République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur

Et de la Recherche Scientifique

UNIVERSITE SAAD DAHLAB DE BLIDA

FACULTE DES SCIENCES

DEPARTEMENT INFORMATIQUE

MEMOIRE DE FIN D'ETUDE

Pour l'obtention

d'un diplôme de Master en Informatique

Option : Systèmes Informatiques et Réseaux

Thème

Ebauchage de Pièces Complexes par la Stratégie « Contours Décalés » sur Fraiseuses Numériques à 03-Axes

Réalisé par :

Melle. MENOUS Meriem

Melle. KEMMOUM Nour El Houda

Soutenu devant :

Mr. BEY Mohamed	CDTA	Encadreur
Melle. FERHAT Sahla	CDTA	Encadreuse
Mr. HAMOUDA Mohamed	USDB	Promoteur
Mr. KAMECHE Abdallah	USDB	Président
Mme. ZAHRA Fatma Zohra	USDB	Examinatrice

2020/2021

Résumé :

Ce travail s'insère dans le cadre de développement d'une plateforme logicielle pour la production des pièces de formes complexes, sur des fraiseuses numériques multiaxes, 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).

Dans ce projet nous nous intéressons à l'ébauchage des pièces complexes, définies par leurs modèles « STL » sur des fraiseuses numériques à 03-axes, en considérant la stratégie d'usinage « Contour Décalés ». Il s'agit de concevoir, de développer et d'intégrer à la plateforme logicielle de l'équipe « CFAO » un module logiciel graphique et interactif permettant de déterminer les outils optimums évitant les problèmes d'interférences et de collisions, leurs combinaisons et le trajet d'outils global représentant la succession des points de positionnement de l'outil par rapport à la pièce. L'objectif est atteint, le trajet d'outil est générer à partir des décalages successifs.

Mots Clés : Pièce Complexe, Modèle STL, Contours Décalés, Trajet d'outils, Ebauchage, Interférence, Collision, Fraiseuse 03-axes.

Abstract:

This work is part of a software platform for the production of parts of complex shapes, on multi-axis CNC milling machines, initiated by the Computer Aided Design and Manufacturing (CFAO) team of the Production and Robotics Division (DPR) of the Centre for the Development of Advanced Technologies (CDTA).

In this project, we are interested in the roughing of complex parts, defined by their "STL" models on 03-axis CNC milling machines, considering the "Offset Contour" machining strategy. This involves designing, developing and integrating into the software platform of the "CFAO" team a graphical and interactive software module making it possible to determine the optimum tools avoiding interference and collision problems, their combinations and the global tool path representing the succession of tool positioning points relative to the part. The objective is reached, the tool path is generated from the successive offsets.

Keywords: Complex Part, STL Model, Offset Contours, Tool Path, Roughing, Interference, Collision, 03-axis CNC Milling Machine.

ملخص:

هذا العمل هو جزء من تطوير منصة برمجية لإنتاج قطع ميكانيكية ذات الأشكال المعقدة، على آلات التفريز الرقمية متعددة المحاور، والتي بدأها فريق التصميم والتصنيع بمساعدة الكمبيوتر التابع لشعبة الإنتاج والروبوتات (DPR) في مركز تطوير التقنيات المتقدمة (CDTA).

الخاصة بها على "STL" في هذا المشروع، نحن مهتمون بتخشين الأجزاء المعقدة، المحددة من خلال نماذج ، مع الأخذ في الاعتبار "Contours Décalés"التصنيع آلات التفريز الرقمية ذات 03 محاور، بواسطة استراتيجية وحدة "CFAO" يتضمن ذلك تصميم وتطوير ودمج منصة برمجية لفريق ."Offset Contour" استراتيجية التصنيع .برمجية تفاعلية و تجعل من الممكن تحديد الأدوات المثلى لتجنب مشاكل التداخل والاصطدام ومجموعاتها ومسار الأداة يتم الوصول إلى الهدف الكلي، ويتم إنشاء مسار الأداة من عمليات الإزاحة المتتالية

الكلمات المفتاحية: قطع معقدة، نموذجSTL ، " Contours Décalés "، مسار الأداة، تخشين، تشطيب، التداخل، التصادم، آلة تفريز ذات 03 محاور

Remerciements

En tout premier lieu, nous remercions Allah le tout Puissant et le Miséricordieux de nous avoir donné la santé, la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés et la volonté d'entamer et d'achever ce modeste travail.

Al Hamdoulilah

Nous tenons à exprimer nos vifs remerciements avec un grand plaisir et un grand respect à notre encadreur, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide de Monsieur BEY MOHAMED, nous le remercions par la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant notre préparation du mémoire.
Nous sommes profondément reconnaissantes envers vous pour votre compétence, vos qualités, votre sérieux et vos encouragements.
Nous avons eu l'honneur et la chance de côtoyer au quotidien au sein de votre équipe et de travailler avec vous. Veuillez bien Monsieur recevoir nos sincères remerciements pour votre bonne volonté d'accepter de nous encadrer.

Nos remerciements s'adressent aussi à Melle FERHAT Sahla, notre encadreuse pour son aide pratique et son soutien moral et ses encouragements afin de compléter ce travail. Nous sommes très heureuses d'avoir la chance de travailler avec vous.

Nos remerciements s'adressent également à notre promoteur Monsieur HAMOUDA Mohamed pour son aide et son soutien malgré ses charges académiques et professionnelles.

Nous tenons à exprimer vivement nos remerciements avec une profonde gratitude à toutes les personnes qui ont contribué de près ou de loin à sa réalisation, car un projet ne peut pas être le fruit d'une seule personne.

Dédicace

Louange à Dieu tout puissant, qui m'a permis de voir ce jour tant attendu

Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont très chers :

A MES TRES CHERS PARENTS

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consentis pour mon instruction et mon bien être.
Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.
Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices. Puisse Dieu, le très Haut vous accorde santé, bonheur et longue vie.

A MA SŒUR NAWEL

A tous les moments d'enfance passés avec toi, engage de ma profonde estime pour l'aide que tu m'as apporté, tu m'as réconforté et encouragé, pour ton amour si profond. Puissent nos liens fraternels se consolider et se pérenniser encore plus.

A MON FRERE YOUNES
Tu m'as toujours offert soutien et réconfort, j'exprime envers toi mon frère une profonde admiration et reconnaissance. Pour tous les moments d'enfance passés avec toi, puissent nos liens fraternels se consolider et se pérenniser encore plus.

A MON TRES CHER, Missoum

Tu m'as toujours offert soutien et réconfort, j'exprime envers toi une profonde admiration, reconnaissance et attachement inconditionnels.

A TOUTE MA FAMILLE

Aucun langage ne saurait exprimer mon respect et ma considération pour votre soutien et encouragements. Je vous dédie ce travail en reconnaissance de l'amour que vous m'offrez quotidiennement et votre bonté exceptionnelle. Que Dieu le Tout Puissant vous garde et vous procure santé et bonheur.

A MA BINOME AMIRA
En témoignage de l'amitié qui nous unit et des souvenirs de tous les moments que nous avons passés ensemble, je tiens à te remercier pour ton aide et ton soutien durant la réalisation de ce travail.

A TOUS MES CAMARADES DE PROMO,

Je vous dédie ce travail et je vous souhaite une vie pleine de santé, bonheur et beaucoup de succès.

Meriem

Dédicace

Je dédie ce modeste travail à ceux qui, quels que soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.

A la femme qui a souffert sans me laisser souffrir, qui n'a jamais dit non à mes exigences, ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles

Mon adorable amie, confidente et complice Mama Fatiha

A l'homme de ma vie, à mon support dans ma vie, mon précieux offre du Dieu, qui doit ma vie, ma réussite et tout mon respect, qui me guide dans la bonne direction, merci d'être à la hauteur de mes exigences incessantes

Mon chéri PaPa Moussa

A mes très chers frères Saïd, Adel et Karim. Merci de répondre à mes demandes continues puisse Dieu vous donner santé, bonheur, courage et surtout réussite.

A mes belles sœurs Asmae et Aridj. Merci pour les petits plaisirs, pour les rires et pour les fois où nous avons partagé nos rêves, je vous souhaite une vie pleine de bonheur et de succès.

A ma grand-mère, mes oncles et mes tantes, que Dieu leurs donne une longue et joyeuse vie.

A mes amies que j'ai connu jusqu'à maintenant. Merci pour vos amours et vos encouragements.

Sans oublier mon binôme Meriem pour son soutien moral, merci d'avoir partagé tous ce parcours avec moi.

Une spéciale dédicace à Fairouz et Ikram merci d'avoir étés à mes côtés tout au long de ce projet.

Je vous aime de tout mon cœur

Nour El Houda ♥♥

Sommaire

Chapitre II : Etude conceptuelle

Chapitre III : Implémentation informatique et validation

Chapitre I

Chapitre II

Chapitre III

Chapitre III

Chapitre II

CHAPITRE I : Etude bibliographique

I.1. Introduction :

Dans le domaine de la fabrication mécanique, les industriels doivent faire face à de nombreux facteurs défavorables au sein du marché actuel. Le prix de revient, les délais de livraison demandés sont de plus en plus courts tandis que le prix des matières premières augmente et la qualité doit être d'un niveau bien élevé. Cependant, l'évolution actuelle des marchés impose le choix pertinent des techniques de fabrication pour réduire les coûts et améliorer la qualité [1].

Un procédé de fabrication est un ensemble de techniques visant l'obtention d'une pièce ou d'un objet par la transformation d'une matière brute. Obtenir la pièce désirée, nécessite parfois l'utilisation successive de différents procédés de fabrication. Ces procédés de fabrication font partie de la construction mécanique [2]. Le processus d'élaboration des pièces de formes complexes doit permettre de garantir la fidélité entre la pièce et les spécifications fonctionnelles exprimant l'idée initiale du designer. Ce processus se découple en une activité de conception et une activité de fabrication. Tout d'abord, dans un module de CAO « Conception Assistée par Ordinateur », un modèle géométrique (modèle de référence de la maquette numérique) est construit, à partir des spécifications fonctionnelles. Ensuite, les trajectoires d'outils permettant l'usinage de la pièce sont calculées dans le module de FAO « Fabrication Assistée par Ordinateur ». Finalement la pièce est usinée selon les trajectoires précédemment calculées [3].

Ce chapitre est organisé en cinq (05) parties : 1) généralités sur le processus de production de pièces de formes complexes, 2) étude du format d'échange de données « STL », 3) généralités sur l'opération d'ébauchage sur des fraiseuses numériques à 03-axes, 4) étude de la stratégie d'ébauchage « Contours Décalés » et 5) identification des problèmes d'interférences et de collisions.

I.2. Processus de production des pièces mécaniques

Chaque produit, suit un cycle de vie, qui commence par sa naissance, traverse sa vie et va jusqu'à sa mort. La naissance est la phase de l'élaboration de l'idée puis sa mise en œuvre à travers sa production pendant laquelle le produit est conçu, développé, puis fabriqué. La vie est la phase qui suit la production, pendant laquelle le produit est mis à la disposition de l'utilisateur. La mort est la phase après l'utilisation, quand le produit ne sert plus à rien ou devient obsolète.

Lors de leurs productions, les produits issus de la fabrication mécanique, passent par deux grandes étapes :

- Conception Assistée par Ordinateur « CAO ».
- Fabrication Assistée par Ordinateur « FAO ».

Chaque activité doit garantir la conformité de la pièce et le respect de ses spécifications ainsi que les ressources et les moyens de production disponibles.

I.3. Définition des pièces complexes

Une pièce dite est complexe, lorsqu'il y a présence de topologies nécessitant plusieurs trajectoires d'outils, avec des risques d'interférences, de collisions et des zones difficilement accessibles [4].

I.3.1. Processus de production des pièces mécaniques

Dans ce processus, plusieurs points clefs permettent la maîtrise de l'état de la surface de la pièce finie, en commençant par la phase de conception qui génère un modèle « CAO » « modèle géométrique », puis la phase de planification « gamme d'usinage » et enfin la phase de fabrication suivie par la phase contrôle de la pièce réalisée (Figure 1).

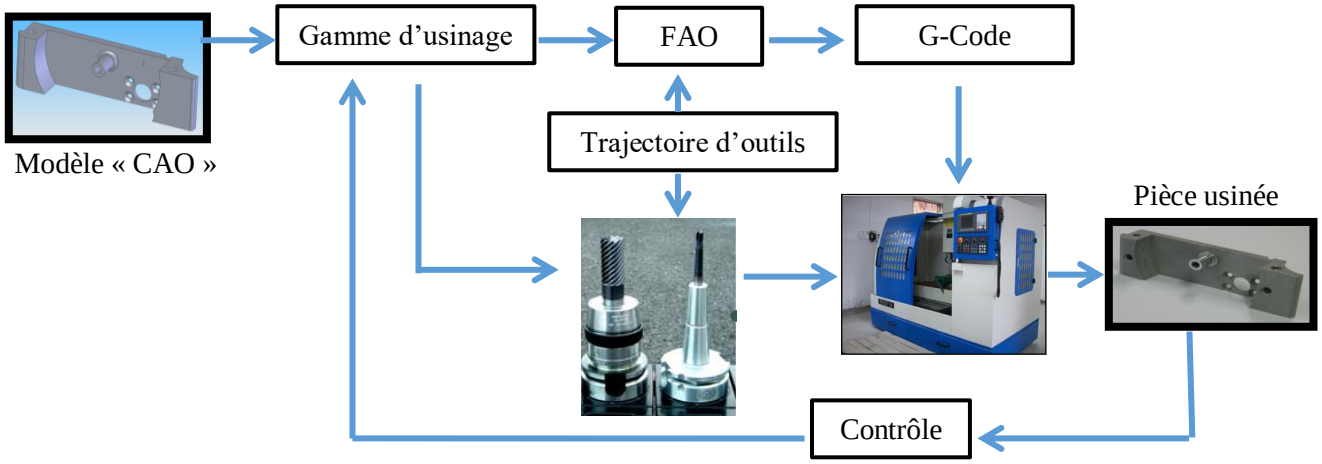


Figure 1 : Processus de production des pièces mécaniques.

I.3.2. Conception et Fabrication Assistées par Ordinateur « CFAO »

La « CFAO » est un processus basé sur la mise en œuvre du traitement continu de l'information (sous forme numérique). Elle permet à partir d'une représentation volumique d'extraire des entités à réaliser [5].

I.3.3. Conception Assistée par Ordinateur « CAO »

L'opération de conception « CAO » d'une pièce en trois dimensions (Figure 2) consiste, à l'aide d'un logiciel de conception (modeleur volumique) à associer ou à soustraire des volumes géométriques simples appelés également solides [5]. Les avantages de la « CAO » sont les suivants [5] :

- Augmenter la productivité du concepteur.
- Améliorer la qualité de la conception.
- Améliorer les communications à travers la documentation.
- Créer une base de données pour la fabrication. [5]

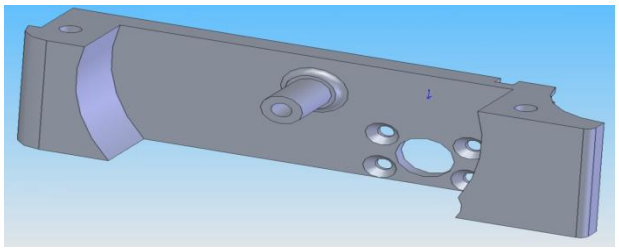


Figure 2 : Modèle « CAO »[1].

I.3.3.1. Gamme d'usinage

L'usinage effectif d'une pièce sur machine-outil nécessite l'établissement d'une gamme d'usinage. C'est une méthode qui permet d'établir l'ordre chronologique des différentes opérations d'usinage, de préciser ce que doit être respecté et les outils à utiliser pour permettre l'obtention d'une pièce finie conforme au dessin de définition du produit fini. Cette méthode est effectuée à partir des données suivantes :

Données relatives à la pièce. Il s'agit généralement d'un dessin représentant la pièce finie d'usinage ou son modèle « CAO » et les informations complémentaires qui sont :

- La qualité des différentes surfaces de la pièce et leur positionnement,
- Le matériau constituant la pièce,
- La présence éventuelle de traitement thermique ou de traitement de surface,
- Le procédé d'obtention de la pièce brute.

Les données économiques. Il s'agit du nombre de pièces à fabriquer, de la taille des lots et des délais de fabrication.

Les données technologiques. Il s'agit des machines-outils, des outils et des outillages disponibles, les possibilités des machines-outils (cinématique, courses maximales et la qualité garantie pour la série de pièces) et des conditions d'usinage (volume de copeau maximal, direction de travail).

I.3.4. Fabrication Assistée par Ordinateur « FAO »

Le processus de fabrication « FAO » consiste à mettre en œuvre une stratégie qui permet le pilotage par ordinateur d'un outil afin de balayer une surface jusqu'à l'obtention de la géométrie prescrite dans le modèle « CAO » (Figure 3).

Figure 3 : Modèle « FAO » (pièce usinée) [2].

I.4. Généralités sur les machines-outils à commande numérique « MOCN »

La « MOCN » est une machine totalement ou partiellement automatique à laquelle les ordres sont communiqués grâce à des codes dans un programme à commande numérique. Lorsque la machine-outil est équipée d'une commande numérique capable de réaliser les calculs des coordonnées des points définissant une trajectoire, on dit qu'elle est à calculateur. Elle est appelée Commande Numérique par Calculateur (CNC) [7].

I.4.1. Architecture d'une MOCN

La « MOCN » est composée de deux principales parties [8] (Figure 4) :

- **Partie commande :** constituée d'une armoire dans laquelle on trouve un clavier pour rentrer les commandes, lecteur de données et un écran de visualisation de toutes les données enregistrées.
- **Partie opérative :** composée d'une base pour qu'elle soit indépendante du sol, d'un bâti, d'un support outil, d'une table support pièce et des moteurs chargés de l'entraînement de la table.

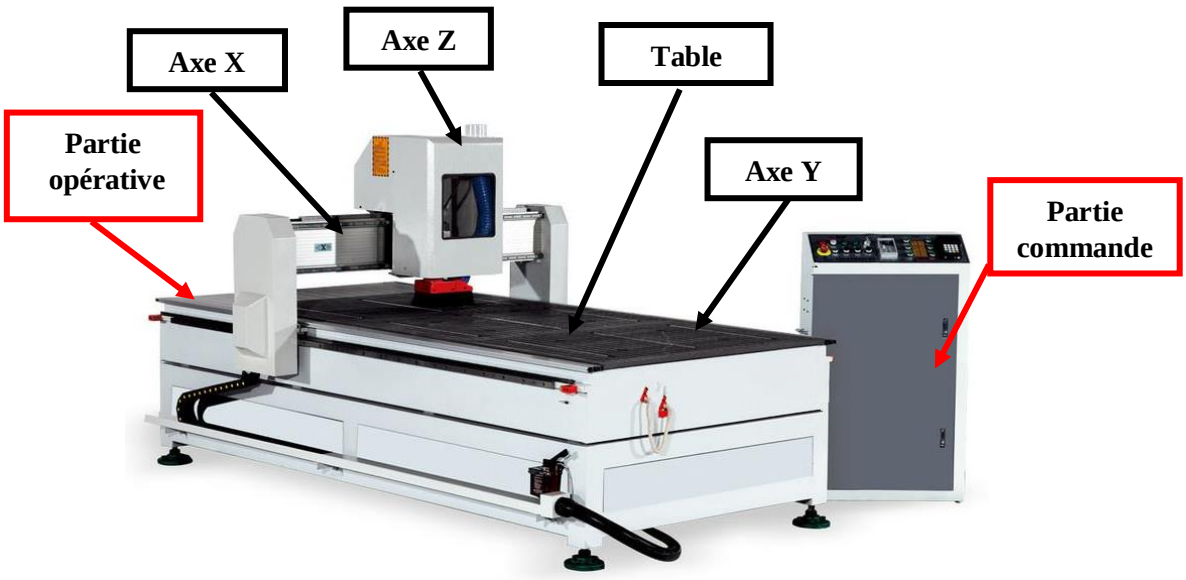


Figure 4 : Machine-outil à commande numérique [3].

I.4.2. Fraiseuse

Une fraiseuse est une machine-outil utilisée pour usiner tous types de pièces mécaniques prismatiques, à l'unité ou en série, par enlèvement de matière à partir de blocs à l'aide d'un outil coupant nommé fraise. La fraise munie de dents est mise en rotation et taille la matière grâce à la combinaison de deux mouvements : sa rotation et le mouvement relatif de la fraise par rapport à la pièce. La forme de la fraise est variable. Elle peut être cylindrique, torique, conique, hémisphérique ou quelquefois de forme encore plus complexe. La fraise et la pièce peuvent se déplacer relativement suivant des coordonnées X, Y ou Z [15].

I.4.2.1. Types de fraiseuses

Selon l'orientation de l'axe de rotation de la broche, on peut classer les fraiseuses en fraiseuses horizontales et fraiseuses verticales (Figure 5) :

- **Fraiseuse horizontale :** la broche se déplace horizontalement par rapport à la table.
- **Fraiseuse verticale :** l'axe de la broche est perpendiculaire à la table et permet le mouvement vertical de l'outil [16].

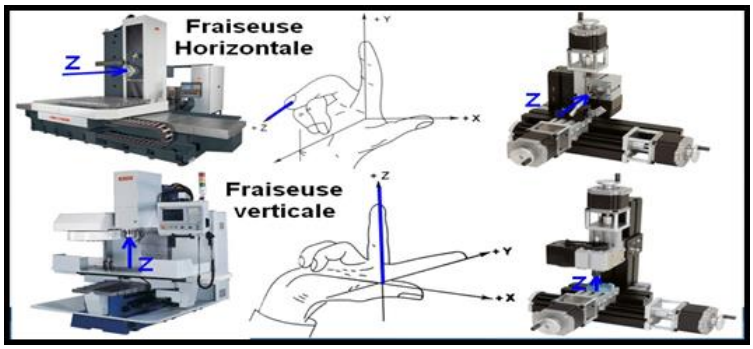


Figure 5 : Types de fraiseuses [4].

Selon le nombre d'axes (mouvements autorisés), les fraiseuses peuvent être classées également en trois types (Figure 6) :

- **Fraiseuse à 03-axes :** le mouvement relatif entre la pièce et l'outil sont trois déplacements linéaires.
- **Fraiseuse à 04-axes :** en plus des trois déplacements linéaires, une rotation est ajoutée.
- **Fraiseuse à 05-axes :** en plus des trois déplacements linéaires, deux rotations sont ajoutées.

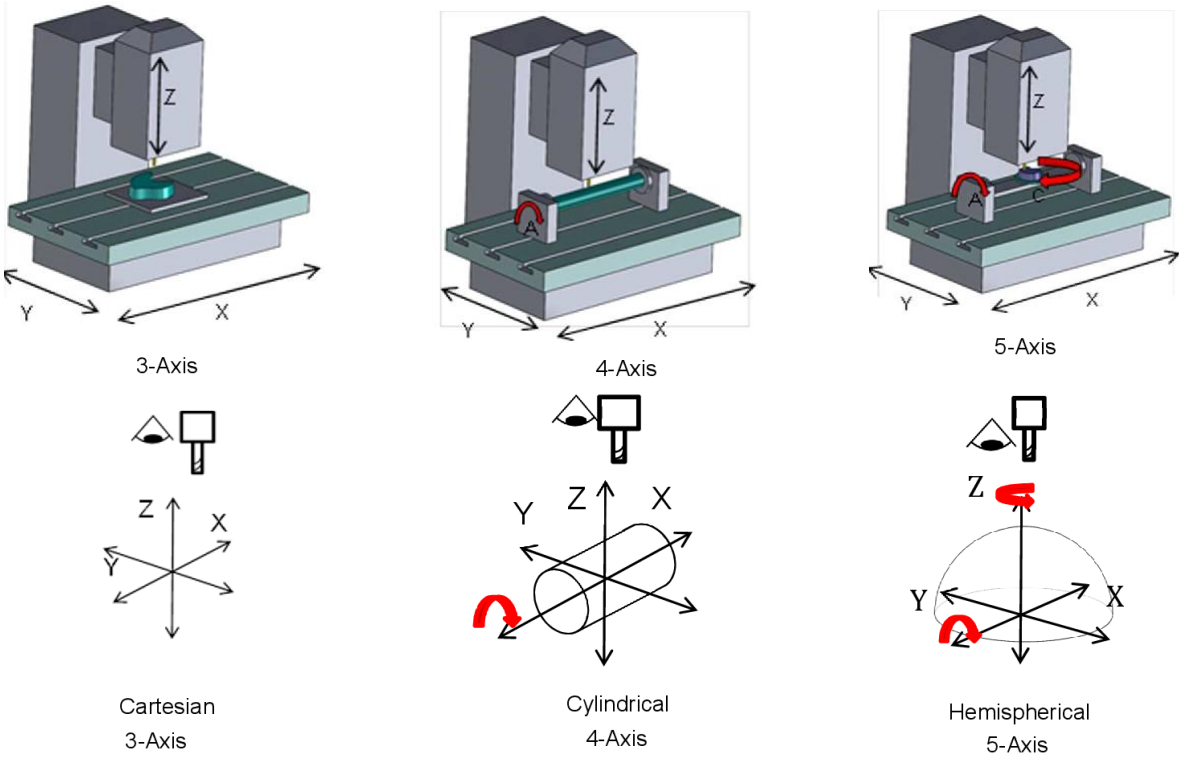


Figure 6 : Types de fraiseuses.

I.4.3. Fraisage

Le fraisage est une technique de fabrication qui combine deux mouvements : la rotation d'un outil de coupe et l'avancée d'une pièce à usiner (Figure 7). Actuellement, on ajoute le déplacement de l'outil (toutes les directions sont possibles) par rapport à la pièce. [13].

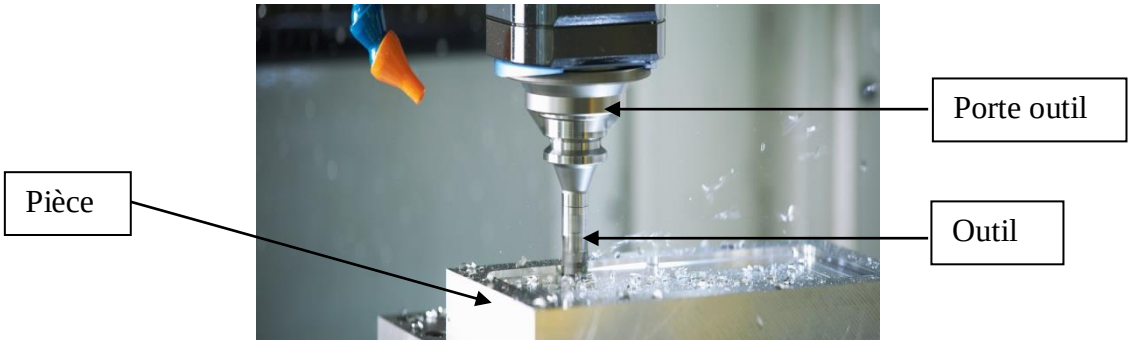
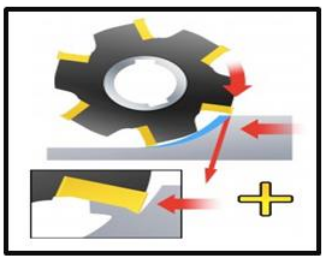


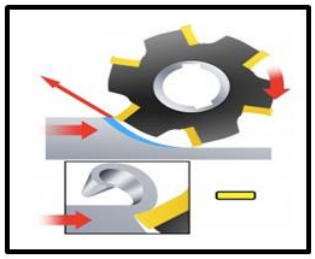
Figure 7 : Fraisage de pièce prismatique [5].

I.4.3.1. Types de fraisage

- **Fraisage en avalant :** l'outil avance dans le sens de sa rotation. Il doit toujours être préféré si la machine-outil, le bridage et la pièce le permettent (Figure 8.a).
- **Fraisage en opposition :** la direction de l'avance est opposée au sens de rotation de l'outil (Figure 8.b). Il est avantageux dans le cas de grandes variations dans la surépaisseur d'usinage [14].



a. Fraisage en avalant.



b. Fraisage en opposition.

Figure 8 : Types de fraisage.[6]

I.5. Etude du format d'échange de données « STL »

Le format du fichier « STL » est un format utilisé dans les logiciels de stéréo-lithographie. Il est largement utilisé dans le domaine de la fabrication assistée par ordinateur. Le format « STL » ne décrit que la géométrie des surfaces d'un objet en 3 dimensions. Ce format comporte des informations contenues dans le modèle « CAO ».

I.5.1. Définition du format « STL »

Le format « STL » est le type de fichier standard de l'industrie pour l'impression 3D. Il utilise une série de triangles pour représenter les surfaces d'un modèle solide (Figure 9) [10].

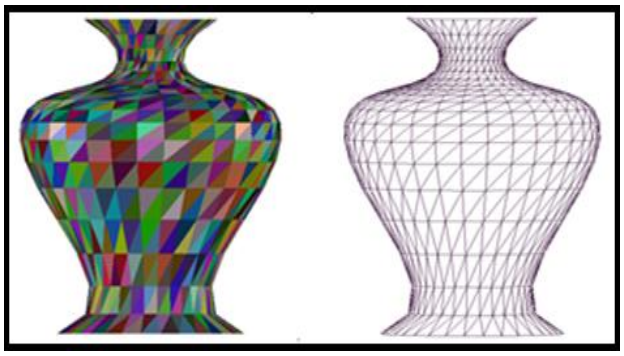


Figure 9 : Exemple du modèle STL.

I.5.2. Structure d'un fichier « STL »

Le fichier « STL » décrit un objet par sa surface externe. Cette surface est nécessairement fermée et définie par une série de triangles. Chaque triangle est défini par les coordonnées cartésiennes (x, y, z) dans un trièdre direct de son vecteur normal unitaire $\vec{n}$ orienté vers l'extérieur de l'objet et de ses trois sommets [11].

I.5.3. Enregistrement d'un fichier « STL »

Le fichier « STL » stocke les informations des triangles par deux manières : un codage binaire et un codage ASCII.

- **Codage binaire** : le codage binaire est plus commun, fichier de faible taille mais non lisible et trop condensé. un fichier « STL » binaire est structuré de la façon suivante (Figure 10) :
  - Les 80 premiers octets sont un commentaire.
  - Les 04 octets suivants forment un entier codé sur 32 bits, qui représente le nombre de triangles présents dans le fichier.

Ensuite, chaque triangle est codé sur 50 octets, selon la décomposition suivante :

- ✓ 3 fois 4 octets, chaque paquet de 4 octets représente un nombre à virgule flottante correspondant respectivement aux coordonnées (x, y, z) de la normale au triangle.
- ✓ 3 paquets de 3 fois 4 octets, chaque groupe de 4 octets représente un nombre à virgule flottante correspondant respectivement aux coordonnées (x, y, z) de chacun des sommets du triangle.
- ✓ 2 octets représentant un mot 16 bits de contrôle.

```
UINT8[80] – en-tête
UINT32 – Nombre de triangles


foreach triangle
REAL32[3] – Vecteur normal
REAL32[3] – Sommet 1
REAL32[3] – Sommet 2
REAL32[3] – Sommet 3
UINT16 – Mot de contrôle
end
```

Figure 10 : Codage binaire d'un fichier STL [7].

- **Codage ASCII :** le codage ASCII est plus descriptif et lisible mais le fichier est gros de taille. un fichier « STL » ASCII commence par la ligne « solid name » où name est une chaîne facultative. Le fichier continue avec n'importe quel nombre de triangles, chacun représenté comme suit : chaque n ou v est un nombre à virgule flottante. Le fichier est clôturé par « endsolid name » [12] (Figure 11).

```
solid name
facet normal ni nj nk
   outer loop
     vertex v1x v1y v1z
     vertex v2x v2y v2z
     vertex v3x v3y v3z
   endloop
endfacet
endsolid name
```

Figure 11 : Codage ASCII d'un fichier [8].

I.5.4. Avantages et inconvenants du modèle « STL »

- **Avantage :**
  - Le format STL est devenu un standard.
  - Très courant et peut-être lu par l'ensemble des logiciels 3D.
  - L'existence de nombreuses bibliothèques de fichiers 3D « STL » [12].
- **Inconvénient :**
  - Le format STL décrit seulement la géométrie, d'autres informations sur l'objet telles que la couleur, matière … ne sont pas décrites.
  - Le fichier « STL » est difficilement modifiable [12].

I.6. Usinage

L'usinage consiste à réaliser des pièces par enlèvement de matière en respectant, l'état de surface, les tolérances de formes et la géométrie spécifiée. À chaque phase de la gamme de fabrication, le bureau des méthodes associe la machine, le type d'usinage à réaliser, l'outil ainsi que le support de pièce permettant l'obtention de tous les éléments de cotation de la surface considérée. L'usinage des pièces complexes est réalisé dans sa grande majorité par fraisage en 03,04 et 05-axes [19] (Figure 12).

Figure 12 : Pièces mécaniques de formes complexes.

I.6.1. Usinage 03-axes

En usinage à 03-axes, l'orientation de l'axe de l'outil est fixe et se fait par trois déplacements linéaires. Par conséquent pour chaque point de contact il existe une et une seule position de l'outil qui soit tangente à la surface à usiner au point considéré (Figure 13).

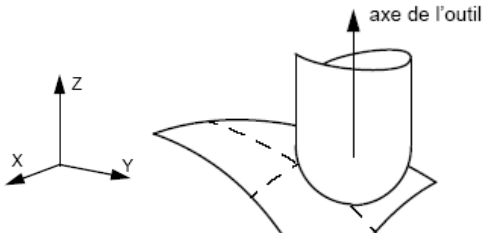


Figure 13 : Usinage 03-axes.

I.6.2. Phases d'usinage

Le processus d'usinage est l'enlèvement de matière d'une pièce brute dans le but de réaliser une pièce finie conforme aux spécifications dimensionnelles et technologiques. Afin de transformer un métal en une pièce finie, il faut passer par trois phases, ébauchage, demi-finition et finition. Chacune de ces phases transforme la forme du métal de manière à se rapprocher du modèle de la pièce désirée (Figure 14).

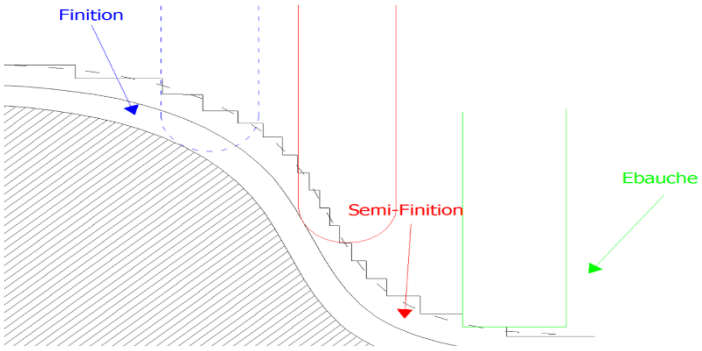


Figure 14 : Phases d'usinage [9].

I.6.3. Caractéristiques des opérations d'usinage

- **Ebauchage :** éliminer la croûte superficielle liée au procédé d'obtention du brut (écroûtage) ainsi que les fortes surépaisseurs d'usinage [20].

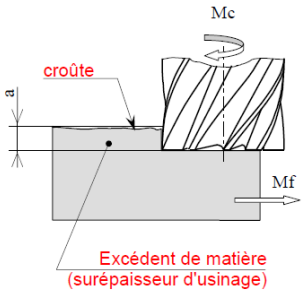


Figure 15 : Opération d'ébauchage [10].

- **Demi-finition :** elle permet d'assurer une surépaisseur constante et faible pour l'opération de finition [20, 22].
- **Finition :** respecter toutes les spécifications imposées par le dessin de définition. En plus de respecter l'état de surface désiré et finir la cote demandée. [20]

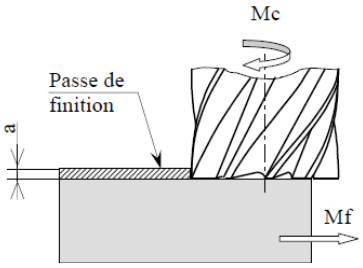


Figure 16 : Opération de finition [10].

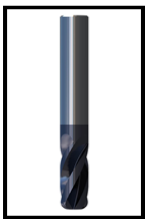
I.6.4. Outils d'usinage

I.6.4.1. Définition de l'outil de coupe

La fraise est un outil qui se place sur une « fraiseuse » pour fabriquer une pièce par enlèvement de matière. Cet enlèvement de matière se réalise grâce à la combinaison de deux mouvements : la rotation de l'outil de coupe et l'avancée de la pièce. Cette opération est désignée sous le nom de « fraisage ». Ils existent plusieurs types de fraises (Figure 17) (cylindrique, hémisphérique et torique). [21]

a. Cylindrique.

b. Hémisphérique.



c. Torique.

Figure 17 : Types de fraises (outils).

I.6.4.2. Paramètres d'outil de coupe

L'objectif final de l'usinage est d'obtenir une pièce usinée avec un bon état de surface. Ceci nous impose de déterminer certains paramètres spécifiques :

- ✓ **Vitesse de coupe :** la relation entre la vitesse de coupe Vc et la vitesse de rotation w de la fraise est donnée par la relation suivante :

  $Vc = R \times w$, où R est le rayon de la fraise.

- ✓ **Vitesse d'avance :** la relation entre la vitesse d'avance de la table et la vitesse de rotation de la fraise est donnée par : $V_a = N_f \times S_t \times N$

Avec :

N_f: Nombre de dents de la fraise,

S_t: Avance par tour ou par dent,

N : vitesse de rotation de la broche en tr/min.

I.6.5. Stratégies d'usinage

I.6.5.1. Stratégie des « Plans Parallèles »

Les passes d'usinage de type parallèle constituent l'une des stratégies les plus courantes. Elles sont perpendiculaires au plan XY et suivent la surface dans la direction Z [23]. Elle s'appuie sur des trajectoires d'outil résultantes des intersections entre la surface à usiner et un ensemble de plans parallèles.

➢ **Avantage :**

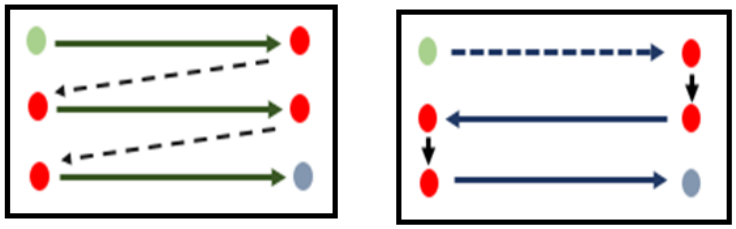
- Ne génère pas de trajectoires d'outils redondantes et permet un gain de temps notable.
- Evite l'apparition de zones non usinées lors de la planification de trajectoires.
- Planifie les trajectoires d'outils en 3D [24].

➢ **Inconvénient :**

- Elle n'est pas forcément optimale [24].

- **Modes de la stratégie « Plans Parallèles »**

Les trajectoires définies par les plans parallèles peuvent être parcourues dans un seul sens « One-Way » (Figure 18.a) ou en « Zig-Zag » [24] (Figure 18.b).



b. One-Way. a. Zig-Zag.

Figure 18 : Modes de la stratégie « Plans Parallèles ».

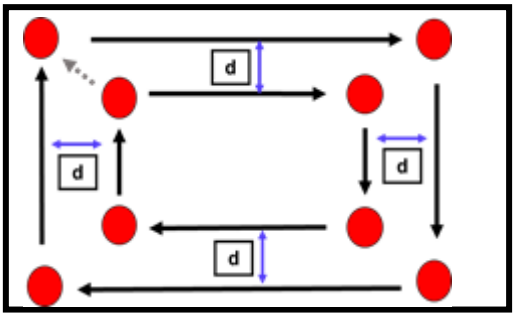
I.6.5.2. Stratégie des « Contours Décalés »

Un contour est une suite de segments consécutifs où chaque segment est défini par deux sommets et une normale (Figure 19). Le sommet est défini par ces trois (03) coordonnées X, Y et Z. La normale est perpendiculaire au segment et est définie par trois (03) composantes Nx, Ny et Nz.

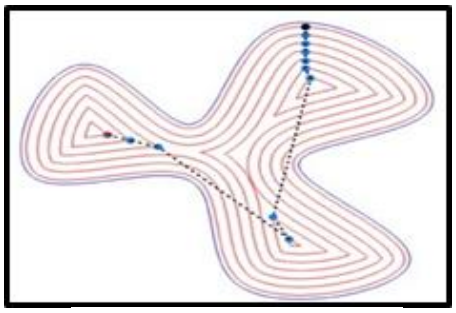
Cette stratégie est composée de deux étapes :

➢ La première étape consiste à calculer les contours d'intersection entre des plans horizontaux et la surface à usiner.

➢ La deuxième étape quant à elle consiste à décaler les contours calculés d'une certaine distance « ω » fixée par l'utilisateur et qui doit être inférieure au rayon de l'outil. Les différents contours décalés constituent le trajet d'usinage final.



a. Contour décalé.



b. Trajectoire d'outil.

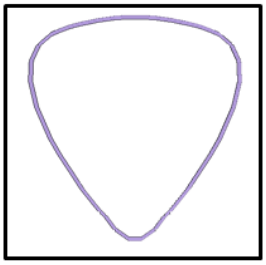
Figure 19 : Stratégie du « Contour Décalé ».

Le décalage d'un contour dépend du sens des normales de ses segments. Il est exprimé en pourcentage du diamètre de l'outil [25].

❖ Types de contours

Le type de contour est soit fermé ou ouvert :

- Un contour est dit « contour fermé » si le point de départ du premier segment est confondu avec le point final du dernier segment (Figure 20.a).
- Dans le cas contraire, il est dit « contour ouvert » (Figure 20.b).



a. Contour fermé.

b. Contour ouvert.

Figure 20 : Types de contours.

❖ Sens d'un contour

Le sens d'un contour est soit horaire ou antihoraire.

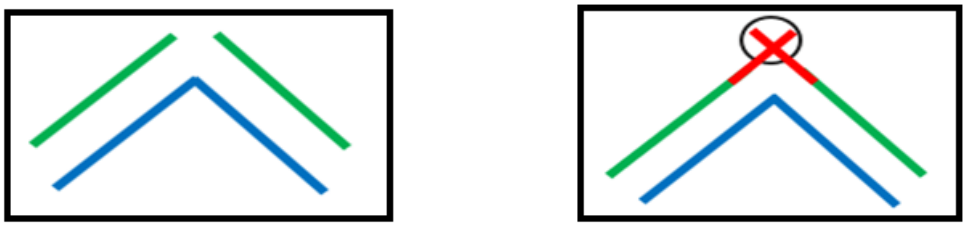
- Un contour orienté dans le sens antihoraire est un contour orienté positivement.
- Un contour orienté dans le sens des aiguilles d'une montre est un contour orienté négativement.

I.6.5.2.1. Décalage des contours

Le décalage d'un contour est une translation ou déplacement de tous ses segments et ses sommets de la même distance, selon la même direction et dans le même sens. C'est-à-dire suivant leurs vecteurs normaux. Les calculs nécessaires pour décaler la géométrie s'effectuent

en fonction du diamètre de l'outil et/ou de la surépaisseur à enlever. Les segments du nouveau contour décalé dans certains cas ne sont pas liés, pour cela apparition de zones indéfinies [26].

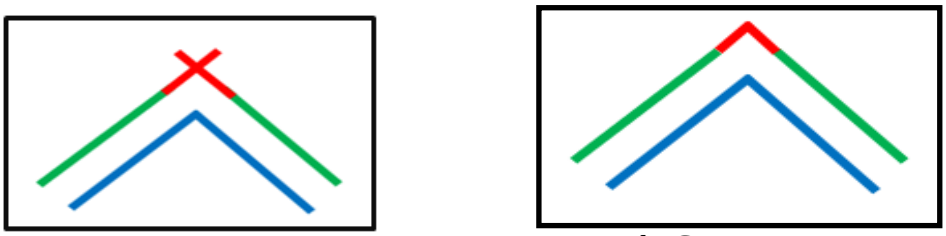
- **Zone indéfinie :** la zone indéfinie apparait lorsque deux segments successifs décalés ne se coupent pas. Le traitement de cette singularité consiste en l'insertion d'un segment / d'un arc de cercle [26] (Figure 21).



a. Zone indéfinie. b. Correction de la zone.

Figure 21 : Correction d'une zone indéfinie.

- **Segments colinéaires :** la correction des zones indéfinies provoque un nouveau problème. C'est l'apparition de segments colinéaires (Figure 22).



a. Segments colinéaires b. Correction.

Figure 22 : Elimination des segments colinéaires

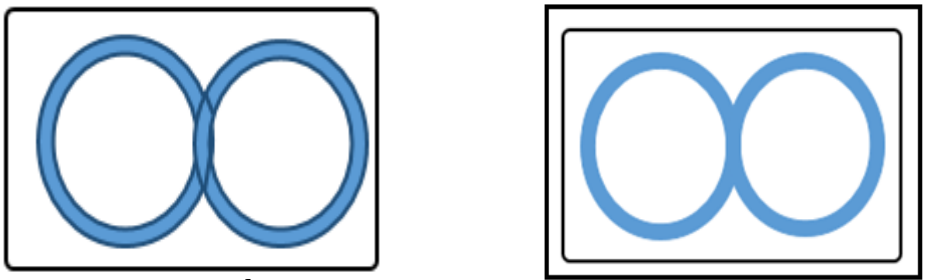
Après le décalage des contours, d'autres intersections peuvent apparaitre ; soit des auto-intersections soit des intersections entre les contours.

- **Zone d'auto-intersection :** la zone de collision apparait lorsque deux segments successifs se coupent à l'intérieur de leurs limites. Dans ce cas il est nécessaire d'éliminer cette collision [26].

a. Auto-intersection. b. Correction.

Figure 23 : Elimination d'auto-intersection.

- **Zone d'intersection :** lorsque le contour décalé recouvre partiellement le contour extérieur décalé ou celui d'un autre, apparaît une boucle inter-contour ou une intersection. La solution permettant l'élimination de ces boucles consiste en la fusion de ces deux contours en un seul contour [26].



a. Intersection entre deux contours. b. Correction de l'intersection.

Figure 24 : Elimination d'intersection entre deux contours

I.6.6. Problèmes d'usinage

Les problèmes de collisions et d'interférences peuvent être générés à cause de la forme complexe des pièces. Ces problèmes sont classés en trois types :

- **Interférence locale :** la principale cause est la différence de courbures entre la partie active de l'outil et la surface à usiner.
- **Interférence globale « collision » :** elle est provoquée entre l'ensemble « corps d'outil, porte outil et la broche » et l'ensemble « pièce, porte pièce et éléments de structures ».
- **Interférence vers l'arrière :** elles sont des pénétrations indésirables de l'arrière de l'outil dans la surface à usiner [27].

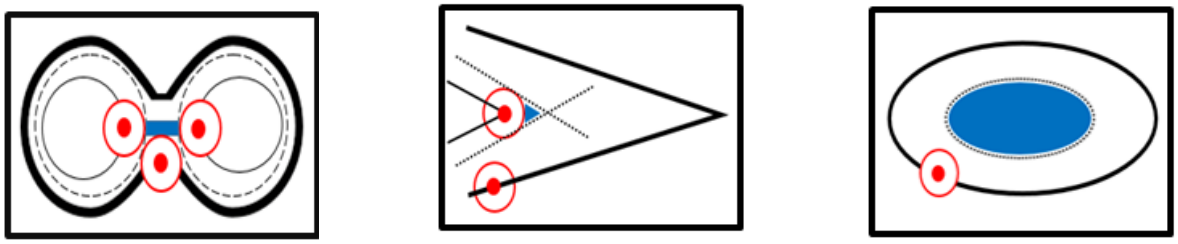


Figure 25 : Zones non usinées avec les contours décalés**.**

I.7. Conclusion

Dans ce chapitre nous avons donné une vue générale sur le processus de conception et de fabrication assistées par ordinateur des pièces. Après la présentation des deux processus « CAO » et « FAO », nous avons décrit la passerelle entre ces deux processus qui manipulent des informations hétérogènes et nous avons détaillé le format « STL » choisi comme format d'échange dans notre travail. Ensuite, nous avons présenté brièvement l'opération d'ébauchage et ses différentes stratégies. Enfin, une étude détaillée sur la stratégie « Contours Décalés » a été explicité.

Dans le prochain chapitre, nous allons modéliser notre étude et présenter la solution proposée avec une conception de l'application.

CHAPITRE II : Etude Conceptuelle

II.1. Introduction

En informatique, la résolution d'un problème donné exige sa décomposition en une suite d'étapes successives pour arriver à une solution appropriée. Dans le chapitre précédent, nous avons mené une étude bibliographique sur l'opération d'ébauchage par la stratégie « Contours Décalés » sur des fraiseuses numériques à 03-axes. Ce chapitre sera consacré à la présentation de la problématique, aux solutions envisagées ainsi qu'à la conception informatique.

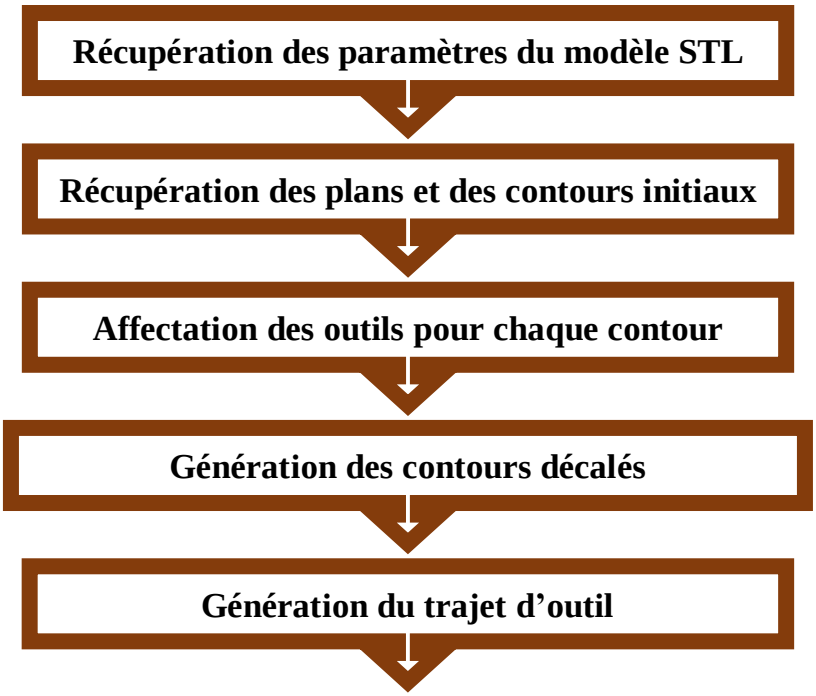
II.2. Problématique

La stratégie « Contours Décalés » permet de réaliser des usinages continus dans le même plan de coupe. Le trajet d'outils résulte des décalages successifs des contours issus de l'intersection de la pièce avec des plans horizontaux où la distance de décalage dépend des outils utilisés. Elle permet d'ébaucher des pièces définies par des modèles continus ou des modèles discrets (STL). Pour atteindre cet objectif, plusieurs problèmes sont rencontrés :

- La complexité de la forme géométrique de la pièce rend la tâche couteuse et difficile à cause de la taille volumineuse du modèle « STL ».
- Délimitation des parties à usiner.
- Identification des outils optimums permettant d'éviter les problèmes d'interférences et de collisions.
- Combinaison d'une façon optimale des outils optimums.
- Détermination des contours d'intersection et des contours décalés.
- Positionnent des outils le long des passes d'usinage.
- Détermination des positions d'outils valides et invalides.
- Détermination du trajet d'outils.

II.3. Architecture générale du système

La résolution du problème est la décomposition de celui-ci en parties cohérentes et simples à résoudre jusqu'à la génération du trajet d'outil. Dans notre cas, l'architecture générale de la solution proposée et ses étapes sont présentées dans **l'Organigramme 1**.



Organigramme 1 : Organigramme de la démarche proposée.

II.3.1. Récupération des paramètres du modèle « STL »

Le fichier « STL » décrivant la pièce à usiner doit être analysé. Cette analyse consiste à vérifier son extension et sa syntaxe. L'étape d'analyse a été déjà faite auparavant. Nous avons seulement récupéré les informations suivantes :

- Paramètres du brut : ses coordonnées limites (Xmin, Xmax, Ymin, Ymax, Zmin, Zmax) et ses dimensions (largeur, longueur et hauteur) (Figure 1). Les dimensions du brut sont données par :

$$\begin{cases} \mathbf{Longueur} = \mathbf{Xmax} - \mathbf{Xmin} \\ \mathbf{Largeur} = \mathbf{Ymax} - \mathbf{Ymin} \\ \mathbf{Hauteur} = \mathbf{Zmax} - \mathbf{Zmin} \end{cases} \quad \textbf{(1)}$$

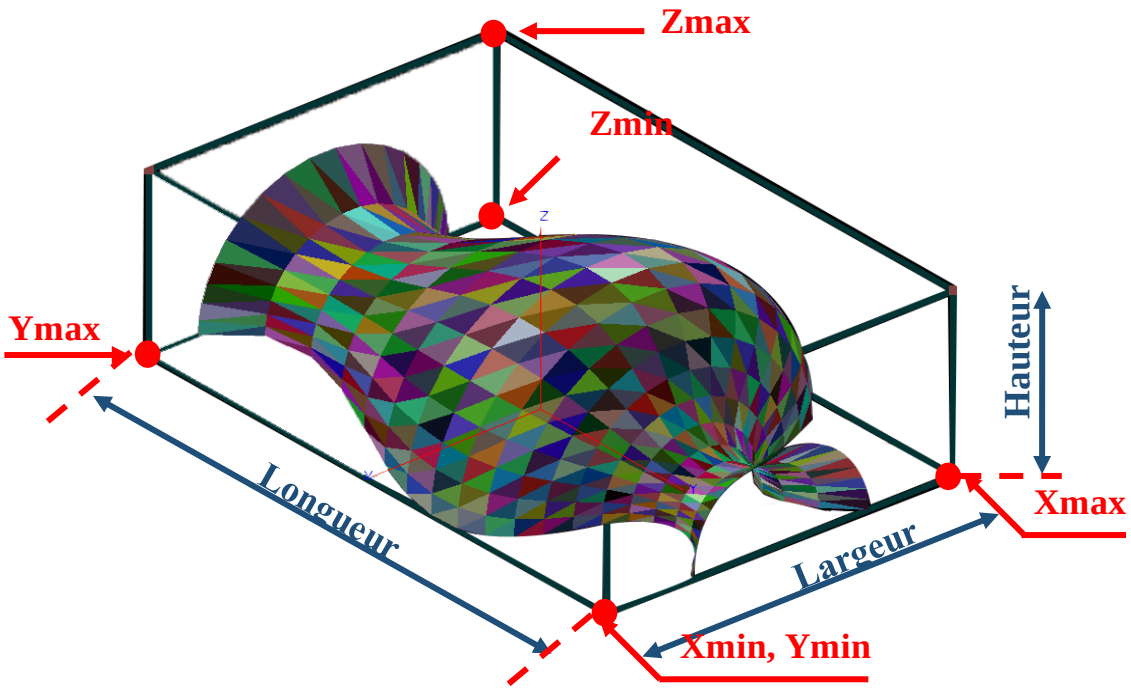


Figure 1: Limites et paramètres du brut.

- Paramètres des triangles : chaque triangle est caractérisé par une normale N définie par ses composantes Nx, Ny et Nz, trois segments reliés entre eux par des sommets (S1, S2, S3) où chaque sommet est caractérisé par ses coordonnées X, Y et Z (Figure 2).

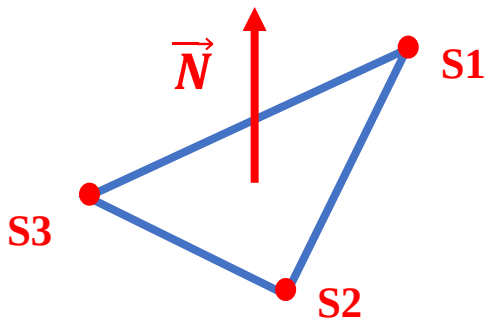


Figure 2: Paramètres d'un triangle.

II.3.2. Récupération des plans et des contours initiaux

L'opération d'ébauchage par la stratégie des « Contours Décalés » consiste à subdiviser le modèle de la pièce à usiner par un ensemble de plans horizontaux « plans de coupe ». La génération des trajets d'outils dans les plans de coupe nécessite la spécification de la profondeur de passe « d ». Le calcul des intersections entre les triangles du modèle « STL » et les plans donne naissance à la génération des contours ce qui rend la délimitation des parties concernées par l'usinage possible. Ce travail a été déjà réalisé précédemment, nous avons seulement récupéré ces informations pour commencer notre travail.

II.3.2.1. Plans de coupe

L'ébauchage d'une pièce mécanique complexe requiert plusieurs plans de coupe où la distance entre deux plans consécutifs est appelé « profondeur de passe ». Chaque plan est caractérisé par ses coordonnées limites (Xmin, Xmax, Ymin, Ymax et Zplan) et un ou plusieurs contours (Figure 3).

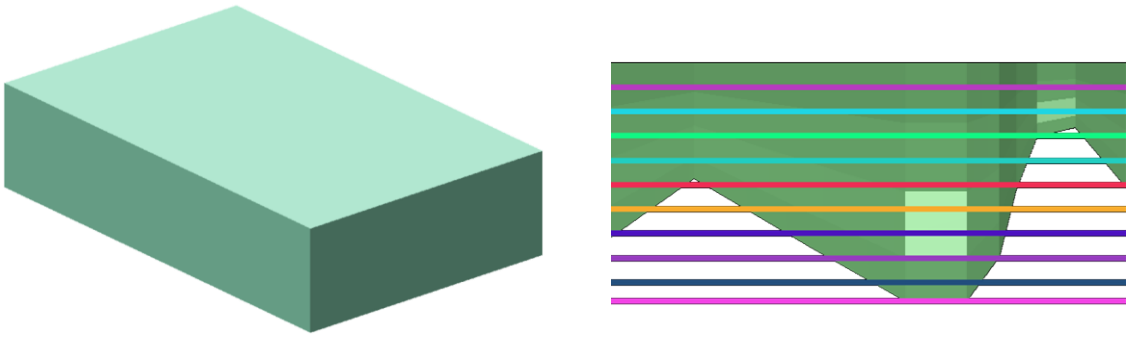


Figure 3 : Plans de coupe.

II.3.2.2. Contours initiaux

Les résultats des intersections entre les plans de coupe et les triangles donnent naissance à une liste de segments. Les contours de chaque plan sont donc construits à l'aide de ces segments. Ces segments sont reliés entre eux pour générer une chaine fermée ou ouverte appelée « contour ». Un plan peut contenir un ou plusieurs contours (Figure 4).

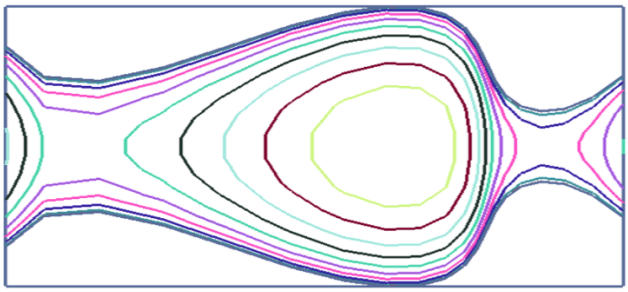
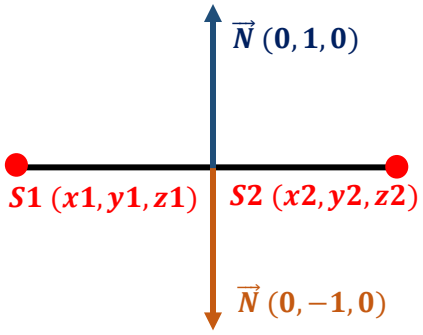


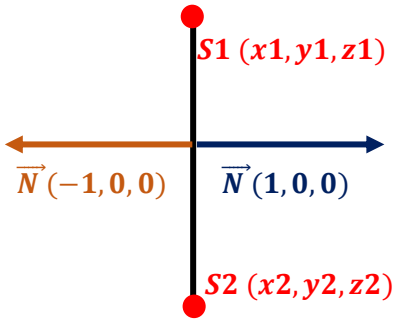
Figure 4: Contours initiaux.

Une fois les contours récupérés, certains paramètres indispensables dans la stratégie « Contours Décalés » doivent être calculés. Ces paramètres sont :

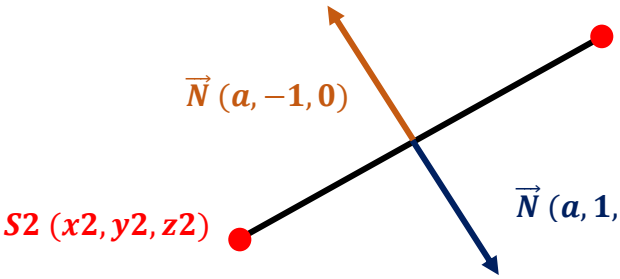
1. **Normale d'un segment :** la stratégie d'ébauchage « Contours Décalé » est basée sur le décalage successif des segments des contours initiaux suivants leurs normales. Donc, il est indispensable de calculer ces normales en fonction de l'orientation des segments. Les composantes du vecteur normal d'un segment en fonction de son type sont données par la Figure 5.



a. Segment horizontal.



b. Segment vertical.



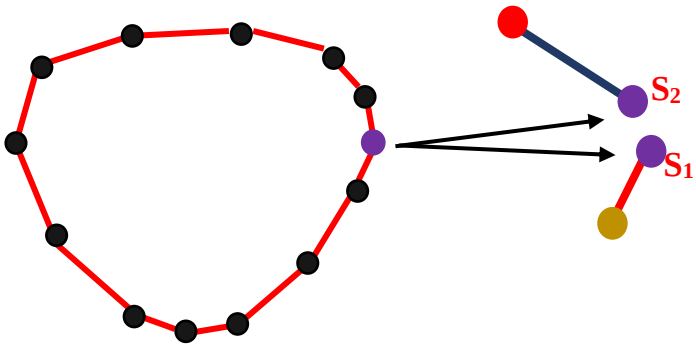
$$a = \frac{y_2 - y_1}{x_2 - x_1}$$

c. Segment incliné.

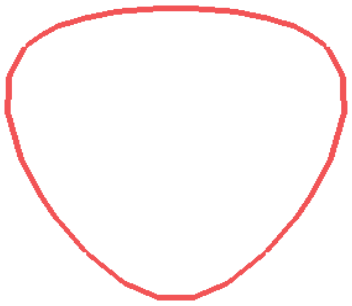
Figure 5 : Normales des différents types de segments.

2. **Type de contour :** la génération des contours est une suite de segments reliés entre eux. Si pour le dernier segment du contour son deuxième sommet est confondu avec le premier sommet du premier segment, alors le contour est dit « fermé » (Figure 6.a-b). Dans le cas contraire, il est dit « ouvert » (Figure 6.c). L'algorithme suivant donne les étapes de calcul.

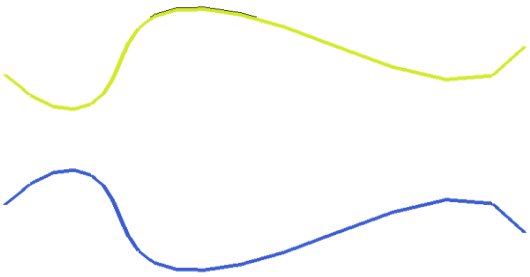
```
Début ;
    Tableau_De_Segments ;
    Dernier_Segment : Type segment ;
    Premier_Segment : Type segment ;
    Si Deuxieme_Sommet_Dernier_Segment = Premier_Sommet_Premier_Segment
        Alors Type_Contour_Fermé == True ;
    Sinon Type_Contour_Fermé == False ;
Fin ;
```



a. Point de fermeture d'un contour fermé.



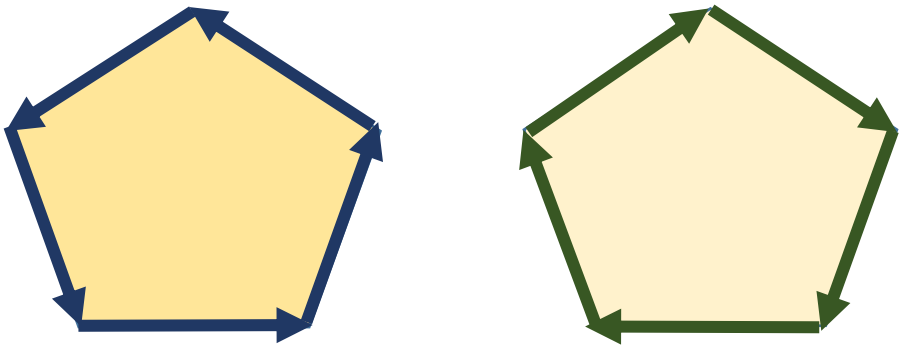
b. Contour fermé.



c. Contour ouvert.

Figure 6 : Types de contour.

3. **Sens de contour :** le critère d'arrêt des décalages successifs est basé sur le sens des contours initiaux. Un contour peut avoir le sens des aiguilles d'une montre « sens horaire » « ClockWise CW » ou le sens inverse des aiguilles d'une montre « sens antihoraire » « Counter-ClockWise CCW » (Figure 7).



a. Sens antihoraire.

b. Sens horaire.

Figure 7 : Sens des contours.

Pour déterminer le sens d'un contour, il faut identifier un sommet convexe. Un polygone est dit convexe si tout segment liant deux points appartenant aux segments du contour est entièrement inclus dans le polygone, sinon, il est dit concave (Figure 8).

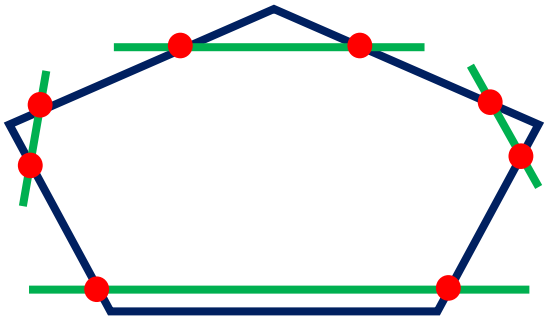


Figure 8 : Contour convexe.

La détermination du sens d'un contour (ouvert ou fermé) commence par la détermination des équations cartésiennes des segments (Figure 9). Le principe de l'identification d'un sommet convexe est basé sur la détermination d'un segment où les deux points, avant le sommet du début du segment et après la fin du segment, soient du même côté par rapport au segment considéré.

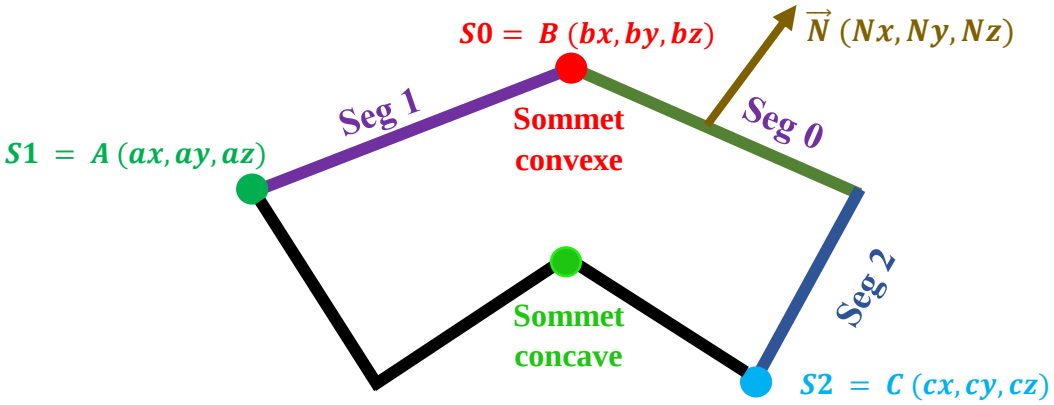


Figure 9 : Types de sommets.

L'équation du segment 0 « Seg 0 » est donnée par :

$$\boldsymbol{Nx} * x + \boldsymbol{Ny} * \mathrm{y} + \boldsymbol{C} = \mathbf{0} \qquad \textbf{(2)}$$

La première étape consiste à remplacer les coordonnées des points A et C dans l'équation du segment 0 pour déterminer leurs signes :

$$\textbf{Signe A} = \boldsymbol{ax\ Nx} + \boldsymbol{ay\ Ny} + \boldsymbol{C} \qquad \textbf{(3)}$$

$$\textbf{Signe C} = \boldsymbol{cx\ Nx} + \boldsymbol{cy\ Ny} + \boldsymbol{C} \qquad \textbf{(4)}$$

Par la suite, calcul du produit des deux signes :

$$\textbf{Signe} = \textbf{Signe A} * \textbf{Signe C} \qquad \textbf{(5)}$$

Si le résultat est supérieur à zéro, alors le sommet B est un sommet « convexe ». Donc, les deux points A et C sont situés du même côté du segment 0. Par conséquent, nous pouvons calculer le sens du contour. Dans le cas contraire, le sommet B est un sommet « concave » et il faut passer au segment suivant.

Si le sommet est convexe, nous devons calculer le produit vectoriel des deux vecteurs des segments ayant ce sommet en commun.

Soient $\vec{U}$ et $\vec{V}$ deux vecteurs de l'espace, le produit vectoriel de ces vecteurs est le vecteur noté $\vec{U} \wedge \vec{V}$:

$$\vec{U}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (6)$$

$$\vec{V}\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} \quad (7)$$

Le produit vectoriel des deux vecteurs est donné par :

$$\vec{U} \wedge \vec{V}\begin{pmatrix} YZ' - ZY' \\ ZX' - XZ' \\ XY' - YX' \end{pmatrix} \quad (8)$$

- Si $\vec{U} \wedge \vec{V} > 0$, alors le sens du contour est « Antihoraire ».
- Sinon, le sens du contour est « Horaire ».

4. **Limite du contour :** la limite du contour est définie par les valeurs minimale et maximale des coordonnées X et Y (Figure 10). La procédure de calcul est donnée par l'algorithme suivant :

```
Début
  Déterminer_limite_Contour ()
    Xmin = 0 ; Ymin = 0 ; Xmax = 0 ; Ymax = 0 ;
  //Pour chaque segment du contour
     for i=0, i← Nbr_Segments, i++
       Xmin = min (Xmin, min (S1_ Segment, S2_ Segment) ;
       Xmax = max (Xmax, max (S1_ Segment, S2_ Segment) ;
       Ymin = min (Ymin, min (S1_ Segment, S2_ Segment) ;
       Ymax = max (Ymax, max (S1_ Segment, S2_ Segment) ;
Fin
```

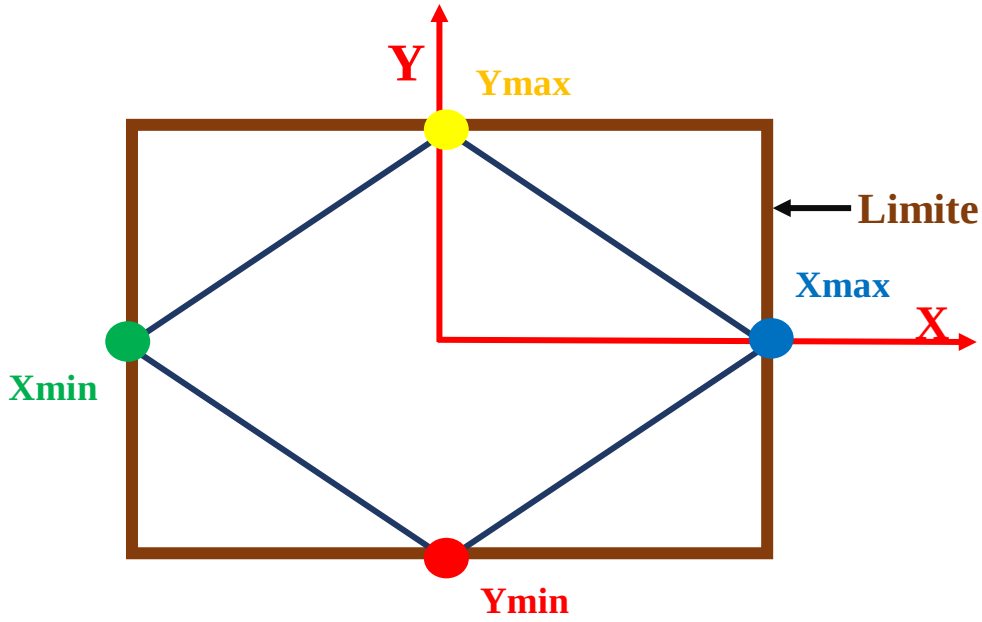


Figure 10 : Limites du contour.

5. Contours fils (contours avec îlots)

En usinage, une part importante des pièces est constituée de cavités ou poches qu'il faut évider. L'usinage de ces poches nécessite en général l'enlèvement d'un important volume de matière en ébauche ou en ébauche et finition combinées. Dans notre cas, un contour peut contenir un ou plusieurs contours fils (Figure 11).

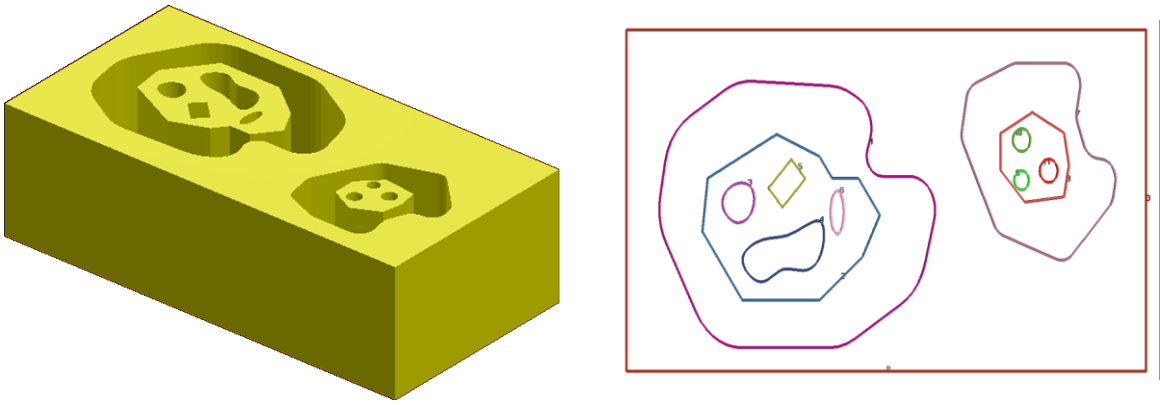


Figure 11 : Contours fils.

Dans ce qui précède, nous avons calculé les limites d'un contour donné (Xmax, Xmin, Ymax, Ymin).

En premier lieu, il faut vérifier pour tous les contours si deux contours ou plus intersectent dans un ou plusieurs points en utilisant les limites des contours. Si ce n'est pas le cas, on dit que le contour ne contient pas des contours fils (Figure 12).

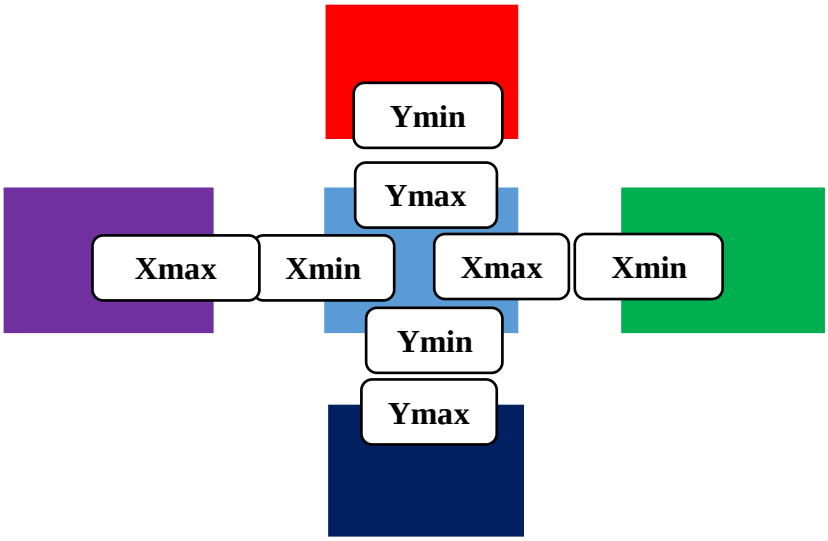


Figure 12 : Comparaison des limites de plusieurs contours.

Après avoir vérifié les limites des contours, il est important de pouvoir déterminer si un certain point est situé à l'intérieur d'un contour donné. Un algorithme bien connu pour répondre à cette question est l'algorithme des demi-lignes (Figure 13). Les étapes de cet algorithme sont :

- Considérons un contour composé de N sommets (Xi, Yi) où i varie de 0 à N-1.
- Le dernier sommet (XN, YN) est supposé être le même que le premier sommet (X0, Y0), c'est-à-dire que le contour est fermé.
- Pour déterminer l'état d'un point (Px, Py) considérons une droite horizontale issue de (Px, Py) vers la droite.
- Si le nombre de fois que cette droite coupe les segments constituant le contour est pair, alors le point P est à l'extérieur du contour.
- Si le nombre d'intersections est impair, le point P est à l'intérieur du contour.

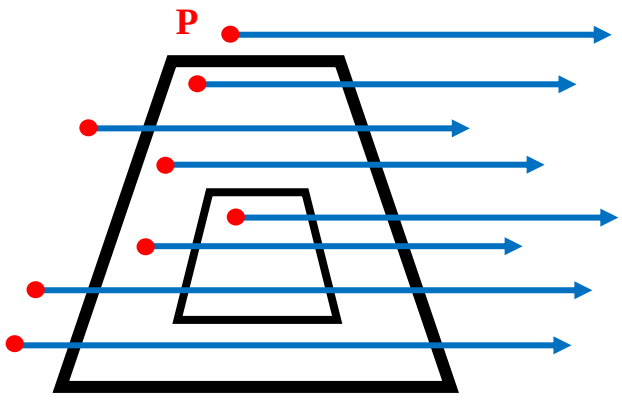


Figure 13 : Position du point « P » par rapport au contour.

Cas particuliers :

Les cas particuliers sont lorsqu'un segment ou un sommet du contour appartient à la droite passant par le point P (Figure 14). Les situations possibles sont illustrées dans la Figure 14 :

- Si le X-maximum des sommets d'un segment est plus grand que Px, alors il n'est pas considéré comme intersection valide.
- Si le Y-maximum des sommets d'un segment est plus petit que Py ou le Y-minimum est plus grand que Py, alors il n'est pas considéré comme intersection valide.
- Si un des sommets du segment a la même valeur de la coordonnée Py, alors que l'autre coordonnée est plus grande que Py, alors il n'est pas considéré une intersection valide.

Les lignes épaisses ne sont pas considérées comme des intersections valides alors que les lignes fines comptent comme des intersections.

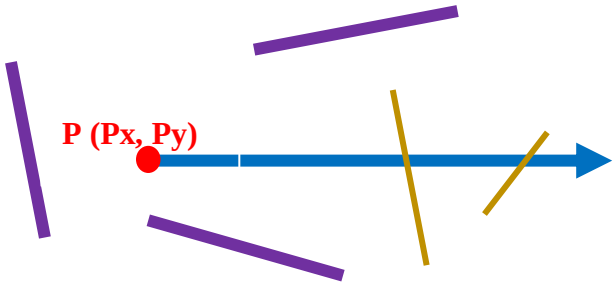
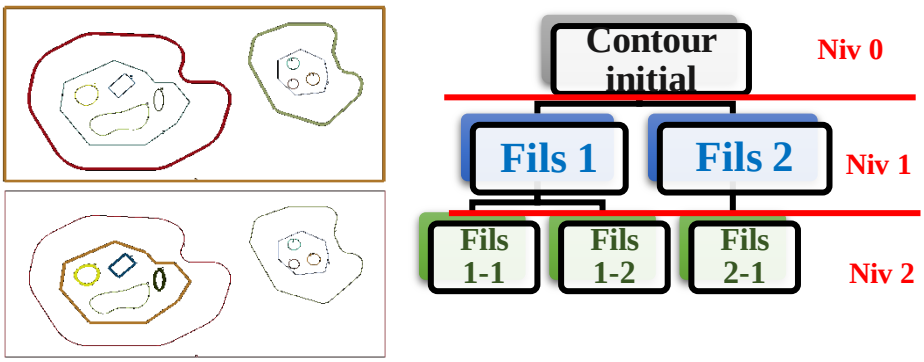


Figure 14 : Cas particuliers des intersections valides et non valides.

Après la détermination des contours fils d'un contour, il faut déterminer les contours fils du premier niveau seulement. Cet algorithme montre la démarche de détermination :

```
Début
    Déterminer_Contours_Fils_1er_Niv
    Contours_Fils_1er_Niv [] ;
    For i=0, i← Nbr_Contour, i++
        Récupérer les contours fils de chaque contour
            For j=0, j← Nbr_Contour_Fils, j++
                Récupérer les contours fils
                For k=0, k← Nbr_Contour_Fils, k++
                    Récupérer les contours fils de chaque contour fils
                    Si Contour_Fils_Contour == Contour_Fils_Du_Contour_Fils
                    Alors Supprimer_Contour_Fils_Contour_Fils
Fin
```

Le principe des contours fils est le même que celui d'un arbre binaire comme l'illustre la hiérarchie suivante (Figure 15).



a. Hiérarchie des contours. b. Contours fils du premier niveau.

Figure 15 : Niveaux des contours fils.

II.3.3. Affectation des outils aux contours

La production des pièces complexes par l'opération d'ébauchage nécessite plusieurs formes d'outils (cylindrique, torique, hémisphérique etc...) car la poche à usiner change de forme d'un plan à un autre. C'est pour cela que nous avons récupéré une base de données d'outils optimums et affecté pour chaque contour son outil optimum. Sachant qu'un outil est caractérisé par sa longueur, son rayon et sa vitesse.

Nous avons créé un tableau d'outils avec différentes formes ou différents rayons que nous avons trié en fonction du nombre d'utilisations, du moins utilisé au plus utilisé. L'utilisateur définit le nombre d'outils désiré mais le programme affectera pour chaque contour un seul outil optimum qui lui convient dans un plan donné (Figure 16).

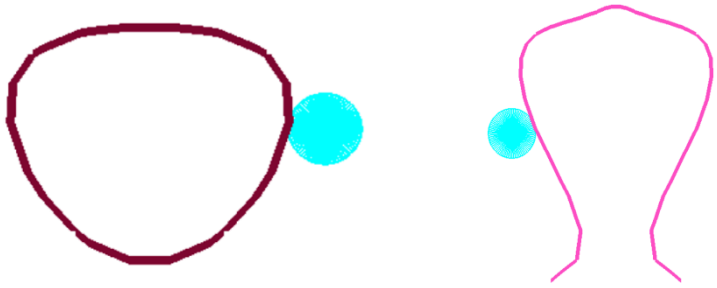


Figure 16 : Outils optimums des contours.

L'utilisation des outils optimums est importante dans l'opération d'évidement de poche pour éviter tous les problèmes d'interférences et de collisions (Figure 17).

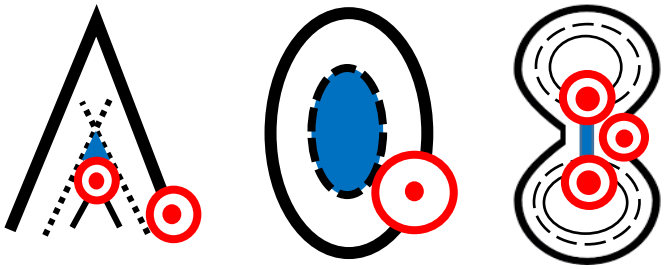
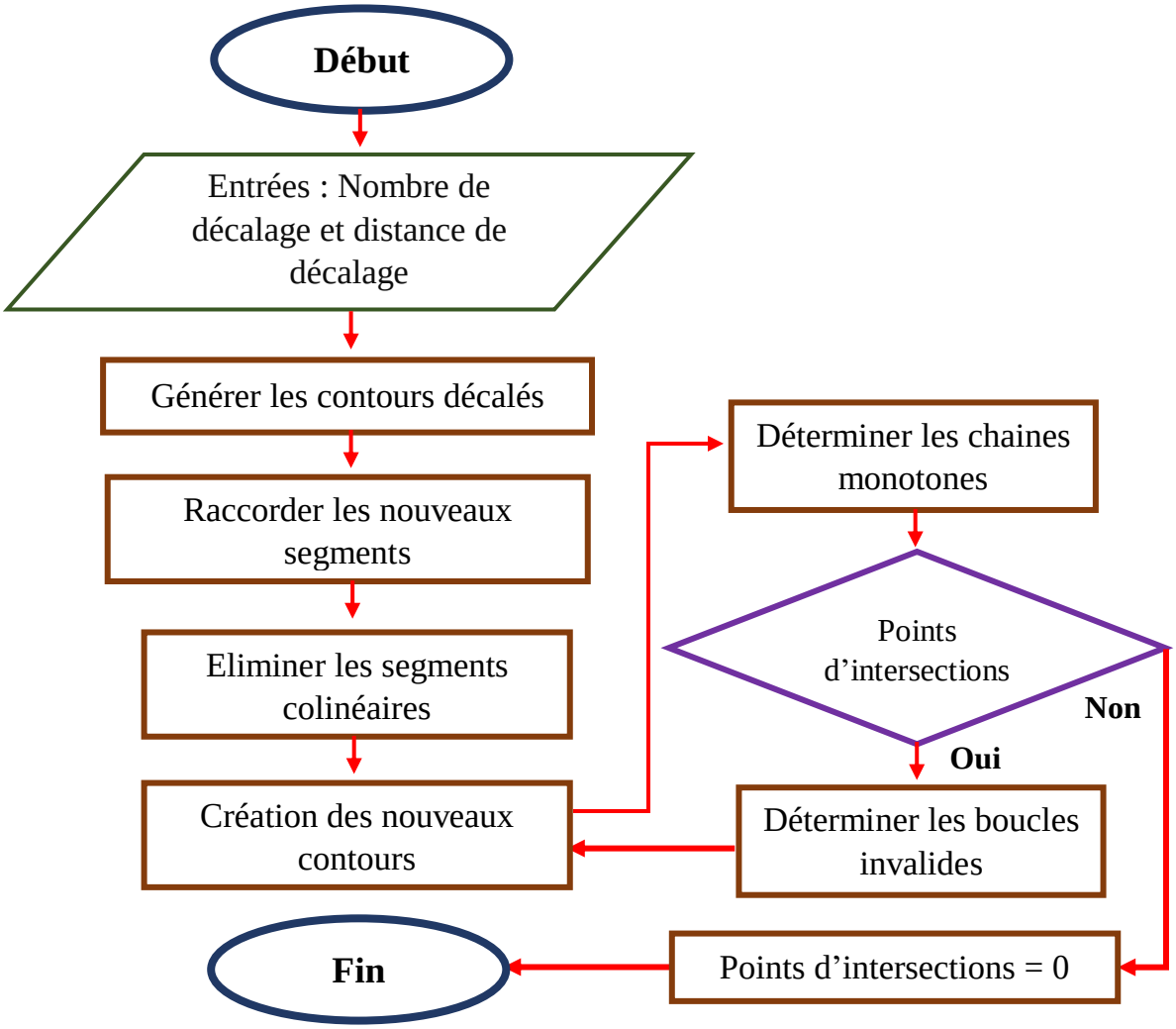


Figure 17 : Zones non usinées avec les contours décalés.

II.3.4. Génération des contours décalés

La génération des contours décalés passent par plusieurs étapes (Organigramme 2) :

1. Générer les contours décalés successifs.
2. Raccorder les nouveaux segments décalés.
3. Eliminer les segments colinéaires.
4. Création des nouveaux contours.
5. Générer le trajet.



Organigramme 2 : Génération des contours décalés.

II.3.4.1. Générer les contours décalés successifs

Comme les contours initiaux sont tangents avec la surface, il faut alors les décaler avec une surépaisseur et un nombre de décalage spécifiés par l'utilisateur (Figure 18).

Soit le segment $[AB]$ dans le plan, le décalage de $[A'B']$ dont $A\ (X_a, Y_a, Z_a)$, $B\ (X_b, Y_b, Z_b)$ et $A'\ (X_{á}, Y_{á}, Z_{á})$, $B'\ (X_{b́}, Y_{b́}, Z_{b́})$

$$X_{\acute{a}} = X_a + Décalage * N_x \; ; \quad (9)$$

$$Y_{\acute{a}} = Y_a + Décalage * N_y \; ; \quad (10)$$

$$X_{\acute{b}} = X_b + Décalage * N_x \; ; \quad (11)$$

$$Y_{\acute{b}} = Y_b + Décalage * N_y \; ; \quad (12)$$

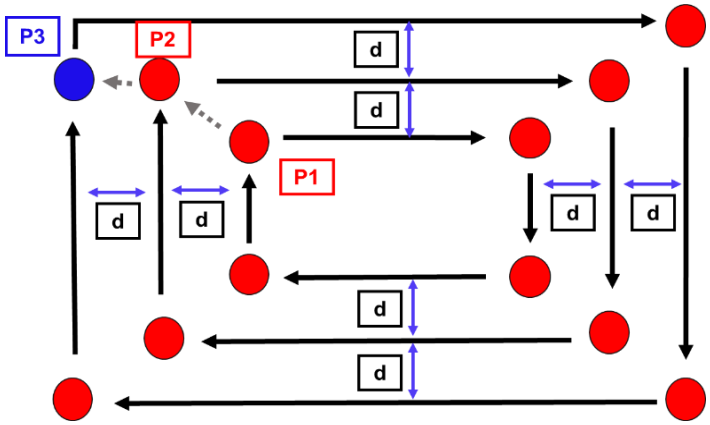


Figure 18 : Décalages successifs.

II.3.4.2. Raccordement des nouveaux segments décalés

Lors du décalage des segments du contour initial, les nouveaux segments décalés ne sont pas raccordés les uns aux autres. Dans ce cas, nous avons utilisés les équations paramétriques afin de trouver le point d'intersection entre les deux segments non raccordés et les relier en ce point (Figure 19).

Soient les segments $[AB]$ et $[A'B']$ dans le plan :

$$\overrightarrow{AM} = \alpha * \overrightarrow{AB} \quad (13)$$

$$\overrightarrow{A'B'} = \beta * \overrightarrow{\acute{A}\acute{B}} \quad (14)$$

D'où :

$$\begin{matrix} x & x_a \\ y & y_a \end{matrix} = \alpha * \begin{matrix} x_b & x_a \\ y_b & y_a \end{matrix} \quad (15)$$

$$\begin{matrix} x & x_{\acute{a}} \\ y & y_{\acute{b}} \end{matrix} = \beta * \begin{matrix} x_{\acute{b}} & x_{\acute{a}} \\ y_{\acute{b}} & y_{\acute{a}} \end{matrix} \quad (16)$$

Alors :

$$x_a + \alpha . (x_b - x_a) = x_{\acute{a}} + \beta . (x_{\acute{b}} - x_{\acute{a}}) \quad (17)$$

$$y_a + \alpha . (y_b - y_a) = y_{\acute{a}} + \beta . (y_{\acute{b}} - y_{\acute{a}}) \quad (18)$$

On pose :

$$a = (x_b - x_a) , b = -(x_{\acute{b}} - x_{\acute{a}}), c = (x_{\acute{a}} - x_a) \quad (19)$$

$$d = (y_b - y_a), e = -(y_{\acute{b}} - y_{\acute{a}}), f = (y_{\acute{a}} - y_a) \quad (20)$$

Alors :

$$a * \alpha + b * \beta = c \quad (21)$$

$$d * \alpha + e * \beta = f \quad (22)$$

Calculons le déterminant :

$$\Delta = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = a * d - c * b \quad (23)$$

$$\Delta_\alpha = \begin{vmatrix} c & b \\ f & e \end{vmatrix} = c * e - f * b \quad (24)$$

$$\Delta_\beta = \begin{vmatrix} c & a \\ f & d \end{vmatrix} = c * d - f * a \quad (25)$$

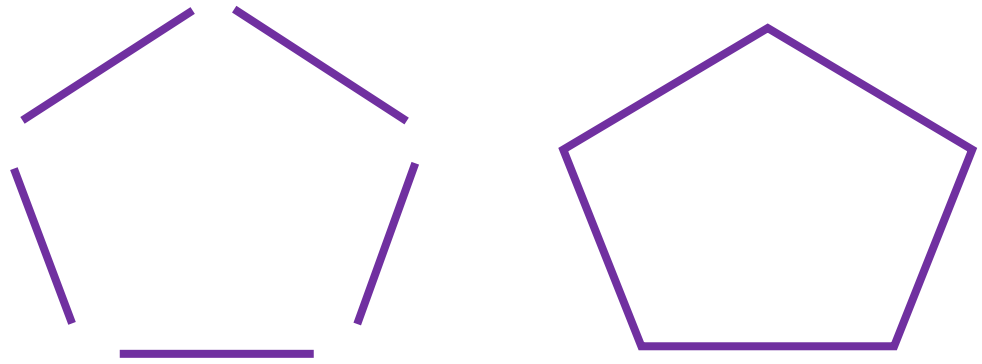
$$Si\ \Delta <> 0\ alors\ \alpha = \frac{\Delta_\alpha}{\Delta}\ \ \beta = \frac{\Delta_\beta}{\Delta} \quad (26)$$

Remplaçons par **α** :

$$X = \alpha * A + x_a \quad (27)$$

$$Y = \alpha * B + y_a \quad (28)$$

M (X, Y) est le point d'intersection entre les deux segments.



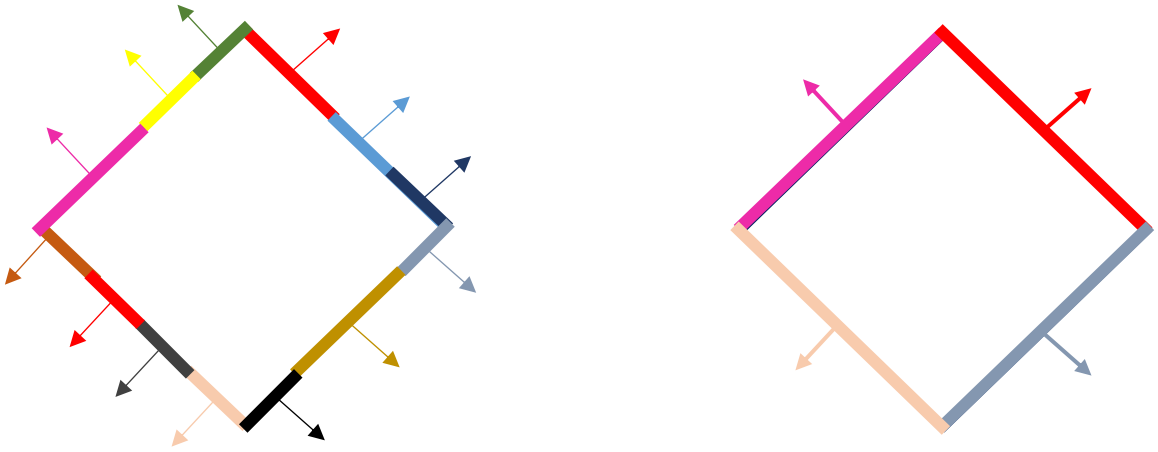
a. Segments avant raccordement. b. Segments après raccordement.

Figure 19 : Raccordement des segments.

II.3.4.3. Fusion des segments colinéaires

On rappelle que deux segments sont colinéaires si leurs vecteurs normaux sont identiques. Ce traitement a pour but de minimiser le calcul par rapport au nombre de segments par le remplacement de deux segments ou plus non colinéaires par un seul segment (Figure 20). L'algorithme suivant explique cette démarche :

```
Début
    Eliminer_Segments_Colinéaires ;
    Si Normal_Segment_1 == Normal_Segment_2 alors
        Remplacer le Sommet_2 du Segment_1 par Sommet_2 du Segment_2
        Nombre_Segments -- ;
    Finsi ;
Fin
```



a. Avant le fusionnement des segments. b. Après le fusionnement des segments.

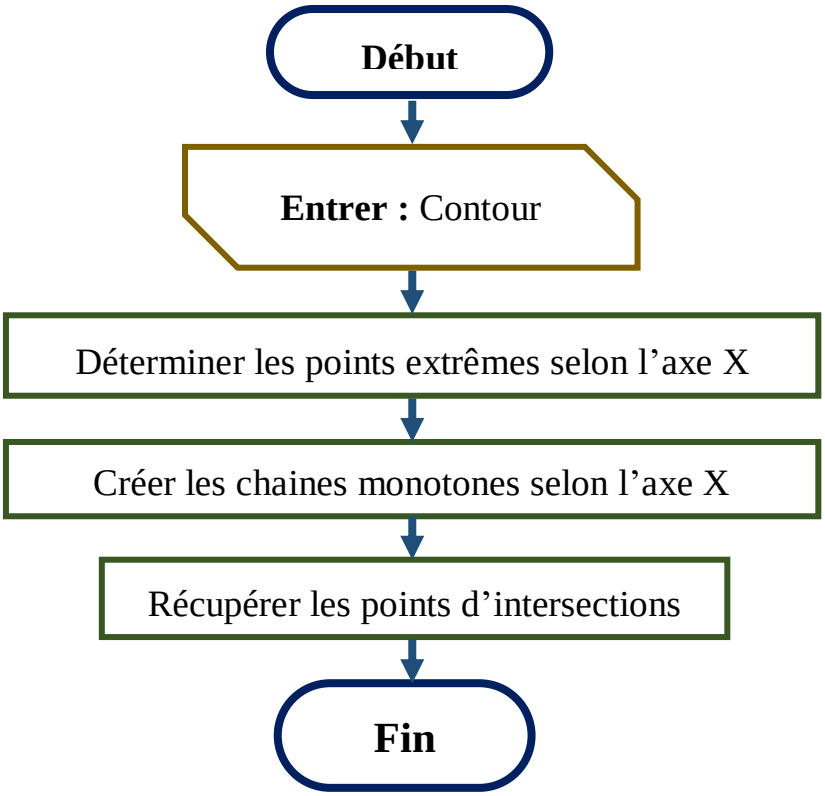
Figure 20 : Fusion des segments colinéaires.

II.3.4.4. Création des nouveaux contours

Les nouveaux contours seront créés après le décalage successif du contour initial. Ces contours risquent d'avoir des points d'auto-intersection ou des points d'intersection entre les contours ce qui forment des boucles invalides et par conséquent des problèmes d'usinage. Afin d'éliminer ou d'éviter ces problèmes nous avons utilisé l'approche de « Sweep-Line » pour détecter ces points d'intersection et donc éliminer les boucles invalides.

Cette approche est valable pour la détection d'un seul point d'intersection entre deux segments seulement. Nous avons pu l'améliorer pour pouvoir détecter plusieurs points d'intersections entre deux segments ou plus.

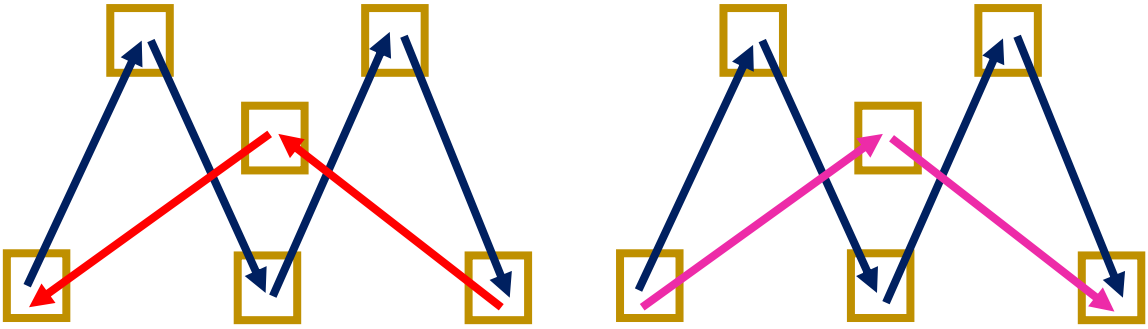
L'Organigramme 3 explique l'architecture de la création des nouveaux contours :



Organigramme 3 : Etapes de détection des points d'intersection.

1. Détermination des points extrêmes selon l'axe X

Avant de déterminer les points extrêmes, nous avons permuté tous les segments d'un contour. Le principe de permutation est de choisir une seule direction pour tous les segments. Nous avons choisi la direction de gauche à droite selon l'axe X (Figure 21).



a. Avant la permutation. b. Après la permutation.

Figure 21 : Permutation des segments.

Cette opération est faite comme suit selon l'axe X :

– Pour chaque segment, on compare le sommet de début avec le sommet de fin.
  - Si la valeur de X du sommet de début est plus grande que la valeur X du sommet de fin, alors on permute les deux sommets.
  - Sinon, on passe au segment suivant.

Après la permutation des segments, nous avons pu détecter les points extrêmes en traitant tous les cas possibles (Figure 22).

- **Cas simple**

  - Points extrêmes négatifs : ce sont les points de début pour chaque segment où plusieurs segments peuvent partager le même point.
  - Points extrêmes positifs : ce sont les points de fin pour chaque segment où plusieurs segments peuvent partager le même point.

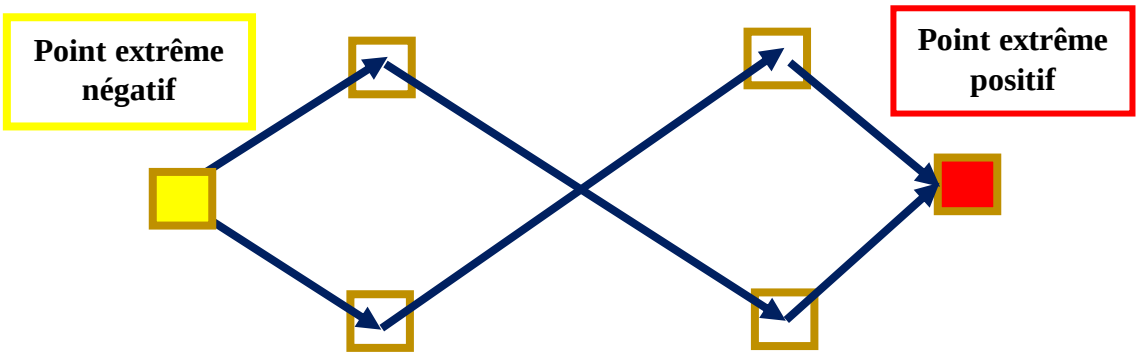


Figure 22 : Extrémités positives et négatives.

- **Cas dégénéré**

  - Un point peut être un point extrême positif et négatif en même temps (Figure 23).

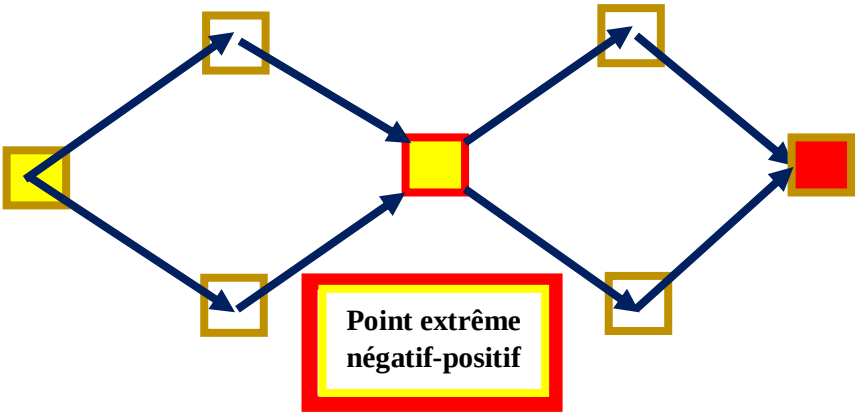


Figure 23 : Point extrême positif-négatif.

- **Cas particuliers**

Ce cas peut être présent quand pour un contour, un de ses segments ou plus sont verticaux selon l'axe X (Figure 24). La détermination des points extrêmes, dans ce cas passe par :

  - Si pour un segment, son sommet de début a les mêmes coordonnées qu'un sommet d'un segment vertical ou horizontal, alors c'est un point extrême négatif.

– Si pour un segment, son sommet de fin a les mêmes coordonnées qu'un sommet d'un segment vertical ou horizontal, alors c'est un point extrême positif.

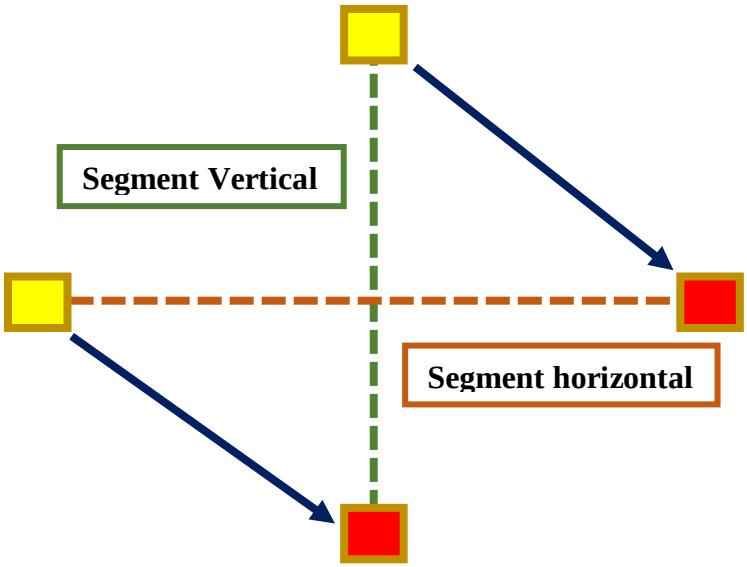


Figure 24 : Extrémités des segments verticaux et horizontaux.

2. Création des chaines monotones

Une chaine monotone est une séquence connectée de segments de lignes selon une direction (axe – X). Le sommet de début du premier segment de la chaine monotone est un point extrême négatif et le sommet de fin du dernier segment de la chaine monotone est un point extrême positif. Il existe deux types de chaines monotones :

– Chaines monotones non parallèles ne contenant pas de segments parallèles à la direction de balayage (Figure 25).
– Chaines monotones parallèles contenant au moins un segment parallèle à la direction de balayage (Figure 26).

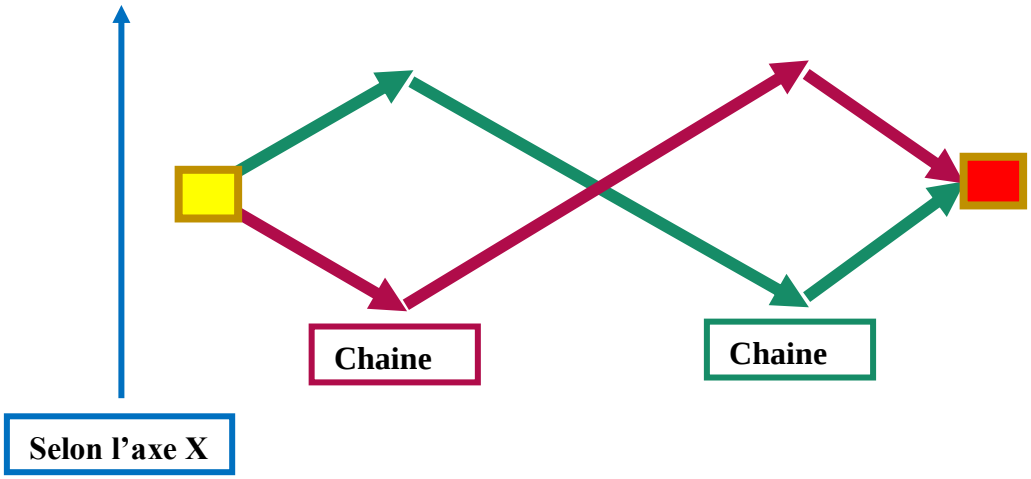


Figure 25 : Segments non parallèles à la direction X.

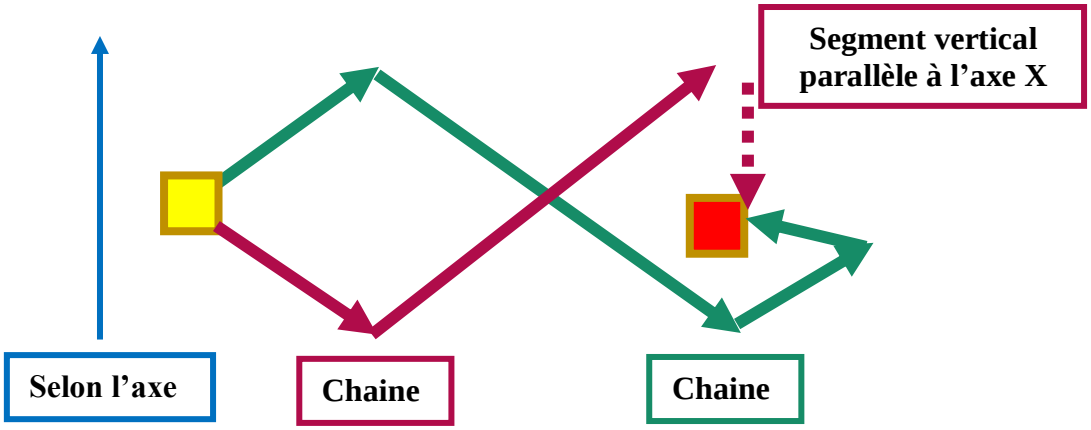


Figure 26 : Segments parallèles à la direction X.

3. Calcul des points d'intersection

Une « Sweep-Line » (ligne de balayage) est utilisée dans l'algorithme de recherche d'intersections entre segments utilisant une "ligne de balayage" virtuelle pour résoudre des problèmes dans l'espace euclidien. Dans celui-ci, chaque point d'évènement est soit un sommet d'un segment, soit une intersection entre deux segments, à chaque point, il est testé si deux segments voisins et traversés par la ligne se croisent. Ce principe est aussi celui de **l'algorithme de Bentley-Ottmann** dont la complexité temporelle est linéarithmique O(nlogn) (Figure 27).

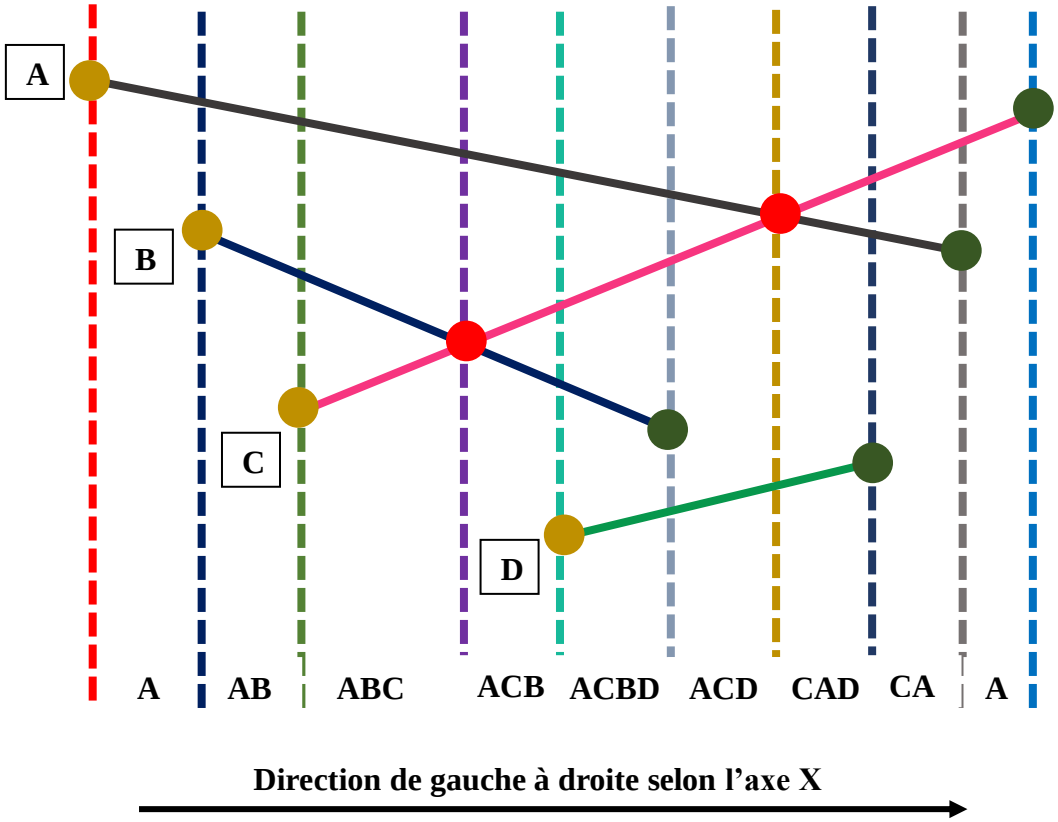


Figure 27 : Principe de l'algorithme de « Sweep-Line ».

La Figure 27 montre le déplacement de la ligne de balayage ***L*** de gauche à droite en coupant les segments en entrée [Sommet 1] dans l'ordre au fur et à mesure qu'elle se déplace. Les étapes sont comme suit :

Etape 1 : pour ce faire, nous avons commencé par le tri des extrémités des segments. La procédure de tri utilisée est le tri par sélection. Les extrémités ont été triées dans l'ordre croissant selon l'axe X.

Etape 2 : à tout instant de l'algorithme, la ligne de balayage va commencer par intersecter un segment à son extrémité gauche. Ce segment doit être stocké dans un tableau de segments.

Etape 3 : à son extrémité droite, le segment doit alors être supprimé de ce tableau, tout en maintenant sa liste triée par un ordre « dessus/dessous », c'est à dire que nous avons utilisé une fonction de tri suivant l'axe Y pour trier ces segments.

Etape 4 : quand deux segments se croisent, on échange leurs positions dans le tableau trié.

Etape 5 : cependant, quand une intersection entre deux segments est trouvée, elle est ajoutée à un tableau des points d'intersection toujours dans le même ordre, en faisant attention à ne pas ajouter deux fois les mêmes intersections.

Etape 6 : on détermine leurs intersections potentielles avec les autres segments, et on ajoute les intersections effectives à la file d'attente des événements.

Etape 7 : quand on traite un événement correspondant à une intersection, on l'ajoute au tableau des intersections et on réorganise ce dernier à chaque ajout.

Lors de ce traitement, une intersection peut être détectée tout en vérifiant si deux chaines monotones sont intersectées car il n'y a pas d'intersections dans la même chaine monotone (Figure 28).

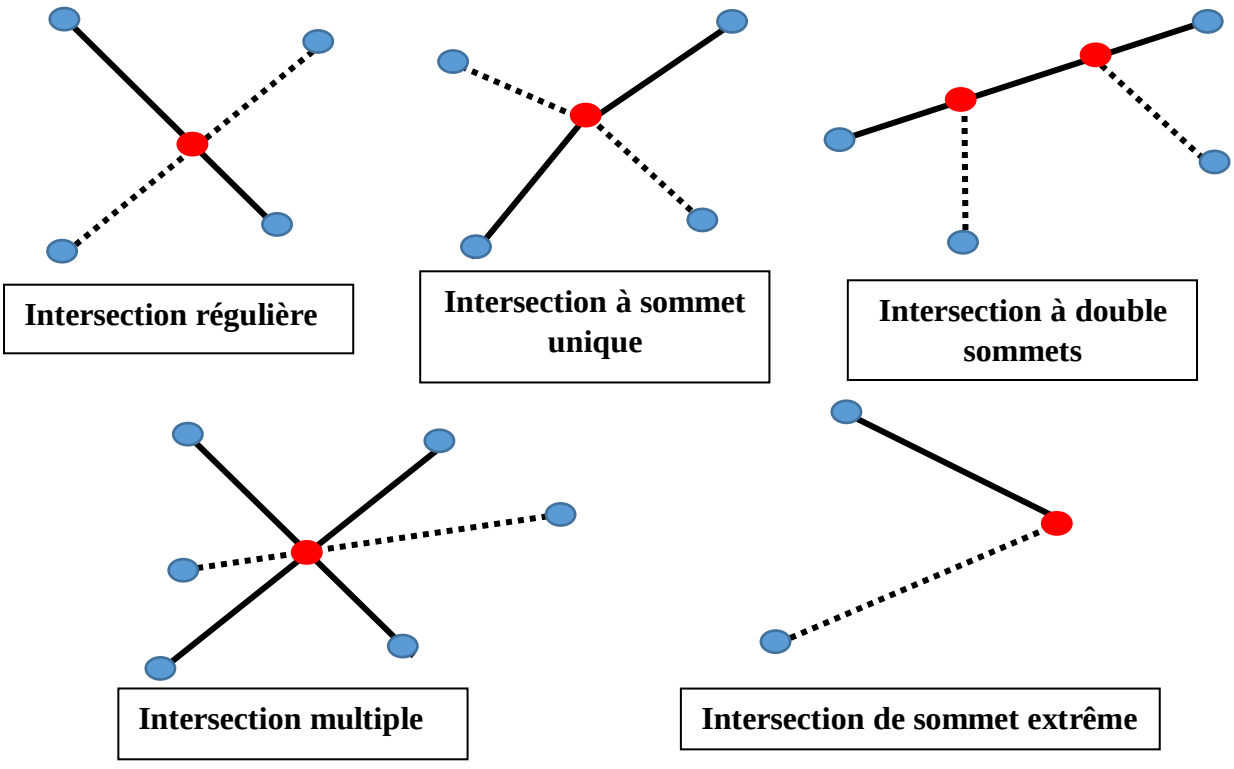


Figure 28 : Différents types d'intersections.

Ce point d'intersection **I** entre deux segments $P_i\ P_{i+1}$ **et** $P_k\ P_{k+1}$ (Figure 29) est calculé comme suit :

$$I = P_i + t\overrightarrow{P_iP_{i+1}} = P_k + s\overrightarrow{P_kP_{k+1}} \quad (29)$$

On note :

$$P_i\ P_{i+1} = A \quad (30)$$

$$P_k\ P_{k+1} = B \quad (31)$$

Soit P le point d'intersection :

$$P = P_i + t.A \quad (32)$$

$$P = P_k + s.B \quad (33)$$

D'où :

$$P_i + t.A = P_k + s.B \quad (34)$$

Les paramètres t et s ont été obtenus par la résolution de l'équation précédente par :

$$t = \frac{\overrightarrow{P_iP_k} * \overrightarrow{P_kP_{k+1}}}{\overrightarrow{P_iP_{i+1}} * \overrightarrow{P_kP_{k+1}}} \quad (35)$$

$$s = \frac{\overrightarrow{P_iP_k} * \overrightarrow{P_iP_{i+1}}}{\overrightarrow{P_iP_{i+1}} * \overrightarrow{P_kP_{k+1}}} \quad (36)$$

Pour que le point d'intersection soit valide, il faut que $0 \leq t \leq 1$ **et** $0 \leq s \leq 1$

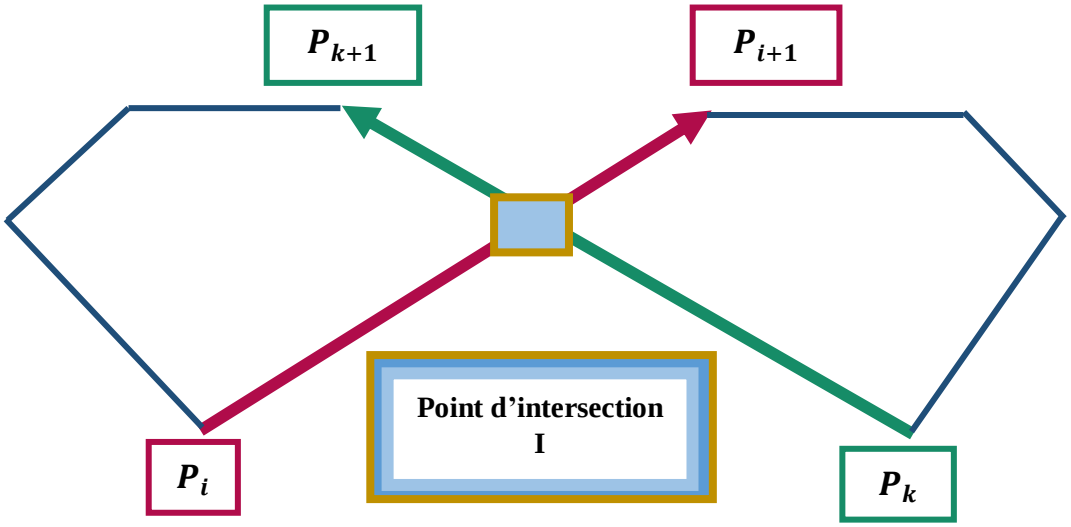


Figure 29 : Calcul du point d'intersection.

La détection des points d'intersection montre les boucles invalides (auto-intersection) à éliminer (Figure 30). L'élimination de ces boucles est nécessaire pour l'obtention d'un trajet d'usinage sain. Ceci, nous mène à une solution de subdivision des segments pour la composition des nouveaux contours fermés.

4. Création des nouveaux contours

Après la détection des points d'intersection, est donc des problèmes d'usinage qui doivent être éliminés. Parmi ces problèmes on cite :

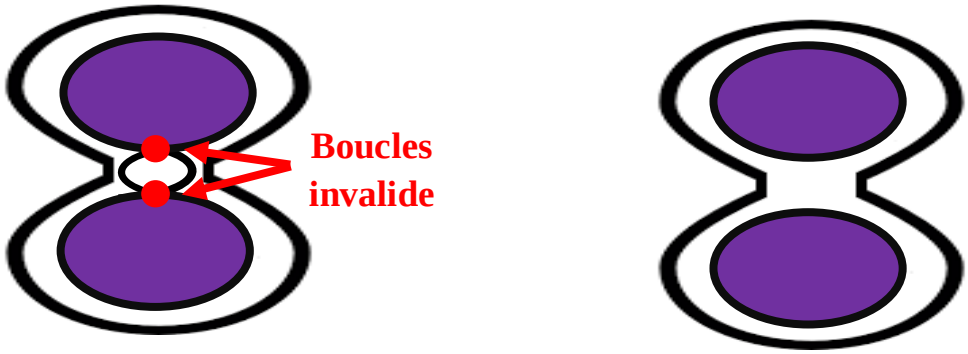
- Zone de rebroussement.
- Boucle invalide inter-contour.
- Boucle invalide intra-contour.

4.1. Zone de rebroussement : la zone de rebroussement peut apparaitre lors un décalage où deux segments se coupent à l'intérieur de leurs limites en formant une boucle invalide. Cette dernière doit être éliminée (Figure 30).

a. Avant l'élimination de la zone. b. Après l'élimination de la zone.

Figure 30 : Zone de rebroussement.

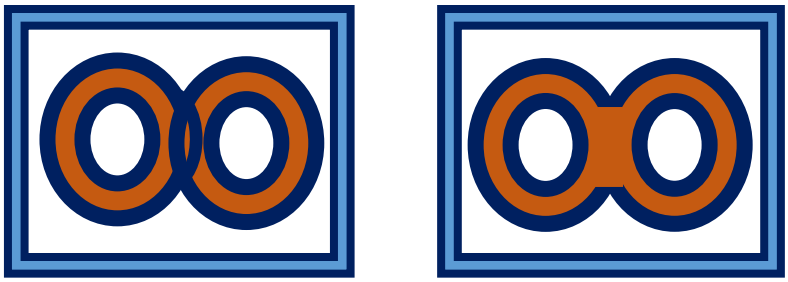
4.2. Boucle invalide intra-contour : lors du décalage d'un contour quelconque, des zones indéfinies ont été créées entre les segments du même contour (Figure 31).



a. Avant la suppression de la boucle. b. Après la suppression de la boucle.

Figure 31 : Boucle invalide intra-contour.

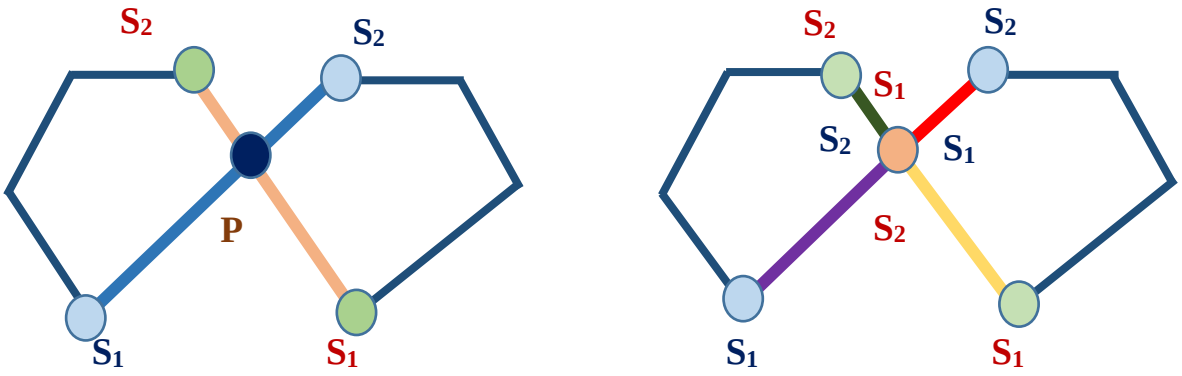
4.3. Boucle invalide inter-contour : ce problème est la conséquence du décalage d'un contour quelconque qui est à l'intérieur d'un autre contour. En d'autres termes, le décalage des contours fils provoquent ce genre d'intersections avec le contour père (Figure 32).



a. Avant la suppression de la boucle. b. Après la suppression de la boucle.

Figure 32 : Boucle invalide inter-contour.

La solution proposée pour l'élimination de ces problèmes, est la subdivision des segments dans un contour donné, lorsque deux segments se coupent en un point donc existence de point d'intersection entre deux segments, nous avons pensés à subdiviser ces deux segments chacun en deux. C'est-à-dire, pour chaque segment on lui a affecté le point d'intersection comme sommet 1 ou sommet 2. Prenons un segment, son premier sommet est aussi le premier sommet du segment subdivisé et son deuxième sommet est le point d'intersection alors c'est la première partie subdivisée du segment. Pour la deuxième partie subdivisée prenons le point d'intersection comme son premier sommet et le deuxième sommet du segment courant comme deuxième sommet (Figure 33).



a. Avant la subdivision. b. Après la subdivision.

Figure 33 : Subdivision des segments.

Pour nous faciliter la tâche, nous avons utilisé des booléens « Entré/Sortie » pour chaque nouveau segment subdivisé. Nous remarquons que le sens des nouveaux contours est différent l'un à l'autre, le sens est inversé (Figure 34).

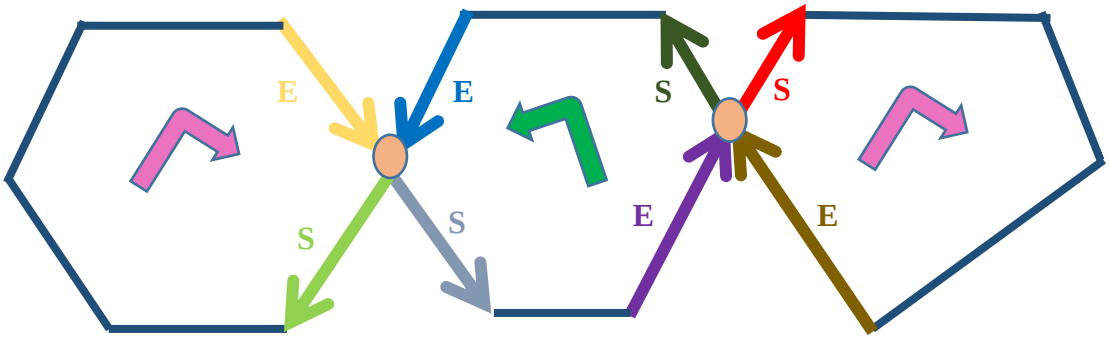


Figure 34 : Nominalisation des segments subdivisés.

La création des nouveaux contours passe par deux étapes :

1. Création du premier contour.
2. Création des autres contours.

Les nouveaux contours sont créés suivant cette nominalisation. Nous avons traité tous les cas possibles :

- Si pour un contour, l'indice de son premier segment est « Entré » alors le segment suivant son indice est « Sortie », ce segment doit être vérifié ;
  - Il faut vérifier que ce segment n'est pas déjà « sélectionné ».
  - Il faut vérifier la normale de ce segment, elle doit être différente du segment précédent.
  - Il faut fermer le contour en vérifiant le premier sommet du premier segment avec le dernier sommet du dernier segment.
- Si pour un contour l'indice de son premier segment est « Sortie » alors le dernier segment son indice est « Entré », ce segment doit être vérifié ;
  - Il faut vérifier que ce segment n'est pas déjà « sélectionné ».
  - Il faut vérifier la normale de ce segment, elle doit être différente de la normale d'un segment indexé par « Sortie ».
  - Il faut fermer le contour en vérifiant le premier sommet du premier segment avec le dernier sommet du dernier segment.

Après la création de ces nouveaux contours, on doit vérifier le sens de chaque contour :

- Si pour un contour, son sens est inversé par rapport au premier contour alors on le supprime (Figure 35).

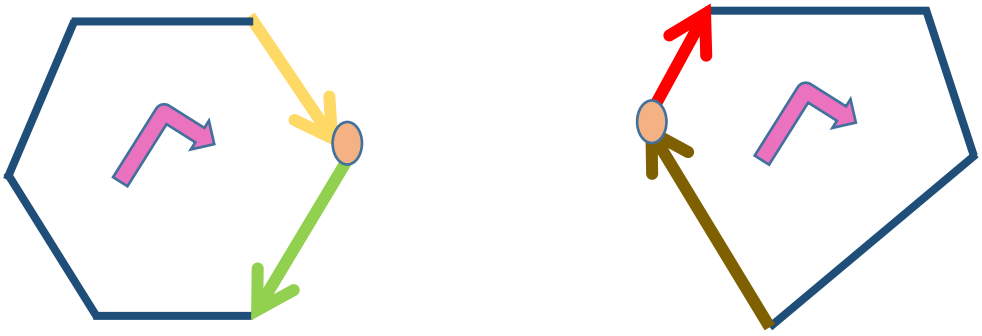


Figure 35 : Nouveaux contours.

II.4. Modélisation avec UML

L'UML *(Unified Modeling Language*) ou Langage de modélisation unifiée en français est un langage graphique de modélisation informatique. Ce langage est désormais la référence en modélisation objet ou programmation orientée objet. Cette dernière consiste à modéliser des éléments du monde en un ensemble d'entités informatiques appelées « objet ». Il est constitué de diagrammes qui servent à visualiser et à décrire la structure et le comportement des objets qui se trouvent dans un système. Dans notre projet, nous avons utilisé les diagrammes suivants :

- Diagramme de cas d'utilisation.
- Diagramme d'activité.
- Diagramme de séquence.
- Diagramme de classe.

II.4.1. Diagramme de cas d'utilisation

Généralement les utilisateurs ne sont pas des informaticiens, pour exprimer leurs besoins, les diagrammes de cas d'utilisation sont le moyen le plus simple. Ils illustrent et définissent le contexte et les exigences d'un système entier, ou des parties essentielles d'un système. Ils décrivent les fonctions générales et identifient également les interactions entre le système et ses acteurs.

➢ **Diagramme de cas d'utilisation général :** le système est divisé en quatre grandes phases (Figure 36) :

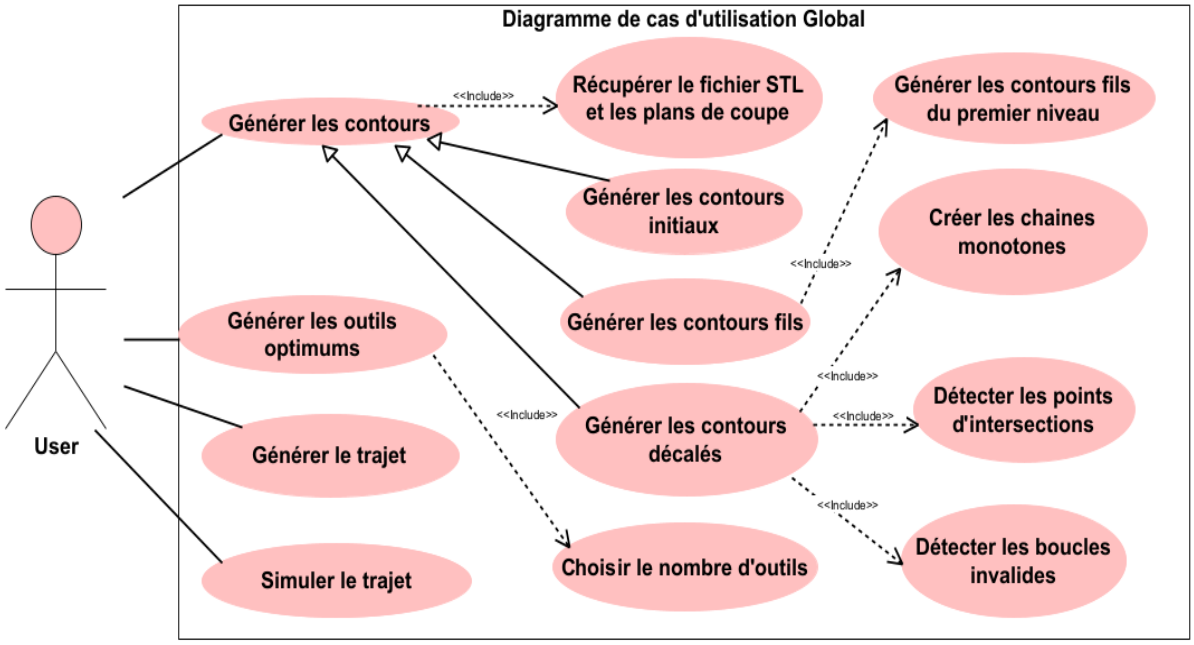


Figure 36 : Diagramme de cas d'utilisation général.

- **Diagramme de cas d'utilisation « Générer les contours initiaux » (**Figure 37)

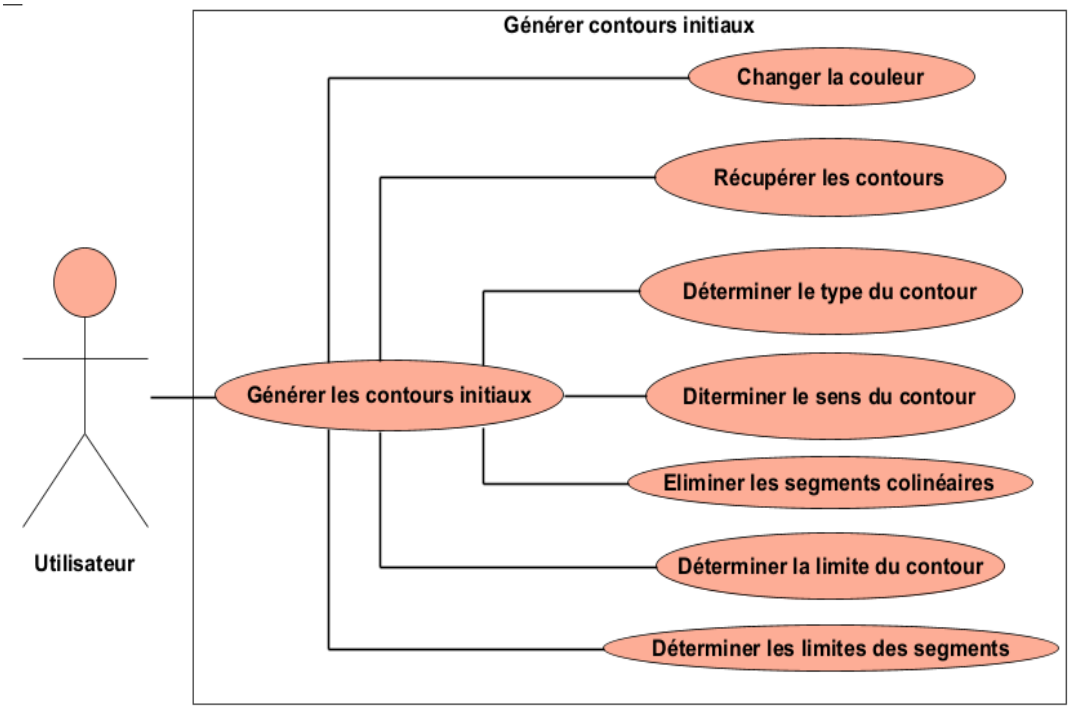


Figure 37 : Diagramme de cas d'utilisation « Générer les contours initiaux ».

- **Diagramme de cas d'utilisation « Générer les contours décalés » (**Figure 38) :

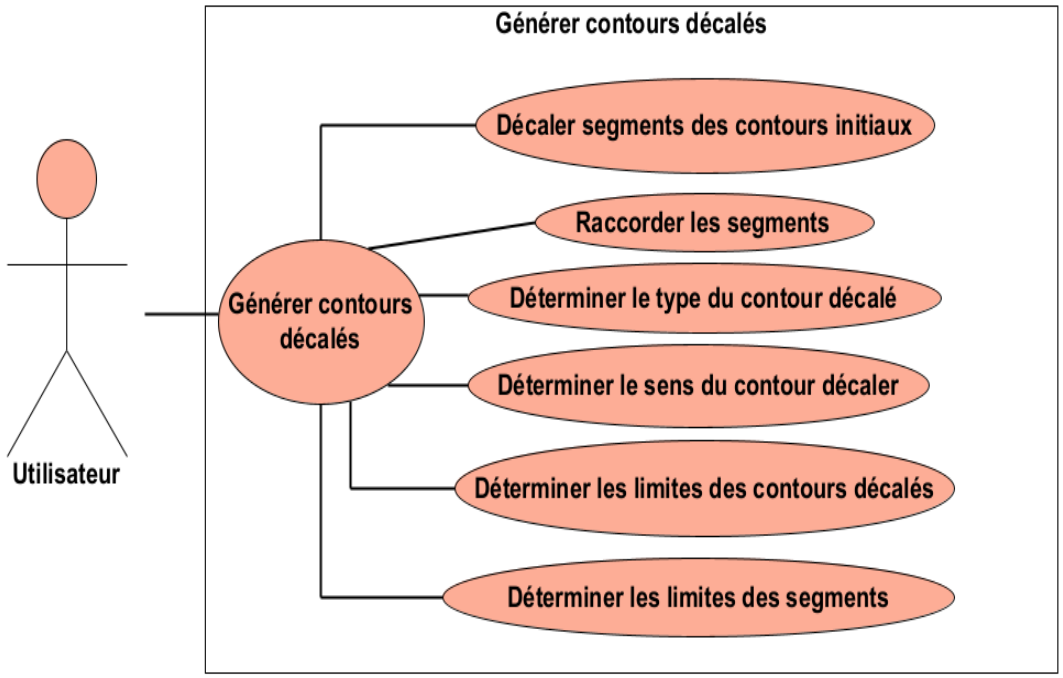


Figure 38 : Diagramme de cas d'utilisation « Générer les contours décalés ».

➢ **Diagramme de cas d'utilisation « Générer les contours fils » (**Figure 39)

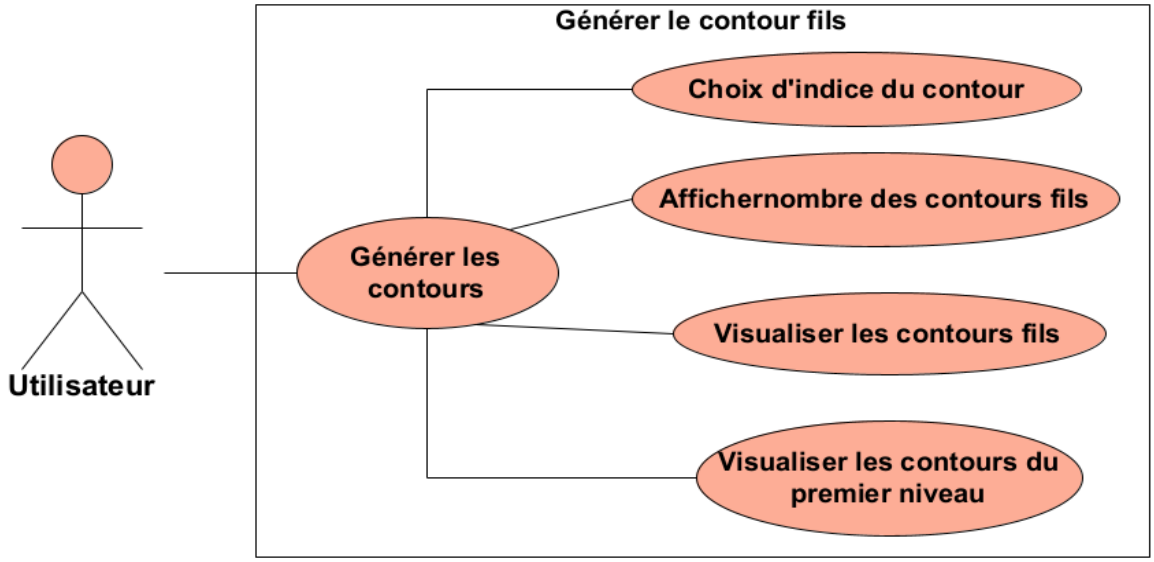


Figure 39 : Diagramme de cas d'utilisation « Générer les contours fils ».

II.4.2. Diagramme d'activités

Avant de se lancer dans la réalisation d'un logiciel, il faut comprendre, clarifier et structurer les attentes et les besoins du client. Un diagramme d'activité fournit une vue du comportement d'un système en décrivant la séquence d'actions d'un processus. Les diagrammes d'activité sont similaires aux organigrammes de traitement de l'information, car ils montrent les flux entre les actions dans une activité.

➢ **Diagramme d'activités général**

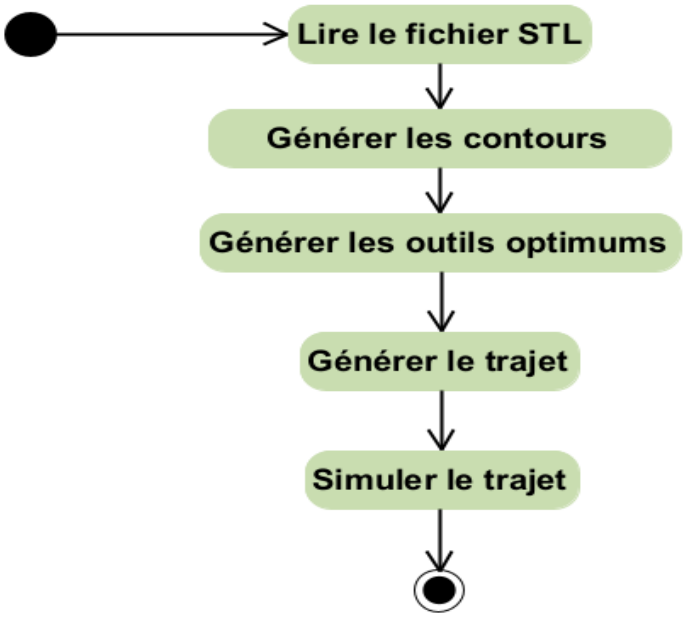


Figure 40 : Diagramme d'activités général.

➢ **Diagramme d'activités « Lecture du fichier STL »**

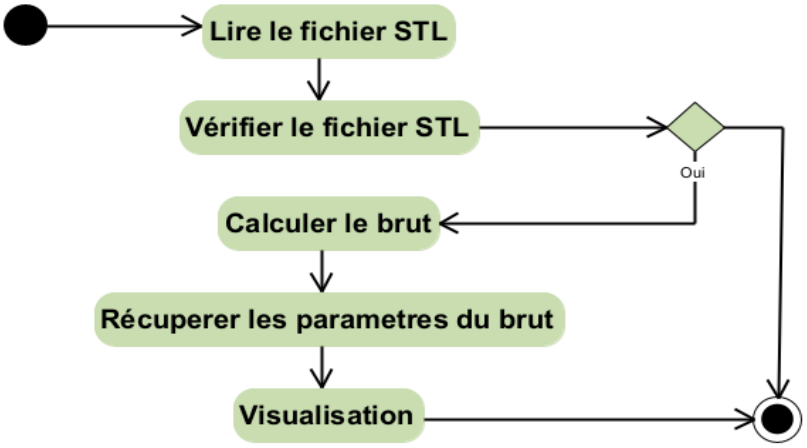


Figure 41 : Diagramme d'activités « Lecture du fichier STL ».

➢ **Diagramme d'activités « Création des chaines monotones »**

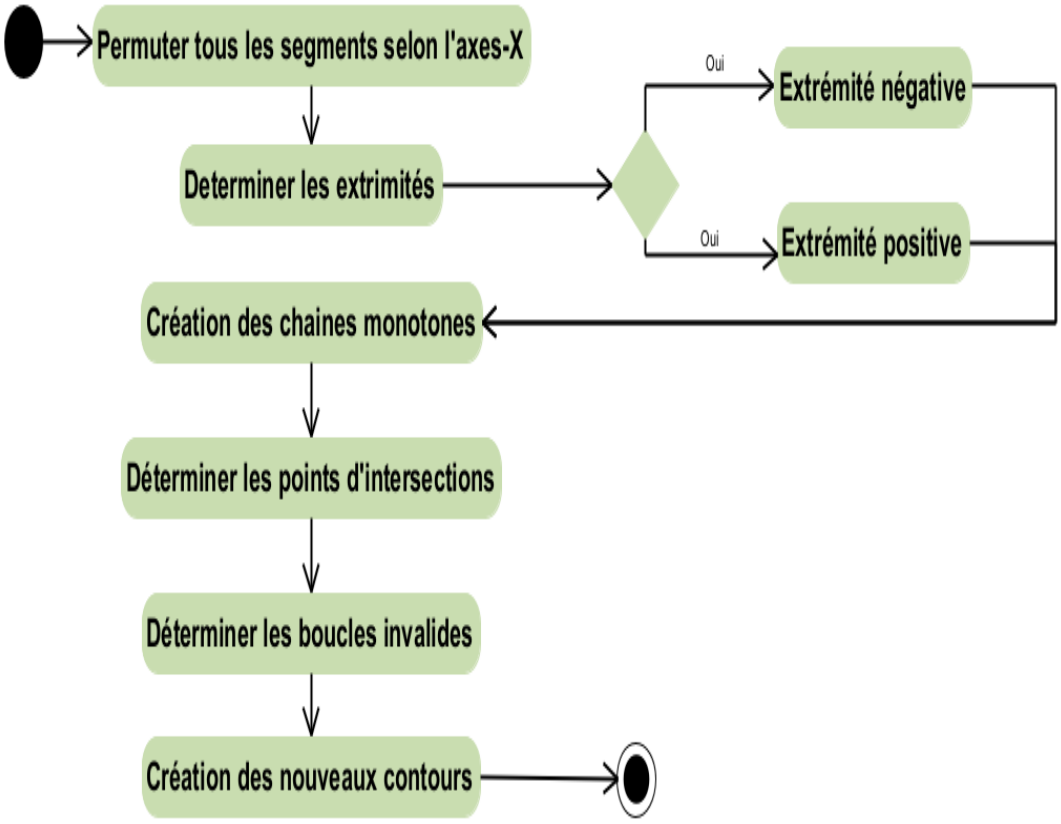


Figure 42 : Diagramme d'activités « Création des chaines monotones ».

II.4.3. Diagramme de séquence :

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language.

➢ **Diagramme de séquence « Sweep-Line »**

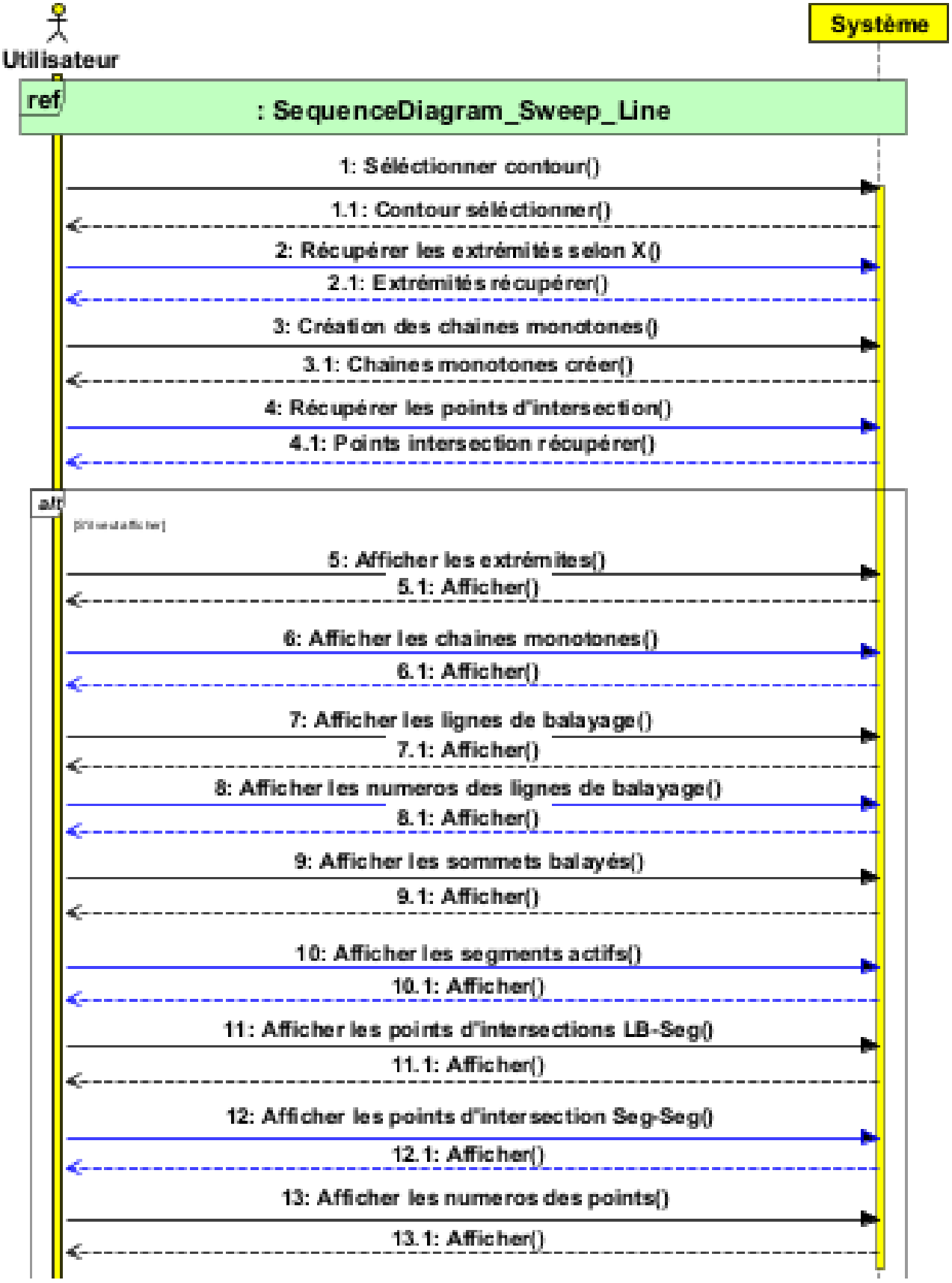


Figure 43 : Diagramme de séquence «Sweep-Line ».

➢ **Diagramme de séquence « Détection des points d'intersections »**

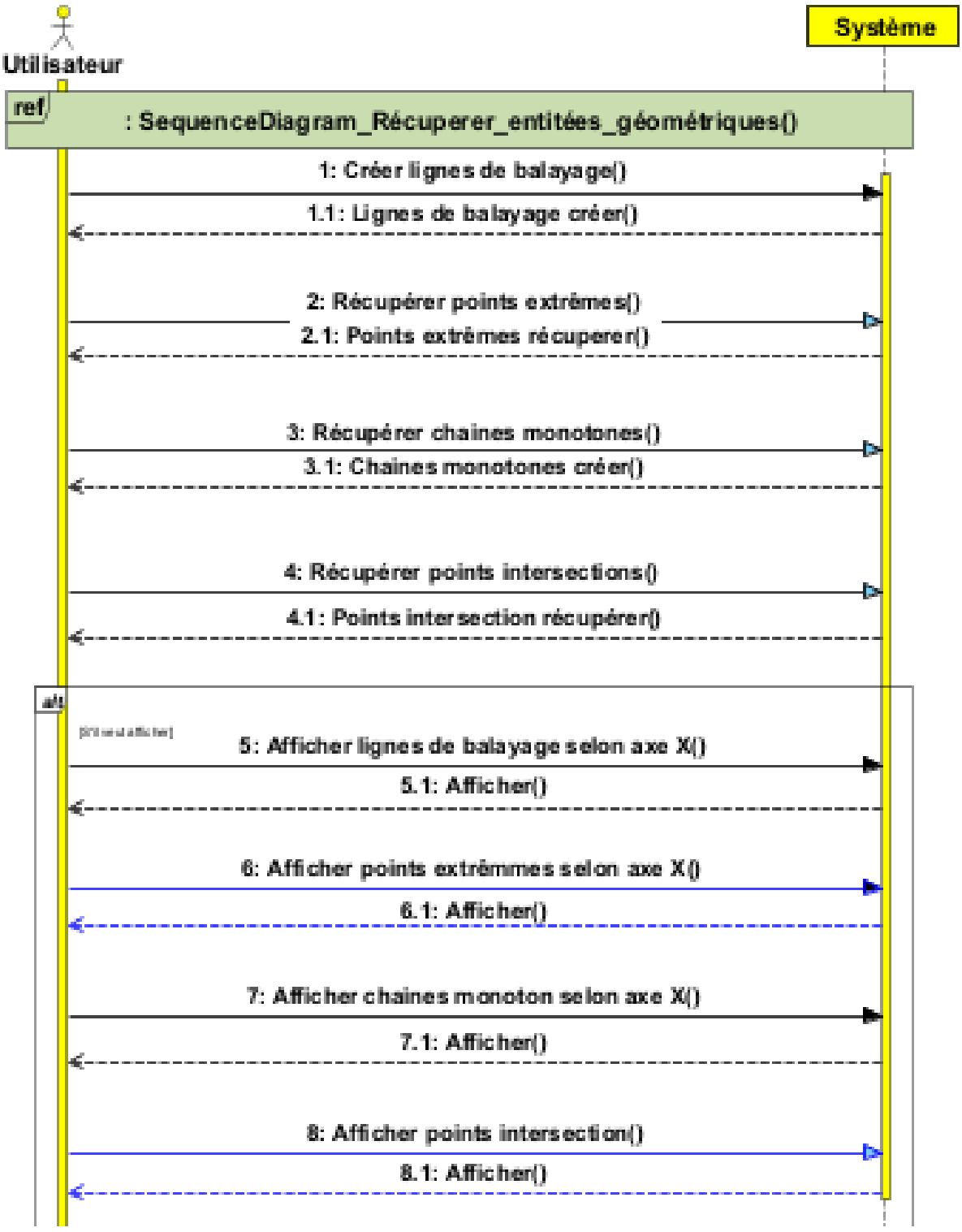


Figure 44 : Diagramme de séquence « Détection des points d'intersection ».

II.4.4. Diagramme de classe

Le diagramme de classe monte une vue globale du logiciel en présentant ses classes et ses attributs, ses opérations et les relations entre ses objets. Il s'agit d'une illustration des modèles de données pour des systèmes d'information, pour mieux comprendre l'aperçu général des schémas d'une application quel que soit leur degré de son complexité. De ce fait, nous présentons le diagramme de classe de notre système (Figure 45) en détaillant les classes nécessaires utilisées :

- Classe Couleur_Ebauchage_Contour_Decale
- Classe Brute_Ebauchage_Contour_Decale
- Classe ModeleSTL_Ebauchage_Contour_Decale
- Classe Sommet_Couleur_Ebauchage_Contour_Decale
- Classe Normal_Ebauchage_Contour_Decale
- Classe Triangle_Ebauchage_Contour_Decale
- Classe Segment_Ebauchage_Contour_Decale
- Classe Outil_Ebauchage_Contour_Decale
- Classe Extreme_Ebauchage_Contour_Decale
- Classe Chaine_Ebauchage_Contour_Decale
- Classe Sweep_Line_Ebauchage_Contour_Decale
- Classe Sommet_Balayer_Ebauchage_Contour_Decale
- Classe_PI_SL_Seg_Ebauchage_Contour_Decale
- Classe_PI_Seg_Seg_Ebauchage_Contour_Decale
- Classe Contour_Ebauchage_Contour_Decale
- Classe Plan_Ebauchage_Contour_Decale
- Classe Trajet_Ebauchage_Contour_Decale

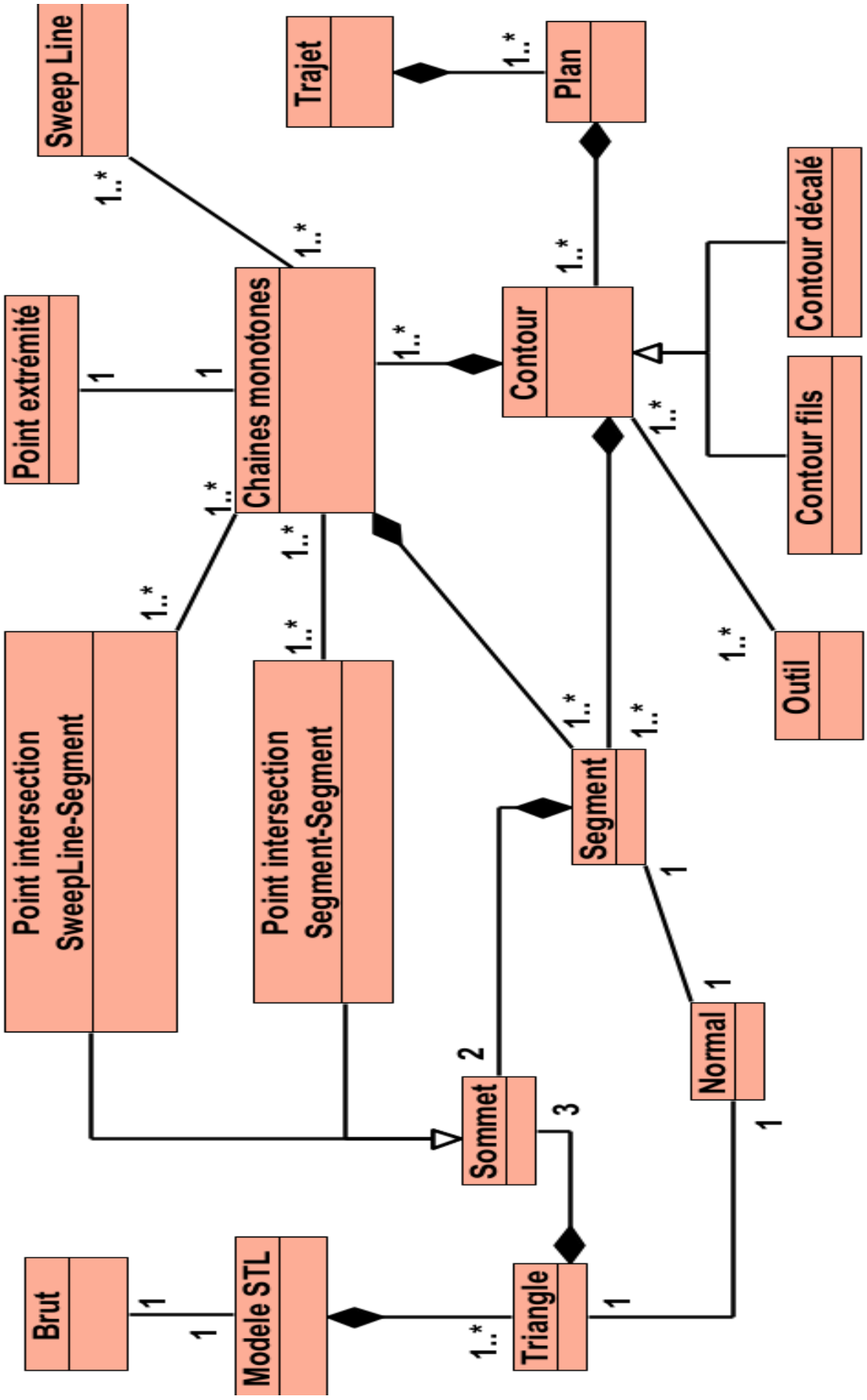


Figure 45 : Diagramme de classe.

Classe « Couleur_Ebauchage_Contour_Decale » : elle regroupe les trois paramètres qui représentent la génération des couleurs des contours aléatoirement.

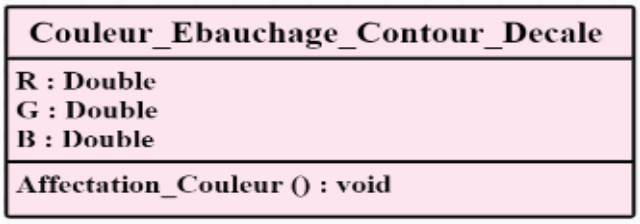


Figure 46 : Classe couleur.

- **Classe « Brute_Ebauchage_Contour_Decale » :** elle contient les limites du brut.

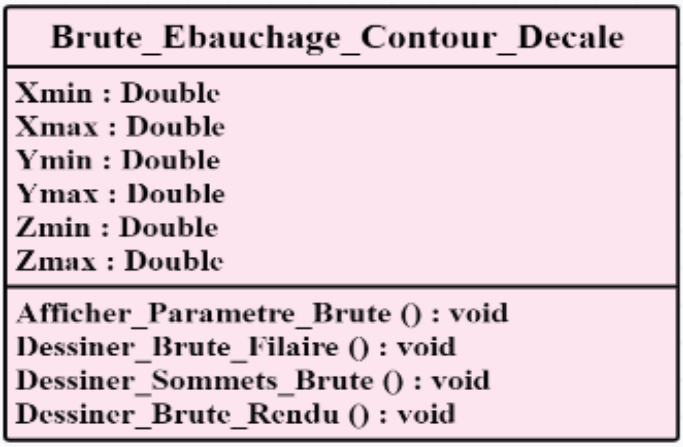


Figure 47 : Classe brute.

- **Classe « Sommet_Ebauchage_Contour_Decale » :** elle regroupe les coordonnées d'un sommet ***(X, Y, Z)***

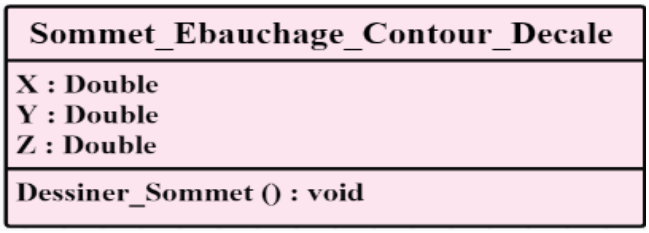


Figure 48 : Classe sommet.

- **Classe « Normal_Ebauchage_Contour_Decale »** : elle décrit les composantes de la normale d'un segment.

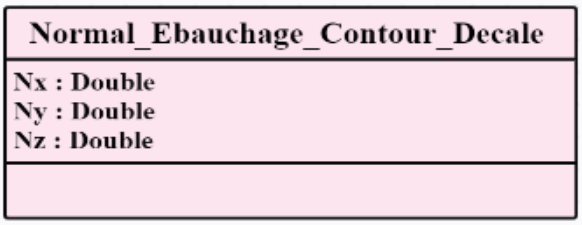


Figure 49: Classe normale.

➢ **Classe « Triangle_Ebauchage_Contour_Decale » :** cette classe contient les trois sommets du triangle ainsi que sa normale.

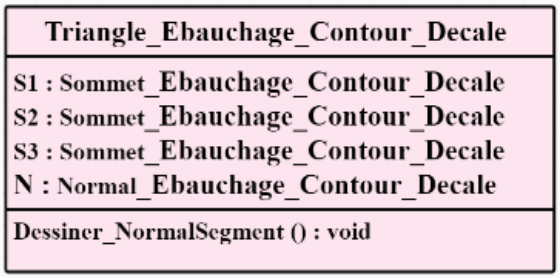


Figure 50 : Classe triangle.

➢ **Classe « Outil_Ebauchage_Contour_Decale » :** elle représente un outil récupéré depuis une base de données.

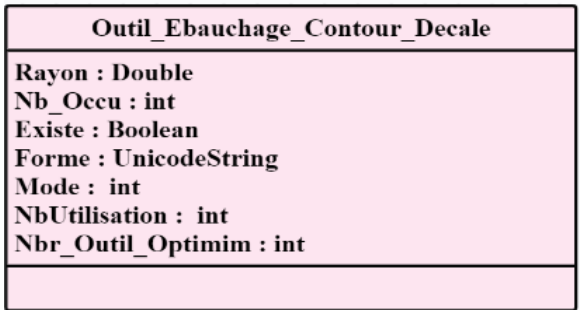


Figure 51 : Classe outil.

➢ **Classe « Segment_Ebauchage_Contour_Decale » :** cette classe définit les différentes caractéristiques d'un segment.

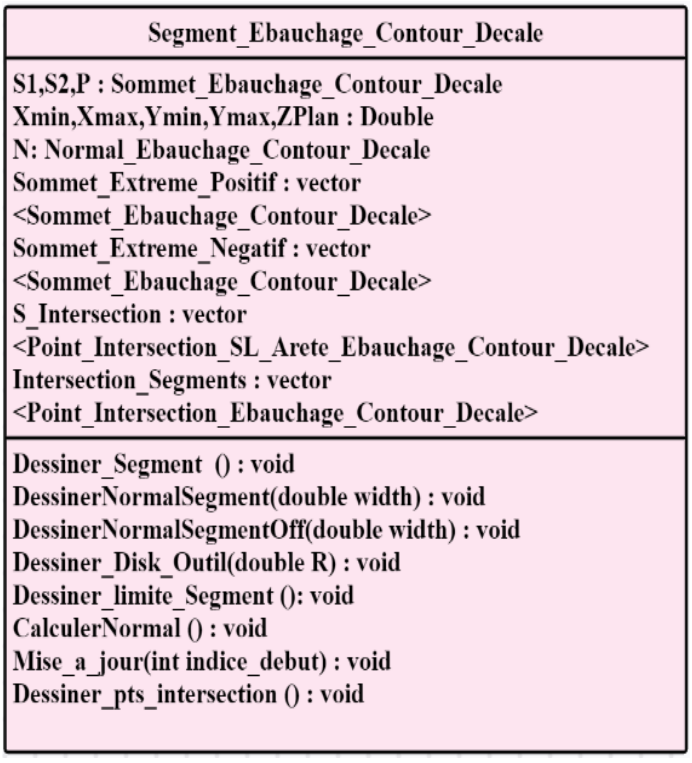


Figure 52 : Classe segment.

➢ **Classe « Extreme_Ebauchage_Contour_Decale » :** elle regroupe les paramètres et les fonctions appliquées sur les extrémités.

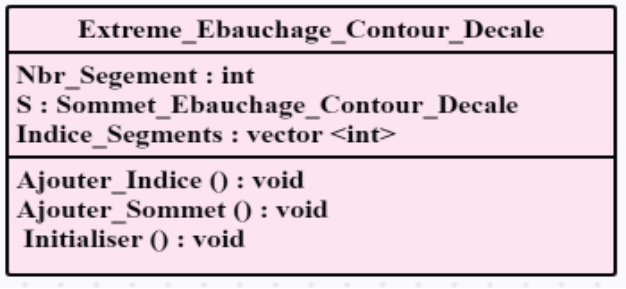


Figure 53: Classe extrêmes.

➢ **Classe « Chaine_Ebauchage_Contour_Decale » :** cette classe représente une chaine monotone qui est un vecteur de segments avec les différentes fonctions de calcul.

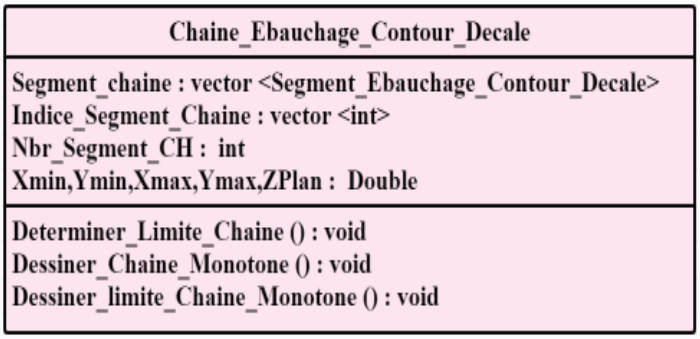


Figure 54 : Classe chaine monotone.

➢ **Classe « Sommet_Balayer_Ebauchage_Contour_Decale » :** la classe du sommet balayer représente le sommet sélectionné pas la ligne de balayage.

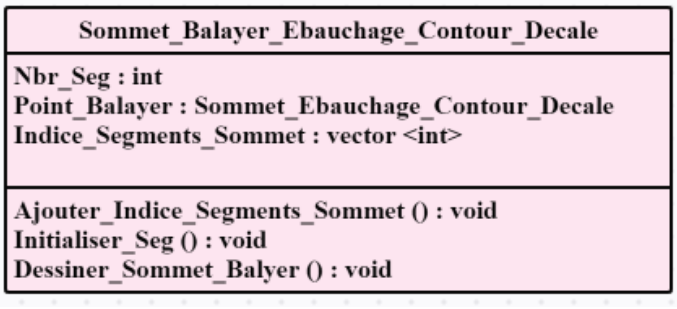


Figure 55 : Classe sommet balayer.

➢ **Classe « PI_SL_SEG_Ebauchage_Contour_Decale » :** cette dernière est le point d'intersection entre la ligne de balayage et le segment.

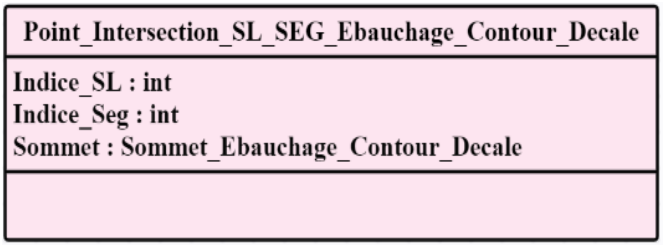


Figure 56 : Classe PI Sweep_Line_Segment.

➢ **Classe « Sweep_Line_Ebauchage_Contour_Decale » :** la classe Sweep-Line représente la ligne de balayage qui se déplace tout au long du contour de gauche à droite. Elle définit chaque ligne de balayage avec les segments dont leurs premier sommet est un sommet balayé ou sommet d'intersection entre la ligne de balayage et le segment ainsi que les fonctions de dessin.

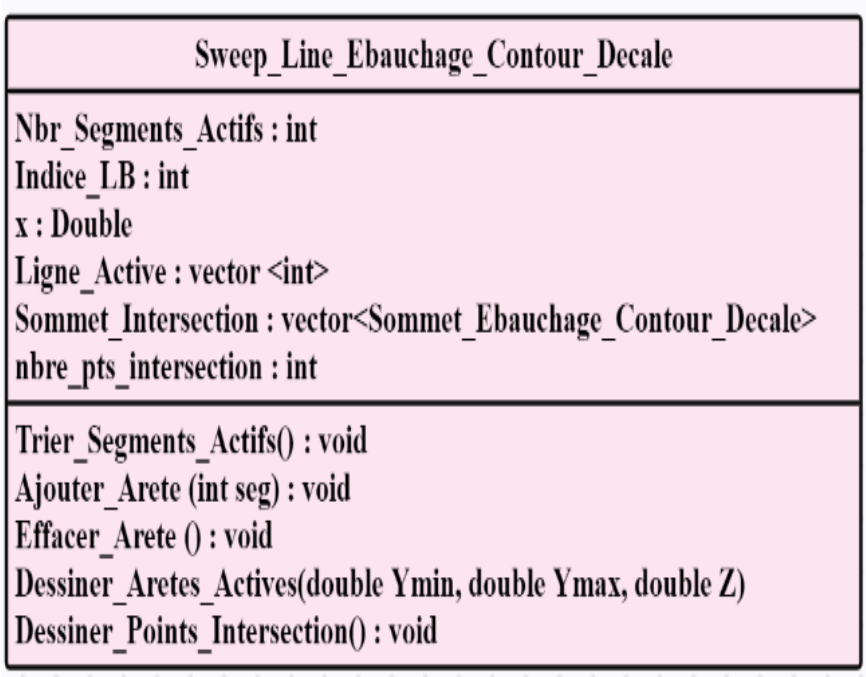


Figure 57 : Classe Sweep-Line.

➢ **Classe « PI_SEG_SEG_Ebauchage_Contour_Decale » :** cette classe regroupe les points d'intersection entre les segments dans un contour :

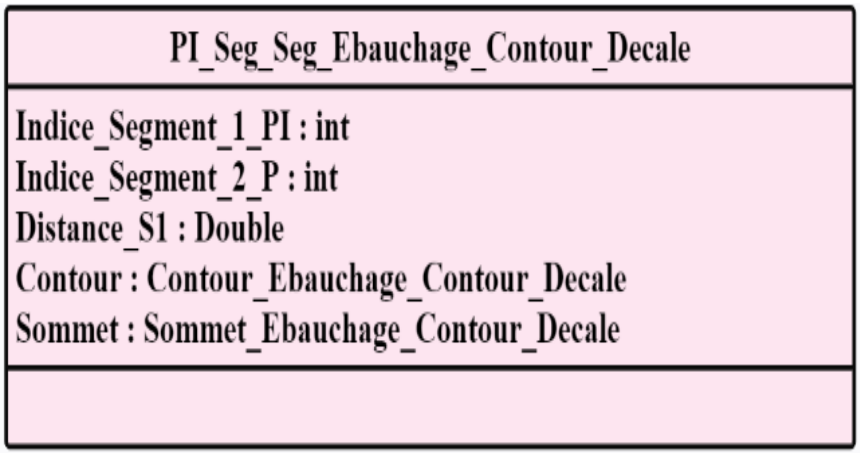


Figure 58: Classe PI Segment_Segment.

➢ **Classe « Contour_Ebauchage_Contour_Decale » :** la classe contour est une classe indispensable dans notre projet car elle regroupe tous les attributs et les fonctions nécessaires.

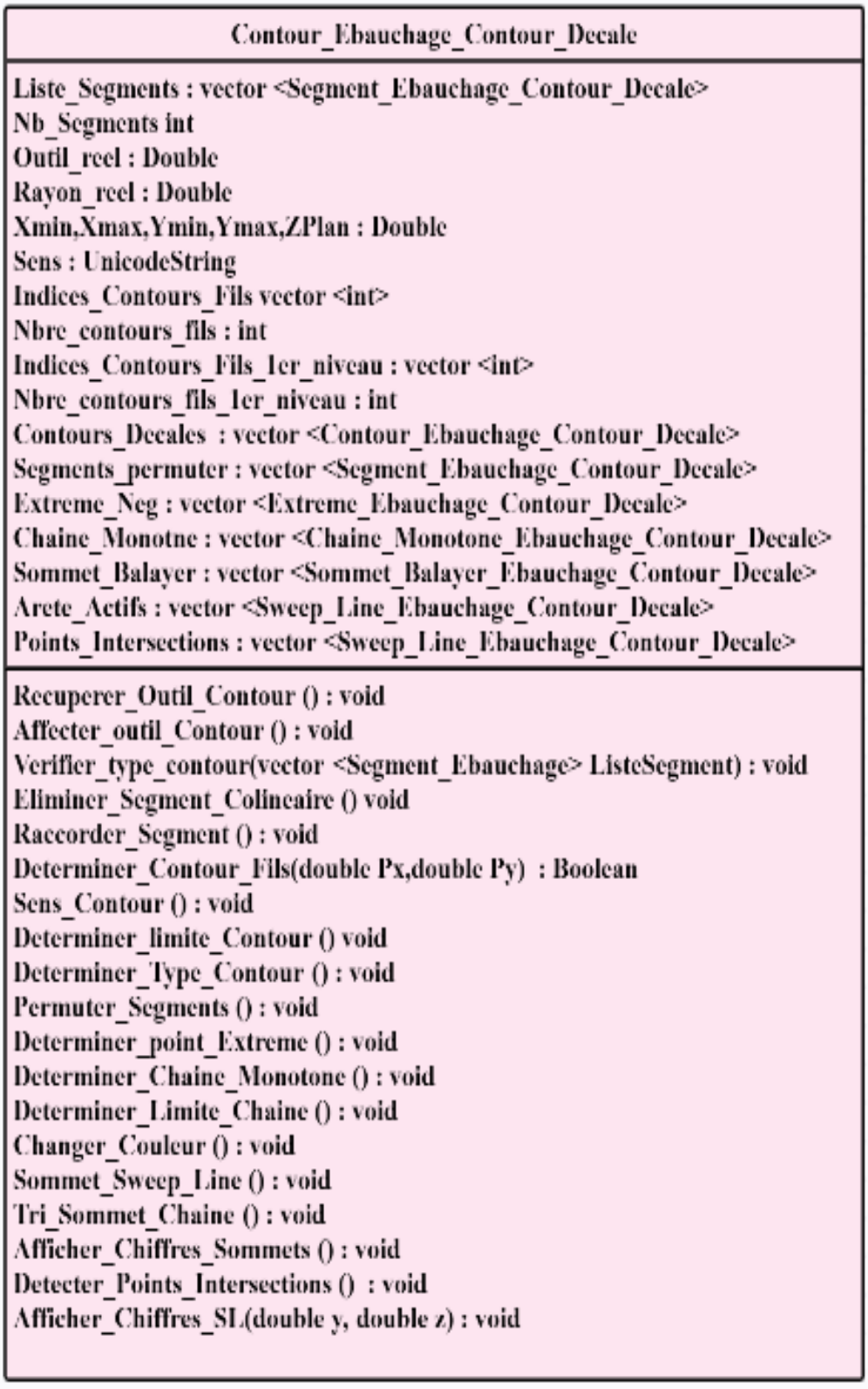


Figure 59: Classe contour.

➢ **Classe « Plan_Ebauchage_Contour_Decale » :** cette classe génère la liste des contours avec les différents attributs importants et les fonctions nécessaires.

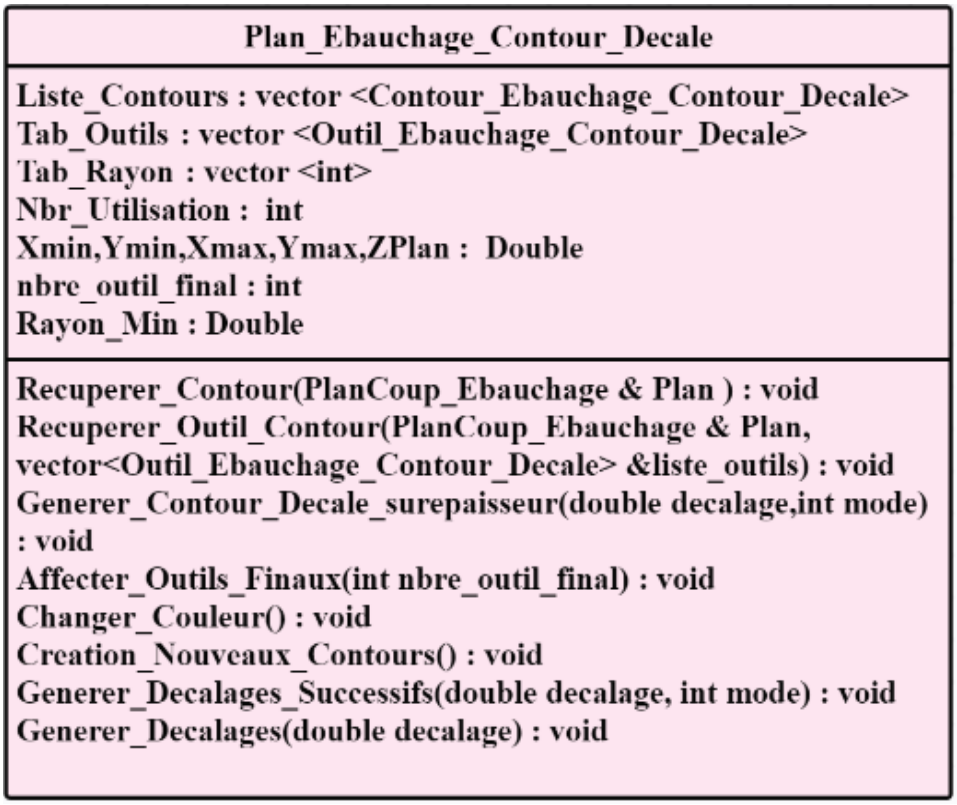


Figure 60: Classe plan.

➢ **Classe « Trajet_Ebauchage_Contour_Decale » :** la classe trajet génère la liste des plans avec les fonctionnalités différentes qu'un trajet aura besoin.

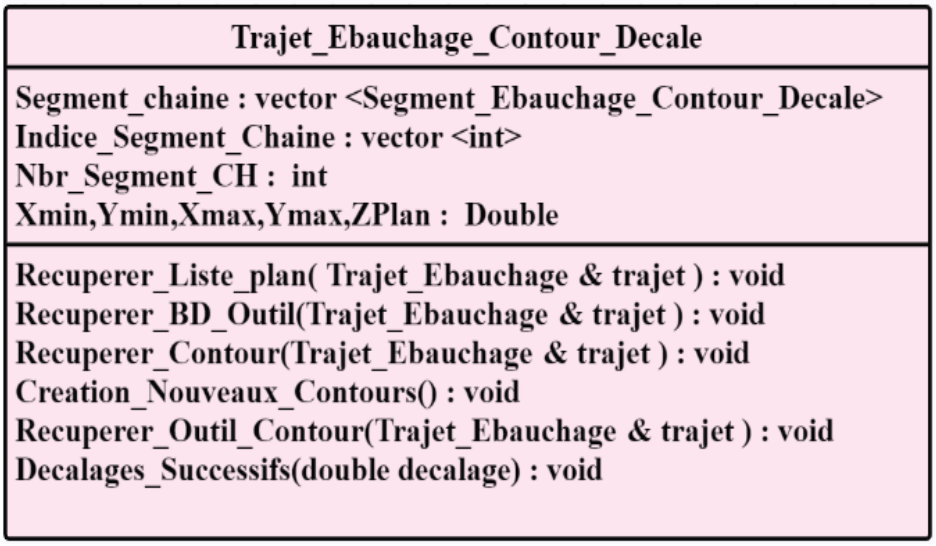


Figure 61: Classe trajet.

➢ **Classe « ModeleSTL_Ebauchage_Contour_Decale » :** c'est une classe qui regroupe toutes les informations contenant un modèle « STL ».

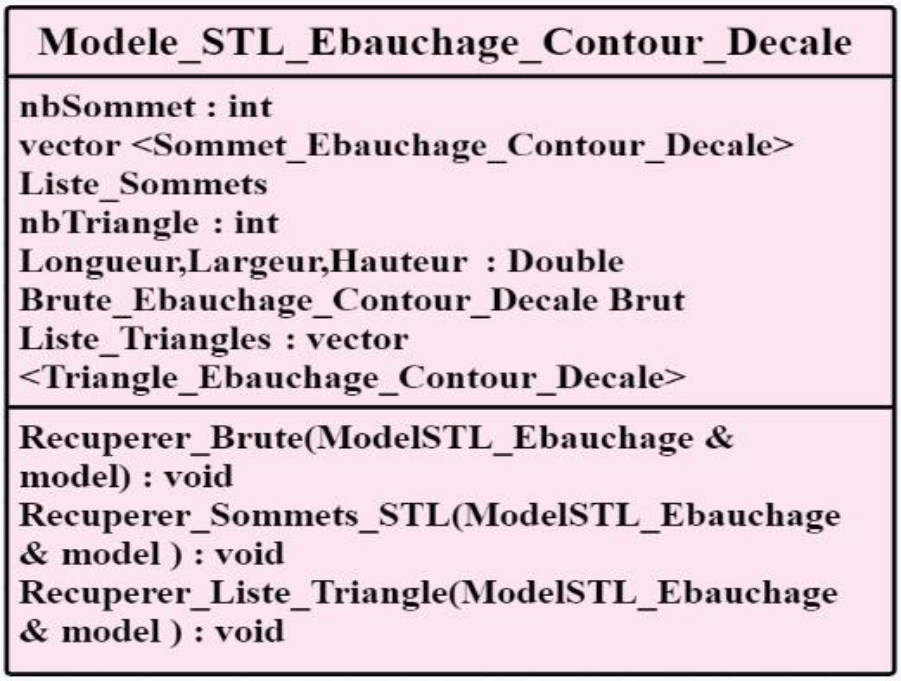


Figure 62 : Classe modèle STL.

II.5. Conclusion

Dans ce chapitre, nous avons présenté la conception de notre application. Cette présentation a été entamée par la définition de la problématique du projet, les différents objectifs visés ainsi que les solutions proposées. Les diagrammes de la conception et la modélisation orientée objet et le langage de modélisation UML ont été utilisés afin de bien comprendre le concept.

Dans le chapitre suivant, nous allons présenter l'implémentation de notre application ainsi que la validation des approches proposées et développées.

CHAPITRE III : Implémentation informatique et validation

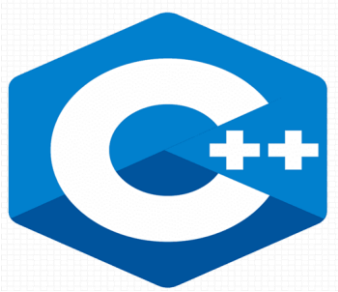
III.1.Introduction

Dans le chapitre précédent, nous avons présenté l'étude conceptuelle de notre système, en mettant en évidence les différents objectifs du module logiciel à développer ainsi que les solutions proposées. Ce chapitre représente la dernière partie du rapport, il est consacré à la présentation des résultats obtenus après l'implémentation à travers des exemples de validation.

III.2.Présentation des langages utilisés

Comme le système développé est réalisé pour être intégré dans la plateforme logicielle de production des pièces complexes développée par l'équipe « CFAO » du « CDTA », les outils de développement utilisés lors de la mise en œuvre (C++, OpenGL) sont les mêmes utilisés par l'équipe « CFAO » pour se conformer à la tendance qui veut que la majorité des systèmes « CFAO » sont développés en utilisant le C++ et la bibliothèque graphique OpenGL.

III.2.1.Présentation du langage C++



Le langage « C » est un langage de programmation inventé par MM. Kernighan et Ritchie au début des années 70. Au début des années 90, Bjarne Stroustrup fait évoluer le langage vers le langage « C++ » en lui rajoutant notamment les notions orientées objet. Toutefois, bien que le « C++ » ait évolué à partir du « C », et ait gardé un grand nombre de notions et de syntaxes de son « ancêtre », il s'agit de deux langages différents. Le « C++ » est un langage compilé : pour écrire un tel programme, il faut commencer par écrire un ou plusieurs fichiers sources. Ensuite, il faut compiler ces fichiers sources grâce à un programme appelé compilateur afin d'obtenir un programme exécutable. Cette phase s'appelle la compilation. Les fichiers sources sont des fichiers textes lisibles dont le nom se termine en général par .c, .cpp ou .h. Les fichiers exécutables portent en général l'extension .exe sous Windows et ne portent pas d'extension sous Linux [28].

III.2.2.Présentation d'OpenGL

Open Graphics Library « OpenGL »est une spécification qui définit une API multiplateformes pour la conception d'applications générant des images 3D. Elle utilise en interne les représentations de la géométrie projective pour éviter toute situation faisant intervenir des infinis. L'interface regroupe environ 250 fonctions différentes qui peuvent être utilisées pour afficher des scènes tridimensionnelles complexes à partir de simples primitives géométriques. Du fait de son ouverture, de sa souplesse d'utilisation et de sa disponibilité sur toutes les plateformes, elle est utilisée par la majorité des applications scientifiques, industrielles ou artistiques 3D. Cette bibliothèque est également utilisée dans l'industrie du jeu vidéo où elle est souvent en rivalité avec la bibliothèque de Microsoft « Direct3D ». Une version nommée « OpenGL ES » a été conçue spécifiquement pour les applications embarquées (téléphones portables, agenda de poche, etc.). On programme très souvent en utilisant un environnement de développement intégré : il s'agit d'un ensemble complet d'outils permettant d'éditer et de modifier des fichiers sources, de les compiler, de lancer l'exécutable, de déboguer le programme, etc... Visual C, C++ Builder, Dev-cpp et Code-Blocks sont des environnements

de développement intégrés [29]. Dans notre projet, le « C++ Builder » est utilisé dans la phase d'implémentation informatique.

III.2.3. Présentation d'Embarcadero Builder 10 Seattle

Embarcadero RAD Studio est un package de développement rapide d'applications qui comprend Delphi, C++ Builder pour la création d'applications natives Windows, Mobile et de base de données. C++Builder est un logiciel de développement rapide d'applications (abrégé RAD pour Rapid Application Développent) conçu par Borland qui reprend les mêmes concepts, la même interface et la même bibliothèque que Delphi en utilisant le langage C++. Il permet de créer rapidement des applications Win32, Win64, MacOS, iOS, Android, ainsi qu'une interface graphique avec son éditeur de ressources [30].

III.3. Matériel utilisé

Le matériel utilisé dans la phase de développement et de validation de l'application se résume en un PC portable sous Windows 10 avec les caractéristiques suivantes :

- Processeur : Intel® Core™ i5-3320M CPU @ 2.60GHz 2.60 GHz.
- Mémoire installée (RAM) : 4.00 Go (3.87 Go utilisable).
- Type du système : Système d'exploitation 64 bits, processeur x64.

III.4. Présentation de la plateforme logicielle

Une plateforme logicielle graphique et interactive sous Windows dédiée à la production numérique des pièces complexes sur fraiseuses numériques est développée par l'équipe « CFAO » du « CDTA ». Cette plateforme est composée en deux parties (Figure 1).

III.4.1.Fenêtre principale de la plateforme CFAO

III.4.1.1. Fenêtre de visualisation

C'est une fenêtre réservée à la visualisation des objets géométrique en 3D (points, surfaces, outils, machine, simulation, trajet d'outils, etc.) en utilisant la bibliothèque graphique « OpenGL ». Dans cette partie, les objets visualisés peuvent subir diverses transformations géométriques telles que translations, rotations, projections, zoom et changement d'échelles.

III.4.1.2. Fenêtre menu principal

Cette fenêtre est composée de trois rubriques :

- **Rubrique fichier :** englobe toutes les fonctionnalités de manipulation de fichiers (ouverture, création, sauvegarde, etc.).
- **Rubrique options :** permet la modification des paramètres des courbes, des surfaces et des trajets d'outils.
- **Rubrique aide :** permet l'affichage des informations de la réalisation et l'utilisation de l'application.

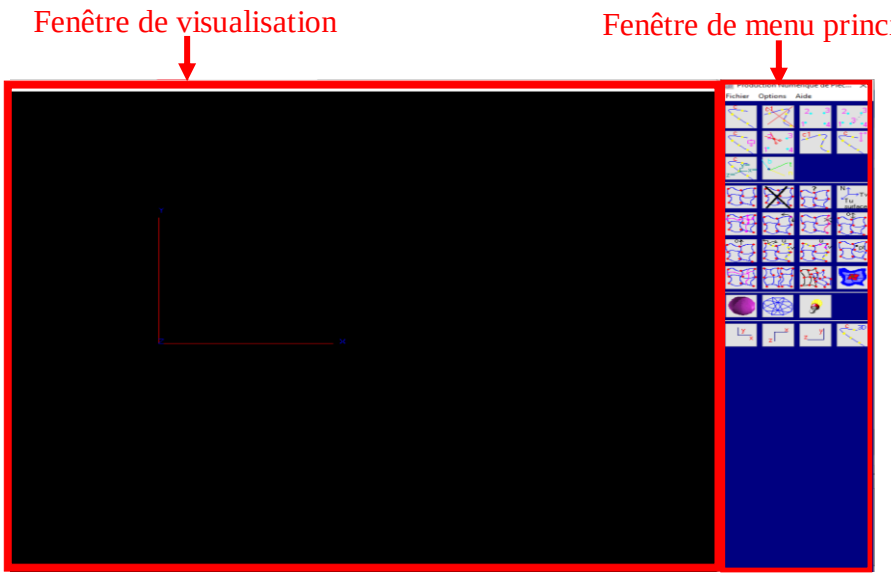


Figure 1: Fenêtres principales de la plateforme logicielle.

III.4.2. Rubrique « Ebauchage en 03-axes par « Contours Décalés » »

Le module logiciel que nous avons développé dans le cadre de ce projet, est intégré à cette plateforme sous le nom **« Ebauchage en 03-axes par «Contours Décalés»** avec un raccourci **«Ctrl+A»** qui est lancé à partir de l'onglet **« Option »**, **« Projet Socio-Economique (Production de Pièces Complexes en 03-Axes) »** et **« Ebauchage en 03-axes par « Contours Décalés »** (Figure 2).

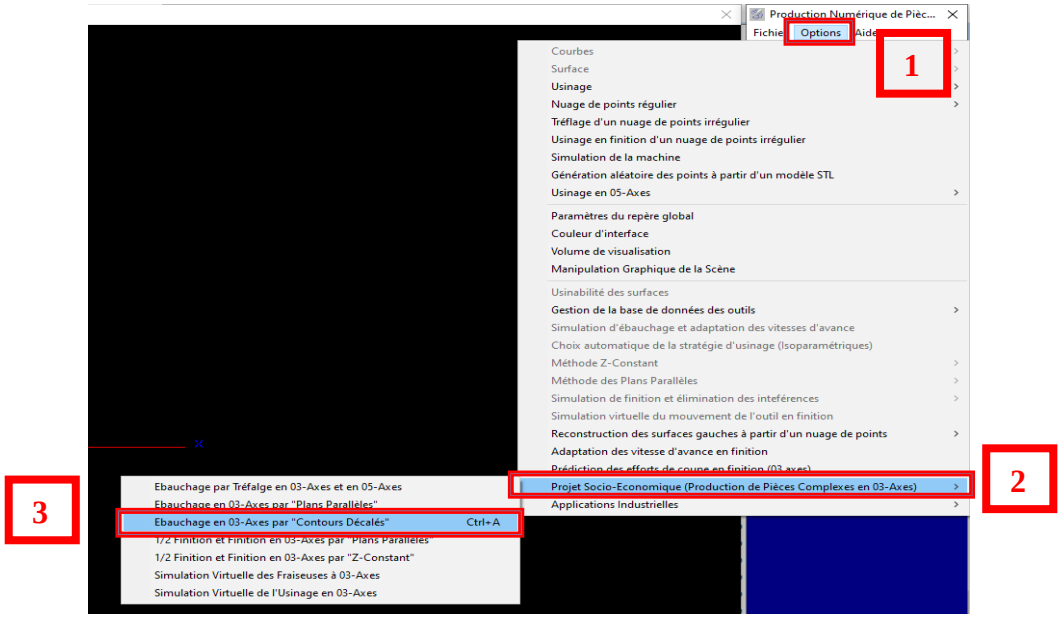


Figure 2: Lancement du module logiciel développé.

III.5. Présentation du module logiciel développé

Afin d'atteindre l'objectif principal, l'application développée est composé de cinq onglets, où chaque onglet est dédié à des phases du processus de l'opération d'ébauchage par la stratégie « Contours Décalés » en 03-axes (Figure 3). Les différents onglets sont :

1. Onglet 1 : Lecture du modèle STL.
2. Onglet 2 : Génération des contours.
3. Onglet 3 : Application Sweep-Line.
4. Onglet 4 : Génération des contours décalés.

Figure 3 : Onglets du module logiciel développé.

III.5.1. Lecture du fichier STL

Le premier onglet **« Lecture du fichier STL »** permet la lecture du fichier texte contenant tous les paramètres des triangles et leurs sommets récupérés. Il est lancé par un simple clic sur le bouton **« Récupérer les paramètres du modèles STL »** suivi par la sélection du fichier à ouvrir. Par la suite, les limites du brut sont calculées et affichées. Une fois le fichier est lu, les dimensions du brut (largeur, longueur et hauteur) peuvent être récupérées ainsi que le nombre de sommets et nombre de triangles seront affichés.

Donc, il est possible de visualiser les informations suivantes (Figure 4) :

- Les limites de la pièce brute : (Xmin, Xmax, Ymin, Ymax, Zmin et Zmax), la longueur, la largeur et la hauteur.
- Le nombre de sommets et le nombre de triangles.
- Le brut du modèle en deux modes (filaire et rendu).
- Les sommets du brut.
- Les sommets du modèle.
- Les triangles du modèle en deux modes (filaires et rendus).

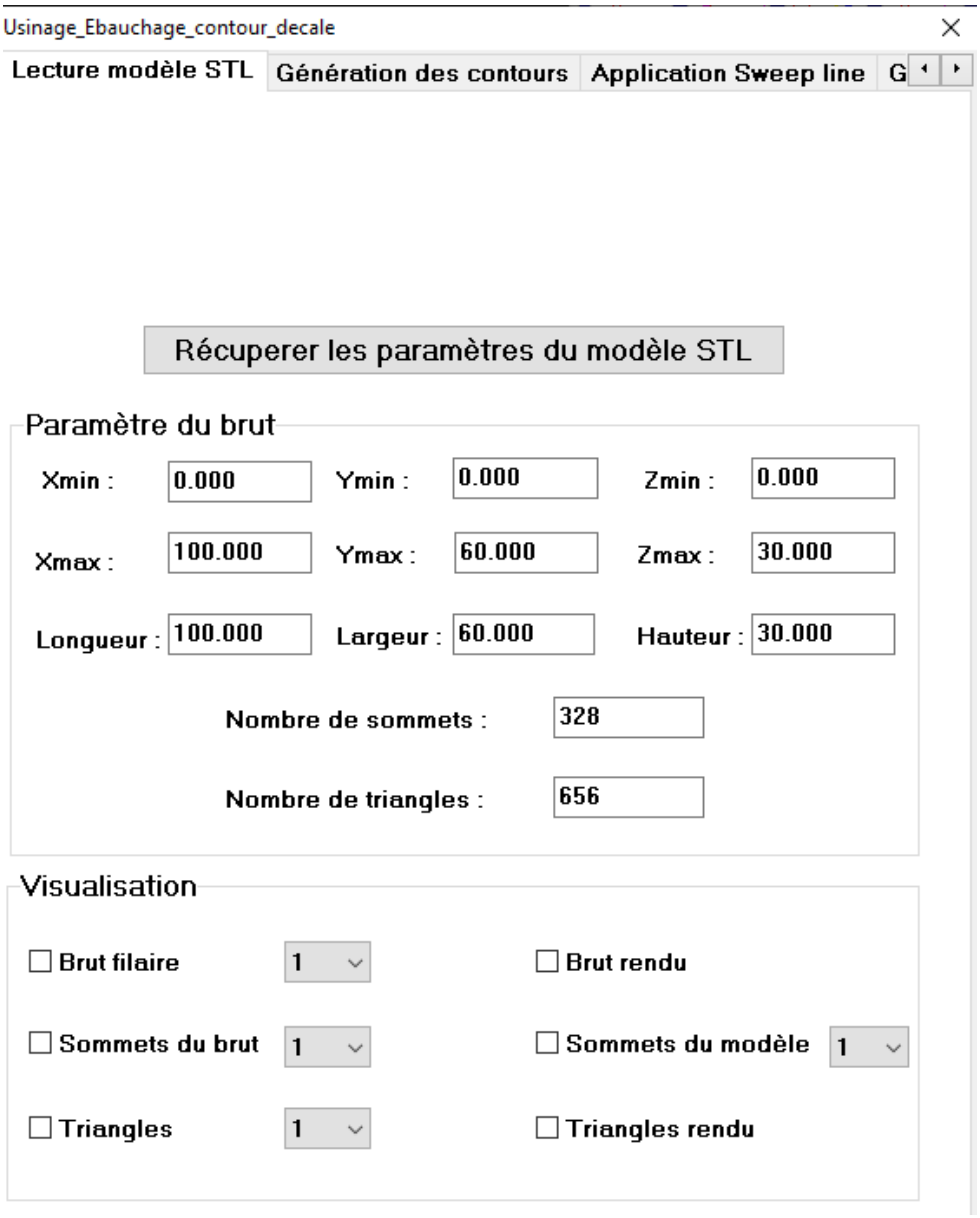


Figure 4: Onglet « Lecture du fichier STL ».

III.5.2. Génération des contours

Après la première étape qui est la lecture du fichier STL, elle vient cette étape qui consiste à récupérer les entités géométriques d'un modèle (sommets, segments, contours et plans) dans l'onglet **« Génération des contours »** (Figure 5). Les visualisations possibles sont **:**

- **Visualisation globale** qui consiste à :
  - Visualiser l'ensemble des contours initiaux de tous les plans.
- **Visualisation partielle** qui consiste à visualiser :
  - Contours initiaux de chaque plan avec leurs normales.
  - Numérotation des segments et limites des segments et des contours.
  - Sens des contours initiaux.

- Contours fils et contours fils du premier niveau.
- Outils optimums des contours et sélection des outils finaux.
- Contours décalés et leurs sens.

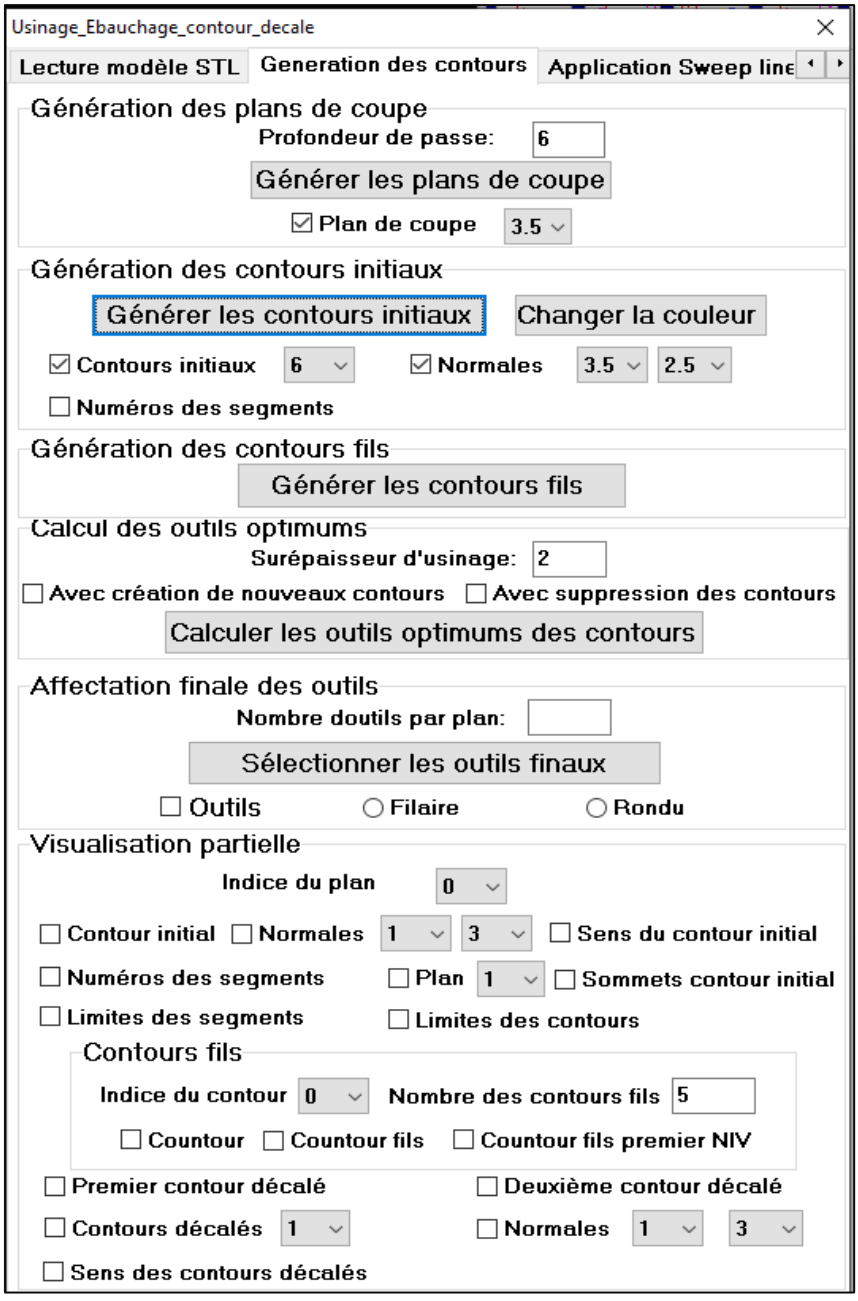


Figure 5 : Onglet « Génération des contours ».

Dans cet onglet, l'utilisateur doit passer par plusieurs étapes :

1. Visualisation globale :

Etape 1 : saisir la profondeur de passe dans un champ de texte et ensuite cliquer sur le bouton **« Générer les plans de coupe »** (Figure 6). A ce niveau, l'utilisateur peut visualiser les informations suivantes :

- Un clic sur le check-box **« plan de coupe »**, un ou plusieurs plans seront affichés (selon la forme géométrique de la pièce).
- Un combo-box de surépaisseur a été réalisé pour bien préciser les plans sur le dessin.

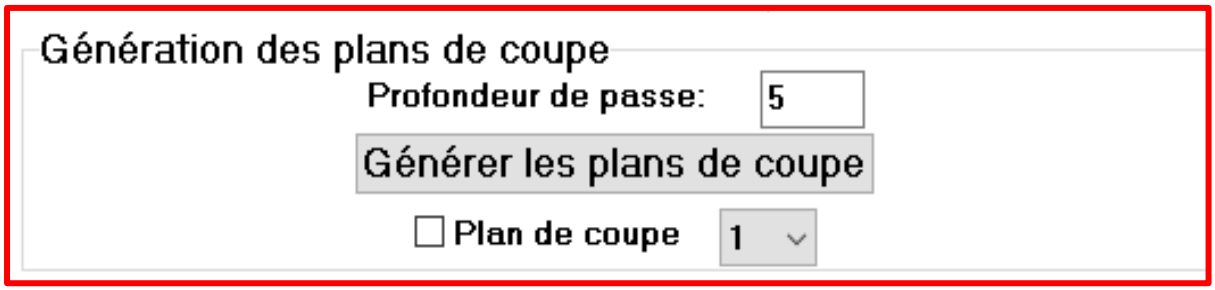


Figure 6 : Génération des plans de coupe.

Etape 2 : après la récupération des plans, l'utilisateur doit générer les contours initiaux. Cela est réalisé par un clic sur le bouton **« Générer les contours initiaux»**. Par la suite, il est donc possible de visualiser les entités géométriques (Figure 7).

Nous avons utilisé un check-box **« contours initiaux »** pour afficher tous les contours et un combo-box de surépaisseur des contours initiaux à sélectionner par l'utilisateur. Nous avons utilisé un check-box **« normales »** qui affiche toutes les normales et deux combo-box de surépaisseur à sélectionner par l'utilisateur (1. Longueur, 2. Largeur). Le bouton **« Changer la couleur »** permet à l'utilisateur de changer les couleurs des contours initiaux. Les segments des contours sont numérotés par un simple clic sur le check-box **« numéros des segments »**.

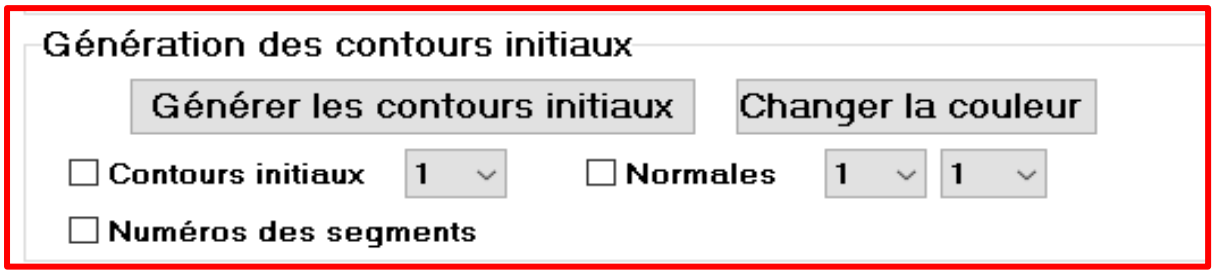


Figure 7: Visualisation globale.

2. Visualisation partielle

Etape 1 : le choix d'un seul plan est réalisé par un combo-box qui contient l'indice de chaque plan. Le **bouton « changer la couleur du plan »** a été utilisé pour changer la couleur de plan.

Nous avons utilisé un check-box **« normales »** qui affiche toutes les normales des segments des contours dans un seul plan donné et deux combo-box de surépaisseur à sélectionner par l'utilisateur (1. Longueur, 2. Largeur).

Un check-box est dédié à la détermination du sens des contours pouvant être **« Horaire »** ou **« Antihoraire »** :

Les limites des contours et des segments ont été calculés. Pour les afficher, il suffit de cliquer sur un check-box pour dessiner les limites des contours et un check-box pour dessiner les limites des segments des contours (Figure 8).

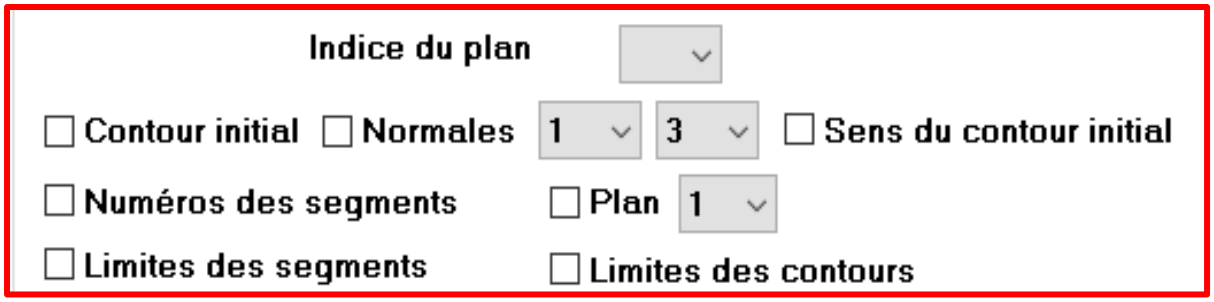


Figure 8 : Visualisation partielle.

Etape 2 : cette partie de l'onglet est réalisée pour l'arborescence des contours dans un plan donné. Le bouton **« Générer les contours fils »** permet de déterminer tous les contours fils pour chaque contour ainsi que ses contours fils du premier niveau (Figure 9). Une fois la détermination est effectuée, il est possible de :

- Visualiser le contour initial en cliquant sur le check-box **« contour ».**
- Visualiser les contours fils en cliquant sur le check-box **« contour fils »**.
- Visualiser les contours fils du premier niveau en cliquant sur le check-box **« contour fils du premier niveau ».**
- Pour confirmer notre travail, nous avons utilisé un check-box des indices du contour, une fois l'indice est sélectionné, le nombre de ses fils sera affiché sur un label.

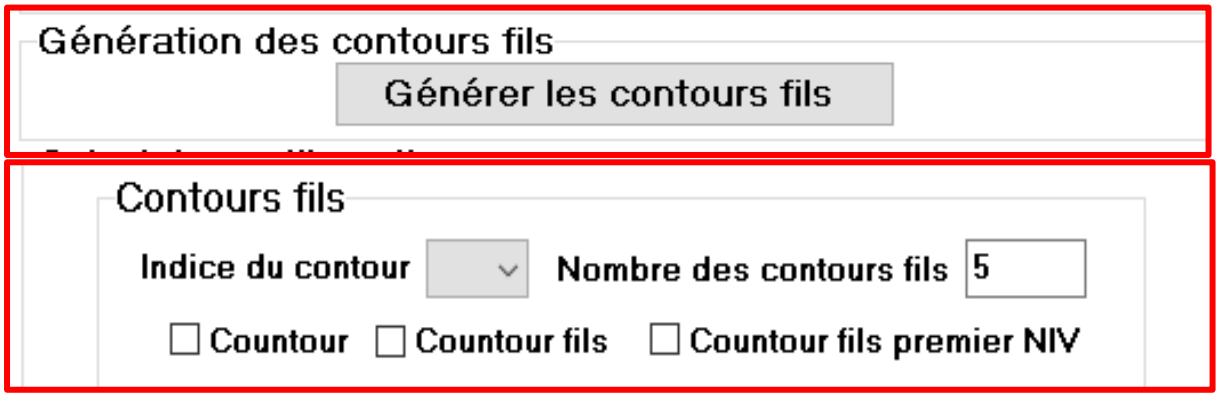


Figure 9 : Génération des contours fils.

Etape 3 : la récupération de la base de données des outils a été utilisée dans cette partie. Nous avons calculé les outils optimums en fonction de leurs rayons dans le but de générer les contours décalés et pouvoir observer les problèmes des boucles invalides. Un simple clic sur un bouton permet à l'utilisateur de générer les contours décalés en fonction du rayon de l'outil (Figure 10). Cela permet à l'utilisateur de :

- Calculer les outils optimums en cliquant sur **« Calculer les outils optimums »**.
- Visualiser les contours décalés avec la création des nouveaux contours.

- Visualiser les contours décalés après la suppression des boucles invalides.
- Visualiser les contours décalés par un clic sur le check-box « **Contours Décalés »** et leurs normales par un clic sur le check-box « **Normales ».**
- Visualiser le sens des contours décalés en cliquant sur le check-box « **Sens des contours décalés »** (Figure 11).

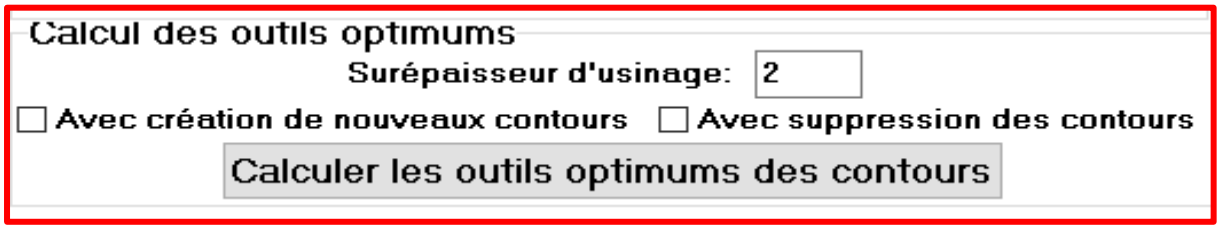


Figure 10: Calcul des outils optimums.

Figure 11 : Génération des contours décalés.

Etape 4 : cette partie est réservée pour la détermination des outils. L'utilisateur peut choisir le nombre d'outils souhaité qui est récupéré à partir de la base de données d'outils optimums. Ensuite, un simple clic sur le bouton **« Sélectionner le nombre d'outils »**, les outils optimums seront affectés pour chaque contour selon le nombre d'utilisation et leurs rayons (Figure 12). Une fois la détermination est faite, l'utilisateur peut visualiser les outils sur les contours en deux modes (filaire et rendu).

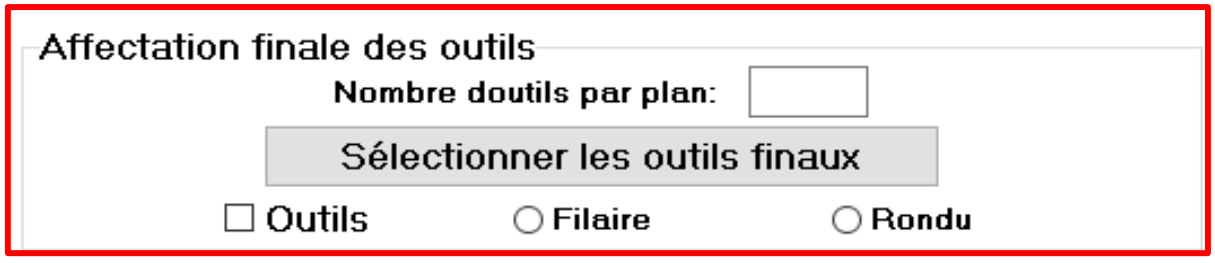


Figure 12 : Sélection des outils finaux.

III.5.3. Application Sweep-Line

Dans cet onglet l'utilisateur peut choisir jusqu'à cinq (05) contours pour pouvoir tester et observer les étapes de notre approche. Il est divisé en trois parties (Figure 13).

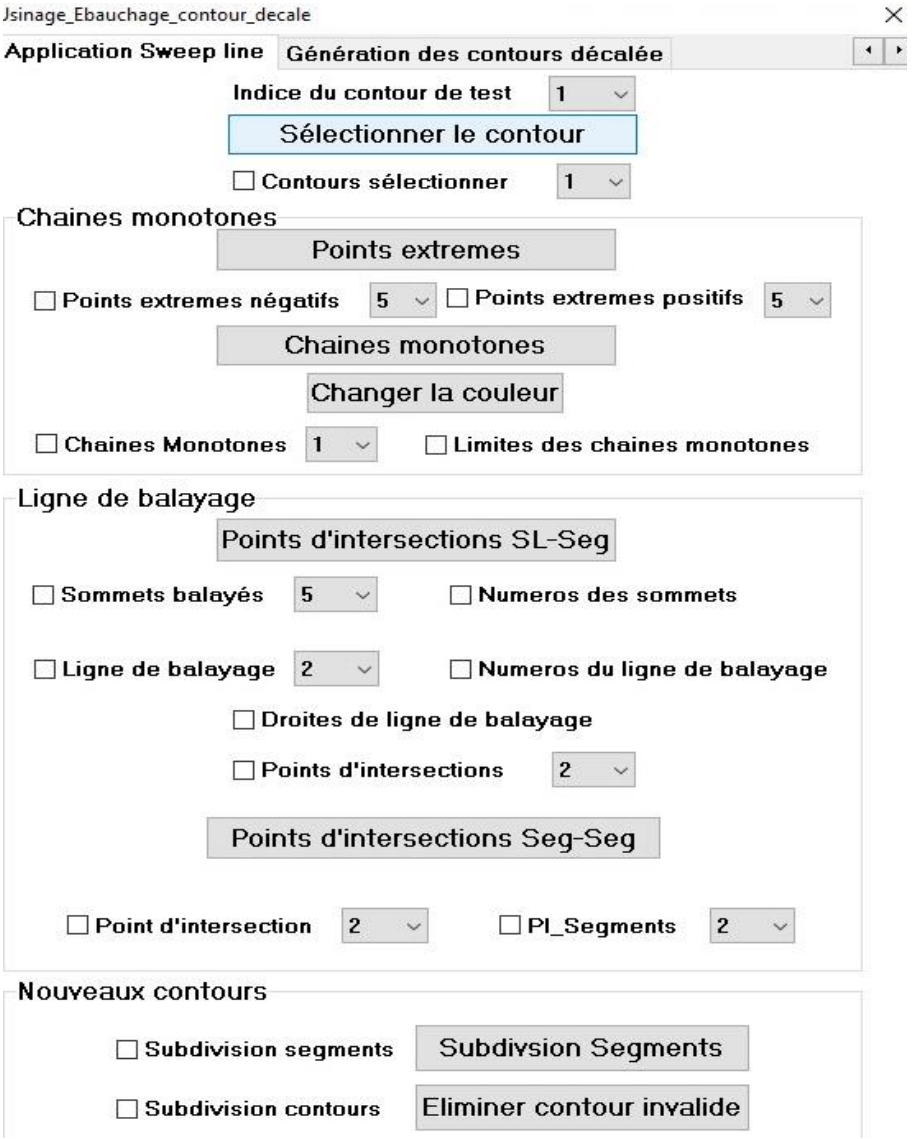


Figure 13 : Onglet « Sweep-Line Algorithme ».

1. **Création des chaines monotones :** pour un contour donné, l'utilisateur peut (Figure 14) :
   - Déterminer les extrémités : pour ce faire un clic sur le bouton « **Points extrêmes** ». Par la suite, il peut :
     - Afficher les points extrêmes négatifs.
     - Afficher les points extrêmes positifs.
     - Changer la surépaisseur du point sur le dessin.
   - Déterminer les chaines monotones : la réalisation de cette fonctionnalité est un clic sur le bouton « **Déterminer les chaines monotones** ». Il est suivi par :
     - Un check-box d'affichage des chaines monotones.
     - Un bouton pour changer la couleur des chaines monotones.
     - Un check-box pour déterminer les limites des chaines monotones.

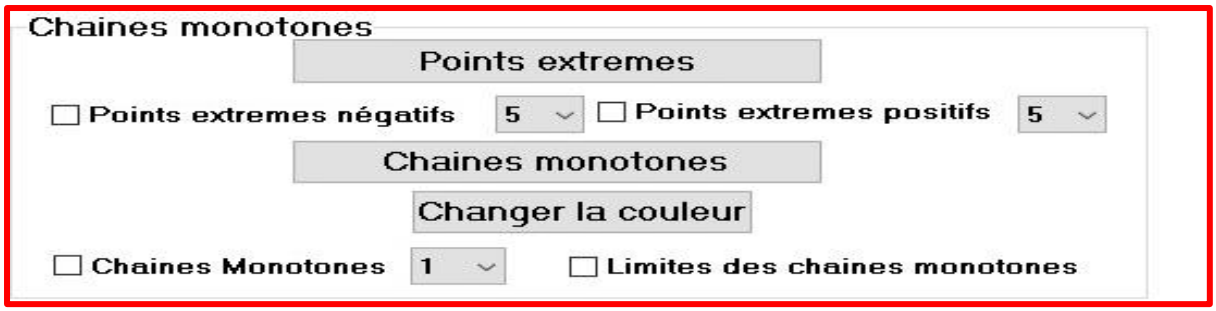


Figure 14 : Création des chaines monotones.

2. **Ligne de balayage :** dans cette partie, l'utilisateur peut appliquer l'algorithme de **« Sweep-Line »** « **ligne de balayage** » étape par étape pour qu'il puisse regarder en réalité comment cet algorithme fonctionne (Figure 15).

   En cliquant sur le bouton « **Sweep-Line** », ces informations seront récupérées :

   - Les sommets balayés : c'est des sommets triés suivant l'axe X que la ligne de balayage balaye en affichant leurs numéros par un check-box.
   - Les lignes de balayage : ce sont des lignes triées suivant l'axe X qui passent par tous les sommets du contour de gauche à droite en affichant leurs numéros par un check-box.
   - Les droites de la ligne active : c'est-à-dire les segments appartenant à chaque ligne de balayage.
   - Les points d'intersection : ils se divisent en deux types :
     - Points d'intersection entre la ligne de balayage et les segments.
     - Points d'intersection entre les segments.

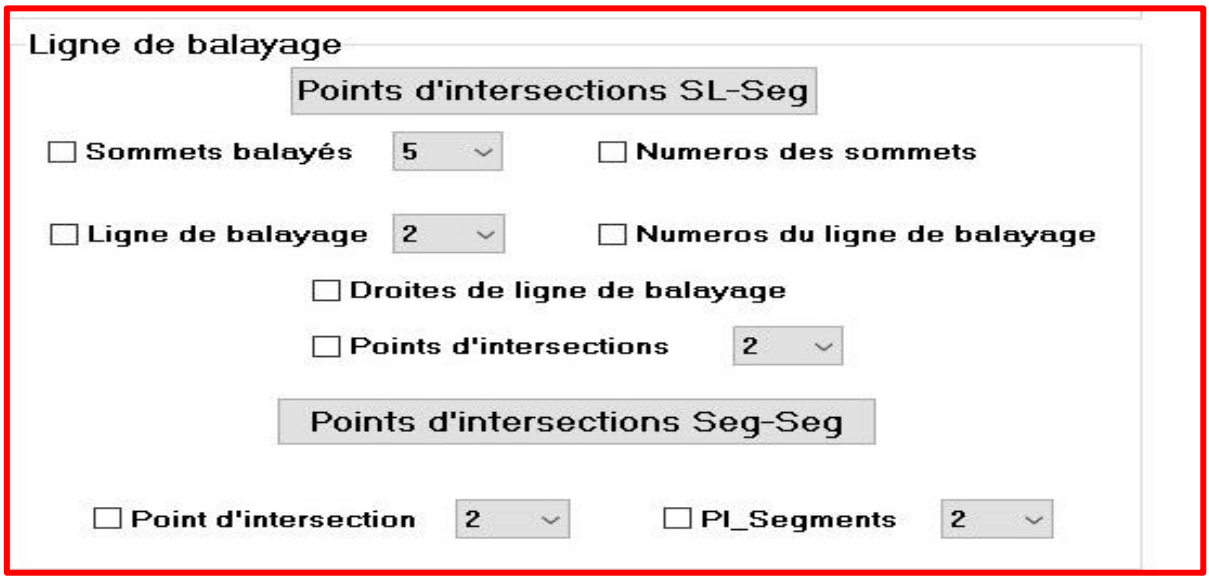


Figure 15 : Ligne de balayage.

3. **Nouveaux contours :** dans cette dernière partie de l'onglet, nous avons défini (Figure 16) :
   - La subdivision des segments : elle consiste à subdiviser les segments depuis les points d'intersection entre les segments par le bouton « **Subdiviser les segments ».**
   - La subdivision des contours : après la subdivision des segments, les nouveaux contours sont créés. Par un clic sur le bouton « **Subdiviser les contours »**, les contours à sens inverse avec le contour père sont éliminés.

Figure 16 : Nouveaux contours.

III.5.4. Décalages Successifs

Cet onglet montre la génération successive des contours décalés en fonction du nombre de décalage et de la distance choisie (Figure 17).

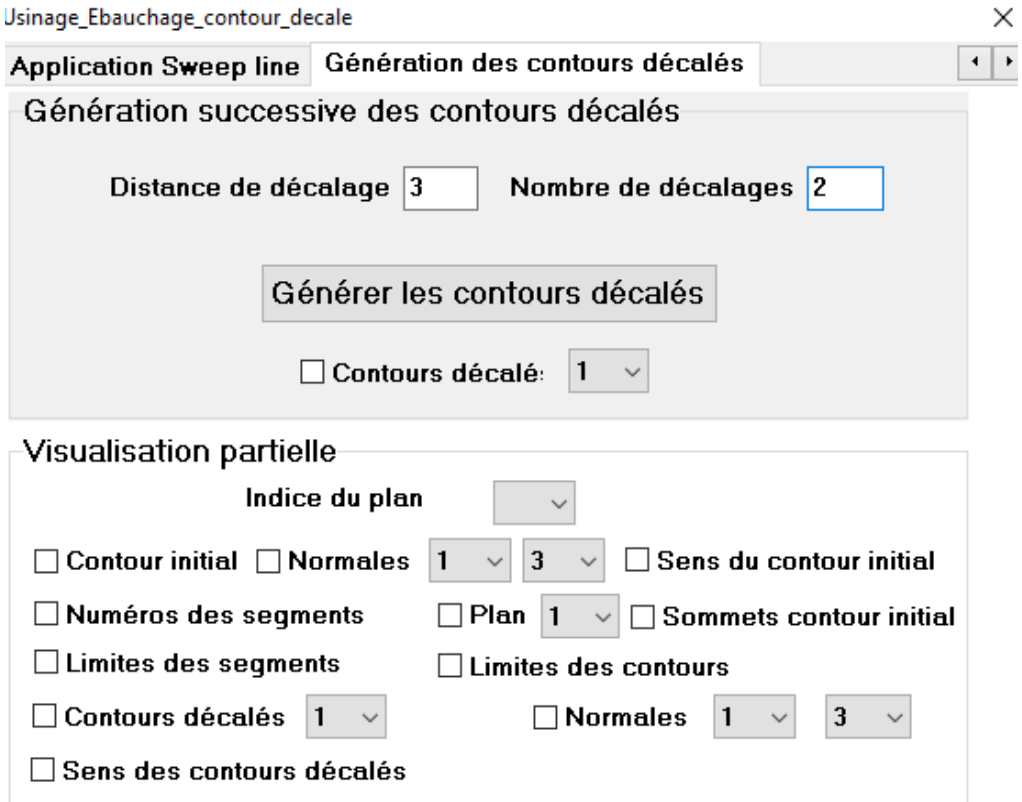


Figure 17 : Onglet « Génération des contours décalés ».

Pour la génération successive des contours décalés, nous passons par plusieurs étapes :

Etape 1 : dans cette étape, la visualisation des contours est globale, choisir le nombre de décalages est nécessaire ainsi que la distance entre les contours décalés, l'utilisateur doit choisir ces deux nombres en cliquant sur le bouton « **Générer les contours décalés »** (Figure 18).

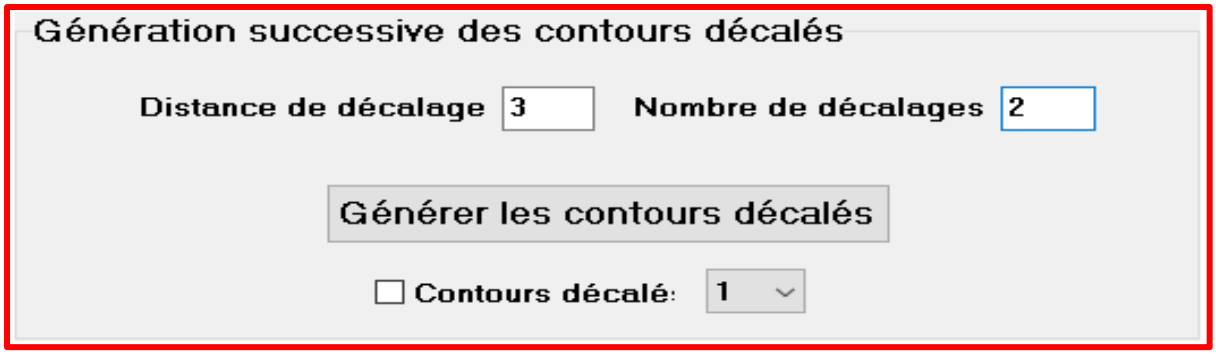


Figure 18 : « Génération des contours décalés ».

Etape 2 : cette partie est réservée à la visualisation partielle. L'utilisateur saisi le nombre de décalages ainsi que la distance de décalage. Il peut visualiser par la suite le contour initial d'un plan quelconque, ses normales, ses sommets, ses limites et les contours décalés et leurs sens (Figure 19).

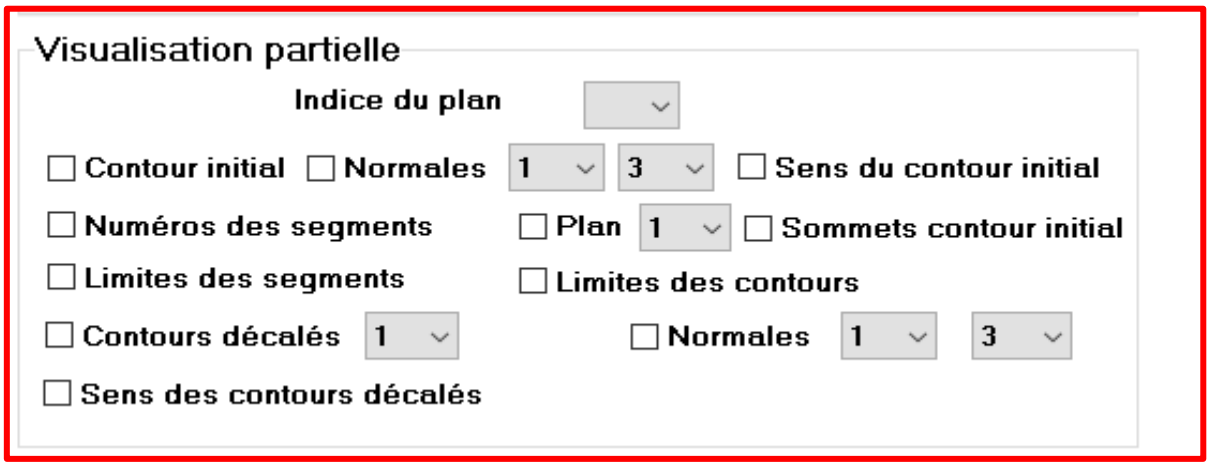


Figure 19 : Visualisation partielle des contours décalés.

III.6. Test et validation

Arrivée au terme de la présentation de l'application, la phase de tests et de validations de notre travail et les différents résultats auxquelles nous avons abouti ont été faite sur un PC ayant les caractéristiques suivantes :

- Windows 10.
- Processeur Intel(R) Core (TM) i9-10900K CPU @ 3,70GHz.
- Mémoire installée (RAM) : 32.00 Go.
- Type de système : système d'exploitation 64 bits.

Le test de validation s'est fait sur six (06) modèles STL de pièces issues d'une conception générée dans un logiciel de « CAO » et cinq (05) autres modèles crées par nous-même.

Parmi les six (06) modèles STL, seuls deux exemples sont explicités dans la suite de ce travail et un modèle créé est utilisé pour montrer les étapes des approches proposées lors de la phase de conception.

III.6.1. Premier modèle STL

L'exemple considéré est représenté par la Figure 20. Les résultats des différentes étapes présentés dans les paragraphes suivants sont relatifs à ce modèle STL.

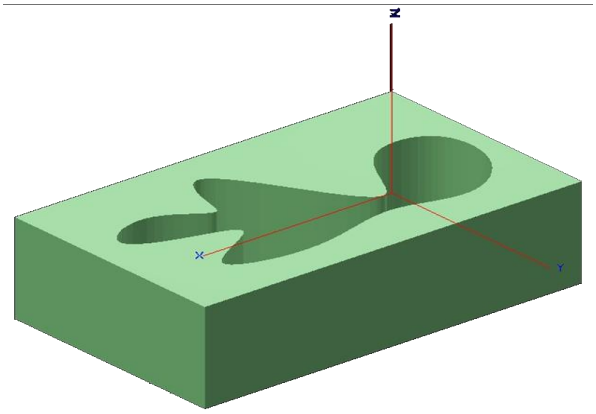


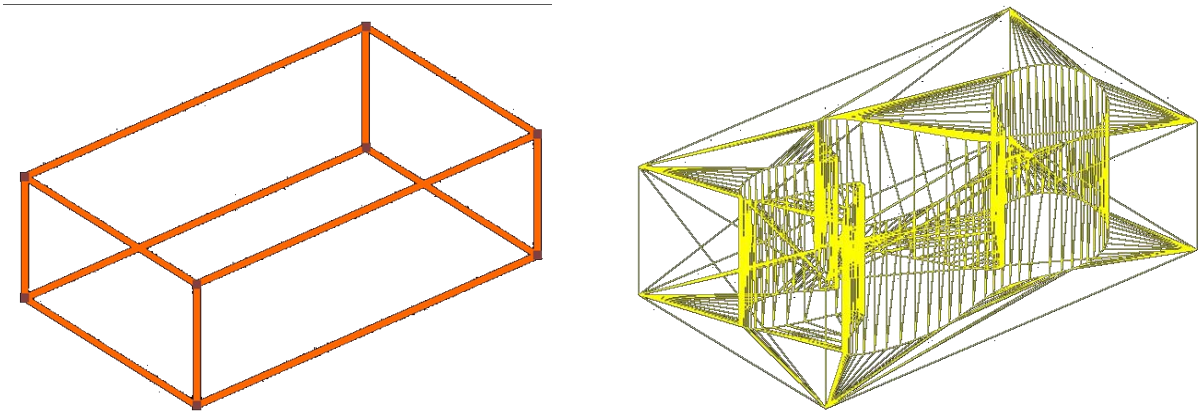
Figure 20: Brute rendu.

Etape 1 : lecture du fichier STL et calcul des limites du brut. Ces opérations donnent les résultats suivants (Tableau 1) :

Tableau 1 : Paramètres du premier modèle STL.

Xmin : 0.000 mm	**Ymin :** 0.000 mm	**Zmin :** 0.000 mm
Xmax : 100.000 mm	**Ymax :** 60.000 mm	**Zmax :** 30.000 mm
Longueur : 100.000 mm	**Largeur :** 60.000 mm	**Hauteur :** 30.000 mm
Nombre de sommets : 328	**Nombre de triangles :** 656	

D'autres paramètres de visualisation sont donnés par la Figure 21.



a. Brut en filaire. b. Triangles du modèle STL.

Figure 21 : Brut et triangles du modèle STL.

Etape 2 : génération des plans de coupe pour une profondeur de passe égale à 6 mm (Figure 22).

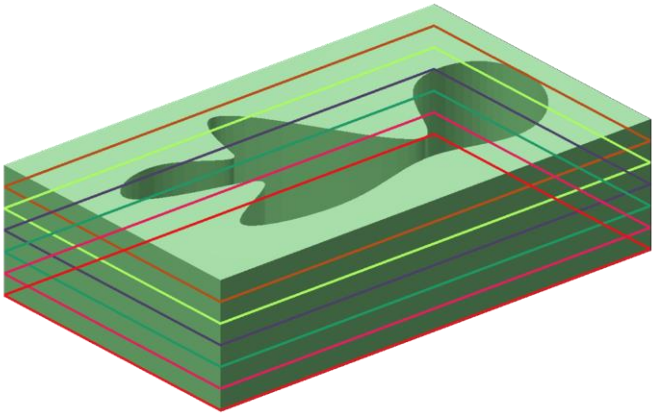


Figure 22: Plans de coupe.

Etape 3 : génération des contours initiaux de tous les plans (Figure 23).

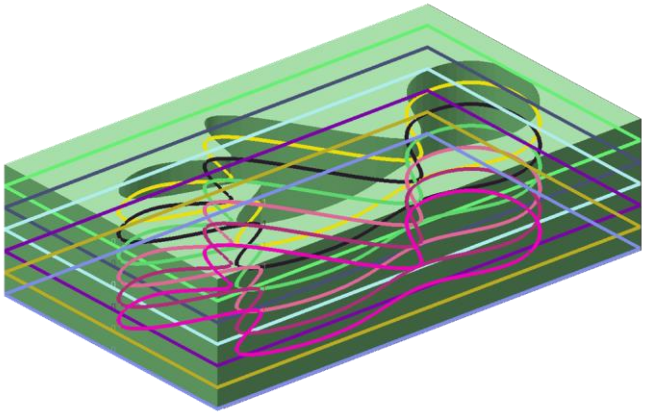


Figure 23 : Contours initiaux.

Etape 4 : génération des contours initiaux pour la visualisation partielle. Dans cet exemple, nous avons pris le plan numéro deux (02) (Figure 24)**.**

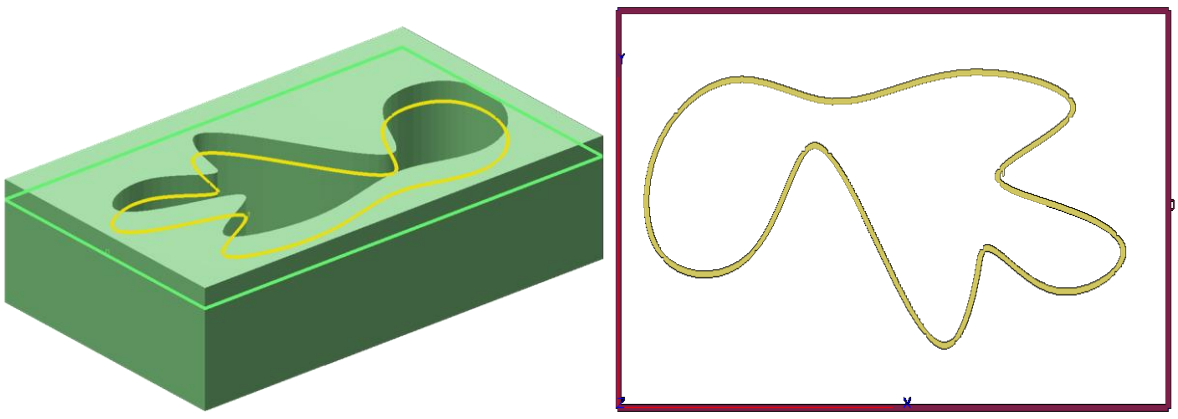


Figure 24 : Contour initial d'un plan de coupe.

L'utilisateur peut visualiser les normales et les numéros des segments (Figure 25).

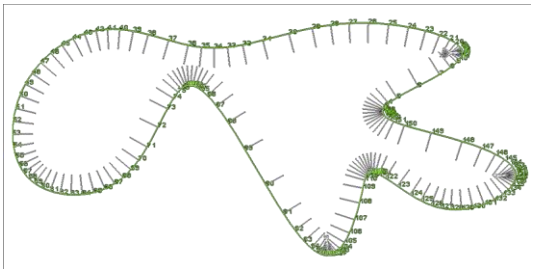


Figure 25 : Normales et numéros des segments.

La Figure 26 montre le sens des contours. La couleur jaune pour le sens « Horaire » et la couleur rouge pour le sens « Antihoraire ».

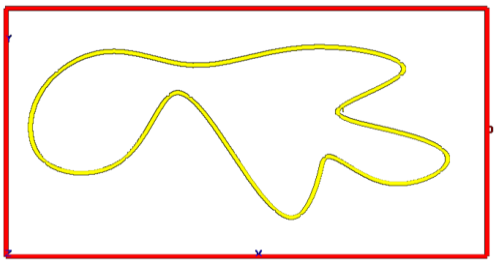


Figure 26 : Sens des contours initiaux.

Les limites des contours et les limites des segments sont montrées par la Figure 27.

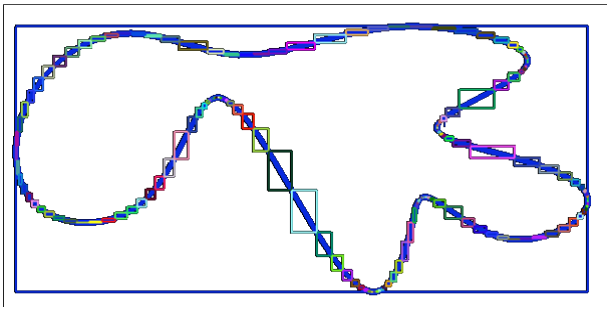


Figure 27 : Limites des contours et des segments.

Etape 5 : les outils optimums des contours sont calculés pour une surépaisseur d'usinage égale à 2mm. Les contours décalés avec leurs normales sont représentés dans la Figure 28.

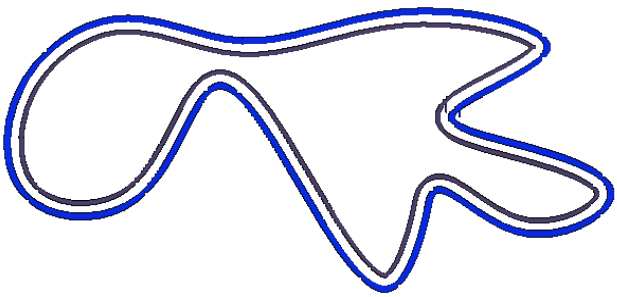
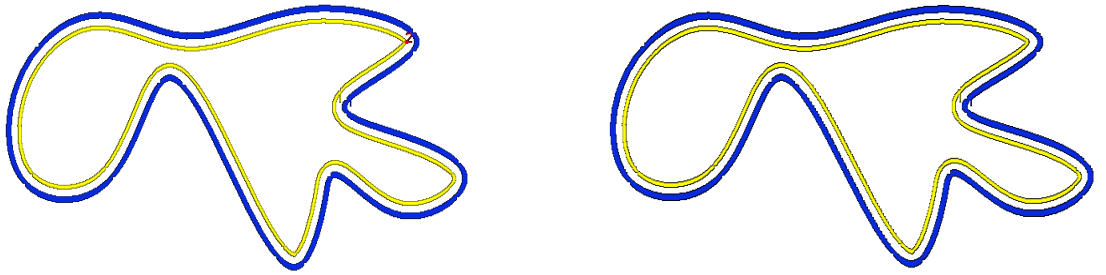


Figure 28 : Contour décalé.

Dans la Figure 29, le sens des contours décalés est précisé avec les couleurs jaune et rouge. Nous allons tester le décalage avec différentes surépaisseurs et montrer les contours décalés avant et après l'élimination des boucles invalides.

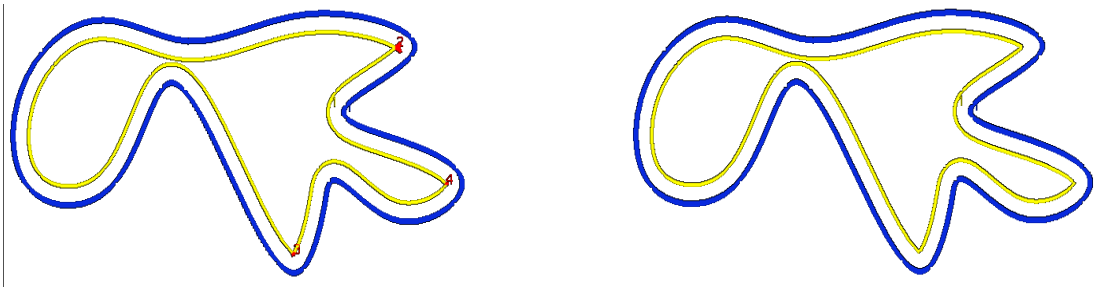
- **Surépaisseur = 2mm**



a. Avant la suppression des boucles. b. Après la suppression des boucles.

Figure 29 : Contours décalés.

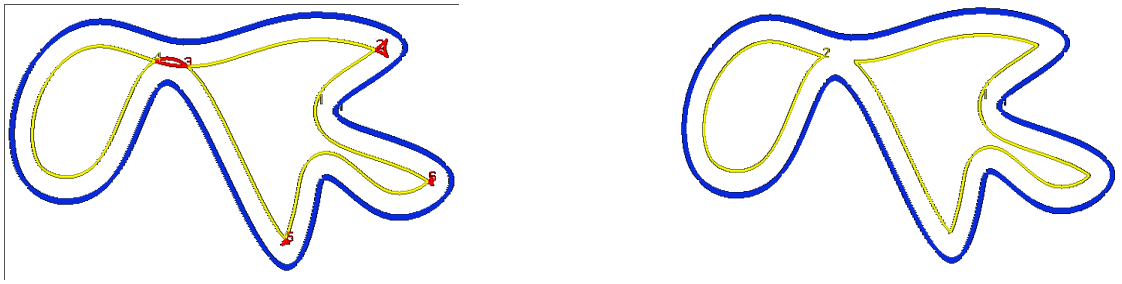
- **Surépaisseur = 3mm**



a. Avant la suppression des boucles. b. Après la suppression des boucles.

Figure 30 : Contours décalés.

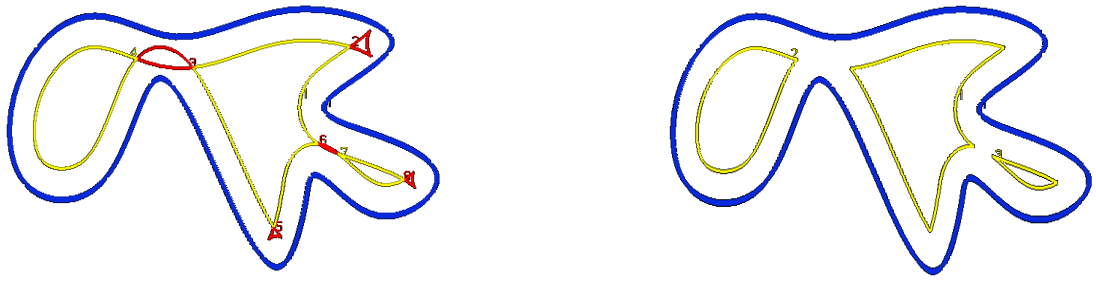
- **Surépaisseur = 4mm**



a. Avant la suppression des boucles. b. Après la suppression des boucles.

Figure 31 : Contours décalés.

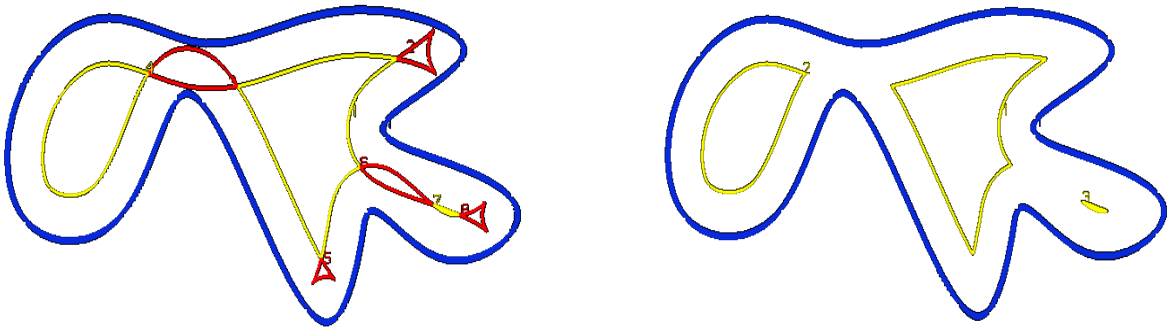
- **Surépaisseur = 5mm**



a. Avant la suppression des boucles. b. Après la suppression des boucles.

Figure 32 : Contours décalés.

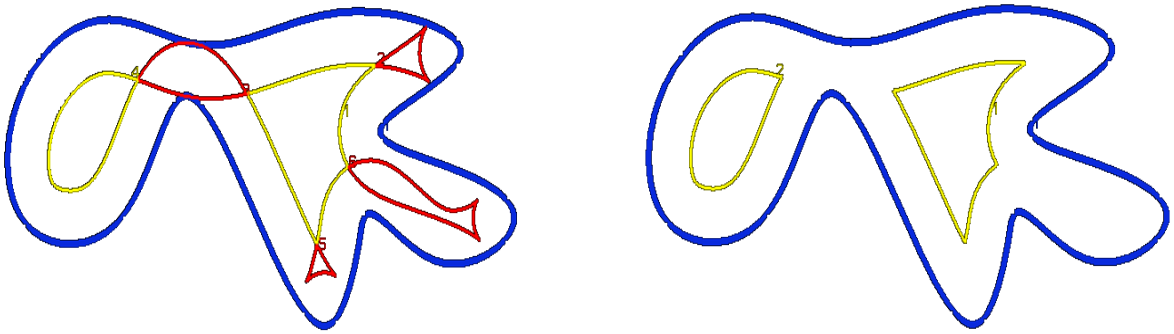
- **Surépaisseur = 6mm**



a. Avant la suppression des boucles. b. Après la suppression des boucles.

Figure 33 : Contours décalés.

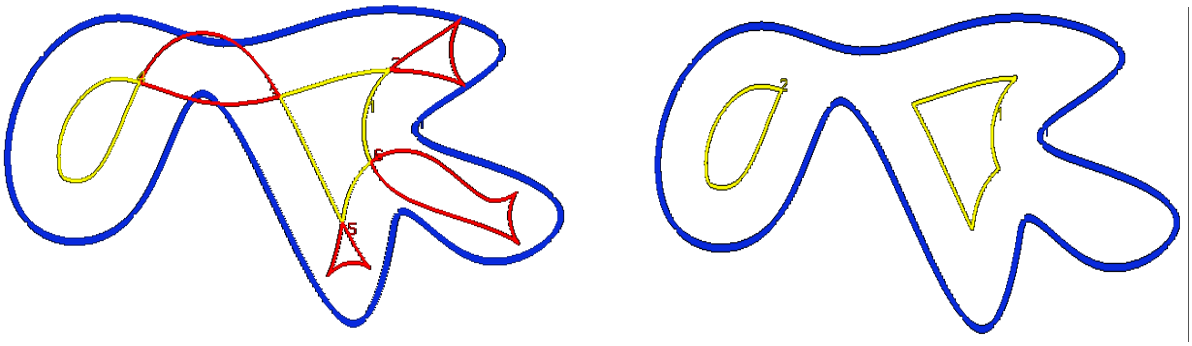
- **Surépaisseur = 7mm**



a. Avant la suppression des boucles. b. Après la suppression des boucles.

Figure 34 : Contours décalés.

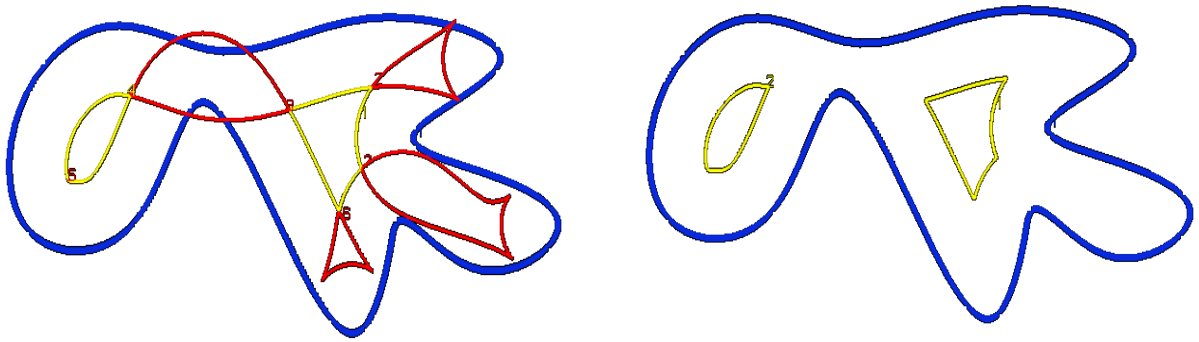
- **Surépaisseur = 8mm**



a. Avant la suppression des boucles. b. Après la suppression des boucles.

Figure 35 : Contours décalés.

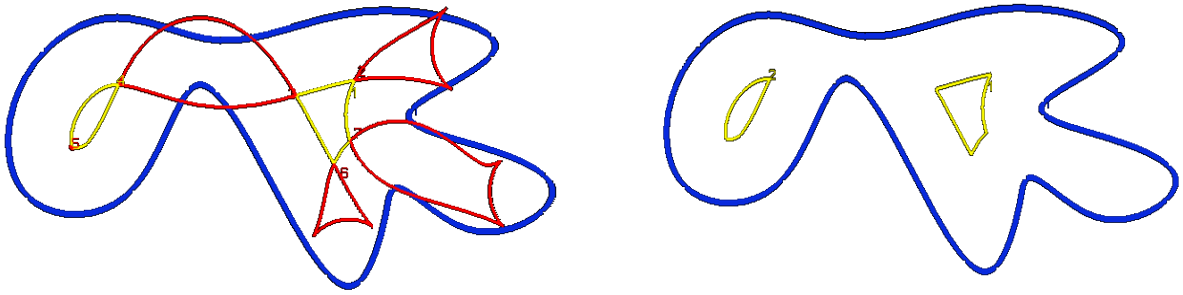
- **Surépaisseur = 9mm**



a. Avant la suppression des boucles. b. Après la suppression des boucles.

Figure 36 : Contours décalés.

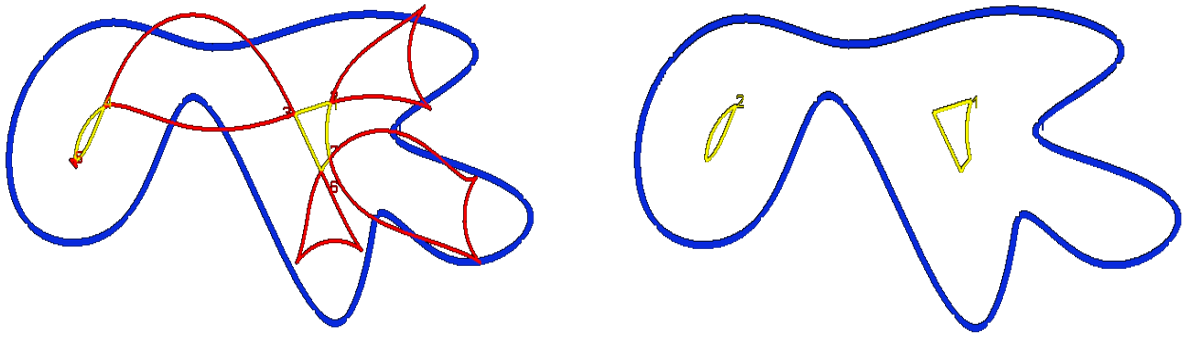
- **Surépaisseur = 10mm**



a. Avant la suppression des boucles. b. Après la suppression des boucles.

Figure 37 : Contours décalés.

- **Surépaisseur = 11mm**

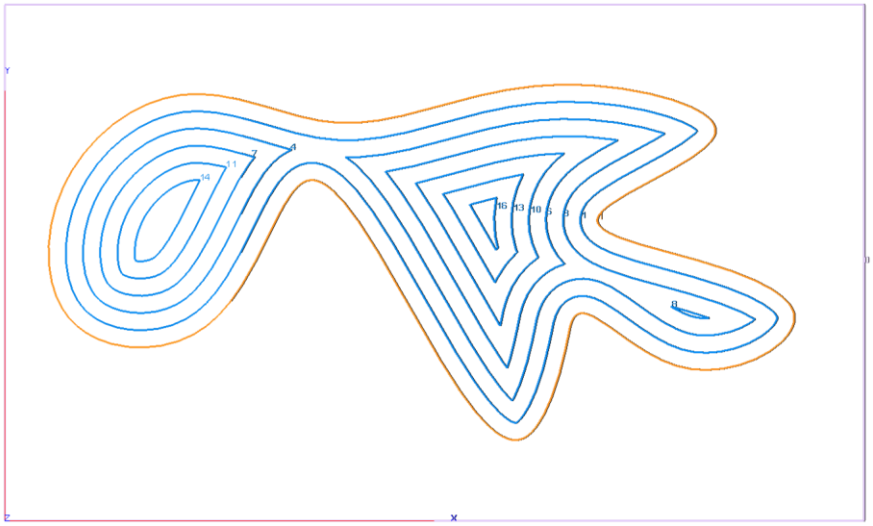


a. Avant la suppression des boucles. b. Après la suppression des boucles.

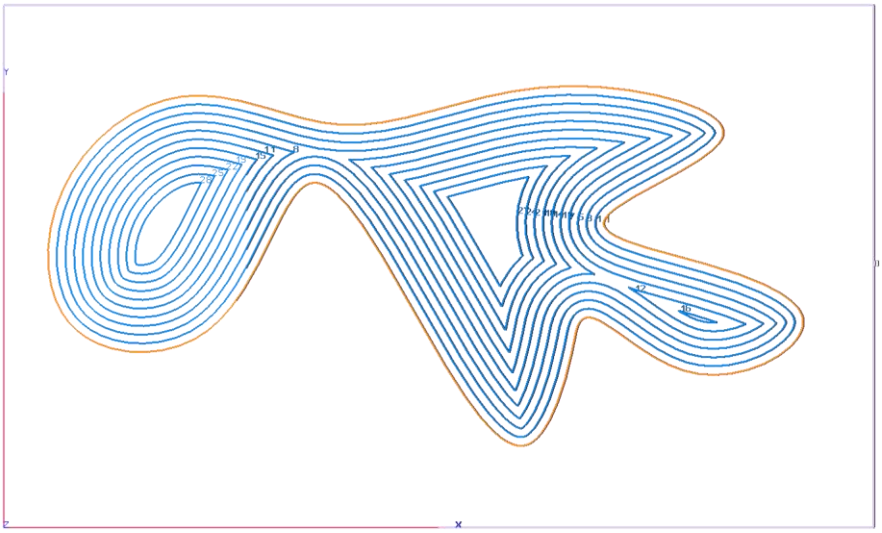
Figure 38 : Contours décalés.

A travers les différentes illustrations des contours décalés, que l'approche que nous avons proposée génère correctement les contours valides et détecte et supprime les contours invalides. Ces résultats nous permettent de générer des trajets d'outils sans problèmes d'interférences.

Les contours décalés suite au décalage successif du contour initial d'une distance égale à 2mm et d'une distance égale à 1mm sont donnés par la Figure 37. Il ressort d'après les résultats que tous les contours invalides sont éliminés et ils ne restent que les contours valides.



a. Distance de décalage égale à 2mm.



b. Distance de décalage égale à 1mm.

Figure 39 : Contours décalés.

III.6.2. Deuxième modèle STL

L'étape de la lecture du modèle STL est essentielle pour récupérer les entités géométriques du modèle utilisé (Figure 38). Cet exemple est considéré afin de montrer l'arborescence des contours initiaux dans chaque plan de coupe.

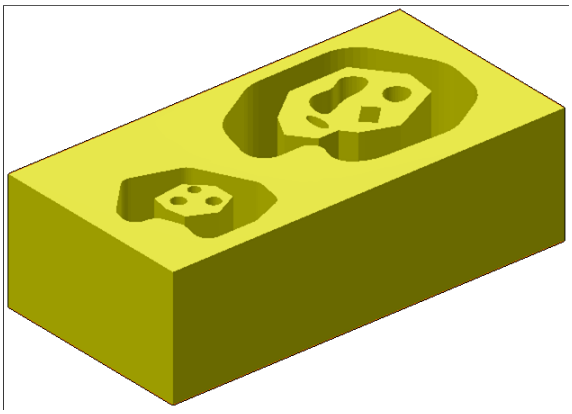


Figure 40: Deuxième modèle STL.

La génération des contours fils pour la visualisation partielle a été faite sur le deuxième modèle STL pour bien préciser l'arborescence des contours. La Figure 39 montre l'emplacement des contours sur différents plans de coupe.

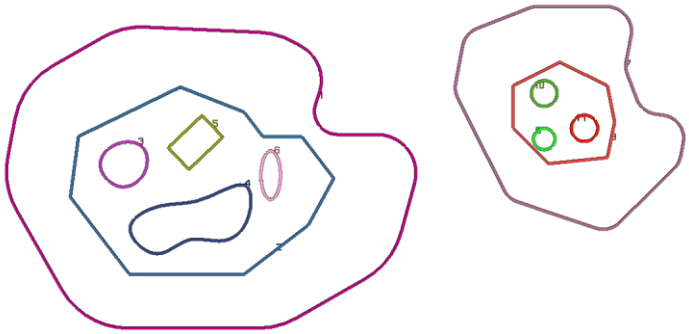


Figure 41 : Contours initiaux sur différents plans de coupe.

Sur le premier plan de coupe, les contours fils du contour 2 sont 3,5 et 6 (Figure 40).

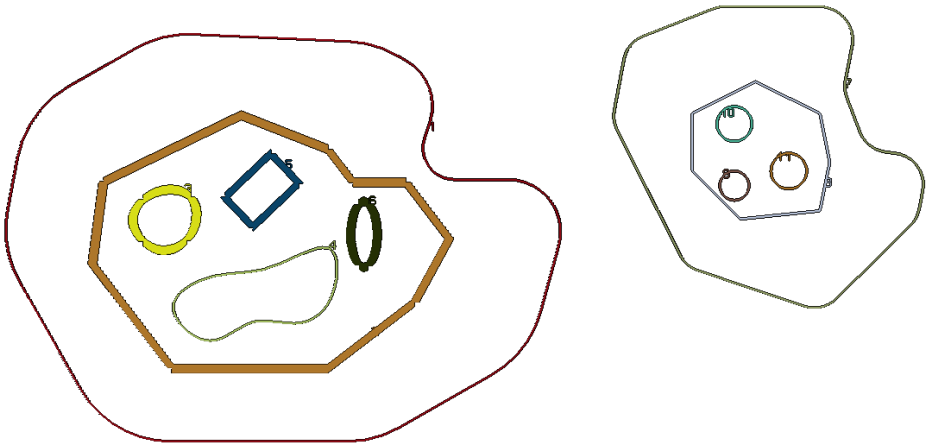


Figure 42 : Contours fils.

Les contours fils du premier niveau du premier contour sont 1 et 7 (Figure 41).

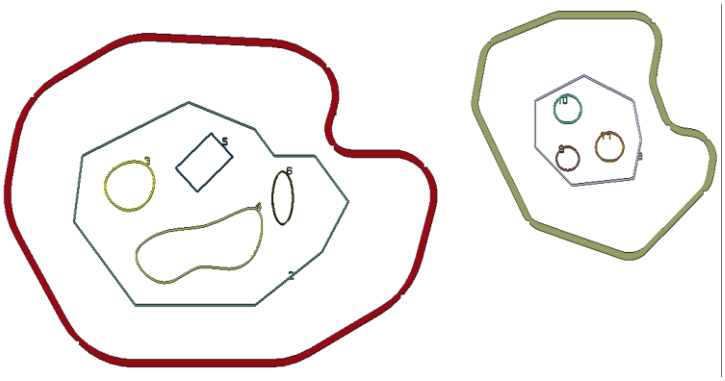


Figure 43 : Contours fils du premier niveau.

III.6.3. Troisième modèle

Dans l'onglet **« Application Sweep-Line »,** ce modèle est utilisé pour montrer les différentes étapes de l'approche développé. Un ensemble de modèles sont créés manuellement avec les coordonnées x, y et z de chaque sommet les segments des polygones sont dessinés sans l'utilisation du logiciel CAO.

Etape 1 : parmi 05 modèles, nous avons choisi le modèle 5 qui est le plus compliqué pour montrer tous les détails.(Figure 42).

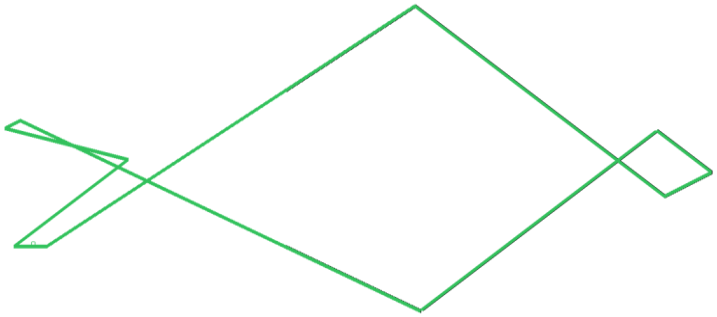


Figure 44 : Contour du modèle.

Etape 2 : la détermination des points extrêmes est nécessaire pour créer les chaines monotones. Les sommets en rouge sont les extrémités négatives et les sommets en bleu sont les extrémités positives (Figure 43).

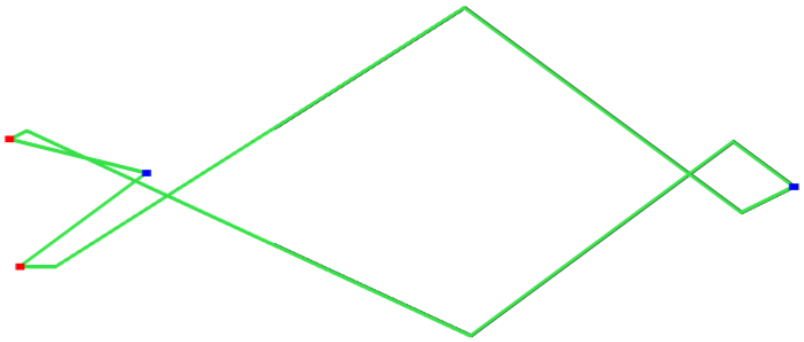


Figure 45 : Extrémités négatives et positives.

Etape 3 : suite à la détermination des extrémités, les chaines monotones sont créées (Figure 44).

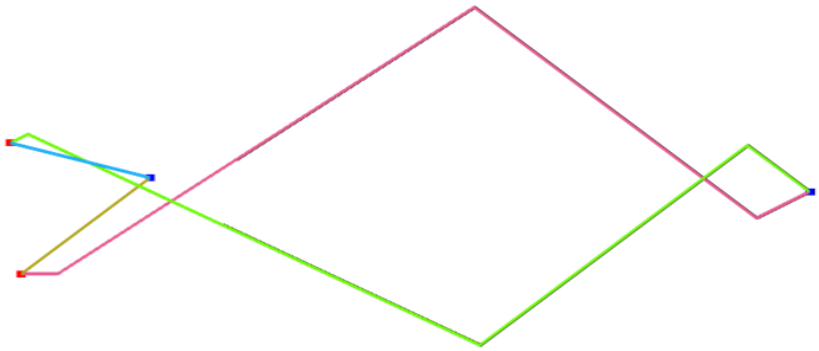


Figure 46 : Chaines monotones.

La Figure 45 montre les limites des chaines monotones.

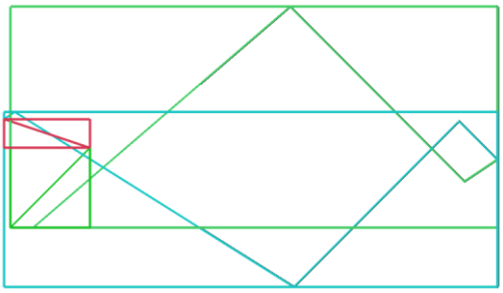


Figure 47: Limites des chaines monotones.

Etape 4 : dans cette étape, nous calculons les points d'intersection entre les lignes de balayage et les segments. La direction des chaines monotones est de gauche à droite alors que la ligne de balayage se déplace dans ce sens selon l'axe X. Ces lignes sont numérotées successivement, nous remarquons qu'on a 9 lignes (Figure 46).

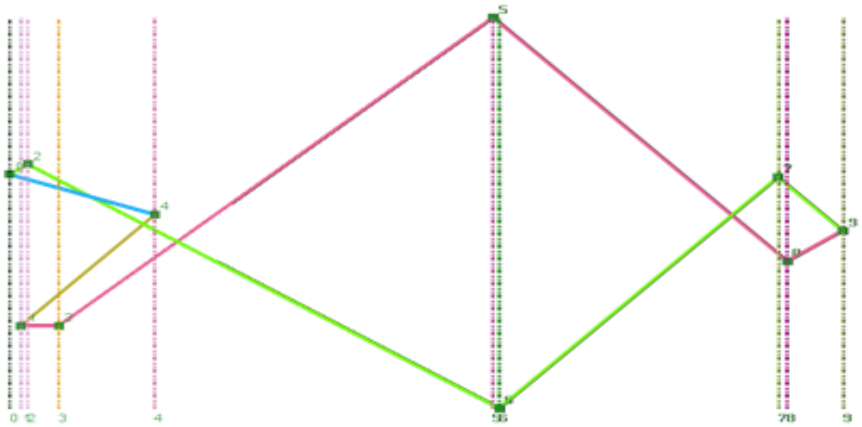


Figure 48 : Lignes de balayage.

La Figure 47 montre les sommets balayés numérotés dans l'ordre croissant.

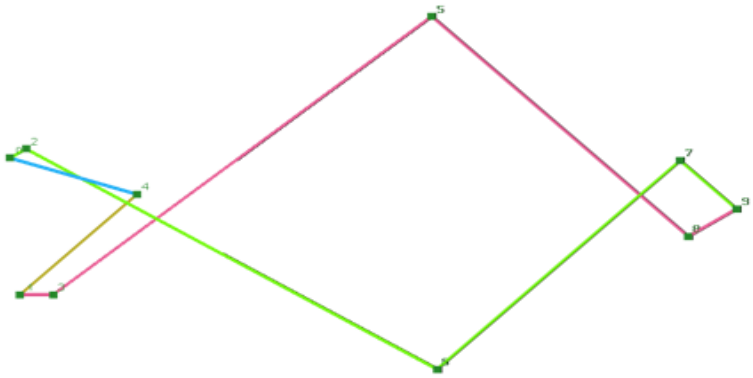


Figure 49: Sommets balayés.

Les points d'intersection entre la ligne de balayage et les segments des chaines monotones sont représentés par la Figure 48. Nous remarquons que le nombre des lignes de balayage est devenu 13 suite à l'insertion de quatre (04) de balayage associées aux quatre (04) points d'intersection des segments.

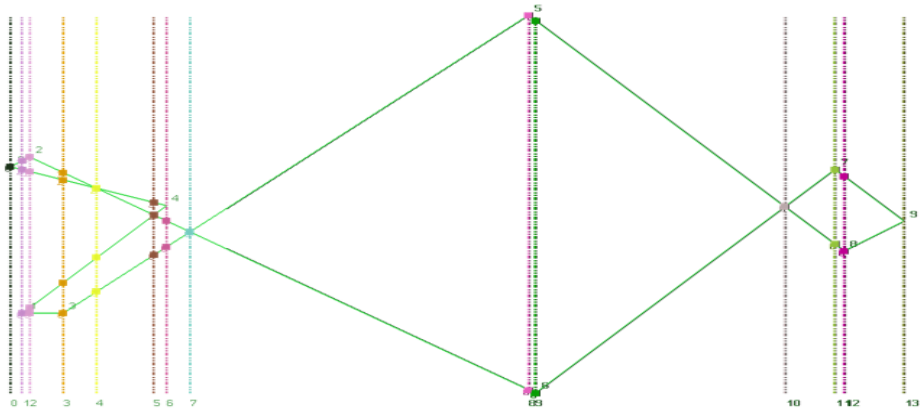


Figure 50 : Points d'intersection entre les lignes de balayage et les segments.

Etape 5 : cette étape consiste à calculer les points d'intersection entre les segments. Ici, nous remarquons que les points d'intersection sont entre plus de deux segments (Figure 49) dont les segments considérés sont sélectionnés.

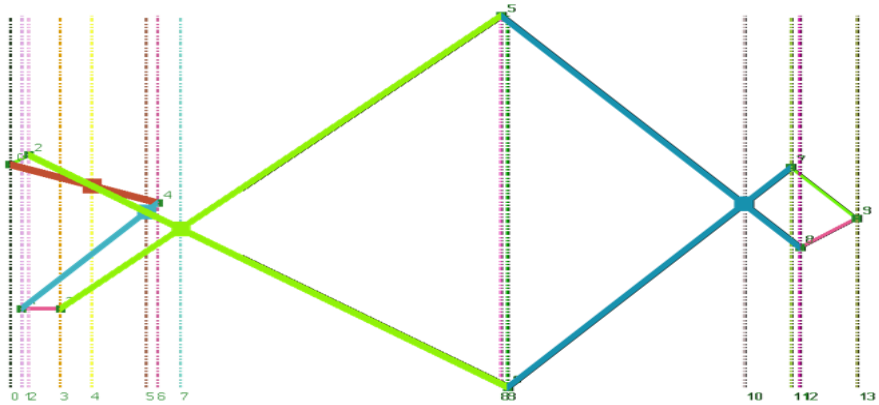


Figure 51 : Segments des points d'intersection.

Dans ce cas, la Figure 50 montre les points d'intersection sans la sélection des segments.

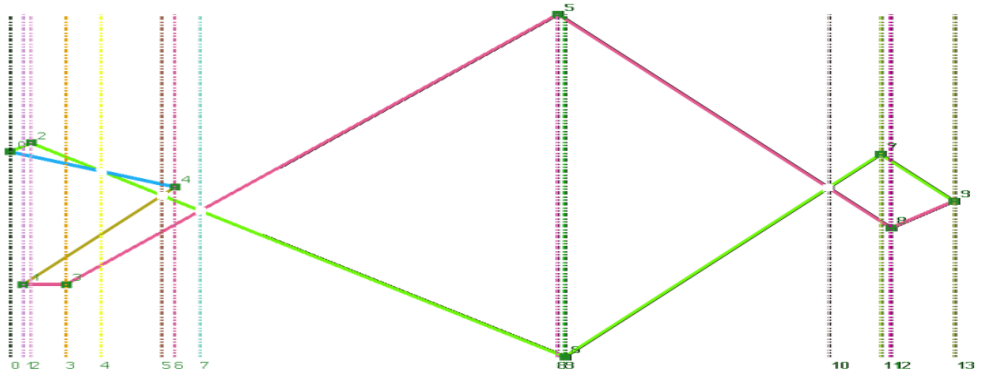


Figure 52 : Points d'intersection entre segments.

Etape 6 : après la détection des points d'intersections, la subdivision des segments est importante. La Figure 51 montre les segments numérotés avant et après la subdivision.

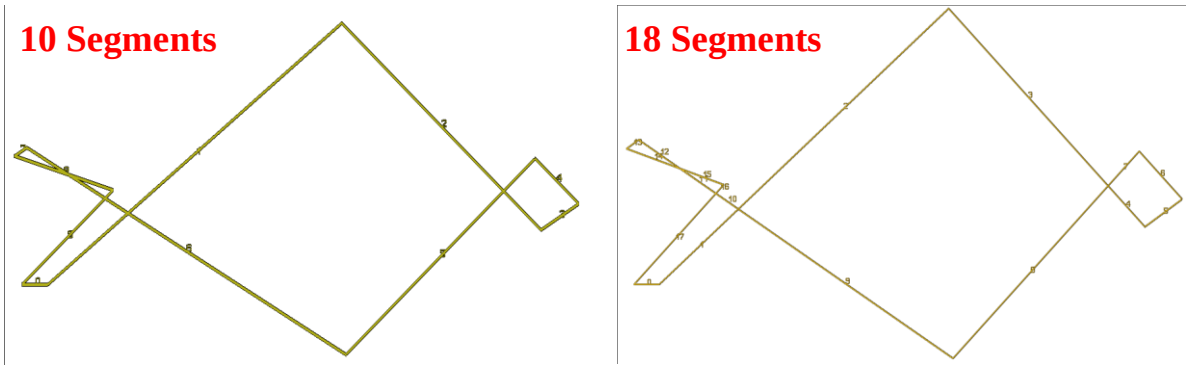


Figure 53 : Avant et après la subdivision des segments.

La subdivision des segments est réalisée au niveau des points d'intersection (Figure 52).

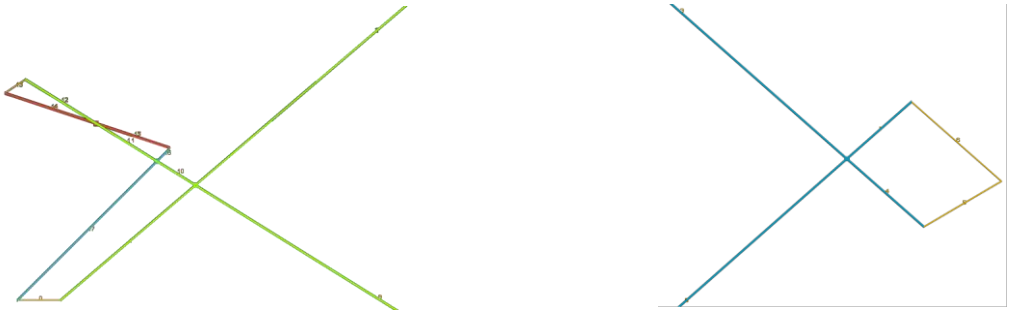


Figure 54 : Subdivision des segments.

Etape 7 : la création des nouveaux contours dépend de la subdivision des segments. La Figure 53 montre les nouveaux contours créés. Ils sont numérotés pour montrer la successivité de création.

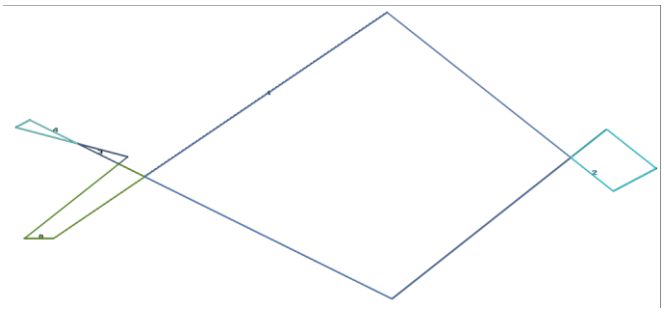


Figure 55 : Subdivision des contours.

Etape 8 : dans cette étape, les contours sont subdivisés après la détermination de leurs sens (Figure 54).

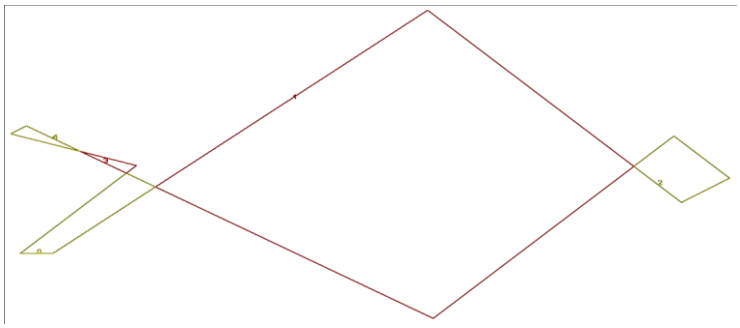


Figure 56 : Sens des nouveaux contours.

Le sens joue un rôle important dans l'élimination des contours. En plus, les contours qui n'ont pas le même sens de leurs pères doivent être éliminés. La Figure 55 montre les contours après élimination des contours invalides.

Figure 57 : Elimination des contours invalides.

III.7.Conclusion

Dans ce chapitre, les différentes étapes d'implémentation de notre module logiciel et les résultats de test menés ont été présentés. D'abord, nous avons commencé par la présentation de l'environnement de développement, des langages de programmation utilisés et du matériel utilisé. Ensuite, les différentes interfaces et leurs fonctionnalités ont été présentées afin de comprendre l'enchainement des tests dans la partie suivante. Enfin, l'utilisation des différents modèles STL et modèles crées ont pour but de valider l'approche proposée.

Dans ce projet, nous avons mis en évidence l'importance de la génération des contours décalés pour assurer des usinages continus suite à l'élimination des engagements et des dégagements des outils. Ce résultat permet de réduire les temps d'ébauchage et par conséquent les coûts.

Conclusion générale et perspectives

Le travail réalisé au sein de l'équipe Conception et Fabrication Assistées par Ordinateur « CFAO », a pour but, la conception et l'implémentation d'un module logiciel graphique et interactif permettant la génération de la trajectoire d'outils pour l'opération d'ébauchage des pièces mécaniques de formes complexes définies par leurs modèles « STL » sur des fraiseuses numériques à 03-axes, en utilisant la stratégie d'ébauchage « Contours Décalés ». L'objectif principal est la détermination des outils optimums évitant les problèmes d'interférences et de collisions, leurs combinaisons et le trajet d'outils global représentant la succession des points de positionnement de l'outil par rapport à la pièce.

Lors de la réalisation de ce projet, nous avons mené une étude bibliographique sur le processus de production des pièces mécaniques de formes complexes, le format d'échange de données « STL », les fraiseuses numériques, la stratégie d'ébauchage « Contours Décalés ». Par la suite, nous avons proposé une approche permettant de générer automatiquement les contours décalés valides, sur les différents plans de coupe, pour des contours de formes quelconques. En dernier lieu, nous avons présenté le module logiciel développé et ses fonctionnalités avec une validation de l'approche proposée sur différents modèles de pièces.

Le résultat de notre travail est le développement d'un module logiciel graphique et interactif sous Windows, en utilisant le langage C++, l'environnement de développement Embarcadero et la bibliothèque graphique OpenGL, et son l'intégration à la plateforme logicielle de l'équipe « CFAO ». Ses principales fonctionnalités sont :

- Récupération des principaux paramètres du modèle « STL » de la pièce.
- Récupération des plans et des contours de coupe.
- Détermination des types des contours.
- Raccordement des segments et élimination des segments colinéaires.
- Détermination du sens et des limites des contours.
- Détermination des contours fils et des contours fils du premier niveau.
- Détermination de l'outil optimal pour chaque contour.
- Limitation du nombre d'outils pour chaque plan de coupe.
- Génération des contours décalés.
- Détection des points d'intersections (auto-intersection).
- Subdivision des segments des contours.
- Détection et élimination des boucles invalides.
- Génération des décalages successifs.

Le projet que nous avons mené au niveau de l'équipe « CFAO » nous a permis de voir réellement l'importance de l'informatique dans le domaine de la fabrication mécanique et la nécessité de travailler dans une équipe pluridisciplinaire regroupant des compétences de différentes spécialités pour résoudre une problématique donnée.

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

1. Ebauchage des pièces de formes complexes sans et avec îlots en appliquant la stratégie « Contours Décalés » sur des fraiseuses numériques à 05-axes.
2. Ebauchage des pièces de formes complexes avec îlots en appliquant la stratégie « Contours Décalés » sur des fraiseuses numériques à 03-axes.
3. Combinaison des outils hémisphérique et cylindriques pour ébaucher les parties à fortes courbures de la pièce.
4. Intégration du parallélisme en utilisant des cartes graphiques GPU pour accélérer les calculs.
5. Adaptation des vitesses d'avance des outils en fonction du volume de la matière enlevée.
6. Optimisation du trajet d'outil lors de l'ébauchage par la stratégie « Contours Décalés » par la minimisation du nombre de dégagements des outils.
7. Identification des zones à usiner avec le même outil sur des plans de coupe successifs pour assurer un usinage en continu.
8. Génération du programme d'usinage « G-Code » en fonction des différentes configurations des fraiseuses numériques à 03-axes.

Références bibliographiques

[1] : El Hachemi Bahloul, « Techniques de fabrication conventionnelles et avancées », Polycopié de cours, Université Batna 2, Algérie, 2017/2018.

[2] : Rob Thompson, « Les procédés de fabrication » [archive], sur *Édition Vial*, 26 septembre 2012.

[3] : Christophe Tournier, « Contribution à la conception des formes complexes : la surface d'usinage en fraisage 05-axes isocrête », Thèse de Doctorat, Laboratoire Universitaire de Recherche en Production Automatisée, Ecole Normale Supérieure de CACHAN, France, 2001.

[4] : En ligne : https://blog.topsolid.fr/programmation-pieces-complexes/

[5] : Maroudi Mourad, « Conception et fabrication assistées par ordinateur du logo 3D », mémoire de Master II, Abou Bekr Belkaid-Tlemcen, Algérie, juillet 2012.

[6] : En ligne :

http://support.ptc.com/help/creo/creo_pma/french/index.html#page/manufacturing%2Fnc%2Fonline_help%2Faux_files%2Fmanufacturing.html%23

[7] : Zahia Hessainia, « Machines-outils à commande numérique », Cours de construction mécanique, Master II, Université des Frères Mentouri, Constantine 1, Algérie, 2021.

[8] : Khalil, Benabderazag «Contribution A La Realisation D'une Mocn », mémoire de Master II, Université Mohamed Boudiaf M'sila, Algérie, 2019.

[9] : Mikell P. Groover, « Automation, production systems, and computer-integrated manufacturing », Programme de baccalauréat en génie des opérations et de la logistique, École de technologie supérieure, Montréal, Canada, 2011.

[10] : Ken Giang, « Fichier STL pour l'impression 3D », guide étape par étape, 2021.

[11] : En ligne, https://en.wikipedia.org/wiki/STL_(file_format)

[12] : Lossendiere, « Adapter les fichiers maillés STL ou OBJ pour les modifier avec fusion 360 », Article Design organique, juillet. 13, 2017. https://www.makerslide-machines.xyz/fr/2017/07/13/adapter-les-fichiers-stl-ou-obj-pour-les-modifier-avec-fusion-360/

[13] : David Prat, « Développement et modélisation de stratégies de fraisage 5 axes de finition - Application à l'usinage de veines fermées », Thèse de Doctorat, Laboratoire Bourguignon des Matériaux et Procédés, France, 9 décembre 2014.

[14] : Aurelien Maurel-Pantel, « Etude expérimentale et modélisation par éléments finis du procédé de fraisage : Applications à l'identification paramétrique des lois de comportement », Mécanique [physics.medph]. Université de Franche-Comté, France, 2009.

[15] : Slamani Mohamed, Kermiche Younes, « Évaluation de la qualité d'alésages obtenus par fraisage circulaire », Master II En Génie Mécanique, Université Mohamed Boudiaf - M'sila, Algérie, 2015/2016.

[16] : M. Benghersallah, Aouassa Izzeddine, « Etude de la gamme d'usinage d'un support d'enrouleuse de tôles et étude de l'effet des paramètres de fraisage sur la rugosité », mémoire de Master II, Universite Badji Mokhtar Annaba, Algérie, 2013/2014.

[17] : Bahloul El hachemi, « CAO - FAO - FEM - Optimisation en usinage », Cours-TP, Université Batna 2, Algérie, 2015 à 2017.

[18] : Lagred Ahmed, Tahraoui Mohamed El Haddi « Contribution à l'amélioration de la qualité d'usinage en fraisage à sec », mémoire de Master II, Universite Badji Mokhtar Annaba, Algérie, Année 2018.

[19] : Tammam Salloum, « Conception et réalisation de pièces prototypes fonctionnelles en usinage sur machines-outils à commande numérique 5 axes ». Automatique / Robotique. École normale supérieure de Cachan - ENS Cachan, France, 2009.

[20] : Éloy Letellier, « Opération d'usinage et d'assemblage réalisation des surfaces », Présentation, https://slideplayer.fr/slide/505273/

[21] : Meziane El Hadj, « Technologie de base », Polycopié de cours, Université Hassiba Benbouali, Chlef, Algérie, Novembre 2017.

[22] : Mohamed Bey, Tchantchane Zahida « Optimisation de l'opération d'ébauchage des surfaces gauches par l'intégration des outils optimums et des vitesses d'avance », first international conference on industrial engineering & manufacturing, ICIEM 2010, May 9-10, Batna, Algérie, 2010.

[23] : Christophe Tournier, Emmanuel Duc « Stratégies d'usinage et productivité en fraisage des surfaces complexes », 16ème Congrès Français de Mécanique, Nice, France, Septembre 2003.

[24] : Sonia Djebali, « Optimisation globale du processus d'usinage des surfaces gauches », Thèse de Doctorat, Université Toulouse 3, Paul Sabatier, France, 13/11/2014.

[25] : El hachemi Bahloul, « Génération de trajectoires et choix d'outils dans l'usinage de cavités de formes quelconques », Thèse de Doctorat, Université De Batna 2 Faculté de Technologie, Département de Génie Mécanique, Algérie, 27/ 02/ 2017.

[26] : Abdelmadjid Hatna, « Contribution à l'élaboration d'une méthodologie d'évidement de poches complexes – application à l'usinage des surfaces gauches », Thèse de Magister, Institut national de Génie Mécanique (IGNM), Boumerdès, Algérie, 13 Décembre 1995.

[27] : Benmimouna Amina Et Elouchefoune Imene, « Mise en place d'un environnement logiciel pour la virtualisation de l'usinage des pièces complexes sur des fraiseuses numériques à 5 axes», mémoire de Master II, Université Saad Dahleb, Blida Faculté des Sciences Département d'Informatique, Algérie, 2019/2020.

Figures

[1] : https://fr.wikipedia.org/wiki/Fraiseuse

[2] : https://fr.wikipedia.org/wiki/Fraiseuse

[3] : https://www.zoneindustrie.com/Produit/Fraiseuse-CNC-3-axes-11606.html

[4] :https://www.commerce-machines-occasion.fr/fonctionnement-differents-types-de-fraiseuses/

[5] : https://www.fanuc.eu/be/fr/applications/fraisage-avec-centres-d%27usinage-verticaux

[6] : https://www.sandvik.coromant.com/fr-fr/knowledge/milling/pages/up-milling-vs-down-milling.aspx

[7] : https://en.wikipedia.org/wiki/STL (file format)

[8] : https://en.wikipedia.org/wiki/STL (file format)

[9] : https://slideplayer.fr/slide/1140042/

[10] : https://slideplayer.fr/slide/505273/