REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLAB DE BLIDA
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

MEMOIRE DE FIN D'ETUDES

Pour l'obtention

D'un Diplôme de Master en Informatique

Option : Ingénierie de Logiciel

THÈME :

Optimisation de la Triangulation de Delaunay d'Objets à Partir d'un Nuage de Points 3D

Réalisé par :

M^elle^. MEBARKI Fedwa

M^elle^. MECHTA Rim

Soutenu devant :

Mr. BEY Mohamed	CDTA,	Encadreur
Mr. BENDIFALLAH Hassène	CDTA,	Encadreur
Mme. TCHANTCHANE Zahida	CDTA,	Encadreur
Mr. KAMECHE Abdallah Hicham	USDB,	Promoteur
Mme.TOUBALINE	USDB,	Président
Mr.OULD AISSA	USDB,	Examinateur

2018/2019

Résume :

Ce travail s'intègre dans le cadre de développement d'une plateforme logicielle pour la production des surfaces de formes complexes initiée par l'équipe Conception et Fabrication Assistées par Ordinateur « CFAO » de la Division Productique et Robotique « DPR » du Centre de Développement des Technologies Avancées « CDTA ».

L'objectif de ce travail consiste à proposer et à implémenter deux approches, une pour la reconstruction des modèles 3D à partir d'un nuage de points quelconque, et l'autre vise à optimiser le temps de cette reconstruction. Pour cela, nous avons utilisé le concept de la triangulation de Delaunay 3D qui consiste à approximer le nuage de points par des éléments géométriques simples (triangles, tétraèdres, etc.) qui permettent d'arriver à un modèle STL tout en optimisant le temps de traitement.

Mots-Clés : Reconstruction, Triangulation de Delaunay, Optimisation, Modèle STL, Nuage de Points.

ملخص

نهتم في هذه المذكرة بتطوير مجال تصنيع القطع ذات الاسطح المعقدة التي يتم معالجتها على مستوى وحدة التصنيع و الروبوتيك التابعة لمركز تنمية التكنولوجيات المتطورة . الهدف من هذا العمل هو اقتراح وتنفيذ طريقتين الاولى تهدف إلى إعادة تصنيع النماذج ثلاثية الأبعاد انطلاقا من سحابة نقط والثانية تهدف إلى الاقتصاد في زمن إعادة التصنيع والتشكيل. ولهذا الغرض قمنا باستخدام طريقة التثليث المتزامن حيث، يتم تقريب المخطط بعناصر هندسية بسيطة مثل المثلثات، رباعيات الاوجه، ... إلخ. الدى يسمح لك بالوصول الى نموذج STL مع تحسين الوقت. الحل المقترح موافق عليه والنتائج مرضية

الكلمات المفتاحية: اعادة التصنيع، ثلاثية دو لوناي، نموذج ستل، سحابة من النقاط.

Abstract

This work is a part of the software developed by "CAD/CAM" team of the Center for the Development of Advanced Technologies "CDTA" dedicated to the machining of free form surfaces on CNC milling machines.

The objective of this work is to propose and to implement two approaches, one allows the reconstruction of 3D models from a cloud of points, and the other is to optimize the time of reconstruction. To solve this problem, the cloud of points is approximated by simple geometric elements (triangles, tetrahedra, etc.) allowing us to reach a STL model while optimizing time.

Keywords: Reconstruction, Delaunay Triangulation, Optimization, STL Model, Cloud of points.

Remerciement

Louanges tout d'abord à ALLAH qui est à l'origine de toute réussite dans notre vie.

Nous adressons nos sincères remerciements à Mr. Bey Mohamed qui nous a confié ce sujet et qui nous a assuré l'encadrement de notre projet et l'intérêt qu'il nous a porté pour notre travail, sa bienveillance, sa rigueur scientifique, ses discussions fructueuses, ses hautes qualités humaines, qui ont constitué une aide précieuse et qui nous a permis de mener à terme ce travail.

Nos remerciements vont à l'encontre de Mr. Bendifallah Hassène qui a été d'une aide appréciable et pour ses efforts qui nous ont guidé et qui ont enrichi notre travaux tout au long du stage. Ainsi que Mme. Tchantchane Zahida.

Par la même occasion, nous adressons nos remerciements à Mr Kameche Abdallah Hichem pour ses efforts et sa bonne coordination durant le stage.

Nous tenons à remercier aussi toute l'équipe du CDTA .

Nous remercions les membres du jury pour avoir accepté d'évaluer notre travail.

Enfin, nos remerciements s'adressent aussi à tous ceux qui ont participé, de près ou de loin, à l'élaboration de ce projet de fin d'études et en particulier à nos familles et nos amis.

Merci

Dédicace

Je dédie ce projet :

A ma chère mère, A mon cher père,

Qui n'ont jamais cessé de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.

A mes deux frères, A ma chère sœur,

Pour leur soutien et leurs conseils précieux tout au long de mes études.

A mes encadreurs, Mr. BEY ET Mr. BENDIFALLAH

A mes chers amis, Pour leur aide et support dans les moments difficiles.

RIM MECHTA

Dédicace

Je dédie ce mémoire à :

Mes parents :

Ma chère maman Hayet, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

Mon cher père Mohamed, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Allah faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutient permanent venu de toi.

Mon petit prince frère Mohamed tadj el Mouloke et mes amours sœurs Amani, Hanine « Manar » et ma petites princesse Djoumana, ma meilleure sœurs copine Sabrina .

Fedwa. Mebarki

Sommaire

Liste des figures

Introduction générale :

L'outil informatique à connu une évolution très importante ces dernières années de tel sorte qu'il est introduit dans tous les domaines et en particulier dans les domaines de la conception et de la fabrication mécanique assistées par ordinateur (CAO, FAO et CFAO) tels que l'automobile, l'aéronautique, l'industrie du moulage, …etc.

Son utilisation a permis d'augmenter la productivité et la qualité des produits réalisés. Ces produits de formes complexes sont obtenus selon la nature des matériaux par des procédés de moulage, d'injection, d'emboutissage, d'usinage, …etc. En raison de la réduction des temps de fabrication de produits, des logiciels de CFAO sont utilisés par les différentes entreprises pour assister leurs ingénieurs dans les phases de conception, d'analyse et de planification de l'usinage sur des machines outils à commande numérique.

Néanmoins, la concordance entre le produit réalisé et la géométrie de la forme définie par le concepteur découle de l'aptitude de chacune des activités du processus de conception et de fabrication à modéliser ou à produire la géométrie attendue. En effet, les techniques de construction des surfaces restent limitées ou insuffisantes pour accorder la réalisation des formes souhaitées par le concepteur ainsi que les raccordements entre les surfaces. Ce dernier est obligé de suivre les fonctionnalités du logiciel qui sont mis à sa disposition. Pas forcément, que le résultat final correspond toujours à l'intention du concepteur, mais c'est la plus proche représentation géométrique qu'il peut obtenir.

Dans la pratique industrielle, les modèles des surfaces sont obtenus soit en utilisant un logiciel de CAO si les formes sont plus ou moins simples, soit le processus du « Reverse Engineering » si les formes sont très complexes ou si les modèles CAO ne sont pas disponibles. La deuxième méthode a comme objectif, la reconstruction du modèle CAO de l'objet en se basant sur un nuage de points acquis par digitalisation d'un objet réel sur des Machines à Mesurer Tridimensionnelle «MMT». La reconstruction du modèle CAO est une étape très délicate et consommatrice de temps, ce qui nécessité le développement de méthodes permettant d'optimiser le temps de calcul afin de générer le modèle STL souhaité sans avoir à remonter au modèle CAO. Puisque ce modèle est un format d'échange de données standard, il peut être généré pour n'importe quelle conception dans les logiciels de CAO.

Objectifs et besoins du travail :

Ce travail s'insère dans le cadre de développement d'une application logicielle graphique et interactive sous Windows automatisant la tâche de la production des surfaces de formes complexes 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).

Les travaux menés dans le cadre de ce mémoire portent sur « Optimisation de la triangulation de Delaunay d'objets à partir d'un nuage de points 3D ». Ce concept a été introduit pour pallier aux problèmes de réalisation de produit dont le modèle géométrique n'existe plus ou impossible à reconstruire. Autrement dit, il s'agit de mettre en place une application logicielle graphique et interactive pour la reconstruction d'objet 3D qui prend en charge ; la localisation des tétraèdres à modifier en un temps optimal et à optimiser la zone de mise à jour de la triangulation , ce qui minimisent les temps de la triangulation ainsi que la génération d'un modèle STL.

Notre projet est une continuité d'un ensemble de travaux traitant la reconstruction d'objets 3D à partir de nuages de points structurés et non structurés réalisés dans le cadre de l'activité de recherche de l'équipe CFAO du CDTA et prend en charge les points suivants :

- Recherche des approches efficaces pour une modélisation en trois dimensions.
- Recherche d'une stratégie efficace pour déterminer le tétraèdre de départ dans le but de minimiser la zone de recherche.
- Développement d'une méthode qui permet de localiser les tétraèdres à modifier en un temps optimal.
- Génération d'une triangulation 3D (modèle STL) approximant la peau extérieure d'objets de n'importe quelles formes à partir de nuages de points quelconques.
- Minimisation de la zone de mise à jour de la triangulation toute en permettant la réduction du temps de traitement.

Pour atteindre cet objectif, une étude sur l'état de l'art du processus de reconstruction d'objet 3D est nécessaire.

Structuration du mémoire :

Le présent mémoire est composé de trois chapitres :

- Le premier chapitre est consacré à l'étude des notions sur les surfaces complexes et leurs modèles de représentation, le processus du Reverse Engineering et les différentes méthodes de reconstruction d'objets ainsi que le modèle d'échange de données STL.
- Dans le deuxième chapitre, une modélisation explicite de l'approche adoptée et le développement de tous les algorithmes utilisés est détaillé.
- Le dernier chapitre présente l'implémentation informatique de l'application et les tests de validation des résultats du travail effectué.

Nous terminons ce mémoire par une conclusion générale et des perspectives.

Chapitre I

Etude Bibliographique

Introduction

Tous les secteurs industriels sont aujourd'hui soumis à une pression économique importante et à la concurrence internationale imposée par le phénomène de la globalisation des marchés. Pour répondre à cet état de fait, les entreprises sont contraintes d'innover et d'améliorer le cycle d'élaboration des produits et/ou du processus depuis l'idée jusqu'à la mise sur le marché des produits. Les demandes de clients et/ou du marché correspondent à de profondes changements et imposent aux entreprises de fournir un produit ou/et des services respectant les exigences de « prix toujours plus faible, en proposant un choix varié, disponible et de bonne qualité ». Les stratégies mises en place par les entreprises sont diverses et conduisent à des choix industriels répondant partiellement à ces exigences (délocalisation, fabrication de produits intermédiaires, transfert de technologie, automatisation de la production, nouveaux systèmes manufacturiers, développement de la logistique, etc.).

Depuis l'avènement de l'industrialisation, les ingénieurs ont développé des systèmes manufacturiers et ont cherché à augmenter leurs efficacités par le développement de la productivité puis de la flexibilité concomitamment à l'évolution de la gestion industrielle et à l'automatisation. Or, il faut aujourd'hui faire preuve d'une très grande réactivité, pouvoir répondre rapidement à l'apparition d'un nouveau produit (évolution du design) et/ou une nouvelle demande (fluctuation de volume) afin de rester compétitif. Prévoir la tendance du marché reste extrêmement difficile et celui-ci rentre dans une logique d'une vision à long terme (suivant les demandes de clients, étude de l'évolution du marché) [1].

L'évolutivité rapide des produits et de la concurrence ont conduit les entreprises industrielles à faire un pas supplémentaire en imaginant de nouveaux systèmes de fabrication pour assurer simultanément une haute productivité et une haute flexibilité avec une contrainte nouvelle de changement rapide de famille de pièces. C'est le processus du « Reverse Engineering » qui est présentée en détail dans la partie 1 de ce chapitre.

La partie 2 de ce chapitre, est consacrée à la présentation de la triangulation de Delaunay support de la reconstruction d'objets 3D ainsi qu'à la description de la méthode de Flip, objet de notre étude.

1. Reverse Engineering

Les concepts de la conception et de la fabrication assistées par ordinateur « CFAO » sont utilisés dans différents domaines afin de concevoir et de produire des objets physiques à partir

Partie I:

Généralités sur la modélisation et la reconstruction d'objet

de modèles numériques. Cependant, la Retro-Conception ou Reverse Engineering est l'opération inverse qui consiste à déterminer le modèle numérique d'un objet physique [2].

1.1. Définition

La rétro-conception (RC) ou la rétro-ingénierie (Reverse Engineering), est une activité qui peut être utilisée dans plusieurs domaines tels que l'ingénierie mécanique, électrique, informatique, etc. L'objet principal de la RC est l'analyse d'un produit existant (logiciel, pièce mécanique, etc.) afin de produire une copie ou une version améliorée de celuici (Bernard et al. 2010), (Gameros et al., 2015) [3]. La Figure 1 illustre l'organigramme qui d'écrit le processus général de la rétro-conception par rapport au cycle de vie du produit.

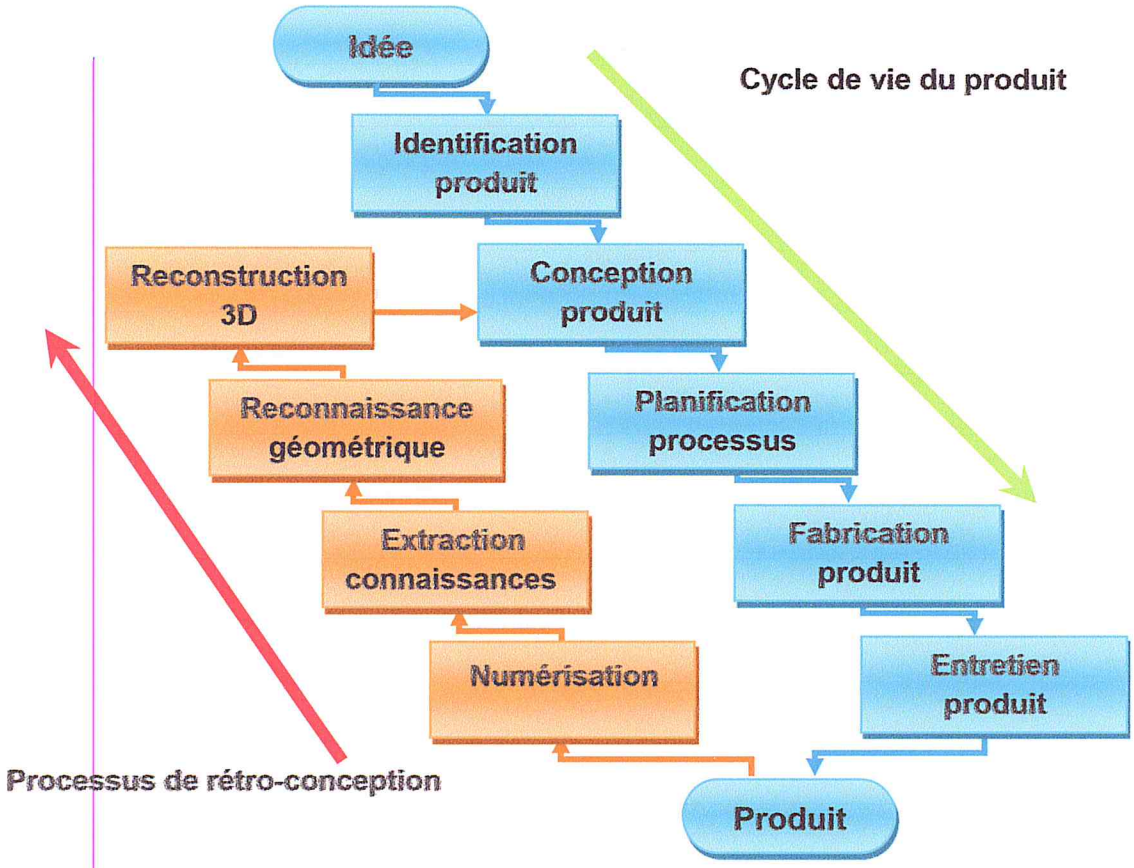
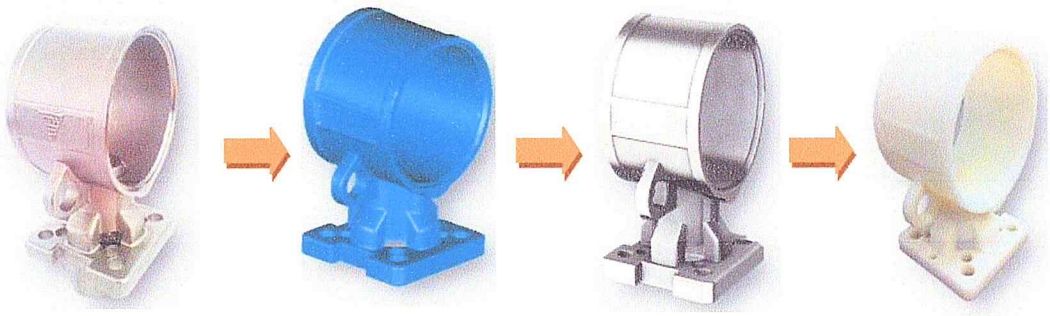


Figure 1. Processus de la rétro-conception par rapport à un cycle de vie d'un produit

Dans une procédure d'ingénierie directe, le concepteur élabore un dessin du produit avec les moyens de fabrication puis ce dernier est fabriqué selon le dessin de conception.

En ingénierie mécanique, la RC suit la procédure inverse de l'ingénierie directe. Dans un premier temps, les ingénieurs identifient les composants du système et leurs interactions. Le système est démonté afin d'en découvrir sa structure et son fonctionnement. La duplication peut alors se faire en capturant les dimensions physiques, les fonctionnalités et les propriétés matérielles des objets pour la réalisation de fichiers CAO « Conception Assistée par Ordinateur

» à partir des objets physiques. La phase finale consiste à reproduire un modèle représentant le produit original avec une précision conformément au dessin établi (Figure 2).



a. Corps de vanne. b. Maillage. c. Modèle CAO. d. Prototype.

Figure 2. Rétro conception du corps de vanne.

1.2. Motivation du Reverse Engineering

Le Reverse Engineering est utilisé et adopté dans les cas suivants :

- Le producteur original n'existe pas, mais le client a besoin de ce produit.
- Le producteur original a arrêté de fabriquer le produit.
- La documentation de conception du produit original est perdue.
- Inspection et contrôle d'une pièce produite avec son modèle CAO.
- Analyse des produits des concurrents.
- Création des données en 3D d'un modèle.
- Documentation architecturale, construction et de mesure.

1.3. Domaines d'utilisation

La reconstruction d'objets est une discipline relativement jeune qui intéresse plusieurs domaines d'applications et de recherches. Parmi ceux-ci :

- Génération d'un modèle CAO d'une pièce à partir de sa maquette.
- Clonage et recopiage des œuvres d'art.
- Inspection et contrôle du composant fabriqué avec son modèle CAO.
- Technologie des prothèses.
- Vision artificielle et robotique.

1.4. Phases de rétro-conception

La rétro conception se déroule en trois phases :

Phase 1 : Acquisition des données numériques 3D

Cette phase de numérisation consiste à acquérir des points 3D en digitalisant la surface de la pièce. Ces mesures sont tridimensionnelles (3D) et peuvent être réalisées sur tout type de

pièces. Différentes technologies de numérisation peuvent être utilisées : avec contact (palpage) ou sans contact (triangulation par laser, optique sans laser, etc.) (Figure 3).

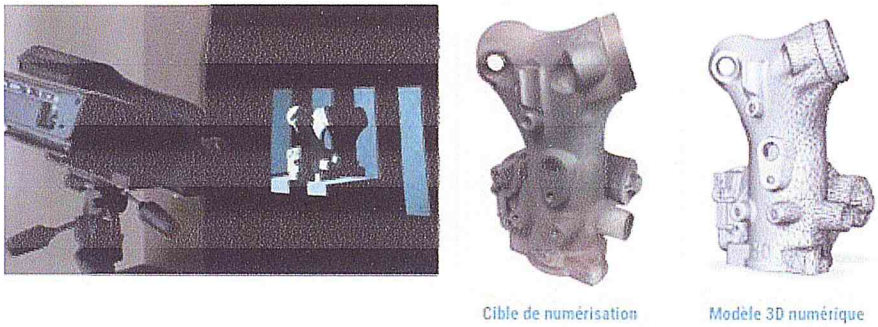


Figure 3. Scanner 3D.

- **Phase2 : Triangulation**

Cette phase regroupe les étapes suivantes (Figure 4) :

- Traitement et alignement des données obtenues : elle comporte les tâches d'alignement, de réduction du bruit de points et lissage des points.
- Modélisation polygonale : elle passe par la création du maillage polygonale suivie de la réparation du modèle obtenu.

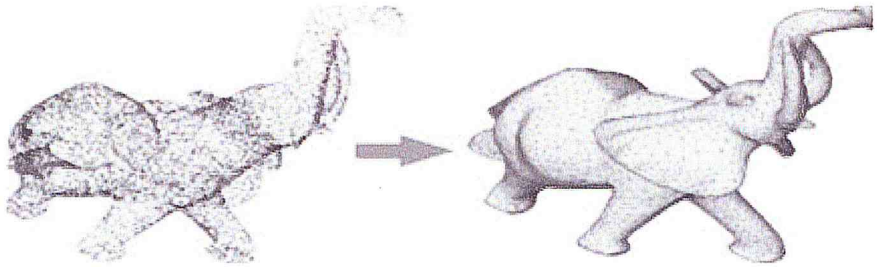


Figure 4. Triangulation.

- **Phase 3 : Reconstruction 3D**

La dernière étape de la reconstruction consiste à générer sur le modèle polygonal un réseau de courbes afin de reconstruire les surfaces continues épousant la pièce numérisée [4].

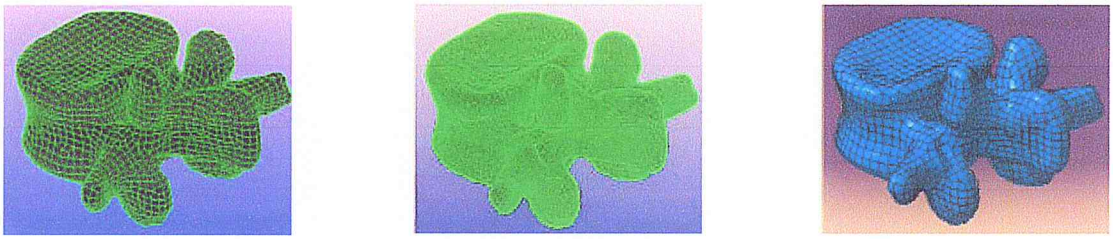
Par rapport aux deux phases, il est possible de regrouper la phase de la triangulation et de la reconstruction dans une seule phase. Au niveau de l'équipe « CFAO » du « CDTA », une méthodologie basée sur le principe de la triangulation de Delaunay est proposée pour l'approximation de nuages de points non structurés. La démarche est inspirée de la méthode Bowyer-Watson [5].

2. Reverse Engineering de forme complexe

La reconstruction des modèles en trois dimensions « 3D » avec une bonne précision est une tâche délicate, couteuse, dépendant des possibilités de modélisation disponibles dans les

logiciels de « CAO » et exigeant une grande interactivité entre le logiciel et le concepteur. En outre, le modèle résultant de l'objet est une approximation du nuage de points et nécessite très souvent des ajustements. Pour résoudre ce problème, le nuage de points est approximé par des éléments géométriques simples (triangles, tétraèdres, etc.). Les nuages de points sont très denses et volumineux et exigent des calculs intensifs ce qui augmente les temps de traitement et donc les coûts pour la génération d'une triangulation [6]. La Figure 5 donne un exemple d'une méthode effective pour la reconstruction 3D par occupation spatiale de modèle biomécanique : disque lombaire humain modélisé à partir de nuage de points tirés de séquences d'images d'un scanner (CT).

a. image CT. b. Surface implicite. c. Discrétisation des données.



d. Projection du rectangle. e. Intérieur des points générés. f. lissage de surfaces.

Figure 5 : Reconstruction de l'os du disque de la colonne vertébrale.

Le processus de reconstruction d'objet 3D à partir d'un nuage de points comporte cinq étapes principales nécessaires à la génération du modèle STL (Figure 6).

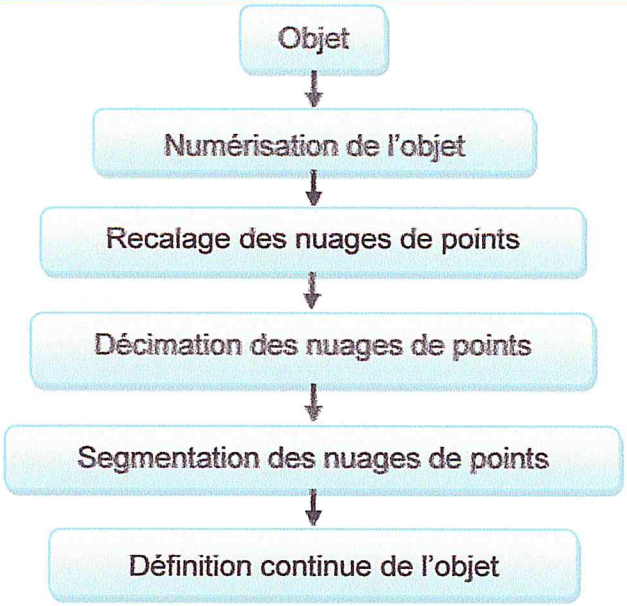
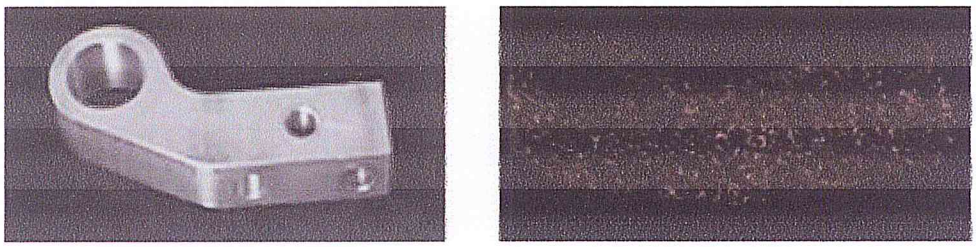


Figure 6. Processus de reconstruction d'objets.

2.1. Numérisation de l'objet

La mesure d'objets tridimensionnels se traduit par un ou plusieurs ensembles de points 3D qui, une fois assemblés, forment l'image en profondeur de l'objet. Ces ensembles sont constitués de suites de points P_i de coordonnées (x_i, y_i, z_i). Ces triplets offrent une définition des surfaces de l'objet sans indications concernant son type topologique ni sa géométrie [7].



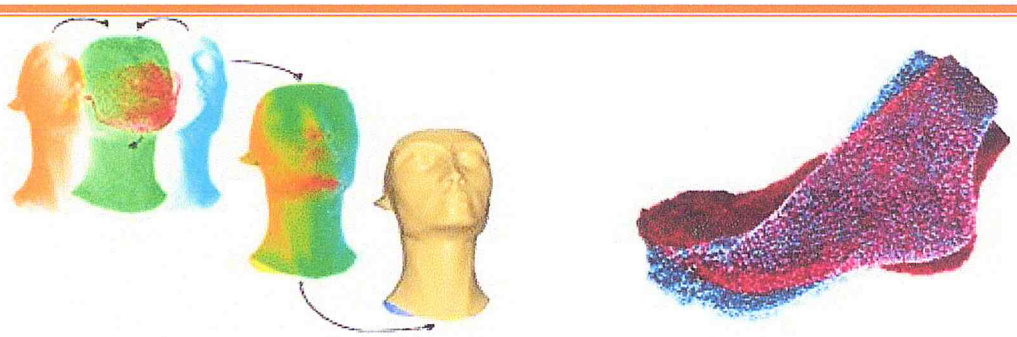
a. Pièce avant numérisation. b. Pièce après numérisation.

Figure 7. Numérisation de l'objet

2.2. Recalage

Un objet de forme complexe ne peut pas être numérisé en prenant en compte une seule face, plusieurs points de vue sont nécessaires. Dans ce cas, plusieurs nappes de points sont créées. Le recalage consiste à mettre en correspondance ces derniers suivant deux modes :

- **Recalage simple** : c'est un recalage deux à deux. Deux nuages de points sont recalés pour former un seul. Ce dernier sera recalé avec un autre et ainsi de suite (Figure 8.a).
- **Recalage globale** : permet de faire une correspondance simultanément de plusieurs nuages de points en utilisant l'algorithme ICP « Iterative Closest Point » (Figure 8.b).



a. Recalage simple. b. Recalage avec l'algorithme ICP. **Figure 8.** Recalage des nuages de points.

2.3. Décimation

La décimation sert à simplifier le nuage de points pour le rendre manipulable sur ordinateur par diminution du nombre de ces points. Il est important de signaler que la densité des points dans les régions fortement courbées doit être plus grande par rapport aux endroits de faibles courbures [8].

a. Objet sans décimation. b. Objet avec décimation. **Figure 9.** Objet ayant subi une décimation.

2.4. Segmentation

C'est une étape clé dans la reconstruction 3D. Elle identifie chaque sous ensemble comme entité indépendante en utilisant une des méthodes de segmentation (Figure 10) :

- Segmentation par reconnaissance de formes : consiste à estimer les paramètres géométriques d'un objet (plan, cylindre, etc.…) à partir d'un ensemble de points.
- Segmentation par agrégation de points : c'est une étude de critères locaux des nuages de points autour de graines. Ces critères une fois définis, sont testés au voisinage de la graine. Si les points voisins répondent aux critères, alors ils sont acceptés dans ladite surface segmentée à partir de certains critères (proximité, planéité locale et normalité) [9].

Figure 10. Segmentation d'un objet.

2.5. Définition continue de l'objet

Cette étape est celle qui présente les plus grandes difficultés dans le processus de reconstruction. Dès que les régions sont identifiées, une approximation est faite par des modèles géométriques continus (Figure 11).

a. Nuage de points. b. Segmentation. c. Modèle continu.

Figure 11. Définition de l'objet continue.

3. Méthode de modélisation des surfaces complexes

La modélisation de surfaces complexes est devenue un outil incontournable dans l'industrie moderne. Elle permet de représenter et d'étudier les surfaces constituant des objets. Ils existent trois méthodes de modélisation des surfaces basées sur l'utilisation des points.

3.1. Interpolation surfacique

Cette méthode est basée sur un nuage de points décrivant la forme de la surface désirée (Figure 12.a). Ensuite, une surface passant par ces points est construite par interpolation. [10].

a. Nuage de points b. Interpolation du nuage de points

Figure 12. Interpolation de la surface.

3.2. Approximation de la surface

L'approximation de la surface est fréquemment utilisée dans la conception de formes. Elle consiste à déterminer la surface qui approxime le mieux tous les points mesurés mais qui ne passe pas nécessairement par ces points (Figure 13) [10].

a. Nuage de points b. Approximation du nuage de points

Figure 13. Approximation de la surface.

3.3. Surface Paramétrique

C'est une surface continue définie par un réseau de contrôle (Figure 14).

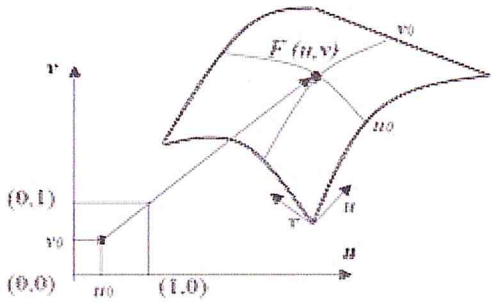


Figure 14. Localisation d'un point sur une paramétrique.

Les surfaces paramétriques ne sont pas utilisées séparément, mais plusieurs morceaux de surfaces sont joints ensemble pour obtenir une surface plus complexe, Les surfaces Bézier, Bézier-rationnelle, B-SPline et NURBS sont très utilisées en CFAO/CAO (Figure 15) [11].

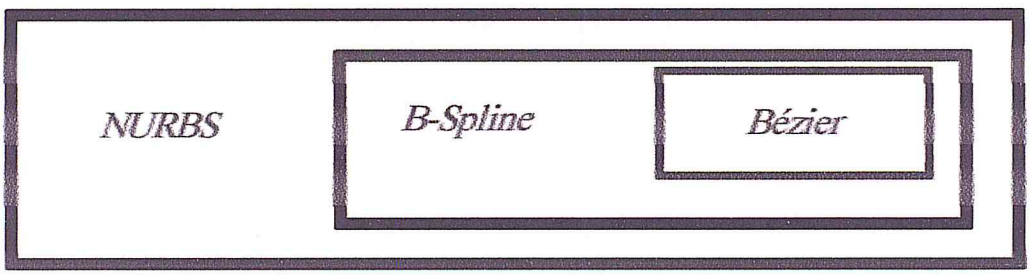


Figure 15. Relation entre les différentes représentations.

4. Généralités sur la CFAO

4.1. Introduction

Pour aider les métiers de la conception et de la fabrication de produits manufacturiers, des logiciels de Conception Assistée par Ordinateur (CAO) et de Fabrication Assistée par Ordinateur (FAO) ont été développés. La CAO et la FAO correspondent aux activités de deux processus complémentaires permettant de traduire une idée en un objet réel [9].

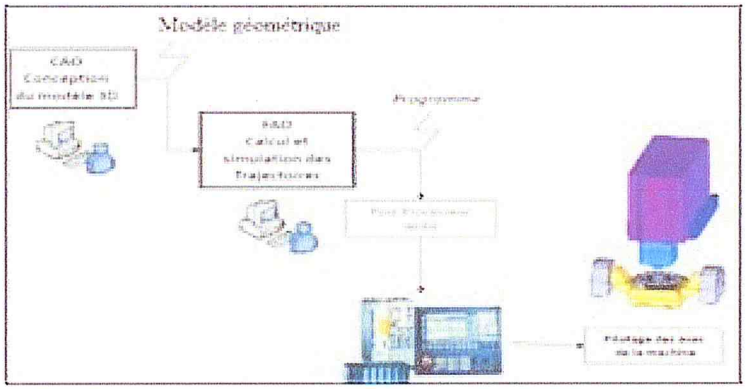


Figure 16. Composants de CFAO. 4.2.

Conception assistée par ordinateur (CAO)

4.2.1. Définition :

La conception assistée par ordinateur « CAO » rassemble des outils informatiques (logiciels et matériels) permettant la modélisation géométrique d'un objet 3D. La conception d'un produit est basée sur la modélisation géométrique.

4.2.2. Modélisation géométrique

Trois catégories de modèles géométriques peuvent être utilisées pour décrire les surfaces frontières d'une pièce ou d'un objet au cours de sa conception.

- **Première catégorie :** appelée modèles paramétriques. Ces modèles permettent la création d'objets par un langage et des algorithmes paramétrés.
- **Deuxième catégorie :** appelée modèles polyédriques. Les facettes sont les seules entités de surface à visualiser par les environnements graphiques (Figure 17) [16-17].

a. Polyèdre pour la visualisation. b. Polyèdre pour l'analyse de structure.

Figure 17. Exemple de la géométrie polyédrique.

➢ **Troisième catégorie :** appelée modèles implicites. Les surfaces implicites, utilisées initialement dans le cas de primitives volumiques simples, restent pour le moment un modèle de préférence de l'animation (Figure 18) [12].

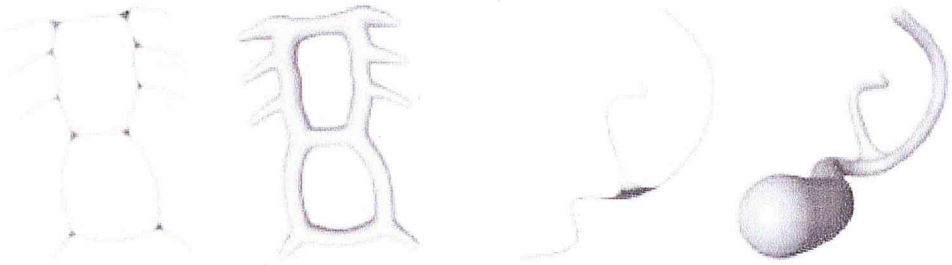


Figure 18. Exemples d'objets décrits par des modèles implicites.

5. Fabrication assistée par ordinateur (FAO)

Au sens strict, la fabrication assistée par ordinateur (FAO) désigne les logiciels d'assistance à la programmation des machines-outils à commande numérique [9].

5.1 Objectif de l'intégration CAO/FAO

En CFAO, l'intégration consiste à associer des applications hétérogènes (CAO, Calcul, FAO, etc.) dans un même environnement, avec l'objectif de favoriser l'action simultanée ou concourante de ces applications au cours du processus global et de minimiser la traduction de données qui constituent le facteur principal de perte de temps et de sémantique [9].

6. Format d'échange de données

Les erreurs de conversion ou les pertes d'informations lors d'un transfert sont fréquentes. Ils existent deux grands types de solutions :

- Utiliser un logiciel de conversion : permet de convertir un fichier natif du logiciel A vers un logiciel B, sans perte d'information mais très dépendant de la version des logiciels, non universel et solution généralement couteuse.
- Utiliser un format d'échange standard (ACIS, STEP, IGES, STL, etc.).

Dans ce travail, le format STL est considéré.

6.1. Format d'échange de données « STL »

C'est une abréviation du mot STereo Litography . Un fichier STL stocke des informations sur les modèles 3D. Ce format décrit uniquement la surface externe d'un objet, c'est-à-dire des modèles sous formes triangulaires contigües, dont chaque triangle est défini de manière unique par sa normale unitaire sortante et les coordonnées X, Y et Z de ses trois sommets (Figure 19). Chaque triangle doit partager deux sommets avec chacun des triangles le juxtaposant. Il ne

représente pas la couleur, la texture ou autres attributs de l'objet. Il est généralement généré par un programme de conception assistée par ordinateur (CAO).

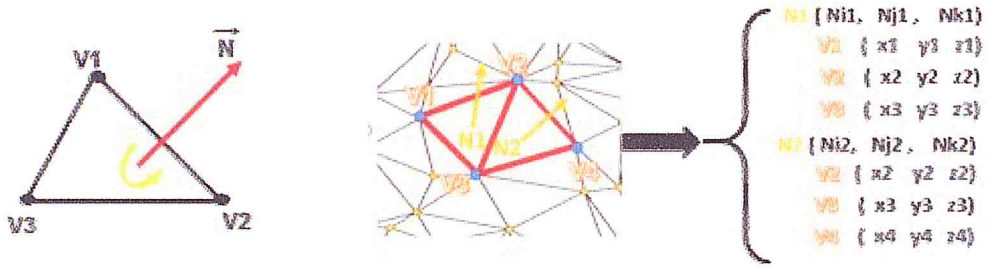


Figure 19. Paramètres d'un triangle.

6.2. Pourquoi STL ?

Les données STL permettent de créer un objet en déposant une succession de calques constituées de plastiques, métaux ou matériaux composés. Les pièces ou modèles obtenus sont généralement utilisés dans les cas suivants : visualiser les conceptions, créer des prototypes de produits, tester les formes, identifier les problèmes de conception, etc. [13].

6.3. Différents types du format « STL »

- Un fichier binaire, moins de taille mais non lisible et trop condensé (Figure 20).
- Un fichier ASCII lisible mais gros de taille. Il commence par une ligne de description de (nom, nom de l'auteur…) précédée par le mot réservé « **Solid** » suivie des composantes de la normale et des coordonnées des sommets des triangles. Il se termine par « **end solid** ».

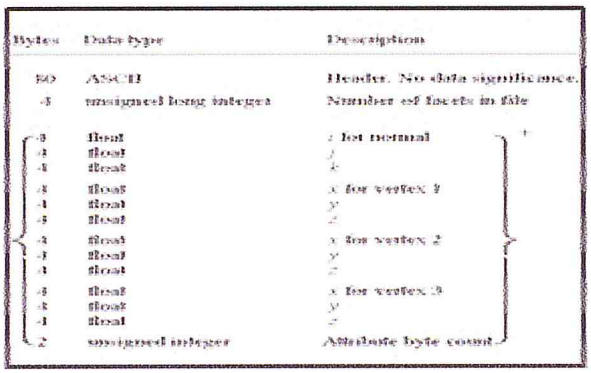


Figure 20. Structure d'un fichier STL binaire.

6.4. Perspectives du format STL

Avantages : le format STL est devenu un standard, il est très courant et utilisé par l'ensemble des logiciels 3D, de nombreuses bibliothèques de fichiers 3D STL existent. **Inconvénients :** le format STL décrit seulement la géométrie, les autres informations sur l'objet telles que la couleur, matière ne sont pas décrites.

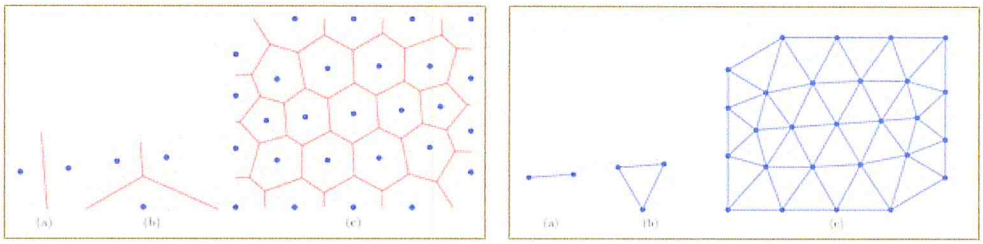
Partie II:
Méthode de Flip

1. Triangulation de Delaunay

La notion de triangulation de Delaunay a été proposée en 1934 par un mathématicien russe Boris Delaunay l'élève de Georgy Voronoï [13].

1.1. Définition

On appelle triangulation de Delaunay de l'ensemble P, et on note Del(P), le dual du diagramme de Voronoï de P : les sommets de la triangulation de Delaunay sont les points pi ∈ P, et deux sommets sont reliés par une arrête dans la triangulation si les cellules de Voronoï correspondantes sont voisines. Les sommets sont des objets, qui à eux tous constituent une partition de l'enveloppe convexe de ces objets. La Figure 21.a donne des exemples de diagramme de Voronoï et la Figure 21.b représente la Triangulation de Delaunay.

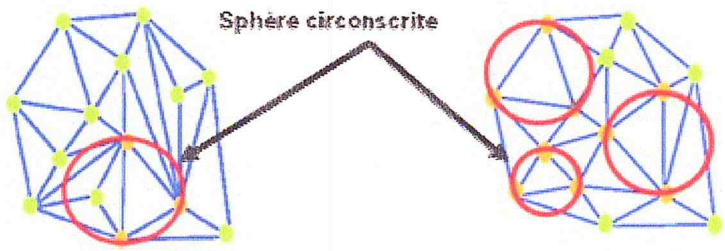


a.: Diagramme de Voronoï. b. Triangulation de Delaunay.

Figure 21. Triangulation de Delaunay duales des diagrammes de Voronoï.

1.2. Propriétés de la triangulation de Delaunay

Propriété 1 : chaque triangle est entouré d'un cercle vide qui passe par les sommets du triangle et ne contient aucun autre site ou sommet à l'intérieur.



a. Non-Delaunay triangulation b. Delaunay triangulation.

Figure 22. Triangulation de Delaunay.

Propriété 2 : les sommets et les côtés ouverts de l'ensemble des triangles de Delaunay sont deux à deux disjoints.

Propriété 3 : la triangulation de Delaunay maximise le minimum des angles des triangles (Figure 23).

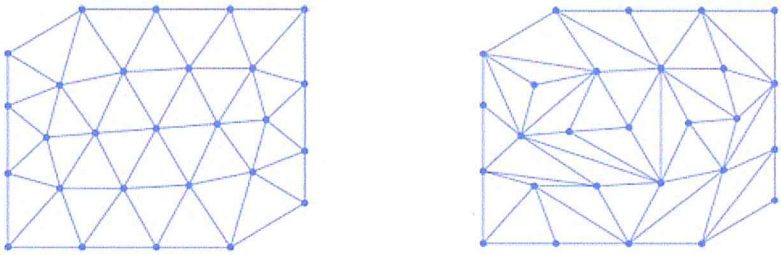


Figure 23. Propriété angulaire.

Propriété 4 : soient pi et pj deux points de P. Alors le segment pipj est une arrête de la triangulation de Delaunay de P si et seulement s'il existe un cercle passant par pi et pj tel que le disque correspondant ne contient aucun autre point de P (Figure 24).

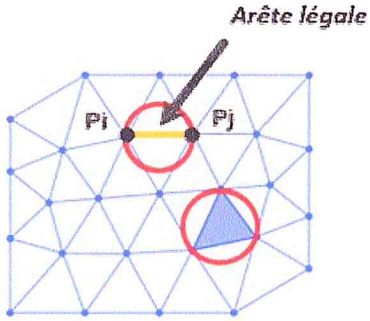


Figure 24. Illustration de la caractérisation arête légale.

En trois dimensions, une triangulation 3D de Delaunay est une tétraédrisation de l'ensemble des points (Figure 25). Dans ce cas, le nuage de points défini la surface d'un objet ainsi que son intérieur, chaque tétraèdre possède une sphère circonscrite vide (Figure 26). Ces dernières ne se chevauchent jamais et chaque tétraèdre est adjacent à quatre tétraèdres sauf ceux qui sont situés sur la frontière de l'ensemble de points [13].

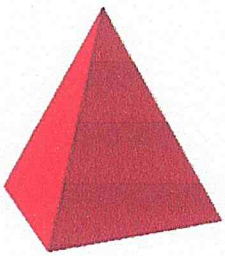


Figure 25. Tétraèdre.

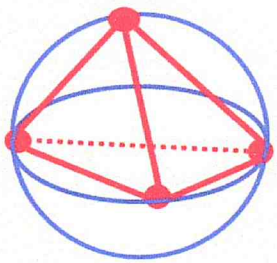


Figure 26. Sphère circonscrite.

1.3. Méthodes basées sur la triangulation de Delaunay

Ils existent plusieurs algorithmes qui génèrent, de façon incrémentale, la triangulation de Delaunay. On parle de techniques incrémentales lorsque des triangles sont recalculés au fur et à mesure que des points sont ajoutés à une triangulation de départ. Dans la suite, les algorithmes les plus performants sont présentés.

1.3.1. Triangulation par Destruction/Construction

Elle se fait à partir d'une triangulation partielle du semis, et se poursuit par la triangulation existante augmentée d'un point. Donc, partant de quatre points de départ formant deux triangles (donc quatre points déjà triangulés), l'algorithme rajoute des points qui sont intérieurs à la triangulation et qui sont triangulés au fur et à mesure. On procède ainsi : on utilise un triangle et un point choisi, puis tous les triangles dont le cercle circonscrit contient le point considéré sont détruits. Une fois les triangles illégaux détruits, on construit ceux qui sont engendrés par ce nouveau point, en ajoutant des arêtes reliant au point inséré les extrémités de celles qui n'appartenaient qu'à un seul triangle présent dans la liste des triangles supprimés [14].

1.3.2. Algorithme Crust

Permet de reconstruire des surfaces. La méthode s'appuie sur la triangulation de Delaunay et le diagramme de Voronoï. Le Crust se compose des 4 étapes suivantes :

- Construction de la triangulation de Delaunay sur le nuage de points.
- Calcul des pôles des cellules de Voronoï.
- Construire la triangulation de Delaunay de l'union des points du nuage et des pôles définis dans l'étape précédente.
- Garder uniquement les triangles dont les trois sommets sont des points de l'échantillon de départ.

1.3.3. Algorithme DeWall

L'algorithme DeWall est basé sur l'approche « diviser pour régner ». La démarche de cet algorithme consiste en: 1) la subdivision du nuage, 2) triangulation (TD) des sous nuages et 3) concaténation de ces triangulations [15].

1.3.4. Algorithme de Flip

Si dans une triangulation quelconque, on remplace localement un côté illégal par un côté légal, on obtient une triangulation dont la suite d'angles est lexico-graphiquement strictement supérieure. Il en résulte que, lorsqu'on supprime successivement les côtés illégaux d'une triangulation, on obtient une suite de triangulations deux à deux distinctes. Comme le nombre total de triangulations possibles d'un ensemble S donné est fini, on obtient, après un nombre fini d'échanges, une triangulation qui n'a plus de côté illégal. Cette triangulation est la triangulation de Delaunay. On en déduit un algorithme (dit de « Flip ») qui permet de

transformer une triangulation quelconque T en triangulation de Delaunay par une suite de modifications locales.

1.4. Complexité et performance

Un problème (un algorithme) nécessite des ressources pour être traité : on parle alors de complexité. La complexité s'exprime généralement en fonction du nombre de données à traiter mais il arrive parfois qu'elle s'exprime aussi en fonction du nombre de solutions au problème. Ils existent deux types de complexité :

- **La complexité en temps** : donne le nombre d'opérations élémentaires nécessaire pour traiter les n données.
- **La complexité en place** : donne la place mémoire nécessaire pour traiter les n données.

L'algorithme '**Destruction-Construction**' est plus facile à aborder, mais de complexité assez importante en O (n^3) [15]. L'algorithme '**Dewall**' se réduit à un algorithme incrémental de complexité O ($(^d/_2)+1$) [16]. Dans ce travail, l'algorithme de '**Flip**' est adopté dont la complexité est O(n*log(n)).

1.5. Méthode de Flip

Cet algorithme est plus rapide par rapport aux autres algorithmes précédents. C'est une grande fonction récursive qui peut être implémentée en itératif et qui vérifie si les côtés sont légaux. Un côté est défini comme étant l'arête commune à deux triangles. Pour savoir s'il est illégal, il faut regarder les points qui ne sont présents que dans un des triangles, Un côté est considéré comme étant illégal si au moins un de ces deux points est contenu dans le cercle circonscrit du triangle opposé (Figure 27).

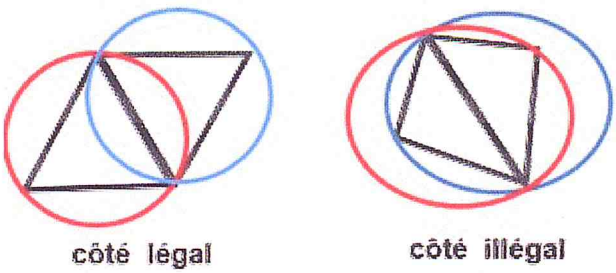


Figure 27. Côté légal et illégal.

Si c'est le cas, le côté illégal est remplacé par un côté relié par les deux points opposés. Et on testera les quatre côtés qui restent pour voir si ce changement a provoqué un autre conflit. L'image qui suit montre un côté illégal et le Flip qui permet de rendre ce côté légal.

Donc l'algorithme peut être décrit par :

- Insérer le point, détruire le triangle le contenant.
- Créer trois triangles qui ont ce nouveau point comme sommet.
- Vérifier chaque côté externe pour une illégalité [14].

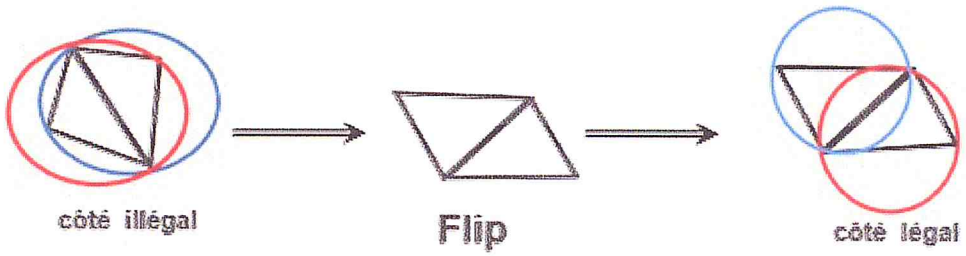


Figure 28. Transformation utilisant Flip.

Conclusion

L'étude effectuée dans ce chapitre portait sur une méthode de construction inverse dite « Ingénierie Inverse » qui a pour objectif de reconstruire un objet en donnant une représentation numérique (mathématique) à un modèle physique existant. Une présentation des phases de la Rétro Conception a été introduite et l'accent a été mis sur le processus de reconstruction d'objet 3D à partir de nuages de points et les différents formats d'échange de données entre la CAO et la FAO ont été présentés en particulier le format STL.

Ils existent plusieurs algorithmes qui génèrent, de façon incrémentale, la triangulation de Delaunay. Parmi les algorithmes existant, il y'a ceux qui sont réservés à la triangulation dans le plan, et ceux qui destinés à la triangulation 3D. La méthode de 'Flip' qui modifie localement les triangles d'un maillage pour obtenir une triangulation de Delaunay est la plus indiquée pour la reconstruction d'objets 3D.

Le prochain chapitre est réservé à la conception de notre application.

Chapitre II

Conception de l'application

Introduction :

Afin de copier un produit existant sans disposer de dessins, de documentation ou de modèles informatiques (Modèle CAO) on utilise le « Reverse Engineering ». Cette technique est introduite dans plusieurs domaines tels que le génie mécanique, la vision artificielle, la robotique, la médecine, ...etc. Pour faciliter ce processus, il est préférable de représenter ce produit par un nuage de points et le nuage de points par des formes géométriques élémentaires (triangles, tétraèdres, ...etc.). Par la suite, plusieurs algorithmes sont utilisés pour construire des surfaces approchant le mieux la surface physique de l'objet. Enfin, cette surface peut être représentée sous un espace 2D ou un espace 3D.

Le but recherché est de produire en sortie une surface STL approchant au mieux la forme de l'objet physique et d'une façon optimale. Pour cela, la méthode de Flip en 3D appliquée à la Triangulation de Delaunay a été utilisée dans ce projet. Cette méthode a été adoptée au vu de ses propriétés géométriques remarquables.

1. Problématique :

Les objets ou les surfaces de formes complexes sont utilisés dans la conception, et la réalisation des pièces mécaniques, médicales, robotique...etc. Ces modèles 3D peuvent être obtenus par deux méthodes : la première méthode est basée sur l'utilisation des modèles CAO **« construction »** et la deuxième méthode utilise la technique de reconstruction des modèles 3D **« Reverse Engineering ».** Les étapes et les détails de ces deux méthodes sont présentés dans le chapitre précédent. Cependant, la méthode de reconstruction d'objet qui repose sur la digitalisation de l'objet, reste une tâche très compliquée et couteuse en temps à cause de la taille volumineuse des nuages de points et du traitement effectué sur les point pour triangulation.

L'objectif assigné à notre travail, se résume comme suit : étant donné un nuage de points 3D représentant la peau d'un objet, il s'agit de proposer une méthode de triangulation 3D permettant une localisation rapide et efficace des tétraèdres sur lesquels des Flips et des Splits doivent être appliqués à même de respecter les critères de la triangulation de Delaunay et d'optimiser le temps de traitement et par conséquent les coûts de réalisation. Le but recherché est de reproduire en sortie une surface approchant au mieux la forme de la surface physique de l'objet échantillonné. Pour atteindre cet objectif, le processus de reconstruction

(Figure 1) résume l'approche adoptée.

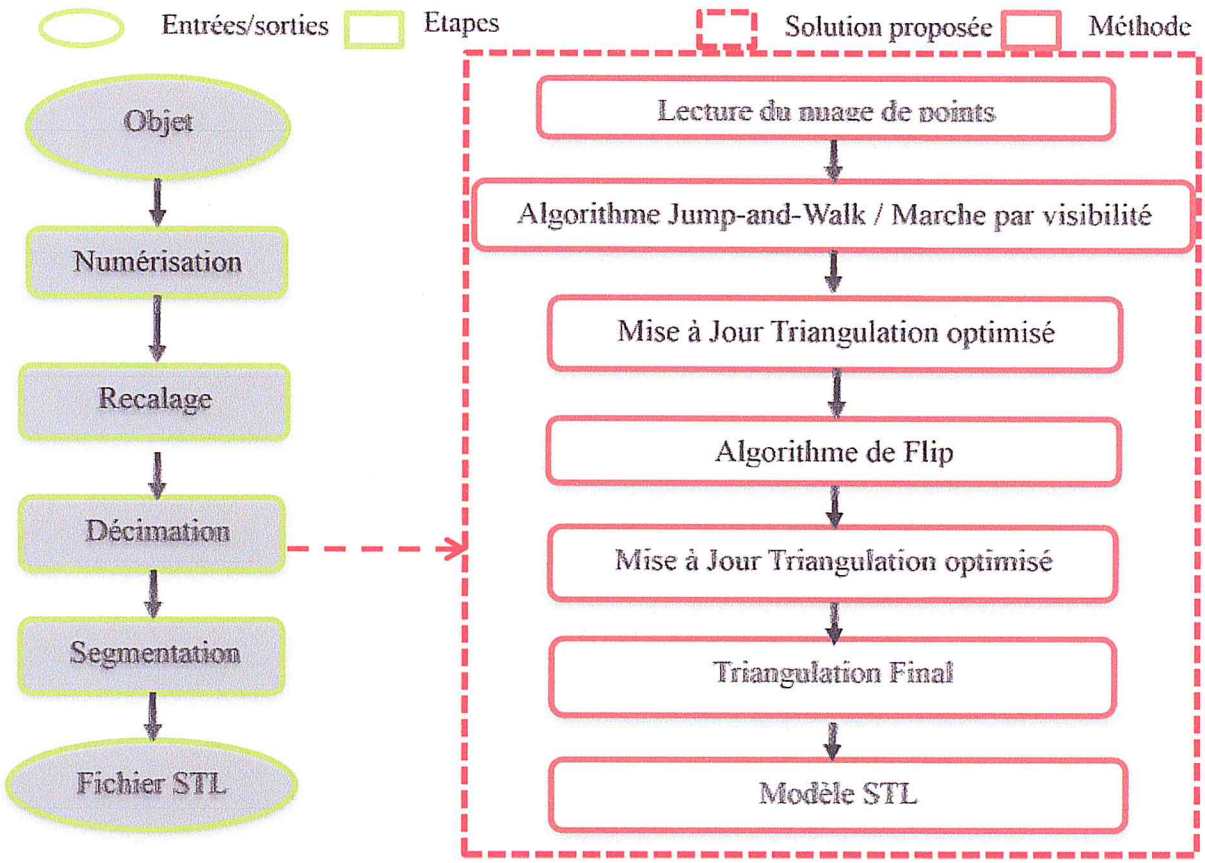


Figure 29. Approche adoptée pour la reconstruction de surfaces en 3D.

2. Architecture globale de l'application :

L'architecture générale de l'application proposée est constituée des quatre (04) étapes suivantes :

- Développement des algorithmes d'approximation du nuage de points par des tétraèdres en utilisant la triangulation de Delaunay 3D.
- Proposition d'une méthode permettant la localisation des tétraèdres à modifier en un temps optimal.
- Optimisation de la zone de mise à jour de la triangulation.
- Génération du modèle STL et du fichier STL d'un objet digitalisé.

L'architecture globale est illustrée par l'organigramme suivant :

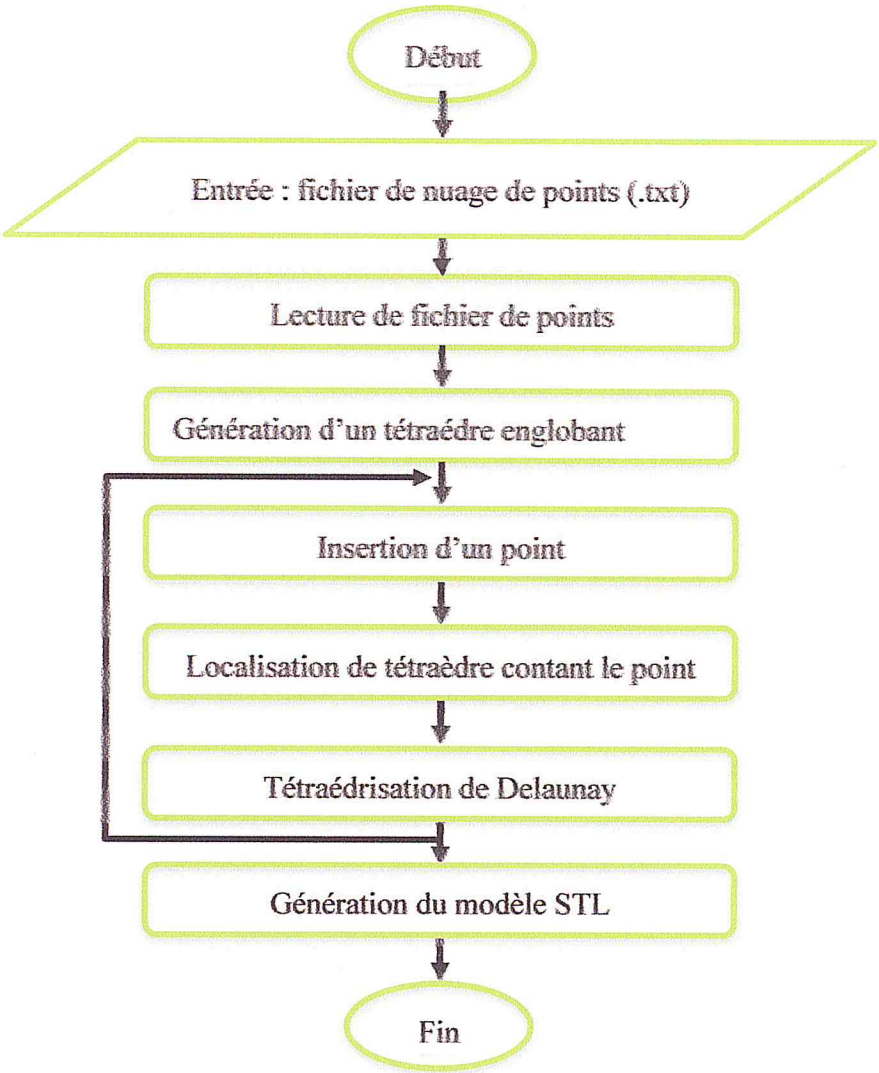


Figure 30.Organigramme de l'Architecture globale.

2.1. Lecture du fichier de points

La lecture du fichier consiste à vérifier la conformité syntaxique du fichier et à structurer les coordonnées de chaque point du nuage. Cette conformité est une représentation unique et correcte de l'objet. La procédure de la lecture du nuage de point est illustrée par la procédure suivante :

1. Sélection du fichier texte.
2. Récupération des coordonnées (x, y, z) des points du nuage à partir du fichier.
3. Stockage de chaque point dans une classe créer à cet effet « Point_Delaunay_Optim ». Cette classe est représentée par champs double (x, y, z) et de type booléen « point Fictif ».
4. Enregistrement des points dans un tableau « tableau_pts » de type « Point_Delaunay_Optim ». Les types booléens des points du fichier sont mentionnés des points non fictifs.

5. Calcul des limites du brut en utilisant une classe « Brut » (Figure 31) :

- Calcul des coordonnées du brut de P_{min} (x_{min}, Y_{min}, Z_{min}) et P_{max} (x_{max}, Y_{max}, Z_{max}) à partir des coordonnées des points du nuage.
- Calcul des dimensions du brut à savoir la hauteur, la longueur et la largeur :

Hauteur = z_{max}-Z_{min}.

Longueur = x_{max}-x_{min}.

Largeur = y_{max}-y_{min}.

- Visualisation du nuage de points. (Dessiner le brut, le nuage de points)

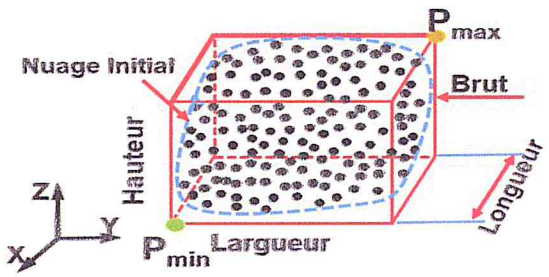


Figure 11. Limites du brut.

L'organigramme de lecture du fichier de points est illustré ci-dessous :

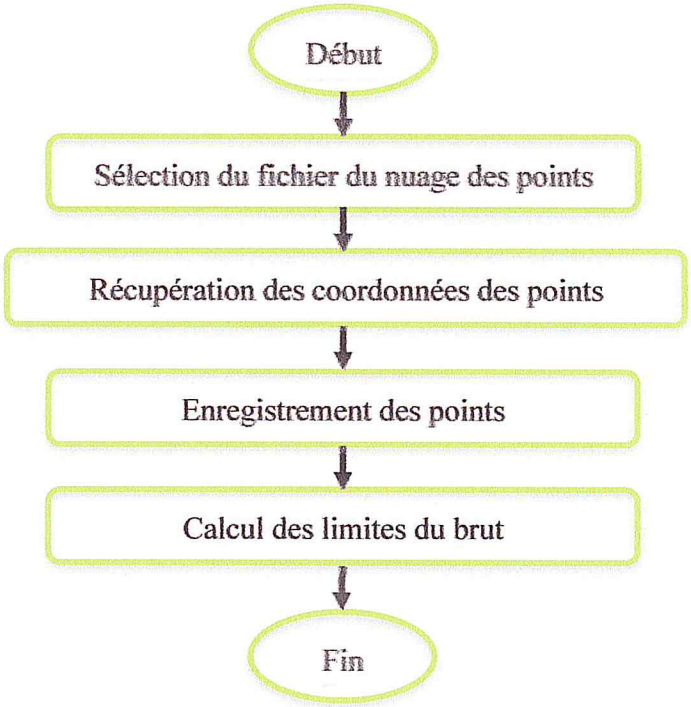


Figure 32. Organigramme de lecture du nuage de points.

2.2. Génération d'un tétraèdre englobant :

La tétraédrisation à partir d'un tétraèdre englobant est faite en cinq (05) étapes :

1. Calcul du centre du brut (Figure 33)

$$Xcentre = (Xmax + Xmin)/2$$

$$Ycentre = (Ymax + Ymin)/2$$

$$Zcentre = (Zmax + Zmin)/2$$

2. Calcul du Rayon de la sphère définie par la distance maximale entre le centre de la sphère et les différents points d'extrémités du brut (X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{min}, Z_{max}) (Figure 33).

$$\boldsymbol{Rayon} = \sqrt{(\boldsymbol{Xmax} - \boldsymbol{Xcentre})2 + (\boldsymbol{Ymax} - \boldsymbol{Ycentre})2 + (\boldsymbol{Zmax} - \boldsymbol{Zcentre})2}$$

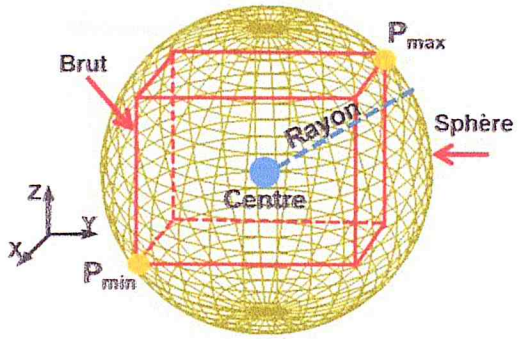


Figure 33. Rayon de la sphère.

3. Génération des quatre points sur la sphère permettant de définir les plans tangents en utilisant les coordonnées sphériques.
4. Génération des quatre plans tangents sur la sphère passant par les points définis précédemment.
5. Création du tétraèdre englobant à partir des points qui sont obtenus par la méthode du Pivot de Gauss en faisant l'intersection entre les différents plans pour définir les sommets (sommets fictifs).
6. Calcul du centre et du rayon de la sphère circonscrite du tétraèdre englobant.
7. Calcul des normales du tétraèdre englobant.

2.2.1. Caractéristiques d'un tétraèdre :

Chaque tétraèdre est caractérisé par :

1. Quatre (04) points (P_1, P_2, P_3, P_4) (Figure 34).
2. Quatre (04) normales dirigées vers l'extérieur (normalisée) (Figure 35)
3. Une sphère englobante (Figure 36).
4. Le centre et le rayon de la sphère (Figure 36).
5. Entre un (01) et quatre (04) voisins (Figure 37).
6. Les coordonnées limites d'extrémités de sa boite englobante (X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{min}, Z_{max}) (Figure 10).

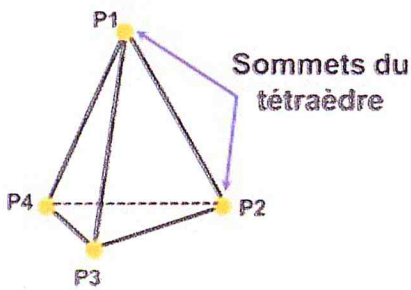


Figure 34. Sommets d'un tétraèdre.

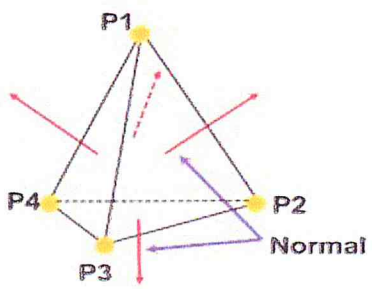


Figure 35. Normales d'un tétraèdre.

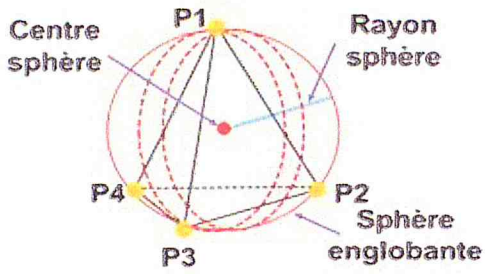


Figure 36. Rayon et centre de la sphère circonscrite.

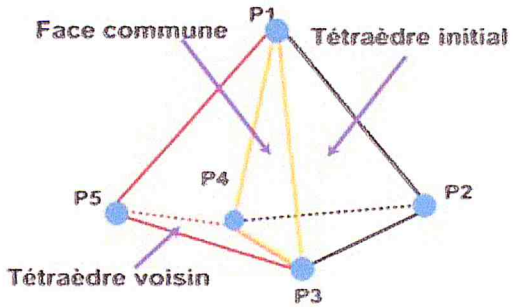


Figure 37. Tétraèdres voisins.

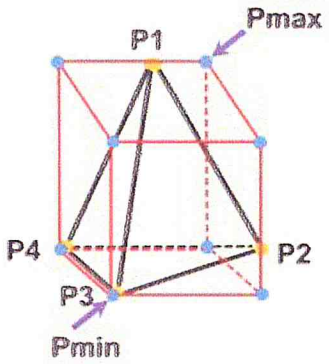
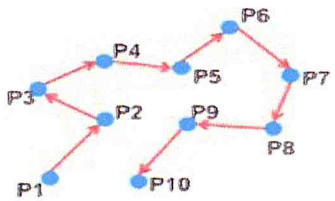


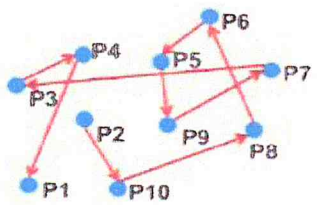
Figure 38. Représentation du brut englobant.

2.3. Insertion d'un point :

Il y'a plusieurs techniques pour l'insertion des points. Dans ce travail, l'algorithme incrémental qui permet l'insertion des points un par un soit d'une manière aléatoire ou séquentielle a été utilisé (Figure 39).



a. Séquentiel.



b. Aléatoire.

Figure 39. Modes d'insertion des points.

2.4. Localisation du tétraèdre contenant le point inséré

L'un des problèmes les plus étudiés dès le début de la géométrie algorithmique est celui du positionnement rapide des points dans la subdivison pour avoir une triangulation de Delaunay. La première phase de l'optimisation de la triangulation de Delaunay commence par la localisation du tétraèdre contenant le point inséré. La performance moyenne de la localisation du point ne peut pas être facilement estimée car elle dépend fortement du choix rapide du tétraèdre de départ et le plus court parcours à suivre pour arriver finalement au tétraèdre

contenant le point inséré. Pour cela, l'organigramme (Figure 40) propose une méthode pour l'implémentation rapide afin d'optimiser le temps de recherche du point inséré.

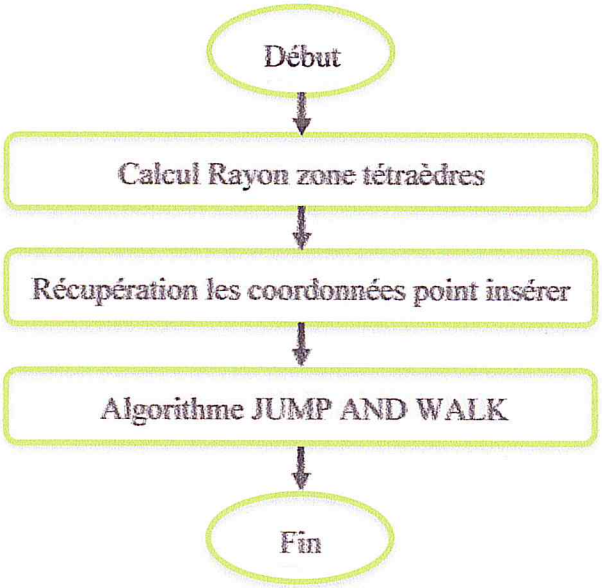


Figure 40. Organigramme de localisation du point inséré.

2.4.1. Calcul du rayon zone tétraèdres

Afin de déterminer le rayon d'une façon judicieuse, les limites du brut sont utilisées :

Rayon sphère=max (largeur, max (hauteur, longueur)) ;

2.4.2. Récupération des coordonnées du point inséré

Dans cette phase, les coordonnées du point inséré (x, y, z) sont récupérées afin de les utiliser dans l'étape qui suit pour déterminer le tétraèdre qui contient ce dernier.

2.4.3. Algorithme JUMP AND WALK:

Pour sélectionner un tétraèdre de départ l'algorithme de jump and walk a été mis en œuvre(le calcul de la distance 'd'est fait par rapport à chaque voisins) :

- La première étape permet de sélectionner un tétraèdre de départ .Les étapes de cette méthode sont :
  - Choix d'un Rayon de la sphère de manière judicieuse.
  - Calcul des limites de la sphère (X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{min}, Z_{max}) en utilisant le rayon et les coordonnées de points de départ (P_X, P_Y, P_Z).

$X_{min} = P_X$ - rayon; $X_{max} = P_X$ + rayon;

$Y_{min} = P_Y$ - rayon; $Y_{max} = P_Y$ + rayon; Z_{min}

$= P_Z$ -rayon ; $Z_{max} = P_Z$ + rayon.

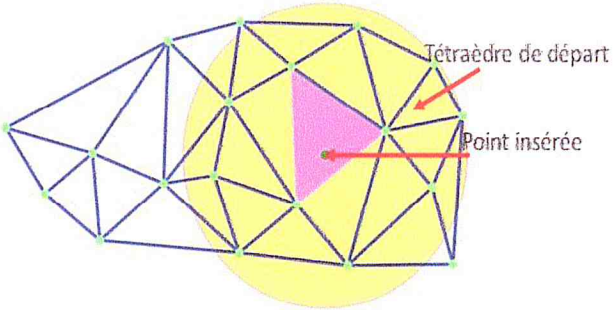


Figure 42. Jump And Walk.

➢ La deuxième étape est la **Marche par visibilité** est une recherche itérative. Une fois le rayon de la sphère introduit, une recherche du premier tétraèdre contenu dans la sphère inscrite est identifiée. Par la suite, le produit scalaire du vecteur allant du centre du tétraèdre de départ au point inséré, par le vecteur allant du centre du premier tétraèdre aux vecteurs normaux du premier tétraèdre contenu dans la sphère inscrite est calculé. Selon le signe des quatre produits scalaires, le tétraèdre contenant le point inséré est identifié. L'algorithme est ensuite répété jusqu'à ce que le tétraèdre contenant le point recherché soit trouvé ou qu'une

face limite soit atteinte (Figure 42)

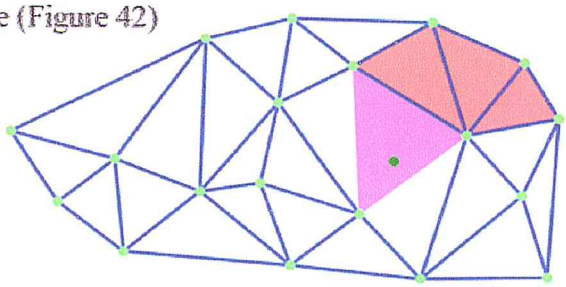


Figure 32. Marche par visibilité.

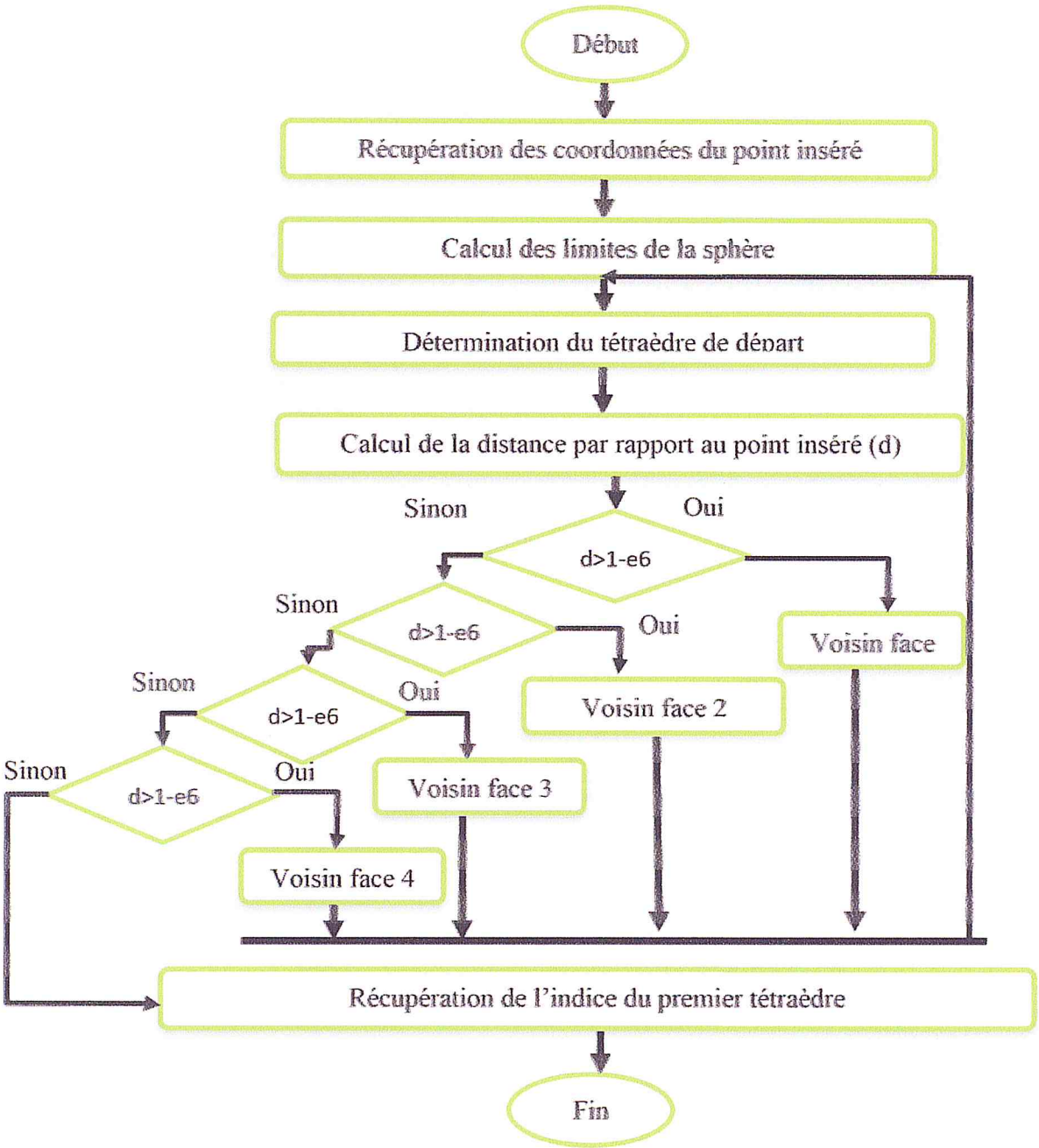


Figure 43. Organigramme Jump And Walk.

2.5. Tétraédrisation de Delaunay

La triangulation de Delaunay est importante dans le domaine de modélisation géométrique mais aussi dans l'analyse de données. Elle consiste à construire des triangules les moins plats possible à partir d'un ensemble de points. La triangulation de Delaunay d'un ensemble de n points est l'unique triangulation telle que les cercles circonscrits de chaque triangle ne contiennent aucun autre point de l'ensemble. Cette notion peut être généralisée à n'importe quelle dimension. En 3D, les tétraèdres et les sphères remplacent les triangles et les cercles circonscrits. La triangulation de Delaunay 3D consiste à générer des tétraèdres à partir du nuage des points. La procédure de la tétraédrisation est donnée par la Figure 44).

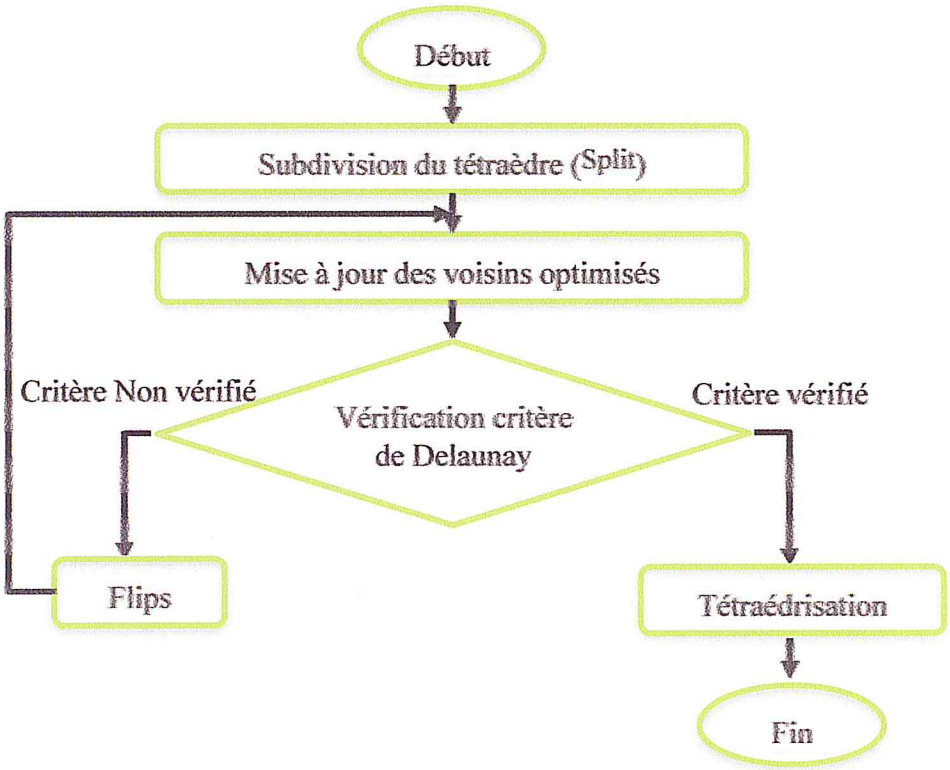


Figure 44. Organigramme de la triangulation de Delaunay.

2.5.1. Subdivision du Tétraèdre (Split)

Après l'insertion du point et après l'identification du tétraèdre inclus dans la sphère circonscrite, la procédure suivante est appliquée :

- Récupérer et stocker les quatre (4) voisins du tétraèdre qui contient le point inséré. Par la suite, on appellera ce tétraèdre le tétraèdre en question.
- Remplacer les coordonnées du point et la normale dans l'équation du plan d'une face, afin de récupérer les quatre plans.

➢ Déterminer la position du point inséré selon les quatre cas qui se présentent par la suite:

- Les quatre produits scalaires négatifs, le point est à l'intérieur du tétraèdre.
- Trois produits scalaires négatifs et un positif, le point est situé sur une face.
- Deux produits scalaires négatifs et deux positifs, le point est situé sur une arrête.
- Un produit scalaire négatif et trois positifs, le point est un sommet.

Le processus de subdivision se fait comme suit (Figure 45).

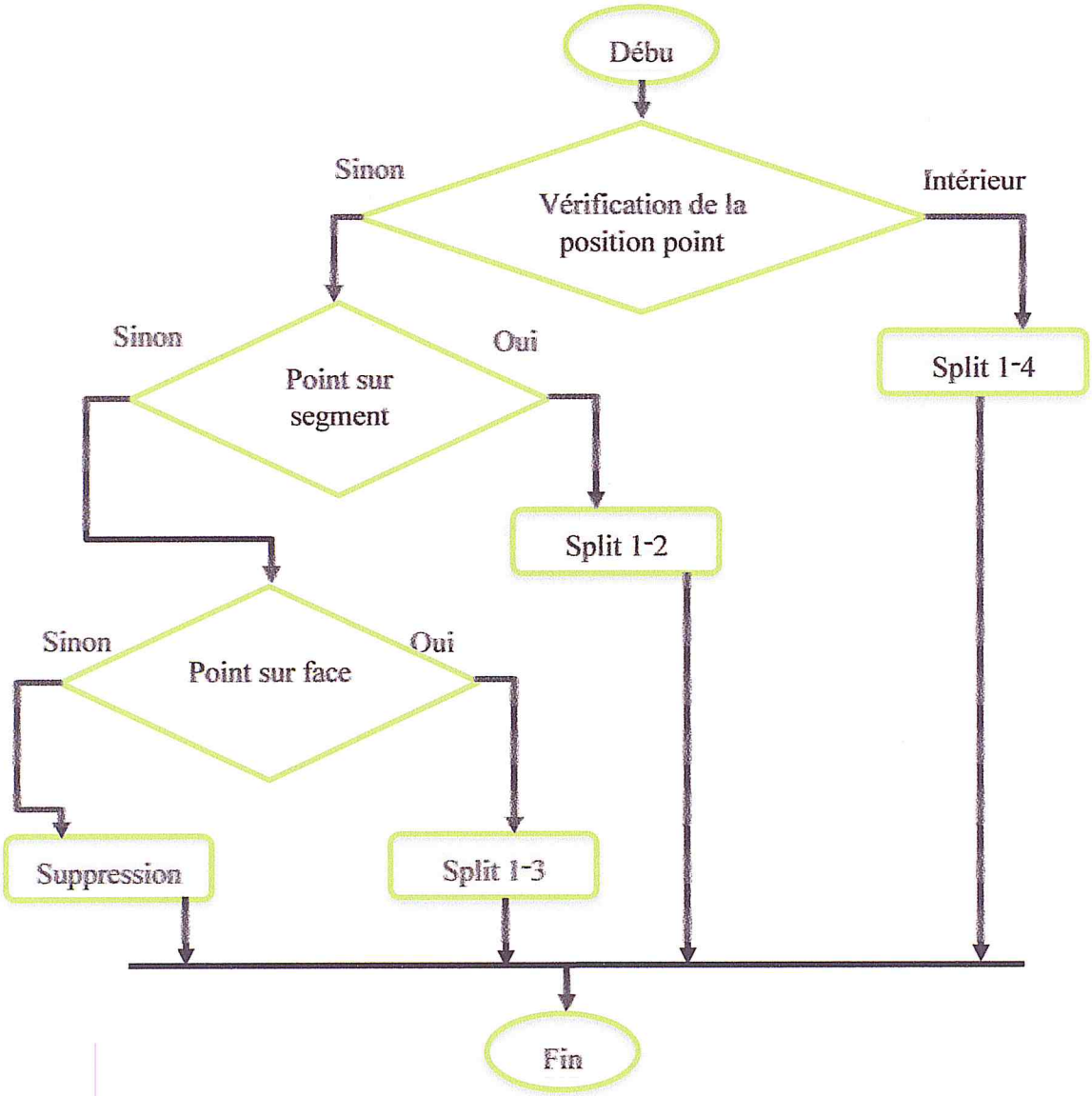


Figure 45. Subdivision du tétraèdre.

2.5.1.1. Point inséré est confondu avec un des sommets du tétraèdre

Dans ce cas, le point inséré n'est pas pris en considération, il est donc supprimé.

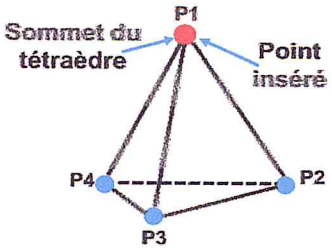


Figure 46. Point inséré sur le sommet.

2.5.1.2. Point inséré est à l'intérieur du tétraèdre

Si le point inséré est à l'intérieur du tétraèdre, les étapes effectuées sont (Figure 47) :

1. Stocker l'indice du tétraèdre dans les deux tableaux « Tableau voisin » et « Tableau vérifier ».
2. Subdiviser le tétraèdre en quatre tétraèdres (split 1-4).
3. Remplacer le tétraèdre initial.
4. Stocker les trois derniers dans les deux tableaux « Tableau voisin » et « Tableau vérifier ».
5. Déterminer les voisins pour tous les nouveaux tétraèdres.
6. Calculer la zone voisine toute en récupérant les voisins des voisins et les stocker dans un tableau zone afin de minimiser la zone des tétraèdres à modifier.
7. Mettre à jour les voisins.

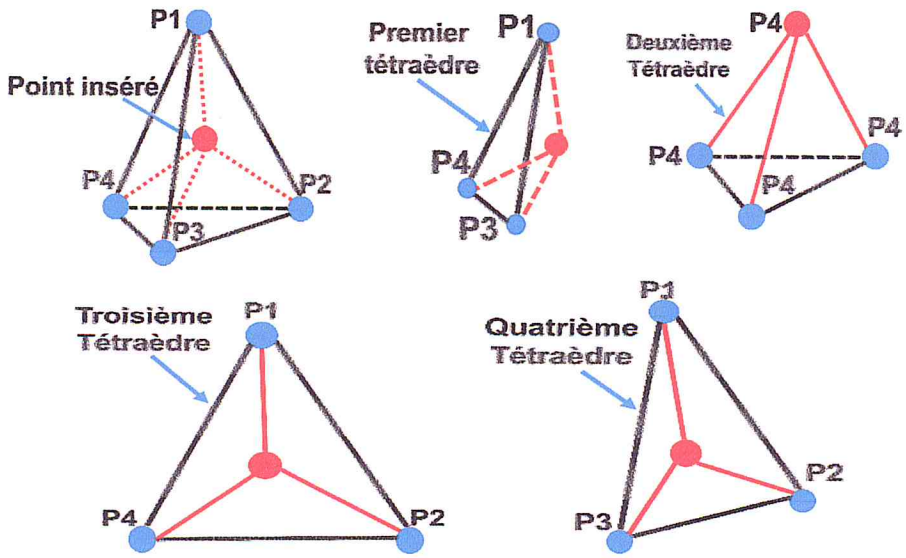


Figure 47. Split d'un point à l'intérieur du tétraèdre.

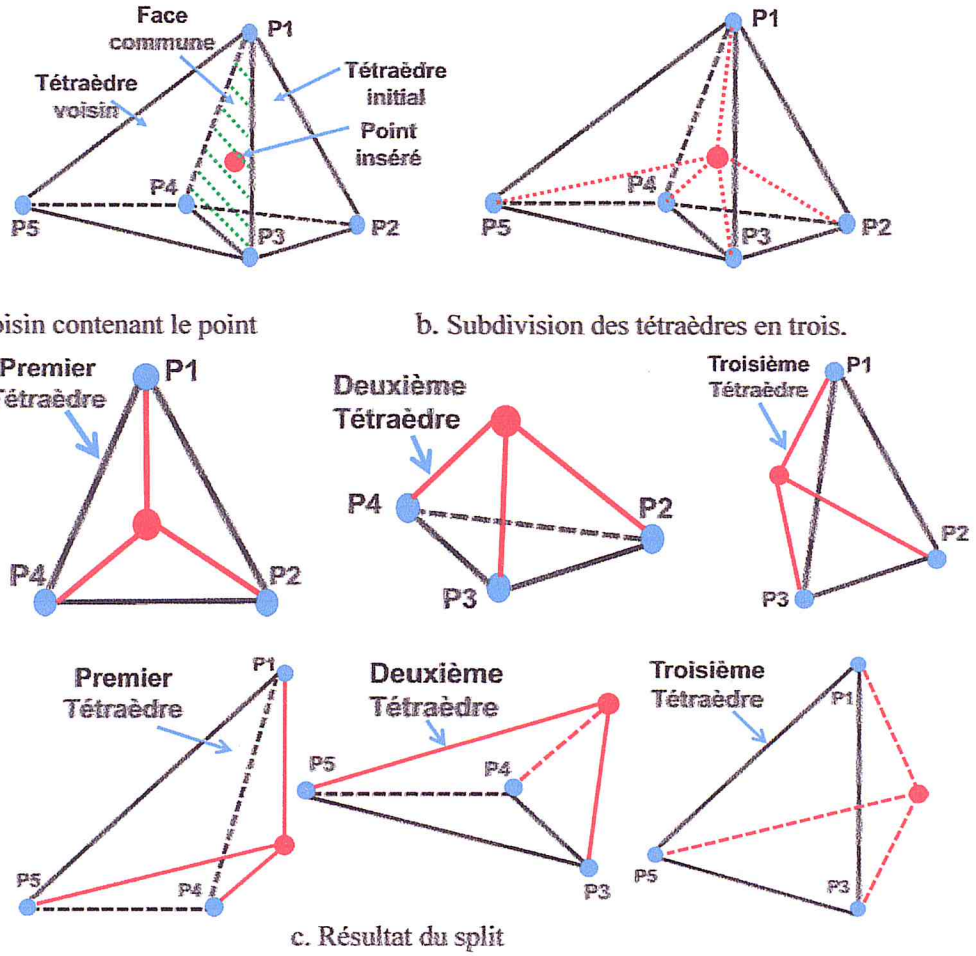
2.5.1.3. Point inséré est sur une face du tétraèdre

Si le point inséré se trouve sur une face du tétraèdre, le processus suivant est appliqué:

1. Stocker l'indice du tétraèdre dans les tableaux « Tableau voisin» et «Tableau vérifier».
2. Identifier le voisin commun à cette face.

3. Subdiviser le tétraèdre voisin en créant trois nouveaux tétraèdres (split 1-3).
4. Remplacer le tétraèdre voisin.
5. Subdiviser le tétraèdre initial en créant trois nouveaux tétraèdres.
6. Remplacer le tétraèdre initial.
7. Stocker tous les nouveaux tétraèdres dans les deux tableaux «Tableau voisin» et «Tableau vérifier».
8. Déterminer les voisins pour tous les nouveaux tétraèdres.
9. Calculer les voisins.
10. Mettre à jour les voisins.

Ce processus est illustré par la figure suivante (Figure 48) .



a. Voisin contenant le point b. Subdivision des tétraèdres en trois.

c. Résultat du split

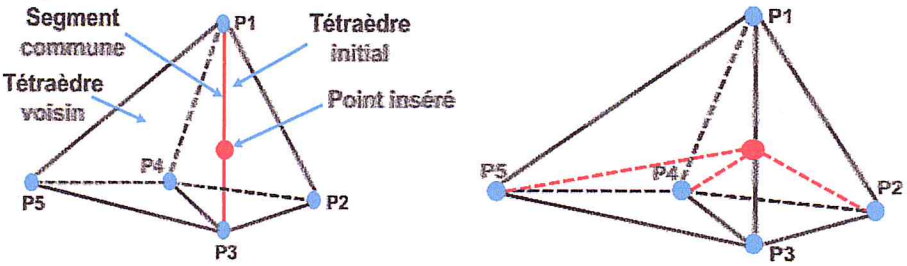
Figure 48. Split d'un point sur la face d'un tétraèdre.

2.5.1.4. Point inséré est sur un segment du tétraèdre

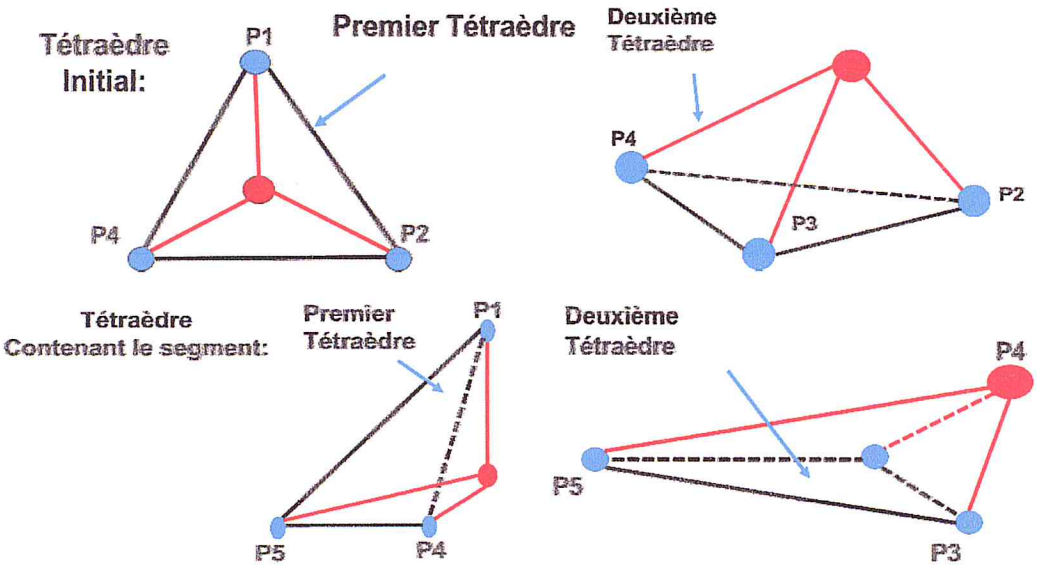
Si le point inséré se trouve sur un segment du tétraèdre, le processus est (Figure 49) :

1. Stocker l'indice du tétraèdre dans « Tableau vérifié » et « Tableau partage segment ».
2. Déterminer et stocker les tétraèdres partageant le même segment.
3. Subdiviser le tétraèdre en créant deux nouveaux tétraèdres (split 1-2) (Figure 49).

4. Remplacer le tétraèdre initial.
5. Subdiviser chaque tétraèdre ayant ce segment commun en créant deux nouveaux tétraèdres.
6. Remplacer chaque tétraèdre initial.
7. Stocker tous les nouveaux tétraèdres dans « Tableau voisin » et « Tableau vérifié ».
8. Déterminer les voisins pour tous les nouveaux tétraèdres.
9. Calculer les voisins.
10. Mettre à jour les voisins.



a. Segment contenant le point. b. Subdivision des tétraèdres en deux.



c. Résultat du split

Figure 49. Split d'un point situé sur le segment d'un tétraèdre.

2.5.2. Mise à jour des voisins

La localisation de la zone de modification des voisins et l'optimisation du temps de mise à jour des voisins se fait en deux niveaux (Figure 50) :

- ✓ Premier niveau ; création d'un tableau voisin, qui contient les voisins de chaque tétraèdre créé.
- ✓ Deuxième niveau ; création d'un tableau zone, afin de spécifier la zone de modifications ; c'est-à-dire, on doit définir pour chaque voisin ses voisins dans le but d'optimiser la zone de recherche et de minimiser le nombre de tétraèdres à modifier (Figure 51).

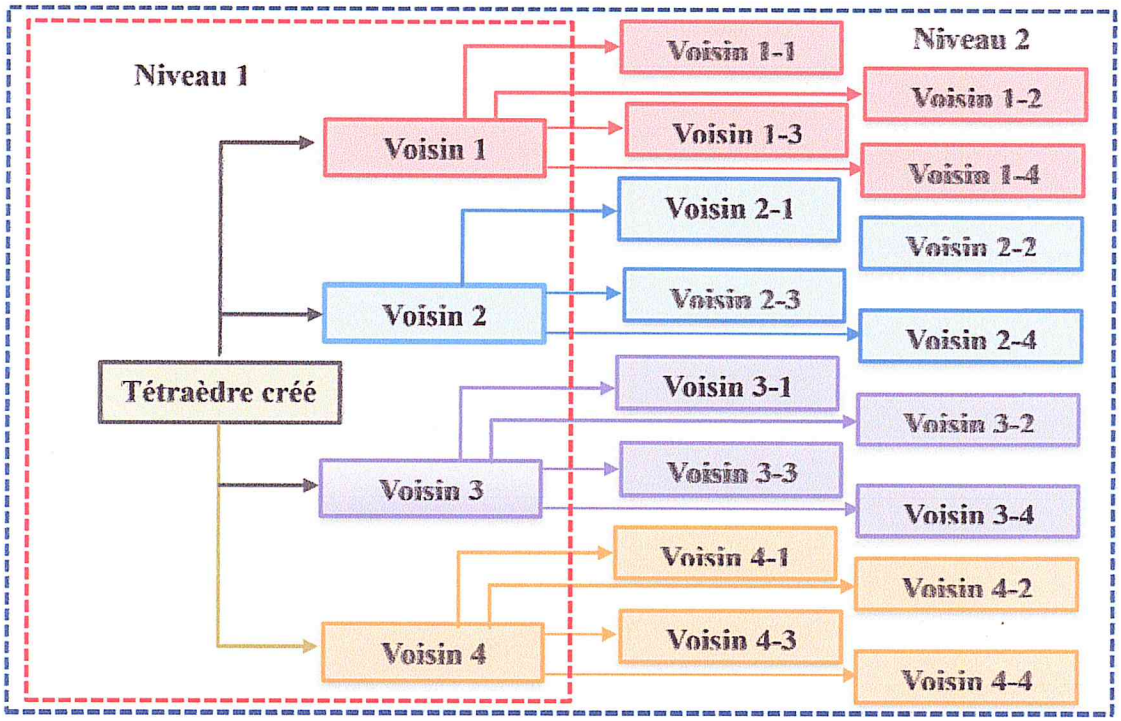


Figure 50. Tableau des niveaux de mise à jour des voisins optimisés.

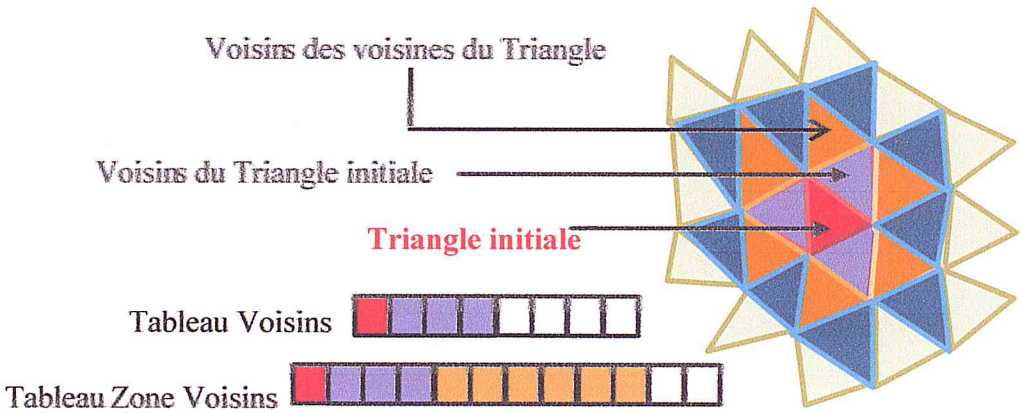


Figure 51. Mise à jour des voisins optimisée.

2.5.3. Vérification des critères de Delaunay

L'insertion d'un point dans un tétraèdre peut engendrer des tétraèdres non Delaunay. Pour cela, une étape de vérification est indispensable. Elle est résumée dans l'organigramme suivant (Figure 52) :

1. Si le point est à l'intérieur de la sphère circonscrite : ➢
   Réaliser un flip pour les deux tétraèdres.
2. Si le point est sur la sphère circonscrite :
   ➢ Modifier les coordonnées du point.
3. Si le point est à l'extérieur de la sphère circonscrite : ➢
   Le tétraèdre est valide.

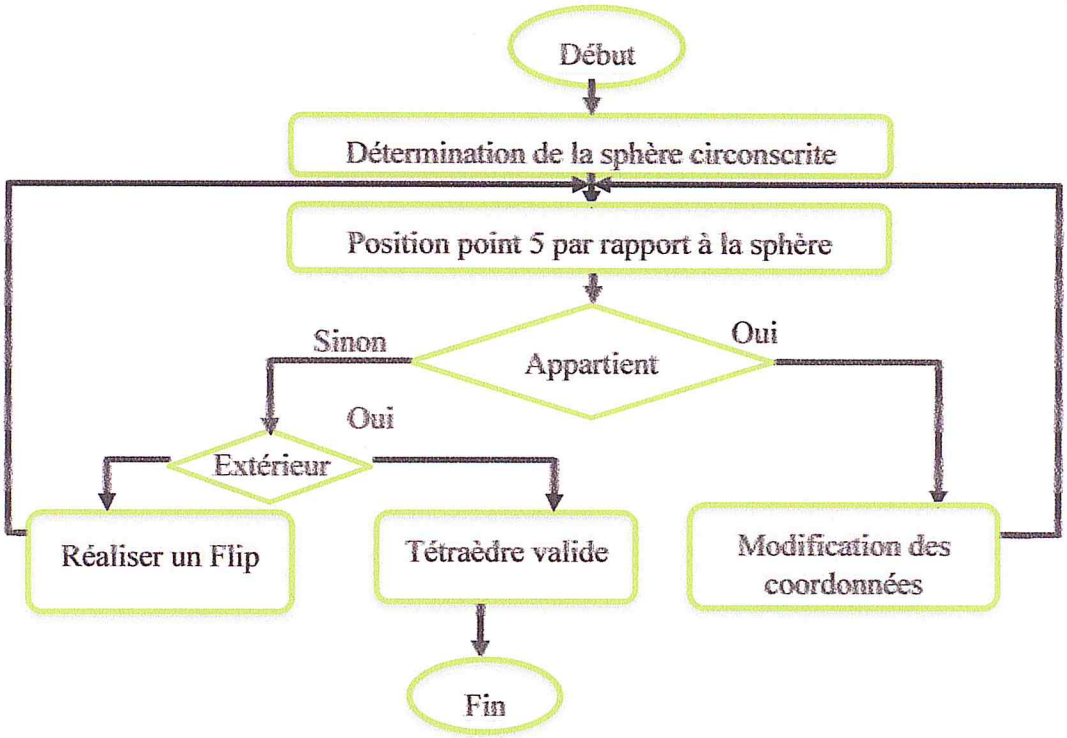


Figure 52. Organigramme de Vérification critère de Delaunay.

2.5.4. Flips

Cette opération est effectuée si le critère de Delaunay n'est pas vérifié. Pour cela, trois modes de FLIP sont envisagés à savoir FLIP 2-3, FLIP 3-2 et FLIP 4-4 pour corriger la triangulation 3D. Ce processus est illustré dans la Figure 53. Pour déterminer quel FLIP utilisé, un test sur un cône formé à partir d'un tétraèdre est effectué en suivant les étapes suivantes (Figure 54) :

1. Déterminer le cône pour ce tétraèdre.
2. Identifier le voisin de ce tétraèdre.
3. Vérifier de quel côté se trouve le point **P5** par rapport au cône. Le point 5 étant le point hors la surface commune entre le tétraèdre en question et son voisin.

Le cône test consiste à déterminer de quel côté se trouve le point par rapport au cône :

➢ Remplacer les coordonnées du point 5 dans les équations des plans des trois (03) faces du cône.

$plan1 = X_{Normale1} * X_{point5} + Y_{Normale1} * Y_{point5} + Z_{Normale1} * Z_{point5} + D1$

$plan2 = X_{Normale2} * X_{point5} + Y_{Normale2} * Y_{point5} + Z_{Normale2} * Z_{point5} + D2$

$plan3 = X_{Normale3} * X_{point5} + Y_{Normale3} * Y_{point5} + Z_{Normale3} * Z_{point5} + D3$ Récupérer la face laquelle le point 5 est situé.

➢ Calculer le produit des trois (03) équations :

$$Resultat = plan1 * plan2 * plan\ 3.$$

- Si $Resultat < 0$: le point est à l'intérieur du cône, un « FLIP 2-3 » est effectué (Figure 55).
- Si $Resultat > 0$: le point est à l'extérieur du cône, un « FLIP 3-2 » est effectué (Figure 56).
- Si $Resultat = 0$: le point est sur une des faces du cône, un « FLIP 4-4 » est effectué (Figure 57).

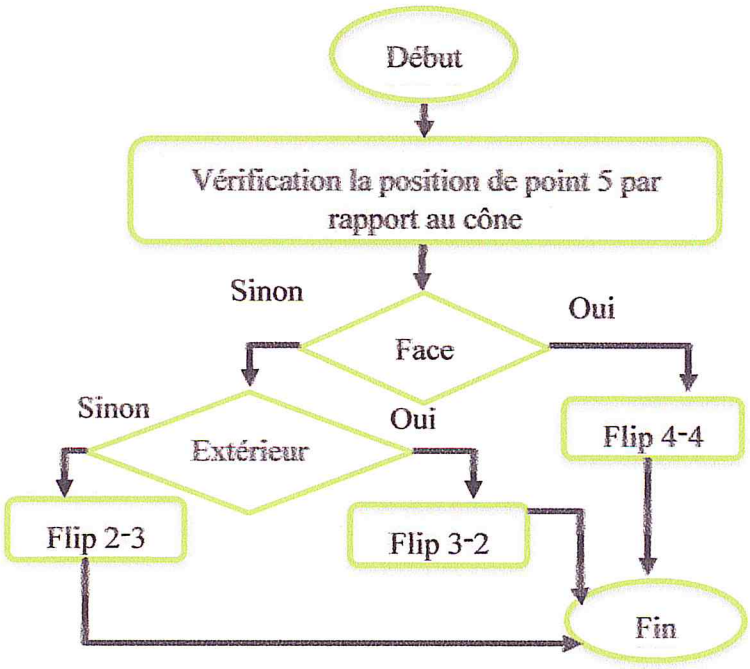


Figure 53. Organigramme du Flip.

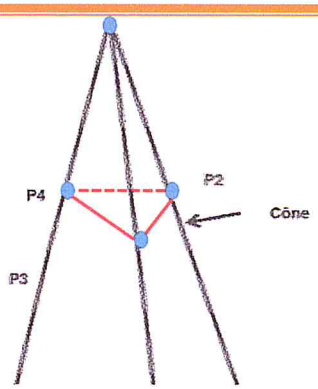
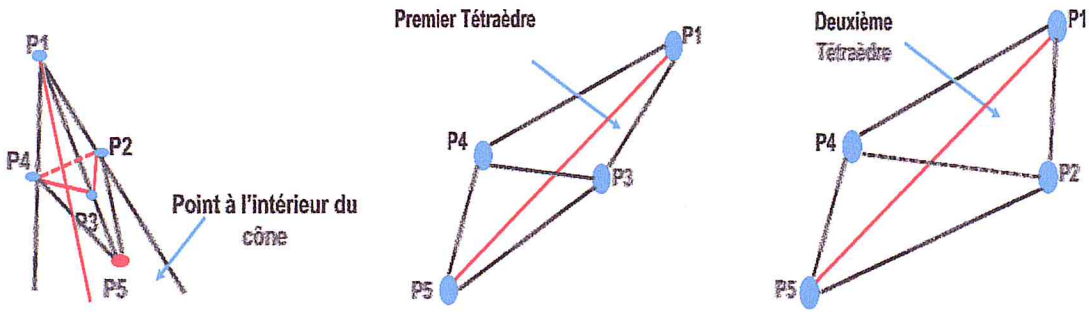


Figure 54. Cône test.

2.5.4.1. FLIP 2-3

Ce FLIP permet de passer de deux tétraèdres à trois tétraèdres. Il est effectué suivant les étapes suivantes (Figure 55) :

1. Vérification de l'appartenance du point au cône (intérieur).
2. Stocker l'indice du tétraèdre dans les deux tableaux «Tableau voisin» et «Tableau vérifier».
3. Récupérer et stocker les voisins du tétraèdre dans un tableau.
4. Stocker l'indice du tétraèdre voisin du tétraèdre cible.
5. Récupérer et stocker les voisins 6. Calculer la zone voisine pour chacun.
7. Réinitialiser les voisins à la valeur 1.
8. Réaliser un flip 2-3 toute en créant trois tétraèdres à partir des deux initiaux.
9. Remplacer les deux tétraèdres initiaux.
10. Stocker tous les nouveaux tétraèdres dans les deux tableaux «Tableau voisin» et «Tableau vérifié».
11. Calculer les voisins
12. Mettre à jour les voisins.



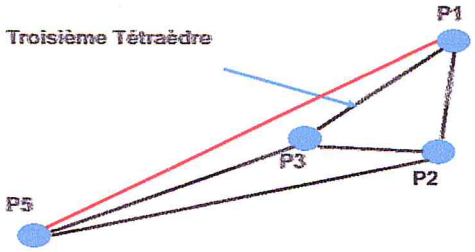
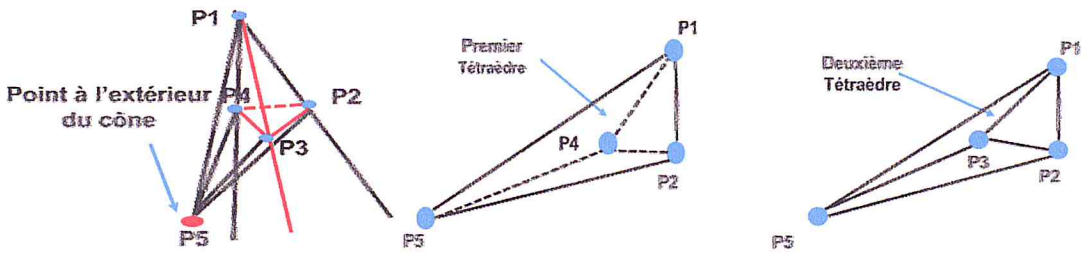


Figure 55. Flip convexe « 2-3»

2.5.4.2. FLIP 3-2

Ce FLIP permet de passer de trois tétraèdres à deux tétraèdres (Figure 56) de la manière suivante :

1. Vérification de la position du point (à l'extérieur du cône).
2. Vérification de l'existence d'un voisin commun entre les deux premiers tétraèdres.
3. Si oui :
   - Récupérer l'indice du voisin commun.
   - Stocker l'indice du tétraèdre et du tétraèdre voisin.
   - Récupérer et stocker les voisins
   - Calculer la zone voisine pour chacun.
   - Réinitialiser les voisins à -1.
   - Création de deux tétraèdres à partir des trois initiaux (Flip 3-2).
   - Remplacer les deux tétraèdres initiaux.
   - Supprimer le voisin commun.
   - Stocker tous les nouveaux tétraèdres dans les deux tableaux «Tableau voisin» et «Tableau vérifier».
   - Calculer les voisins.
   - Mettre à jour les voisins.
4. Si non, ce cas ne sera pas traité, d'autres FLIPs régulariseront ce problème par la suite.



a. Point à l'extérieur du cône

b. Résultat du «FLIP 3-2 »

Figure 56. FLIP non convexe «FLIP 3-2 ».

2.5.4.3. FLIP 4-4

Ce FLIP permet de passer de quatre tétraèdres à quatre autres (Figure 57). Il se fait de la façon suivante :

1. Vérifier si le point est situé sur une face du tétraèdre initial.
   - La face commune entre le tétraèdre initial et le tétraèdre voisin « j » est obtenue.
2. Déterminer le voisin « j ».
3. Déterminer le voisin commun du tétraèdre contenant le point et le tétraèdre « j ».
4. Si le voisin existe :
   - Déterminer les points communs (pour la création des nouveaux tétraèdres).
   - Créer quatre nouveaux tétraèdres à partir des quatre initiaux.
   - Remplacer les deux tétraèdres initiaux.
   - Stocker tous les nouveaux tétraèdres dans les tableaux voisins et vérifier.
   - Calculer les voisins.
   - Mettre à jour les voisins.
5. Si non, ce cas ne sera pas traité, d'autres FLIPs régulariseront ce problème par la suite.

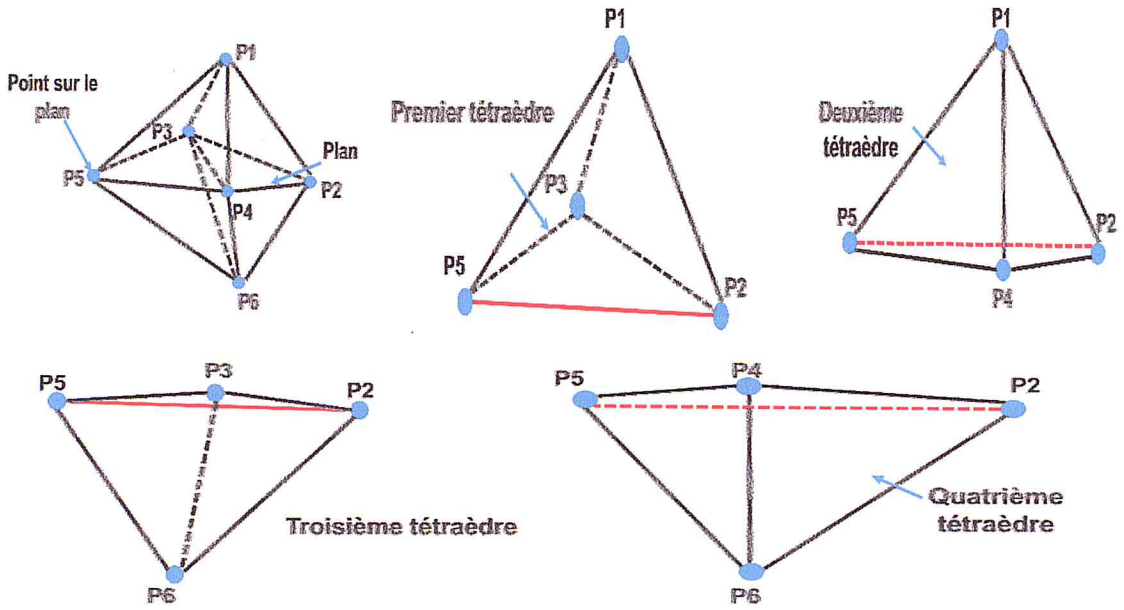


Figure 57. FLIP « 4-4 ».

2.6. Suppression des points fictifs :

Une fois la triangulation terminée, la suppression des points fictifs est effectuée (Figure 58) :

1. Parcourir la liste des tétraèdres résultants :

- Si un des quatre points du tétraèdre est fictif, alors il est supprimé.

2. Après avoir tout parcouru, le critère de l'enveloppe convexe est ainsi vérifié et la triangulation est une triangulation de Delaunay.

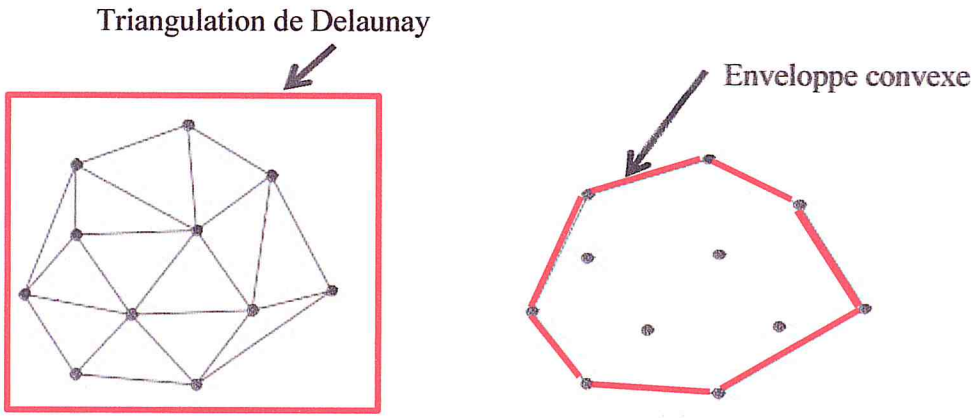


Figure 58. Enveloppe convexe.

2.7. Génération du fichier STL

L'objectif final de ce travail est la génération du modèle STL de l'objet digitalisé pour être exploité dans la phase de fabrication. Le modèle STL est généré à partir de la triangulation finale de Delaunay 3D en suivant les étapes suivantes :

1. Calcul des voisins de tous les tétraèdres.
2. Recherche des faces dont les voisins sont nulles (faces non communes avec d'autres tétraèdres).
3. Sauvegarde de la face identifiée.
4. Sauvegarde des faces en suivant la structure du fichier STL
5. Visualisation du modèle STL

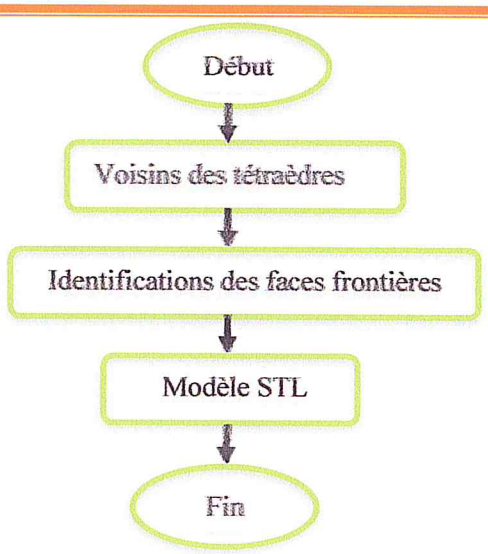


Figure 59. Organigramme de génération du fichier STL.

3. Modélisation de l'application avec UML

Dans cette partie, les besoins sont analysés et modélisés avec le langage UML, c'est l'acronyme anglais pour «Unified Modeling Language». On le traduit par «langage de modélisation unifié». La notion UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc, des diagrammes pour représenter le logiciel à développer :

son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel. Ainsi, il vaut mieux modéliser un système avant de réaliser de manière visuelle et graphique les besoins, les solutions fonctionnelles et techniques du projet. La modéliser permet de :

- Obtenir une application de très haut niveau indépendante des langages et des environnements.
- Faire collaborer des participants de tous horizons autours d'un même document de synthèse.
- Faire des simulations avant de construire un système.

Dans notre projet, les diagrammes suivants ont été utilisés.

❖ **Expression des besoins** : pour ce cas, le diagramme de cas d'utilisation est utilisé chacun de ces cas d'utilisation est explicité à travers les organigrammes qui ont été vus au préalable.

❖ **Conception :** pour la conception de l'application, le diagramme de classe a été utilisé.

3.1 Diagramme de cas d'utilisation

Les cas d'utilisation (use cases) permettent de :

- Modéliser les besoins des utilisateurs.
- Modéliser les objectifs correspondants d'un système.
- Représenter les interactions entre le système et ses utilisateurs.

Dans ce qui suit, les différentes fonctionnalités du système développé ainsi que les offres proposées aux utilisateurs avec un diagramme de cas d'utilisation sont présentées (Figure 60) voir Annexe A.

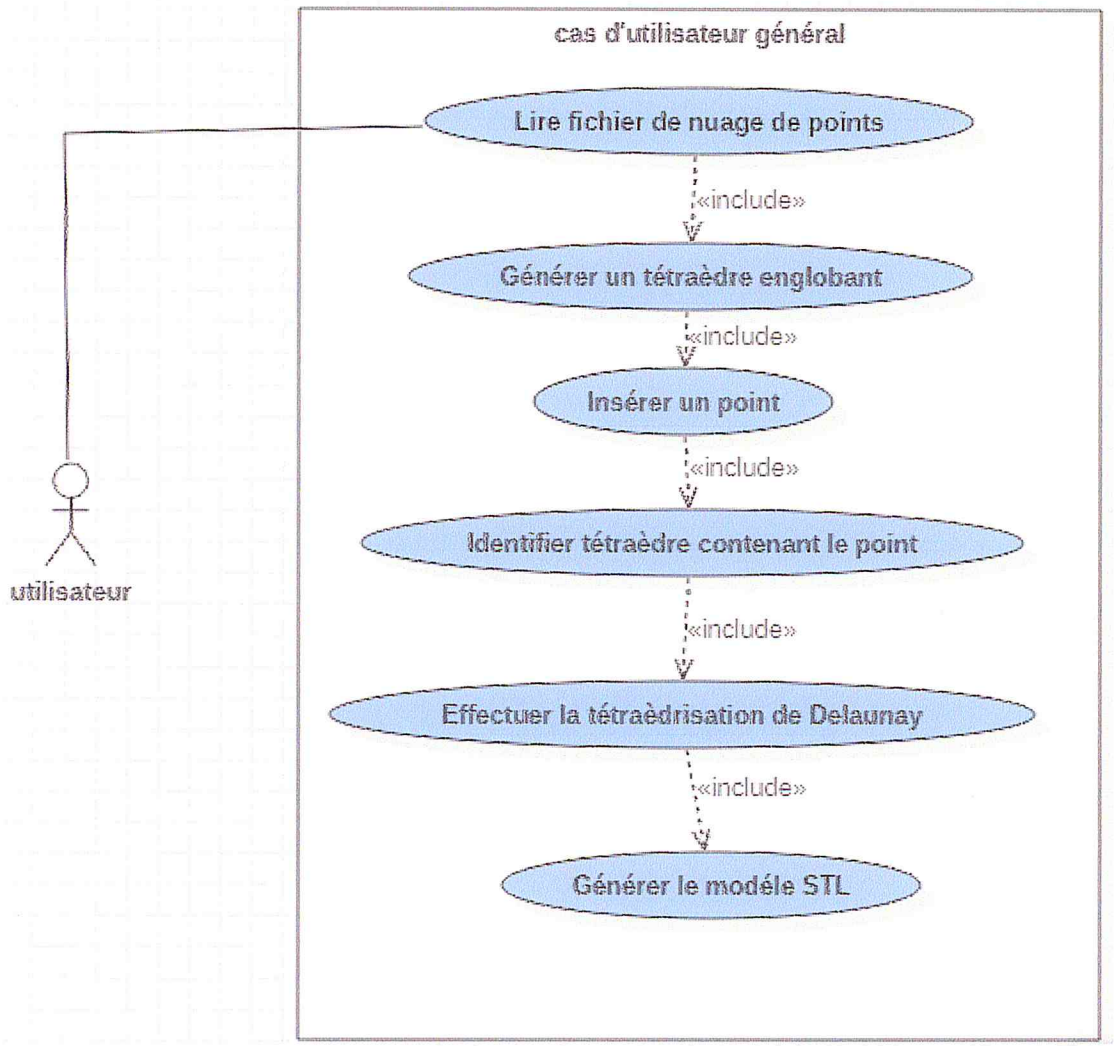


Figure 60. Diagramme de cas d'utilisation général.

3.2 Diagramme de classes

Un diagramme de classes décrit le type des objets ou données du système ainsi que les différentes formes de relations statiques qui les relient entre eux. Classiquement, ils existent deux principaux types de relations entre objets :

•Les associations, bien connues des vieux modèles entité/association utilisés dans la conception des bases de données depuis les années 70.

•Les sous-types, particulièrement en vogue en conception orientée objets, puisqu'ils s'expriment très bien à l'aide de l'héritage en programmation.

Dans ce qui suit, le digramme de classes global développé est présenté en tenant compte des relations entre les classes (Figure 61). Les principales classes utilisées sont détaillées. Les classes implémentées dans ce travail sont :

•Classe Point_Delaunay_Optim □ Classe Plan_Optim.
•Classe Nuage_Points_Optim.
•Classe Brut_Optim.
•Classe Normale_Optim.
•Classe Face_stl_fr.
•Classe Segment.
•Classe Sphère.
•Classe Tetraedre_Optim.
•Classe Triangulation_Delaunay_Optim

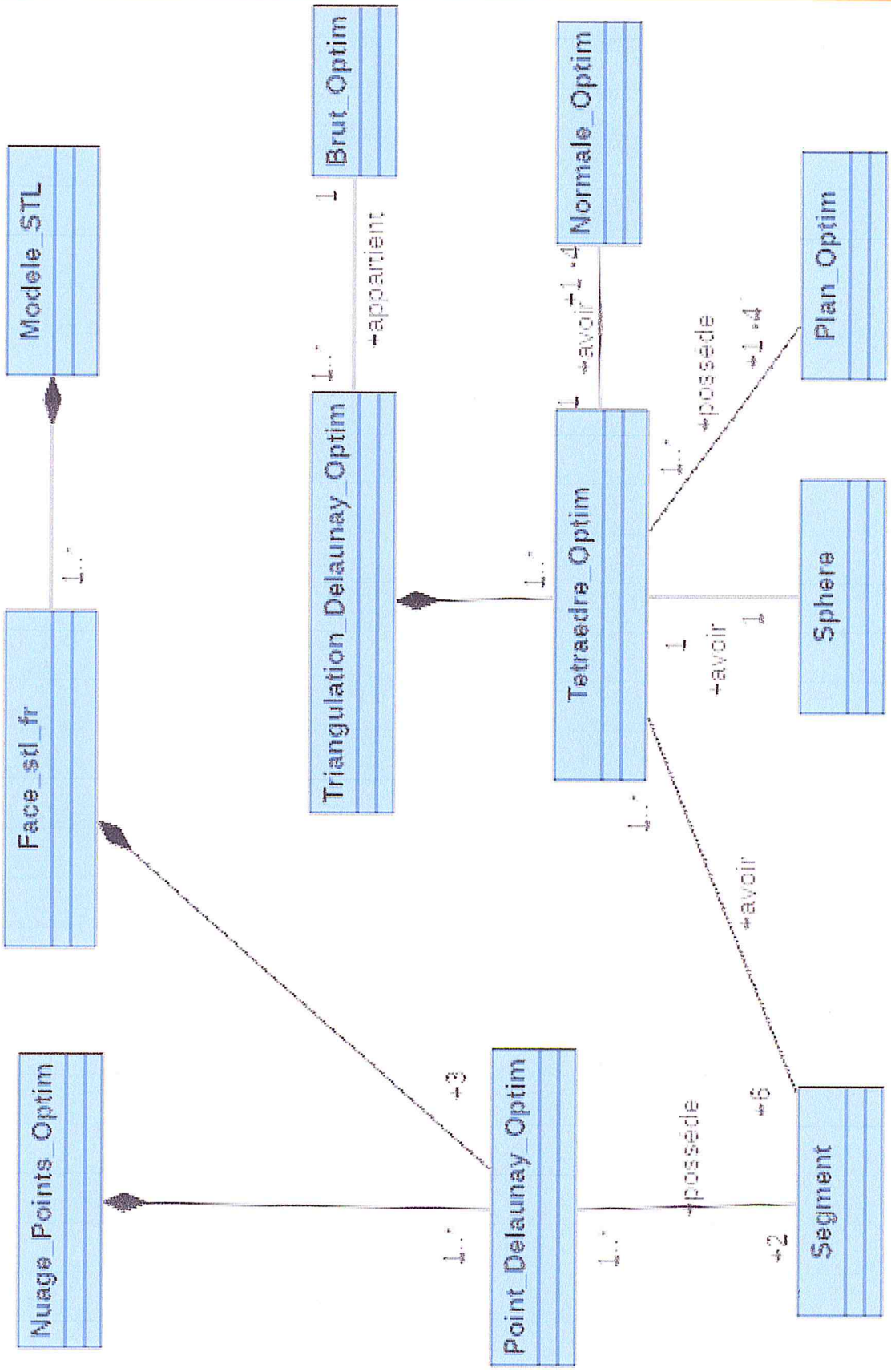


Figure 61. Diagramme de classe général.

La représentation détaillée (attributs et fonctions) de différentes classes Implémentée est décrites ci-dessous :

✓ **Classe « Nuage_points_Optim» :** c'est la classe principale qui regroupe un ensemble de points enregistrés dans un tableau, des tétraèdres sauvegardés dans un tableau, des segments et le nombre total des points ainsi que des tétraèdres. Les méthodes de cette classe permettent de récupérer les principales informations du fichier txt (Figure 62).

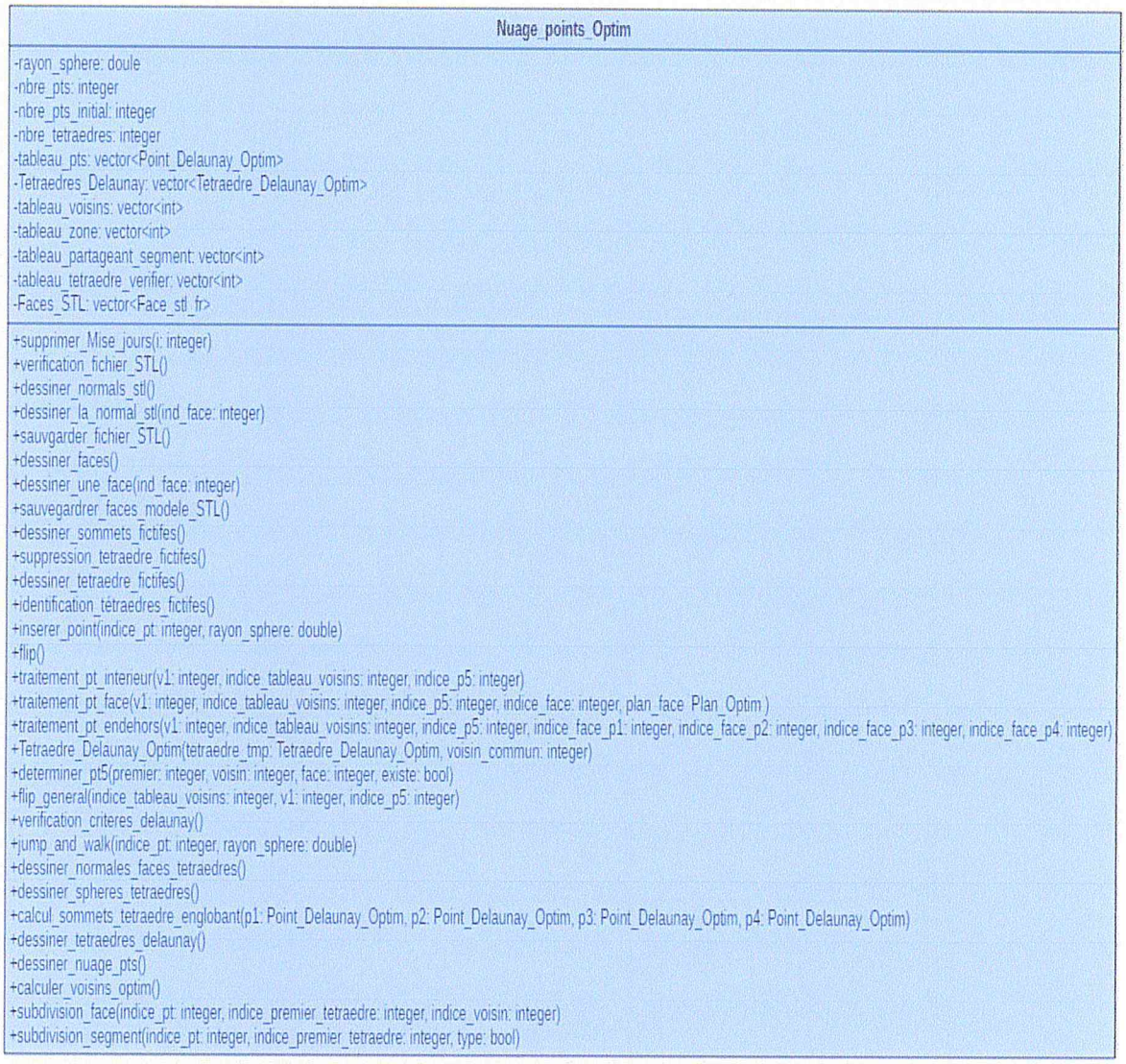


Figure 62. Classe Nuage_points_Optim.

✓ **Classe «Point_Delaunay_Optim» :** c'est la classe qui regroupe les coordonnées X, Y, Z des points ainsi que les indices des points fictifs (Figure 63). Les méthodes de cette classe permettent de déterminer les différents paramètres du point.

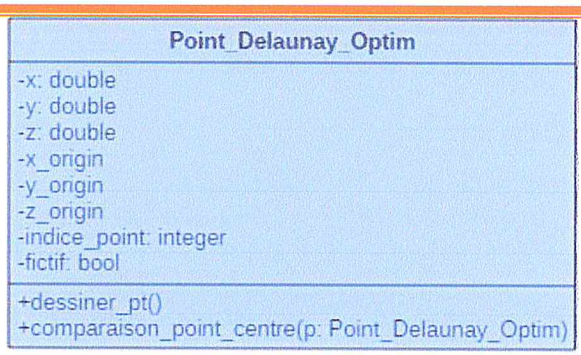


Figure 63. Classe Point_Delaunay_Optim.

✓ **Classe «Brut_Optim»**: regroupe les limites du nuage de points. Pour une visualisation claire du brut, on utilise des fonctions de dessin du brut (Figure 64).

Figure 64. Classe Brut _Optim.

✓ **Classe «Normale_Optim.» :** cette classe faite pour déterminer la normale d'une face (Figure 65).

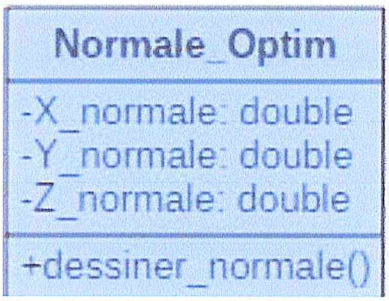


Figure 65. Classe Normale _Optim.

✓ **Classe «Tetraedre_Optim » :** contient les différentes caractéristiques du tétraèdre, ses attributs et ses fonctions sont données. Ces fonctions sont données pour enrichir un tétraèdre. (Figure 66).

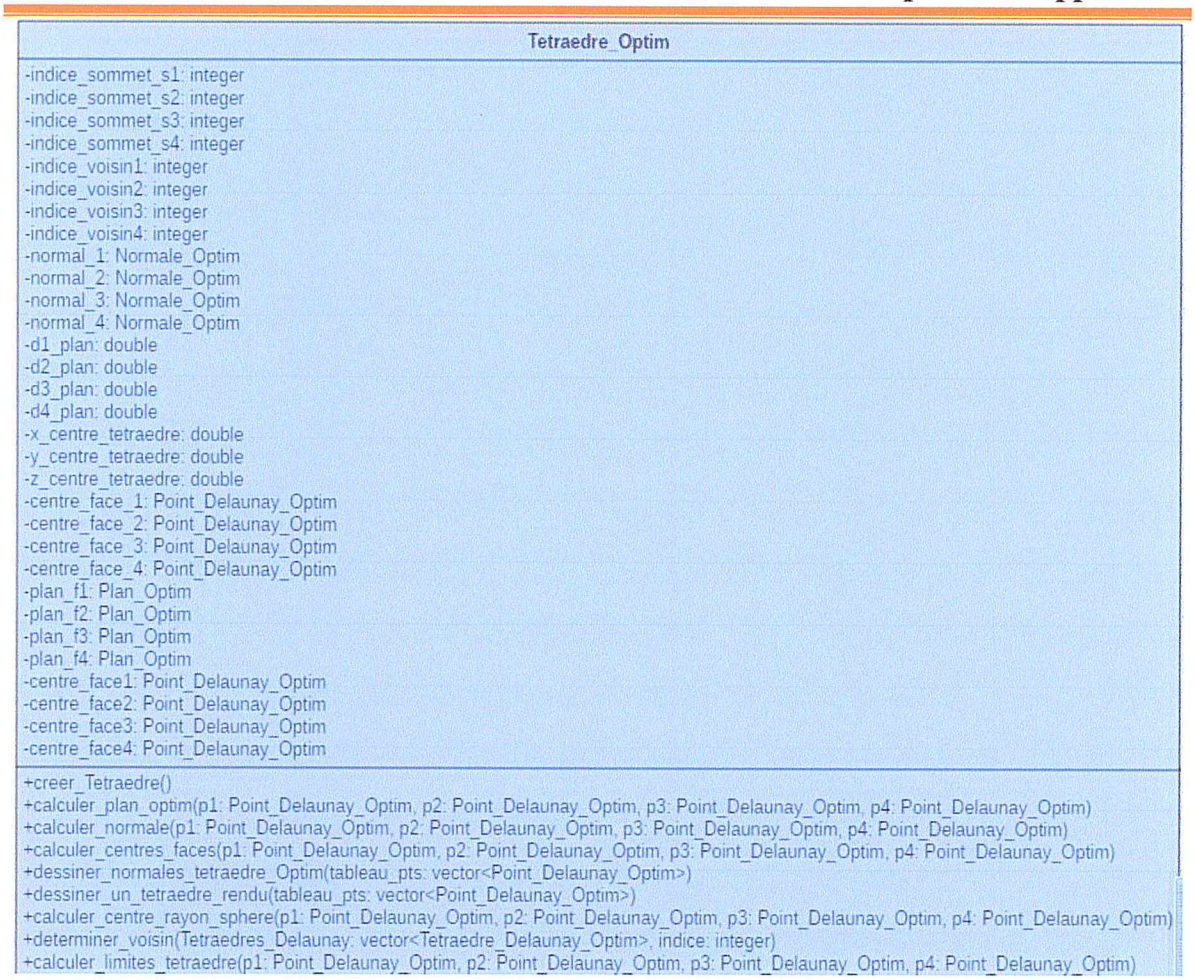


Figure 66. Classe Tetraedre_Optim.

✓ **Classe «Face_stl_fr»:** c'est une classe qui englobe les différentes caractéristiques des triangles (Figure 67). Ses fonctions sont données pour enrichir un triangle.

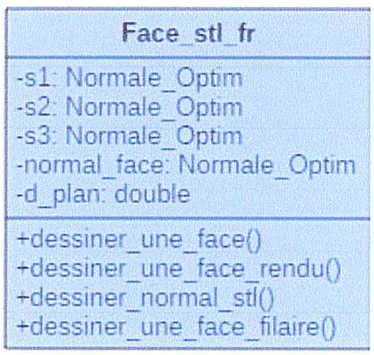


Figure 67. Classe Face_stl_fr.

Conclusion

Dans ce chapitre on a expliqué en détail la partie conception et on a présenté tous les détails nécessaires pour la réalisation de notre application avec des organigrammes pour les algorithmes utilisée dans le développement de notre application.

Dans le chapitre suivant, nous allons prendre un exemple réel pour tester le bon fonctionnement de notre application, en détaillant le processus de reconstruction implémenté.

Chapitre III

Implémentation et Validation

les algorithmes les plus usuels et aussi permet d'appliquer les techniques de la programmation objet. Il apporte notamment la gestion des exceptions, la gestion des références (remplaçant partiellement l'usage quelque peu délicat des pointeurs), la surcharge des opérateurs et les Template (liste non exhaustive). Enfin, une rétrocompatibilité a été gardée ; les programmes en C compilent sans difficulté avec un compilateur C++. Comme tout langage, C++ dispose d'une bibliothèque standard, c'est-à-dire de fonctions et de classes prédéfinies. Voici une figure qui représente les meilleurs environnements de développement.

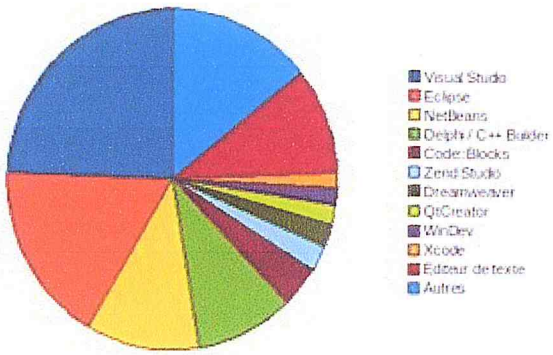


Figure 68. Meilleurs environnement de développement.

1.1.2. Présentation d'OpenGL

Open Graphic Library (ou OpenGL) est une API graphique multiplateforme open-source, de bas niveau, dédiée pour les applications générant des images 2D ou 3D. OpenGL est sortie en 1992 par la Sillicon Graphics Inc. (ou SGI) en 1992 [16], et est actuellement gérée par la société - à but non lucratif - Khronos Group. La version actuelle de l'API est le 4.2 (sortie le 8 août 2011). OpenGL est conçue pour développer des applications graphiques portables, et elle favorise l'innovation et la simplicité d'utilisation en offrant aux développeurs de nombreuses fonctionnalités de visualisations. Etant libre, ouverte et portable, la bibliothèque graphique a pu se vendre dans le milieu scientifique et industrielle et être utilisé dans de nombreuses applications artistiques ou de traitement d'images.

Du fait de ses performances, l'API OpenGL est aussi présent dans le marché des jeuxvidéoludiques, en concurrence direct avec la bibliothèque graphique propriétaire DirectX. En ajout, OpenGL est de plus en plus utilisée dans les applications web (en particulier grâce à la bibliothèque WebGL), et est présente dans les systèmes embarqués (OpenGL ES, ou OpenGL for Embedded Systems), dont les téléphones portables (les systèmes Android et iPhone) et les PDA.

OpenGL est une API très performante et simple d'utilisation, qui permet de réaliser des rendus et des scènes complexes, selon nos besoins. Cette technologie est ouverte et est implémentée par de nombreuses bibliothèques, sous des langages de programmations différents (dont des langages de script comme Python ou Ruby). L'API est gratuite. Par ailleurs, OpenGL est portable et multiplateforme. En théorie, un même code OpenGL fonctionne sur tous les systèmes d'exploitation, dont Windows, Linux, et Mac, et sur toutes les plateformes.

1.1.3. Présentation embarcadero C++ Builder 10 Seattle

C++Builder est un logiciel de développement rapide d'applications conçu par Borland qui reprend les mêmes concepts, la même interface et la même bibliothèque que Delphi en utilisant le langage C++. Ce dernier est développé par Embarcadero Technologies avec un système d'exploitation « Microsoft Windows » et un langage de programmation C ++, Pascal Objet. Ces auteurs originaux sont : Borland, CodeGear, embarcadero technologies. Embarcadero C ++ Builder 10 Seattle est sorti fin août 2015 est le moyen le plus rapide de créer et de mettre à jour des applications riches en données, hyper connectées et visuellement engageantes pour Windows 10 32 et 64 bits, Mac, Mobile, IoT et plus encore en utilisant le standard C ++ et aussi mettre à jour les applications VCL et FMX vers Windows [17].

1.2. 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, Intel® Core™ I5-2450M CPU @2.50 GHz 2.50 GHz 4,00Go de RAM.

La configuration minimale pour l'exécution de l'application développée est la suivante :

- RAM : 04 Go
- Carte graphique : Intel(R) HD Graphics 3000.
- Processeur multi-cœurs.

1.3. Présentation de la fenêtre principale de plateforme de « CFAO » :

Cette fenêtre est la première fenêtre qui apparait après l'exécution de la plateforme logicielle pour la production numérique de pièces complexes de l'équipe « CFAO ». La plateforme est composée de deux parties (Figure 69) :

- **Scène de visualisation** : c'est un écran d'affichage des résultats des projets réalisés au niveau de l'équipe « CFAO ». Des paramètres géométriques permettent la visualisation des résultats de différents ongles.

➢ **Barre du menu principal :** la barre du menu principal contient tous les travaux précédents de l'équipe « CFAO » y compris notre application (Figure 3).

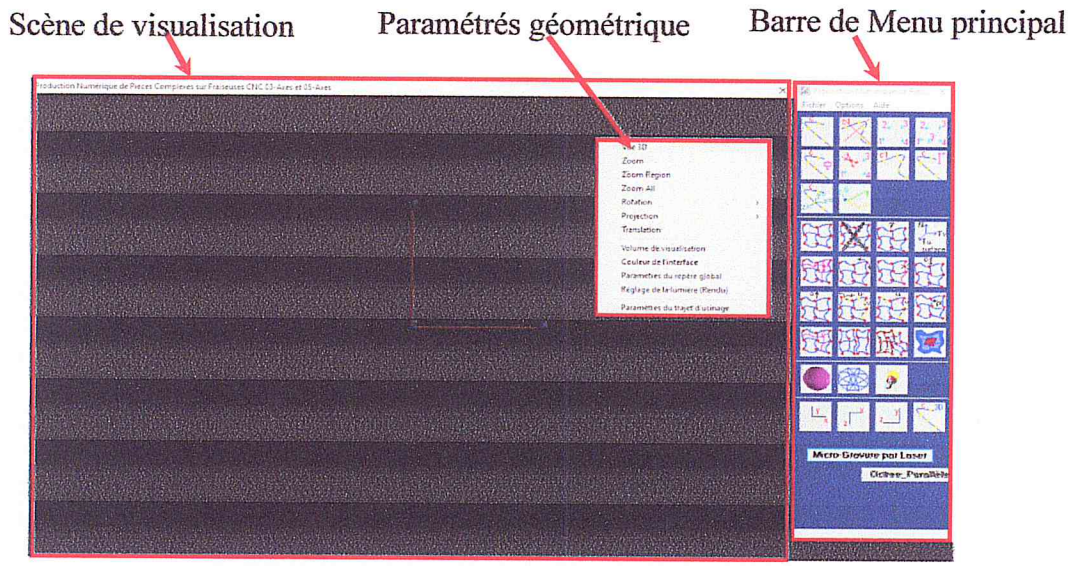


Figure 69. Environnement de « **CFAO** » .

1.4. Présentation de l'application développée

Pour accéder à la partie relative à notre application, il suffit d'activer le bouton reconstruction des surfaces gauches à partir d'un nuage de points puis sur le bouton Optimisation de la Triangulation de Delaunay par FLIP (Figure 70).

L'application développée est composée de trois (03) ongles. Chaque onglet est dédié à des phases du processus de génération de la triangulation de Delaunay (Figure 71) :

- Onglet 1 : Lecture du Fichier.
- Onglet 2 : Tétraédrisation.
- Onglet 3 : Génération du fichier STL.

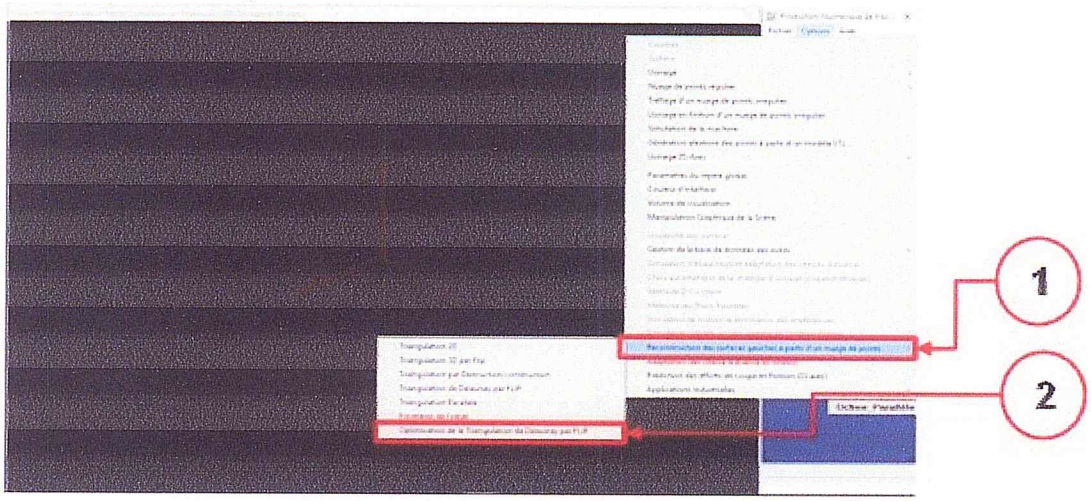


Figure 70. Etapes de démarrage de notre application logicielle.

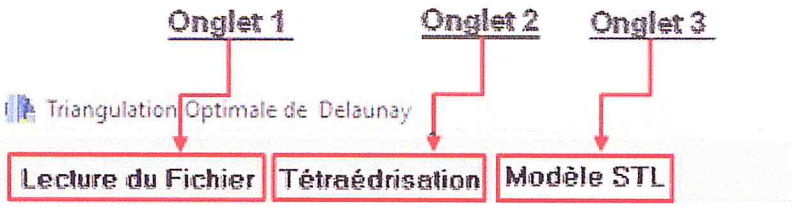


Figure 71. Onglets de l'application développée.

1.4.1. Lecture du fichier du nuage de points

Le premier onglet « **Lecture du Fichier**» (Figure 72) permet la lecture du fichier texte contenant l'ensemble des paramètres du nuage de points. Il est divisé en deux parties :

La première partie permet la lecture du fichier contenant le nuage de points par simple clic sur le bouton «Ouvrir le fichier» et la sélection du fichier à ouvrir.

Une fois le fichier est lu, un clic sur le bouton **«calculer les limites»** permet de passer au calcul des paramètres du brut :

- Les coordonnées des limites du brut (X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{min}, Z_{max}).
- Dimensions du brut (Hauteur, Largeur et Longueur) ainsi que le nombre de points du nuage.

La seconde partie est consacrée à la visualisation. Cette partie permet d'afficher les nuages de points ainsi que le brut soit en filaire ou en rendu avec les différentes dimensions.

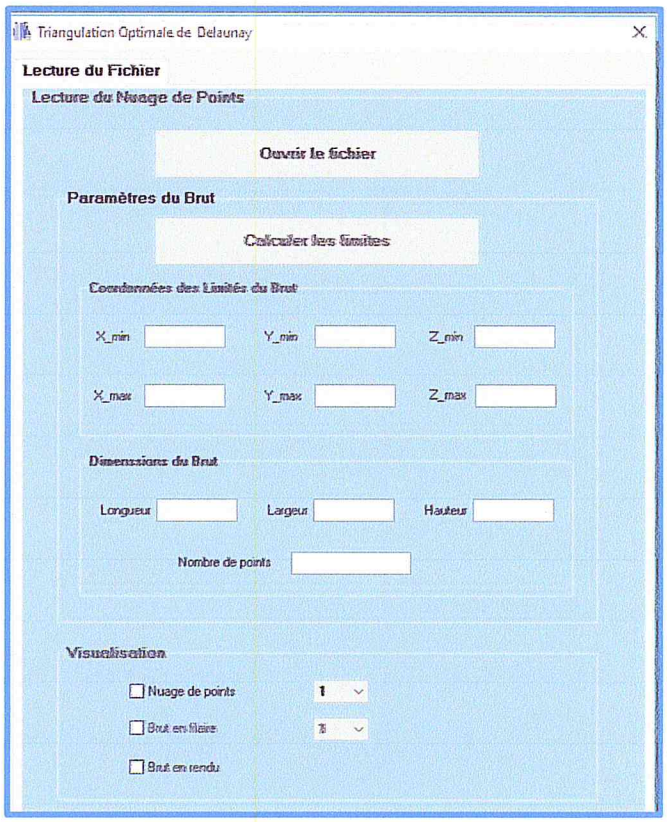


Figure 72. Onglet « Lecture du Fichier».

1.4.2.Tètraèdrisation

Ce deuxième onglet est réservé à la partie de la Tétraèdrisation de Delaunay. La première étape est le **Tétraèdre englobant**. Un clic sur le bouton **«déterminer le tétraèdre englobant »** permet de générer un seul tétraèdre qui englobe tout le nuage de points.

Une fois cette tache terminée, l'utilisateur peut visualiser le tétraèdre englobant, en filaire et en rendu, sa sphère circonscrite ainsi que les normales et les limites du tétraèdre englobant.

La deuxième étape; la **Tétraédrisation de Delaunay** permet l'insertion de points selon deux modes:

- Mode1 : « **Triangulation Manuelle** » .
- Mode2 : « **Triangulation Automatique** » .

Le mode de triangulation manuelle consiste à insérer point par point en cliquant sur le bouton **«insérer»**. Ceci permet de faire une optimisation de la zone de recherche de chaque point inséré. Par la suite, un clique sur le bouton «**flip**» permet d'effectuer les différentes opérations de flips toute en vérifiant les critères de Delaunay avec une intégration de l'optimisation de la mise à jour des zones voisines après chaque split et flip au niveau des deux boutons.

Le mode de triangulation automatique consiste à insérer les points de manière séquentielle, tout en vérifiant les critères de Delaunay. Ceci est exécuté par un simple clic sur le bouton **«Lancer automatiquement Delaunay».**

Une fois la triangulation terminée, l'utilisateur peut visualiser (Figure 73) :

- Le pourcentage d'avancement du traitement.
- L'ensemble ou une partie des tétraèdres, leurs voisins, en filaire et en rendu, leurs sphères circonscrites et les normales.
- Chaque tétraèdre avec ses voisins, la sphère circonscrite et les normales
- Le point inséré, le point 5 et la zone des voisins optimisée en plus le temps de calcul.

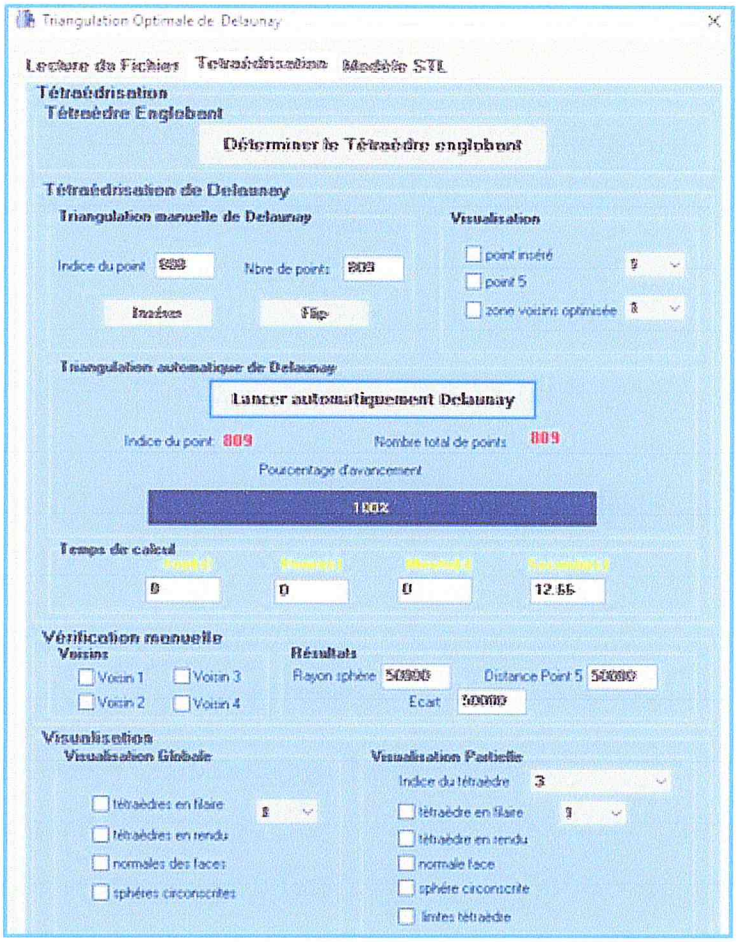


Figure 73. Onglet « Tétraèdrisation ».

1.4.3. Modèle STL

Cet onglet est dédié à la phase de génération du modèle STL. Tout d'abord , il faut appuyer sur le bouton **«Identifier les Tétraèdres fictifs»** qui sert à déterminer tous les tétraèdres fictifs. Ces dernier sont visible en en filaire et en rendu ainsi que les sommets fictifs (Figure 74).

Le second bouton **«Supprimer les Tétraèdres fictifs »**permet de supprimer les tétraèdres fictifs qui sont identifiés par le premier bouton. A ce stade, les tétraèdres finaux sont visualisés en filaire ou en rendu.

Par la suite, le modèle STL est généré par simple clic sur le bouton **« faces du modèle STL »** qui permet de distinguer les différents faces du modèle STL.

Une fois cette tâche est assurée, l'utilisateur à une visualisation sur chaque face soit en filaire ou en rendu en plus des normales STL.

Une action sur le bouton **« sauvegarder modèle STL»** permet la sauvegarde du fichier avec l'extension « **.STL** ».

Enfin, le bouton **« vérifier le fichier STL»** permet de vérifier la validation du fichier généré.

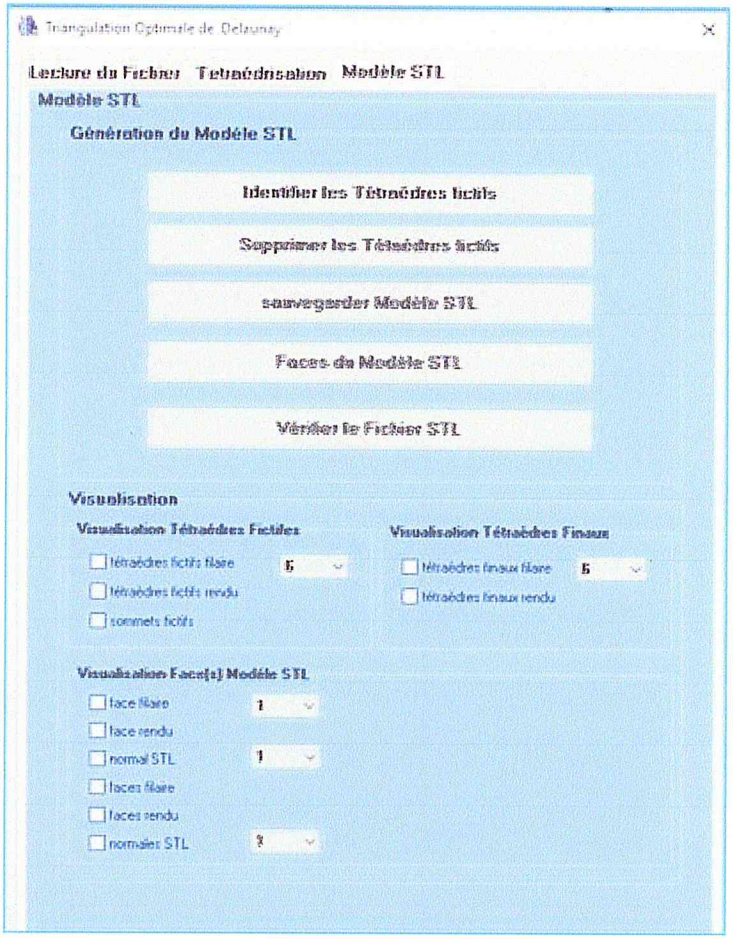


Figure 74. Onglet « modèle STL ».

2. Validation

La phase de validation consiste à tester les approches proposées. Pour cela, la pièce de la (Figure 75) est prise comme support de validation. Le traitement est effectué sur le nuage de points issus de la digitalisation de l'objet (Figure 76) depuis la lecture du fichier jusqu'à la génération du modèle STL.

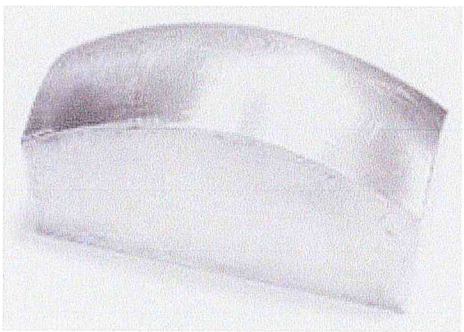


Figure 75. Pièce de test.

Figure 76. Nuage de points de la pièce.

Etape 1 : Lecture du fichier

La première étape consiste à lire le fichier contenant le nuage points (Figure 77) et à calculer les paramètres du brut. Les résultats des calculs des attributs sont présentés dans la Figure 78 et visualisation du nuage de point dans la Figure 79.

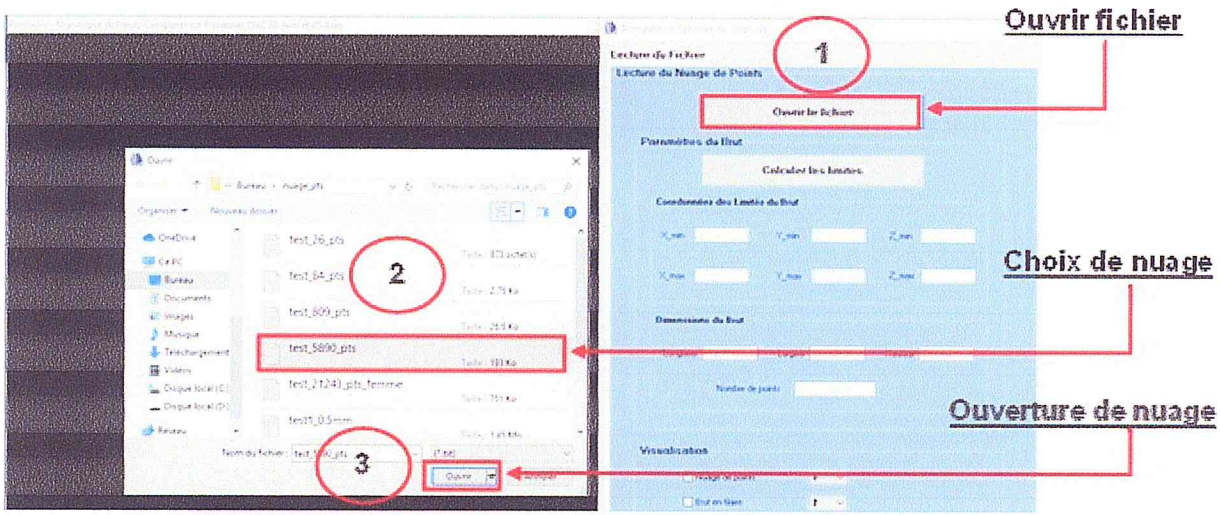


Figure 77 : Lecture du fichier de nuage de points

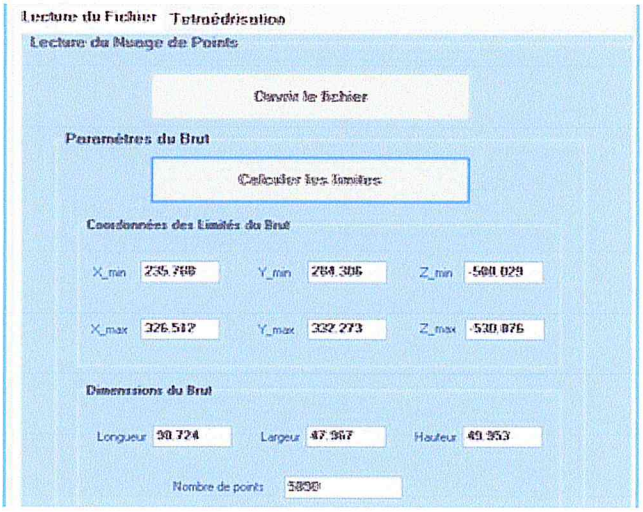


Figure 78. Résultat du calcul limites du nuage de points.

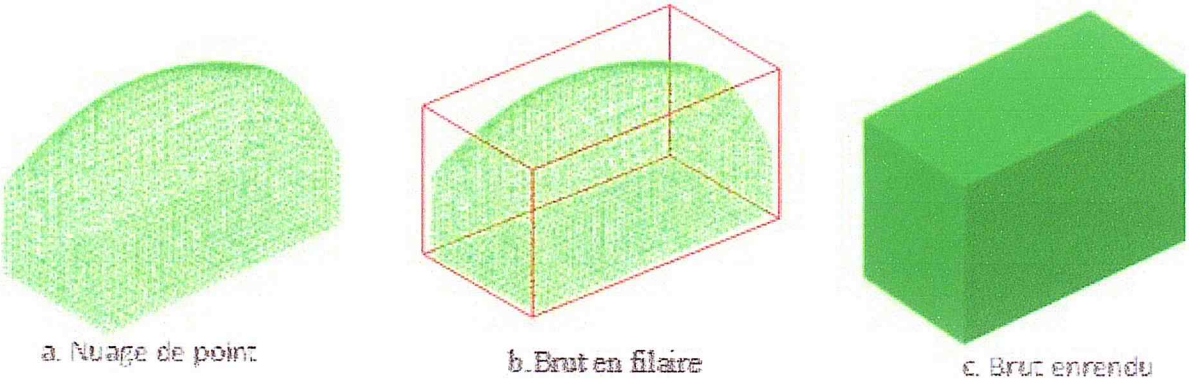
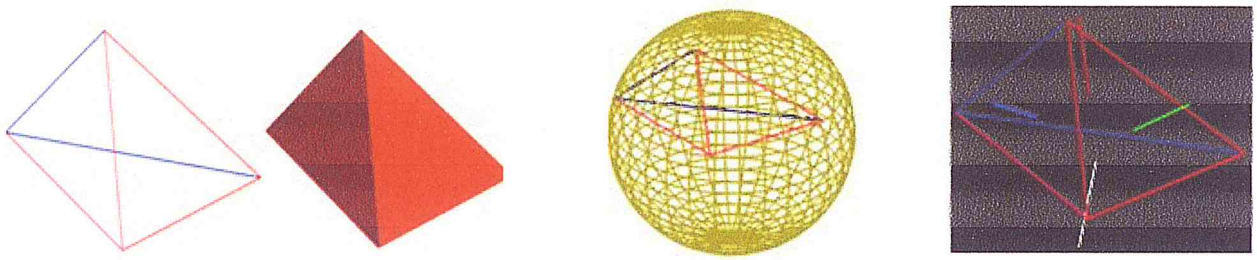


Figure 79. Résultat de la visualisation du nuage de points.

- **Etape 2 :Tètraèdrisation de Delaunay**

Le mode d'exécution automatique est adopté. Les résultats de cette étape sont illustrés par la Figure 80 et la Figure 81. L'utilisateur peut visualiser la Triangulation de Delaunay finale obtenue en filaire, en rendu et les normales des tétraèdres.



a. Tétraèdre filaire.et Tétraèdre rendu. b. Sphère circonscrite. c. Normales du tétraèdre.

Figure 80 . Visualisation du tétraèdre englobant le nuage de points.

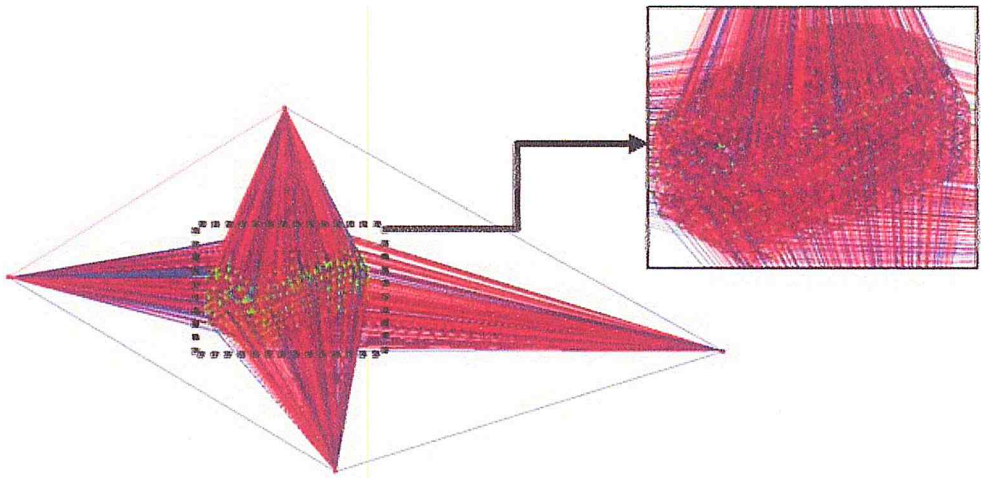


Figure 81. Visualisation de la Triangulation de Delaunay final.

- **Etape 3 : Génération du model STL:** la génération du modèle STL se fait par l'exécution séquentielle des action suivantes :

- ✓ Clic sur le bouton « **Identification des tétraèdres fictifs**», le tétraèdre fictif ainsi que les sommets fictifs sont visibles (Figure 82).
- ✓ Clic sur le Bouton « **Supprimer les Tétraèdres fictifs**», tous les tétraèdres fictifs et le tétraèdre englobant sont supprimés (Figure 83) .
- ✓ Clic sur bouton « **Sauvegarder Modèle STL**» le fichier STL est sauvegardé sous format «.STL » en 3D (Figure 84) (Figure 85).
- ✓ Clic sur le bouton « **Faces du Modèle STL**» seules les faces (triangles) extérieures du modèle sont gardées (Figure 86).
- ✓ Clic sur le bouton « **Vérification le Fichier STL**», une vérification automatique du fichier est effectuée (Figure 87).

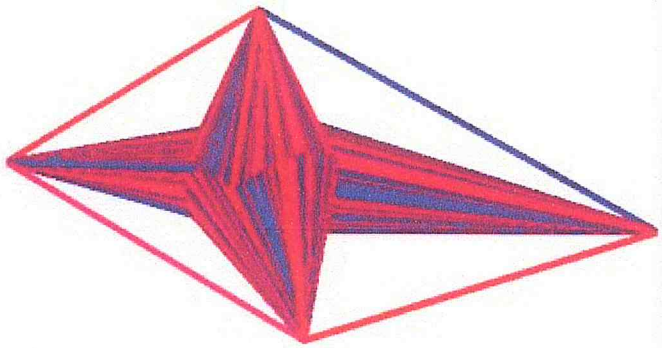
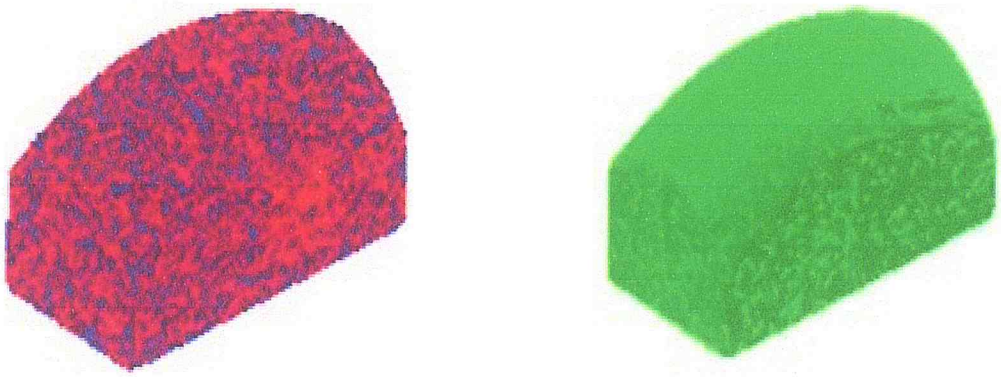


Figure 82. Identification des tétraèdres fictifs en filaire .



a.Tétraèdrisation Filaire. b. Tétraèdrisation Rendu.

Figure 83.Tétraédrisation sans tétraèdres fictifs.

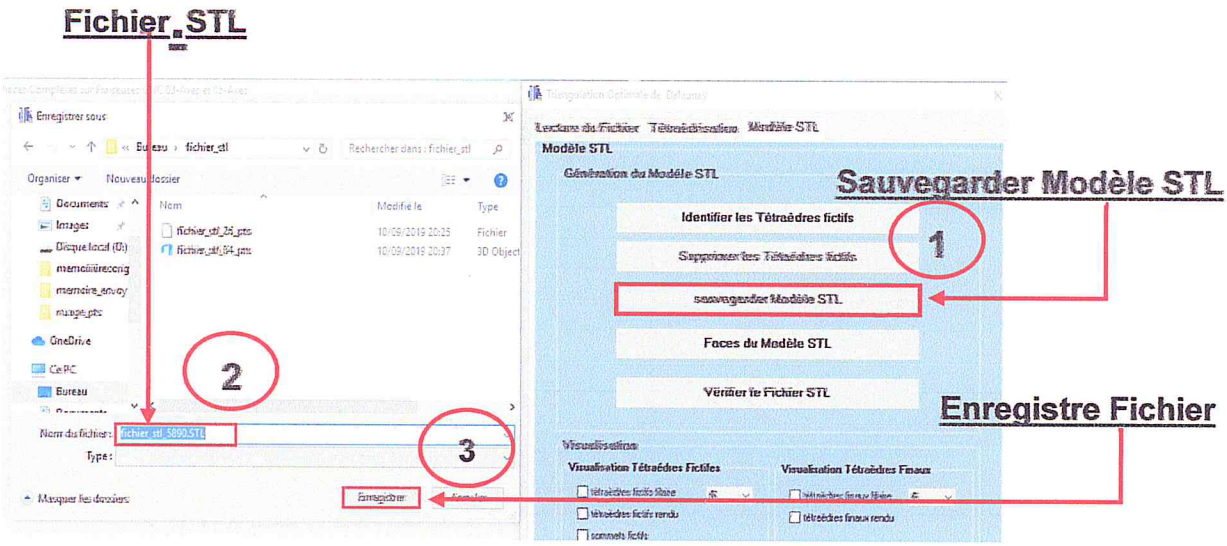
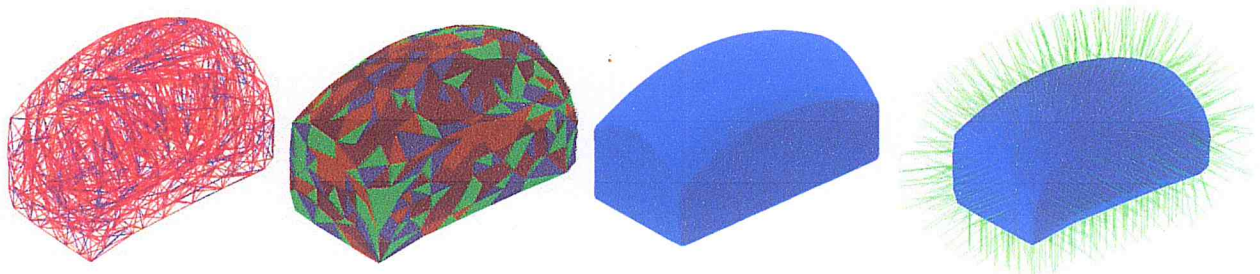


Figure 84. Onglet sauvegarder modèle STL .

```
solid surfaces
   facet normal   -0.32080112    -0.0033968507    -
0.94714049
      outer loop
         vertex   236.93377    298.49344    -577.80492
         vertex   236.78414    303.86659    -577.77351
         vertex   240.9417    304.29155    -579.18322
      endloop
   endfacet
   facet normal   -0.31423947    -0.0084473927    -
0.94930617
      outer loop
         vertex   236.93377    298.49344    -577.80492
         vertex   241.01737    299.20661    -579.16302
         vertex   240.9417    304.29155    -579.18322
      endloop
   endfacet
   facet normal   -0.3204047    -0.0076382786    -
0.94724996
      outer loop
         vertex   236.78414    303.86659    -577.77351
         vertex   240.9417    304.29155    -579.18322
         vertex   240.86909    309.1777    -579.19806
      endloop
   endfacet
   facet normal   -0.31409861    -0.0061441704    -
0.94937048
```

Figure 85. Sauvegarde d'un fichier STL.



a. Tétraèdres filaire. b. Tétraèdres rendu. c. model STL obtenu. c. normales

Figure 86. Visualisation de la tétraédrisation finale.

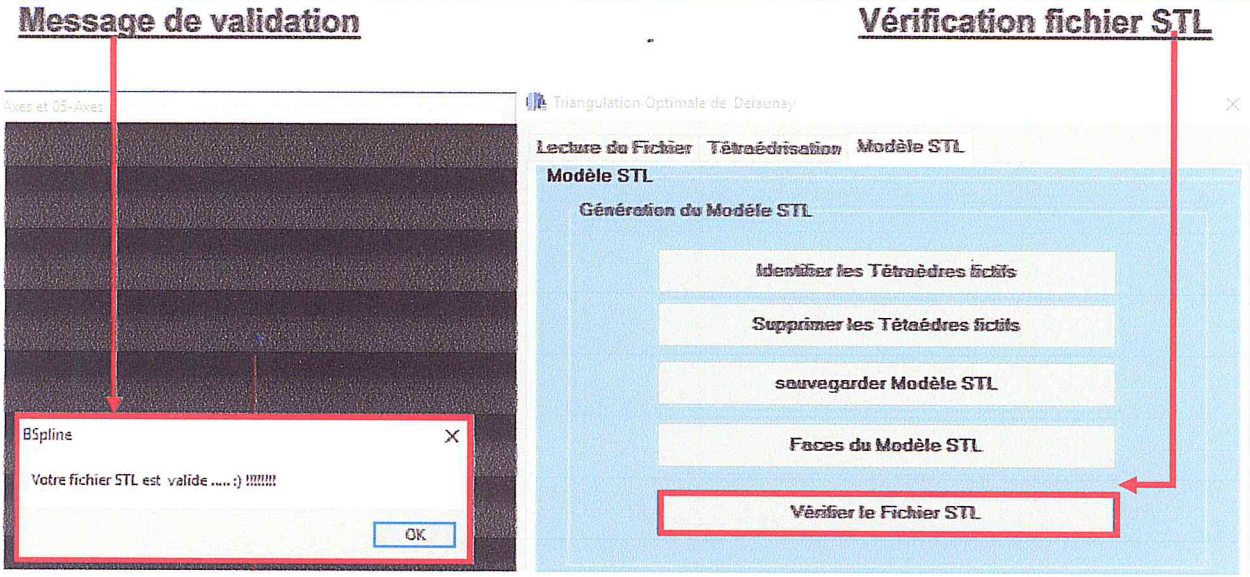
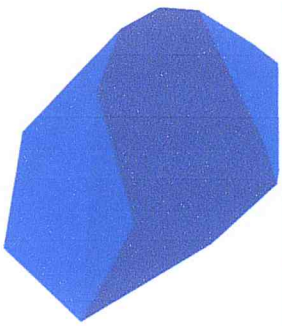
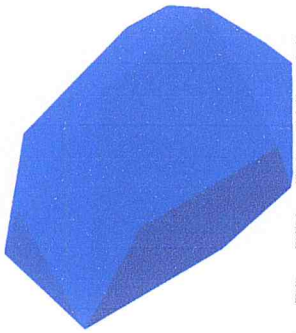
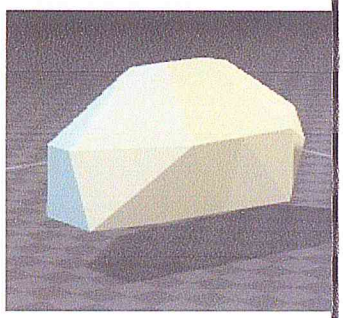
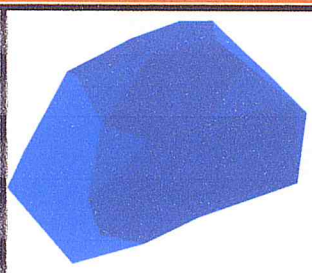
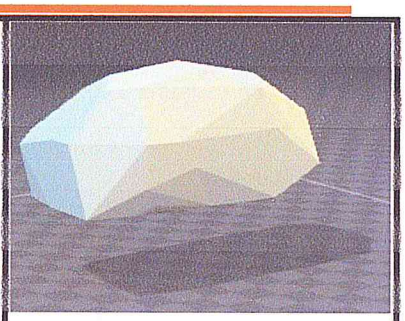
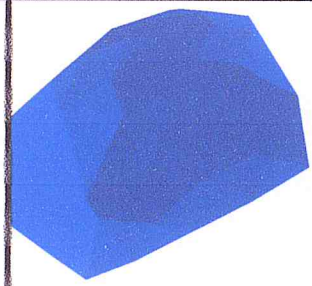
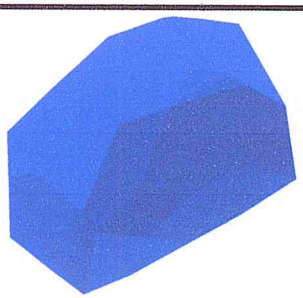
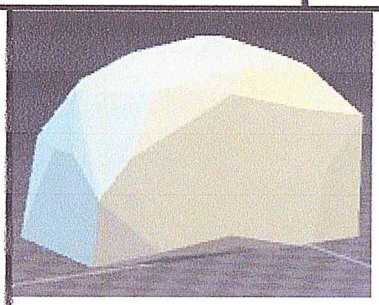
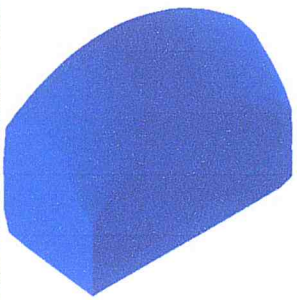
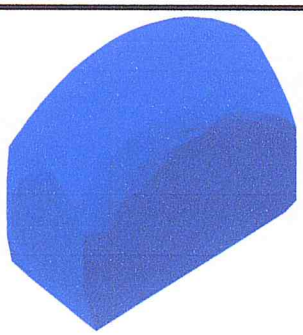
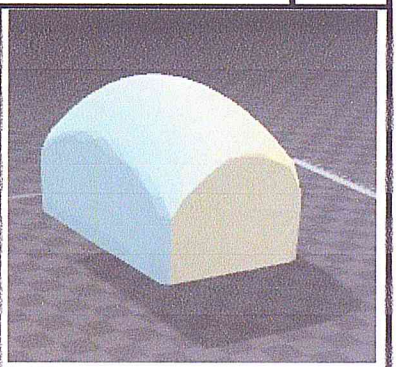


Figure 87. Fichier STL valide.

3. Etude comparative

Afin de valider notre travail, une comparaison est faite entre les résultats obtenus par notre application et les résultats obtenus par une application développée antérieurement. Cette comparaison est appliquée sur plusieurs nuages de points de densités différentes. Les résultats sont présentés dans le tableau ci-dessous (Figure 88).

Nombre de points	**Modèle STL (non optimisé)**	**Modèle STL (optimisée)**	**Visualisation en 3D Builder**
26 points			
Temps d'exécution	**5.10 s**	**0.28 s**	Modèle STL (non optimisée)
GAIN		**94.50%**	

55 points			
Temps d'exécution	**6.96 s**	**0.58 s**	
GAIN		**91.66%**	
84points			
Temps d'exécution	**8.20 s**	**0.98s**	
GAIN		**88.04%**	
809points			
Temps d'exécution	**1h07mn 25 s**	**12min 42 s**	

GAIN		83%	

Figure 88. Tableau comparatif.

L'analyse des résultats obtenus, fait ressortir une différence remarquable dans les temps d'exécution pour tous les nuages de points traités. De plus, autant le nuage de points est dense, plus le modèle obtenu est similaire au modèle voulu avec un temps d'exécution élevé.

Afin de consolider la fiabilité de notre approche, d'autres tests avec des nuages de points plus dense sont effectués. Les résultats des tests sont résumés dans le tableau suivant (Figure 89).

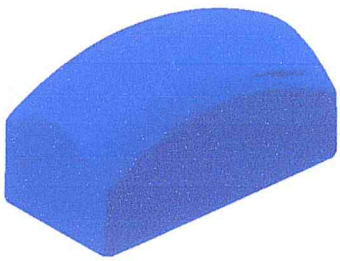
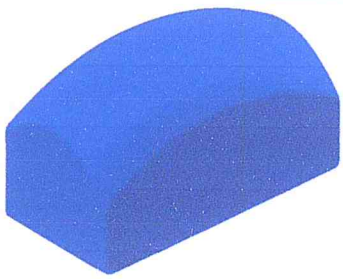
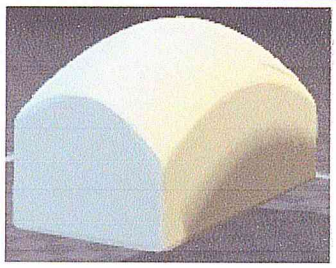
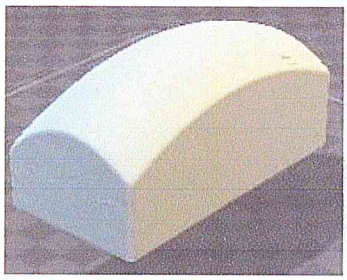
Nombre de points	**5890**	**14838**
Modèle STL		
Visualisation en 3D Builder		
Temps d'exécution	**4 mn 4.22 s**	**20 mn 58.00 s**

Figure 89. Tableau des tests.

Les résultats des différents test effectués donnent un aperçu sur les gains de temps de traitement de la triangulation de Delaunay pour des nuages de points de densités différentes. Ceci permet d'affirmer que le but assigné à ce travail est atteint.

Conclusion

Dans ce chapitre, nous avons présenté l'application développée en détaillant les différentes étapes du processus d'optimisation de la triangulation de Delaunay depuis la lecture du fichier de points jusqu'à la génération du fichier STL. Afin de valider notre approche, plusieurs exemples de nuage de points de densités différentes ont été traités.

Certe, la validation à travers des exemples de tailles plus ou moins modérées (quelque milliers de points) ne peut refléter l'intérêt de l'application dans le domaine de la reconstruction d'objet surtout pour les applications du réverse engineering, mais elle constitue un nouveau maillon dans l'environnement de reconstruction d'objets développé par l'équipe de CFAO du CDTA.

Conclusion générale :

Le projet que nous avons présenté dans ce mémoire consiste en la conception et le développement d'une application logicielle graphique et interactive de CAO. Ce travail est proposé par l'équipe CFAO de la Division Productique et Robotique du CDTA. Il s'articule autour de l'optimisation de la triangulation de Delaunay d'objet et à la génération du fichier STL associé et ce à partir d'un nuage de points 3D non structuré. Cette application utilise des techniques d'optimisation et d'approximation de nuages de points par des éléments géométriques élementaires (triangles en 2D) ou (tétraèdres en 3D) pour réaliser une triangulation de Delaunay en 2D ou 3D avec la méthode de flip.

Lors de la réalisation de ce projet, nous avons commencé par une étude bibliographique sur le processus de production de pièces mécaniques. Par la suite, nous avons mené une étude sur le processus du « Reverse Engineering » et sur les formats d'échange de données en particulier le format STL. Suite à cela, nous avons passé à la représentation d'objet 3D, en présentant les différentes méthodes concourant à la tétraédrisation de Delaunay. Finalement, nous avons présenté la conception et l'implémentation de l'approche adoptée et les solutions proposées ainsi que les tests et les validations.

Les principaux résultats obtenus par notre application logicielle sont les suivants :

- Lecture de n'importe quel nuage de points.
- Génération de tétraédrisation avec un (01) super tétraèdre englobant.
- Minimisation de la zone des voisinages.
- Localisation des tétraèdres à modifier en un temps optimal.
- Optimisation du temps d'exécution.
- Génération d'une triangulation 3D pour générer une tétraédrisation Delaunay.
- Génération d'un modèle STL à partir de d'un ensemble de tétraèdres obtenus de la triangulation de Delaunay.

En perspective, nous recommandons de traiter les thématiques suivantes :

1. Développement de nouvelles méthodes de traitement pour améliorer la qualité de la triangulation.
2. Reconstruction des frontières de l'objet en utilisant la technique « alpha shape ».
3. Intégration des cartes graphiques dans les calculs « GPU ».

Références bibliographiques

[1] Hasan Aladad. Conception du système de fabrication de pièces mécaniques en grand série : for-malisation de la configuration géométrique (enveloppe) et cinématique de Machine-Outil Reconfig-urable (MOR). Sciences de l'ingénieur [physics]. Arts et Métiers ParisTech, 2009. Français. NNT :2008ENAM0012. Pastel-00005257

[2] Mohamed Islem OUAMER ALI, Florent LAROCHE, Sébastien REMY, Alain BERNARD. (2017). Nouvelle méthodologie de rétro conception intégrée.url : aipprimeca2017.sciencesconf.org.

[3] Salam ALI. (2015). La Rétro-conception de composants mécaniques par une approche « concevoir pour fabriquer ». Université de technologie de Troyes.

[4]Mohamed Serraji, Energie éolienne Ingénierie de la conception énergie .NTS Inc-Ingénierie du logiciel PLM Montréal Québec, 2013.

url :https://www.ntsconsulting.ca/retroconception_fr.php.

[5]SAFAR Khansa & ZEKKARI Sara. (2017). Conception et Réalisation d'une Application pour la Reconstruction de Surface d'objet 3D à Partir d'un Nuage de Points par la Méthode de FLIP. Master 2 USDB.

[6] Bey Mohamed, Bendifallah Hassène, Tchantchane Zahida, N. Ghoulmi, O. Azouaoui. (2018). Triangulation d'un Nuage de Points 3D Non-Structuré par la Méthode de FLIP. Centre de Développement des Technologies Avancées – CDTA.

[7]Zouaoui, O & Sissani, D, (2016). Rapport conception et développement d'une application distribuée pour la reconstruction des modèles 03D à partir d'un nuage de points quelconque. Master 2 USDB.

[8] Challali, M. O., Belaidi, I., Mohammedi, K., Belaidi, A., & Ishiomin, G (2001). Application et adaptation de la triangulation de Delaunay pour la reconstruction de surfaces. Surface reconstruction using and adapting Delaunay triangulation.

[9]AMEDDAH Hacène. (2013). Modélisation CAO et Stratégies d'usinage pour la réalisation des surfaces à géométrie compliquée (Surfaces Libres). Université Hadj Lakhdar Batna

[10]B. Delaunay, « Sur la sphère vide », Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikhi Estestvennykh Nauk, 7 :793800, 1934

[11]Houman Borouchaki, Paul Louis George.Maillage, modélisation géométrique et simulation numérique, l'institut de physique du globe paris ,2007.

[12] Mounir Ait Mansour, M. Extraction d'éléments géométriques dans un nuage de points LiDAR terrestre : application aux relevés de façades. Sciences de l'ingénieur [physics]. 2014. url : https://dumas.ccsd.cnrs.fr/dumas-01164570.

[13] Gold, H. L. C. (2005). La modélisation de données océanographiques à l'aide du diagramme de Voronoï tridimensionnel. Revue internationale de géomatique. Volume (1).

[14]Beyler, J C & Isenmann, B. (2006, Mai). Génération de terrain et triangulation de Delaunay. Article de la triangulation de Delaunay.

url : http://fearyourself.developpez.com/tutoriel/jeu/Delaunay/.

[15]Hong-Tzong Yau, Ming-Tzong Lin, Meng-Shiun Tsai. Real-time NURBS interpolation using FPGA for higt speed motion control. Elsevier CAD. Vol. 38,2006.S.Kumar et D. Manocha. Efficient rendering of trimmed NURBS surfaces, Computer Aide

Design.27 (7): 509-521, 1995.

[16]Jiang, Z., Jaillet, F., & Zara, F,(2010). Reconstruction et complétion de maillages sous contraintes. Laboratoire d'Informatique en Image et Systèmes d'information LIRIS UMR 5205 CNRS/INSA de Lyon, Université Claude Bernard Lyon 1. url : https://liris.cnrs.fr/page-membre/fabrice-jaillet .

[17] site officiel : https://www.embarcadero.com/.

Annexe A

Diagrammes de séquence

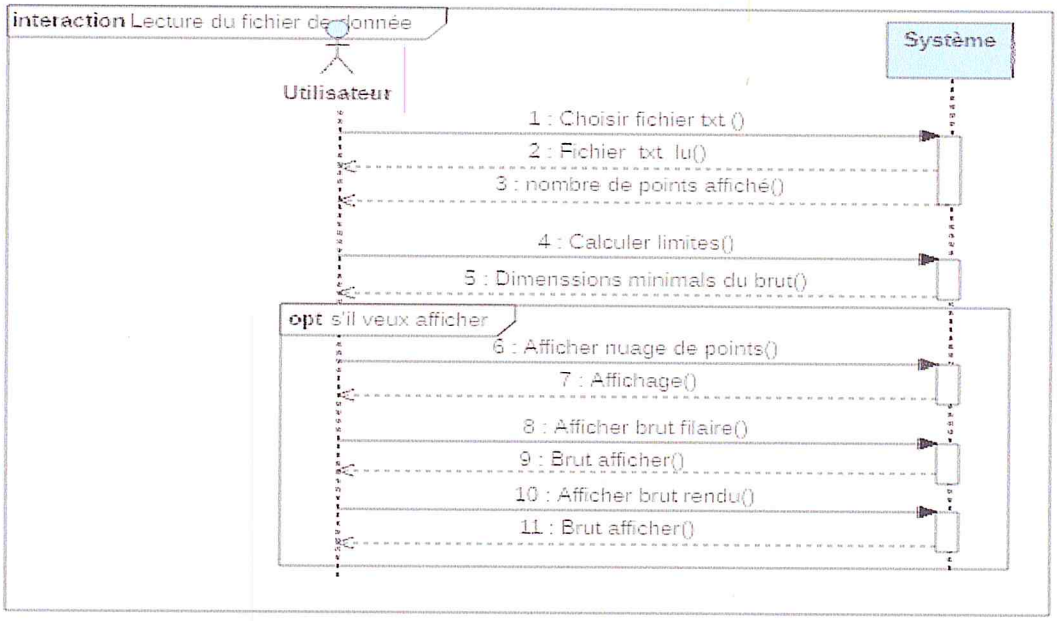


Figure A.1. Diagramme de séquence relatif au traitement la lecture du fichier.

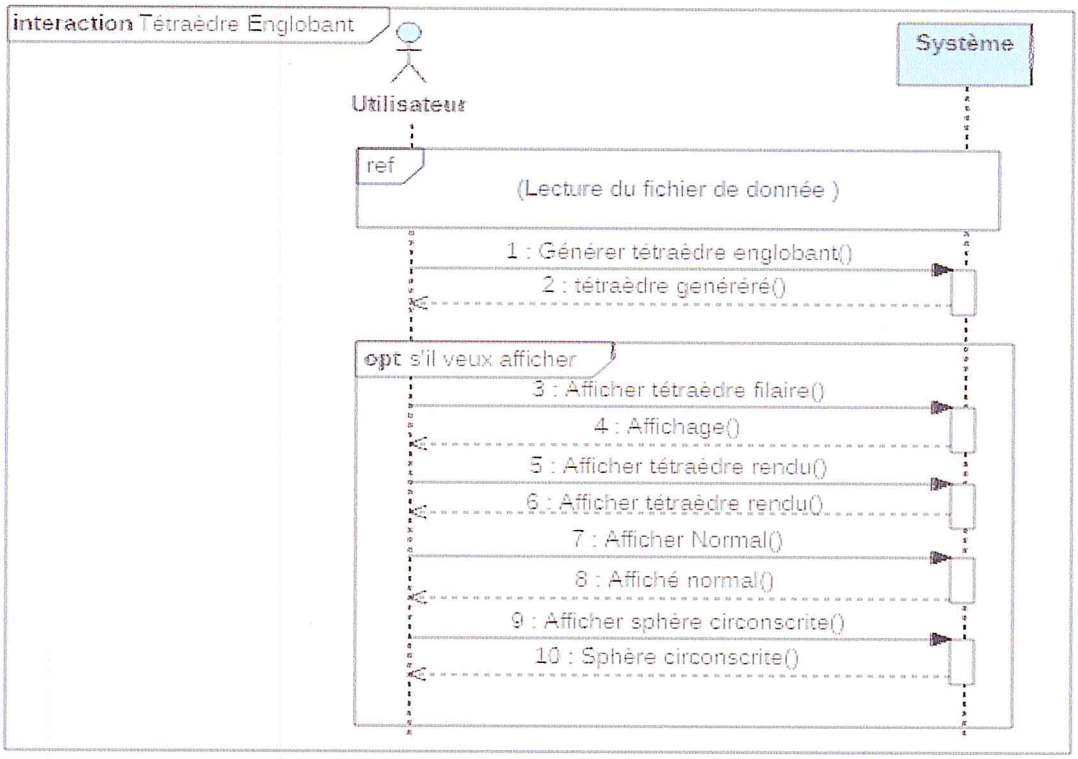


Figure A.2. Diagramme de séquence relatif à la génération du tétraèdre englobant.

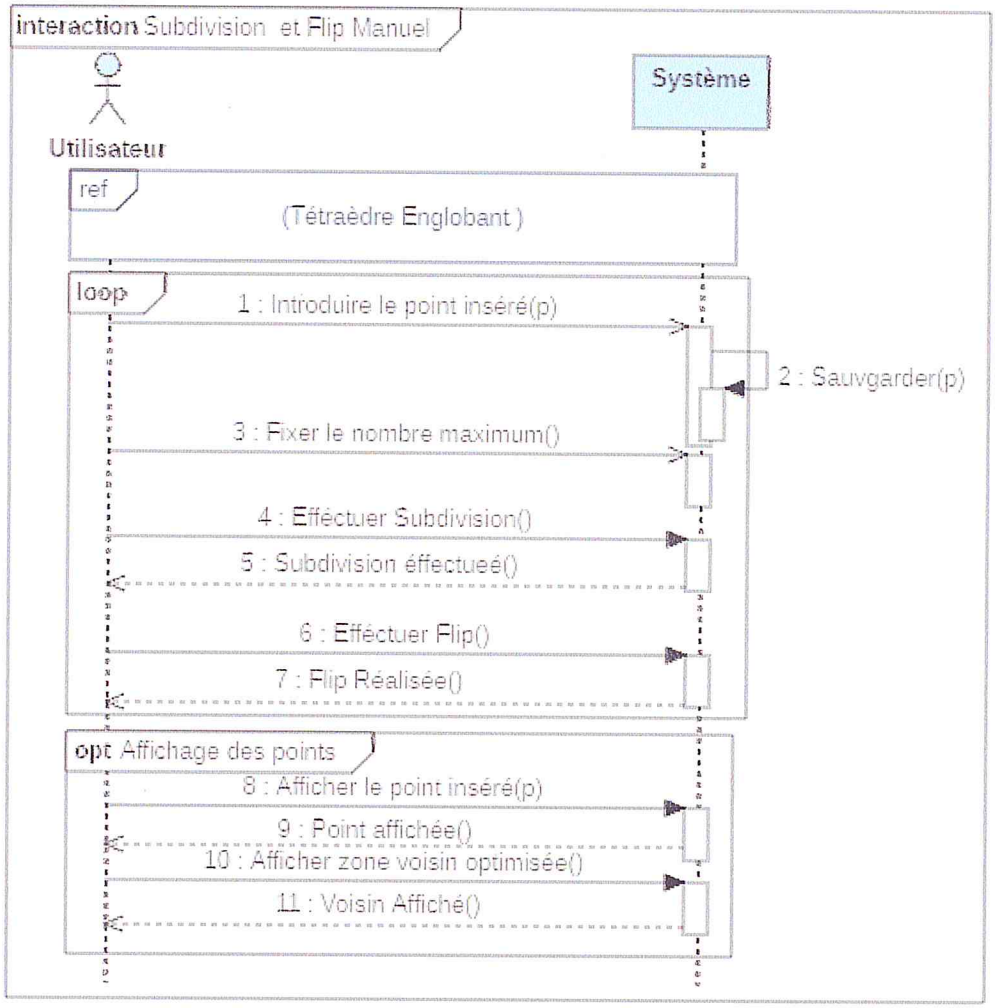


Figure A.3. Diagramme de séquence relatif à la tétaèdrisation de Delaunay manuelle.

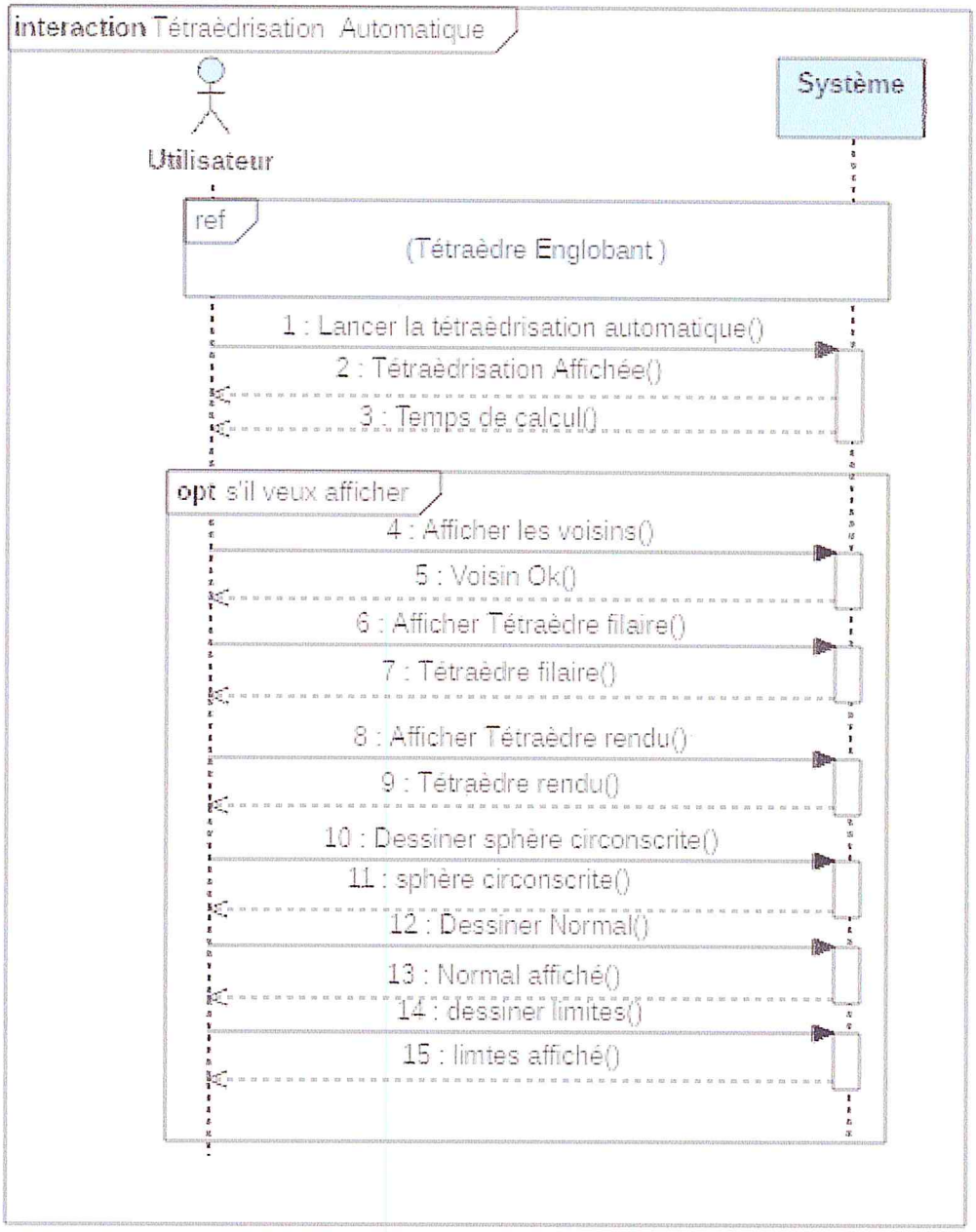


Figure A.4. Diagramme de séquence de la triangulation automatique de Delaunay.

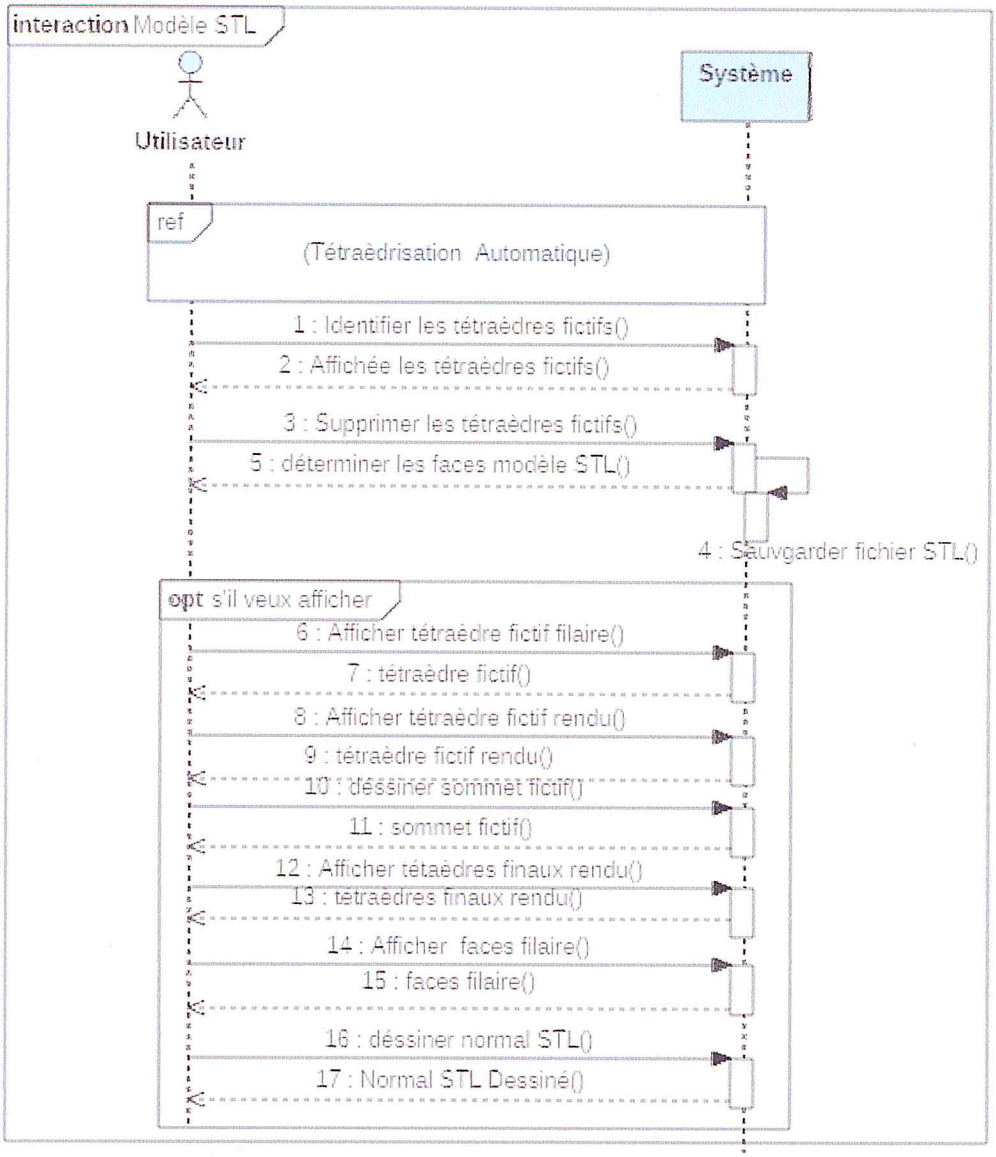


Figure A.5. Diagramme de séquence de traitement de la génération du modèle STL.