. L'affichage des informations récoltées par l'opération précédente est assuré par les « boutons à cocher » (*CheckBox*) comme c'est présenté dans la (Figure VI.13).

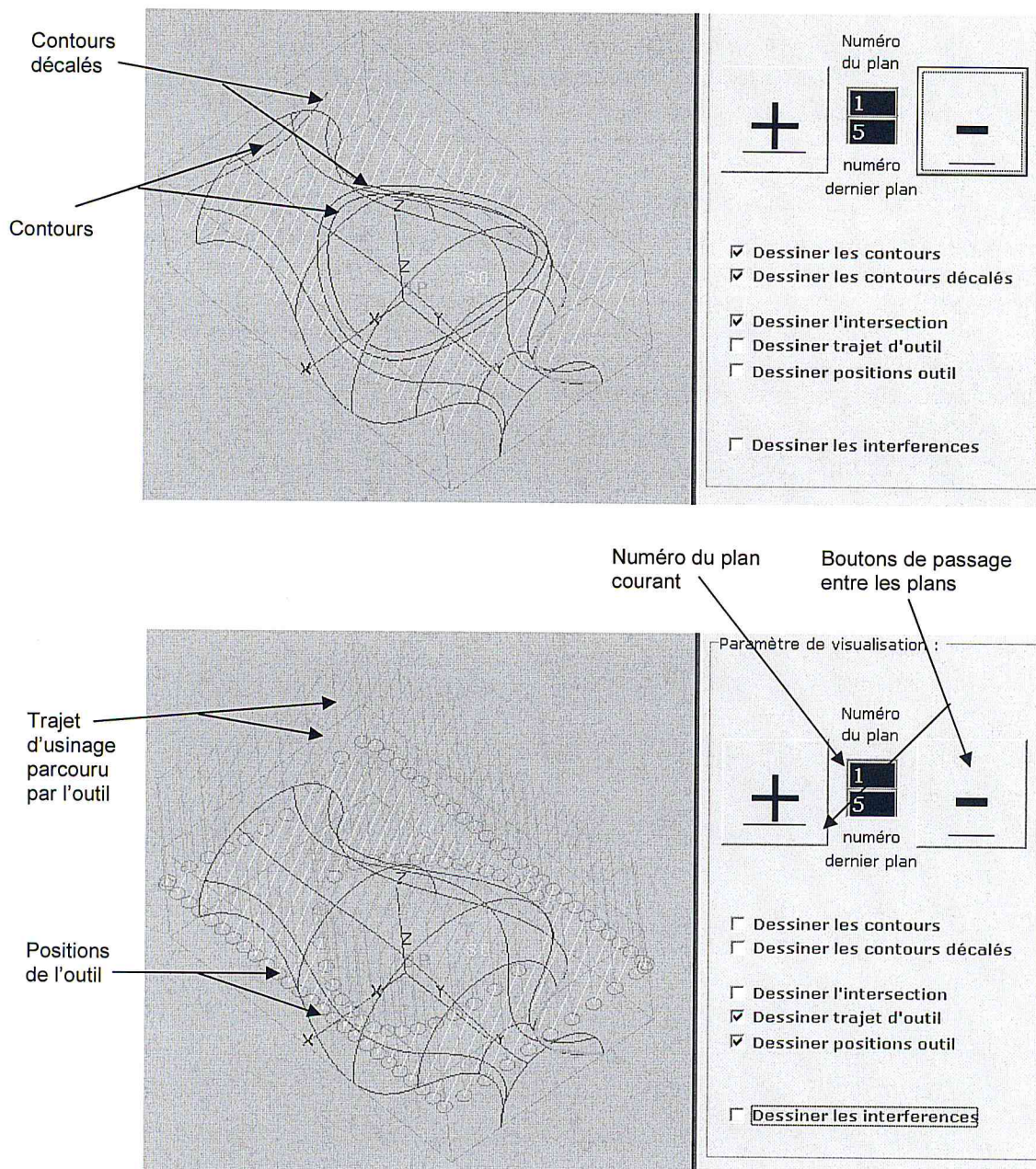


Figure VI.13 : Affichages des données.

Le bouton «*Simulation d'usinage*» de la (Figure VI.12) lance la simulation (voir Figure VI.14), la fenêtre qui est à droite possède les outils d'aide à la simulation.

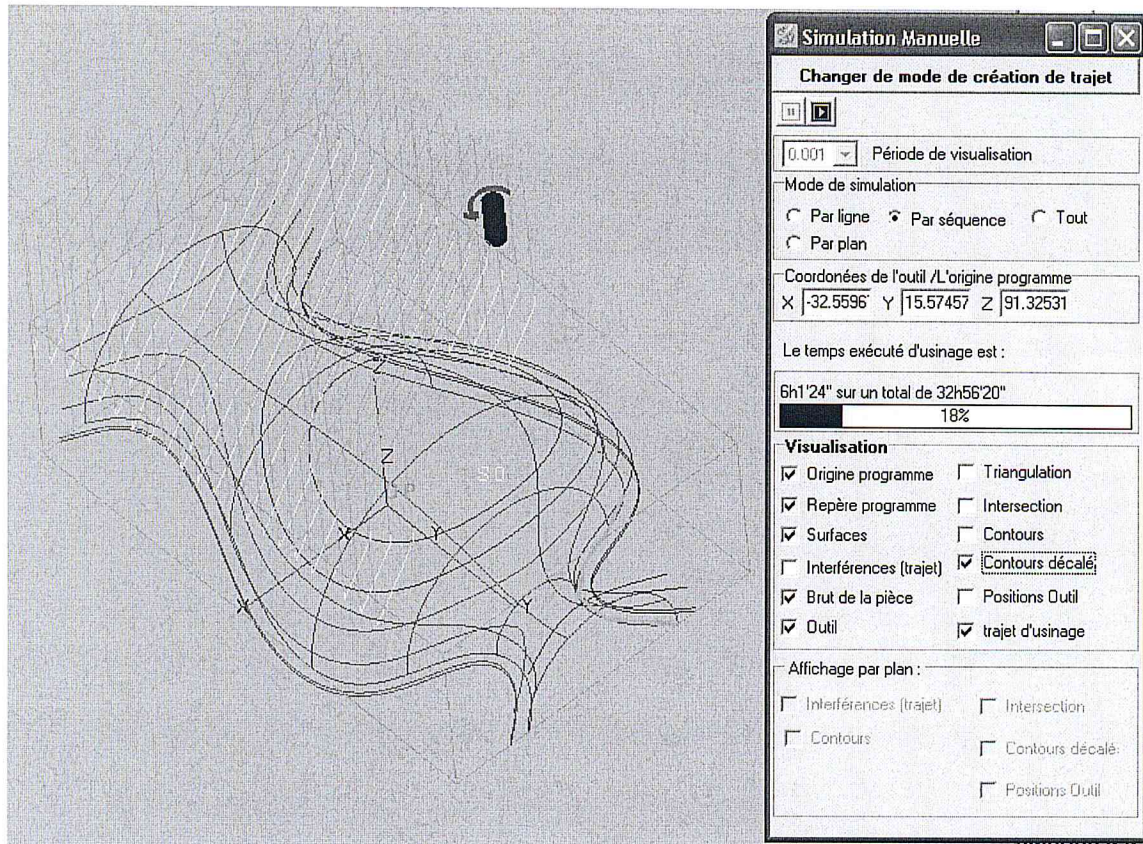


Figure VI.14 : Affichages de la simulation.

Le bouton «*Génération du G-Code*» affiche la fenêtre de la (Figure VI.15). Avec le bouton *Enregistrer*, on peut l'enregistrer dans un fichier séparé.

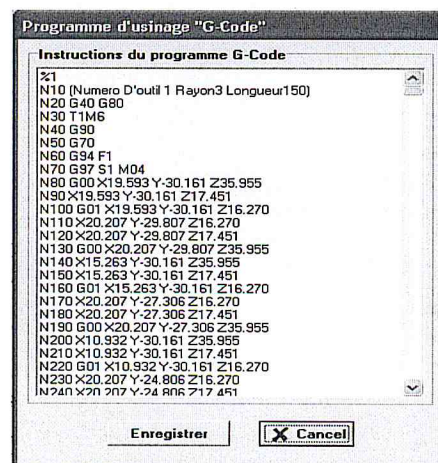


Figure VI.15 : Génération du programme G-Code.

La fenêtre ci-dessous (voir (Figure VI.16)) réalise le calcul du rayon d'outil optimal pour chaque plan. Elle est appelée par le bouton « *Choix automatique des outils* » de la (Figure VI.12).

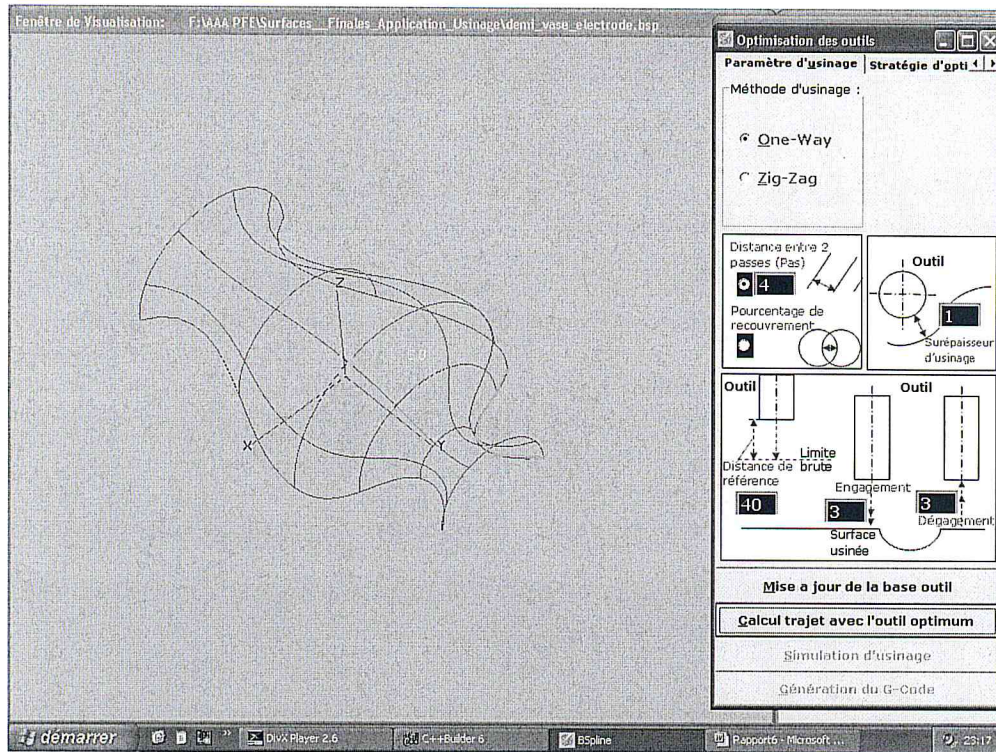


Figure VI.16 : Fenêtre choix automatique des outils.

Le reste des affiches est exposé dans la (Figure VI.17).

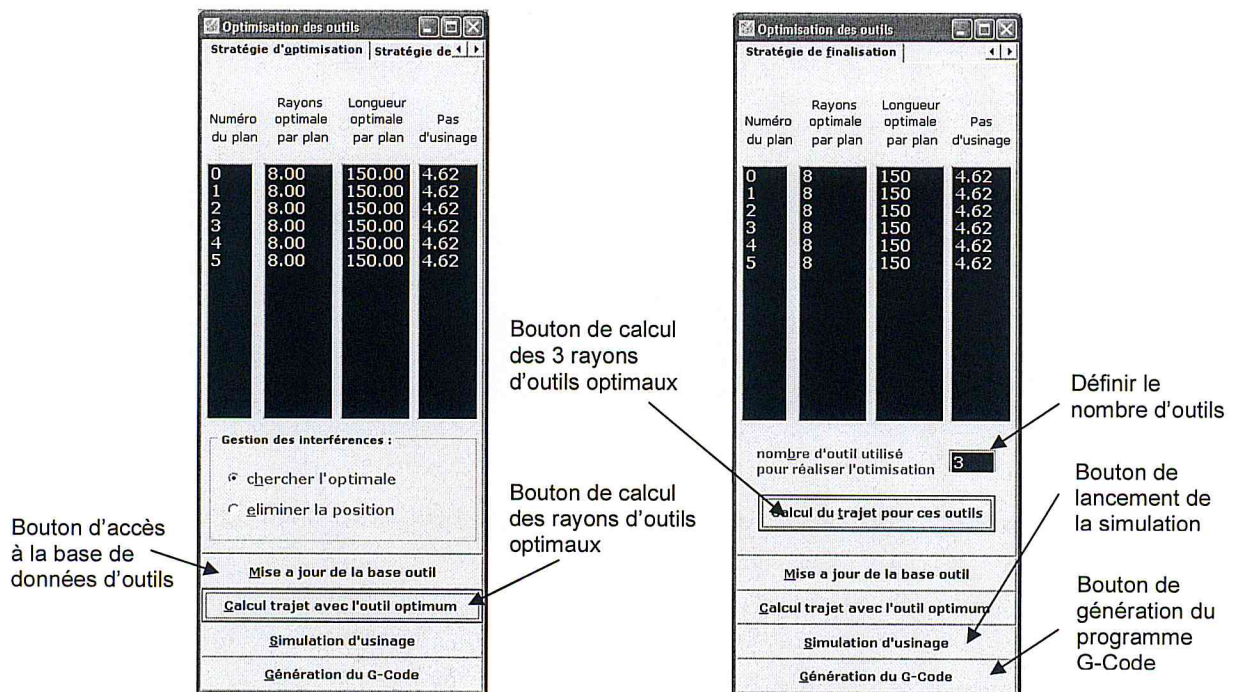
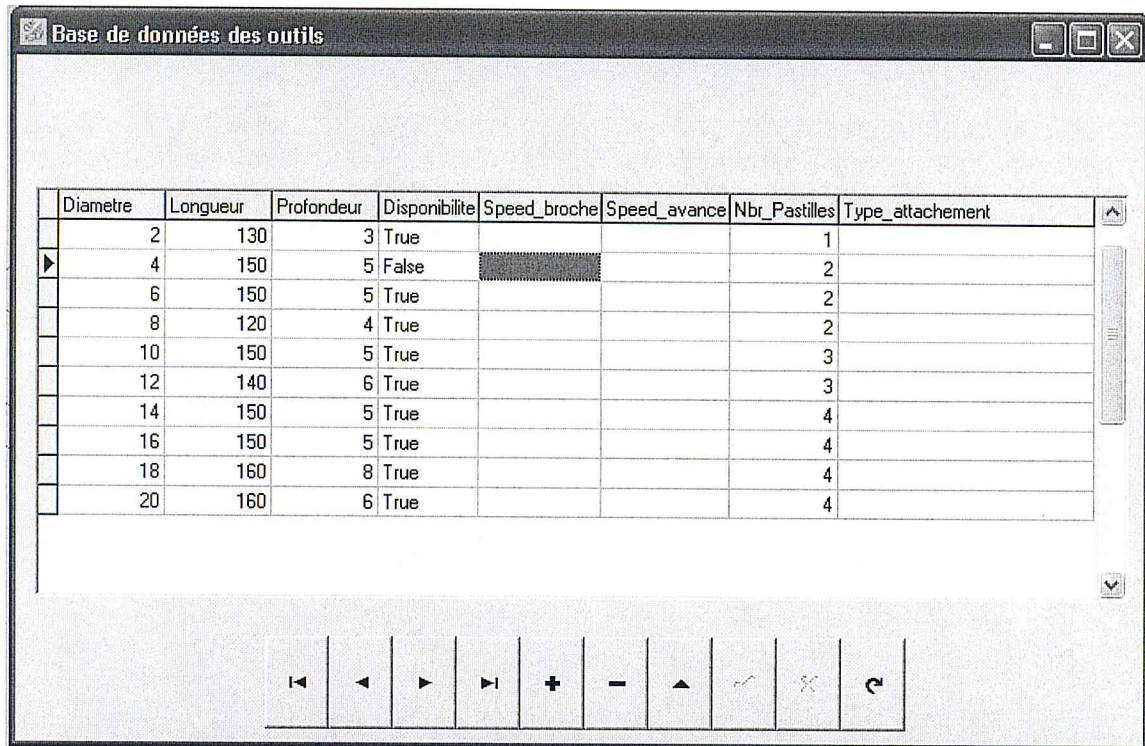


Figure VI.17 : Affichage des rayons optimaux pour chaque plan.

La base de données d'outils est interfacé au moyen de la fenêtre de la (Figure VI.18). On peut ajouter ou modifier ou supprimer un outil au moyen de cette fenêtre. Cette dernière est appelée par le bouton «*Mise à jour de la base outil*» (voir Figure VI. 17).



The screenshot shows a window titled "Base de données des outils" with a table of tool parameters. The table has columns for Diameter, Length, Depth, Availability, Drill Speed, Feed Speed, Number of Inserts, and Attachment Type. Below the table is a control panel with buttons for navigation and editing.

| | Diametre | Longueur | Profondeur | Disponibilite | Speed_broche | Speed_avance | Nbr_Pastilles | Type_attachement |
|---|----------|----------|------------|---------------|--------------|--------------|---------------|------------------|
| | 2 | 130 | 3 | True | | | 1 | |
| ▶ | 4 | 150 | 5 | False | | | 2 | |
| | 6 | 150 | 5 | True | | | 2 | |
| | 8 | 120 | 4 | True | | | 2 | |
| | 10 | 150 | 5 | True | | | 3 | |
| | 12 | 140 | 6 | True | | | 3 | |
| | 14 | 150 | 5 | True | | | 4 | |
| | 16 | 150 | 5 | True | | | 4 | |
| | 18 | 160 | 8 | True | | | 4 | |
| | 20 | 160 | 6 | True | | | 4 | |

Figure VI.18 : interface base de données d'outils.

Le bouton « *Simulation d'usinage* » appel la fenêtre de simulation (voir Figure VI.14) qui ressemble à la première sauf que les paramètres diffèrent, rayon d'outil et paramètre d'usinage ainsi que le trajet d'usinage sont différents.

Le bouton « *Génération du G-Code* » appel la fenêtre de génération du G-Code (voir Figure VI.15).

V. CONCLUSION :

L'application est réalisée d'une façon qu'elle soit simple et efficace dans son utilisation. Les interfaces suivent toutes une même harmonie dans le but de rendre la manipulation de cette application logicielle simple, et peut être utilisé par des personnes autres que les informaticiens, puisque toutes les interfaces sont interactives, et dotées de schémas explicatifs et illustrées par des commentaires.

CONCLUSION GENERALE

Le travail que nous avons présenté consiste en une automatisation de l'opération d'ébauchage des surfaces gauches sur des fraiseuses à commande numérique à 03 axes.

Dans ce mémoire nous avons étudié en premier lieu les méthodes de conception et de représentation des surfaces gauches en appuyant notre étude sur les surfaces B-Spline et NURBS. Ensuite, nous avons présenté les machines utilisées dans l'usinage de ces surfaces et la syntaxe des programmes G-Code. Dans les parties suivantes nous avons présenté en détail l'ébauchage de surfaces gauches ainsi que l'application logicielle développée.

Le résultat de notre application est l'enrichissement de l'application logicielle graphique avec des fonctions qui permettent :

- ✓ D'automatiser la génération des fichiers G-Code.
- ✓ D'ébaucher les surfaces gauches en One-Way ou en Zig-Zag.
- ✓ De modifier les différents paramètres d'usinage pour arriver au résultat souhaité.
- ✓ De simuler virtuellement les mouvements de l'outil par rapport à la surface et de calculer les temps d'usinage.
- ✓ De choisir automatiquement les outils pour usiner chaque plan sans avoir des interférences.
- ✓ De choisir la combinaison optimale des outils permettant d'usiner la totalité de la surface en un minimum de temps.
- ✓ De gérer une base de données des outils.

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

- ✓ Développement de méthodes de recherche de la direction optimale d'usinage en ébauche.
- ✓ Variation adaptative de la vitesse d'avance en fonction de la quantité de matière enlevée.
- ✓ Ebauchage des surfaces gauches par d'autres méthodes.
- ✓ Simulation d'enlèvement de matière lors de l'usinage avec des outils cylindriques et toriques.
- ✓ Usinage des surfaces gauches sur des fraiseuses à 5 axes.

Annexe I — *Fonctions G-Code.*

I.FONCTIONS PREPARATOIRES G [10, 11] :

Ce sont des fonctions préparent la logique de contrôle à une action ou bien à un type de calcul. Les fonctions ci-dessous représentent l'ensemble des fonctions préparatoires :

- G00** Interpolation linéaire à vitesse rapide.
- G01** Interpolation linéaire à vitesse d'avance programmée.
- G02** Interpolation circulaire sens anti-trigonométrique à vitesse d'avance programmée.
- G03** Interpolation circulaire sens trigonométrique à vitesse d'avance programmée.
- G04** Temporisation programmable.
- G05** Exécution d'un déplacement suivant l'axe incliné.
- G06** Ordre d'exécution d'une courbe Spline.
- G07** Positionnement initial de l'outil avant un usinage suivant l'axe incliné.
- G09** Arrêt précis en fin de bloc avant enchaînement sur bloc suivant.
- G10** Bloc interruptible.
- G12** Survitesse par manivelle.
- G16** Définition de l'orientation de l'axe de l'outil avec les adresses P, R.
- G20** Programmation en coordonnées polaires (X, Z, C).
- G21** Programmation en coordonnées cartésiennes (X, Y, Z).
- G22** Programmation en coordonnées cylindriques (X, Y, Z).
- G23** Interpolation circulaire définie par trois points.
- G33** Cycle de filetage à pas constant.
- G38** Filetage enchaîné.
- G40** Annulation de correction de rayon.
- G41** Correction de rayon à gauche du profil à usiner.
- G42** Correction de rayon à droite du profil à usiner.
- G48** Définition d'une courbe Spline.
- G49** Suppression d'une courbe Spline.
- G51** Miroir.
- G52** Programmation absolue des déplacements par rapport à l'origine mesure.
- G53** Invalidation des décalages PREF et DEC1.
- G54** Validation des décalages PREF et DEC1.
- G59** Décalage d'origine programmé.
- G63** Cycle d'ébauche avec gorge.
- G64** Cycle d'ébauche paraxial.
- G65** Cycle d'ébauche de gorge.
- G66** Cycle de défonçage.
- G70** Programmation en pouce.
- G71** Programmation en métrique.
- G73** Invalidation du facteur d'échelle.
- G74** Validation du facteur d'échelle.
- G75** Déclaration d'un sous programme de dégagement d'urgence.

- G76** Transfert des valeurs courantes des paramètres «L» et «E» dans le programme pièce.
- G76+/-** Création/suppression de programme ou de bloc ISO
- G77** Appel inconditionnel de sous-programme ou d'une suite de séquences avec retour
- G77 -i** Appel du bloc de retour d'un sous programme.
- G78** Synchronisation des groupes d'axes.
- G79** Saut conditionnel ou inconditionnel à une séquence sans retour.
- G79 +/-** Suspension momentanée de la préparation du bloc suivant dans une séquence avec mouvements.
- G80** Annulation de cycle d'usinage.
- G81** Cycle de perçage centrage.
- G82** Cycle de perçage chambrage.
- G83** Cycle de perçage avec débouillage.
- G84** Cycle de taraudage rigide.
- G84** Cycle de taraudage.
- G85** Cycle d'alésage.
- G87** Cycle de perçage avec brise-copeaux.
- G89** Cycle d'alésage avec temporisation en fin de trou.
- G90** Programmation absolue par rapport à l'origine programme.
- G91** Programmation relative par rapport au point de départ du bloc.
- G92** Présélection de l'origine programme.
- G92 R..** Programmation de la vitesse l'avance tangentielle.
- G92 S..** Limitation de la vitesse de broche.
- G94** Vitesse d'avance exprimée en millimètre, pouce ou degré par minute.
- G95** Vitesse d'avance exprimée en millimètre ou en pouce par tour.
- G96** Vitesse de coupe constante exprimée en mètres par minute.
- G999** Suspension de l'exécution et forçage de la concaténation des blocs.
- G97** Vitesse de broche exprimée en tours par minute.
- G98** Définition de la valeur du X de départ pour interpolation sur l'axe C.
- G110** Optimisation des trajectoires de manutention
- G997** Validation et exécution des fonctions mémorisées dans l'état **G999**.
- G998** Validation de l'exécution des blocs et d'une partie des fonctions traitées dans l'état **G999**.

II.FONCTION AUXILLIAIRES M (MODES) :

Ces dernières actionnent les moteurs en émettant des ordres de mouvement, de sélection de vitesse ainsi que d'arrosage.

Une fonction est dite "modale" lorsqu'elle reste active (mémorisée) après le bloc où elle est écrite jusqu'à sa révocation.

Toutes les fonctions **M** décodées citées ci-dessous sont modales sauf **M06**.

- M00** Arrêt programmé, Pause (redémarrage avec le bouton START).
- M01** Arrêt programmé optionnel.
- M02** Fin de programme, retour de sous-programme.
- M03** Rotation de broche sens antitrigonométrique.
- M04** Rotation de broche sens trigonométrique.
- M05** Arrêt de broche.
- M06** Appel d'outil, changement d'outils + **T1** ou **T2**
- M07** Arrosage numéro 2.
- M08** Arrosage numéro 1.
- M09** Arrêt d'arrosage.
- M10** Blocage d'axe, pause si la pompe à vide n'est pas activée.
- M11** Déblocage d'axe, pompe à vide activée.
- M12** Arrêt d'usinage programmé.
- M19** Indexation de broche.
- M40 à M45** Gammes de broche.
- M48** Validation des potentiomètres de broche et d'avance.
- M49** Inhibition des potentiomètres de broche et d'avance.
- M61** Libération de la broche courante dans le groupe d'axes.
- M62 à M65** Commande des broches 1 à 4.
- M66 à M69** Mesure des broches numéro 1 à 4.
- M997** Forçage de l'enchaînement des blocs.
- M998** Réactivation des modes modification (MODIF), immédiat (IMD) et des appels de sous programme par fonction automatisme.
- M999** Neutralisation programmée du mode modification (MODIF), du mode immédiat (IMD) ou des appels de sous programme par fonction automatisme.

Annexe II

*Etude de la méthode
de modélisation
"UML".*

ETUDE DE LA METHODE DE MODELISATION « UML »

1. Définition d'UML [22, 29]:

UML n'est pas une méthode mais un langage de modélisation en objets graphique adopté par toutes les méthodes objet, permettent de spécifier, construire et documenter le développement d'un système, il est constitué d'un ensemble de diagrammes, notes, exigences présenté dans un document (rédiger lors du développement d'un système).

Il est indépendant de toute plateforme logiciel/matériel de domaine ^{du} gestion, ^{de} ingénierie, finances...

UML standardisé par OMG (Object Management Group).

2. Histoire d'UML [19, 30]:

Tout a débuté en 1970 lors de la sortie de la première méthode de conception de Yordon 1979 qui été de type structurée, puis la sortie du premier langage orienté objet Simula-67 (1967), les années 80 été marqué par la sortie de plusieurs langages de programmation objet : Smalltalk, objectif C, C++, Eiffel, Clos...

En 1988 sortie de la première méthode orientée objet de OOSA (Schlaer & Mellor).

1991 : OOD Object Oriented Design (Grady Booch) chez General Electric , OOA/OOD (Coad & Yourdon), OMT Object Modeling Technique (James Rumbaugh) chez Rational Software.

1992 : OOSE Object oriented software engineering (Jacobson & al.) chez Ericsson et plus de 50 autres méthodes.

1993 : HOOD Hierarchical Object Oriented Design (Delatte & al.), OOM Object oriented Merise (Bouzeghoub & Rochfeld).

1994 : Tentative d'unification : Fusion (Coleman & al.). Fin 1994 : Rumbaugh (OMT) rejoint Booch (OOD) chez Rational Software afin de créer une méthode en commun.

1995 : Ivar Jacobson rejoint rational Soft^ware Corp présentation v0.5.

Octobre **1995** : The Unified Method, version 0.8.

Octobre **1996**: addendum 0.91: UML, The Unified Modeling Language for Object-oriented Development. Dans la même année une appelle d'offre de OMG (RFP) pour une approche normalisée pour les développements objets.

Janvier 1997 : UML version 1.0, soumission à l'OMG, **Novembre 1997** : UML version 1.1 adoptée par l'OMG.

Avril 1999 : UML version 1.4, **Mars 2000** : UML version 1.5.

2001 : début des travaux sur UML 2.

Automne 2002 : spécifications de UML 2.0.

3. Qu'est ce que UML [22, 30]?

Est une notation et possède une sémantique, composé de diagrammes afin de modéliser les aspects statiques, dynamiques et organisationnels du système en unifiant les connaissances acquise dans le domaine orienté objet.

La modélisation d'un système s'effectue indépendamment de toute méthode ou langage de programmation. Adapté aux méthodes objets : Unified Software Development Process, Catalysis. Convient à tous les langages objets : C++ (Héritage multiple, Template), Java (Interface), SmallTalk.

C'est un langage comprenant un vocabulaire et des règles centrées sur la représentation physique d'un système logiciel, ses domaines d'utilisations sont la visualisation, spécification, construction et documentation d'un système.

Lors du développement avec UML, le processus passe par les étapes.

- *Expression des besoins* : définition des attentes de l'utilisateur.
- *Analyse* : aboutis à un modèle proche de la solution en faisant abstraction aux particularités de l'implémentation.
- *Conception* : son résultat est un modèle adapté au monde réel qui est indépendant du langage de implémentation qui défini la structure et le comportement.
- *Architecture* : exploitation des différentes vues d'UML pour englober tous les aspects du système à développer.
- *Validation* : des parties livrables à l'utilisateur.
- *Itérative & incrémental* : créer un prototype puis le raffiner au fur et à mesure de l'avancement du projet.

Puisque UML est issue de plusieurs méthodes, il tire le meilleur d'eux, OOSE (*Cas d'utilisation «Use Cases»*), OMT (*Analyse*), OOD (*conception, Architecture*).

4. Objectif d'UML [29, 30]:

L'objectif majeur d'UML est la simplification du développement en objet, mais pour le satisfaire un ensemble d'autres objectifs sont parus.

- Représenter des systèmes entiers par sa syntaxe riche et expressive.
- Indépendance de toute technologie et langages d'implémentation.
- Fournir une base formelle pour la modélisation.
- Fournir des mécanismes d'extension.
- Créer un langage de modélisation utilisable à la fois par les humains et les machines.
- Recherche d'un langage commun unique.

la simplification du développement en objet

5. Le langage OCL [22, 31, 32]:

OCL (pour Object Constraint Language) est un langage déclaratif qui permet d'exprimer des contraintes sur des modèles UML. Exprimer une contrainte consiste à donner un contexte, typiquement une *classe* (type englobant les objets) et une méthode, ainsi que des pré-conditions et des post-conditions. Le sens d'une contrainte OCL pourrait être le suivant :

"Pour que la méthode *m* définie dans la classe *C* s'exécute, il faut que la pré-condition soit réalisée. Après son exécution la post-condition sera réalisée".

6. Les vues de UML [21, 22] :

Il existe cinq différentes vues :

- La vue logique se concentre sur l'abstraction et l'*encapsulation* (consiste à englober un objet et cacher sa structure interne), classes, stéréotypes, diagramme de classes.
- La vue d'implantation est une vue de bas niveau composée de modules logiciels.
- La vue des processus utilisé dans un environnement multitâches.
- La vue de déploiement est utilisée dans un environnement distribué pour représenter l'architecture et la topologie du système.
- La vue des cas d'utilisation ou de *scénarios* (définis le comportement du système) cette vue qui sert de guide à toutes les autres vues.

7. Les constituants d'UML [22, 31] :

Notion de base:

Classe : est une description d'un ensemble d'objets, qui partage les mêmes attributs, opération, relation et contraintes.

Interface : type particulier de classe.

Objet : une entité avec une identité bien définie qui englobe un *état* (l'ensemble des attributs et de la relation à un instant donné), le comportement est représenté par des opérations et des méthodes. Un objet est une instance d'une classe.

Package : regroupement de plusieurs classes et interfaces.

Un modèle conceptuel comprend trois éléments de base :

Élément : représente une abstraction d'un système qui peut être structurel (classe, interface, collaboration...), comportementale (interaction, automate à état...), de regroupement (package), d'annotation (note).

Relation : lien entre les éléments de type : dépendance, association, généralisation ou réalisation.

Diagramme : regroupe des éléments avec leurs liens.

Modéliser en UML revient à schématiser le système par un ensemble de notes, de *diagrammes* (support graphique de modélisation), qui sont au nombre de 9 principaux (il existe en tout 12), chaque diagramme propose un point de *vue* différent (vue statique, dynamique, organisationnelle, ...). Avec un mécanisme d'extension.

7.1. Relation entre diagrammes et étapes du processus avec les différentes vues [21, 22, 28, 29, 30]:

Pour montrer les limites d'un système et ses fonctions principale on utilise les *cas d'utilisation* (Use Case) 'OOSE' et les *acteurs* (élément d'un système peut être une personne, un service ou un autre système) présentent une vue externe du système illustré par un *diagramme de cas d'utilisation* qui présente une *vue statique (fonctionnelle)* parmi les 4 autres type de diagramme, ce diagramme représente l'étape de *découverte des besoins* dans le cycle de vie du logiciel.

Diagramme du cas d'utilisation : Un cas d'utilisation est un moyen de représenter les différentes possibilités d'utiliser un système.

- . Il exprime toujours une suite d'interactions entre un acteur et l'application.
- . Il définit une fonctionnalité utilisable par un acteur.

Voici ci-dessous les différents composants graphiques (Figure.1) d'un cas d'utilisation avec un exemple (Figure.2).

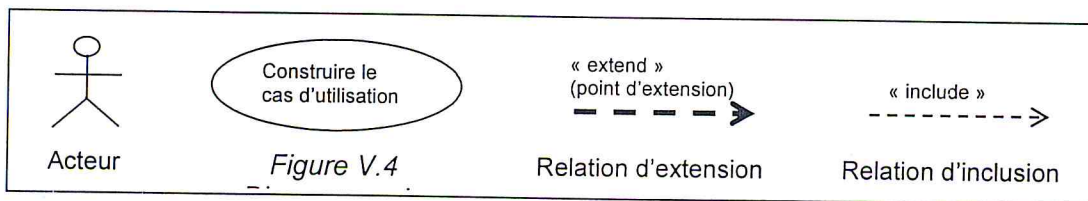


Figure.1: composant graphique du cas d'utilisation.

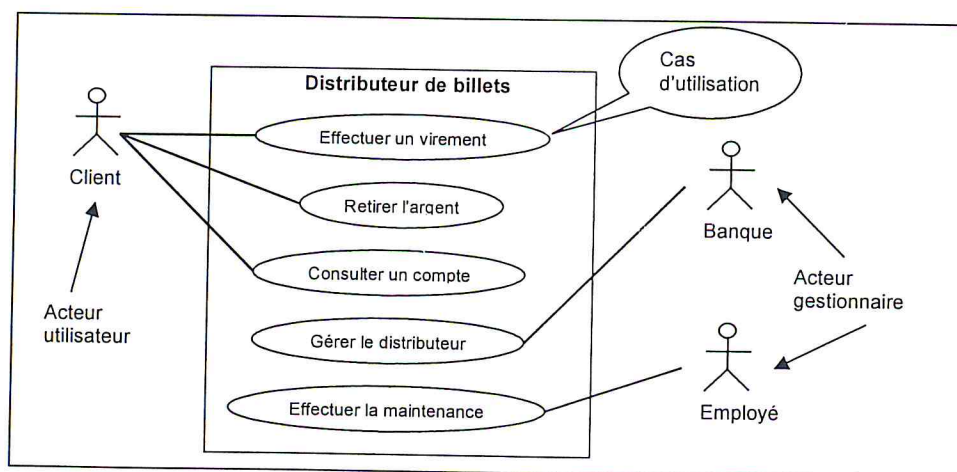


Figure.2: schéma représentant cas d'utilisation d'un distributeur de billets.

Pour montrer la *structure (vue) statique (Architecture)* d'un système on utilise les quatre diagrammes qui suit .

Diagramme de classes 'OMT, BOOCH', structure des données du système définies comme un ensemble de relations entre classes, se trouve à l'étape d'Analyse dans le cycle de vie du logiciel. La (Figure.3) schématise une classe.

Remarque. Les compartiments d'une classe peuvent être omis.

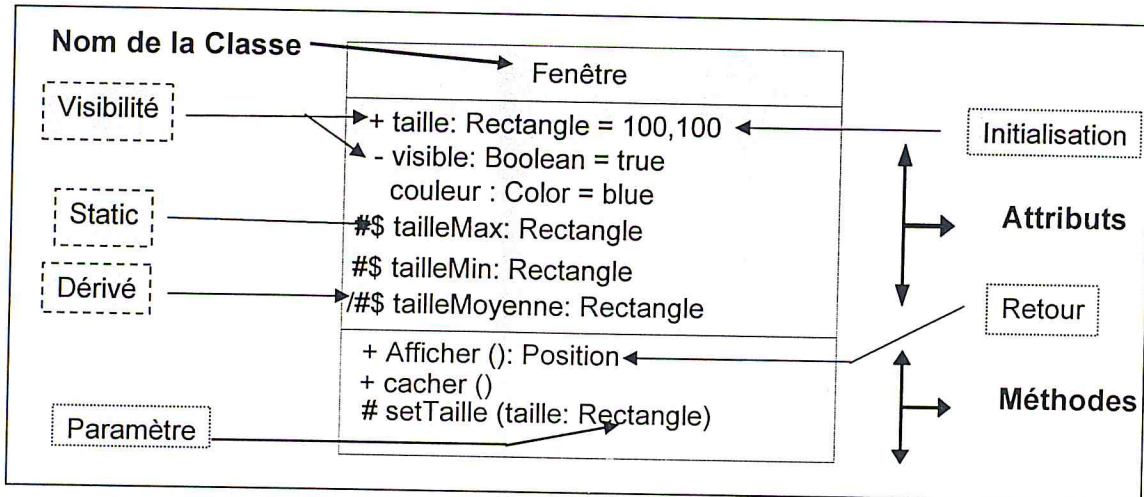


Figure.3 : schéma représentant une classe.

La syntaxe des attributs et des méthodes est représentée par la (Figure.4), celle des liens (Figure.5), avec un exemple récapitulatif et en (Figure.6) montrons le tout.

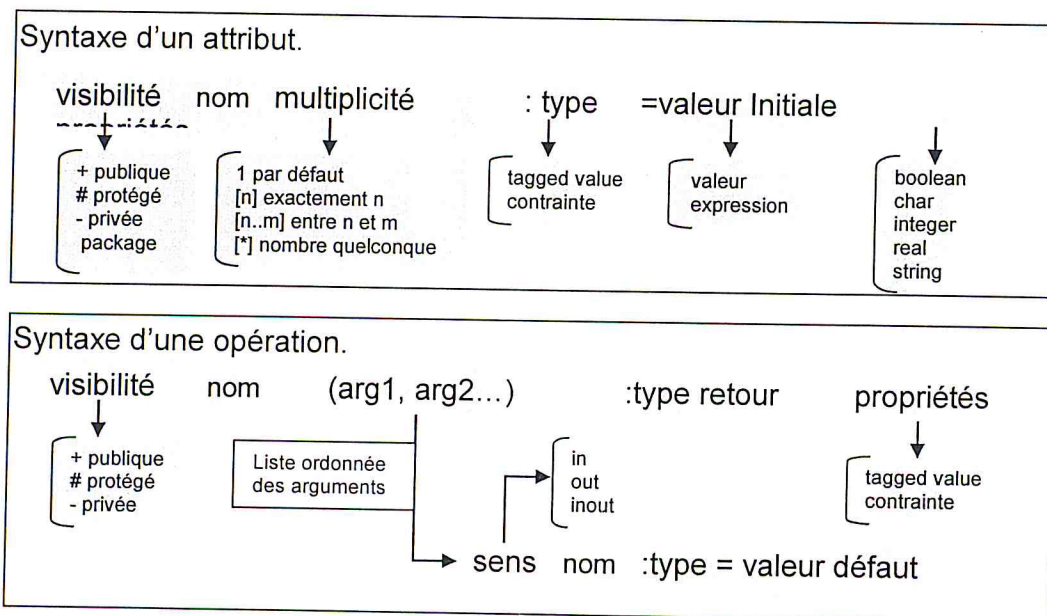


Figure.4: Définition des attributs et des méthodes.

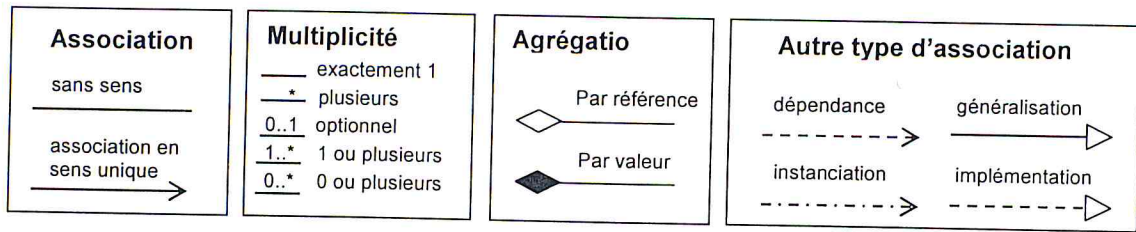


Figure.5: schématisation des liens entre les classes.

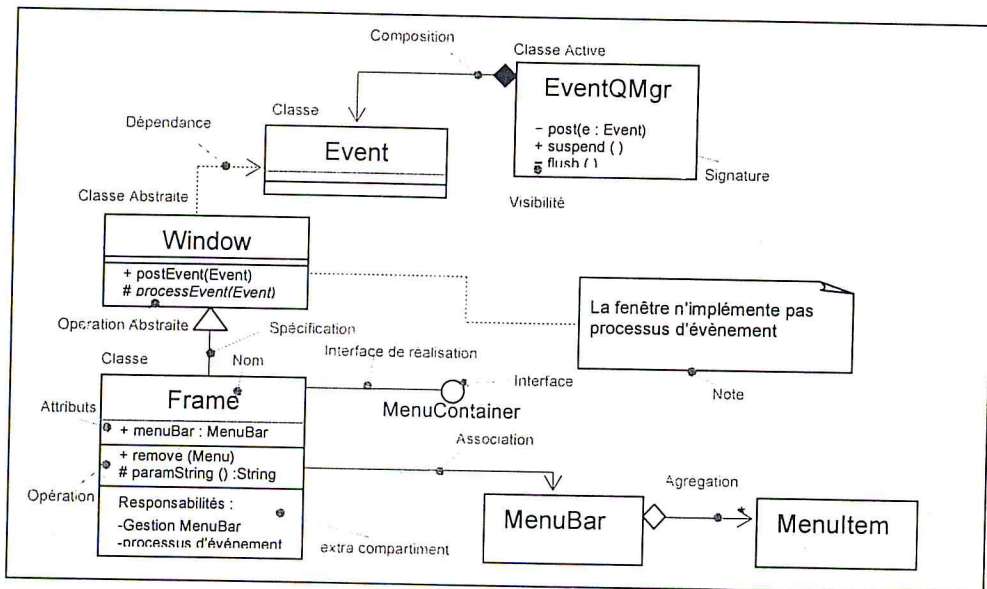


Figure.6 : exemple d'un diagramme de classes [23]

Diagramme d'objets 'OMT, BOOCH', représentation des objets et de leurs relations, correspondent à un diagramme de collaboration simplifié, sans représentation des envois de messages, correspond l'étape d'Analyse.

Pour le reste c'est la même chose, sauf que le nom d'objet est souligné, et suivi du : nom de la classe à qui il appartient; exemple: Airbus A380 : Avion. (Voir Figure 7,8).

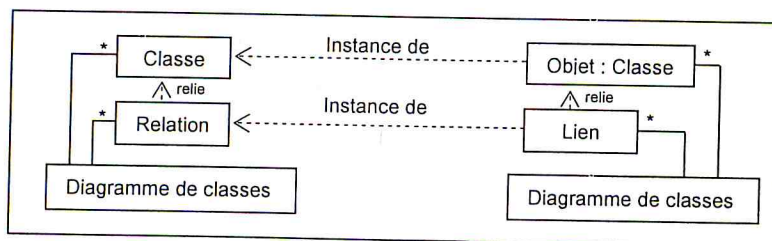


Figure.7 : relation entre diagramme de classes et diagramme d'objets.

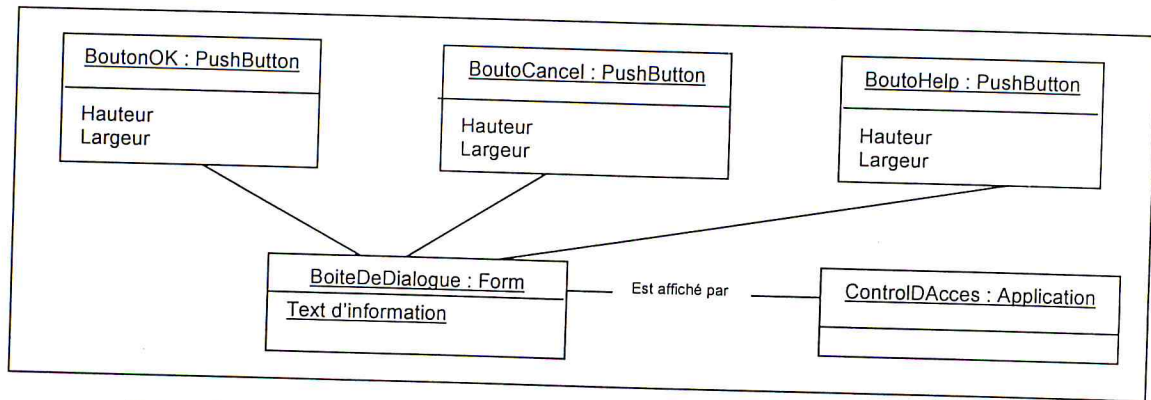


Figure.8 : exemple d'un diagramme d'objet cas de boîte de dialogue.

Pour représenter le comportement du système (vue dynamique) on utilise 4 diagrammes comportementaux.

Diagramme de séquence 'OMT, OMSC', pour chaque cas d'utilisation, description temporelle de l'interaction d'un acteur sur le système = scénario, dans la réalisation d'une interface Homme/Système, l'étape de *Conception*. (Figure.9)

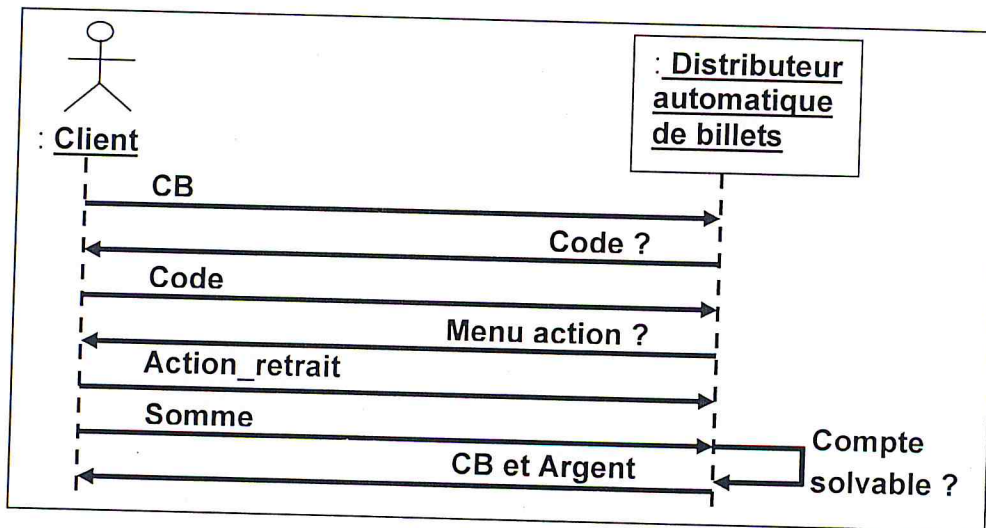


Figure.9 : Diagramme de séquence (cas d'un DAB).

Diagramme de collaboration [28] 'BOOCH, FUSION', Représentation spatiale des objets, établissement des responsabilités qui coopère avec qui pour l'obtention d'un but, l'étape d'*Analyse* dans le cycle de vie du logiciel. (Figure.10).

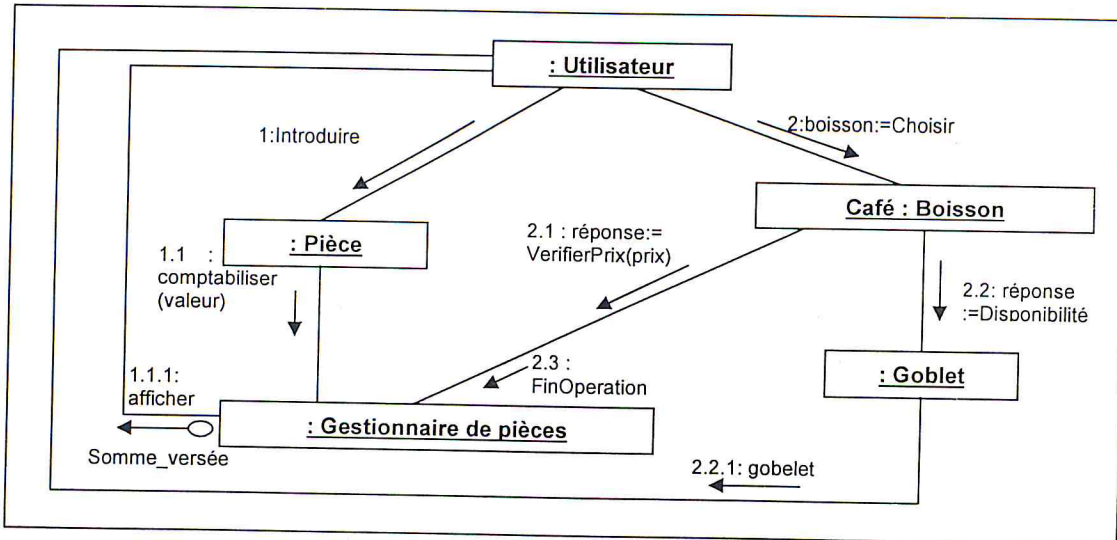


Figure.10 : Diagramme de collaboration (cas d'un distributeur de boisson).

Diagramme d'activités 'HAREL, WORKFLOW', Il s'agit d'une variante du diagramme de transition d'état, organisé par rapport aux actions et destiné à représenter le comportement interne d'une méthode ou d'un cas d'utilisation. Contenu dans l'étape d'Analyse. (Figure.11, 12).

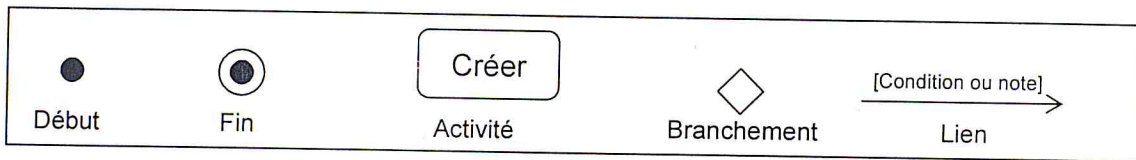


Figure.11: composants d'un diagramme d'activité.

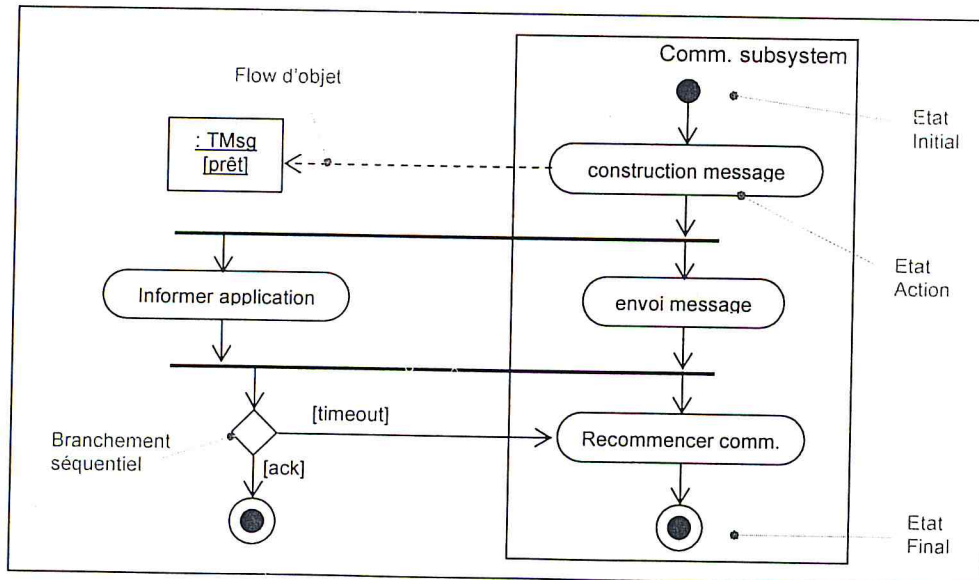


Figure.12: Diagramme d'activité (cas d'une communication) [23]

Diagramme de transition d'état 'HAREL', représentation du comportement Des objets d'une classe en terme d'états et de transitions, lié au cycle de vie des objets, l'étape d'Analyse. (Figure.13,14).

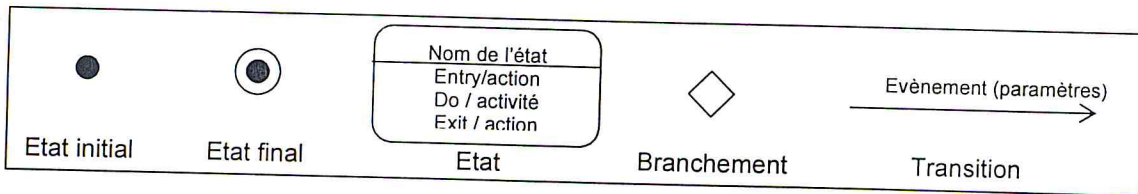


Figure.13 : composants d'un diagramme de transition d'état.

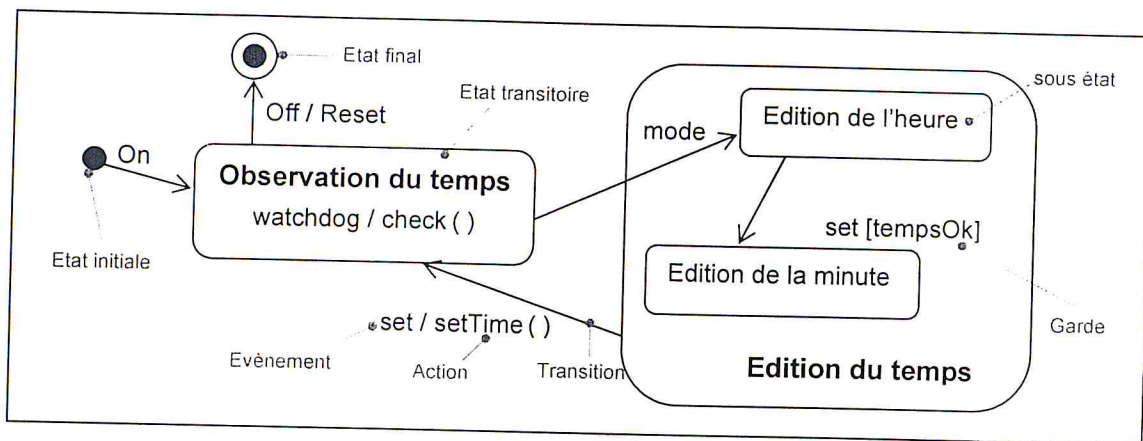


Figure.14 : Diagramme de transition d'état (cas d'une montre) [23].

Diagramme de composant 'BOOCH', architecture des composants physiques d'une application, (Vue organisationnelle) l'étape de Conception dans le cycle de vie du développement de l'application (Figure.15, 16).

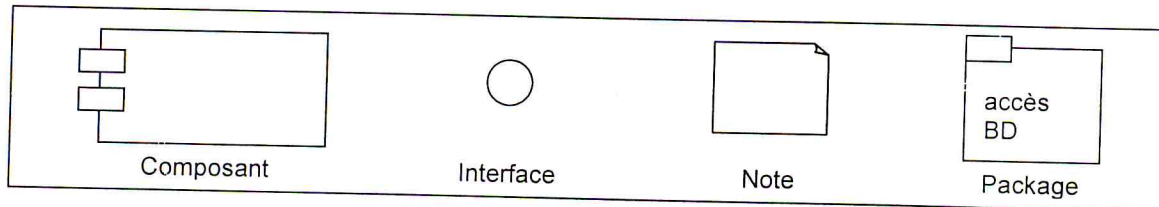


Figure.15 : composants d'un diagramme de composant.

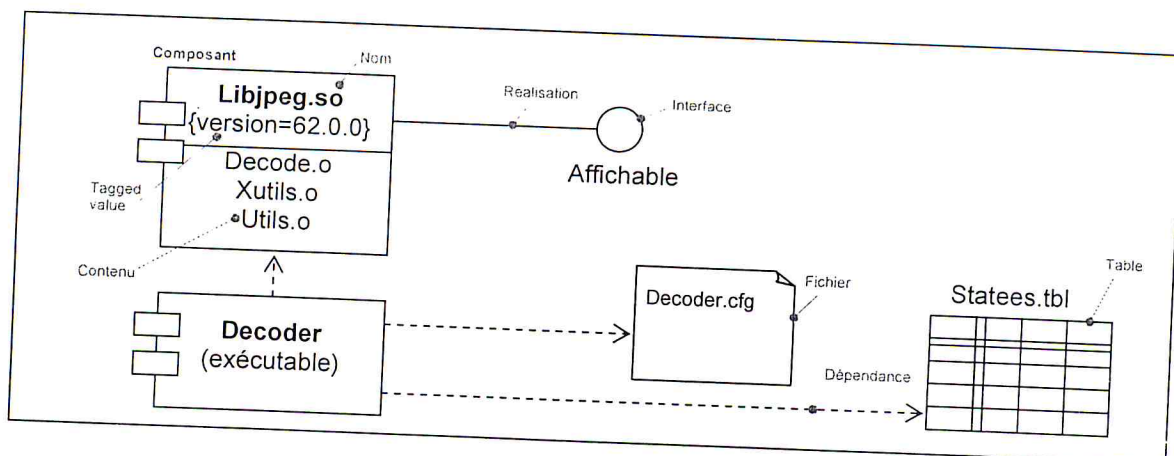


Figure.16 : exemple d'un diagramme de composant cas décodage d'un fichier JPEG [23]..

Diagramme de déploiement 'BOOCH', Le diagramme de déploiement décrit les différentes ressources matérielles (Figure.17), l'implémentation du logiciel dans ces ressources, et les modes de communication entre les composants physiques du système. On y retrouve le processeur (de données ou de traitements) et ses périphériques, (*Vue de déploiement*) correspond l'étape de Conception dans le cycle de vie du logiciel.

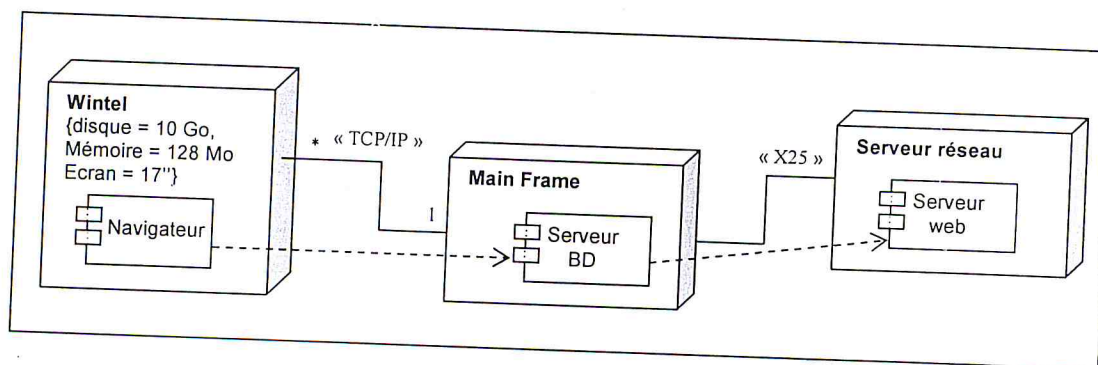
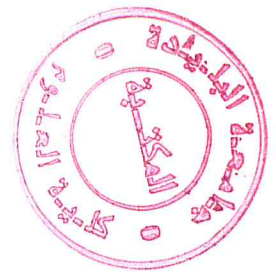


Figure.17 : Diagramme de déploiement avec composants.

REFERENCES BIBLIOGRAPHIQUES



- [1] Technique de l'ingénieur, Traité informatique H3750, YVON GARDAN (Directeur de l'institut de formation technique supérieure (IFTS)).
- [2] Technique de l'ingénieur, traité mécanique B7160 B,7130
- [3] " modélisation des courbes et des surfaces ", rapport de recherche CDTA Bey Mohamed Avril 2000
- [4] " Géométrie des courbes et des surfaces". Y. Kerbrat, J-M. Braemer Edition Hermann.1996.
- [5] " Méthode des plans parallèles pour l'usinage des surfaces gauches sur des fraiseuses à commande numérique à 3 axes " Projet de fin d'études 2003 BOUDJOUAD Salim. TAFAT-BOUZID Nazim, USTHB Algérie
- [6] "Mathématique des courbes et des surfaces, modèles de Bézier des B-splines, de NURBS". Gilbert Demengle, Jean-Pierre Pougep Editions Ellipse 1998.
- [7] "Approximation des courbes et des surfaces" L. Granger, F. Guibault, G. Pesant Ecole polytechnique. Montréal Août 1999.
- [8] Mark Daniel " Modélisation de courbes et surfaces par des B-plines, application à la conception et à la visualisation formes". Thèse de Doctorat. Université de Grenoble, Mai 1989.
- [9] "Machine outils a commandes numériques structure, modélisation et réglage", Emmanuel Duc, Emmanuel Lefur Mémoire de DEA à l'école normale supérieure de Cachan, France Septembre 1997.
- [10] " Numerically controlled machine tools", Ellis Horwood, G.W Vickers, M.H Ly, R.G Oetter (ISBN 0-13-625526-4), University of Victoria, Canada.
- [11] Catalogue Commandes numériques Num Power 1020-1040-1050-1060-1080 Entraînements NUM DRIVE.
- [12] "Usinage de formes gauches; contribution à l'amélioration de la qualité des trajectoires d'usinage ". Emmanuel Duc Thèse de Doctorat à l'école normale supérieure de Cachan, France 1998.

- [13] "Evaluation des modes de génération des trajets outil pour l'usinage de formes gauches. Etude de cas". Christophe Tournier Mémoire de DEA à l'école normale supérieure de Cachan, France 1996.
- [14] "Corner cutting curves and new characterization of Bézier and B-splines curves". Günter Aumann Computer Aide Geometric Design, volume 14, pp 449-474, 1997.
- [15] "Interference detection and optimal tool selection in 3- axis NC machining of free form surfaces", Daniel Yang, Zhonglin Han volume 31. Elsevier Ltd, USA 1999.
- [16] "Evaluation des modes de génération des trajets outil pour l'usinage de formes gauches ". Emmanuel Duc Mémoire de DEA à l'école normale supérieure de Cachan, France Juillet 1996.
- [17] "Cut distribution & cutter for sculptured surface cavity machining ", document "numéros 06 (1447-1470) vol 30" du protocole de recherche au centre. d'ingénieur de la recherche pour la fabrication intelligente des système, a l'université de Purdue ouest Lafayette USA.
- [18] Surfaces à Subdivision Geri's Game (1989) : Pixar Animation Studios
- [19] " l'intégral des maths " Site Internet de Etienne Mauvais 1998
- [20] LOG2000 – Éléments du génie logiciel © 2002 – Bayomock André-Claude
- [21] Unified Modeling Language Modélisation Objet Omar El Beqqali (MIL-2001)
- [22] Analyse et Conception Orientées Objet avec UML et réalisation en C++ (RéférenceOO-226) Guide étudiant Sun Microsystems France S.A. Service Formation
- [23] *Modélisation et Programmation Orientées –Objet* Mai 2004 Pierre Chevaillier
- [24] CAHIER DE CHARGE DE CONSULTATION " EXTERNALISATION DE LA FONCTION INFORMATIQUE" Thierry ROBERT société "PORTZAMPARC" juillet 2002
- [25] Comment faire un cahier des charges ? société "roflo&cie" octobre 2003
- [26] CAHIER DES CHARGES "PROJET ING2 – Jeu d'Echecs" société "Laboratoire des Technologies .Net" JURY Frédéric, MORLE Ralph-Gabriel, ELBAZ Pascal Alexandre, Pascal Grégory, HUPKES Thomas décembre 2003
- [27] Document Officiel - Comment Rediger Un Cahier Des Charges
- [28] Analyse et Conception avec UML Mireille Blay-Fornarino 2005
- [29] Conception avec UML Gerson Sunyé IRIN - Université de Nantes 2003

- [30] UML (Unified Method Language) - Langage unifié pour la modélisation objet - Frédéric Julliard Université de Bretagne Sud UFR SSI - IUP Vannes - année 2001-2002
- [31] Méthode d'analyse orientée objet UML F. Bernardi, 2002
- [32] Etude de Transformations de modèle UML Rapport de TER Maîtrise informatique 2001-2002 Laurent Morin
- [33] UML Quick Reference Card Copyright © 2001 Laurent Grégoire
- [34] Notation UML Antonio GONCALVES Antonio GONCALVES 1998
- [35] UML Fiche technique Audet, Lucie 2003 © Gilles E. St-Amant, cours ADM7705
- [36] La genèse d'UML Pierre-Alain Muller ESSAIM (1997)