

F.S.D .....N°D'ordre .....

Université Saad DAHLAB de Blida



Faculté des Sciences  
Département d'Informatique

Mémoire présenté par :  
M<sup>lle</sup> KELLOU Ratiba  
M<sup>lle</sup> BETTOU Mahdia

En vue d'obtenir le diplôme de Master

**Domaine:** MI

**Filière:** Informatique

**Spécialité:** Informatique

**Option:** Ingénierie de logiciel

Sujet :

**Tréflage des Surfaces Gauches  
Discrètes par la Combinaison  
de Différents Outils**

**Encadreur :** BEY Mohamed  
TCHANTCHANE Zahida

**Promoteur :** TOUBALINE Nesrine

**Organisme d'accueil :** CDTA (Centre de Développement des Technologies Avancées).  
Division Productique et Robotique

**Soutenue le:** 01/07/2012 , devant le jury composé de :

Mr Aitacache président  
Mlle Bacha examinateur 1  
Mme Boumahdi examinateur 2

Année universitaire : 2011/ 2012

## REMERCIEMENT



*Au terme de ce travail nous tenons à remercier tout d'abord ALLAH tout puissant de nous avoir donné le courage et facilité la réalisation de ce travail.*

*Nous exprimons notre grande reconnaissance et remerciement à notre encadreur Mr. Bey Mohamed pour son aide, ces conseils et de nous avoir orienté et mis à notre disposition tous les moyens nécessaires durant notre travail.*

*Nous exprimons également notre gratitude à Mme Tchantchane, à notre promotrice Melle Toubaline, à Mr Gaham et à Mr Kouider qui ont collaboré à nous éclairer.*

*Et enfin, que nos chers parents et amis, et bien avant tout, trouvent ici l'expression de nos remerciements les plus sincères et les plus profonds en reconnaissance de leurs sacrifices, aides, soutien et encouragement pour atteindre nos buts.*

*Merci*

# Dédicace

*Je dédie ce modeste travail à mes très chers parents que je leurs exprime ma grande reconnaissance et gratitude pour leurs aide et l'encouragement qu'ils m'ont continuellement apportés, je leur souhaite le bonheur et la bonne santé.*

*A ma petite famille : mes grands-parents que j'adore ,ma Sœur Radhia et mon petit frère Ibrahim ,*

*A toute ma famille oncles tantes cousins cousines ;*

*Et a tous mes amis, et en particulier à :ma très chère Amina, à Salim, imene, Wassila, khaled, Abdelah , Samir , Nadir , Riadh , Chahine, lyes, mohamed, Ahmed , Nacer Amine et nesrine.*

*A mon amie et binôme Mehdia et sa famille;*

*Merci à tous ceux qui m'ont aidé de près ou de loin*

# *Dédicace*

*Je dédie ce modeste travail à mes très chers parents  
qui m'ont toujours soutenu pendant toute ma vie.*

*A toute ma famille ; mon frère et mes Sœurs*

*Et surtout ma grande mère;*

*A ma collègue de binôme Amina, et sa grande  
famille;*

*A tous mes collègues.*

*A toute personne porte des sentiments pour moi ;*

*Merci à tous ceux qui m'ont aidé de près ou de loin.*

***Mehdia***

## Résumé

Ce projet s'insère dans le cadre de développement d'outils de conception et de fabrication des surfaces gauches initié par l'équipe Conception et Fabrication Assistées par Ordinateur (CFAO) de la Division Productique et Robotique du Centre de Développement des Technologies Avancées (CDTA). Dans ce projet et à partir des modèles STL des surfaces gauches, nous optimisons l'opération de Tréflage des pièces de formes complexes sur des fraiseuses numériques à 03 axes par la détermination de la combinaison optimale des outils permettant de minimiser le temps d'usinage tout en maximisant le volume de matière à enlever. La finalité de ce travail est le développement d'une application logicielle graphique et interactive sous Windows permettant à partir des modèles STL, de déterminer la liste des outils utilisables, de calculer la combinaison optimale des outils, de déterminer les positions de plongée valides, les profondeurs de plongée associées, le chemin le plus court et le programme d'usinage « G-Code ».

**Mots clés :** Modèle STL, Optimisation, Tréflage, Interférence, Collision, Outils Optimums.

## Abstract

This project fits into the context of developing tools for designing and manufacturing freeform surfaces initiated by Computer Aided Design and Manufacturing Team at the "CDTA". In this project and from STL models of free form surfaces, we optimize the operation of plunging milling machining strategy for roughing parts of complex shapes on 03-axis CNC milling machines by determining the optimal combination of tools minimizing the machining time while maximizing the volume of material to be removed. The purpose of this work is the development of a graphical and interactive software application under Windows in order to determine and to calculate the optimal combination of tools, plunging positions and their plunging depth and finally the optimum tool path and its associated machining program "G-Code".

**Keywords :** STL Model, Optimization, Plunging, Interference, Collision, Optimum Tools.

## SOMMAIRE

INTRODUCTION GENERALE .....	1
I. SITUATION DU PROBLEME .....	2
II. OBJECTIF DU TRAVAIL .....	2
III. STRUCTURATION DU MEMOIRE .....	2
<b>CHAPITRE 1 : GENERALITES SUR LE PROCESSUS D'USINAGE DES SURFACES GAUCHES</b> .....	3
I. INTRODUCTION .....	3
II. FORMAT STL .....	3
II.1. Définition.....	3
II.2. Caractéristiques du fichier STL .....	4
III. MACHINE OUTIL A COMMANDE NUMERIQUE (MOCN).....	4
III.1. FONCTIONS D'UNE MOCN .....	4
III. FRAISEUSE .....	5
III.1. Définition .....	5
III.2. Types de fraiseuses .....	5
III.3. Avantages des fraiseuses à commande numérique .....	5
IV. USINAGE DES SURFACES GAUCHES .....	5
IV.1. Outils d'usinage .....	6
IV.2. Problèmes d'usinage .....	6
IV.3. Ebauchage des surfaces gauches .....	7
IV.3.1. Stratégies d'ébauchage.....	7

IV.3.1.1. Stratégie des « Plans Parallèles ».....	7
IV.3.1.2. Stratégie des « Contours Décalés ».....	8
IV.3.1.3. Stratégie de « Tréflage ».....	8
IV.3.1.3.1. Outils en Tréflage .....	9
IV.3.1.3.2. Paramètres limites de coupe.....	9
IV.3.1.3.3. Types de géométrie de plaquettes.....	10
IV.3.1.3.4. Paramètres pilotant l'opération de Tréflage.....	11
IV.3.1.3.5. Mouvements de l'outil.....	11
IV.3.1.3.6. Mouvement de coupe .....	11
IV.3.1.3.7. Mouvement d'avance .....	12
IV.3.1.3.8. Mouvement résultant de coupe .....	12
IV.3.1.3.9. Trajectoire de la partie active de la fraise en Tréflage.....	12
IV.3.1.3.10. Trajectoires d'usinage .....	12
IV.3.1.3.11. Modes de balayage de l'outil .....	13
IV.3.1.3.12. Avantages du Tréflage .....	14
V. CONCLUSION .....	14
<b>CHAPITRE 2. CONCEPTION DE L'APPLICATION LOGICIELLE ...</b>	<b>15</b>
I. INTRODOCTION .....	15
II. METHODE DE MODELISATION .....	15
III. REALISATION DE L'APPLICATION .....	15
III.1. Cahier de charges .....	15

III.1.1. Présentation du sujet.....	15
III.1.2. Problématique .....	15
III.1.3. Objectifs du travail .....	16
III.1.4. Plateforme exigée .....	17
III.1.5. Solutions proposées de la problématique .....	17
IV. MODELISATION DE L'APPLICATION EN UML .....	18
IV.1. Diagramme cas d'utilisation .....	18
IV.2. Diagramme de séquence .....	20
IV.4. Diagramme de collaboration .....	23
IV.5. Diagramme de classe .....	25
V. CONCLUSION .....	30
<b>CHAPITRE 3: DEVELOPPEMENT ALGORITHMIQUE ET IMPLEMENTATION</b> <b>INFORMATIQUE .....</b>	<b>31</b>
I. INTRODUCTION .....	31
III. IMPLEMENTATION.....	31
III.1 Fenêtre principale .....	31
III.2. Barre du menu principal.....	31
III.3. Rubrique d'ébauchage par Tréflage à partir d'un fichier STL .....	32
IV. PRESENTATION DES FENETRES.....	32
IV.1. Fenêtre d'ouverture d'un fichier STL.....	32
IV.1.1. Lecture du fichier STL .....	33
IV.1.2. Création des cellules des triangles .....	33
IV.1.3.Création des dexels .....	33

IV.2. Fenêtre de calcul des outils utilisables .....	36
IV.2.1. Calcul du rayon initial .....	37
IV.2.2. Calcul du rayon théorique (intermédiaire) .....	38
IV.2.3. Calcul du rayon final .....	38
IV.3. Fenêtre de calcul des positions de plongée .....	39
IV.3.1. Calcul des positions de plongée .....	40
IV.3.2. Calcul des profondeurs de plongée, du volume enlevé et du temps.....	43
IV.4. Fenêtre de détermination de la combinaison optimale des outils.....	43
IV.4.1. Fonction objective .....	44
IV.4.2. Algorithme du recuit simulé.....	44
IV.4.3. Perturbation.....	45
IV.5. Fenêtre de détermination du trajet d'usinage .....	46
IV.5.1. Perturbation .....	47
IV.5.2. Fonction objective .....	48
IV.5.3. Génération du trajet d'usinage .....	48
V. CONCLUSION .....	49
<b>Chapitre 4 : Test et Validation .....</b>	<b>50</b>
I. INTRODUCTION .....	50
II. TESTS ET VALIDATIONS .....	50
II.1. Récupération des points et des triangles et détermination des contours .....	50
II.2. Détermination des triangles des cellules.....	50
II.3. Détermination des parties à usiner du modèle STL .....	51
II.4. Détermination des rayons utilisables.....	52

II.4.1. Déterminer du rayon final .....	52
II.4.2. Détermination des rayons minimum et maximum de chaque contour.....	53
II.5. Détermination des combinaisons .....	54
II.6. Détermination de la combinaison optimale théorique .....	55
II.7. Détermination du trajet d'usinage .....	58
II.7.1. Optimisation du trajet d'usinage.....	58
II.7.2. Génération du trajet d'usinage .....	60
II.8. Génération du programme d'usinage .....	63
III. CONCLUSION .....	63

**Conclusion générale**

**INTRODUCTION**

**GENERALE**

# INTRODUCTION GENERALE

## I. SITUATION DU PROBLEME :

Le développement incessant de l'outil informatique et son utilisation dans tous les domaines et en particulier en mécanique par la mise sur le marché de logiciels dédiés à la conception et à la fabrication assistées par ordinateur (CAO, FAO et CFAO) a permis aux entreprises industrielles de réduire le cycle de développement de nouveaux produits, d'augmenter la productivité et d'améliorer la qualité des produits réalisés. Ces logiciels assistent les ingénieurs dans les phases de conception, d'analyse et de planification de l'usinage sur des machines numériques.

Les pièces avec des surfaces de forme libres (surfaces gauches) sont utilisées dans diverses industries : aéronautique, automobile, moules, matrices ...etc. Leurs complexités géométriques exigent l'utilisation des logiciels de CFAO (Conception et Fabrication Assistées par Ordinateur) dans les phases de conception et de fabrication. Ces surfaces sont usinées sur des fraiseuses numériques à 03, à 04 ou à 05 axes par le choix des stratégies d'usinage, des formes et des dimensions d'outils, des conditions de coupe ...etc. La forme finale de ces surfaces est obtenue en trois étapes : ébauche, demi-finition et finition en générant le programme d'usinage « G-Code ».

Le travail que nous présentons dans ce mémoire s'inscrit dans le cadre de développement d'outils de conception et de fabrication des surfaces gauches initiés par l'équipe Conception et Fabrication Assistées par Ordinateur (CFAO) de la Division Productique et Robotique du Centre de Développement des Technologies Avancées « CDTA ». Notre projet est une continuité de plusieurs travaux :

- Modélisation et conception des courbes et des surfaces gauches,
- Finition des surfaces gauches avec différentes stratégies de finition,
- Ebauchage des surfaces gauches avec différentes stratégies d'ébauchage,
- Adaptation des vitesses d'avance en ébauche et en finition,
- Simulation virtuelle de l'usinage en ébauche et en finition,
- Choix des outils optimaux à partir d'un nuage de points irrégulier,
- Finition des surfaces gauches à partir d'un nuage de points quelconque,
- Tréflage des surfaces gauches en utilisant les modes de balayage « Plans Parallèles » (One-Way et Zig-Zag) et « Contours Décalé »,
- Combinaison de plusieurs formes d'outils pour l'opération de finition.

Les modèles des surfaces gauches sont obtenus en utilisant soit un logiciel de CAO si les formes sont plus ou moins simples, soit le processus du « Reverse Engineering » si les formes sont très complexes ou si les modèles CAO ne sont pas disponibles. Dans la deuxième méthode, la reconstruction du modèle CAO à partir d'un nuage de points est une tâche très délicate et consommatrice de temps. Pour la simplifier, le nuage de points est transformé en un modèle STL. En même temps, ces modèles STL peuvent

être générés à partir des modèles CAO dans des logiciels de CAO afin d'effectuer des analyses spécifiques dans des logiciels spécialisés.

L'opération d'ébauchage consiste généralement à enlever l'excédent de matière dans des plans horizontaux par le choix entre deux stratégies « Plans Parallèles » et « Contours Décalés ». Afin de minimiser les temps d'ébauchage des cavités profondes, une nouvelle stratégie a été développée c'est le « Tréflage » où plusieurs modes de balayage d'outils peuvent être choisis (One-Way, Zig-Zag, Spirale, ...etc.). Cette stratégie nécessite un choix judicieux du nombre et des dimensions d'outils, des modes de balayage des outils, des conditions de coupe, des positions de plongée valides des outils et des profondeurs associées. Le résultat du Tréflage est une géométrie en escaliers d'épaisseurs non uniformes. Cette stratégie devient plus complexe si elle est appliquée sur des pièces avec des surfaces gauches ce qui nécessite le développement de procédures spécifiques dans le but d'automatiser et d'optimiser le Tréflage. Pour optimiser cette stratégie, il est nécessaire de choisir judicieusement la séquence des outils à utiliser afin d'enlever le maximum de matière en un minimum de temps tout en évitant les problèmes d'interférences et par la suite déterminer les profondeurs de plongée et le chemin d'outils le plus court. La forme géométrique très complexe des surfaces gauches rend cette tâche très difficile.

## II. OBJECTIF DU TRAVAIL :

Dans ce projet et à partir des modèles STL des pièces avec des surfaces gauches à ébaucher, nous nous intéressons à l'optimisation de l'opération de Tréflage sur les fraiseuses à commande numérique à 03 axes par la détermination de la combinaison optimale des outils permettant d'enlever le maximum de matière en un minimum de temps tout en évitant les problèmes d'interférences et de collisions, des positions de plongée valides, des profondeurs de plongée, du chemin le plus court reliant les positions de plongée valides et le programme d'usinage « G-Code ». La finalité de ce travail est le développement d'une application logicielle graphique et interactive sous Windows permettant d'automatiser toutes ces tâches.

## III. STRUCTURATION DU MEMOIRE :

Le présent mémoire est composé des chapitres suivants :

- Le premier chapitre est consacré à la présentation du processus d'ébauchage et de Tréflage des surfaces gauches et le format d'échange de données « STL ».
- L'étude conceptuelle de notre application logicielle est menée dans le deuxième chapitre.
- L'application développée, les fonctions et les algorithmes utilisés pendant le développement informatiques sont présentés dans le troisième chapitre.
- Le dernier chapitre présente les tests et la validation des résultats.

**CHAPITRE I**

**GENERALITES SUR LE  
PROCESSUS D'USINAGE  
DES SURFACES GAUCHES**

## I. INTRODUCTION :

Dans la pratique industrielle, pour concevoir de nouveaux produits, l'ingénieur concepteur peut choisir un logiciel de CAO parmi un nombre très important de logiciels du marché. Une fois la phase de conception est terminée, une deuxième phase commence et qui consiste à analyser, étudier les performances, déterminer les contraintes maximales admissibles, planifier les trajectoires d'usinage, ...etc. Ces études nécessitent généralement l'utilisation de logiciels spécifiques. Le problème qui se pose est l'échange des informations de conception entre ces logiciels. D'où la nécessité de développer des outils permettant de réaliser ces échanges dans un format neutre indépendant des logiciels. Pour cela, plusieurs formats d'échange de données sont créés tels que IGES, STL, VRML, DWG, DXF, STEP, ...etc. Le domaine de la conception et de la fabrication des pièces mécaniques avec des surfaces gauches est un domaine très complexe qui doit répondre aux différents paramètres lors de l'usinage comme le respect de la forme et des contraintes fonctionnelles et/ou esthétiques. En raison de la complexité géométrique de ces pièces, les différentes opérations d'usinage sont effectuées sur des machines-outils à commande numérique. La forme finale des surfaces gauches est obtenue en trois étapes : ébauchage, demi-finition et finition. Pour l'ébauchage, différentes stratégies peuvent être utilisées. Une nouvelle technique d'ébauchage appelée « Tréflage » a fait son apparition pour remédier aux inconvénients des autres stratégies d'ébauchage.

Ce chapitre est organisé en deux parties. Dans la première partie nous allons étudier le format d'échange de données « STL ». Dans la deuxième nous allons présenter le processus d'usinage des surfaces gauches tout en mettant l'accent sur la nouvelle stratégie d'ébauchage le « Tréflage ».

## II. FORMAT STL :

### II.1. Définition [1]:

Par définition, un modèle « STL » (Stéréolithography Tessellation Language) est une représentation des formes tridimensionnelles par des facettes triangulaires dont la taille et le nombre de ces facettes sont déterminés par une tolérance appliquée à une surface d'origine ou à un modèle solide (Figure 1). Donc, le fichier contenant le modèle « STL » est composé d'un ensemble de triangles où chaque triangle est défini par les coordonnées X, Y et Z de ses trois sommets et par les composantes du vecteur normal  $\vec{N}$  du triangle qui indique de quel côté est situé l'extérieur (Figure 2).

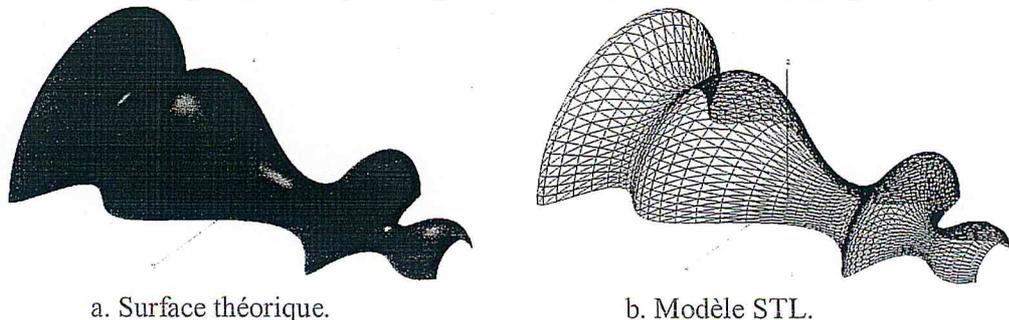
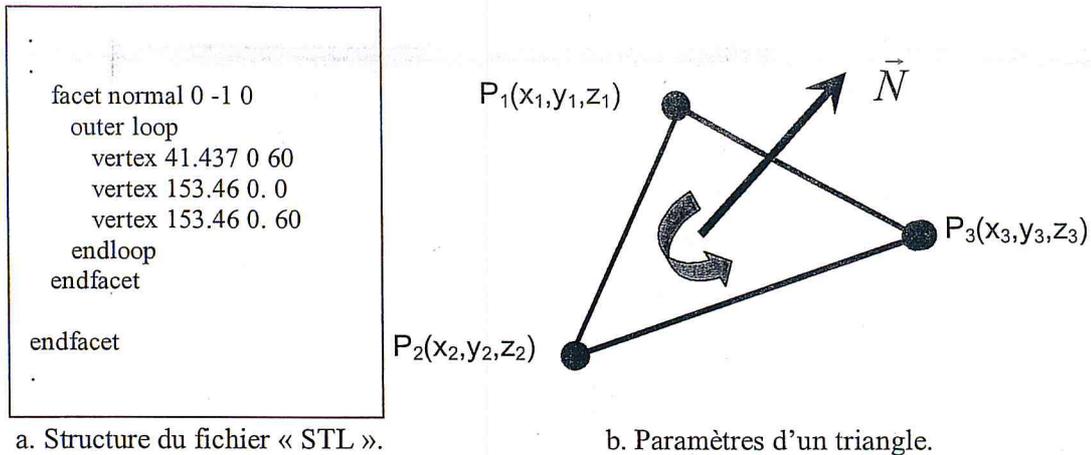


Figure 1 : Format STL d'une surface théorique.



**Figure 2 :** Fichier « STL » et paramètres d'un triangle.

## II.2. Caractéristiques du fichier STL :

Les principales caractéristiques du fichier STL sont les suivantes :

- Format de très faible niveau puisqu'il ne contient aucune notion de couleur, de texture ou d'autres attributs.
- Approximation de la frontière par des facettes triangulaires.
- Le fichier « STL » est volumineux. Le modèle CAO est remplacé par un nombre important de triangles dont le nombre dépend de la précision désirée, de la forme de l'objet et de ses dimensions.
- Le fichier « STL » ne contient pas d'information d'échelle : les coordonnées sont dans des unités arbitraires.

## III. MACHINE OUTIL A COMMANDE NUMERIQUE (MOCN) [2] :

Une machine-outil à commande numérique (MOCN) c'est une machine-outil programmable équipée d'une commande numérique par ordinateur (CNC). Une MOCN assure la réalisation automatisée des pièces et les mouvements nécessaires sont décrits dans un programme. Ces machines sont composées essentiellement de deux parties complémentaires :

- Partie opérative : composée des organes agissant directement sur le produit final.
- Partie commande : contrôle la machine en envoyant des consignes et récupère des informations sur le déroulement des opérations.

Si la machine-outil à commande numérique est équipée d'un magasin d'outils et d'un changeur automatique d'outils, alors est appelée un centre d'usinage.

### III.1. FONCTIONS D'UNE MOCN [3] :

Les principales fonctions d'une MOCN sont les suivantes :

- Réaliser physiquement les mouvements de coupe et d'avance nécessaires à l'obtention d'une surface par l'enlèvement de matière.

- Obtenir des pièces respectant les spécifications fonctionnelles.
- Assurer la réalisation automatisée des pièces de différentes formes en décrivant les mouvements d'outils dans un programme d'usinage.

### III. FRAISEUSE [3]

#### III.1. Définition:

La fraiseuse est un des types des MCON qui sert principalement à usiner des pièces prismatiques et des pièces avec des formes gauches où l'outil nommé « fraise » est animé d'un mouvement de rotation (mouvement de coupe) et la pièce est animée d'un mouvement de translation (mouvement d'avance).

#### III.2. Types de fraiseuses :

Les fraiseuses sont classifiées selon l'axe de la broche comme suit :

- Fraiseuse horizontale : l'axe de la broche est horizontal (Figure 3.a).
- Fraiseuse verticale : l'axe de la broche est verticale (Figure 3.b). Ce type de fraiseuses est utilisé dans l'usinage des surfaces gauches.

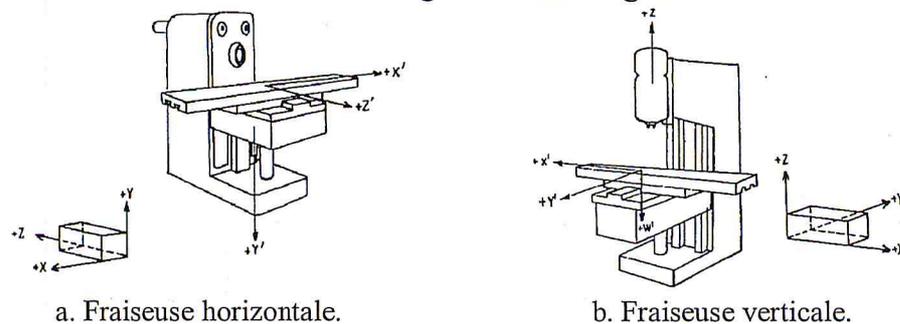


Figure 3 : Types de fraiseuses.

#### III.3. Avantages des fraiseuses à commande numérique [3] :

Parmi les avantages des fraiseuses à commande numérique :

- Fonctionnement sans aucune assistance pendant leur cycle d'usinage, laissant l'opérateur libre d'accomplir d'autres tâches en dehors du poste de travail.
- Changement facile du programme d'usinage des pièces.
- Réduction des temps de préparation et de réglage du poste de travail.
- Prise en compte rapide des modifications d'usinage.
- Définition plus rapide et plus fiable des conditions optimales d'usinage.
- Réalisation des pièces complexes.

### IV. USINAGE DES SURFACES GAUCHES [4] :

La complexité géométrique des surfaces gauches rend leur usinage très difficile. Pour cela, leur fabrication passe par plusieurs étapes en considérant les formes et les dimensions des outils, les contraintes géométriques, le débit de matière et la précision. La fabrication d'une pièce, passe par trois étapes :

### IV.3. Ebauchage des surfaces gauches :

L'ébauchage c'est l'étape de dégrossissage de la matière sans tenir en compte ni de la qualité d'usinage ni de la forme mais ce qui importe le plus est l'enlèvement du maximum de matière en un temps réduit. L'ébauchage avec un outil cylindrique est une série d'évidement de poches dans les différents plans de coupe (plans 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. Le meilleur moyen d'usinage serait de le faire dans un mode continue (nombre minimum de dégagements d'outils) en commençant du haut vers le bas (Figure 6).

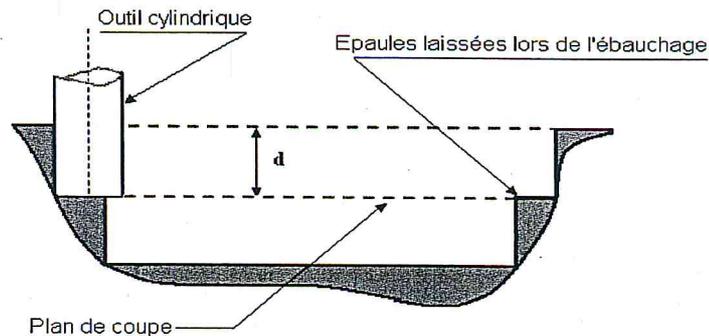
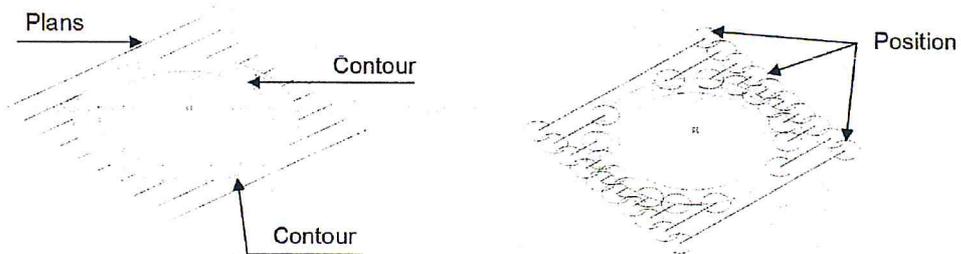


Figure 6 : Ebauchage avec un outil cylindrique.

#### IV.3.1. Stratégies d'ébauchage :

##### IV.3.1.1. Stratégie des « Plans Parallèles » :

Cette stratégie comporte deux étapes. La première étape consiste à calculer les contours d'intersection entre des plans horizontaux et la surface à usiner. La deuxième étape consiste à calculer les intersections entre les contours et des plans parallèles et verticaux ayant une certaine orientation par rapport au plan XZ (Figure 7.a). Le trajet d'usinage est calculé à partir des segments d'intersection (Figure 7.b).



a. Contours et segments d'intersection.

b. Trajet d'usinage.

Figure 7 : Stratégie des « Plans Parallèles ».

Pour cette stratégie, deux modes de balayage de l'outil sont possibles :

- **One-Way** : l'outil suit une droite jusqu'à sa fin. Par la suite, l'outil quitte la surface et reprend l'usinage au début de la droite suivante (Figure 8.a).
- **Zig-Zag** : l'outil suit une droite jusqu'à sa fin. Par la suite, l'outil se déplace à la fin de la droite suivante sans quitter la surface et reprend l'usinage en sens inverse (Figure 8.b).



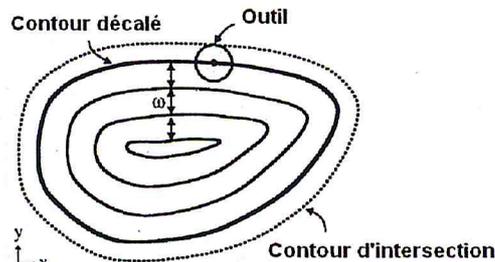
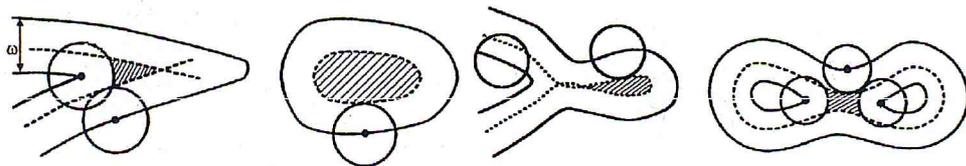
a. One-Way.

b. Zig-Zag.

**Figure 8 :** Modes de balayage de l'outil.

#### IV.3.1.2. Stratégie des « Contours Décalés » :

Cette stratégie comporte deux étapes. La première étape consiste à calculer les contours d'intersection entre des plans horizontaux et la surface à usiner. La deuxième étape consiste à calculer les contours décalés successifs d'une certaine distance «  $\omega$  » fixée par l'utilisateur et qui doit être inférieure au rayon de l'outil (Figure 9). Le trajet d'usinage est généré à partir des différents contours décalés. Son inconvénient est la possibilité d'existence de zones non usinées si la distance de décalage est plus grande que le rayon de l'outil ou si le contour est très courbé (Figure 10) et le calcul des contours décalés successifs est très difficile.

**Figure 9 :** Stratégie des « Contours Décalés ».**Figure 10 :** Zones non usinées avec les « Contours Décalés ».

Pour minimiser le temps d'ébauchage, une nouvelle stratégie a été développée « Tréflage ».

#### IV.3.1.3. Stratégie de « Tréflage » [5] :

Le « Tréflage » est une opération d'usinage utilisée dans le fraisage, essentiellement pour les opérations d'ébauche. Le principe de cette opération est fondé sur l'enlèvement de matière en ayant des avances selon la direction de l'axe de l'outil principalement. La fraise travaille, comme en perçage, suivant l'axe Z de la machine. Cette opération se justifie particulièrement lorsque la hauteur de l'usinage est importante. Le tréflage est apprécié pour réaliser des cavités profondes des moules et des matrices et permet de réduire les vibrations en sollicitant l'axe le plus rigide de la machine à savoir l'axe Z (Figure 11).

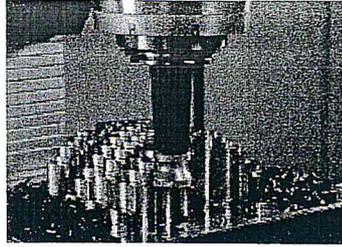


Figure 11 : Opération de Tréflage.

#### IV.3.1.3.1. Outils en Tréflage :

Les outils de Tréflage peuvent être classés de plusieurs manières (monoblocs, brasés, indexés ou à plaquettes rapportées, nombre de tailles, montage, ...etc.). La première spécificité de ces fraises porte sur la coupe dite centrale pour effectuer des plongées en pleine matière. L'outil de Tréflage est composé d'une porte plaquette montée sur un système d'attachement. Ce système permet de positionner l'outil dans la broche de la machine (Figure 12). Les caractéristiques comme le diamètre, la longueur, le pas, le nombre de dents, la géométrie des plaquettes permettent de décrire l'outil de Tréflage :

- Le choix du diamètre d'une fraise est guidé, en règle générale par les dimensions de la pièce.
- Le Tréflage n'est limité en profondeur que par le corps et les rallonges de la fraise utilisée.
- Le pas d'une fraise est la distance entre deux points correspondants sur deux arêtes successives. Une fraise est caractérisée par le nombre de dents ( $Zn$ ).
- La géométrie de la partie active doit être définie pour assurer correctement un enlèvement de matière.

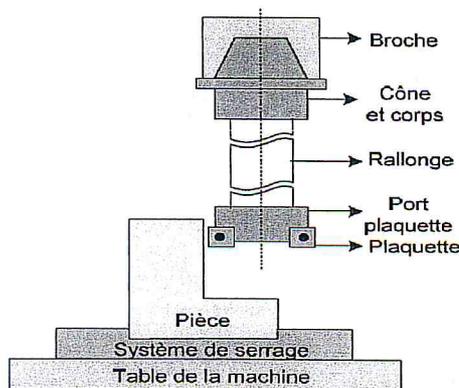


Figure 12 : Montage de Tréflage.

#### IV.3.1.3.2. Paramètres limites de coupe [6-7] :

Les paramètres limites de coupe en Tréflage sont les suivants (Figure 13) :

- **Engagement radial ( $a_e$ )** : exprimé en mm est la distance d'engagement de l'outil dans la pièce. Il est limité par la dimension de la plaquette utilisée. Il est déterminé en fonction de la surépaisseur d'usinage, de la géométrie de la pièce, de la puissance et de la rigidité de la machine et de l'outil.

- **Épaisseur de coupe (h)** : exprimée en mm est l'épaisseur axiale de copeau. Elle est constante, à l'exception lors de l'entrée de la fraise dans la pièce, elle passe de zéro à la valeur constante et inversement en sortie de la pièce.
- **Pas radial (P)** : exprimé en mm est la distance entre deux plongées de la fraise dans la matière. Plus le pas radial augmente, plus la longueur d'arête en contact avec la matière augmente. Le pas radial est un paramètre significatif influençant la topographie de la surface.
- **Avance par dent (f<sub>z</sub>)** : exprimée en mm/dent est la distance linéaire parcourue par une dent de l'outil. Elle représente aussi la distance couverte entre la pénétration de deux dents successives dans la pièce. La vitesse d'avance (V<sub>f</sub>) est exprimée en fonction du nombre de dents Z<sub>n</sub>, de l'avance par dent f<sub>z</sub> et de la vitesse de rotation de la fraise (N) et par :

$$V_f = Z_n f_z N$$

- **Vitesse de coupe (V<sub>c</sub>)** : exprimée en m/min est la vitesse périphérique à appliquer à l'arête de coupe quand celle-ci travaille la pièce. Elle dépend des matériaux de la pièce et de l'outil et de l'opération.

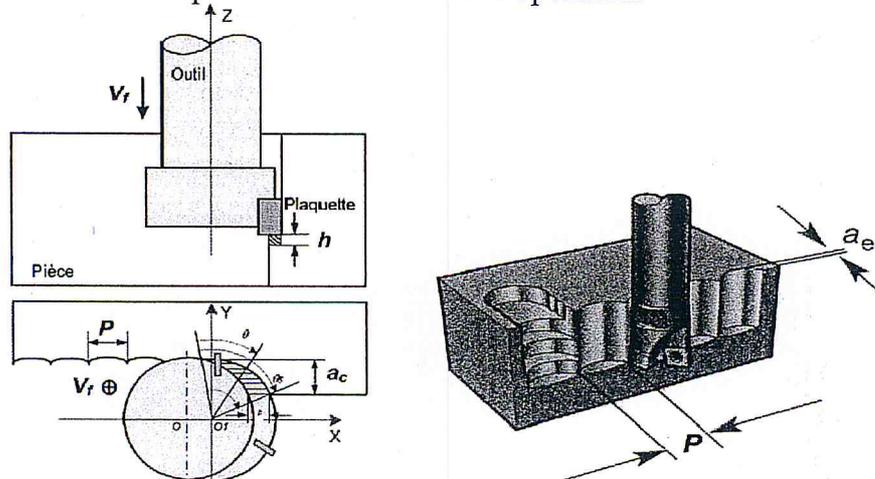
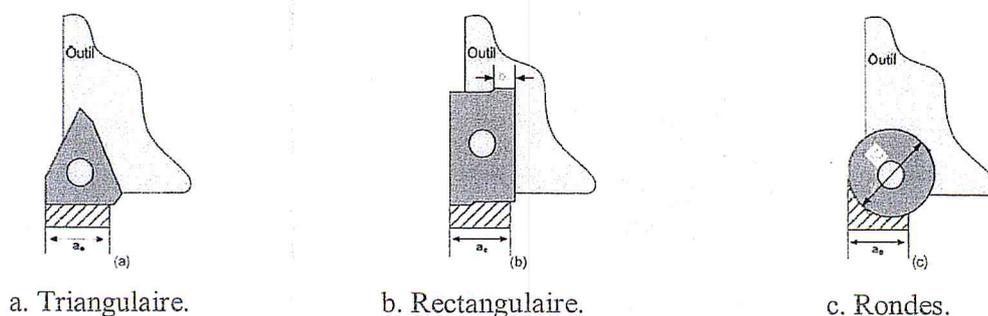


Figure 13 : Paramètres limites de coupe.

#### IV.3.1.3.3. Types de géométrie de plaquettes :

Les plaquettes employées en Tréflage peuvent être rondes, rectangulaires (carrées) et triangulaires (Figure 14). Les plaquettes rectangulaires et carrées sont les plus utilisées en Tréflage.



a. Triangulaire.

b. Rectangulaire.

c. Rondes.

Figure 14 : Types de plaquettes.

#### IV.3.1.3.4. Paramètres pilotant l'opération de Tréflage [6-7] :

Les principaux paramètres pilotant l'opération de Tréflage sont :

- Paramètres limites : engagement radial, épaisseur de coupe, pas radial, avance par dent, vitesse de coupe.
- Paramètres de liaison : débit de copeaux, section du copeau, ...etc.
- Paramètres auxiliaires : énergie spécifique de coupe, puissance de coupe, efforts de coupe, ...etc.) (Figure 15).

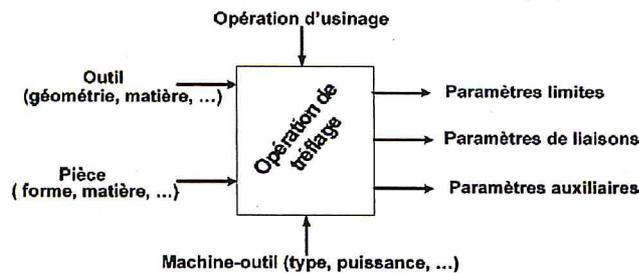


Figure 15 : Paramètres pilotant l'opération de Tréflage.

#### IV.3.1.3.5. Mouvements de l'outil [8, 10, 11] :

Lors de l'usinage, la course de l'outil peut être descendante ou ascendante dans la pièce, bien que l'usinage en descendant soit le plus courant. Le nom d'usinage en plongée est souvent adopté pour désigner l'usinage par Tréflage. La cinématique de l'opération de Tréflage est caractérisée par des mouvements spécifiques de l'outil. Pour enlever de la matière en Tréflage, deux mouvements sont définis (Figure 16) :

- Mouvement de coupe : développé par la broche et qui correspond à la rotation de l'outil autour de son axe.
- Mouvement d'avance : qui correspond au déplacement de l'outil parallèlement à son axe (suivant l'axe Z).

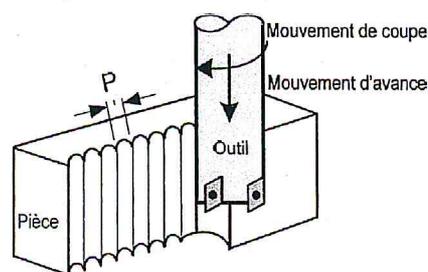


Figure 16 : Mouvements de l'outil en Tréflage.

#### IV.3.1.3.6. Mouvement de coupe :

Le mouvement de coupe est le mouvement principal transmis à l'outil par la broche de la machine-outil. Il est caractérisé par la direction et la vitesse de coupe :

- Direction de coupe : direction instantanée du mouvement de coupe du point considéré de l'arête par rapport à la pièce.
- Vitesse de coupe ( $V_c$ ) : exprimée en m/min est la vitesse instantanée du mouvement de coupe du point considéré de l'arête par rapport à la pièce.

#### IV.3.1.3.7. Mouvement d'avance :

C'est le mouvement relatif entre l'outil et la pièce nécessaire à la génération de la surface de la pièce. Il peut être composé de plusieurs mouvements mais seulement de façon à ce qu'au moins une de ses composantes soit rectiligne. Le mouvement d'avance est caractérisé par la direction et la vitesse d'avance ( $V_f$ ) :

- Direction d'avance : direction instantanée du mouvement d'avance du point considéré de l'arête par rapport à la pièce.
- Vitesse d'avance  $V_f$  : exprimée en mm/min est la vitesse instantanée du mouvement d'avance du point considéré de l'arête par rapport à la pièce.

#### IV.3.1.3.8. Mouvement résultant de coupe :

Le mouvement de coupe et le mouvement d'avance combinés constituent le mouvement résultant de coupe ( $V_e$ ). La vitesse résultante de coupe ( $V_e$ ) est la somme vectorielle de la vitesse de coupe ( $V_c$ ) et de la vitesse d'avance ( $V_f$ ) :

$$\vec{V}_e = \vec{V}_c + \vec{V}_f$$

#### IV.3.1.3.9. Trajectoire de la partie active de la fraise en Tréflage [7] :

La trajectoire d'une dent dans la matière pour le Tréflage est une courbe spirale définie par le mouvement d'un point fixe sur l'arête de coupe (Figure 17). L'équation qui définit cette trajectoire en Tréflage est donnée par :

$$\vec{om} = R \cos(2\pi Nt) \vec{X} + R \sin(2\pi Nt) \vec{Y} + Nht \vec{Z}$$

Les paramètres intervenant dans cette équation sont : le rayon de la fraise ( $R$ ), l'épaisseur axiale du copeau ( $h$ ) ainsi que la fréquence de rotation ( $N$ ).

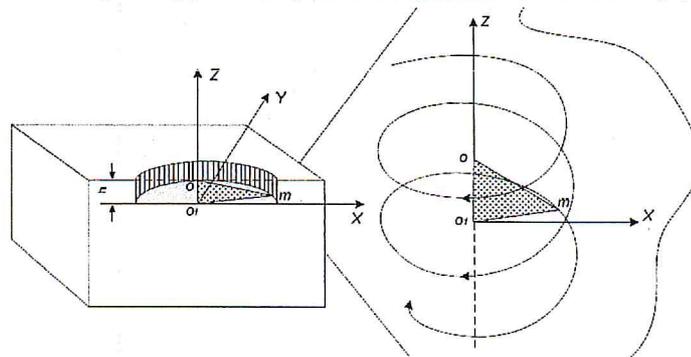


Figure 17 : Trajectoire d'une dent dans la matière.

#### IV.3.1.3.10. Trajectoires d'usinage [8, 10, 11] :

Une bonne génération de la trajectoire de l'outil permet de minimiser son usure et par conséquent d'augmenter sa durée de vie. De manière générale, un trajet d'usinage est un ensemble de mouvements de type « Approcher-Usiner-Dégager ». Les actions d'approcher et de dégager correspondent à des trajectoires hors matière pouvant être linéaires ou non. L'action d'usiner comporte une garde d'engagement,

la coupe proprement dite et une garde de dégagement (Figure 18). L'approche est un mouvement de l'outil en vitesse rapide dans l'espace. Les contraintes essentielles à ce mouvement sont principalement d'éviter les collisions « outil / pièce » ou encore « outil / porte-pièce ». Le dégagement est un mouvement spécifique respectant les mêmes principes que l'approche. En Tréflage, trois modes de dégagement de l'outil en fin de plongée sont possibles (Figure 18) : 1) dégagement de l'outil selon l'axe Z, 2) dégagement de l'outil de manière incliné et 3) dégagement de l'outil à l'aide de deux demi-cercles. Le dégagement selon l'axe Z demande des performances spécifiques à la machine (arrêt en fin de plongée et changement de direction) et à l'outil (résistance au choc). Le dégagement incliné et celui en demi-cercles permettent une remontée d'outil sans contact avec la surface usinée.

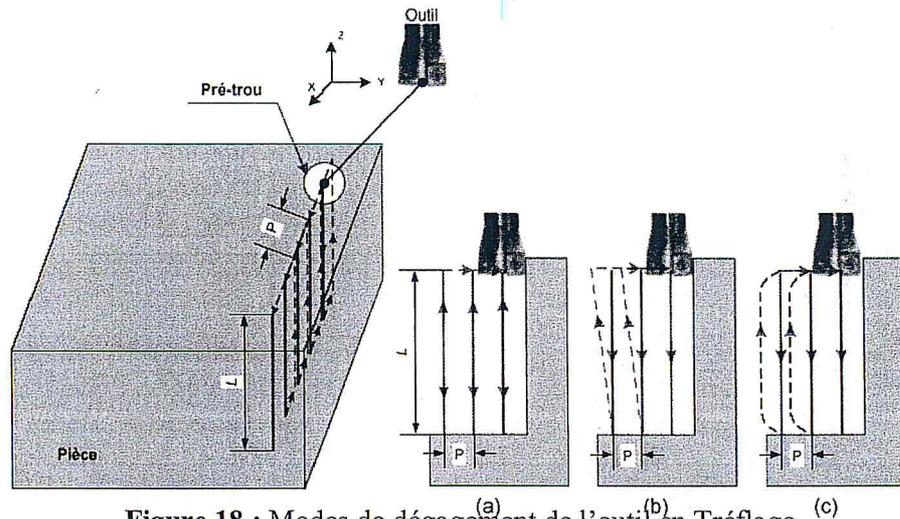


Figure 18 : Modes de dégagement de l'outil en Tréflage.

#### IV.3.1.3.11. Modes de balayage de l'outil [10] :

Consiste à proposer les trajectoires successives suivies par l'outil pour usiner partiellement ou entièrement le volume de matière. Les principaux objectifs d'une stratégie d'usinage efficace sont le respect des conditions d'usinage, de la qualité de surface et la productivité. En Tréflage, le choix d'une stratégie d'usinage tient compte du type d'outil utilisé, de la variation d'engagement du pas radial et des contraintes technologiques. Les différentes stratégies utilisées en Tréflage sont (Figure 19) :

- « Zig-Zag » : le déplacement de l'outil suit une direction de forme rectangulaire parallèle à la surface. Les trajectoires sont construites à partir de segments de droites parallèles entre elles.
- « Contour Parallèle » : la direction de déplacement de l'outil correspond à des contours parallèles au contour de la poche. Les contours de la poche sont utilisés pour construire des trajectoires décalées.
- « Spirale » : la direction de déplacement de l'outil correspond à une opération d'usinage homothétique au contour de la poche. L'outil peut parcourir la poche de l'intérieur vers l'extérieur ou inversement.

- « Spirale-Contour parallèle » : la direction de déplacement de l'outil correspond au type « Spirale » d'une partie en suivant un « Contour Parallèle ».
- « One-Way » : le déplacement de l'outil suit une seule direction parallèle à la surface usinée.
- « Zig-Zag-Triangle » : le déplacement de l'outil suit une direction de forme triangulaire imposée.

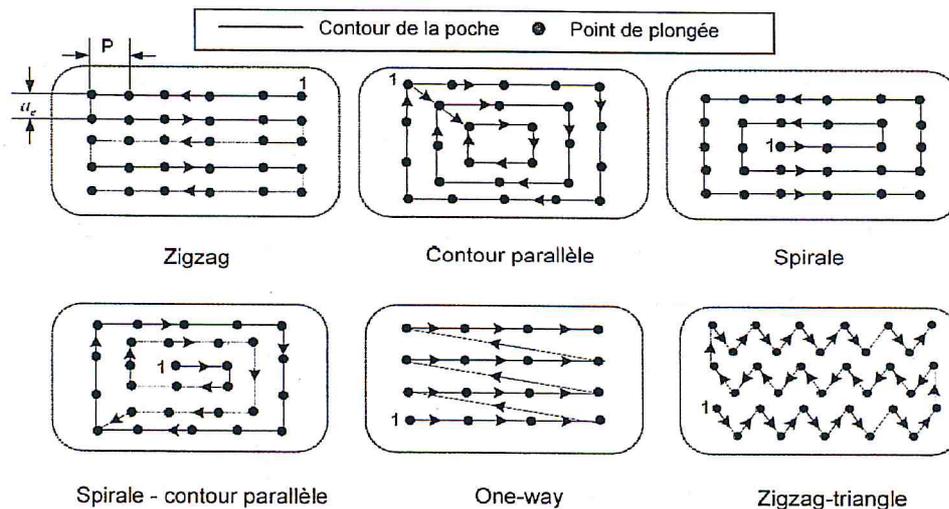


Figure 19 : Modes de balayage de l'outil.

#### IV.3.1.3.12. Avantages du Tréflage [7, 11] :

- Débit de copeau élevé, incomparable aux autres méthodes d'usinage.
- Réduction de la charge appliquée à la broche de la machine-outil.
- Limitation de l'effort radial.
- Utilisation des fraises de longueur supérieure au fraisage classique.
- Sens de travail principalement en poussant ou en tirant suivant l'axe Z.
- Diminution de la déflexion de l'outil et amélioration de la qualité des surfaces.

#### V. CONCLUSION :

Dans ce chapitre nous avons étudié le format d'échange de données « STL », l'opération d'ébauchage des surfaces gauches et les stratégies utilisées. Ensuite, nous avons mis l'accent sur l'opération de Tréflage tout en mettant en évidence les différents paramètres à considérer lors de la génération du trajet de Tréflage.

Les chapitres suivants seront consacrés à la présentation de la problématique, de la conception et des différentes solutions proposées.

**CHAPITRE II**

**CONCEPTION DE**

**L'APPLICATION**

**LOGICIELLE**

## **I. INTRODOCTION :**

Dans le chapitre précédent, après avoir étudié le format d'échange de données « STL » et les stratégies d'ébauchage des surfaces gauches et en particulier la stratégie du Tréflage, nous passons à l'étape suivante c'est l'étape de conception où nous allons formuler les objectifs et les besoins de notre système en utilisant le modèle orienté objet et le langage de modélisation UML (Unified Modeling Language).

## **II. METHODE DE MODELISATION :**

Avant de passer à l'implémentation informatique d'une application logicielle en utilisant un langage de programmation, il est nécessaire de passer par une étape de conception. Dans notre travail, la conception a été effectuée en utilisant le langage de modélisation UML (Unified Modeling Language). Ce langage est indépendant du langage de programmation utilisé et permet de spécifier, de construire, de visualiser et de décrire tous les aspects d'un système logiciel. Il est composé d'un ensemble de notations et de diagrammes que nous pouvons utiliser selon nos besoins.

## **III. REALISATION DE L'APPLICATION :**

Pour atteindre notre objectif, nous devons passer par les étapes suivantes :

### **III.1. Cahier de charges :**

#### **III.1.1. Présentation du sujet :**

Notre travail s'insère dans le cadre de développement d'outils de conception et de fabrication des surfaces gauches initié par l'équipe Conception et Fabrication Assistées par Ordinateur « CFAO » de la Division Productique et Robotique « DPR » du Centre de Développement des Technologies Avancées « CDTA ».

L'objectif principal de ce projet consiste à optimiser l'opération de Tréflage des pièces de n'importe quelle forme à partir de leurs modèles STL sur des fraiseuses à commande numérique à 03 axes par la détermination de la combinaison optimale des outils permettant d'enlever le maximum de matière en un minimum de temps tout en évitant les problèmes d'interférences et de collisions, des positions de plongée valides et des profondeurs de plongée associées, du chemin le plus court reliant les positions de plongée valides et en fin la génération du programme d'usinage « G-Code ».

#### **III.1.2. Problématique :**

Les pièces avec des surfaces de formes libres (surfaces gauches) sont utilisées dans la conception et la fabrication des moules, des matrices, des formes esthétiques,...etc. En raison de leurs formes géométriques très complexes, ces pièces sont usinées sur des fraiseuses à commande numérique à 03, à 04 et à 05 axes par la prise en compte de plusieurs paramètres : modèles des surfaces, formes et dimensions des outils, conditions de coupe, problèmes d'interférences et de collisions, ...etc. Pour obtenir la forme finale de ces pièces, trois opérations sont nécessaires : ébauche, demi-finition et finition.

L'opération d'ébauchage consiste généralement à enlever l'excédent de matière dans des plans horizontaux par le choix entre deux stratégies « Plans Parallèles » et « Contours Décalés ». Afin de minimiser les temps d'ébauchage des cavités profondes, une nouvelle stratégie a été développée c'est le «Tréflage» où plusieurs modes de balayage d'outils peuvent être choisis (One-Way, Zig-Zag, Spirale, ...etc.). Cette stratégie nécessite un choix judicieux du nombre et des dimensions d'outils, des modes de balayage des outils, des conditions de coupe, des positions de plongée valides des outils et des profondeurs associées. Le résultat du Tréflage est une géométrie en escaliers d'épaisseurs non uniformes. Cette stratégie devient plus complexe si elle est appliquée sur des pièces avec des surfaces gauches ce qui nécessite le développement de procédures spécifiques dans le but d'automatiser et d'optimiser le Tréflage. Afin d'optimiser cette stratégie, il est nécessaire de choisir judicieusement la séquence des outils à utiliser afin d'enlever le maximum de matière en un minimum de temps tout en évitant les problèmes d'interférences et par la suite déterminer les profondeurs de plongée et le chemin d'outils le plus court. La forme géométrique très complexe des surfaces gauches rend cette tâche très difficile.

### III.1.3. Objectifs du travail :

Notre objectif principal est le développement d'une l'application logicielle graphique et interactive sous Windows permettant d'optimiser l'opération de Tréflage des surfaces gauches à partir de leurs modèles STL par la détermination de la combinaison optimale d'outils et du chemin d'outils le plus court. Cet objectif peut être subdivisé en plusieurs sous objectifs :

**Premier objectif :** consiste à structurer le fichier STL. Cet objectif nécessite le passage par les étapes suivantes :

- Création de la liste des points ;
- Création de la liste des triangles.

**Deuxième objectif :** consiste à déterminer les contours sur la face supérieure de la pièce finie. Cet objectif nécessite le passage par les étapes suivantes :

- Calcul des segments d'intersection entre les triangles et le plan horizontal passant par la face supérieure du brut ;
- Détermination de la normale de chaque segment ;
- Création des contours.

**Troisième objectif :** consiste à identifier les parties à enlever et les parties à ne pas usiner. Cet objectif nécessite le passage par les étapes suivantes :

- Création des « dexels » approximant le modèle « STL » ;
- Détermination du type de chaque « dexel » ;
- Détermination de la pièce finie ;
- Détermination de la matière à enlever.

**Quatrième objectif :** consiste à calculer les rayons d'outils finaux à utiliser. Cet objectif nécessite le passage par les points suivants :

- Calcul du rayon initial ;
- Calcul du rayon intermédiaire ;
- Calcul du rayon final à utiliser ;
- Détermination de la liste des outils à utiliser ;
- Détermination des combinaisons d'outils possibles.

**Cinquième objectif:** consiste à déterminer la combinaison optimale d'outils permettant d'enlever le maximum de la matière en un temps minimum. Cet objectif nécessite le passage par les étapes suivantes :

- Détermination des positions de plongée valides ;
- Calcul de la profondeur de plongée pour chaque position de plongée valide ;
- Calcul du temps et du volume de matière enlevés par chaque outil ;
- Calcul du temps d'usinage et du volume de la matière enlevée pour chaque combinaison d'outils;
- Choix de la combinaison optimale d'outils.

**Sixième objectif:** consiste à optimiser le trajet d'usinage et à générer le programme « G-Code ». Pour atteindre cet objectif, nous devons passer par les étapes suivantes :

- Détermination du trajet d'usinage optimal pour chaque outil ;
- Simulation virtuelle des mouvements des outils ;
- Simulation virtuelle de l'usinage ;
- Génération du programme « G-Code ».

#### III.1.4. Plateforme exigée :

- Système d'exploitation : Windows.
- Recommandations matérielles (configuration minimale nécessaire) :
  1. Une RAM de 1 Go minimum.
  2. Carte accélératrice 3D de 128 Mo compatible OpenGL ou équivalente.
- Logiciels requis :
  1. Builder C++, Version 6.
  2. Bibliothèque graphique OpenGL.

#### III.1.5. Solutions proposées de la problématique :

Pour atteindre notre objectif, il est nécessaire de développer un ensemble d'algorithmes pour réaliser les tâches suivantes :

- Structurer le fichier STL;
- Déterminer les contours de la face supérieur;
- Déterminer la liste des outils à utiliser;
- Déterminer la combinaison optimale d'outils.
- Déterminer les positions de plongée valides et les profondeurs de plongée;
- Optimisation du trajet d'usinage;
- Génération du programme « G-Code ».

#### IV. MODELISATION DE L'APPLICATION EN UML :

Dans ce qui suit, nous allons présenter le fonctionnement de notre système en utilisant les diagrammes suivants du langage de modélisation UML :

- Diagramme de cas d'utilisation.
- Diagrammes de séquence.
- Diagramme de classe.
- Diagramme de collaboration.

##### IV.1. Diagramme cas d'utilisation :

Pour bien exprimer les besoins de notre système, il faut un moyen simple. C'est précisément le rôle du diagramme de cas d'utilisation qui permet de recueillir, d'analyser et d'organiser les besoins et de recenser les grandes fonctionnalités d'un système. Le diagramme cas d'utilisation est caractérisé par :

- **Acteurs** : ce sont des personnes ou systèmes extérieurs à l'application qui interagissent avec lui.
- **Cas d'utilisation**: représentent une fonctionnalité (un objectif à atteindre) du système à construire. Ils sont en relation avec des acteurs et d'autres cas d'utilisation.
- **Relations** : nous avons deux types de relation : Inclusions et Extensions.

Dans ce qui suit, nous allons présenter une vue générale sur les principaux cas d'utilisations de notre système. La Figure 1 représente le diagramme cas d'utilisation global de notre système. Le système est divisé en plusieurs parties :

- Traitement du fichier STL.
- Création des contours de la face supérieure.
- Calcul des rayons d'outils à utiliser.
- Détermination de la combinaison optimale d'outils.
- Détermination du trajet d'usinage.
- Optimisation du trajet d'usinage.

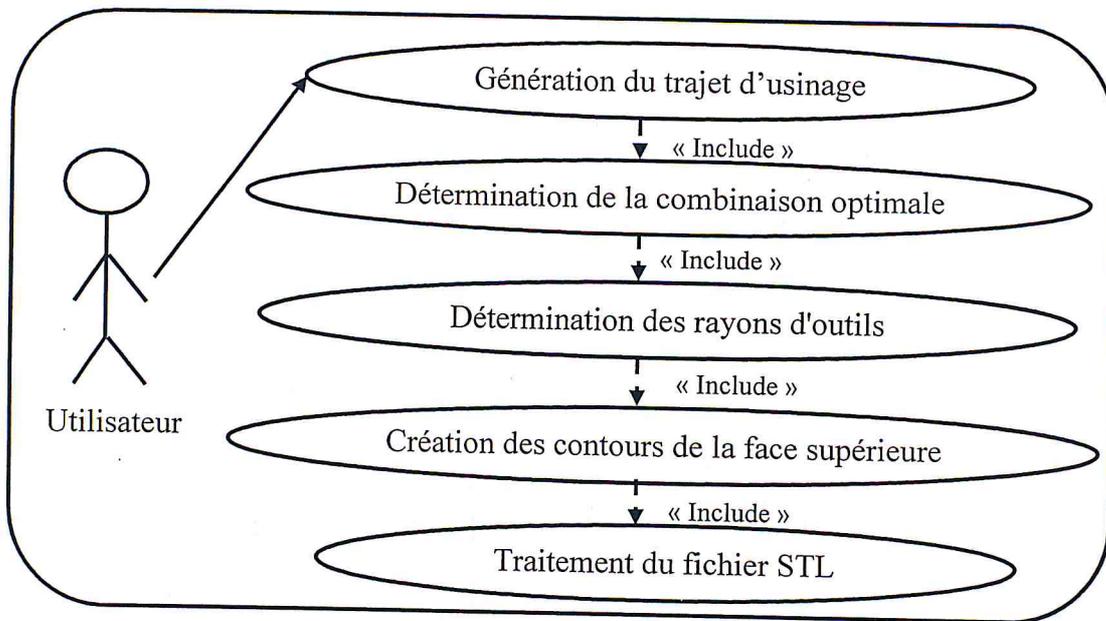


Figure 1 : Diagramme cas d'utilisation global.

Nous allons détailler maintenant les différents cas d'utilisation du diagramme global et nous allons commencer par le traitement du fichier STL (Figure 2) qui consiste à créer une liste de points non redondants et une liste de triangles.

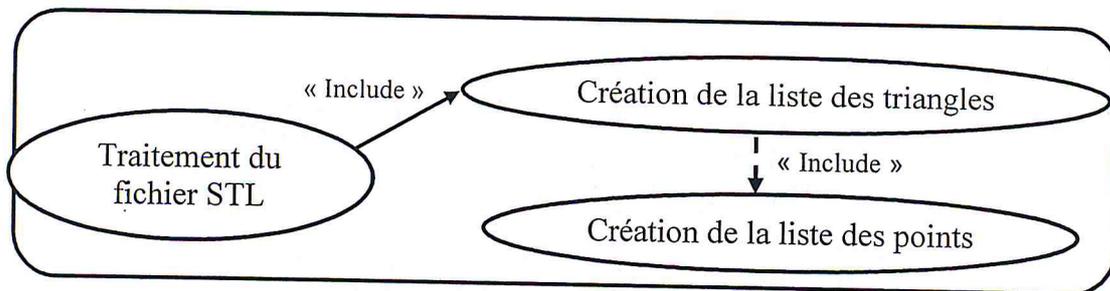


Figure 2 : Diagramme cas d'utilisation du traitement du fichier STL.

Le diagramme cas d'utilisation suivant consiste à déterminer les contours de la face supérieure (Figure 3).

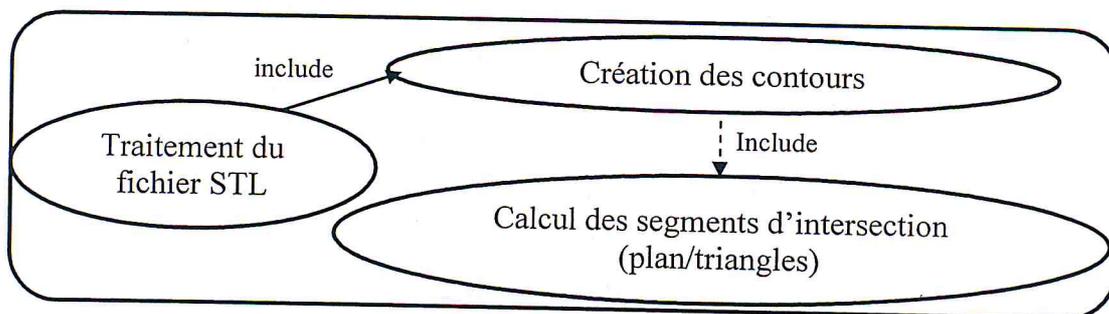


Figure 3 : Diagramme cas d'utilisation du traitement du fichier STL.

Les rayons d'outils finaux qui représentent les outils à utiliser lors de l'usinage de la pièce sont donnés par la Figure 4.

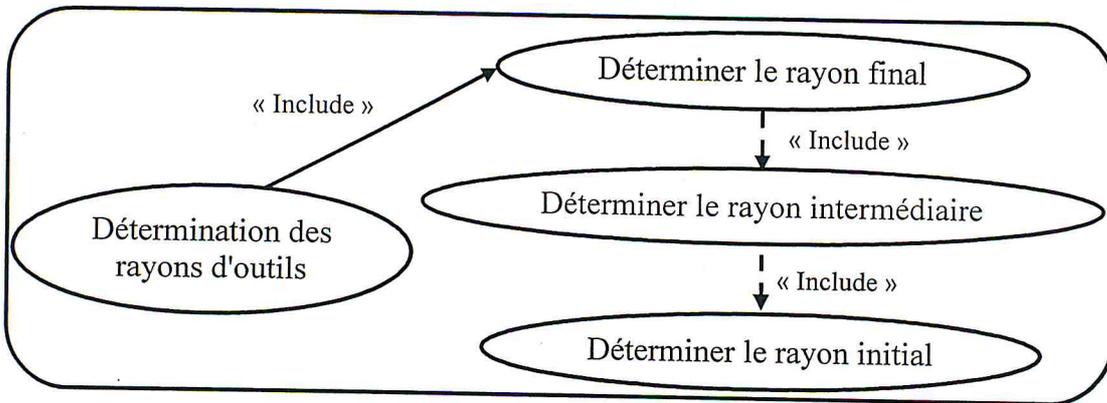


Figure 4 : Diagramme cas d'utilisation de la détermination des rayons finaux.

Le diagramme cas d'utilisation relatif à la détermination de la combinaison optimale d'outils est donné par la Figure 5.

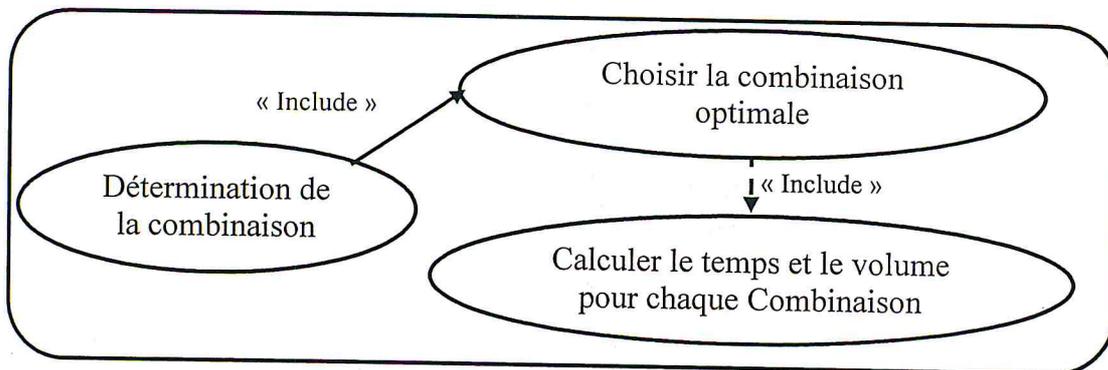


Figure 5 : Diagramme cas d'utilisation de détermination de la combinaison optimale.

Le diagramme cas d'utilisation suivant détermine le trajet d'usinage optimal de l'outil lors de l'usinage qui est donné par la Figure 6.

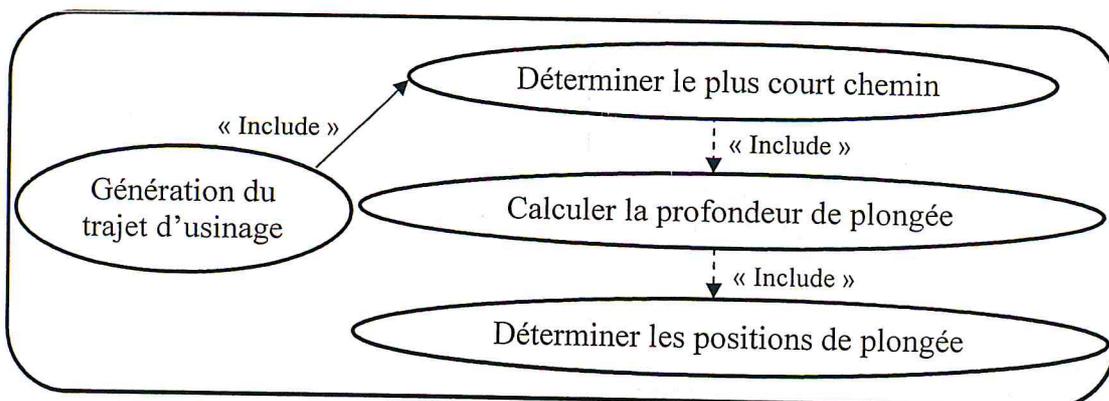


Figure 6 : Diagramme cas d'utilisation de détermination du trajet d'usinage optimal.

#### IV.2. Diagramme de séquence :

Le diagramme de séquence met en évidence l'aspect chronologique de l'envoi des messages « échanges de messages ». Souvent, le diagramme de séquence permet de compléter le diagramme des cas d'utilisation en mettant en évidence les objets et leurs interactions.

Dans ce qui suit, nous allons présenter les différents diagrammes de séquence de notre système à savoir :

- Diagramme de séquence pour exploiter le fichier STL;
- Diagramme de séquence pour générer les contours de la face supérieure;
- Diagramme de séquence pour déterminer les rayons d'outils finaux;
- Diagramme de séquence pour déterminer la combinaison optimale;
- Diagramme de séquence pour déterminer le trajet d'usinage;

Le diagramme de séquence relatif à la structuration du fichier STL est donné par la Figure 7.

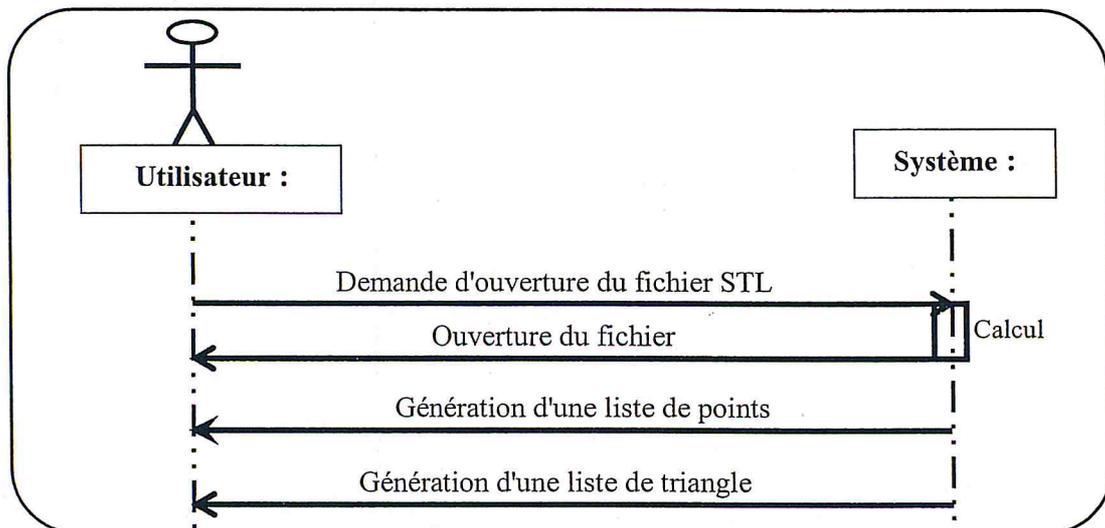


Figure 7: Diagramme de séquence de traitement du fichier STL.

Le diagramme de séquence relatif à la structuration du fichier STL est donné par la Figure 8.

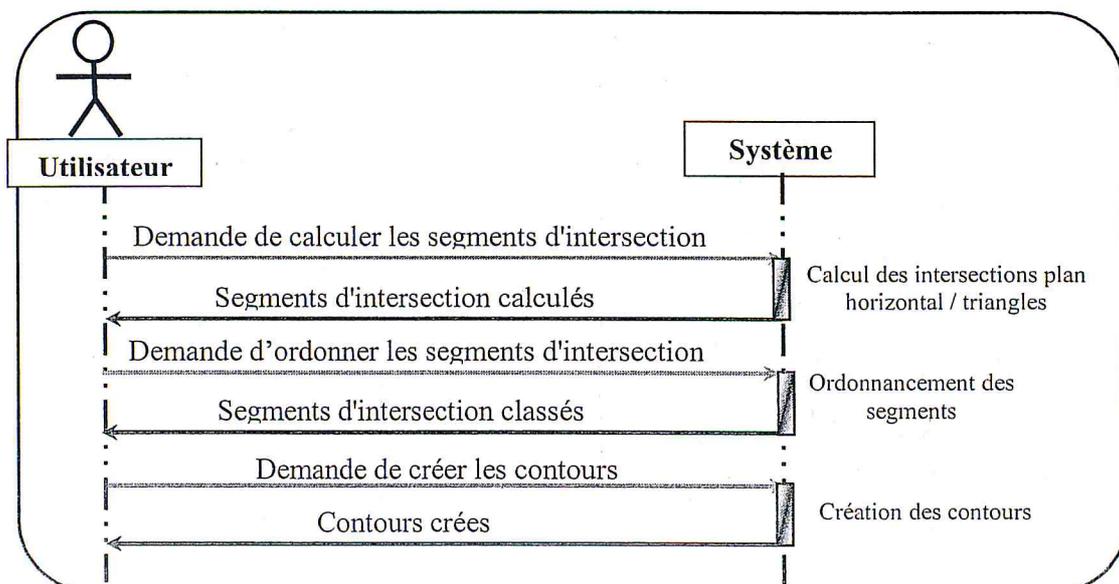


Figure 8: Diagramme de séquence de génération des contours de la face supérieure.

Pour le diagramme de séquence relatif à la détermination des rayons des outils finaux est représenté par la Figure 9.

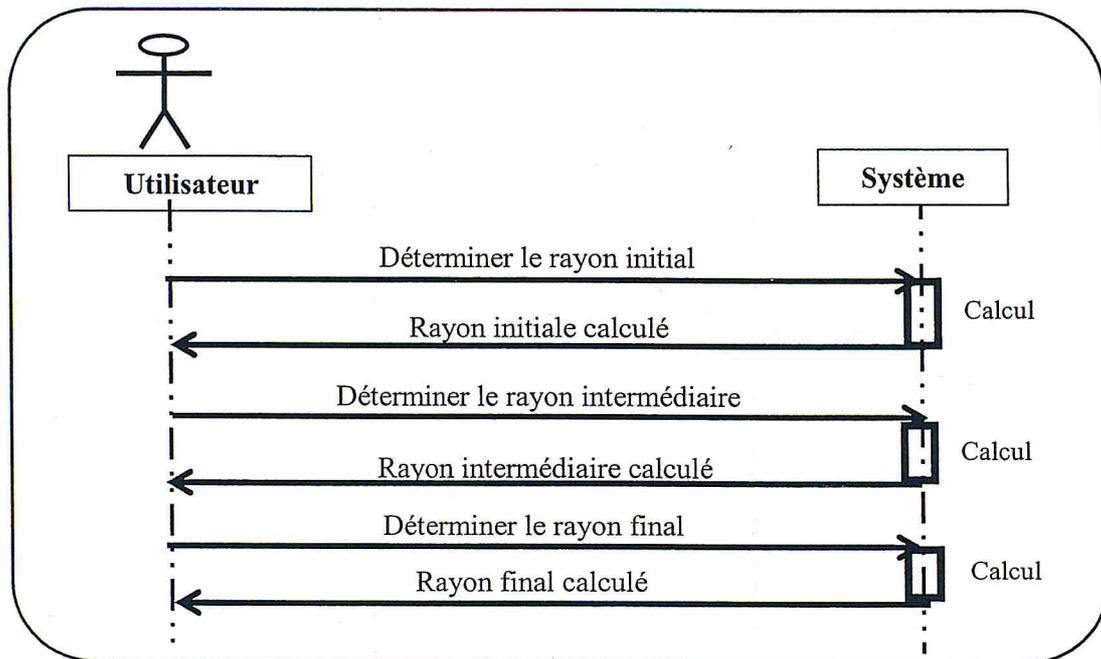


Figure 9 : Diagramme de séquence de calcul des rayons.

Le diagramme de séquence concernant la détermination de la combinaison optimale est donné par la Figure 10.

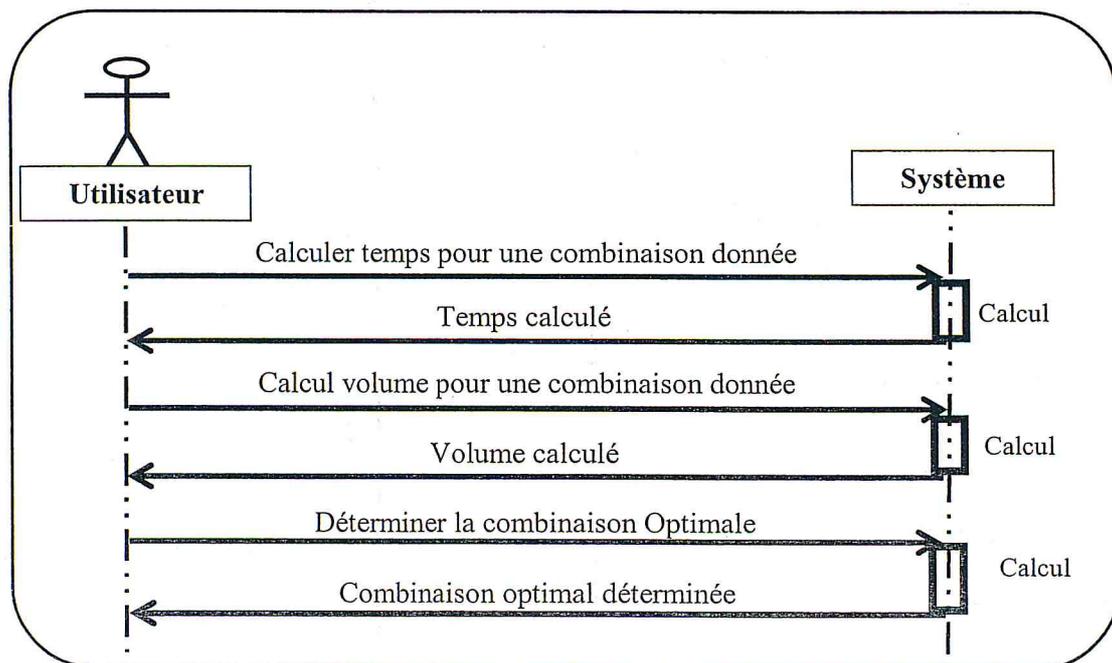


Figure 10 : Diagramme de séquence déterminant la combinaison optimale.

Le diagramme de séquence pour la génération du trajet d'usinage optimal est donné par la Figure 11.

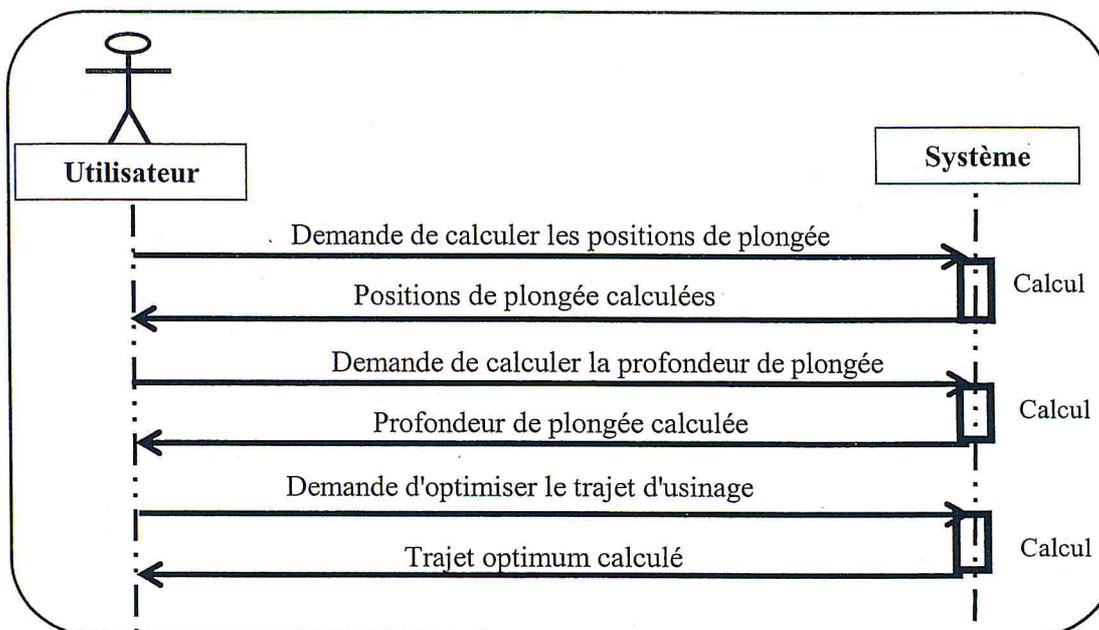


Figure 11 : Diagramme de séquence de la détermination du plus court chemin.

#### IV.4. Diagramme de collaboration :

Les diagrammes de collaboration permettent d'établir un lien entre les différents diagrammes de cas d'utilisation et les diagrammes de classes. Ils montrent comment des objets (des instances de classes) communiquent pour réaliser une certaine fonctionnalité. Ils apportent ainsi un aspect dynamique à la modélisation du système. La Figure 12 représente le diagramme de collaboration de notre système.

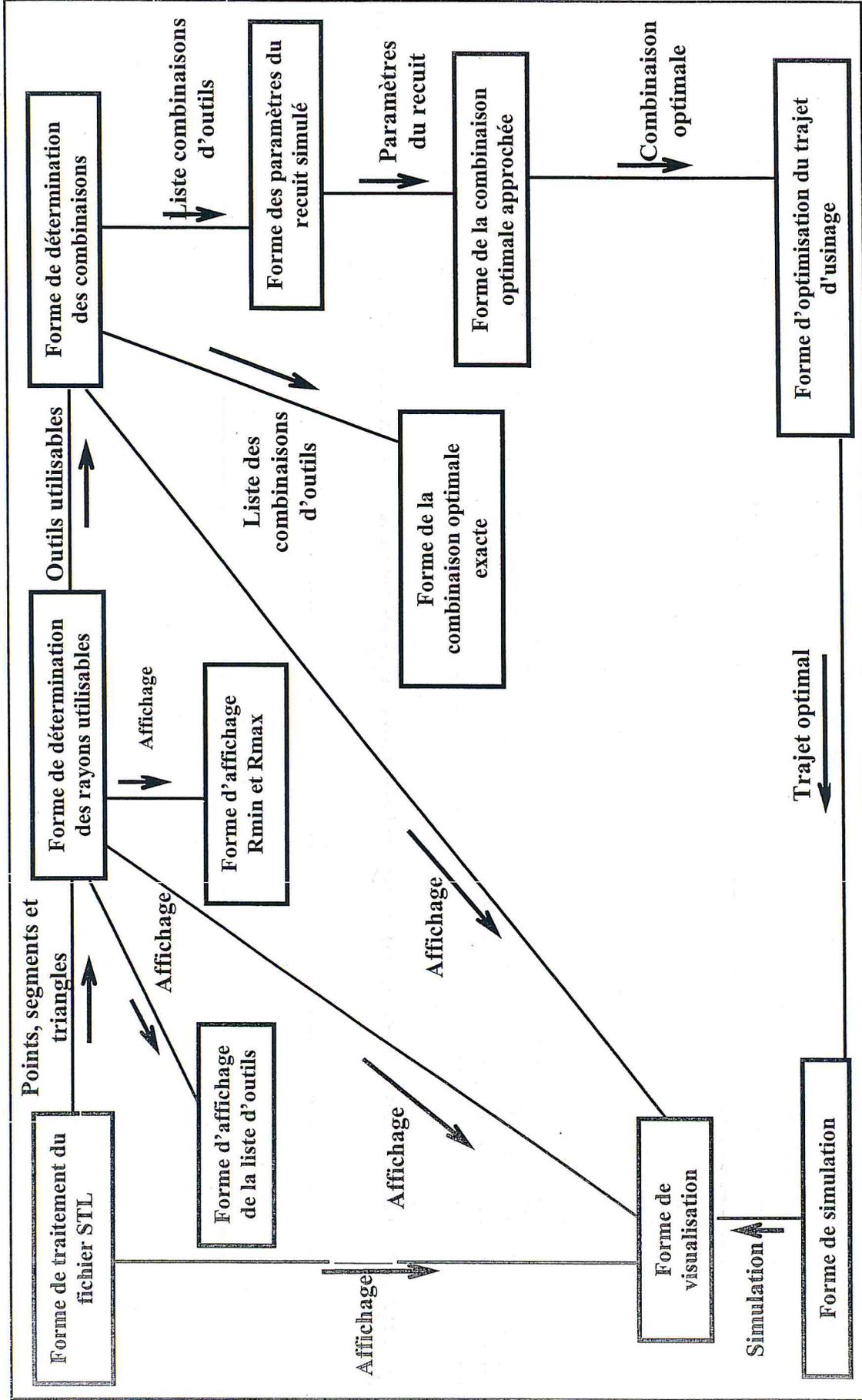


Figure 12: Diagramme de collaboration.

Dans notre système, la forme (**Forme de Traitement du fichier STL**) s'occupe du traitement du fichier STL à savoir la génération de la liste de points et de la liste des triangles ainsi que le calcul des limites du brut....etc. Par la suite, tous ces résultats vont être utilisés pour la génération des contours de la face supérieure.

La forme (**Forme de Traitement du fichier STL**) envoie tous ces résultats à la forme (**Forme Détermination des rayons utilisables**). Cette dernière effectue le calcul des rayons initial, théorique et final. Ensuite, la (**Forme Détermination des rayons utilisables**) envoie la listes des outils utilisables (les rayons des outils finaux) à la forme (**Forme Détermination des combinaisons**) qui effectue le calcul du volume à enlever et détermine la liste des combinaisons possible des outils. Dans notre cas, nous avons utilisé une combinaison de cinq outils. Par la suite, cette dernière envoie les résultats à la forme (**Forme Combinaison optimale exacte**) qui détermine la combinaison optimale.

En parallèle, la (**Forme Détermination des combinaisons**) envoie la liste des combinaisons possibles des outils à la forme (**Forme Paramètres du recuit simulé**) que cette dernière envoie les paramètres du recuit simulé à la forme (**Forme Combinaison optimale approchée**) pour déterminer la combinaison optimale en solution approché afin d'envoyer ce résultat a la forme (**Forme Optimisation de trajet d'usinage**).

Pour déterminer le trajet optimal d'usinage, en utilisant la combinaison optimale approché, le trajet est généré et envoyé à la forme de génération du trajet d'usinage (**Forme de simulation**). La forme (**Forme de simulation**) permet de simuler les mouvements d'outils

Tous les résultats obtenus (points, triangles, normales, segments, rayons, trajet, positions de plongée, ...etc.) sont affichés dans la forme principale d'affichage.

#### IV.5. Diagramme de classe :

Après l'étape de définition et d'analyse des besoins, nous allons passer à l'étape suivante pour montrer les solutions que nous avons proposées pour résoudre la problématique en prenant en considération les besoins définis précédemment et en décrivant l'architecture du système et les relations entre ses différents composants à l'aide du diagramme de classe. Les différentes classes utilisées dans notre conception sont les suivantes :

1. Classe liste\_pt\_eb\_stl.
2. Classe liste\_triangle.
3. Classe cellule.
4. Classe usiner\_dexel.
5. Classe positions\_plongée.
6. Classe interval\_outil.
7. Classe combinaison\_outil.
8. Classe liste\_contour\_eb\_stl.
9. Classe centre\_rayon.
10. Classe list\_pt\_plongee.
11. Classe liste\_pt\_stl.
12. Classe noeud\_triangle.

13. Classe triangle.
14. Classe nœud\_pt.
15. Classe neuod\_segment.
16. Classe contour\_new.
17. Classe contour\_eb\_stl .

La Figure 13 montre les relations qui existent entre les classes de notre system.

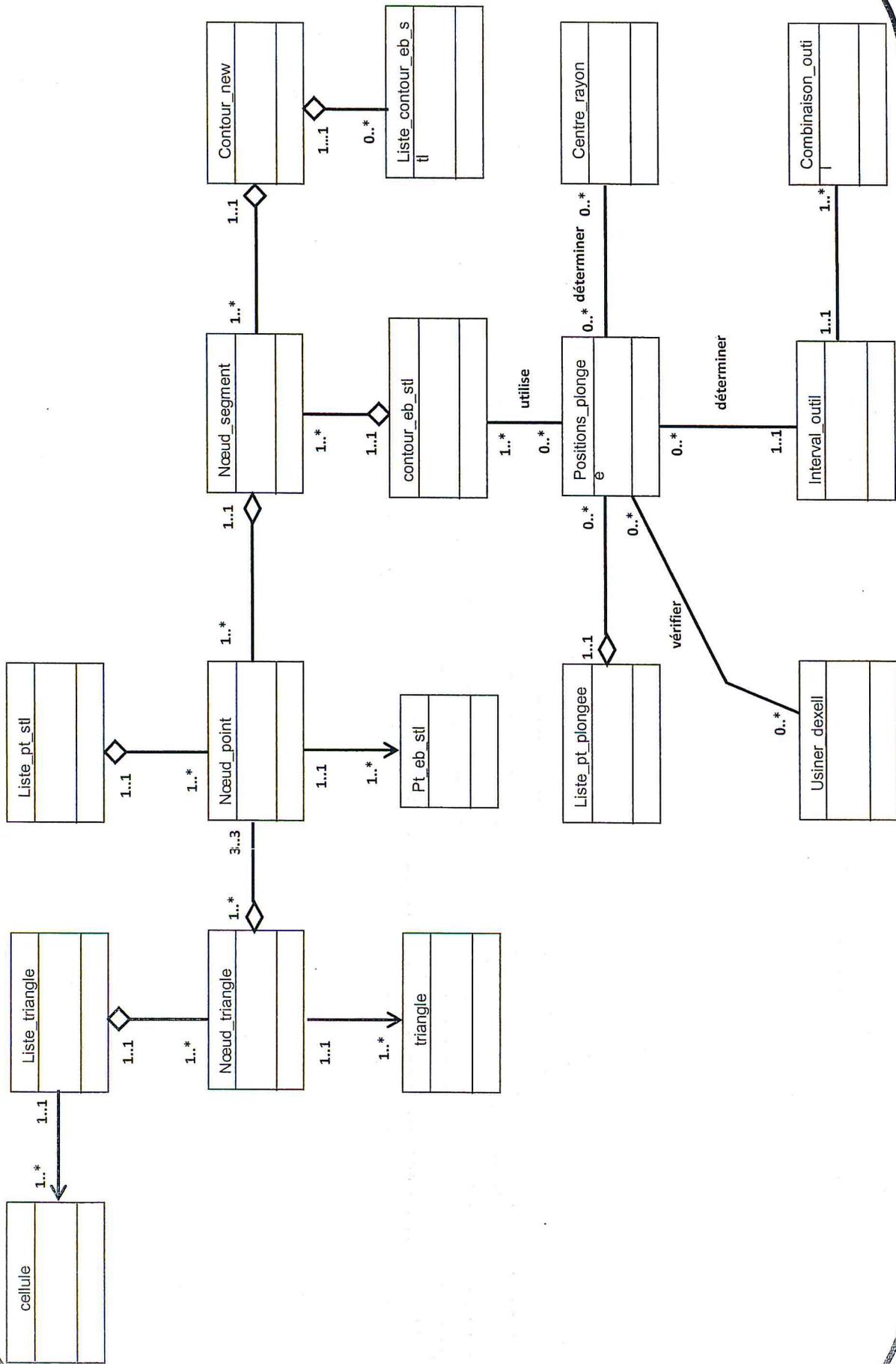
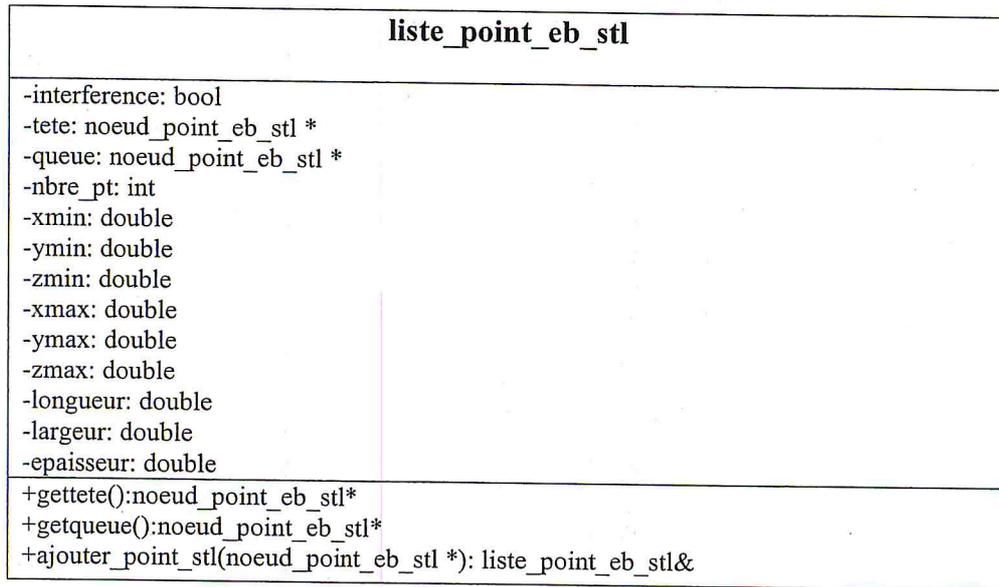


Figure 13 : Diagramme de classe.

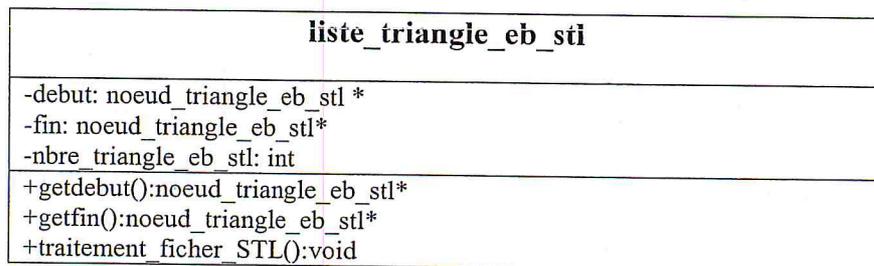
Dans ce qui suit, nous allons détailler les différentes classes utilisées avec leurs attributs et fonctions.

- Classe liste point ébauchage STL : c'est la classe qui regroupe les points du fichier STL. A un emplacement de la liste, nous pouvons trouver les coordonnées du point précédent ou suivant. Nous pouvons se déplacer vers l'avant ou vers l'arrière. La liste comporte une tête qui définit le premier point de la liste et une queue qui définit le dernier point de la liste (Figure14).



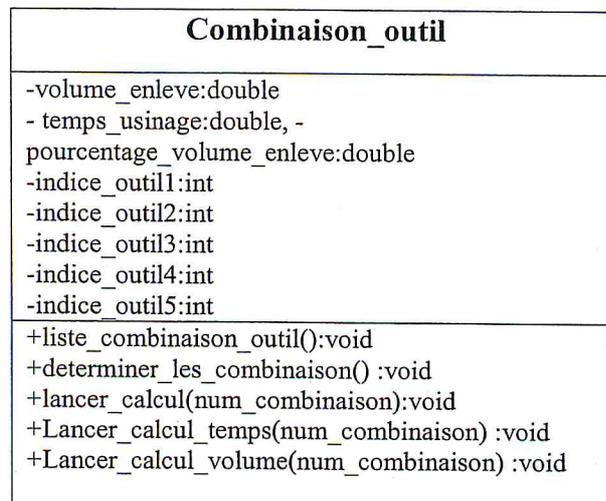
**Figure 14 :** Classe liste point ébauchage STL.

- Classe liste triangle ébauchage STL : regroupe un ensemble de triangles dans une liste. Les attributs et fonctions de cette classe sont donnés par la Figure 15.



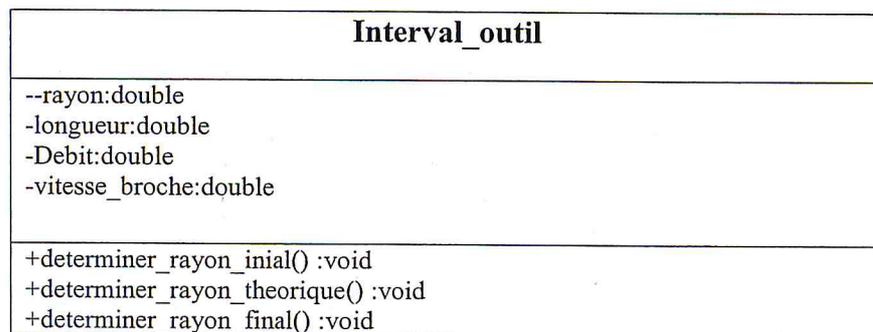
**Figure 15 :** Classe liste triangle ébauchage STL.

- Classe Combinaison\_outil: c'est une combinaison de cinq outils. L'ensemble des combinaisons sont insérées dans une liste de combinaison. Les attributs et quelques fonctions de cette classe sont donnés par la Figure 16.



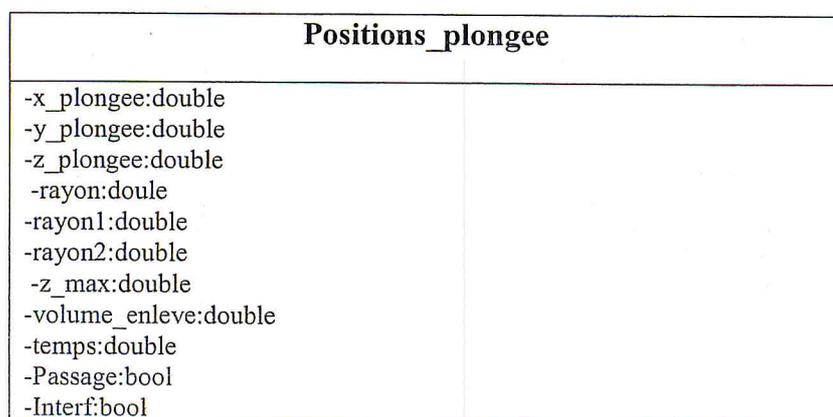
**Figure 16 :** Classe Combinaison\_outil.

- Classe Interval\_outil: les paramètres d'outil nécessaire sont introduits dans cette classe. Par la suite, nous allons calculer les rayons initiaux et les rayons finaux. L'ensemble des outils initiaux et finaux sont regroupés dans une liste chaînée. Les attributs et fonctions de cette classe sont donnés par la Figure 17.



**Figure 17:** Classe Interval outil.

- Classe position\_plongee: elle définit la position de l'outil lors de l'usage. Les attributs et fonctions de cette classe sont donnés par la Figure 18.



```

+verifier_passage_outil():void
+dessiner_nv_centre():void
+dessiner_cercle():void
+calcul_nv_centre():void
+calcul_nv_centre_differeent(outil1, outil2, outil3, outil4, outil5):void
+intercection_droite_cercle():void
+intersection_droite_cercle_differeent():void
+test_interference_cercle():void
+profondeur_plongee():void
+recuit_position_plongee():void
+dessiner_premier_plongee():void
+determiner_aire_usiner():void
+determiner_nb_cercle_outil1():void
+determiner_nb_cercle_outil2():void
+determiner_nb_cercle_outil3():void
+determiner_nb_cercle_outil4():void
+determiner_nb_cercle_outil5():void

```

**Figure 18:** Classe position plongé.

- Classe liste point plongée : l'ensemble de positions de plongée sont insérées dans une liste des point de plongée valides pour l'outil donné. Les attributs et fonctions de cette classe sont donnés par la Figure 19.

<b>Liste_pt_plongee</b>
-debut: noeud_triangle_eb_stl * -fin: noeud_triangle_eb_stl* -nbre_triangle_eb_stl: int
+dessiner_centre_cercle_valide():void +determiner_plongee_successive():void +determiner_plonger_outil2_outil1():void +determiner_plongee_outil3_outil2_outil1():void +determiner_plongee_outil4_outil3_outil2_outil1():void +determiner_plongee_outil5_outil4_outil3_outil2_outil1():void

**Figure 19 :** Classe liste points plongée.

- Classe centre rayon : permet de stocker les coordonnées du centre du cercle avec son rayon. Les attributs et fonctions de cette classe sont donnés par la Figure 20.

<b>Centre_rayon</b>
-x_center:double -y_center:double -z_center:double -r_initial:double -r_intermediaire:double -r_final:double

**Figure 20 :** Classe Centre-rayon.

- Classe `usiner_dexel` : permet de stocker les limites de chaque dexel ainsi que son type et la profondeur de plongée. Les attributs et fonctions de cette classe sont donnés par la Figure 21.

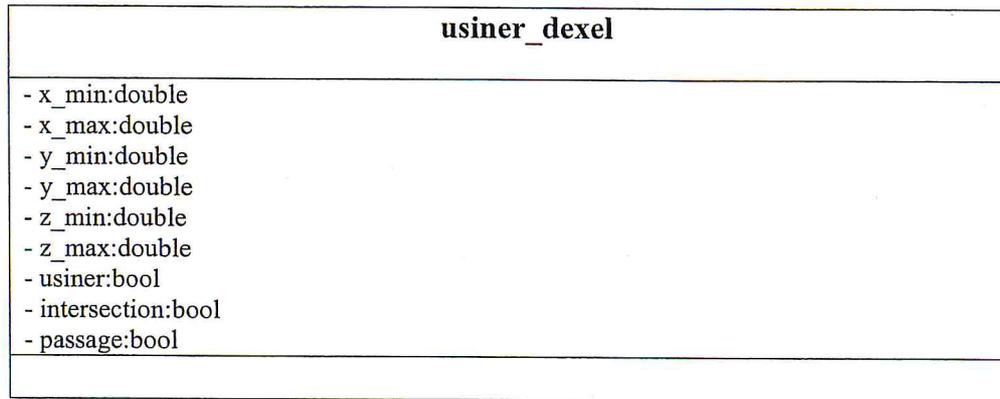


Figure 21 : Classe `usiner_dexel`.

## V. CONCLUSION :

Dans ce chapitre, nous avons défini notre problématique et les différents objectifs à atteindre. Ensuite, nous avons présenté les solutions proposées afin d'atteindre les différents objectifs en utilisant les différents diagrammes de conception UML. Dans le chapitre suivant, nous allons présenter l'implémentation de notre application.

**CHAPITRE III**

**DEVELOPPEMENT  
ALGORITHMIQUE ET  
IMPLEMENTATION  
INFORMATIQUE**

## I. INTRODUCTION

Après avoir exprimé les objectifs de notre travail ainsi que les solutions proposées en utilisant le modèle orienté objet et le langage de modélisation UML (Unified Modeling Language), nous allons passer maintenant à la présentation de notre application logicielle développée en utilisant le langage de programmation Builder C++ et la bibliothèque graphique OpenGL (Open Graphics Library) en mettant en évidence son interface, ses fenêtres et les principaux algorithmes que nous avons développés.

## II. PRESENTATION DE L'APPLICATION

Une application logicielle graphique et interactive de CFAO sous Windows a été développée afin d'optimiser l'opération de Tréflage par le choix de la combinaison optimale des outils et l'optimisation du trajet d'usinage. Plusieurs fenêtres sont intégrées à cette dernière pour lancer les fonctionnalités intégrées.

## III. IMPLEMENTATION

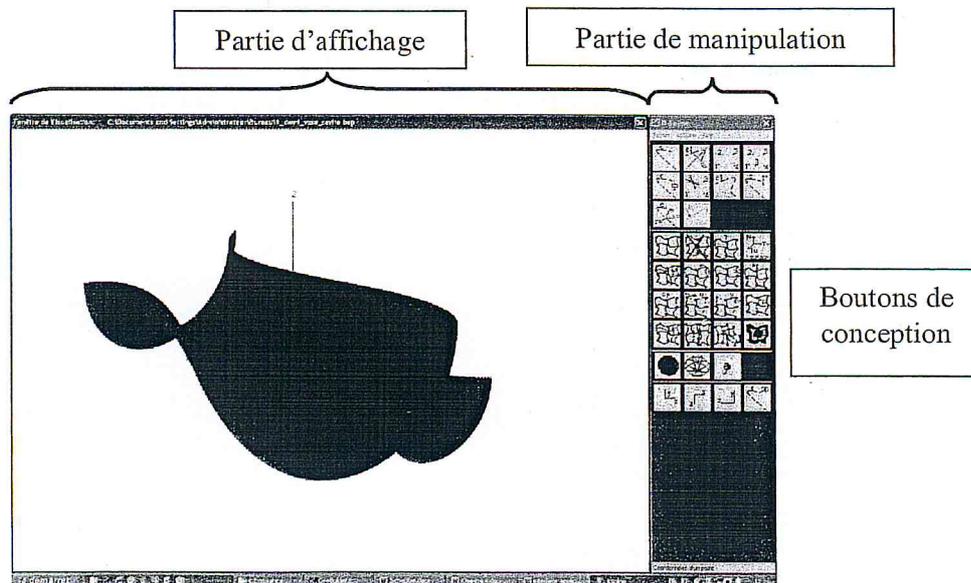
### III.1 Fenêtre principale

La fenêtre principale est composée de deux parties, une partie gauche pour l'affichage et la visualisation graphique des différents objets 3D, et une partie droite pour la manipulation (Figure 1). Cette dernière est composée d'une barre de menu et d'un ensemble de boutons pour lancer les différentes fonctions.

### III.2. Barre du menu principal :

La barre du menu principal est composée de trois rubriques :

- Rubrique Fichier : comporte toutes les fonctionnalités de manipulation des fichiers (ouverture, création, sauvegarde, ...etc.).
- Rubrique Option : permet de modifier les différents paramètres des courbes ainsi que les paramètres des surfaces et les paramètres d'usinage, ...etc.
- Rubrique Aide : pour l'affichage de l'aide.



### III.3. Rubrique d'ébauchage par Tréflage à partir d'un fichier STL :

La partie contenant les modules développés et intégrés à l'application logicielle de CFAO peut être lancée à partir du menu au niveau de la rubrique Option, Usinage, Ebauchage, «Ebauchage par Tréflage (modèle STL)» (Figure 2). Pour pouvoir lancer et exécuter les différentes fonctions de Tréflage, il faut tout d'abord que l'utilisateur ouvre un fichier STL contenant les modèles des surfaces à usiner.

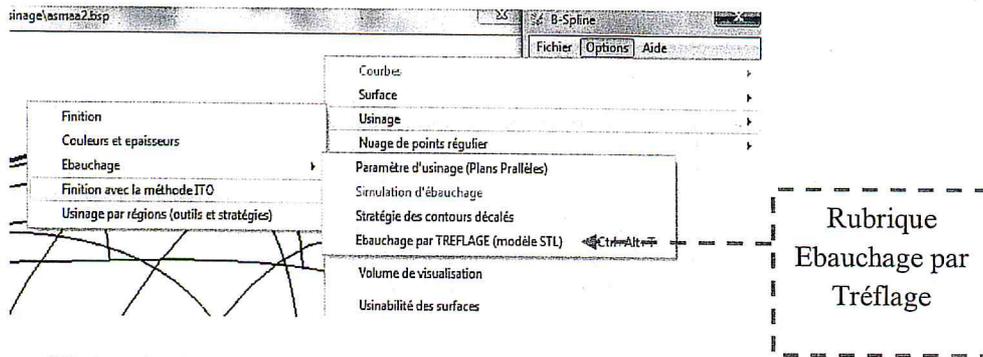


Figure 2: Accès à la rubrique « Ebauchage par Tréflage (modèle STL) ».

### IV. PRESENTATION DES FENETRES :

Dans ce qui suit, nous allons présenter toutes les fenêtres créées dans notre projet ainsi que les principales étapes suivies pour chaque fenêtre.

#### IV.1. Fenêtre d'ouverture d'un fichier STL :

C'est la première fenêtre qui apparaît lorsque l'utilisateur lance l'ébauchage par Tréflage à partir d'un fichier STL (Figure 3). Elle permet de lire le fichier « STL », de générer les contours, de créer des cellules de triangles et de créer les dexels.

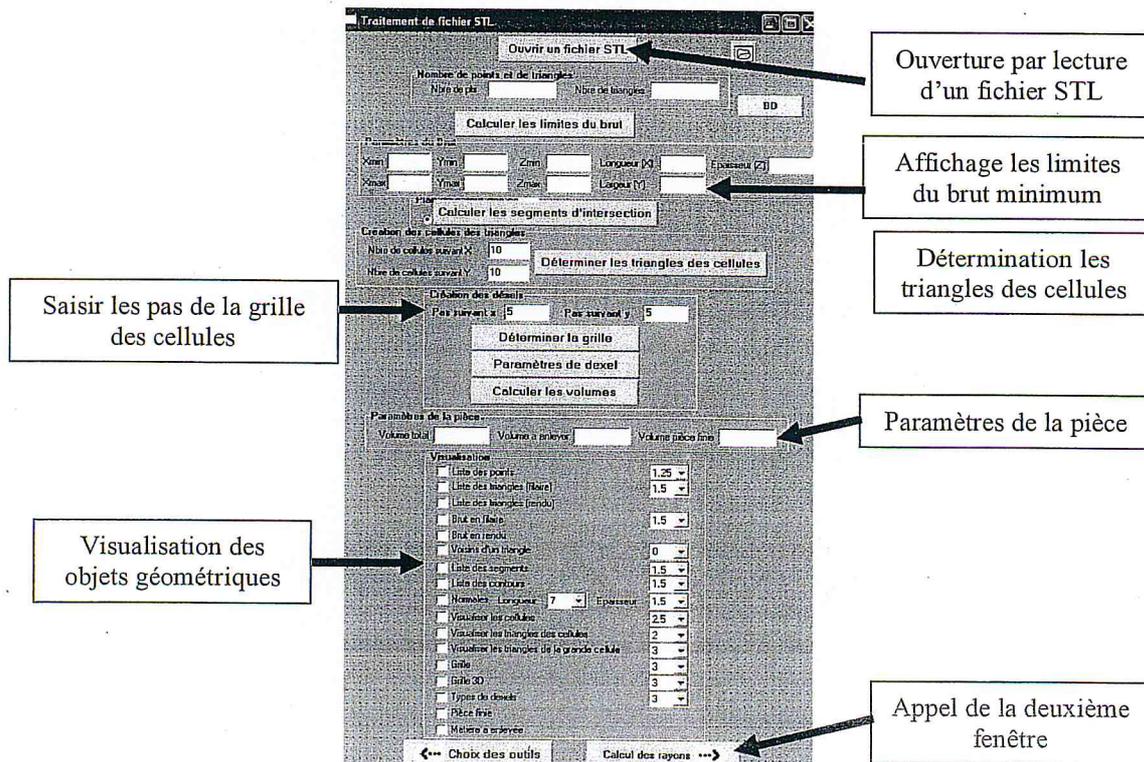


Figure 3: Fenêtre de traitement du fichier STL.

#### IV.1.1. Lecture du fichier STL :

Une fois que le fichier est ouvert, une procédure d'élimination des points redondants est lancée. Par la suite, les limites des surfaces  $X_{MIN}$ ,  $X_{MAX}$ ,  $Y_{MIN}$ ,  $Y_{MAX}$ ,  $Z_{MIN}$  et  $Z_{MAX}$  sont calculées pour pouvoir déterminer les dimensions minimales du brut. Après cette étape, les contours sur la face supérieure sont créés. Il est possible de visualiser la liste des points, la liste des triangles, le brut minimum et les contours.

#### IV.1.2. Création des cellules des triangles :

Afin d'accélérer les calculs de détection des intersections entre les triangles de la surface et les dexels et par conséquent minimiser les temps de calcul, nous avons procédé à la création des cellules de triangles (Figure 4). Pour cela, nous avons généré deux listes, la première liste c'est la liste des triangles des cellules, qui regroupe tous les triangles appartenant à l'enveloppe d'une cellule et la deuxième liste c'est la liste des triangles de la cellule supplémentaire qui regroupe tous les triangles appartenant à plus d'une cellule. La création des cellules nécessite la spécification du nombre de cellules suivant l'axe X et le nombre de cellules suivant l'axe Y. A partir de ces données, le pas suivant les axes X et Y et les limites de chaque cellule sont calculés (Figure 5). Nous obtenons une grille des cellules de triangles.

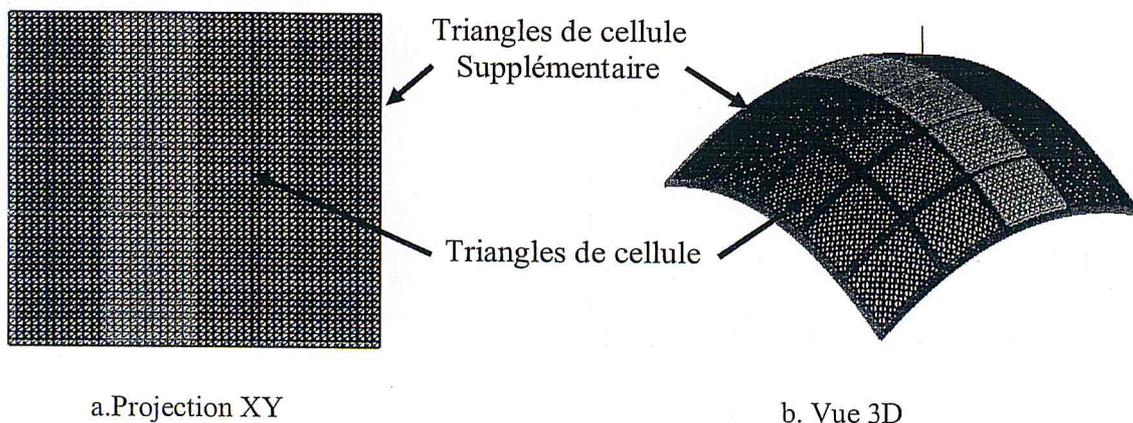


Figure 4 : Triangles des cellules et de la cellule supplémentaire.

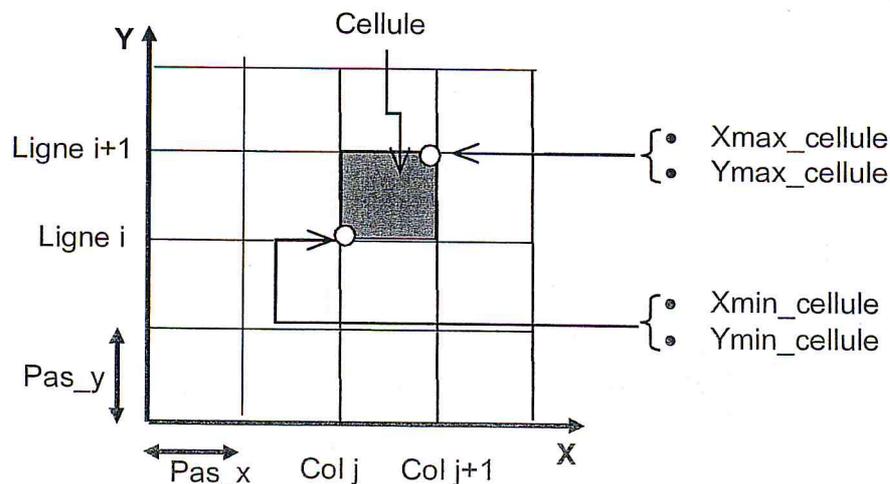


Figure 5 : Création des limites de chaque cellule.

#### IV.1.3. Création des dexels :

Pour modéliser une pièce volumique donnée par des dexels, deux étapes sont nécessaires. La première étape consiste à déterminer les limites d'un dixel suivant les axes X et Y, tandis que la deuxième étape consiste à calculer la limite suivant l'axe Z où le dixel est intersecté avec un triangle de la surface.

#### a) Génération de la grille des dexels:

Pour générer la grille des dexels (Figure 6), l'utilisateur doit spécifier le pas de la grille suivant l'axe X et le pas de la grille suivant l'axe Y.

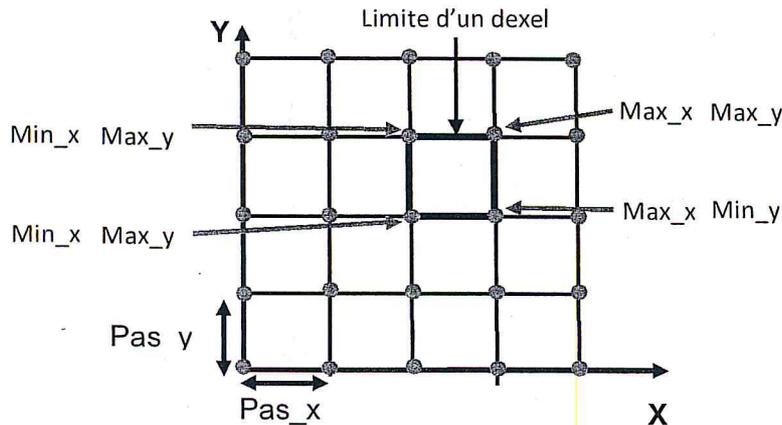


Figure 6 : Création de la grille des dexels.

#### b) Calcul des paramètres des dexels :

Lorsque l'utilisateur clique sur le bouton « *paramètres de dexels* », la détermination de la hauteur de chaque dixel passe par les étapes suivantes (Figure 7) :

1. Pour chaque point des limites d'un dixel :
  - Vérifier l'appartenance de ce point à un triangle de la liste des triangles des cellules ou à la liste des triangles de la cellule supplémentaire.
  - Si le point appartient au triangle, calculer l'intersection de la droite verticale passant par ce point avec les triangles (coordonnée Z).
2. La hauteur du dixel est prise égale à la valeur maximale des Z calculés.

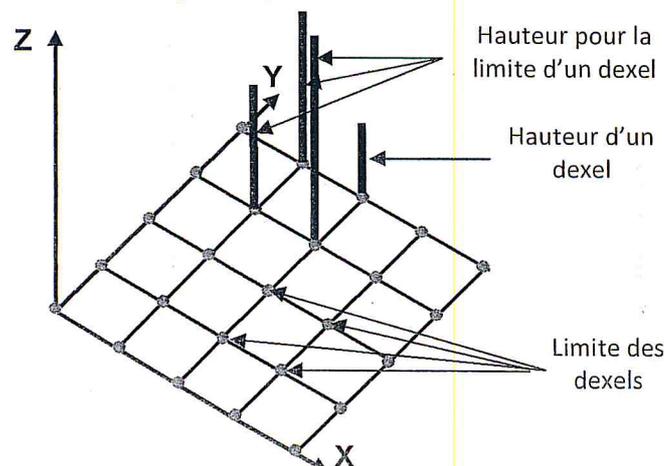


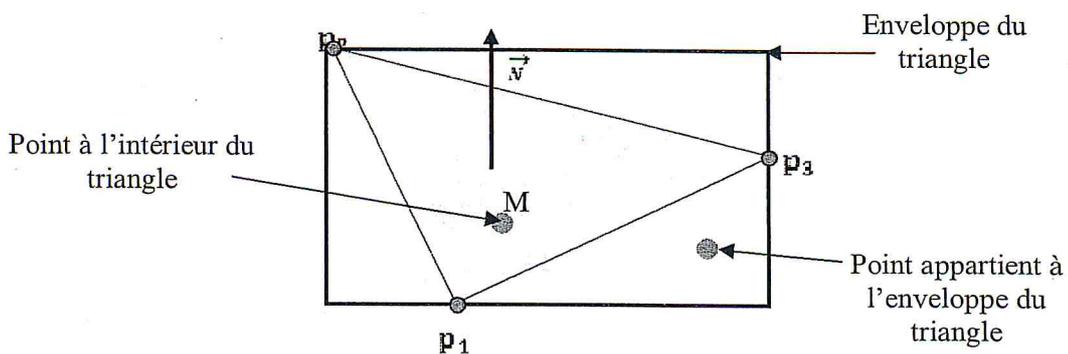
Figure 7 : Hauteur d'un dixel.

Pour vérifier l'appartenance d'un point  $M(x_m, y_m)$  à un triangle, c'est-à-dire déterminé si ce point est à l'extérieur ou à l'intérieur du triangle (Figure 8). Etant donné que chaque triangle est défini par ses trois sommets  $p_1(x_1, y_1, z_1)$ ,  $p_2(x_2, y_2, z_2)$  et  $p_3(x_3, y_3, z_3)$  et son vecteur normal  $\vec{N}(n_x, n_y, n_z)$ , deux cas peuvent être rencontrés selon la position du triangle dans le plan :

- **Cas 1** : si ( $n_z \neq 0$ ) : pour savoir si le point M est à l'intérieur ou à l'extérieur du triangle, le système d'équations suivant est utilisé :

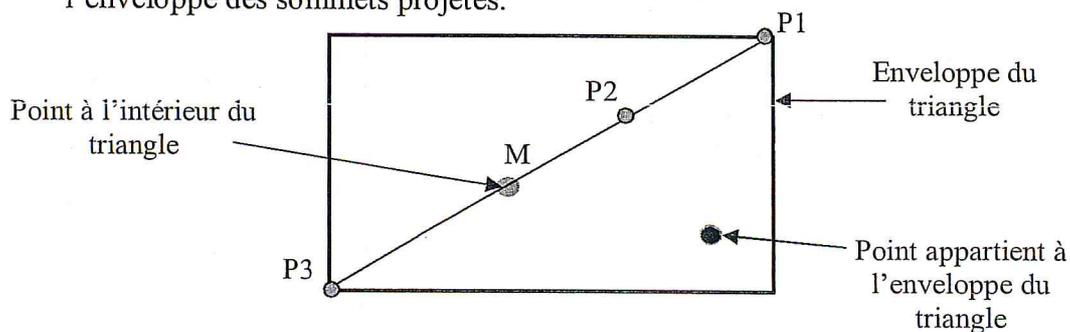
$$\begin{cases} (\overrightarrow{p_1p_2} \wedge \overrightarrow{p_1M}) * (\overrightarrow{p_1M} \wedge \overrightarrow{p_1p_3}) \geq 0 \\ (\overrightarrow{p_2p_1} \wedge \overrightarrow{p_2M}) * (\overrightarrow{p_2M} \wedge \overrightarrow{p_2p_3}) \geq 0 \\ (\overrightarrow{p_3p_1} \wedge \overrightarrow{p_3M}) * (\overrightarrow{p_3M} \wedge \overrightarrow{p_3p_2}) \geq 0 \end{cases} \quad (4.1)$$

Le point M appartient au triangle si les trois conditions sont vérifiées (Figure 8).



**Figure 8** : Appartenance d'un point à un triangle non vertical.

- **Cas 2** : si ( $n_z = 0$ ) : c'est le cas d'un triangle vertical. Donc, les trois sommets projetés sont alignés (Figure 9). Dans ce cas, le point M appartient au triangle si sa projection est alignée avec les sommets projetés et s'il appartient à l'enveloppe des sommets projetés.



**Figure 9** : Appartenance d'un point à un triangle vertical.

Pour déterminer analytiquement l'intersection d'une droite et d'un triangle, le triangle est considéré un plan infini. Donc le calcul d'intersections revient à réaliser l'intersection entre un plan et une droite parallèle à l'axe Z (droite verticale).

Après détermination de la hauteur de chaque dixel, l'étape suivante consiste à identifier sa nature (usinable ou non usinable). Cela, passe par les étapes suivantes :

- Calculer l'intersection de chaque droite verticale passant par les quatre coins du dixel avec les triangles :
  - Si tous les points d'intersection sont au-dessous de la face supérieure de la pièce, alors le dixel est à usiner.

- S'il existe au moins un point d'intersection appartenant à la face supérieure de la pièce, alors le dixel est à ne pas usiner.

#### IV.2. Fenêtre de calcul des outils utilisables :

Cette fenêtre apparaît lorsque l'utilisateur clique sur le bouton « calcul des rayons » de la première fenêtre (Figure 10). L'objectif principal de cette fenêtre est la détermination de la liste des outils pouvant être utilisés dans l'opération de Tréflage. Pour cela, il est impératif de passer par les étapes suivantes :

- Calcul des points d'intersection entre la droite normale à chaque segment et passant par son milieu avec tous les segments de la face supérieure.
- Calcul du rayon initial.
- Calcul du rayon théorique.
- Calculer le rayon final à utiliser.
- Calcul des rayons maximums et minimums.
- Création de la liste des outils utilisables.

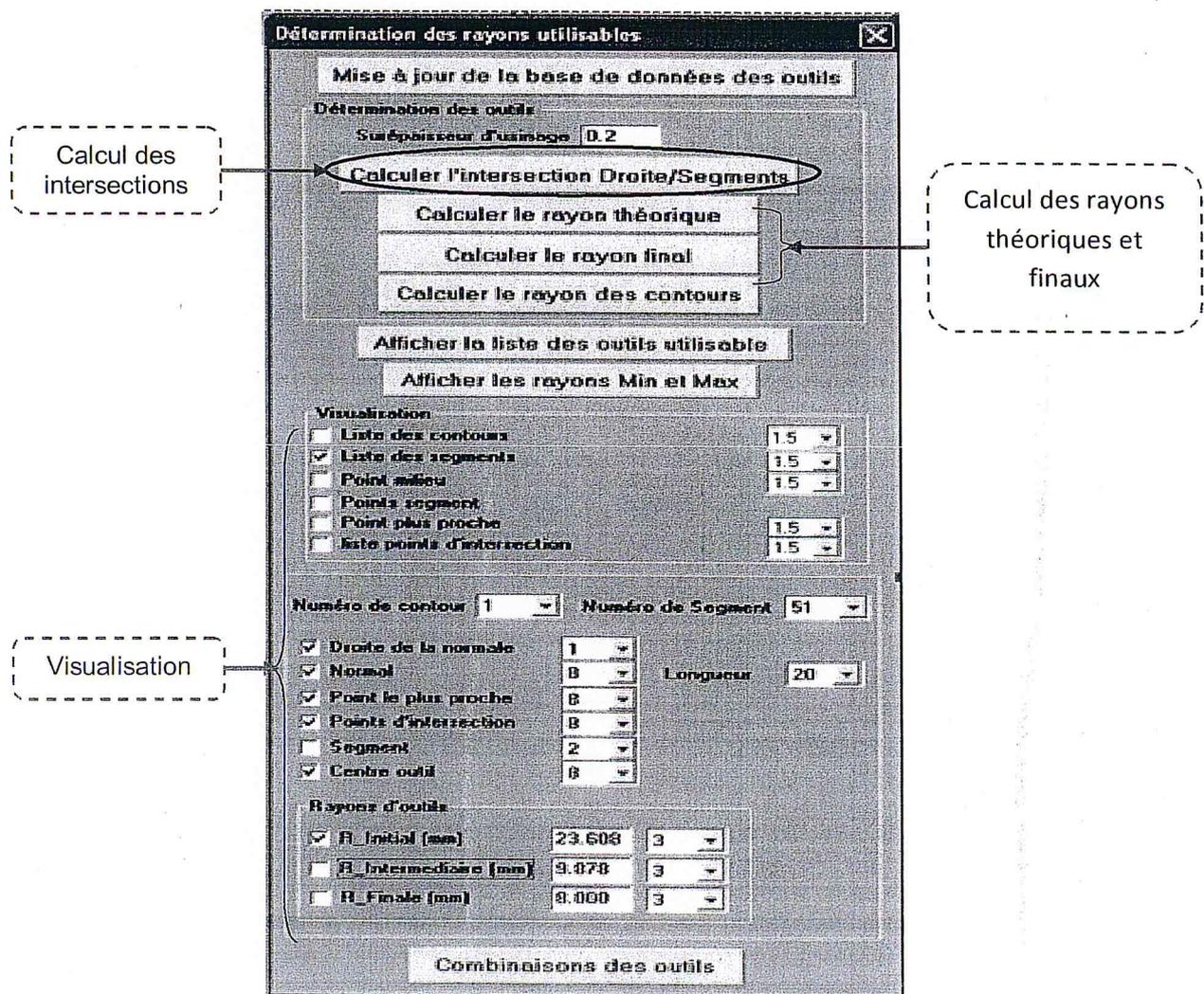


Figure 10 : Fenêtre de détermination des rayons utilisables.

### IV.2.1. Calcul du rayon initial :

Pour chaque segment, le point milieu  $M$  du segment est calculé (Figure 11). Par la suite, les points d'intersection entre la droite passant par  $M$  et parallèle à la normale du segment avec tous les segments de la face supérieure sont calculés. Le cercle dont le rayon est égal à la moitié de la distance entre le point d'intersection le plus proche et le point  $M$  est le cercle initial et son rayon est au rayon initial. Le principe de calcul de l'intersection entre deux segments est donné par les équations suivantes (Figure 11) :

- $MP = \alpha * N$

$$X = X_M + \alpha * X_N \quad (1)$$

$$Y = Y_M + \alpha * Y_N \quad (1)$$

- $P_1P = \beta * P_1P_2$

$$X = X_1 + \beta * (X_2 - X_1) \quad (2)$$

$$Y = Y_1 + \beta * (Y_2 - Y_1) \quad (2)$$

Au point d'intersection, nous avons les deux égalités suivantes :

$$X_M + \alpha * X_N = X_1 + \beta * (X_2 - X_1) \quad (3)$$

$$Y_M + \alpha * Y_N = Y_1 + \beta * (Y_2 - Y_1) \quad (3)$$

Le point d'intersection est valide si  $0 \leq \beta \leq 1$  et  $\alpha > 0$ .

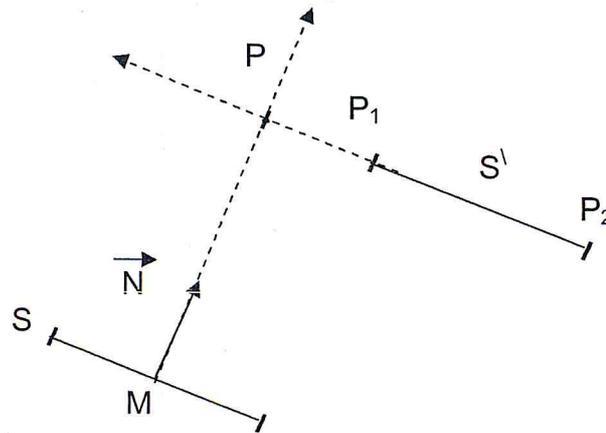


Figure 11 : Intersection de deux segments.

La Figure 12 montre le résultat pour un segment donné.

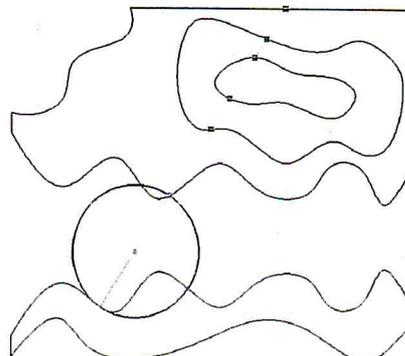


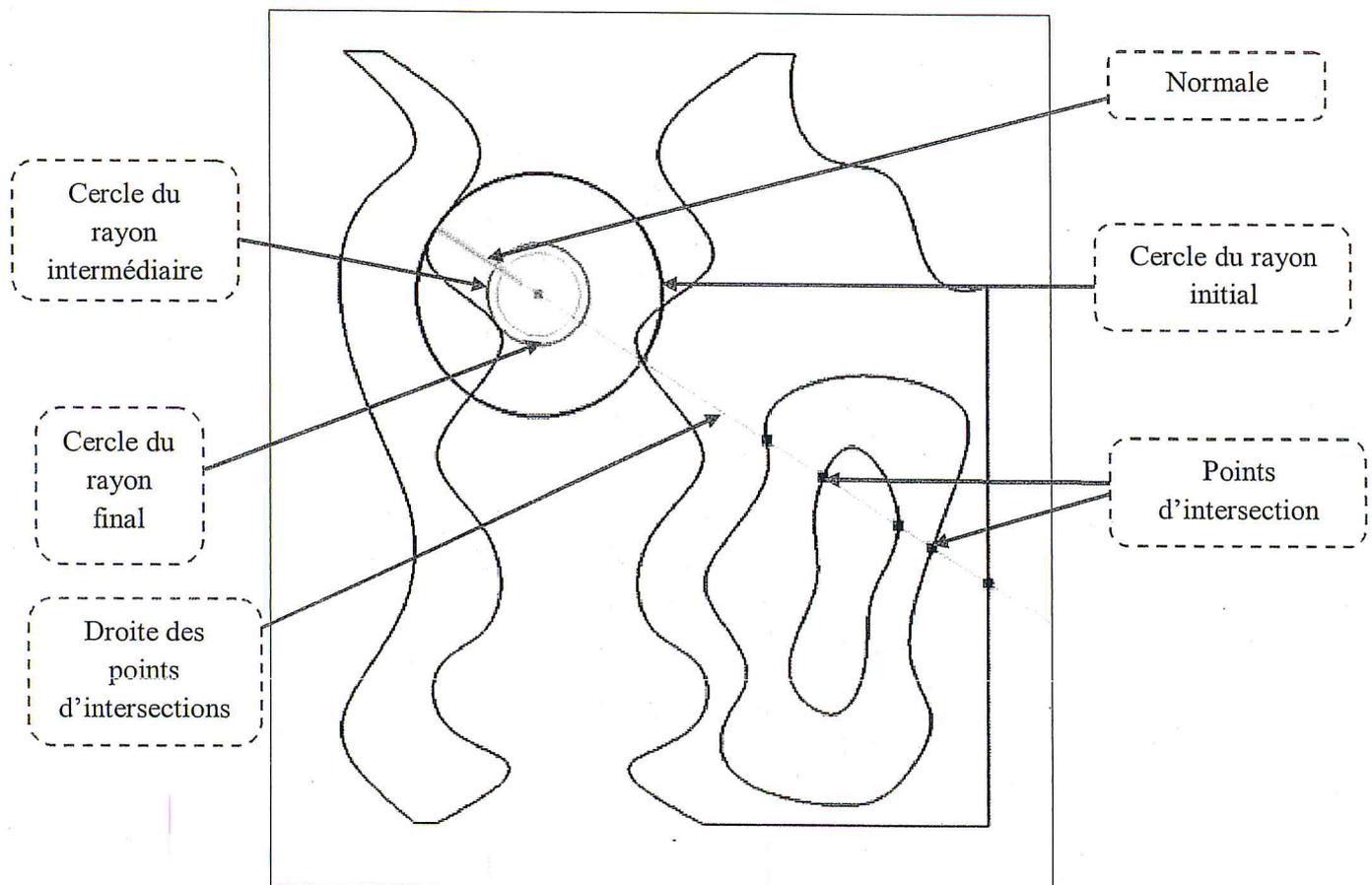
Figure 12 : Rayon initial.

#### IV.2.2. Calcul du rayon théorique (intermédiaire) :

Le rayon théorique ou intermédiaire est déterminé à partir de la distance minimale entre tous les segments et le centre du rayon initial. Pour accélérer les calculs, l'enveloppe du segment est utilisée pour délimiter la zone de recherche.

#### IV.2.3. Calcul du rayon final :

Le rayon final de chaque segment est déterminé à partir de la base de données des outils. Le plus grand rayon de la base de données d'outils inférieur ou égal au rayon théorique est le rayon final à utiliser. La Figure 13 montre les trois rayons pour un segment donné.



**Figure13** : Rayon initial, théorique et final lors d'existence de points d'intersection.

Dans le cas où l'intersection entre la droite normale au segment et passant par le milieu avec tous les segments est un ensemble vide, le rayon final est le plus grand rayon de l'outil choisi à partir de la base de données d'outils qui ne crée pas d'interférences avec les segments et dont le centre est sur la droite à une distance égale au rayon de l'outil plus une surépaisseur d'usinage (Figure 14). Dans ce cas, le rayon final, le rayon intermédiaire et le rayon final sont égaux.

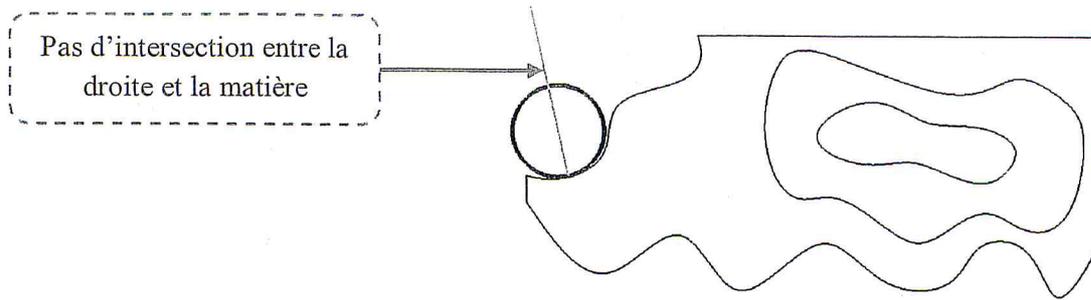


Figure 14 : Rayon initial, théorique et final lors d'absence de points d'intersection.

Les valeurs maximales et minimales des rayons finaux des segments sont utilisées pour déterminer la liste des outils utilisables et par conséquent le calcul des combinaisons possibles des outils.

### IV.3. Fenêtre de calcul des positions de plongée :

Cette fenêtre apparaît lorsque l'utilisateur clique sur le bouton « combinaison des outils » de la fenêtre précédente (Figure 15). L'objectif principal de cette fenêtre est le calcul pour chaque combinaison possible d'outils, les positions de plongée valides, le volume de matière enlevé et le temps d'usinage. Dans ce travail, une combinaison de cinq outils est considérée.

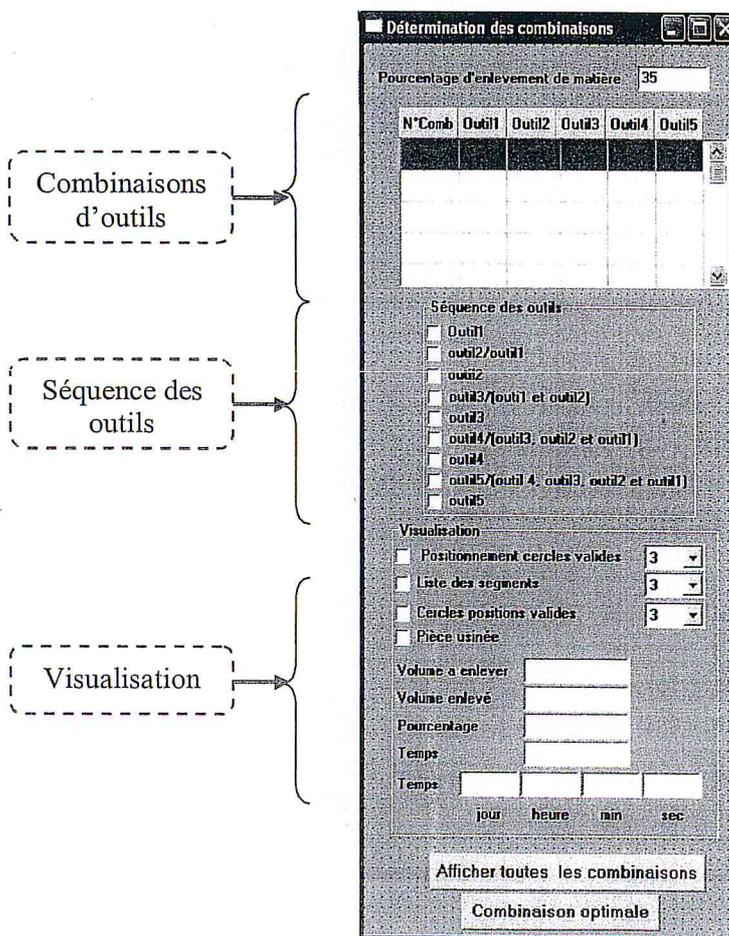
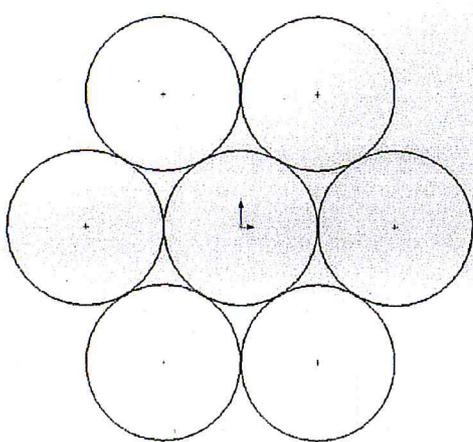


Figure 15 : Fenêtre de calcul du volume enlevé et du temps d'usinage.

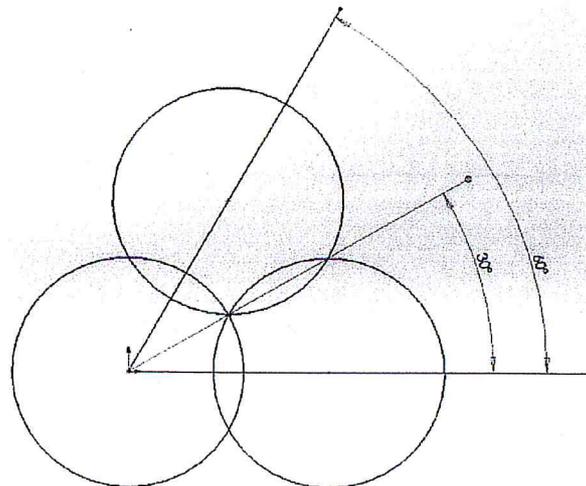
### IV.3.1. Calcul des positions de plongée :

Le principe utilisé dans la démarche adoptée consiste à entourer chaque cercle par un nombre minimum de cercles pour enlever toute la matière autour du cercle central. Pour calculer les positions de plongée, deux cas sont à distinguer :

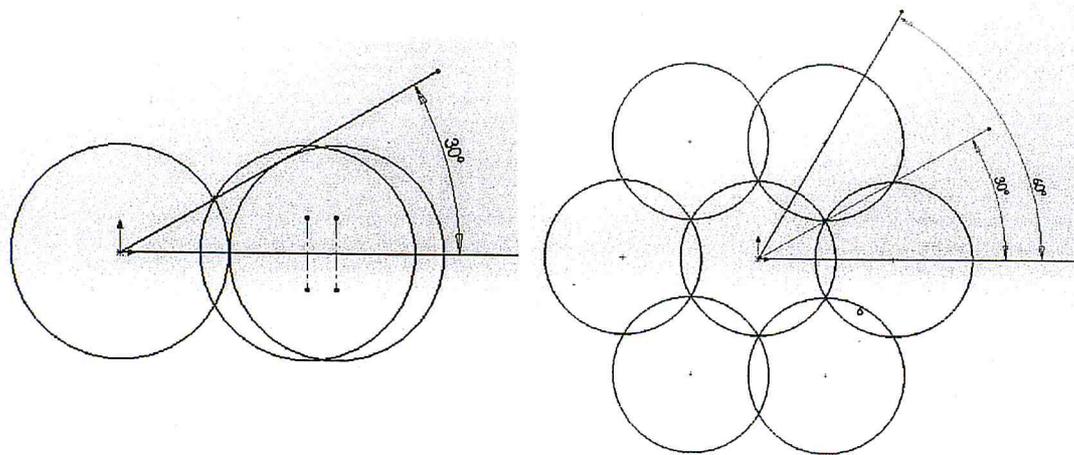
- **Cas 1 :** Rayons des cercles concentriques sont égaux au rayon du cercle central. Le processus de génération des positions de plongée valides autour d'une position de plongée valide passe par les étapes suivantes :
  - Création de six cercles tangents au cercle central (Figure 16.a). Avec cette disposition une crête est laissée entre deux cercles concentriques et le cercle central.
  - Calcul du point d'intersection entre le cercle central et la droite passant par le centre du cercle central et faisant un angle égal à  $30^\circ$  par rapport à l'horizontal (Figure 16.b).
  - Calcul du déplacement horizontal du premier cercle pour que le point d'intersection appartienne au cercle (Figure 16.c). Le déplacement est effectué pour ne pas laisser de crête.
  - Calcul des nouveaux centres des cercles concentriques (Figure 16.d).
  - Pour chaque cercle concentrique :
    - ❖ S'il y a chevauchement entre le cercle et les contours (interférence), alors la position de ce cercle n'est pas une position de plongée valide (centre rouge) (Figure 16.d).
    - ❖ S'il n'y a pas de chevauchement entre le cercle et les contours, alors la position de ce cercle est une position de plongée valide (centre vert) (Figure 16.d).



a. Cercles concentriques.

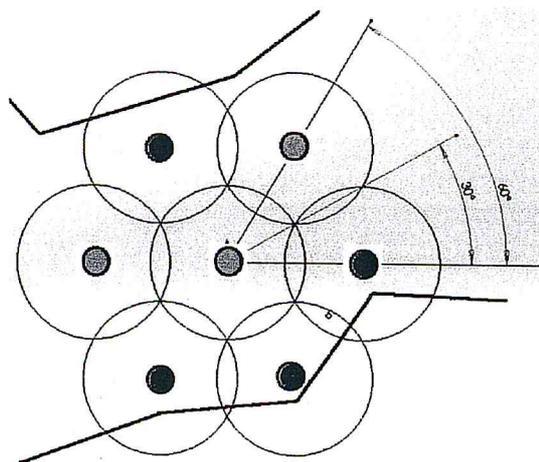


b. Points d'intersection.



c. Déplacement axial du centre de cercle.

d. Répartition finale des cercles.

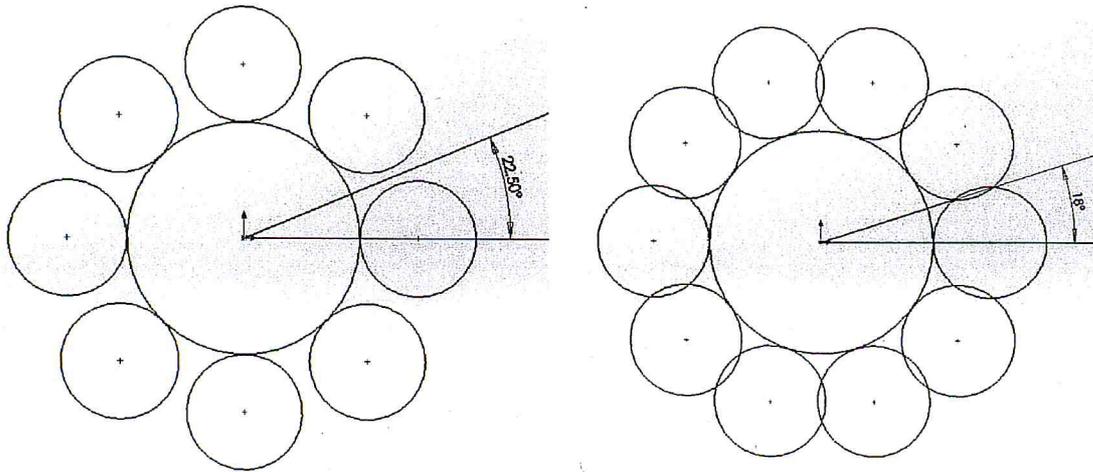


d. Positions de plongée valides et non valides.

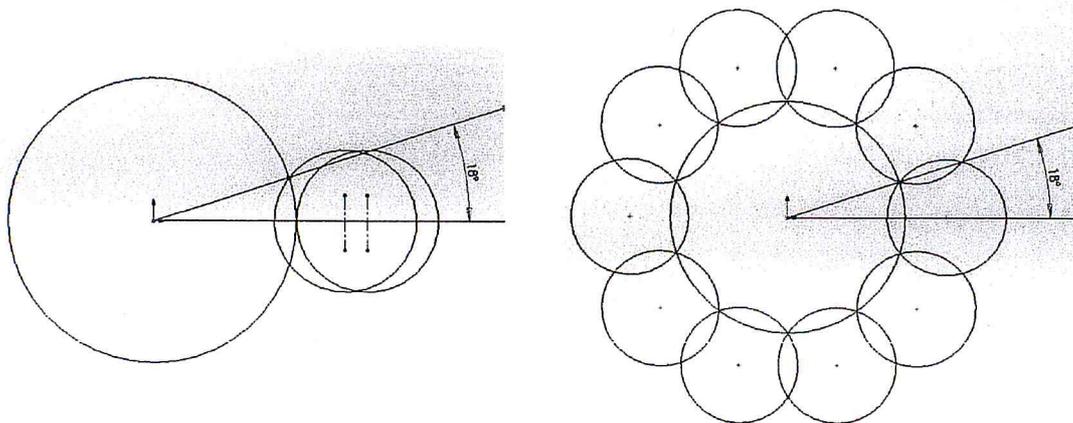
**Figure 16 :** Positions de plongée pour des cercles de mêmes rayons.

- **Cas 2 :** Rayons des cercles concentriques sont égaux et sont plus petits que le rayon du cercle central. Le processus de génération des positions de plongée valides autour d'une position de plongée valide passe par les étapes suivantes :
  - Calcul du minimum nombre pair de cercles concentriques et tangents au cercle central ayant des intersections entre chaque deux cercles concentriques successifs (Figure 17.a). Avec cette disposition une crête est laissée entre deux cercles concentriques et le cercle central.
  - Calcul du point d'intersection entre le cercle central et la droite passant par le centre du cercle central et faisant un angle égal à  $360^\circ / (2 \times \text{nombre\_cercles})$  par rapport à l'horizontal (Figure 17.b).
  - Calcul du déplacement horizontal du premier cercle pour que le point d'intersection appartienne au cercle (Figure 17.c). Le déplacement est effectué pour ne pas laisser de crête.
  - Calcul des nouveaux centres des cercles concentriques (Figure 17.d).
  - Pour chaque cercle concentrique :

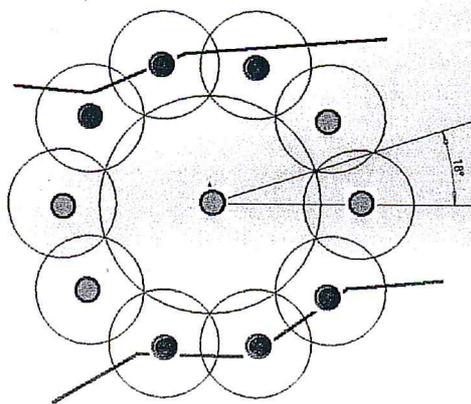
- ❖ S'il y a chevauchement entre le cercle et les contours (interférence), alors la position de ce cercle n'est pas une position de plongée valide (centre rouge) (Figure 17.d).
- ❖ S'il n'y a pas de chevauchement entre le cercle et les contours, alors la position de ce cercle est une position de plongée valide (centre vert) (Figure 17.d).



a. Nombre minimum de cercles concentriques.



b. déplacement horizontal du centre de cercle. c. Répartition finales des cercles concentriques.



d. Positions de plongée valides et non valides.

Figure 17 : Positions de plongée pour des cercles de rayons différents.

### IV.3.2. Calcul des profondeurs de plongée, du volume enlevé et du temps :

Une fois les positions de plongée sont déterminées, l'étape suivante consiste à déterminer la profondeur de plongée associée à chaque position de plongée valide et le volume de matière enlevé. Cela nécessite le passage par les étapes suivantes :

- ❖ Pour chaque position de plongée :
  - Identifier les dexels sous le cercle.
  - Déterminer la coordonnée Z maximale de ces dexels.
  - Calculer la profondeur de plongée.
  - Calculer le volume enlevé pour les dexels sous le cercle.
  - Calculer le temps d'usinage.

### IV.4. Fenêtre de détermination de la combinaison optimale des outils :

Cette fenêtre apparaît lorsque l'utilisateur clique sur le bouton « combinaison optimale » de la fenêtre précédente (Figure 18). L'objectif principal de cette fenêtre est le calcul de la combinaison optimale des outils permettant d'enlever le maximum de matière en un minimum de temps.

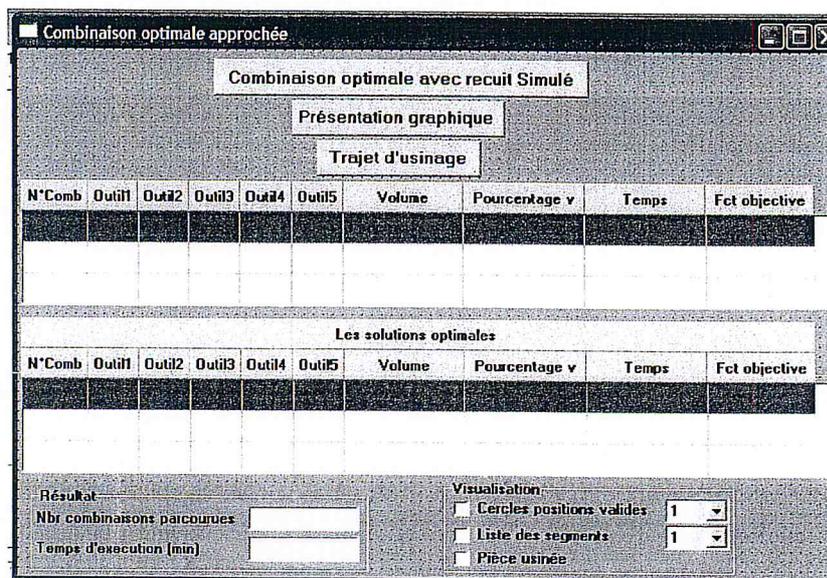


Figure 18 : Fenêtre de détermination de la combinaison optimale.

Etant donné que nous devons choisir cinq outils parmi « n » outils de la liste des outils utilisables pour ébaucher la pièce, le nombre de combinaisons possibles d'outils devient grand et le temps nécessaire pour déterminer la combinaison optimale des outils devient très important. Ce temps dépend également de la précision d'approximation du modèle « STL » par des dexels. L'objectif étant de déterminer la combinaison d'outils permettant d'enlever le maximum de matière en un minimum de temps en un temps de calcul très raisonnable. Ces deux sous objectifs étant contradictoires, il faut donc pondérer ces paramètres, c'est-à-dire affecter chaque paramètre d'un coefficient réel positif compris entre 0 et 1 tout en ayant la somme des deux facteurs égale à 1. Pour

cela, les algorithmes d'optimisation multi objectifs doivent être utilisés. Pour résoudre ce problème, nous avons opté pour l'algorithme du « recuit simulé ».

#### IV.4.1. Fonction objective :

Notre objectif principal est d'obtenir une combinaison d'outils qui puisse minimiser le temps d'usinage tout en maximisant le volume à enlever. La fonction objective est donnée par :

$$F(S) = \alpha \mathbf{1} * \left( \frac{\mathbf{1}}{p_v} \right) + \beta (t_p)$$

$t_p$  : temps d'usinage de la pièce divisé par le temps associé au plus petit outil.

$p_v$  : pourcentage du volume enlevé (rapport entre le volume enlevé et le volume total à enlever).

$\alpha, \beta$  : facteurs de pondération du volume et du temps avec  $\alpha + \beta = 1$ .

#### IV.4.2. Algorithme du recuit simulé :

Cet algorithme s'inspire d'une procédure utilisée depuis longtemps par les métallurgistes qui, pour obtenir un alliage sans défaut, chauffent d'abord à blanc leur morceau de métal, avant de laisser l'alliage se refroidir très lentement (technique du recuit). Pour simuler l'évolution d'un système physique vers son état d'équilibre thermodynamique à une température  $T$ , la méthode du recuit simulé exploite le critère de *Métropolis*. Dans le critère de *Métropolis*, à partir d'une configuration donnée, le système subir une modification élémentaire (perturbation). Si cette perturbation a pour effet de diminuer la fonction objective (ou *énergie*) du système, elle est acceptée. Sinon, elle est acceptée avec la probabilité  $\exp(\Delta E/T)$ . En appliquant itérativement cette règle, une séquence de configurations est engendrée qui tendent vers l'équilibre thermodynamique. L'algorithme du recuit simulé est basé sur deux procédures :

- Procédure d'échantillonnage :
  - ✓ Phase d'exploration - Notion de voisinage
  - ✓ Phase d'acceptation - Utilisation de l'algorithme de Metropolis
- Procédure de refroidissement :
  - ✓ Schéma de décroissance de la température

Le pseudo-code suivant décrit l'algorithme du recuit simulé :

##### Initialisation

Choisir une solution initiale  $S \in X$ ;

$S^* \leftarrow S$  ;  $I \leftarrow 0$  ;(compteur d'itération global)

$Nv\_palier \leftarrow \text{vrai}$  ;

$T \leftarrow T_0$  ;(température initial du système)

##### Processus itératif

Tant que ( $Nv\_palier = \text{vrai}$ ) faire

```

i ← 0 ; Nv_palier ← faux ;
tant que (i < nb_iter_palier) faire
i = i + 1 ;
générer aléatoirement une solution S' ∈ N(S) ;
Δ F ← F(S') - F(S) ;
Si (Δ F < 0) alors
S ← S' ;
Nv_palier ← vrai ;
Sinon
Prob(Δ F, T) ← exp(-Δ F/T) ;
Générer q uniformément dans l'intervalle [0,1[ ;
Si (q < prob(Δ F, T)) alors
S* ← S ;
Nv_palier ← vrai ;
Si (FS < FS*) alors
S* ← S ;
Finsi
Finsi
Finsi
Finsi
Fin
T = α * T ; ( 0 < α < 1 coefficient de refroidissement ) .
Fin

```

Cet algorithme nécessite la spécification de plusieurs paramètres : température initiale, température finale, nombre d'itérations par palier et coefficient de refroidissement.

#### IV.4.3. Perturbation :

Une des étapes essentielles de l'algorithme est la génération d'une solution voisine suite à la perturbation de la solution courante. La procédure utilisée pour la perturbation est décrite par l'algorithme suivant :

##### Entrées :

lp est une liste // la liste de combinaison possibles ;

L'intervalle d'outil (IO) ;

v est un entier ;

l est une liste //liste de combinaison utilisée ;

i est un entier := 0 ;

Sorties : une nouvelle combinaison d'outils

**Début**

```
Extraire aléatoirement à partir de lp une combinaison (C1)
A partir de C1, choisir aléatoirement de 0 a 3 un indice (indice1)
S : A partir de IO, choisir aléatoirement un indice (indice2)
//permutation entre indice1 et indice2
v :=indice1 ;
Indice1 :=indice2 ;
Indice2 :=v ;
Si (i !=4)
    Si (la nouvelle combinaison existe dans l) alors
        i=i+1 ;
        Aller à S
    Sinon
        Aller à l'algorithme recuit simulé
FinSi
Sinon // i :=4
    B : Extraire aléatoirement à partir de lp une autre combinaison
        Si (la nouvelle combinaison existe dans l) alors
            Aller à B
        Sinon
            Aller à l'algorithme recuit simulé
Finsi
Fin
Fin //fin de l'algorithme
```

**IV.5. Fenêtre de détermination du trajet d'usinage :**

Cette fenêtre apparaît lorsque l'utilisateur clique sur le bouton « trajet d'usinage » de la fenêtre précédente (Figure 19). L'objectif principal de cette fenêtre est le calcul du chemin le plus court et la génération du trajet d'outils global et le programme d'usinage « G-Code ». De la même manière, le trajet associé à chaque outil est optimisé par la détermination du plus court chemin reliant toutes les positions de plongée d'un outil donné en utilisant l'algorithme du recuit simulé. Après cette étape, les différents trajets sont connectés entre eux pour former le trajet final.

Le problème du trajet d'usinage, étant donné un ensemble de plusieurs plongés, à trouver le plus court chemin qui relie toutes les plongées d'un même outil. Il s'agit d'un problème d'optimisation pour lequel on ne connaît pas d'algorithme permettant de trouver une solution exacte en un temps polynomial. De plus, la version décisionnelle de l'énoncé (pour une distance D, existe-t-il un chemin plus court que D passant par toutes les positions ?) est connue comme étant un problème NP-complet.

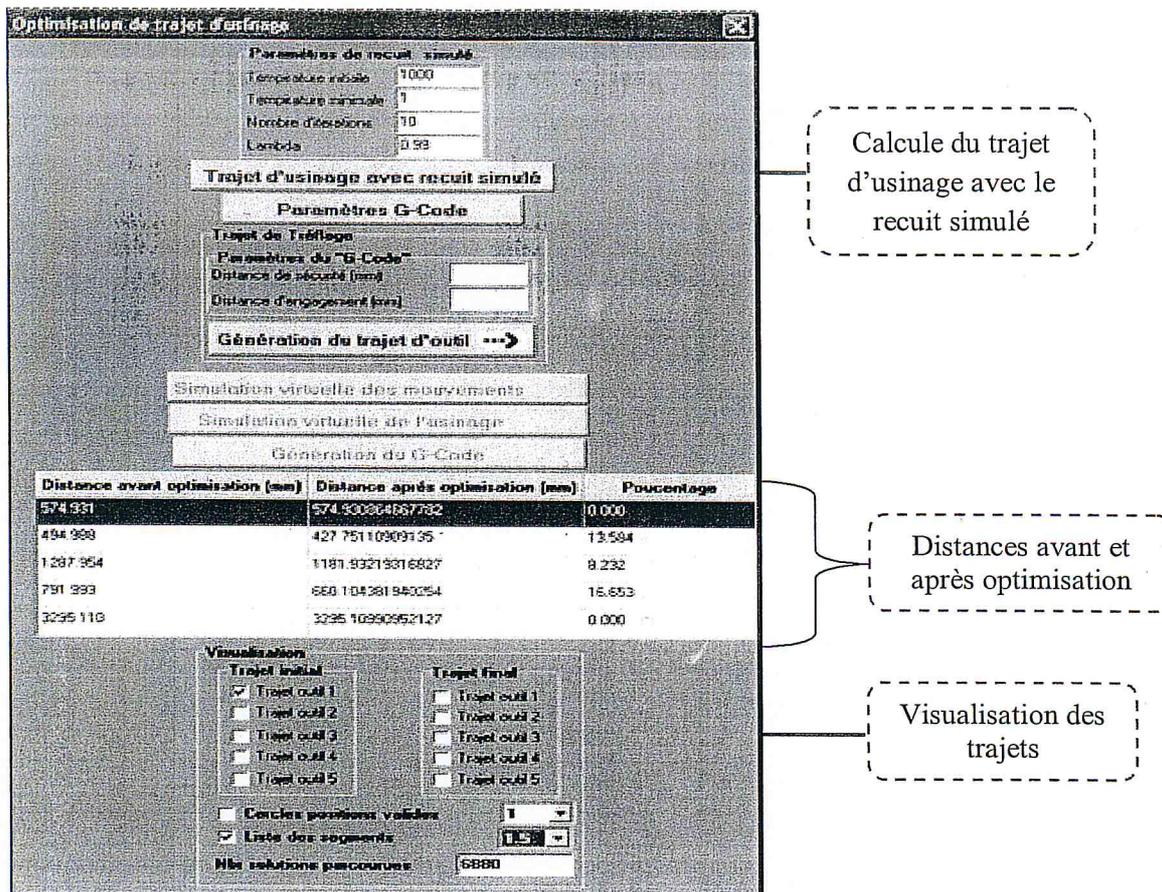


Figure 19 : Fenêtre d'optimisation du trajet d'usinage.

#### IV.5.1. Perturbation :

Lors de l'usinage de la pièce, chaque outil dispose d'une table des positions de plongée générée à partir de la liste globale de toutes les positions de plongée. Puisque nous avons cinq outils, cinq tables sont créées (Figure 20).

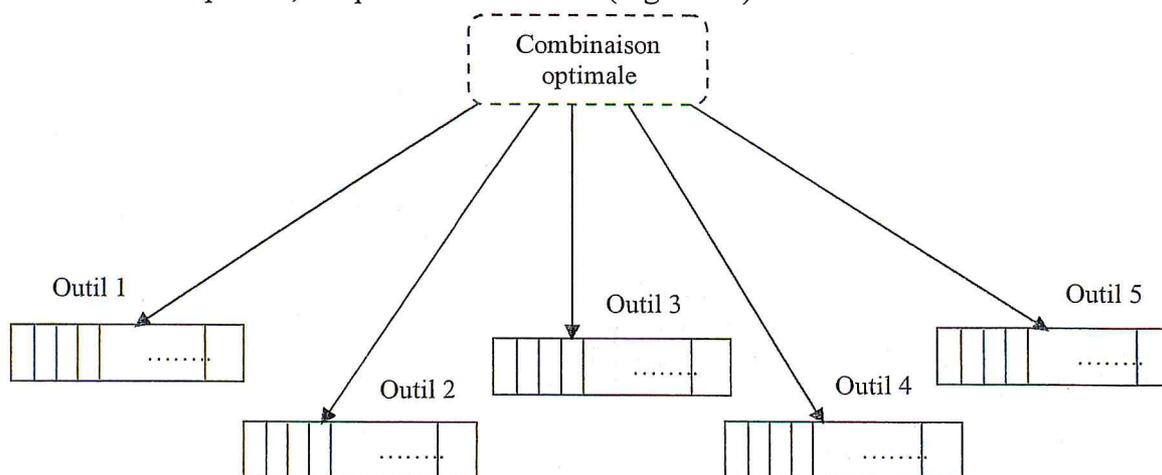


Figure 20 : Création des tables des positions de chaque outils.

La perturbation réalisée consiste à échanger aléatoirement deux positions de la table de chaque outil.

#### IV.5.2. Fonction objective :

La fonction objective dans ce cas est la longueur du chemin reliant toutes les positions de plongées pour chaque outil. La Figure 21 illustre les trajets optimums pour trois outils.

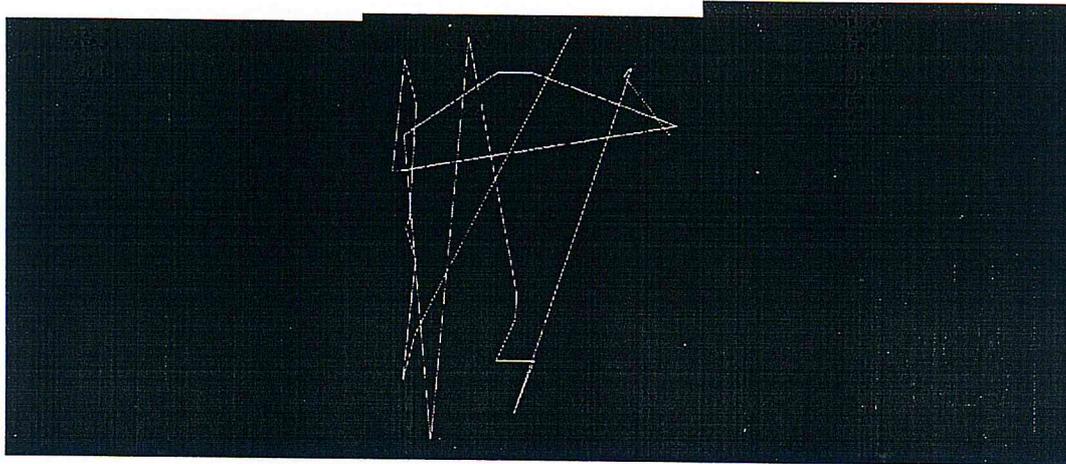


Figure 21 : Trajet optimums.

#### IV.5.3. Génération du trajet d'usinage :

Pour générer le trajet d'outil et le programme d'usinage associé « G-Code », les paramètres suivants doivent être introduits :

1. Distance de sécurité ;
2. Vitesse d'avance ;
3. Surépaisseur d'usinage ;
4. Sens de rotation de l'outil ;
5. Vitesse de broche.

Pendant l'engagement et le déengagement de l'outil, ce dernier se positionne en plusieurs points et doit changer de vitesse à chaque position afin d'éviter de casser l'outil ou bien d'endommager la machine (Figure 22). Pour chaque position de plongée valide, son trajet est généré par l'algorithme suivant :

- Données : liste des positions de plongée valides ;
- Début.
  - Pour chaque position de plongée valide ;
    - ⇒ Déplacement de l'outil jusqu'à  $Z_{max} + \text{distance de sécurité}$  avec une vitesse rapide ;
    - ⇒ Déplacement de l'outil jusqu'à  $Z_{max} + \text{la distance d'engagement}$  avec une vitesse rapide ;
    - ⇒ Engagement de l'outil avec la vitesse d'avance imposée jusqu'à  $Z_{\text{plongée}} + \text{surépaisseur d'usinage}$  ;
    - ⇒ Déengagement de l'outil jusqu'à  $Z_{max} + \text{distance de sécurité}$  avec une vitesse rapide.
- Fin.

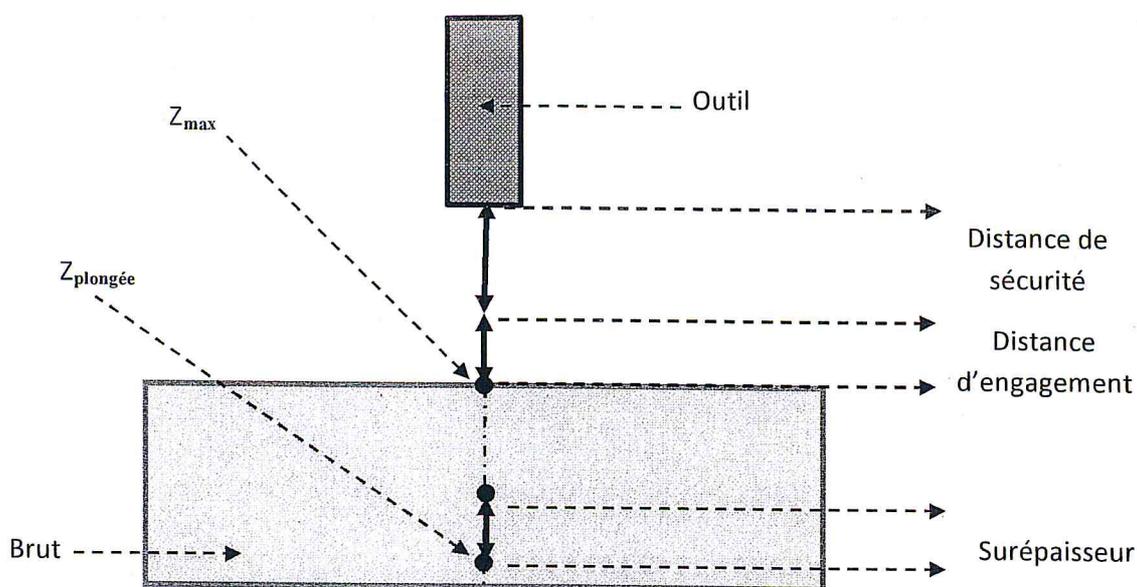


Figure 22 : Points de passage de l'outil pour une position de plongée.

Après la génération du trajet de Tréflage, le programme d'usinage « G-Code » peut être généré et transmis à la machine pour l'usinage réel.

## V. CONCLUSION :

Dans ce chapitre, nous avons présenté les fenêtres que nous avons intégrées à l'application logicielle de CFAO ainsi que les différents algorithmes développés afin d'optimiser l'opération de Tréflage des pièces de n'importe quelle forme en déterminant la combinaison optimale des outils et le chemin le plus court. Le test et la validation de notre application sur des surfaces réelles seront considérés dans le chapitre suivant.

# CHAPITRE IV

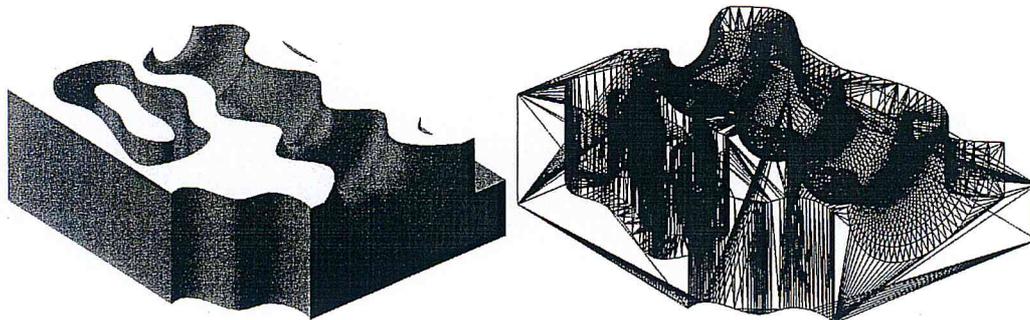
## TEST ET VALIDATION

## I. INTRODUCTION :

Après l'étape d'implémentation et d'intégration des différents modules que nous avons développés, nous passons maintenant à l'étape de la validation de notre travail où nous allons tester notre approche sur un modèle STL d'une pièce. Cette phase permet de détecter les problèmes pouvant apparaître lors de l'exécution pour apporter les corrections nécessaires afin de valider l'application.

## II. TESTS ET VALIDATIONS :

Les tests et les validations sont effectués sur la pièce représentée par la Figure 1.a. Cette pièce est conçue dans un logiciel spécialisé de CAO et elle est composée de plusieurs formes complexes. Dans ce même logiciel, la pièce est approximée par un ensemble de triangles. Ces triangles sont stockés dans un fichier STL (Figure 1.b) qui sera utilisé par la suite.



a. Modèle CAO.

b. Modèle STL.

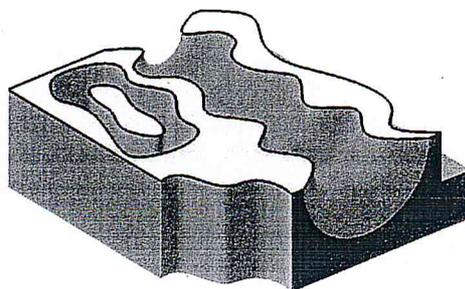
**Figure 1 :** Modèle CAO et modèle STL de la pièce de test.

### II.1. Récupération des points et des triangles et détermination des contours :

Après l'ouverture du fichier STL et la récupération des points et des triangles, nous pouvons visualiser les points, les triangles de l'objet et son brut minimum. Après suppression des points redondants, le nombre de sommets final est égal à 6106 et le nombre de triangles est égal à 11472. Les paramètres du brut sont les suivants :

- $X_{\min}$  est égale à 0 mm,  $Y_{\min}$  est égale à 0mm et  $Z_{\min}$  est égale à 0mm.
- $X_{\max}$  est égale à 140mm,  $Y_{\max}$  est égale à 150mmm et  $Z_{\max}$  est égale à 50mm.
- Dimensions du brut : 150mm×140mm×40mm.

Le calcul des contours sur la face supérieure de la pièce montre l'existence de quatre contours (Figure 2).



**Figure 2 :** Contours de la face supérieure de la pièce.

## II.2. Détermination des triangles des cellules :

Après l'étape précédente, l'utilisateur peut maintenant déterminer les triangles de chaque cellule. La Figure 3 montre les cellules, les triangles des cellules et les triangles de la grande cellule avec un nombre de cellules égale à 10 suivant l'axe X et égale à 10 suivant l'axe Y.

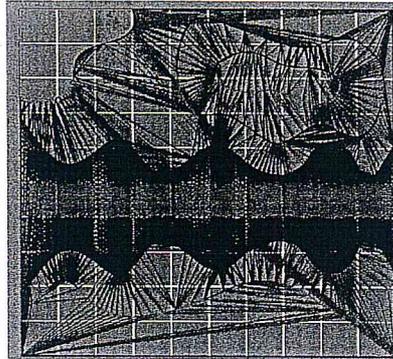
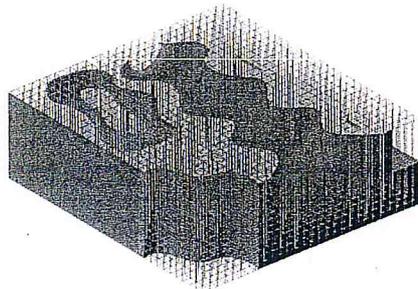


Figure 3 : Cellules, triangles des cellules et triangles de la grande cellule.

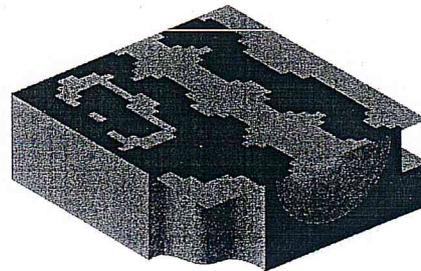
## II.3. Détermination des parties à usiner du modèle STL :

Pour la détermination des parties à usiner et le calcul du volume de la matière à enlever, nous devons passer d'abord par la détermination de la grille représentant les dexels et des paramètres des dexels. L'utilisateur peut visualiser la grille en 2D et en 3D et en même temps il peut distinguer les zones qu'il peut usiner (en vert) et les zones qu'il ne faut pas usiner (en rouge) (Figure 4). Après ces étapes, il est possible de calculer le volume de la pièce finie et le volume à enlever. Pour montrer la précision des résultats, deux cas sont considérés pour les pas des dexels :

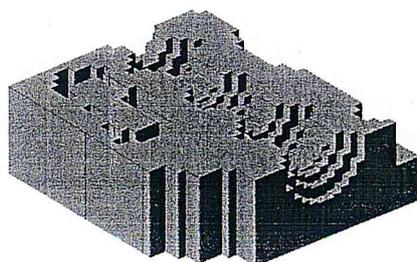
- 1<sup>er</sup> cas : le pas suivant l'axe X est égal à 5 et le pas suivant l'axe Y est égal à 5. Les résultats associés à ces valeurs sont donnés par la Figure 3.



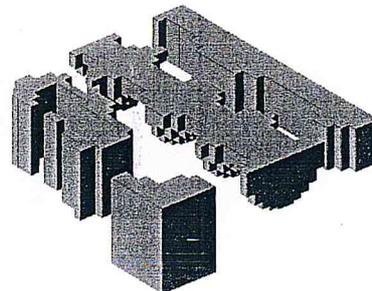
a. Grille en 3D.



b. Type des dexels.



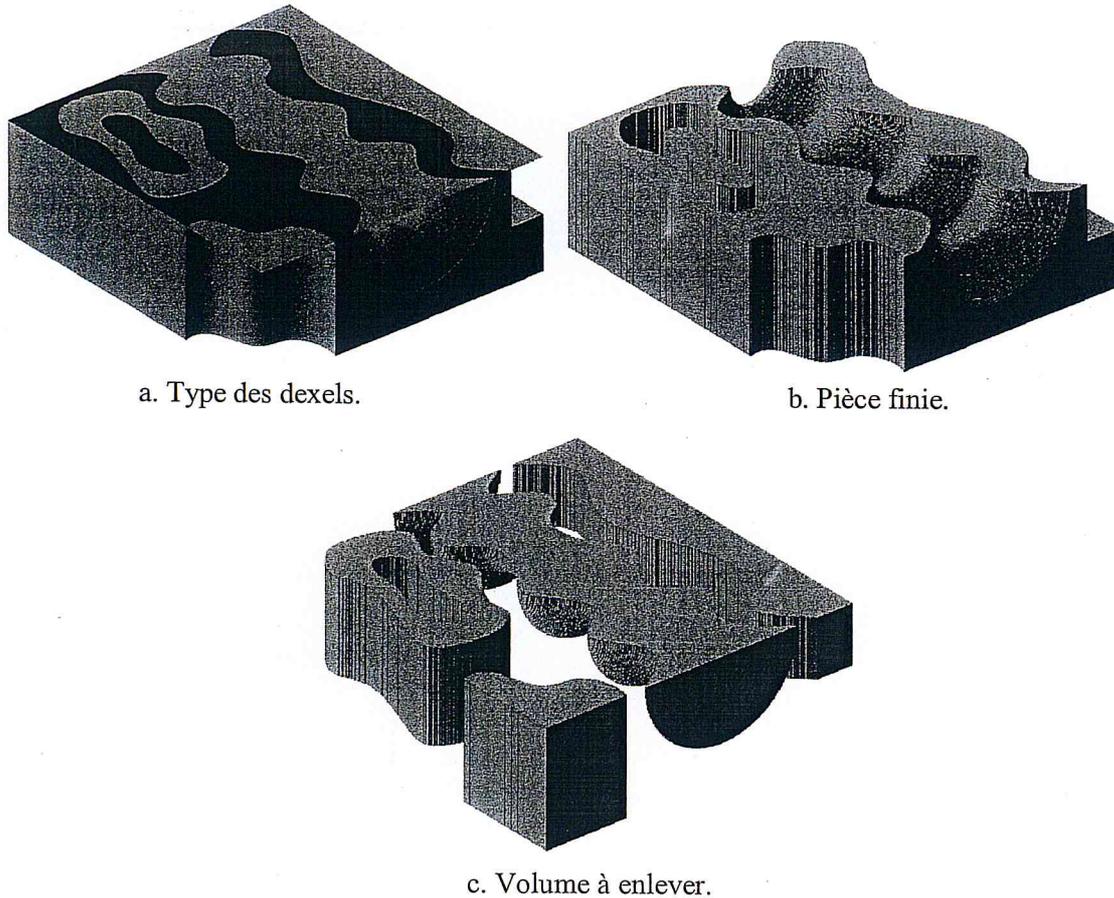
c. Pièce finie.



d. Volume à enlever.

Figure 4 : Approximation de la pièce par des dexels pour le premier cas.

- 2<sup>ème</sup> cas : le pas suivant l'axe X est égal à 0.5 et le pas suivant l'axe Y est égal à 0.5. Les résultats associés à ces valeurs sont donnés par la Figure 5.



**Figure 5 :** Approximation de la pièce par des dexels pour le deuxième cas.

Les résultats des deux cas considérés montrent que plus les dimensions des dexels sont petites (pas suivant l'axe X et pas suivant l'axe Y) plus la pièce finie et le volume à enlever sont obtenus avec une bonne précision.

## II.4. Détermination des rayons utilisables :

### II.4.1. Déterminer du rayon final :

L'identification du rayon final est lancée après la détermination du rayon initial. Pour identifier ce dernier, les points d'intersection de la normale de chaque segment avec la liste des segments sont calculés. Par la suite le point d'intersection le plus proche est sauvegardé. La distance entre ce dernier et le point milieu du segment divisé par deux est le rayon initial de ce segment. A partir du rayon initial, le rayon intermédiaire et le rayon final est déterminé à partir d'une base de données d'outils. La Figure 6 montre les rayons, les points d'intersection et le point le plus proche pour un segment donné de chaque contour.

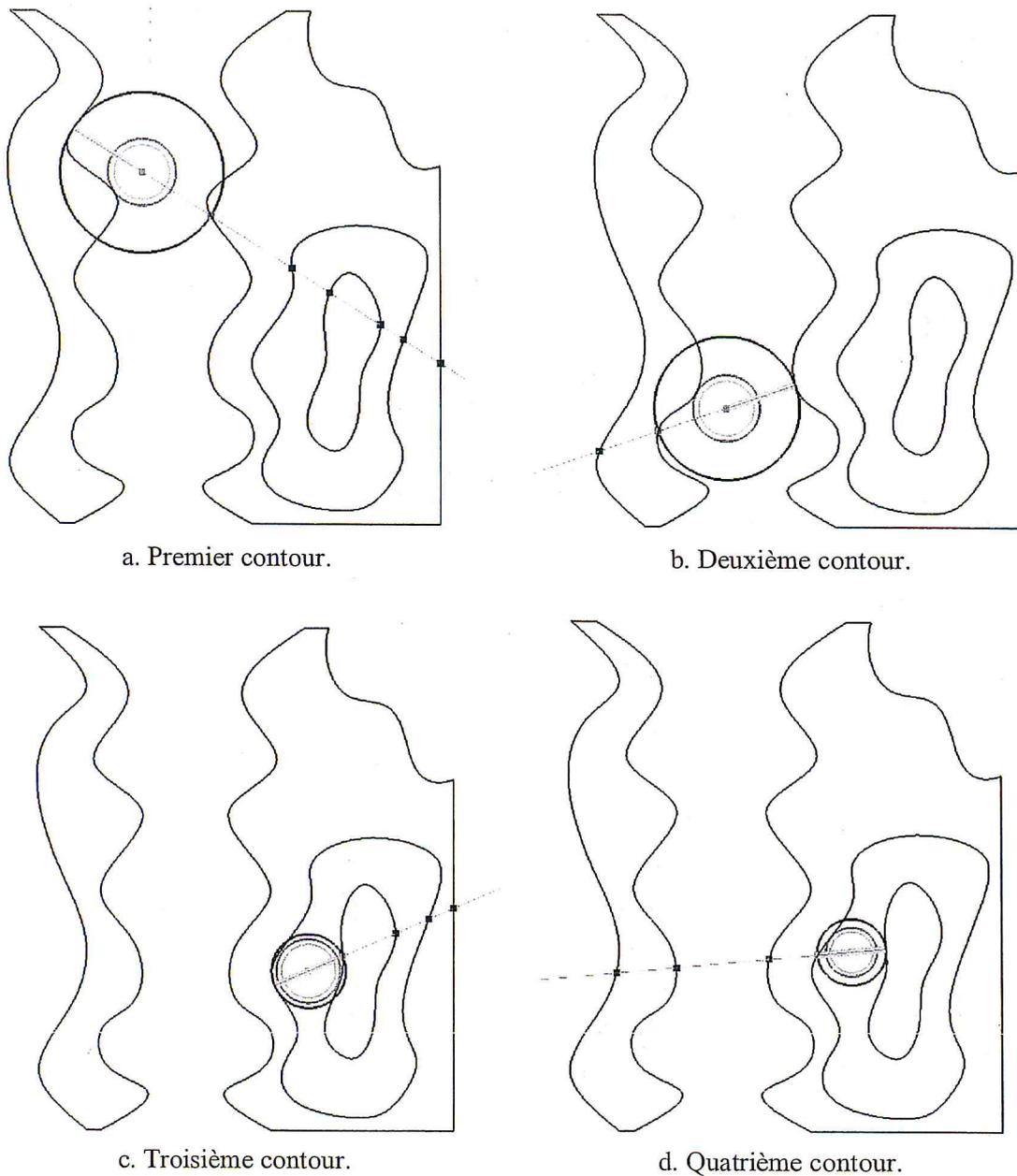
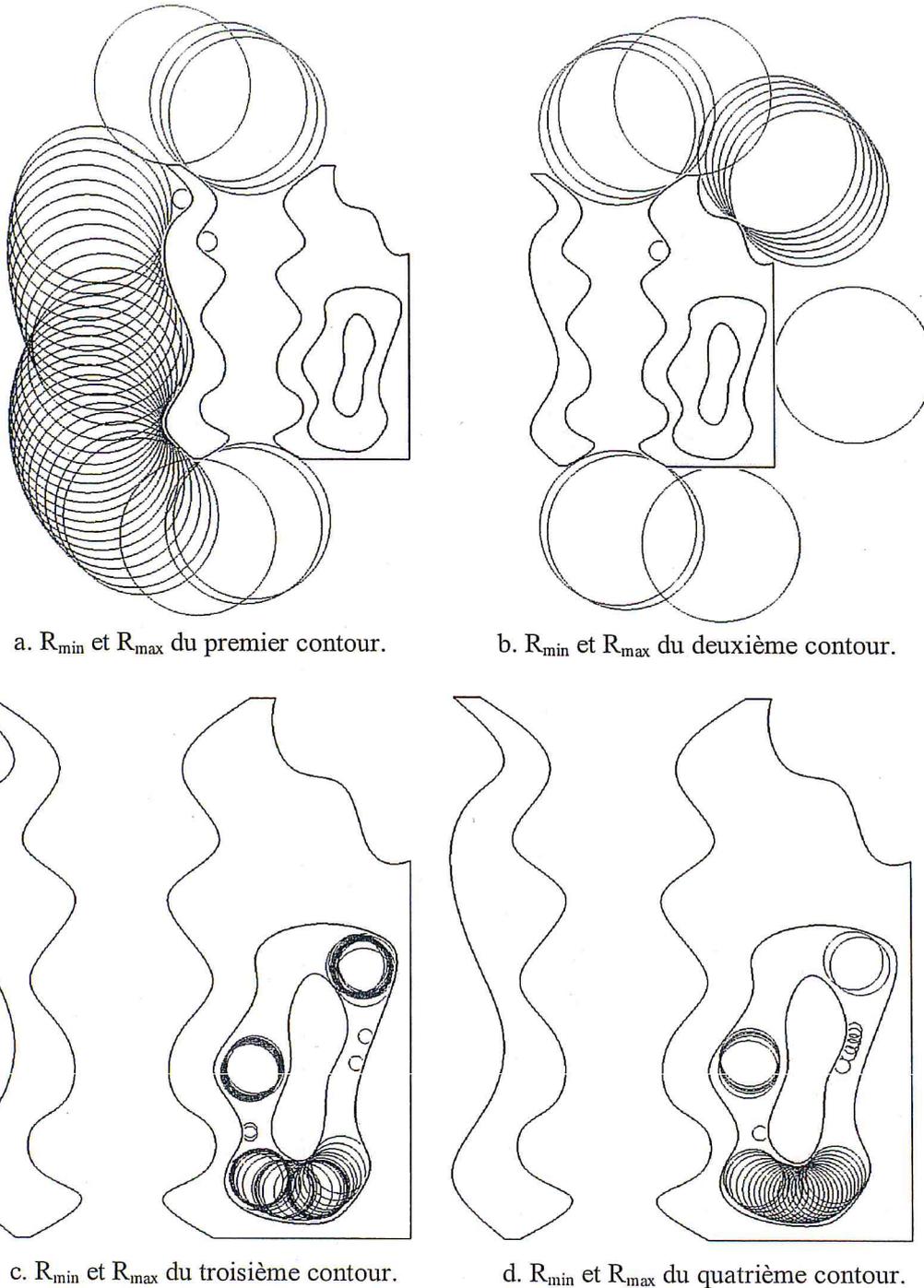


Figure 6 : Détermination des différents rayons.

#### II.4.2. Détermination des rayons minimum et maximum de chaque contour:

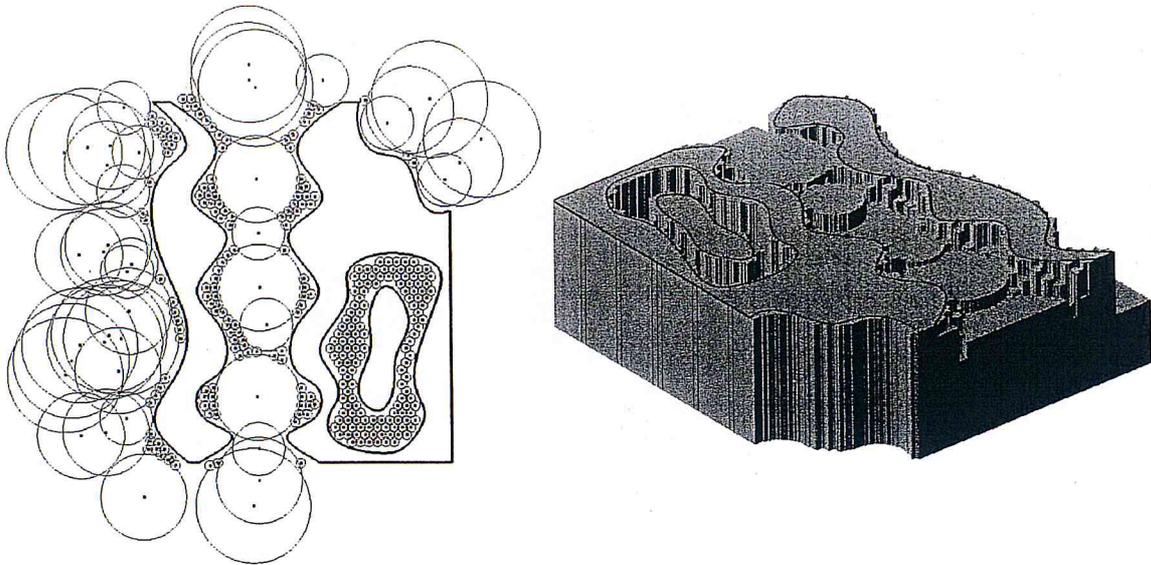
La Figure 7 montre le rayon minimum et le rayon maximum pour chaque contour. Pour le premier et le deuxième contour, les rayons minimum et maximum sont égaux respectivement à 5mm et 40mm; pour le troisième et le quatrième contour, les rayons minimum et maximum sont égaux respectivement à 2mm et 8mm.



**Figure 7 :** Détermination des rayons minimum et maximum de chaque contour.

## II.5. Détermination des combinaisons :

À partir des rayons minimum et maximum de chaque contour, la liste des outils utilisables est créée pour déterminer la liste des combinaisons d'outils possibles pour usiner la pièce. Le nombre d'outils utilisables tirés à partir de la base de données des outils est égal à 17. Par conséquent, le nombre de combinaisons possibles pour choisir 4 outils parmi 16 est égal à 1820. La Figure 8 montre une des combinaisons avec les cercles représentant les rayons d'outils et leurs centres. La combinaison choisie a le numéro 357, le temps d'usinage est égale à 501.87 secondes, le volume enlevé est égale  $231024\text{mm}^3$  avec un pourcentage d'enlèvement de matière égal à 70.83%.



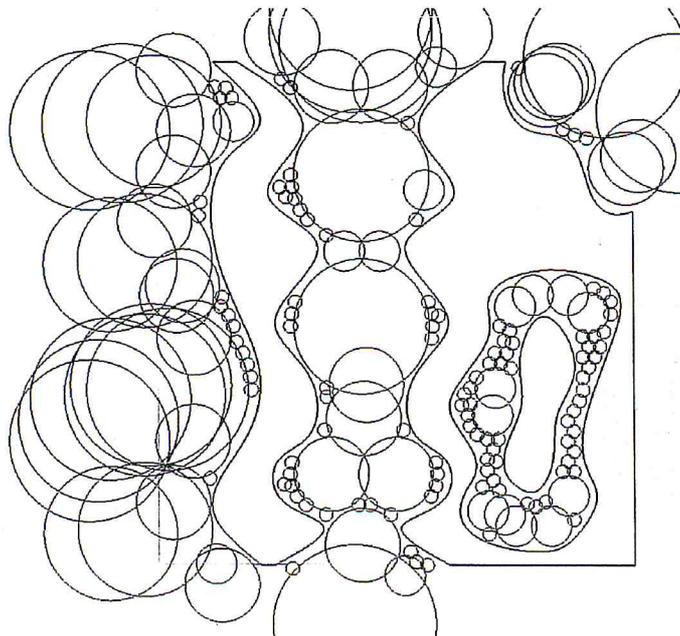
a. Positions de plongée des outils.

b. Forme de la pièce usinée.

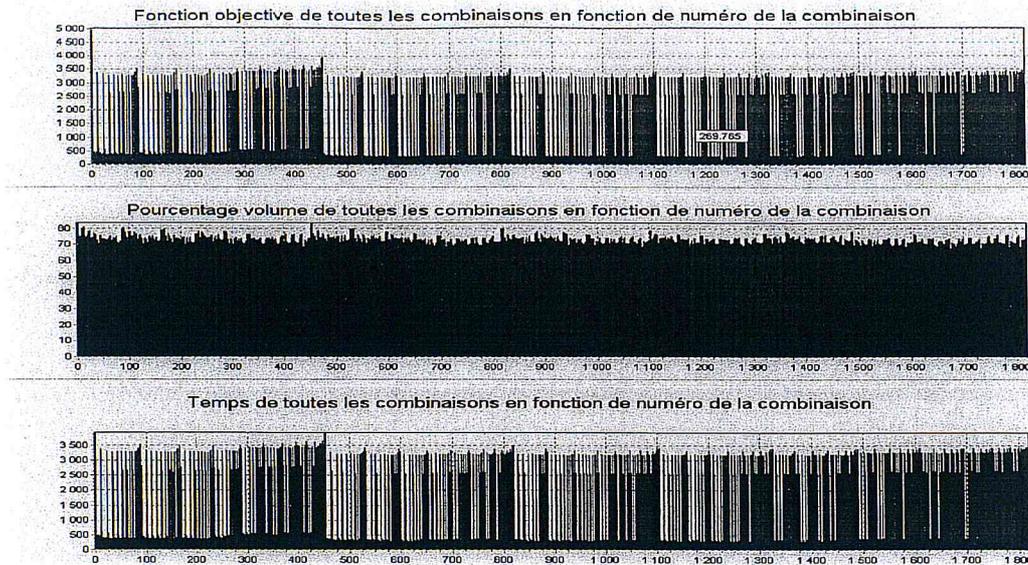
**Figure 8** : Positions de plongées et forme de la pièce pour une combinaison.

## II.6. Détermination de la combinaison optimale théorique :

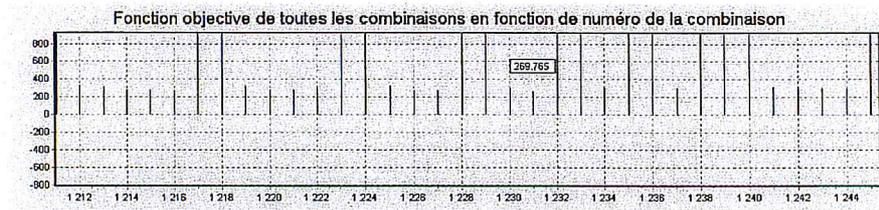
Pour déterminer la combinaison optimale théorique enlevant le maximum de la matière non nécessaire en un minimum de temps, le calcul est lancé pour toutes les combinaisons possibles (1820 combinaisons). Le temps de calcul pour déterminer cette combinaison est très important. La combinaison trouvée a un temps d'usinage égal à 269.76 minutes et un pourcentage d'enlèvement de matière égal à 68.79% (Figure 9.a). La Figure 9.b donne une représentation graphique du volume enlevé, du temps d'usinage et de la fonction objective pour toutes les combinaisons. La Figure 9.c donne la position de la combinaison optimale théorique (combinaison numéro 1231). De même, la combinaison numéro 455 permet d'enlever le maximum de volume et la combinaison numéro 1232 nécessite un temps minimum d'usinage (Figure 9.d).



a. Positions des rayons de la combinaison optimale exacte.



b. Présentation graphique de toutes les combinaisons.



c. Fonction objective de la combinaison optimale exacte.

Combinaisons de volume max et temps min	
<b>Volume max</b>	
N° combinaison	155
Outil 1	40
Outil 2	32
Outil 3	27
Outil 4	2
Outil 5	2
Volume	273067.697
Temps	2 jours 17 heures
<b>Temps min</b>	
N° combinaison	1232
Outil 1	24
Outil 2	20
Outil 3	11
Outil 4	6
Outil 5	2
Volume	224353.581
Temps	0 jours 4 heures

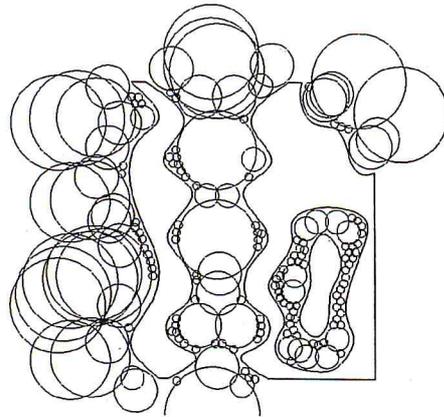
d. Combinaison du temps minimum et combinaison du volume maximum.

Figure 9 : Combinaison optimale exacte.

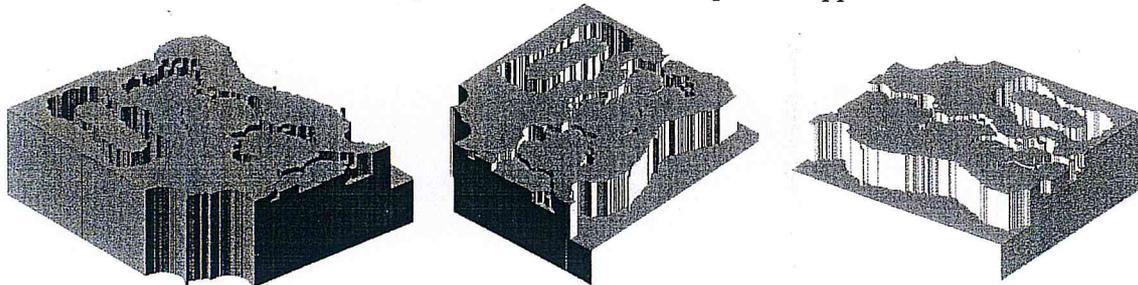
Pour réduire le temps de calcul et pour ne pas lancer les calculs pour toutes les combinaisons possibles, l'algorithme du recuit simulé est utilisé. Dans cet exemple, les paramètres du recuit sont suivants :

- Température initiale est égale à 100.
- Température minimale est égale à 10.
- Nombre d'itérations par palier est égal à 5.
- Lambda est égal à 0.99.
- Pondération du volume est égale à 0.5.
- Pondération du temps est égale à 0.5.

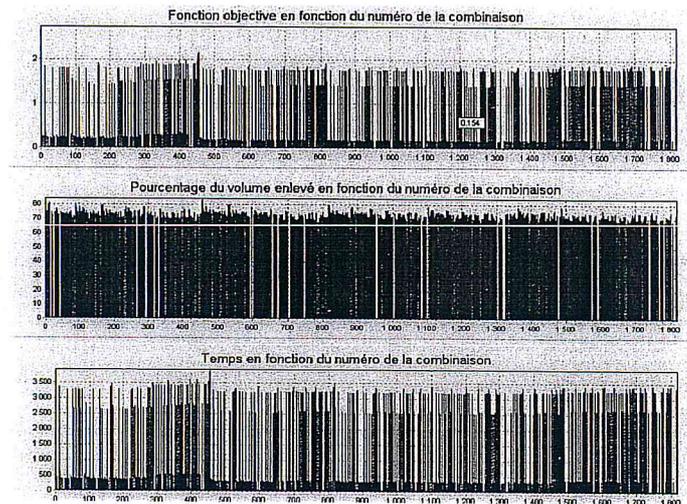
Avec cet algorithme, le nombre de combinaisons parcourues est égal à 1150 et la détermination de la combinaison optimale a nécessité 28 minutes (Figure 10). Le temps d'usinage et pourcentage d'enlèvement de matière associés à la combinaison trouvée sont égaux respectivement à 269.76 minutes et 68.79%.



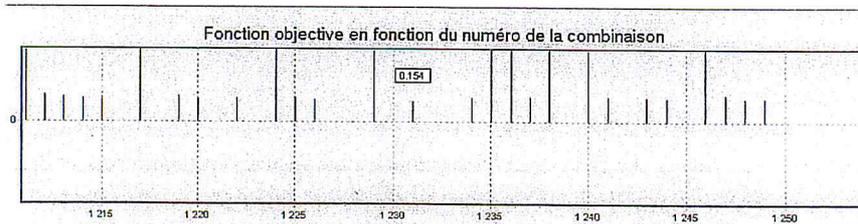
a. Positions des rayons de la combinaison optimale approchée.



b. Pièce usinée.



c. Présentation graphique de la combinaison optimale trouvée.



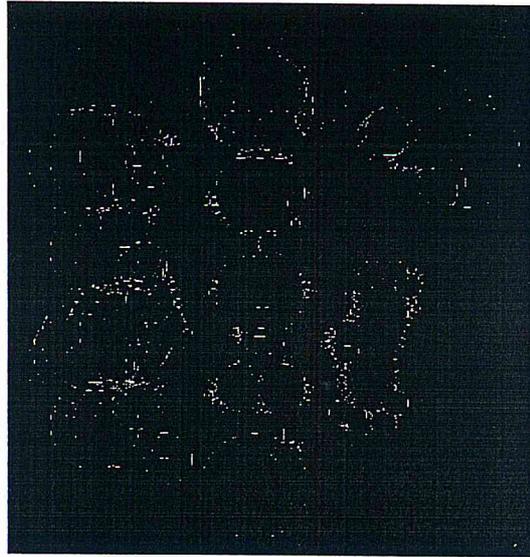
d. Fonction objective de la combinaison optimale approchée.

**Figure 10** : Combinaison optimale approchée.

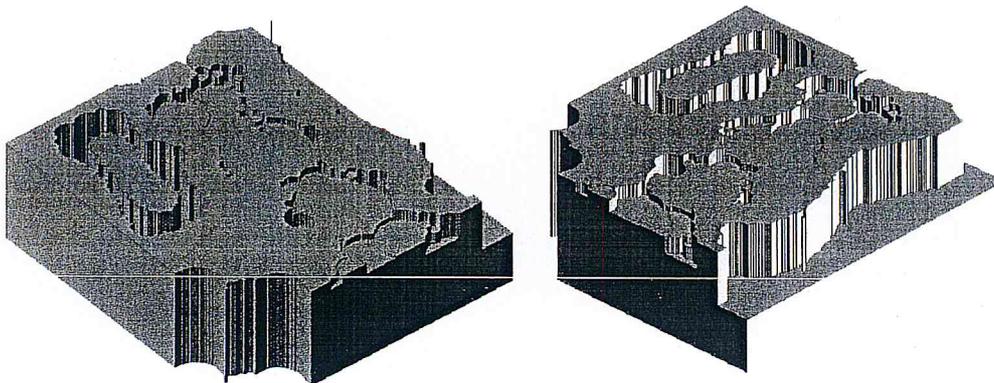
Par la suite, nous avons changé les pondérations dans le recuit simulé :

- Pondération du volume est égale à 0.7.
- Pondération du temps est égale à 0.3.

Les résultats obtenus à ce changement sont donnés par la Figure 11.



a. Positions des rayons de la combinaison optimale approchée.



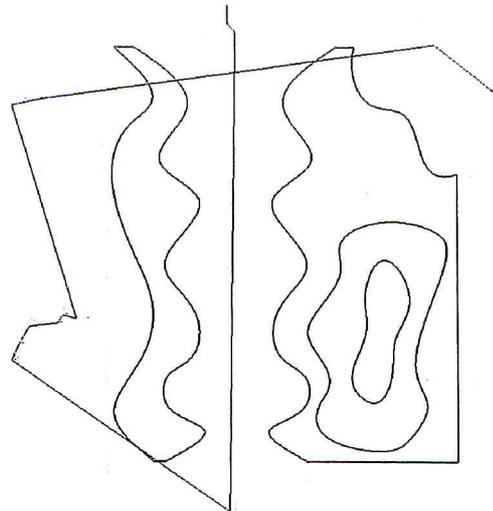
b. Pièce usinée.

Figure 11 : Combinaison optimale approchée.

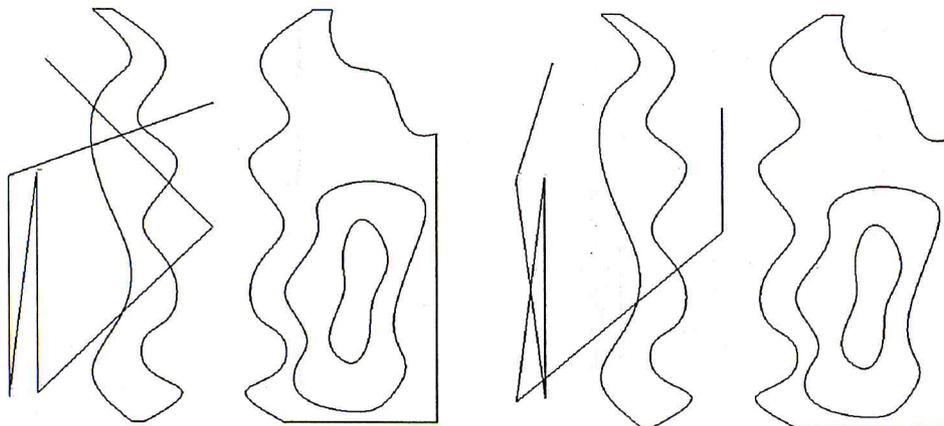
## II.7. Détermination du trajet d'usinage :

### II.7.1. Optimisation du trajet d'usinage :

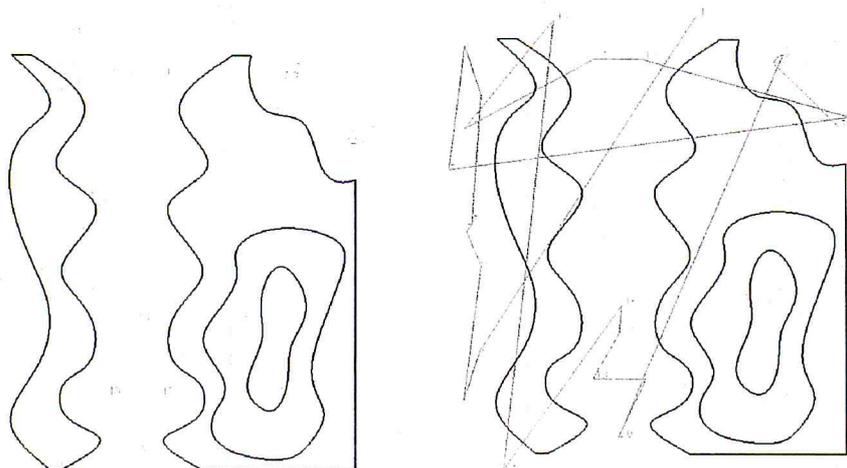
L'algorithme du recuit simulé est appliqué une deuxième fois au trajet d'usinage pour déterminer le chemin d'outils le court pour le premier outil, le deuxième outil, le troisième outil, le quatrième outil et le cinquième outil. La Figure 12 représente les trajets des cinq outils avant et après optimisation. Le déroulement de l'algorithme permet d'optimiser les trajets associés aux outils 2, 3 et 4 tandis que pour les outils 1 et 5 pas de changement des trajets.



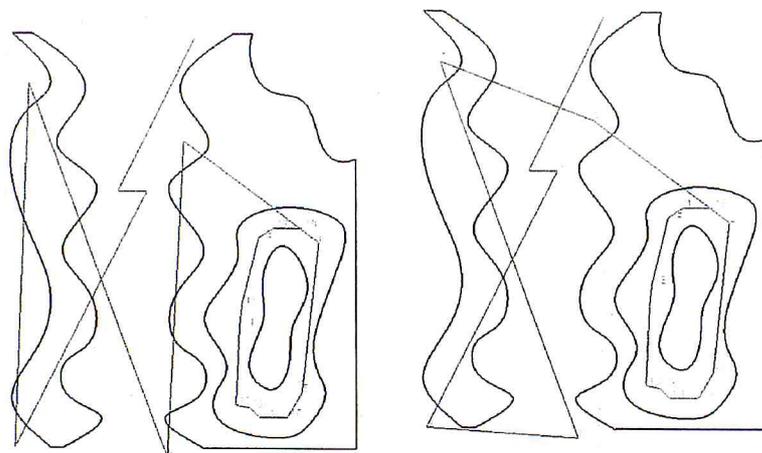
a. Trajet outil 1 avant et après optimisation.



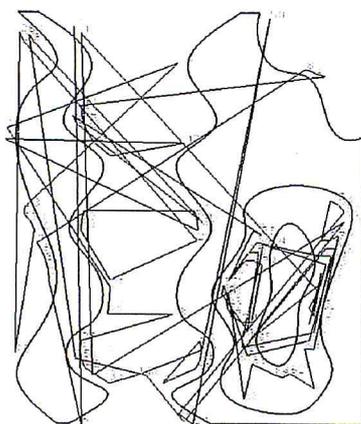
b. Trajet outil 2 avant optimisation (gauche) et après optimisation (droite).



c. Trajet outil 3 avant optimisation (gauche) et après optimisation (droite).



d. Trajet outil 4 avant optimisation (gauche) et après optimisation (droite).



e. Trajet outil 5 avant et après optimisation.

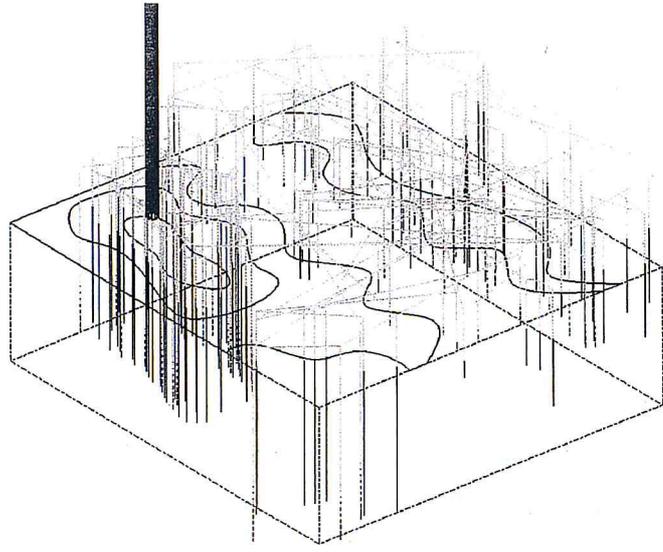
**Figure 12** : trajet d'usinage avant et après optimisation.

### II.7.2. Génération du trajet d'usinage :

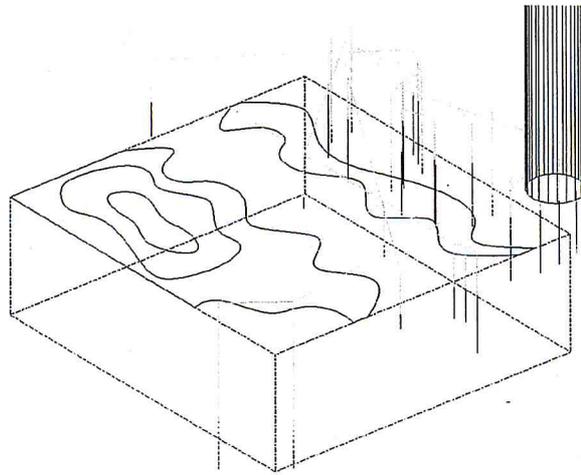
Après avoir déterminé les plongées de l'outil et le trajet optimal, il est possible de passer à la génération du trajet de Tréflage. Pour cela, les paramètres suivants sont définis :

- Distance de sécurité est égale à 20mm.
- Distance d'engagement est égale à 2 mm.

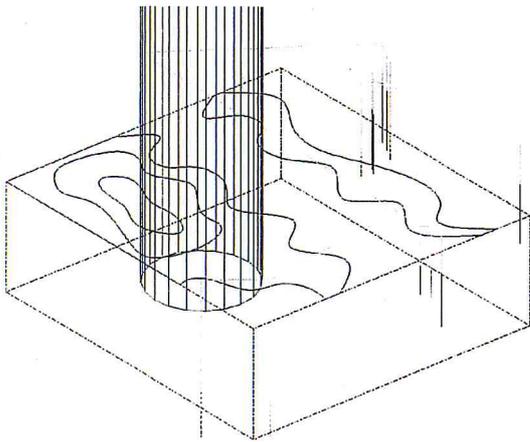
A partir de ces paramètres, le trajet de Tréflage est généré et la simulation des mouvements des outils peut être lancée sous trois formes : point à point, séquence par séquence et globale. La Figure 13 montre ces trois formes. La Figure 14 montre la simulation virtuelle de l'usinage à partir du trajet généré.



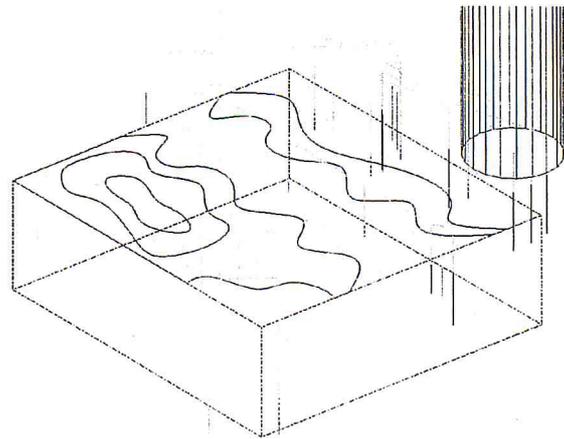
a. Trajet d'usinage global.



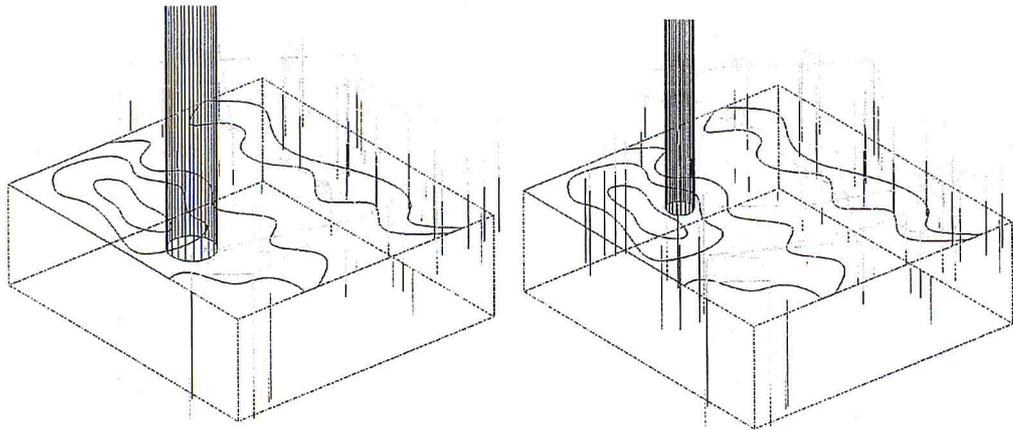
b. Simulation virtuelle du trajet d'usinage points par points.



c. Séquences du premier outil.

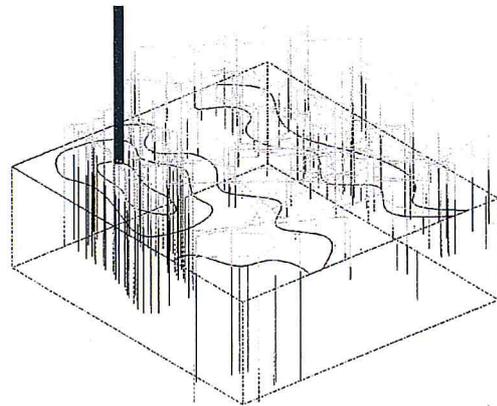


d. Séquences du deuxième outil.



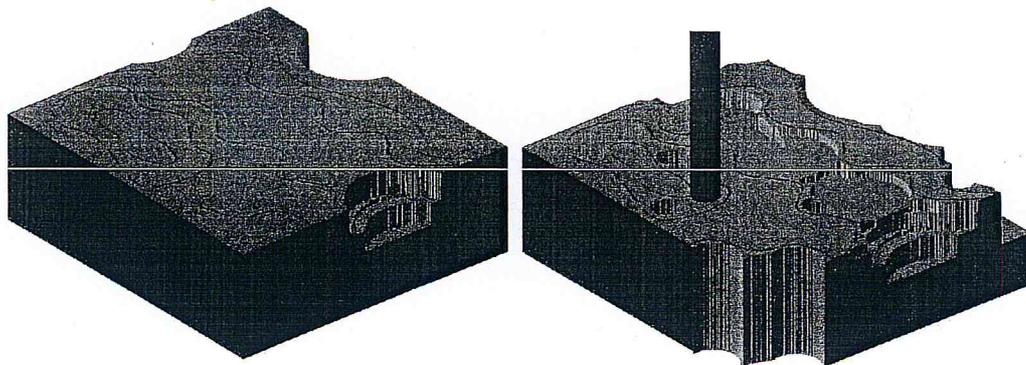
e. Séquences du troisième outil.

d. Séquences du quatrième outil.

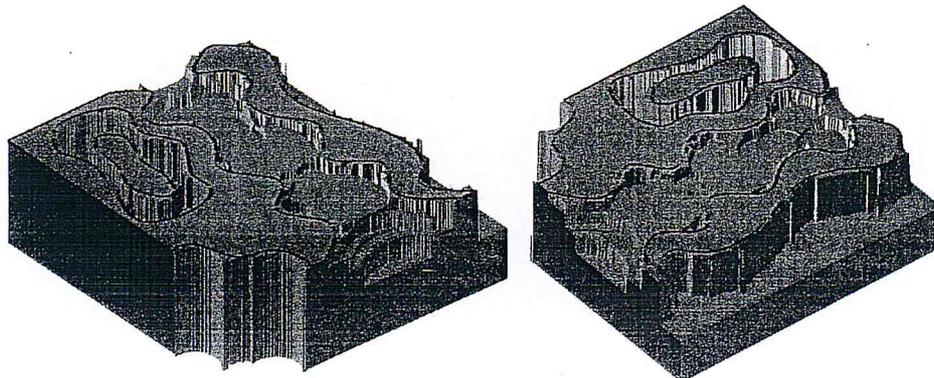


e. Séquences du cinquième outil.

**Figure 13 : Simulation virtuelle des mouvements d'outils.**



a. Outil pendant la simulation de l'usinage.



b. Pièce usinée.



**CONCLUSION**

**GENERALE**

## CONCLUSION GENERALE

Le projet que nous avons présenté dans ce mémoire consiste en la conception et le développement d'une application logicielle graphique et interactive de FAO permettant l'optimisation de l'opération de Tréflage des pièces de n'importe quelle forme à partir de leurs modèles STL sur des fraiseuses à commande numérique à 03 axes. Cette application permet de déterminer la combinaison optimale des outils permettant d'enlever le maximum de matière en un minimum de temps tout en évitant les problèmes d'interférences et de collisions, les positions de plongée valides et les profondeurs de plongée associées, le chemin le plus court reliant les positions de plongée valides et le programme d'usinage « G-Code ».

Lors de la réalisation de ce projet, nous avons commencé par une étude bibliographique sur le format d'échange de données « STL ». Ensuite, nous avons présenté des généralités sur le processus d'usinage des surfaces gauches ainsi que les différentes stratégies d'ébauchage. Par la suite, l'accent est mis sur la stratégie du Tréflage. A la fin de notre mémoire, nous avons présenté la conception, les algorithmes développés et l'implémentation informatique de notre application logicielle ainsi que des tests de validation.

Les principaux résultats de notre application logicielle sont les suivants :

- Création du modèle en « dexels » de la pièce à usiner,
- Détermination des types des « dexels »,
- Identification de la matière à enlever,
- Calcul du rayon initial, du rayon théorique et du rayon final à associer à chaque segment des contours de la face supérieure,
- Détermination de la liste des outils utilisables,
- Calcul de toutes les combinaisons possibles des outils,
- Calcul des positions de plongée valides, du volume enlevé et du temps d'usinage pour toutes les combinaisons d'outils,
- Présentation graphique des résultats,
- Détermination des combinaisons optimales théoriques à partir de toutes les combinaisons d'outils,
- Détermination des combinaisons optimales en appliquant la méthode du recuit simulé,
- Détermination du trajet d'usinage optimal pour chaque outil en utilisant la méthode du recuit simulé,
- Génération du trajet de Tréflage,
- Simulation virtuelle des mouvements des outils et de l'usinage,
- Génération du programme d'usinage « G-Code ».

En perspective de notre travail, nous recommandons de traiter les thématiques suivantes :

- Sélection des conditions de coupe les plus optimales (vitesse d'avance et vitesse de coupe),
- Proposition de méthodes pour uniformiser l'épaisseur de la matière restante,
- Tréflage des surfaces gauches définies par un nuage de points quelconque,
- Raccordement des points de plongée successifs par des arcs de cercles,
- Evitement des frottements entre l'outil et les parois de la pièce dans la phase de remontée de l'outil (dégagement de l'outil suivant une direction inclinée par rapport à l'axe Z de la machine, dégagement suivant deux demi-cercles, ...etc.)
- Intégration de la durée de vie des outils lors de l'optimisation,
- Détermination du nombre minimum d'outils permettant d'enlever le maximum de matière en un minimum de temps,
- Application de d'autres méthodes d'optimisation.

**REFERENCES BIBLIOGRAPHIQUES**

### REFERENCES BIBLIOGRAPHIQUES

1. P. Lefebvre, B. Lauwers, « STL model segmentation for multi axis machining operations planning », Computer-Aided Design and Applications, pages 277-284, 2004.
2. B. Aguenini, H. Messaoudi, « Simulation et vérification de l'opération d'ébauchage des surfaces gauches sur des fraiseuses à commande numérique à 3 axes » PFE, Département d'Informatique, Université Saad Dahlab, Blida, 2005/2006.
3. B. Boukercha, S. Kaouane « Vers l'approche contours décalés dans le processus d'usinage des surfaces libres en ébauche » PFE, Département d'Informatique, Université Saad Dahlab, Blida, 2008/2009.
4. Z. Tchantchane, « Automatisation de l'opération de Tréflage des surfaces gauches à partir d'un model STL » université Amar Telidji, Laghouat, 2010/2011.
5. A. Toumine, « Usinage par enlèvement de copeaux », cours de fabrication, INSA, Lyon, France.
6. L. Kheidri, N. Benhenda, « Automatisation de l'Opération de Tréflage des Surfaces » PFE, Département d'Informatique, Université Saad Dahlab, Blida, 2009/2010.
7. M. Al-Ahmad, « "Industrialisation de procédé : contribution à la maîtrise de l'opération de tréflage ou fraisage vertical - approche analytique et expérimentale » », Thèse de Doctorat, ENSAM, École Nationale Supérieure d'Arts et Métiers, Metz, France, 2008.
8. M. Raucha et J. Y. Hascoet, « "Génération de trajectoires de tréflage et d'usinage trochoïdal pour le vidage de poche » », Mécanique & Industries, Vol. 8, pp 445-453, 2007.
9. H. Tawfik, « "A new algorithm to calculate the optimal inclination angle for filling of plunge-milling » », International Journal of CAD/CAM, Vol.6, 2006.
10. M. Al-Ahmad, A. D'Acunto, P. Martin, « "Influence de différentes stratégies de tréflage sur la qualité de surface et la productivité » », 18ème Congrès Français de Mécanique, CFM, Grenoble, France, 2007.
11. F. Girardin, N. Guillemot, « Maîtrise de l'opération de Tréflage », Rapport de stage, France, Session 2004-2005.

## Liste des figures

### Chapitre 1 :

Figure 1 : Format STL d'une surface théorique .....	3
Figure 2 : Fichier « STL » et paramètres d'un triangle .....	4
Figure 3 : Types de fraiseuses.....	5
Figure 4 : Types de fraises .médiation.....	6
Figure 5 : Problèmes d'interférence et de collision.....	6
Figure 6 : Ebauchage avec un outil cylindrique.....	7
Figure 7 : Stratégie des « Plans Parallèles ».....	7
Figure 8 : Modes de balayage de l'outil.....	8
Figure 9 : Stratégie des « Contours Décalés ».....	8
Figure 10: Zones non usinées avec les « Contours Décalés ».....	8
Figure 11: Opération de Tréflage.. ..	9
Figure 12: Diagramme de cas d'utilisation globale.....	9
Figure 13: Paramètres limites de coupe.....	10
Figure 14: Types de plaquettes.....	10
Figure 15: Paramètres pilotant l'opération de Tréflage.....	11
Figure 16: Mouvements de l'outil en Tréflage.....	11
Figure 17: Trajectoire d'une dent dans la matière.....	12
Figure 18: Modes de dégagement de l'outil en Tréflage.....	13
Figure 19: Modes de balayage de l'outil .....	14
<b>Chapitre 2 :</b>	
Figure 1 : Diagramme cas d'utilisation global.....	19

Figure 2 : Diagramme cas d'utilisation du traitement du fichier STL.....	19
Figure 3 : Diagramme cas d'utilisation du traitement du fichier STL.....	19
Figure 4 : Diagramme cas d'utilisation de la détermination des rayons finaux.....	20
Figure 5 : Diagramme cas d'utilisation de détermination de la combinaison optimale .....	20
Figure 6 : Diagramme cas d'utilisation de détermination du trajet d'usinage optimal .....	20
Figure 7 : Diagramme de séquence de traitement du fichier STL.....	21
Figure 8 : Diagramme de séquence de génération des contours de la face supérieure...	21
Figure 9 : Diagramme de séquence de calcul des rayons.....	22
Figure 10: Diagramme de séquence déterminant la combinaison optimale.....	22
Figure 11: Diagramme de séquence de la détermination du plus court chemin.....	23
Figure 12: Diagramme de collaboration.....	24
Figure 13: Diagramme de classe.....	26
Figure 14: Classe liste point ébauchage STL.....	27
Figure 15: Classe liste triangle ébauchage STL.....	27
Figure 16: Classe Combinaison outil.....	28
Figure 17: Classe Intervalle outil .....	28
Figure 18: Classe position plongé.....	28
Figure 19: Classe liste points plongée.....	29
Figure 20: Classe Centre-rayon.....	29
Figure 21: Classe usiner_dexel.....	30

**Chapitre 3 :**

Figure 1 : Fenêtre principale.....	31
Figure 2 : Accès à la rubrique « Ebauchage par Tréflage (modèle STL) ».....	32
Figure 3 : Fenêtre de traitement du fichier STL.....	32
Figure 4 : Triangles des cellules et de la cellule supplémentaire.....	33
Figure 5 : Création des limites de chaque cellule.....	33
Figure 6 : Création de la grille des dexels.....	34
Figure 7 : Hauteur d'un dexel.....	34
Figure 8 : Appartenance d'un point à un triangle non vertical.....	35
Figure 9 : Appartenance d'un point à un triangle vertical.....	35
Figure 10: Fenêtre de détermination des rayons utilisables.....	36
Figure 11: Intersection de deux segments.....	37
Figure 12: Rayon initial.....	37
Figure 13: Rayon initial, théorique et final lors d'existence de points d'intersection....	38
Figure 14: Rayon initial, théorique et final lors d'absence de points d'intersection.....	39
Figure 15: Fenêtre de calcul du volume enlevé et du temps d'usinage.....	39
Figure 16: Positions de plongée pour des cercles de mêmes rayons.....	41
Figure 17: Positions de plongée pour des cercles de rayons différents.....	42
Figure 18: Fenêtre de détermination de la combinaison optimale.....	43
Figure 19: Fenêtre d'optimisation du trajet d'usinage.....	47
Figure 20: Création des tables des positions de chaque outils.....	47

Figure 21: Trajet optimums.....	48
Figure 22: Points de passage de l'outil pour une position de plongée .....	49

#### **Chapitre 4**

Figure 1 : Modèle CAO et modèle STL de la pièce de test.....	50
Figure 2 : Contours de la face supérieure de la pièce.....	50
Figure 3 : Cellules, triangles des cellules et triangles de la grande cellule.....	51
Figure 4 : Approximation de la pièce par des dexels pour le premier cas.....	51
Figure 5 : Approximation de la pièce par des dexels pour le deuxième cas.....	52
Figure 6 : Détermination des différents rayons.....	53
Figure 7 : Détermination des rayons minimum et maximum de chaque contour.....	54
Figure 8 : Positions de plongées et forme de la pièce pour une combinaison.....	55
Figure 9 : Combinaison optimale exacte.....	56
Figure 10: Combinaison optimale approchée.....	57
Figure 11: Combinaison optimale approchée.....	58
Figure 12: trajet d'usinage avant et après optimisation.....	60
Figure 13: Simulation virtuelle des mouvements d'outils.....	62
Figure 14: Simulation virtuelle de l'usinage.....	63
Figure 15: Génération du programme d'usinage « G-Code ».....	63