

MA - 004 - 453 - 1

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

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

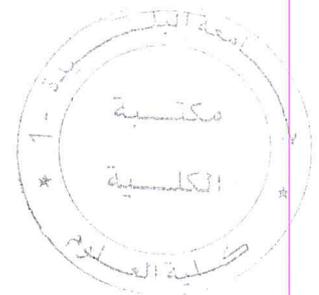


## MEMOIRE DE FIN D'ETUDES

Pour l'obtention

D'un diplôme de master en informatique.

Spécialité : Génie des systèmes informatiques



### THÈME :

**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.**

Réalisé par :

SAFAR Khansa

ZEKKARI Sara

Soutenu le : , devant :

Mr KAMECHE Abdallah Hichem

Mr BEY Mohamed

Mme TCHANTCHANE Zahida

Mr BENHBILES Halim

Mr CHERIF-ZAHAR Sid-Ahmed Amine

Promoteur

Encadreur

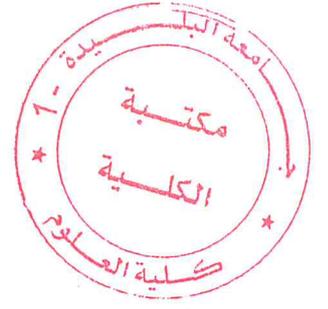
Encadreur

Président

Examinateur

2016/2017

MA-004-453-1



More we learn, more  
we discover our  
ignorance

Percy Shelley

## *Remerciements*

*Tout d'abord, nous tenons à remercier DIEU le tout puissant de nous avoir donné la force, le courage et la patience pour accomplir notre travail.*

*Nous tenons à remercier nos très chers parents pour leur tendresse, leur encouragement et leur soutien moral et matériel dans le but d'assurer notre réussite.*

*Au terme de ce travail, nous tenons à remercier vivement notre promoteur Mr **KAMECHE Abdallah Hichem** pour sa compréhension et sa gentillesse, et un grand merci à nos encadreurs Mr **BEY Mohamed**, Mme **TCHANTCHAN Zahida**, Mr **BENDIFALAH Hassen** pour leur encadrement, orientation, la confiance et la patience qu'ont constitués un apport considérable,*

*Khansa & Sarah*

# *Dédicace Khansa*

*Je dédie ce mémoire de fin d'étude*

*A*

*Ma très chère maman que j'aime énormément*

*Ma sœur Meryouma*

*Ma sœur Amina*

*Ma sœur Imene mon âme sœur*

*Mon frère Abdenour*

*Mes amis Raouf et Asma.*

*Que dieu vous protège et vous bénisse.*

*Safar Khansa*

## *Dédicaces Sarah*

*Je dédie mon travail présenté dans ce mémoire à qui je ne puisse montrer mon amour et ma gratitude :*

*Mes chers parents :*

*Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi, Je ferai de mon mieux pour rester un sujet de fierté à vos yeux avec l'espoir de ne jamais vous décevoir. Que ce modeste travail, soit l'exaucement de vos vœux tant formulés et de vos prières quotidiennes.*

*A mes chères sœurs : Hadjer Manel Yousra et à mon beau frere Mohamed Lamine à qui je souhaite pleine de réussite.*

*Et bien sure à mon très cher neveu Elmehdi et à ma nièce Leyane à qui je souhaite une longue vie pleine de bonheur et de succès.*

*A mes grands-mères, mes oncles et leurs femmes et mes tantes et leurs maris*

*A tous mes cousins et cousines*

*A mon binôme Khansa et à sa famille.*

*A mes chères amies qui m'ont toujours soutenu : Ibtissem, Besma, Imene, Soumia, Dounia et Soumia.*

*A tous mes amis avec qui j'ai partagé des moments inoubliables.*

*Sarah Fekhari*

## *Résumé*

La reconstruction 3D, désigne la technique qui permet d'obtenir une représentation en trois dimensions d'un objet à partir d'un nuage de points dont les coordonnées 3D sont saisies par un dispositif de mesure prise sous différents points de vue de l'objet. Ces objets sont utilisés dans divers domaines. Pour résoudre ce problème, le nuage de points est approximé par des éléments géométriques simples. Le but est de reproduire en sortie une surface approchant au mieux la forme de la surface physique de l'objet échantillonné. La reconstruction des modèles 3D avec une bonne précision est une tâche ardue, coûteuse en temps. Dans ce travail nous nous intéresserons à la génération du modèle STL approximant la peau externe d'un objet de n'importe quelles formes à partir d'un nuage de points en utilisant la méthode de FLIP tridimensionnel.

## *Abstract*

3D reconstruction refers to the technique used to get a three-dimensional representation of an object from a cloud of points whose 3D coordinates are captured by a measuring device taken from different views of the object. These objects are used in various fields. To solve this problem, the point cloud is approximated by simple geometric elements. The aim is to reproduce at the output a surface that best approximates the shape of the physical surface of the sampled object. Reconstructing 3D models with good accuracy is a daunting, time-consuming task. In this work we will focus on the generation of the STL model approximating the outer skin of an object of any shape from a scatter of points using the three-dimensional FLIP method.

## Sommaire

Introduction générale:.....	13
Chapitre I : Modélisation des surfaces et reconstruction d'objets .....	16
Introduction .....	17
1. Représentation des surfaces.....	17
1.1 Définition des surfaces complexes .....	18
2. Méthode de modélisation .....	19
2.1. Interpolation surfacique.....	19
2.2 Approximation de la surface.....	19
2.3 Surface paramétrique .....	20
4. Reverse Engineering .....	20
3.1. Définition:.....	20
3.2. Pourquoi utiliser le Reverse Engineering ? .....	21
3.3 Domaines d'utilisation.....	21
3.4. Phase de rétro-conception.....	21
4. Méthodes de reconstruction d'objets à partir d'un nuage de points:.....	22
4.1 Numérisation de l'objet :.....	23
4.2 Recalage : .....	23
4.3 Décimation : .....	23
4.4 Segmentation : .....	24
4.5 Définition continue de l'objet:.....	25
5. Méthode de modélisation: .....	25
5.1. Modélisation par maillage: .....	25
5.2. Modélisation géométrique: .....	25
6.1 Scanner 3D: .....	25
6.2. Machine à mesurer tridimensionnelle (MMT) .....	26
7. Format d'échange de données.....	27
7.1 Format d'échange de données «STL» .....	27
7.2 Pourquoi STL?.....	28
7.3 Propriétés [9] .....	28
7.4 Avantages .....	29
7.5 Logiciels supportant le format STL :.....	30
Conclusion.....	30
Chapitre II : Méthode de FLIP .....	31
Introduction .....	32
1. Triangulation de Delaunay .....	32
1.1. Diagramme de Voronoi .....	32
1.1.1 Propriétés diagramme de Voronoï 3D: .....	33
1.2 Définition de la triangulation de Delaunay: .....	33
1.2.1. Propriétés de Delaunay 2D: .....	34
1.3. Triangulation Delaunay en 3D: .....	35
2. Méthode d'implémentation de Delaunay:.....	35

2.1. Méthode Destruction-Construction :	35
4.2. CRUST :	36
4.3. DeWall :	37
4.4. Algorithme de flip en 2D :	37
5. Méthode de Flips en 3D:	38
5.1. Structures de données utilisées pour flip :	38
5.1.1. Piles:	38
5.1.2. Pointeurs:	39
5.1.3. Structure DAG :	39
5.1.4. Structure vectorielle	40
5.2. Processus de la triangulation Delaunay 2D utilisant flip :	40
5.3. Problème de légalité des arrêtes :	42
5.4. Processus de la triangulation Delaunay 3D	42
5.4.1. A partir d'une triangulation 3D quelconque	43
5.4.2. A partir d'un nuage de points	44
3.2.1 Problème de légalité 3D :	45
Conclusion :	46
Chapitre III: Conception de l'application	48
Introduction:	49
1. Architecture globale de l'application:	49
2. Problématique	49
3. Vérification du fichier	50
4. Simplification du nuage de points	50
4.1. Création des cellules et des patchs	51
4.1.1. Création des cellules	52
4.1.2 Affectation des points aux cellules	53
4.1.3. Création des patchs	53
4.2. Suppression des points redondants	54
4.2.1 Simplification des points dans un patch :	54
4.2.2. Simplification inter-patch:	55
4.2.3. Simplification inter-cellules	56
5. Triangulation de Delaunay	56
5.1 Triangulation à partir de 12 tétraèdres :	57
5.2 Création d'un tétraèdre englobant :	58
5.3 Caractéristiques d'un tétraèdre :	58
5.4 Insertion d'un point:	59
5.4.1 Identification du tétraèdre contenant le point inséré:	60
5.5 Subdivision du tétraèdre (Split)	61
5.5.1 Point inséré est confondu avec un des sommets du tétraèdre	61
5.5.2 Point inséré est sur un des segments du tétraèdre	61
5.5.3 Point inséré est sur une des faces du tétraèdre	62
5.5.4 Point inséré est à l'intérieur du tétraèdre	62
5.6 Vérification des critères de Delaunay	62

5.7 Réalisation de FLIPs.....	63
5.7.1 Cône test.....	63
5.7.2 FLIP 2-3 .....	64
5.7.3 FLIP 3-2 .....	65
5.7.4 FLIP 4-4 .....	65
6. Suppression des points fictifs: .....	66
7. Génération du fichier STL:.....	66
Conclusion.....	67
Chapitre IV : Test et Réalisation .....	68
Introduction : .....	69
1. langage utilisé :.....	69
1.1. Builder C++ : .....	70
1.2 C++ Builder 6 : .....	70
2. OpenGL .....	70
2. Présentation des fenêtres .....	71
2.1 Présentation de l'application développée .....	71
2.1.1 Lecture du fichier du nuage de points .....	72
2.1.2 Subdivision du brut .....	72
2.1.3 Simplification.....	74
2.1.4 Première tétraédrisation .....	75
3.1.5 Deuxième tétraédrisation .....	75
3. validation: .....	66
3.1 Lecture nuage de point .....	77
3.2 Subdivision: .....	78
3.3 Simplification: .....	80
3.4 Première Tetraedrisation: .....	81
3.5 Deuxième Tetraedrisation: .....	82
4. Etude comparative : .....	83
Conclusion :.....	84
Conclusion générale: .....	85
Références bibliographiques: .....	87

## Liste des figures

Figure 1: Différence entre une surface simple et une surface complexe.....	18
Figure 2 : Interpolation de la surface.....	19
Figure 3 : Approximation de la surface.....	19
Figure 4 : Localisation d'un point sur une paramétrique.....	20
Figure 5 : Objet numérisé.....	23
Figure 6 : Objet ayant subi une décimation.....	24
Figure 7 : Segmentation d'un objet.....	24
Figure 8 : Scanner avec contact.....	26
Figure 9 : Scanner sans contacte.....	26
Figure 10 : Machine a mesuré tridimensionnelle.....	26
Figure 11 : Paramètres d'un triangle.....	28
Figure 12 : Règle d'autorité.....	29
Figure 13 : Triangulation d'un nuage de points.....	32
Figure 14 : Diagramme de Voronoï.....	33
Figure 15 : Triangle entouré de cercle circonscrit.....	34
Figure 16 : Propriété des angles.....	34
Figure 17 : Arête illégale.....	34
Figure 18 : Arête légale.....	34
Figure 19 : Tétraèdre.....	35
Figure 20 : sphère circonscrite.....	35
Figure 21 : Méthode Destruction-Construction.....	36
Figure 22 : Un côté illégal (gauche), le FLIP (droite).....	38
Figure 23 : Structure de pile.....	38
Figure 24 : Structure de donnée avec tétraèdre.....	39
Figure 25 : Evolution du DAG lors d'un split.....	39
Figure 26 : Recherche dans le DAG du triangle contenant le point P.....	40
Figure 27 : Triangle fictif englobant 8 points à trianguler.....	40
Figure 28 : Insertion du point p5.....	41
Figure 29 : Triangle illégal.....	41
Figure 30 : Arête illégale.....	41
Figure 31 : Triangulation Delaunay.....	42
Figure 32 : Tetraédrisation quelconque par la méthode de FLIP.....	43
Figure 33 : étapes de la triangulation Delaunay 3D par la méthode de FLIP.....	44
Figure 34: Différents cas de FLIPs tridimensionnels.....	45
Figure 35: « cône test ».....	46
Figure 36 : Organigramme de l'Architecture globale.....	49
Figure 37 : Organigramme de vérification du fichier.....	50
Figure 38 : Organigramme de la simplification du nuage de points.....	51
Figure 39 : Organigramme de création des cellules et des patches.....	51
Figure 40 : Organigramme de création des cellules.....	52
Figure 41 : Calcul du brut.....	52
Figure 42 : Création des cellules.....	53
Figure 43 : Affectation des points aux cellules.....	53
Figure 44 : Organigramme de la création des patches.....	53
Figure 45 : Processus de création des patches.....	54

Figure 46 : Organigramme de simplification du nuage de points .....	54
Figure 47 : Processus de simplification d'un patch.....	55
Figure 48 : Simplification inter patches .....	56
Figure 49 : Processus de simplification inter cellules .....	56
Figure 50 : Organigramme de la triangulation de Delaunay .....	57
Figure 51 : Création d'un des douze tétraèdres.....	57
Figure 52 : Sommets d'un tétraèdre. ....	58
Figure 53 : Normales d'un tétraèdre. ....	58
Figure 54 : Sphère circonscrite, rayon et centre de la sphère.....	59
Figure 55 : Représentation d'un tétraèdre voisin. ....	59
Figure 56 : Représentation du brut englobant. ....	59
Figure 57 : Organigramme d'insertion d'un point .....	60
Figure 58 : Point inséré sur le sommet .....	61
Figure 59 : Split pour un point inséré sur le segment d'un tétraèdre .....	61
Figure 60 : Split pour un point inséré sur la face d'un tétraèdre .....	62
Figure 61 : Split pour un point à l'intérieur du tétraèdre.....	62
Figure 62 : Cas d'un point à l'intérieur de la sphère circonscrite .....	63
Figure 63 : Cône formé par trois faces du tétraèdre .....	63
Figure 64 : Les trois cas de position du point 5 par rapport au cône. ....	64
Figure 65 : FLIP convexe '2-3' .....	65
Figure 66 : FLIP non convexe '3-2'.....	65
Figure 67 : FLIP point sur le plan d'une des faces du tétraèdre '4-4'.....	66
Figure 68 : Organigramme de génération du fichier STL .....	67
Figure 69 : Onglets de la fenêtre principale. ....	71
Figure 70 : Onglets de la fenêtre principale. ....	72
Figure 71 : Fenêtre première étape de simplification.....	73
Figure 72 : Fenêtre de simplification. ....	74
Figure 73 : Fenêtre de la première tétraédrisation.....	75
Figure 74 : Fenêtre de la deuxième tétraédrisation. ....	76
Figure 75 : La pièce scannée. ....	76
Figure 76 : Résultat après le scan.....	77
Figure 77 : Lecture du fichier de nuage de points.....	77
Figure 78 : Visualisation du nuage.....	78
Figure 79 : Première subdivision spécification le nombre de cellules.....	78
Figure 80 : Visualisation de toutes les cellules. ....	79
Figure 81 : Visualisation d'une seule cellule.....	79
Figure 82 : Fenêtre de création des patches. ....	79
Figure 83 : Visualisation des patches.....	80
Figure 84 : Test de simplification avec 18 mm. ....	80
Figure 85 : Nuage après simplification. ....	81
Figure 86 : Onglets « première tetraedrisation » choix d'un seul tetraedre. ....	81
Figure 87 : Visualisation du tetraèdre englobant le nuage de points. ....	82
Figure 88 : Visualisation de la tétraédrisation finale.....	82
Figure 89 : Sauvegarde d'un fichier STL. ....	83
Figure 90 : Tableau comparatif entre les différentes distances de simplification. ....	83

### **Introduction générale:**

Dans des secteurs d'activités tels que l'automobile, l'aéronautique, l'industrie du moulage, ...etc., la compétitivité conduit à l'élaboration de produits au design toujours plus novateur et de qualité croissante. Ces produits de formes complexes sont obtenus selon la nature des matériaux par des procédés de moulage, d'injection, d'emboutissage, d'usinage, ...etc. ce qui impose la réalisation des outillages associés.

L'adéquation entre le produit réalisé et la géométrie de la forme définie par le concepteur dépend de l'aptitude de chacune des activités du processus de conception et de fabrication à modéliser ou produire la géométrie attendue. En effet, les techniques de construction des surfaces restent limitées et ne permettent pas toujours de réaliser les formes souhaitées par le concepteur ainsi que les raccordements entre les surfaces. L'utilisateur est tributaire des fonctionnalités du logiciel mis à sa disposition. La forme résultante ne correspond donc pas forcément à l'intention du concepteur mais est la plus proche représentation géométrique que le modéleur permette d'obtenir. En outre, les spécifications fonctionnelles ne sont pas exprimées en tant que telles, seule la géométrie les satisfaisant est définie dans le modèle CAO.

En ce qui concerne les surfaces complexes, leur construction repose sur l'association de surfaces par interpolation ou approximation d'éléments géométriques simples, ou bien des nuages de points. L'utilisation de nuages de points ne garantit pas la définition univoque d'une surface si la technique adoptée n'est pas adéquate. Ceci est dû à la propagation des erreurs générées lors de la digitalisation ou lors de la reconstruction des modèles des surfaces.

### **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 la « Reconstruction de surfaces d'objets 3D à partir d'un nuage de points irrégulier ». Ce concept a été introduit pour pallier aux problèmes de réalisation de produit dont le modèle géométrique n'existe plus ou impossible à reconstruire.

## Introduction générale

---

L'objectif assigné à ce travail est donc la mise en place d'une application logicielle graphique et interactive pour la reconstruction d'objet en trois dimensions 3D en utilisant la méthode de Flip.

Autrement dit, étant donné un nuage de points aléatoires issu de la digitalisation d'un objet physique, il s'agit de proposer une démarche pour la reconstruction d'objet 3D qui soit fidèle au modèle d'origine en utilisant des éléments géométriques élémentaires « tétraèdres » en trois dimensions 3D.

Afin de remédier aux problèmes cités ci-dessus, les objectifs assignés à cette étude sont les suivants :

- Remédier aux problèmes de la modélisation en deux dimensions pour les objets complexes.
- Rechercher des approches réalisables et efficaces pour une modélisation en trois dimensions.
- Générer une triangulation 3D (modèle STL) approximant la peau extérieure d'objets de n'importe quelles formes à partir de nuages de points quelconques.
- Proposition d'une méthodologie permettant la réduction des temps de traitement.

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

Le Présent mémoire est structuré comme suit :

Dans une présentation du contexte et des objectifs de 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 sont présentés dans le premier chapitre.

L'intérêt porté au modèle STL résulte du fait que ce travail s'insère dans le cadre d'un projet de recherche initié par l'équipe CFAO du CDTA et qui consiste à utiliser ce modèle dans le processus du Reverse Engineering et dans le processus de fabrication de pièces de formes gauches sur des fraiseuses numériques 05-axes.

Le deuxième chapitre est réservé à la présentation de l'état de l'art de la reconstruction d'objets 3D et quelques méthodes plus performantes. Par la suite, l'accent est mis sur la méthode du Flip qui est utilisée comme support pour développer notre approche.

Dans le troisième chapitre, nous développons une modélisation explicite de notre approche en développant tous les algorithmes utilisés.

## Introduction générale

---

Dans le quatrième chapitre, l'implémentation informatique de l'application est présentée avec des exemples de validation du travail effectué.

Pour achever ce travail, une conclusion et des éventuelles perspectives de ce travail seront présentées.

# Chapitre I

## Modélisation des surfaces et reconstruction d'objets

## Introduction

Depuis quelques années, les modèles géométriques tridimensionnels sont de plus en plus omniprésents. Ils sont apparus avec le développement de l'infographie et de la vision tridimensionnelle, qui ont permis une modélisation 3D d'objets de formes complexes. Ces modèles peuvent être créés par des systèmes de modélisation (logiciels de Conception Assistée par Ordinateur « CAO, ...etc.) ou obtenus grâce à des procédés de numérisation tridimensionnelle d'objets physiques. Ils sont généralement représentés sous forme de maillages surfaciques (surfaces linéaires par morceaux composées de polygones) ou volumiques (représentation de l'objet par un ensemble de polyèdres). Ces modèles peuvent aussi être décrits par des nuages de points, des formes à pôles (courbes de Bézier, NURBS, ...etc.) ou des surfaces implicites [15].

Les systèmes de conception et de fabrication assistés par ordinateur « CFAO » sont utilisés dans diverses industries afin de créer des objets physiques à partir de modèles numériques, cependant la rétro-conception ou Reverse Engineering c'est l'opération inverse qui consiste à conférer un modèle numérique à un objet physique [25].

L'objectif final du processus du Reverse Engineering est la reconstruction du modèle informatique « modèle CAO ». Cependant, cette étape est très importante, fastidieuse et consommatrice de temps et dépend de l'expérience du concepteur.

## 1. Représentation des surfaces

Les surfaces complexes, appelées aussi surfaces gauches, sont caractérisées par une topologie complexe composée d'un ensemble de surfaces de formes concaves, convexes, en selle de cheval ou à double courbure. Outre la complexité de l'opération d'usinage, s'ajoute la difficulté de considérer l'ensemble des formes complexes et du design. La production de ces surfaces est un domaine complexe et est sans cesse confronté à de nouveaux défis: des exigences de précision de plus en plus grandes, des formes de plus en plus complexes, une pression relative aux coûts croissants, une meilleure efficacité et une plus grande productivité.

Les modèles géométriques tridimensionnels sont de plus en plus présents dans notre vie quotidienne, ils sont apparus avec le développement de l'infographie. Leur but est de représenter les formes et les dimensions des objets d'une manière homogène. Cette représentation peut être filaire, surfacique ou volumique [1].

- **Modélisation filaire :**

Elle est basée sur les points et les segments où l'objet est défini par ces sommets et ces arêtes.

- **Modélisation surfacique :**

Elle prend en compte la notion de surface dont elle permet la représentation et la manipulation. Ils existent deux types de modélisation surfacique : par facettes planes et par surfaces gauches.

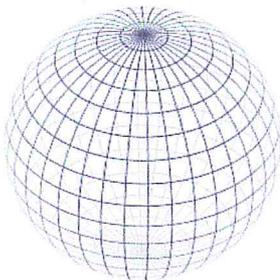
- **Facettes planes :** l'objet est représenté par des facettes élémentaires planes. La structure filaire est habillée avec ces facettes polygonales planes.
- **Par surfaces gauches :** cette méthode est employée quand la surface à définir est trop complexe pour être définie par des surfaces simples.

- **Modélisation volumique :**

Elle englobe les modélisations filaires et surfaciques et permet la représentation de l'objet dans l'espace en incluant la notion de matière.

### 1.1 Définition des surfaces complexes

Une surface complexe est une surface obtenue par raccordement de plusieurs surfaces simples avec des contraintes de continuité d'ordre zéro «  $C^0$  », un «  $C^1$  » ou deux «  $C^2$  ». Les surfaces complexes dites aussi gauches ne sont pas développables sur un plan tel qu'un parabolôïde et ne peuvent pas être décrites par des équations mathématiques. Elles peuvent être définies par des surfaces simples élémentaires ; planes, cylindriques, sphériques, coniques, ...etc. La surface à définir est alors approximée par des fonctions polynomiales paramétrées. L'utilisation de ces fonctions donne une excellente approximation de la surface réelle et permet aussi l'obtention de son profil apparent.



a) Surface simple



b) Surface complexe

**Figure 1:** Différence entre une surface simple et une surface complexe.

### 2. Méthode de modélisation

Il existe trois méthodes de modélisation des surfaces complexes basées sur l'utilisation des points :

#### 2.1. Interpolation surfacique

Cette méthode est basée sur la spécification d'un ensemble de points qui décrivent la forme de la surface désirée (Figure 2.a). Ensuite, une surface qui passe par ces points est construite par interpolation. La surface résultante (Figure 2.b) dépend de la méthode d'interpolation (Bézier, Bézier Rationnelle, B-Spline et NURBS), mais elle capture la forme désirée dans un sens. De plus, par modification de ces points et application de l'interpolation, la forme de la surface peut être manipulée interactivement [2].

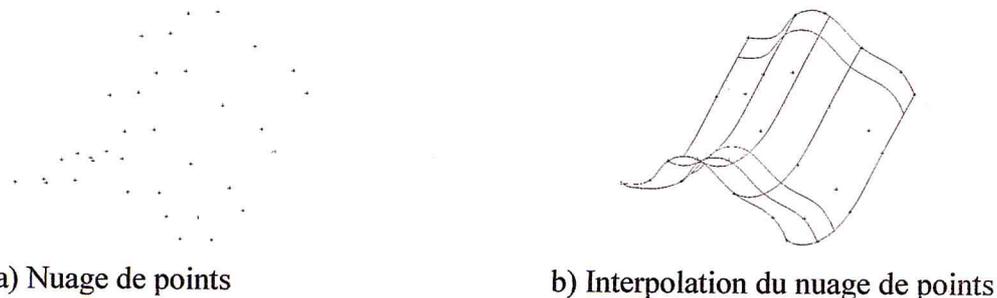


Figure 2 : Interpolation de la surface.

#### 2.2 Approximation de la surface

L'approximation de la surface est une application de modelage qui est fréquemment utilisée dans la conception de formes. Dans plusieurs situations, l'objet physique est disponible et son modèle numérique est recherché. Le processus de modelage commence par un ensemble de points (Figure 3.a) obtenus par mesure sur le modèle physique dans le but de déterminer la surface qui approxime le mieux tous les points mesurés mais qui ne passe pas nécessairement par ces points (Figure 3b) [2].

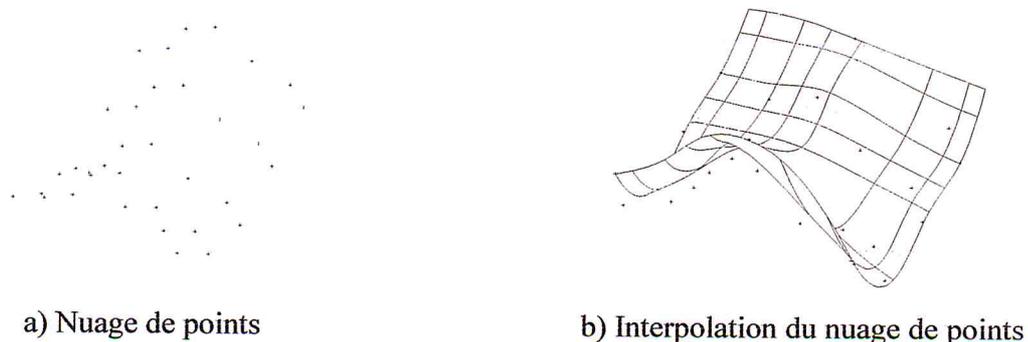


Figure 3 : Approximation de la surface

### 2.3 Surface paramétrique

Les surfaces paramétriques sont des surfaces continues définies par un réseau de contrôle représentant la forme de la surface. Généralement, les surfaces paramétriques ne sont pas utilisées séparément, mais plusieurs morceaux de surfaces sont joints ensemble pour former une surface plus complexe. Bézier, Bézier-rationnelle, B-Spline et NURBS sont les modèles paramétriques de surfaces les plus utilisées (Figure 4). [3]

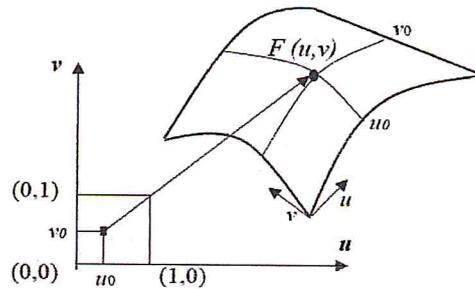


Figure 4 : Localisation d'un point sur une paramétrique

### 3. 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 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 [4].

#### 3.1. Définition:

Le Reverse Engineering (rétro-conception, rétro-ingénierie, ingénierie inverse) consiste à dupliquer un produit existant sans disposer de dessins, de documentations ou de modèles informatiques [5].

Dans une procédure d'ingénierie directe, le concepteur du produit élabore un dessin détaillant les modalités de fabrication d'un objet puis ce dernier est mis en production et fabriqué conformément au dessin de conception.

Le Reverse Engineering suit la procédure inverse. Dans un premier temps, les ingénieurs identifient les composants du système et leurs interactions. L'objet 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 de l'objet.

Après un dessin CAO est généré pour le système analysé. La phase suivante consiste à reproduire un modèle qui représente le produit original avec une précision conformément au dessin établi.

### 3.2. Pourquoi utiliser le Reverse Engineering ?

Le Reverse Engineering est utilisé est 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 3 D d'un modèle.
- Documentation architecturale, construction et de mesure.

### 3.3 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.

### 3.4. Phase de rétro-conception

La rétro conception se déroule en trois phases [1]:

- **Acquisition et digitalisation :**

Cette phase de numérisation consiste à prendre les mesures de la pièce considérée. Ces mesures sont effectuées en 3D et peuvent être réalisées sur tous types de pièces quel que soit leurs dimensions, de la petite pièce de quelques millimètres à une très grosse pièce de plusieurs dizaine de mètres. La mesure donne comme résultat un fichier reprenant le nuage de points très dense représentant la pièce étudiée. L'acquisition du nuage de points peut être repositionnée pour une plus grande précision par une mesure par photogrammétrie de la pièce, cette phase permet de connaître les dimensions de la pièce avec une grande précision.

- **Reconstruction d'objet 3D**

La reconstruction d'objets est une discipline relativement jeune qui intéresse un large spectre de domaines relevant aussi bien du génie mécanique, de vision artificielle, de robotique, du génie médical, du génie civil, de l'environnement ainsi que la génération de modèles CAO ou de la reconstruction par usinage [10].

## Modélisation des surfaces et reconstruction d'objets

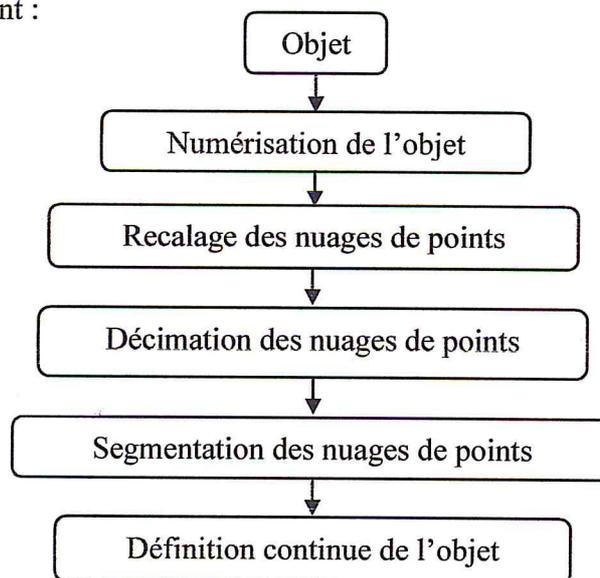
L'objectif de la reconstruction tridimensionnelle des surfaces consiste à générer un modèle géométrique à partir d'informations discrètes. Afin d'arriver à un minimum de précision en respectant la forme de l'objet représentent les applications majeures de la reconstruction d'objets ou Reverse Engineering [11].

Différents chercheurs ont considéré divers problèmes liés à la modélisation de l'objet à partir d'un nuage de points qu'il soit structuré ou non structuré. Au niveau de l'équipe CFAO du CDTA, une méthodologie basée sur le principe de la triangulation de Delaunay est proposée pour l'approximation du nuage de points non structuré. La démarche est inspirée de la méthode Bowyer-Watson et comporte quatre étapes:

1. Filtrage du nuage de points puis calcul des sommets fictifs et la génération du super tétraèdre.
2. Insertion point par point du nuage de points filtré puis recherche des tétraèdres contenant chaque point inséré suivie de modification de la triangulation.
3. Génération de la triangulation finale.
4. Génération du modèle STL.

### 4. Méthodes de reconstruction d'objets à partir d'un nuage de points:

Le processus de reconstruction d'objet 3D à partir d'un nuage de points non structuré comporte cinq étapes principales nécessaires à la génération du modèle STL, décrites par l'organigramme suivant :



**Algorithme 1:** processus de reconstruction d'objets

### 4.1 Numérisation de l'objet :

La mesure d'objets tridimensionnels se traduit par un ou plusieurs ensembles de point 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  $(x_i, y_i, z_i)$ . L'ensemble de ces triplets offre une définition de la ou des surfaces de l'objet mesuré sans indications concernant son type topologique ni sa géométrie. [12]

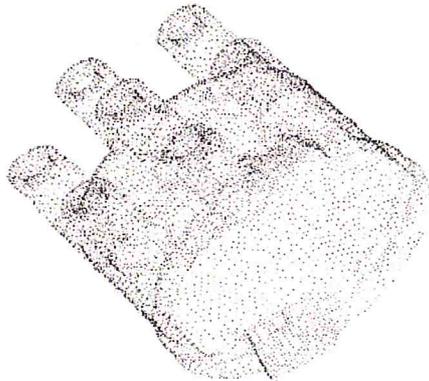


Figure 5 : Objet numérisé.

### 4.2 Recalage :

Un objet de forme complexe ne peut pas être numérisé en prenant en compte une seule face et donc, 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:** permet un recalage deux à deux. Deux ensembles de points sont recalés afin de former un seul. Ce dernier va être recalé avec le troisième et ainsi de suite.
- **Recalage globale:** permet de faire une correspondance simultanément plusieurs nuages de points en utilisant l'algorithme ICP « Iterative Closest Point ».

### 4.3 Décimation :

La décimation sert à simplifier le nuage de points pour le rendre manipulable sur ordinateur par diminution du nombre de ces points. Elle consiste à construire un polyèdre supporté par les points 3D et à trouver un polyèdre simplifié avec un nombre donné de nœuds. Cela est répété jusqu'à satisfaire une déviation donnée. Il est important de signaler que la densité de points dans les régions fortement courbées doit être plus grande par rapport aux endroits de faibles courbures [10].

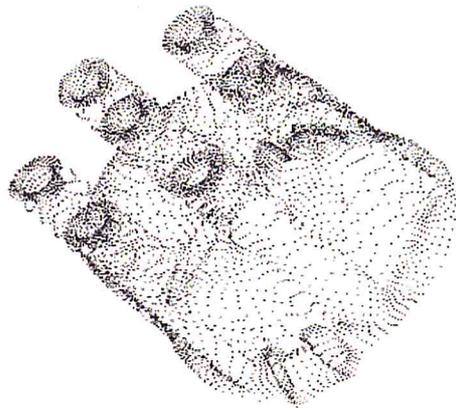


Figure 6 : Objet ayant subi une décimation.

### 4.4 Segmentation :

C'est une étape clé dans la reconstruction du modèle 3D. Elle identifie chaque sous ensemble comme entité indépendante. Il existe deux grandes familles de segmentation :

- Segmentation par reconnaissance de formes : ce type de segmentation estime les paramètres géométriques d'un objet (plan, cylindre,...) 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, ils sont testés au voisinage de la graine. Si les points voisins répondent aux critères, alors ils sont acceptés dans ladite surface segmentée, les critères à prendre en considération sont :
  - Critère de proximité : seuls les  $n$  points voisins peuvent être ajoutés à la surface graine. Le voisinage est défini par un rayon ' $r$ ' fixe ou par un nombre de points ' $n$ ' ;
  - Critère de planéité locale : une fois le plan local ajusté, la distance orthogonale de chaque point au plan est testé. Cette distance doit être inférieure à un certain seuil ;
  - Critère de normalité : les normales aux différents plans voisins sont testées. Leurs différences angulaires ne doivent pas dépasser un certain seuil, sinon la croissance s'arrête. [13]

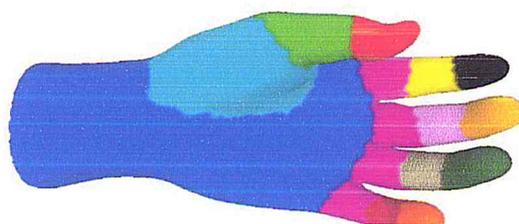


Figure 7 : Segmentation d'un objet.

### 4.5 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.

## 5. Méthode de modélisation:

Pour pouvoir générer le modèle continu de l'objet, il faut passer d'un nuage de points segmenté, vers une maquette numérique 3D. La maquette numérique peut être générée de deux manières : modélisation par maillage et modélisation géométrique.

### 5.1. Modélisation par maillage:

Le maillage consiste à relier les points du nuage entre eux afin de construire l'objet. Le modèle maillé est alors constitué de sommets issus directement des points du nuage et d'arrêtes reliant ces sommets.

### 5.2. Modélisation géométrique:

La modélisation géométrique est basée sur la construction des éléments identifiables et descriptibles d'un point de vue mathématique dans le nuage de points. Autrement dit, il s'agit de construire le modèle à l'aide de primitives géométriques. Cette modélisation a l'avantage de simplifier l'objet à sa plus pure représentation mathématique. La modélisation géométrique permet de trouver le modèle qui correspond au mieux à ce sous ensemble de points. Cette modélisation peut être très efficace si elle est précédée d'une segmentation qui identifiera les principales primitives [14].

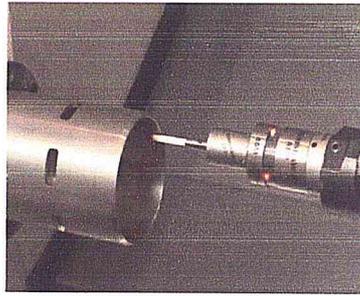
## 6. Machine utilisée dans le Reverse Engineering

**6.1 Scanner 3D:** est un dispositif qui analyse un objet du monde réel ou de l'environnement pour recueillir des données sur sa forme et son apparence (couleur par exemple). Les données recueillies peuvent ensuite être utilisées pour construire numériquement des modèles en trois dimensions utiles pour une grande variété d'applications.

#### ➤ Mode de scanning

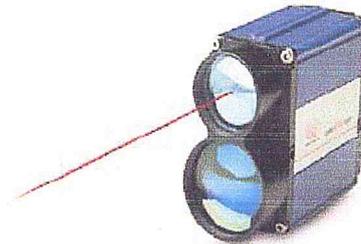
Pour scanner un objet, deux modes peuvent être utilisés.

- **Avec contact :** ces dispositifs utilisent des palpeurs de contact qui suivent automatiquement les contours d'une surface physique en fonction de la taille de la pièce numérisée. Les palpeurs sont des dispositifs tactiles utilisés pour enregistrer un point.



**Figure 8 :** Scanner avec contact.

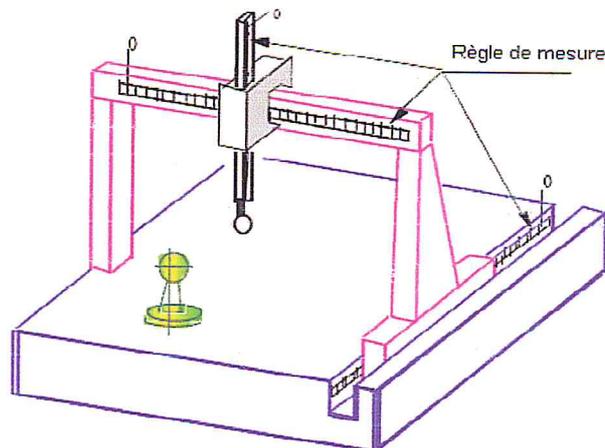
- **Sans contact :** les dispositifs sans contact utilisent des lasers et des caméras CCD « Charge Coupled Device » pour récupérer les données. Le temps de scanning avec ces dispositifs est relativement court.



**Figure 9 :** Scanner sans contacte.

### 6.2. Machine à mesurer tridimensionnelle (MMT)

Ces machines sont des instruments utilisés en métrologie dimensionnelle. Elles permettent d'obtenir les coordonnées des points mesurés (palpés) sur une pièce mécanique. Ces coordonnées permettent de vérifier la validité dimensionnelle de la pièce, de vérifier que les côtes sont respectées. De telles machines sont présentes dans toutes les industries mécaniques



**Figure 10 :** Machine à mesurer tridimensionnelle.

### 7. Format d'échange de données

Le système de Conception et de Fabrication Assistées par Ordinateur « CFAO », permet de dessiner et de réaliser des pièces qui seront ensuite assemblées pour en faire un produit.

L'idée générale de ce procédé est d'utiliser les capacités de l'ordinateur afin de dessiner la pièce en 2D, ou mieux en 3D pour ensuite la fabriquer grâce à des machines connectées directement à l'ordinateur. L'objectif de la « CFAO » est d'obtenir une extrême précision, de réaliser un gain de temps et de minimiser l'intervention humaine. [6]

La structure des systèmes de CFAO est composée de deux modules : le premier est dédié à la conception assistée par ordinateur « CAO » et le second est dédié à la fabrication assistée par ordinateur « FAO ». Ces deux modules peuvent s'exécuter dans des environnements différents ou dans un même environnement (chaîne de « CFAO » intégrée).

- CAO : regroupe les logiciels et les techniques de modélisation géométrique qui permettent de concevoir, de tester virtuellement la réalisation des produits manufacturés.
- FAO : son but est d'écrire le fichier contenant le programme de pilotage d'une machine-outil à commande numérique. Ce fichier décrit précisément les mouvements que doit exécuter la machine-outil pour réaliser la pièce demandée.

Le passage d'un module à l'autre se fait systématiquement, dans le cas où la « CAO » est intégrée à la « FAO ». Dans le cas découplé ou interfacé, le passage se fait par l'utilisation de formats d'échanges de données. Dans ce travail, le format STL est considéré.

#### 7.1 Format d'échange de données «STL»

STL est l'abréviation de « Standard Tessellation Language ». C'est un format d'échange de données utilisé dans les logiciels de Stéréo lithographie ou de prototypage rapide. Il permet de décrire un maillage surfacique, c'est-à-dire des modèles sous formes triangulaires contiguës, 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 8). Chaque triangle doit partager deux sommets avec chacun des triangles le juxtaposant. Dit autrement, le sommet d'un triangle ne doit pas être sur l'arête d'un autre triangle. Il n'y a pas d'information d'échelle et l'unité de longueur est arbitraire. Par rapport à des surfaces d'origines ou à des volumes d'origine, la description par une suite de triangles implique une notion de tolérance d'approximation. Donc le modèle STL est déjà une approximation du modèle théorique de départ [7].

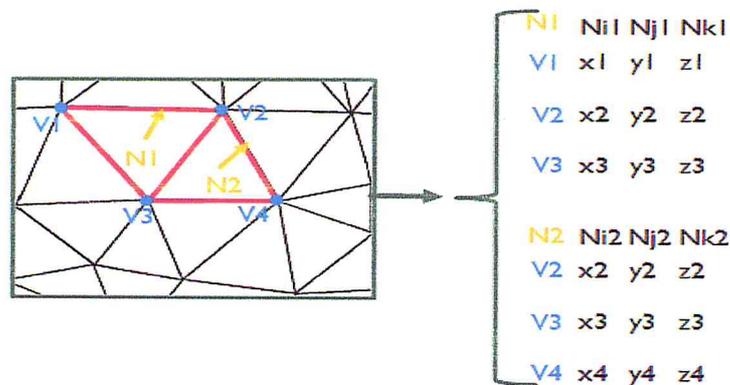


Figure 11 : Paramètres d'un triangle.

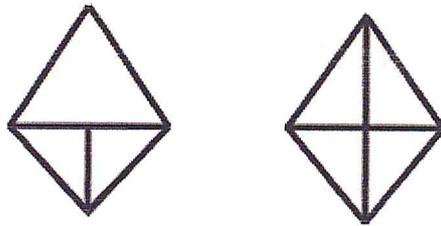
### 7.2 Pourquoi STL?

Les données STL permettent de créer un objet en déposant une succession de calques constituées de plastiques, métaux ou matériaux composites. Les pièces ou modèles obtenus sont généralement utilisés dans les cas suivants :

- Visualiser les conceptions.
- Créer des représentations de produits, des modèles architecturaux et des modèles de terrain.
- Tester les formes.
- Identifier les problèmes de conception.
- Créer des modèles de référence pour des applications de moules positifs [8].

### 7.3 Propriétés [9]

- Approximation de la frontière par des facettes triangulaires.
- Le fichier « STL » est volumineux. Le modèle CAO est remplacé par des triangles dont le nombre dépend de la précision désirée, de la forme de l'objet et de ses dimensions.
- Le fichier « STL » ne contient pas l'information d'échelle et les coordonnées sont dans des unités arbitraires.
- Règle d'autorité : chaque triangle doit partager deux sommets avec chacun de ses triangles adjacents (Figure 9.b). Ainsi, un sommet d'un triangle ne peut pas être localisé sur le côté d'un autre triangle.



a). Non vérifier

b). vérifier

**Figure 12 : Règle d'autorité.**

- Il existe deux types de fichier STL :
  - ASCII : est sous forme de texte lisible par n'importe quel éditeur de texte et génère de gros fichiers. La structure d'un fichier STL ASCII est comme suit:

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

- Binaire : n'est pas lisible mais génère des fichiers de tailles réduites.

### 7.4 Avantages

- Format de fichier compact est simple.
- Format de fichier ne comporte que le minimum nécessaire à la description du modèle 3D ce qui le rend plus simple à l'utilisation.
- Compatible avec de nombreux logiciels de modelage 3D.
- Génération rapide des maillages.
- Description fine des rayons si la précision est correctement définie
- Garantit la confidentialité par ce qu'il est difficilement réutilisable en CAO.
- Très répandu dans l'industrie des prototypes.

### 7.5 Logiciels supportant le format STL :

Le format « STL » est un standard et il est utilisé par différents logiciels tels que :

- Blender
- Meshlab
- Rhinoceros
- Inventor
- Catia
- Solidwork

### Conclusion

L'étude effectuée dans ce chapitre portait sur les différentes activités inhérentes aux processus du Reverse Engineering de produits manufacturés. Dans un premier temps, les différentes méthodes de modélisation des surfaces en CAO ont été explicitées, s'en est suivie une présentation des phases de la Rétro Conception et l'accent a été mis sur le processus de reconstruction d'objet 3D à partir de nuages de points et la description des formats d'échange de données et en particulier le format STL ont été introduits.

La reconstruction d'objets à partir d'un nuage de points est devenue un maillon incontournable du processus du Reverse Engineering et le format STL est devenu le support par excellence de l'usinage des formes complexes et la norme pour la plupart des machines de prototypage rapide qui produisent des modèles physiques en 3D.

Le prochain chapitre est consacré 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.

# Chapitre II

## Méthode de FLIP

### Introduction

L'objectif final du processus du Reverse Engineering est la reconstruction du modèle informatique « modèle CAO ». Afin de faciliter la tâche de reconstruction d'objets, il est préférable d'approximer le nuage de points par un ensemble d'éléments géométriques simples (triangles, tétraèdres, ...etc.). Plusieurs algorithmes ou techniques ont ainsi été développés pour produire en sortie une surface approchant au mieux la forme de la surface physique de l'objet. De façon générale, le terme triangulation désigne une interpolation d'un ensemble de points dans un espace 2D, ou tétraédrisation d'un ensemble de points dans l'espace 3D. [26]

Parmi les méthodes de triangulation citant Delaunay dont FLIP est l'un des algorithmes qui génèrent cette dernière.

### 1. Triangulation de Delaunay

La triangulation consiste à mailler un semis de points, avec des triangles ayant pour sommets les points du nuage et des arêtes qui relient ces derniers pour une représentation 2D ou 3D. Ce maillage doit répondre à certaines propriétés de la triangulation à savoir que l'intersection de deux triangles est soit vide, un point ou une arête. [26]

Le problème consiste à choisir une méthode pour construire cette triangulation. La triangulation la plus utilisée est celle de Delaunay qui possède des propriétés géométriques remarquables et qui permet d'optimiser un certain nombre de critères. De plus, cette triangulation définit une représentation unique et proche de la forme réelle [15]. Celle-ci repose soit sur le diagramme de Voronoï soit sur une approche directe. Dans notre travail, nous nous intéressons à l'approche directe.

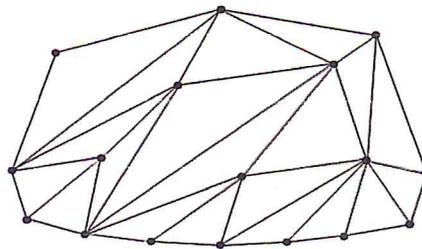


Figure 13 : Triangulation d'un nuage de points.

#### 1.1. Diagramme de Voronoï

Le diagramme de Voronoï pour un ensemble  $S$  de points, appelés sites, dans l'espace euclidien  $R_d$  de dimension  $d$ , est la subdivision de cet espace en régions, nommées cellules de

Voronoi, qui contiennent tous les points de l'espace près du site générateur de la région que des autres sites [15].

Les polyèdres de Voronoi sont convexes, d'intérieurs non vide et forment un recouvrement tel que deux éléments distincts ont une intersection vide ou réduite à un point ou une arête [17].

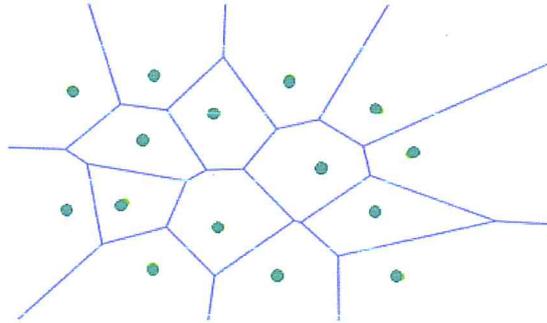


Figure 14 : Diagramme de Voronoï.

### 1.1.1 Propriétés diagramme de Voronoï 3D:

**Propriétés 1:** une cellule de Voronoï 3D sera relativement ronde même si son point générateur fait partie d'un tétraèdre plat.

**Propriété 2:** une cellule de Voronoï est convexe. Il est possible de collecter tous les sommets de Voronoï et ensuite de calculer l'enveloppe convexe des points.

### 1.2 Définition de la triangulation de Delaunay:

Soit  $S$  un ensemble de points en position générale. On appelle triangle de Delaunay de  $S$ , un triangle dont les sommets sont des points de  $S$  et qui est inscriptible dans un cercle vide, c'est-à-dire un cercle qui ne contient aucun point de  $S$  en son intérieur.

On appelle triangulation de Delaunay de  $S$  toute triangulation de  $S$  dont tous les triangles sont des triangles de Delaunay de  $S$  [4].

Ils existent plusieurs algorithmes qui génèrent, de façon incrémentale, la triangulation de Delaunay. On parle de techniques incrémentales lorsque nous partons d'une triangulation et en ajoutant un point, nous générons d'autres triangles.

## 1.2.1. Propriétés de Delaunay 2D:

**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.

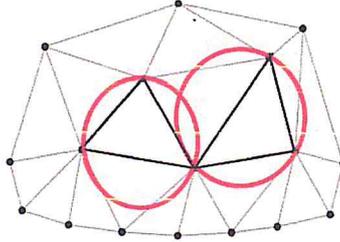
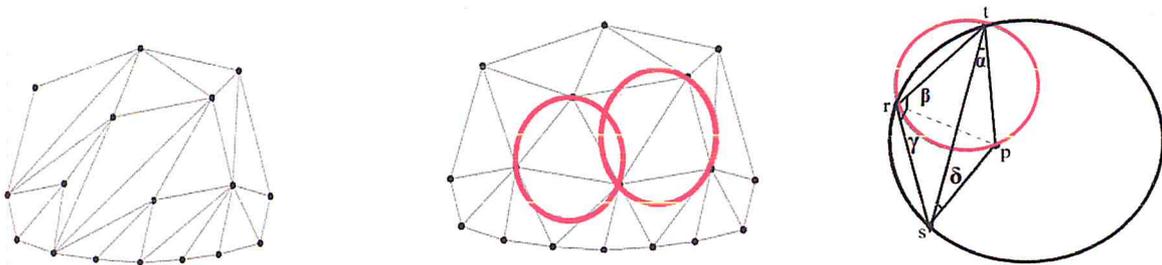


Figure 15 : Triangle entouré de cercle circonscrit.

**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.



a) Triangulation quelconque.

b) Triangulation de Delaunay.

c)  $\beta > \delta$  et  $\gamma > \alpha$ .

Figure 16 : Propriété des angles.

**Propriété 4:** la triangulation de Delaunay de  $S$  est la seule triangulation de  $S$  dont toutes les arêtes sont légales.

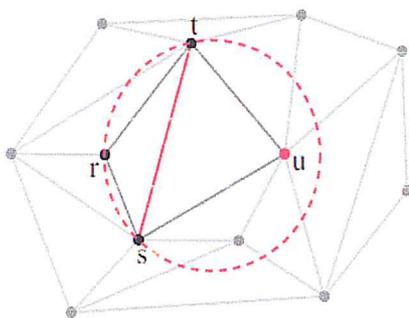


Figure 17 : Arête illégale.

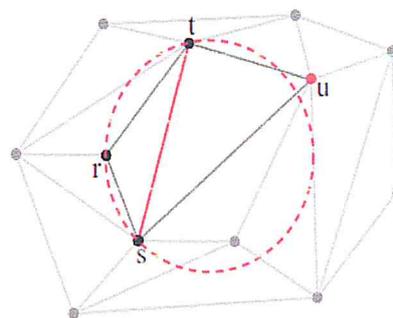


Figure 18 : Arête légale.

### 1.3. Triangulation Delaunay en 3D:

En trois dimensions, une triangulation 3D de Delaunay est une tétraédrisation de l'ensemble des points (Figure 21). Dans ce cas, le nuage de points définit la surface d'un objet ainsi que son intérieur, chaque tétraèdre possède une sphère circonscrite vide (Figure 22). Ces dernières ne se chevauchent jamais et chaque tétraèdre est adjacent à exactement quatre tétraèdres sauf ceux qui sont situés sur la frontière de l'ensemble de points [16].

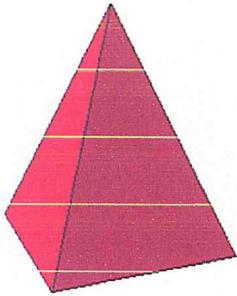


Figure 19 : Tétraèdre.

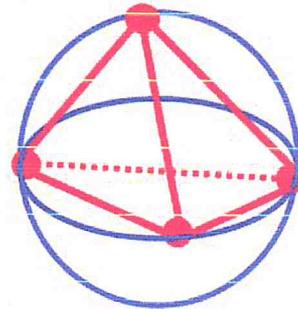


Figure 20 : sphère circonscrite

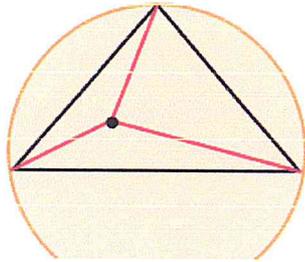
## 2. Méthode d'implémentation de Delaunay:

Plusieurs algorithmes permettent l'implémentation de la triangulation de Delaunay mais les plus utilisés sont la Destruction-Construction et l'algorithme de Flip.

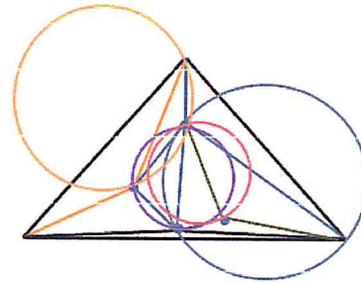
### 2.1. Méthode Destruction-Construction :

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 et donc quatre points déjà insérés, l'algorithme rajoute des points qui sont intérieurs à la triangulation et qui sont triangulés au fur et à mesure.

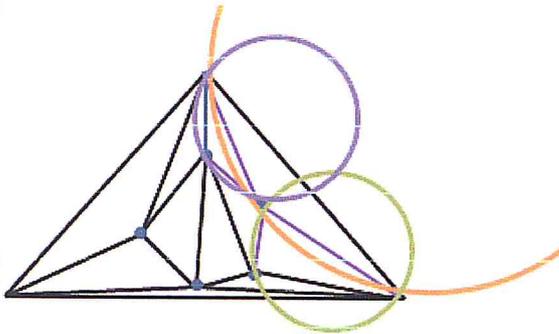
On procède ainsi : on utilise un triangle et un point choisi comme indiqué ci-dessus (Figure 24), puis tous les triangles dont le cercle circonscrit contient le point considéré sont détruits. Une fois les triangles illégaux détruits, on construit ceux qui sont engendrés par ce nouveau point, en ajoutant des arêtes reliant au point inséré les extrémités de celles qui n'appartenaient qu'à un seul triangle présent dans la liste de ceux qui ont été supprimés [18].



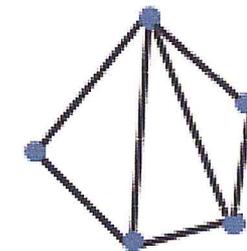
Insertion d'un point dans un « super-triangle »



Insertion 2, 3,4 point.



Suppression des arêtes du super-triangle.



Suppression des arêtes du super-triangle.

**Figure 21 : Méthode Destruction-Construction.**

### 2.2. CRUST :

Le Crust est une méthode pour reconstruire une approximation du contour d'un objet 2D ou de la surface d'un objet en 3D à partir d'un ensemble de points mesurés sur un objet. En dimension 3, la méthode consiste à :

- Construire le diagramme de Voronoï.
- Pour chaque site  $v$  on définira deux pôles, le premier pôle « $pv$ » est le sommet de la cellule de Voronoï de  $v$  le plus loin de  $v$  tandis que le deuxième pôle « $pv'$ » est le sommet de la cellule de Voronoï de  $v$ , le plus loin de « $pv$ ».
- Construire la triangulation de Delaunay des sites originaux des pôles.
- Extraire de cette triangulation les triangles formés uniquement de sites originaux.
- Afficher le résultat [19].

### 2.3. DeWall :

L'algorithme DeWall est basé sur l'approche « diviser pour régner ». La démarche de cet algorithme consiste à : 1) la subdivision du nuage, 2) triangulation (TD) des sous nuages et 3) concaténation de ces triangulations. Les modifications locales lors de la concaténation (insertion, suppression ou mouvement) sont lentes [16].

### 2.4. Algorithme de flip en 2D :

La méthode incrémentale de résolution d'un problème avec un ensemble E de données consiste à trouver la solution pour un ensemble E0 inclus dans E puis à ajouter les autres éléments de E un par un en s'assurant qu'à chaque étape la solution reste valable. Cette méthode est très utilisée dans la résolution des problèmes géométriques où nous avons un ensemble de points de grande taille. Elle peut être randomisée en insérant les points dans un ordre aléatoire ce qui permet d'accélérer le traitement. L'algorithme de Flip est basé sur ce concept.

Dans le cas de la triangulation de Delaunay, nous débutons avec un tétraèdre englobant l'ensemble des points de P, puis nous insérons un par un les points de P dans un ensemble migrant peu à peu vers la solution complète. A l'étape i de l'algorithme, nous avons une triangulation de i points sur n. L'insertion du i+1 ème point risque de contrarier les propriétés de l'ensemble obtenu jusqu'à lors. Pour retrouver la propriété de Delaunay, il faut alors entamer une série de modifications locales constituée de flips.

C'est une fonction récursive (qui pourrait être implémentée en itératif), qui vérifie la légalité des côtés. Un côté est défini comme étant l'arête commune à deux triangles. Pour savoir s'il est illégal, seuls les points qui ne sont présents que dans un des triangles sont regardés (points opposés) [19].

Si c'est le cas, le côté illégal est remplacé par un côté relié par les deux points opposés. Par la suite, les quatre côtés qui restent sont testés pour voir si ce changement a provoqué un autre conflit. La (Figure 22) illustre la position d'un côté illégal et la correction apportée par le flip.

D'autres problèmes peuvent être rencontrés, tel que les triangles allongés. Dans ce cas, un point est ajouté et la triangulation est refaite localement. Ainsi, l'algorithme peut être décrit par :

- Insertion d'un point, destruction du triangle le contenant.
- Création de trois triangles ayant ce nouveau point comme sommet.

- Vérification de chaque côté externe pour une illégalité.

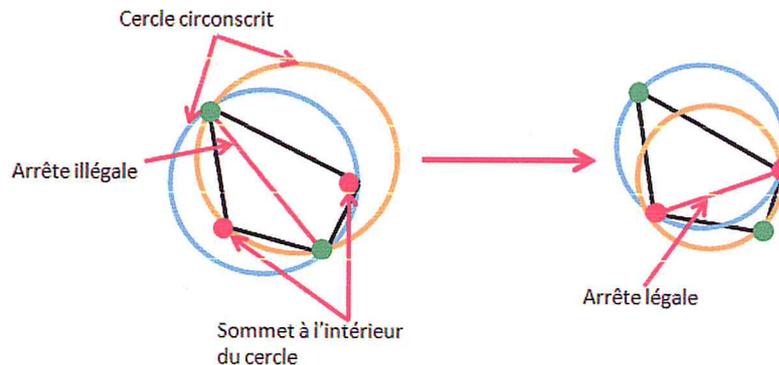


Figure 22 : Un côté illégal (gauche), le flip (droite)

### 3. Méthode de Flips en 3D:

Un « Flip » est une opération topologique qui modifie localement la configuration de quelques tétraèdres adjacents dans une triangulation. C'est donc une opération très utile pour insérer, supprimer ou déplacer un sommet dans une tétraédrisation de Delaunay.

Ils existent plusieurs façons de trianguler un ensemble de cinq points. En effet, ces cinq points peuvent former soit deux, trois ou encore quatre tétraèdres différents. Ainsi, comme le montre la Figure 26, divers flips tridimensionnels peuvent être définis. La séquence des flips est contrôlée par une structure « pile » qui contient tous les tétraèdres qui n'ont pas encore été testés. La pile commence avec les quatre nouveaux tétraèdres créés par l'insertion du point et chaque fois qu'un flip est réalisé, les nouveaux tétraèdres sont ajoutés à la pile. Lorsque la pile est vide, cela signifie que tous les tétraèdres formant la triangulation sont Delaunay. Cette procédure est répétée tant que de nouveaux points sont ajoutés à la triangulation [16].

#### 3.1. Structures de données utilisées pour flip :

##### 3.1.1. Piles:

C'est une structure de données utilisée lors de l'implémentation de la méthode de flip. Elle est composée de cases superposées dont chaque case contient une arête de la triangulation. Cette structure travaille avec FILO (First In First Out,) ou LIFO (Last In First Out).

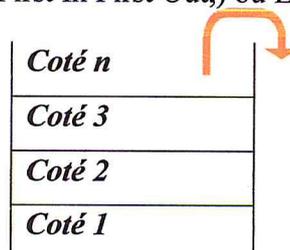


Figure 23: structure de pile.

### 3.1.2. Pointeurs:

La triangulation Delaunay 3D est basée sur une structure de données tétraédrique. Cette structure de données stocke chaque tétraèdre en utilisant quatre pointeurs pour les nœuds constituant ses sommets et quatre autres vers les quatre tétraèdres qui lui sont adjacents [20].

Le tétraèdre cible (2) de la Figure 24, est stocké à l'aide de quatre pointeurs vers ces sommets (a, b, c et d) et quatre pointeurs vers ces tétraèdres voisins (4, 6, 9 et 11).

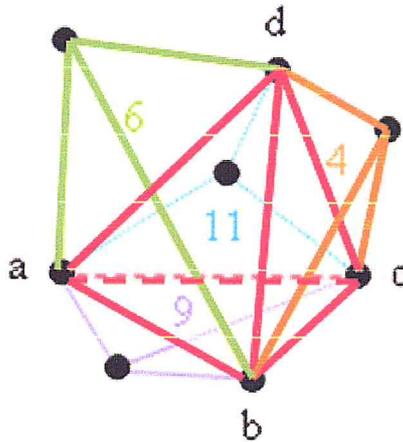


Figure 24 : structure de donnée avec tétraèdre.

### 3.1.3. Structure DAG :

Le DAG (Directed Acyclic Graph) est une structure de données qui permet de localiser le tétraèdre ou triangle T contenant un point P. Cette localisation est une étape fondamentale de la méthode incrémentale. Les étapes de fonctionnement de ce DAG sont illustrées par des schémas associant géométrie et structure de données [21].

La forme initiale du DAG est un nœud qui désigne un tétraèdre ou un triangle englobant les points de P. Le split et le flip sont les deux seules opérations qui modifient le DAG. Lors du split d'un triangle T0, le nœud pointant T0 donne naissance à trois fils pointant sur les triangles T1, T2 et T3 dont la réunion recouvre T0 [21].



Figure 25 : Evolution du DAG lors d'un split.

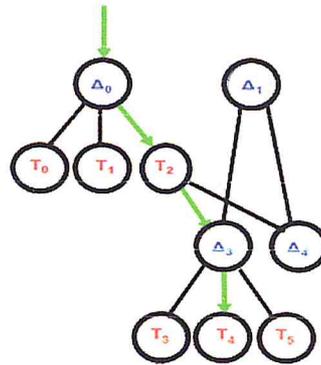


Figure 26: Recherche dans le DAG du triangle contenant le point P.

### 3.1.4. Structure vectorielle

Elle permet de stocker des données de différents types dans des tableaux unidimensionnels.

## 3.2. Processus de la triangulation Delaunay 2D utilisant flip : [23]

### Etape 1: Initialisation

L'algorithme est initialisé avec un triangle fictif, englobant l'ensemble des points à trianguler. Ce triangle étant par nature une triangulation de Delaunay pour ses trois points, permet d'utiliser l'algorithme incrémental qui s'applique sur une triangulation de Delaunay préexistante. L'important étant que cette initialisation fournit une triangulation de démarrage avec les trois points du triangle fictif.

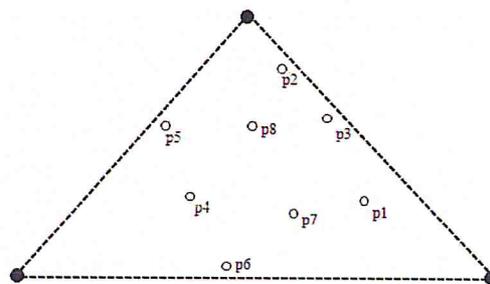
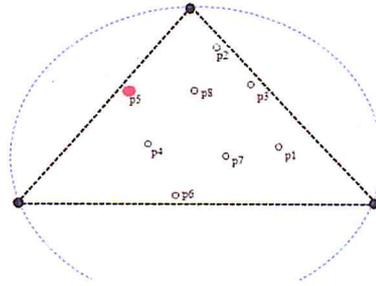


Figure 27: Triangle fictif englobant 8 points à trianguler.

### Etape 2 : Insertion d'un point

À la triangulation existante, un point de l'ensemble des points à trianguler est inséré. Il peut être choisi aléatoirement même si selon la topologie il peut être judicieux d'ordonner l'insertion des points selon certains critères.



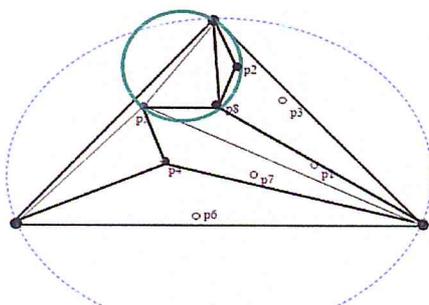
**Figure 28:** Insertion du point p5.

### Étape 3: Suppression et création de triangles :

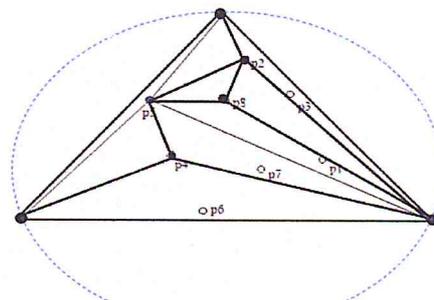
Chaque point inséré est relié avec les trois points du triangle cible pour créer de nouveaux triangles qui seront testés au critère de Delaunay avec les triangles adjacents. Cette étape est décrite par l'algorithme suivant :

Pour chaque point inséré :

- Localisation du triangle cible contenant le point inséré.
- Création de trois nouveaux triangles reliant le point inséré avec les trois sommets du triangle cible.
- Suppression du triangle cible.
- Vérification du critère de Delaunay (cercle circonscrit) :
  - Si un sommet du triangle adjacent est à l'intérieur du cercle circonscrit :  
Effectuer le flip par la suppression de l'arête commune des deux triangles et la remplaçant par une autre qui relie les deux autres sommets opposés des deux triangles.
  - Si non, on passe au triangle suivant.
- La procédure est appliquée jusqu'à l'insertion de tous les points du nuage.



**Figure 29 :** Triangle illégal.



**Figure 30 :** Arête illégale.



### 3.4.1. A partir d'une triangulation 3D quelconque

L'algorithme suivant décrit le processus de la tétraédrisation à partir d'une triangulation 3D quelconque.

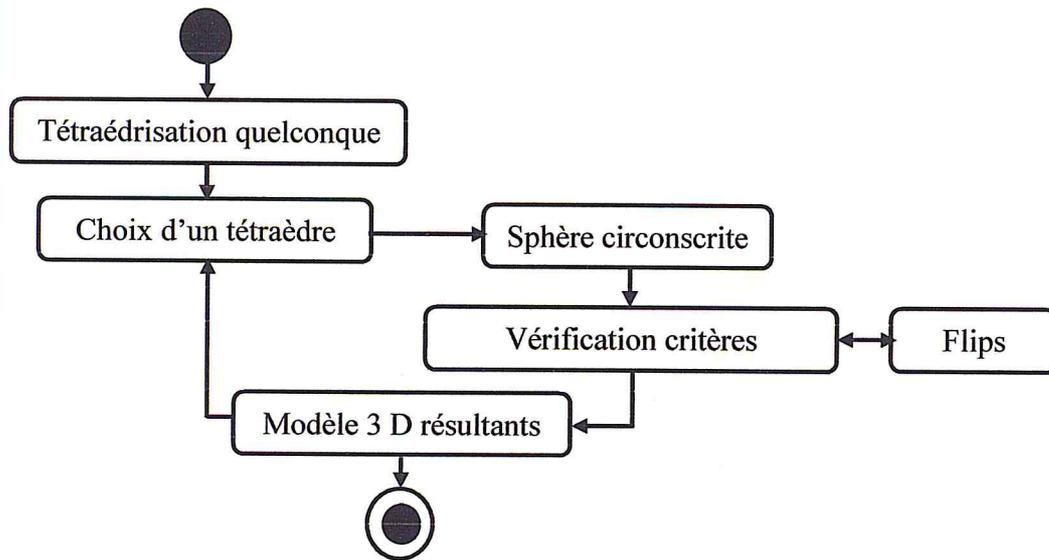


Figure 1: Tétraédrisation quelconque par la méthode de Flip.

Etant donnée une triangulation quelconque, l'algorithme utilisé pour la construction de la triangulation Delaunay 3D comporte quatre étapes :

- Choix d'un tétraèdre de la liste des tétraèdres.
- Vérification du critère de Delaunay
- Si la sphère circonscrite ne contient aucun sommet appartenant aux tétraèdres adjacents :
  - Passer au tétraèdre suivant.
- Si non :
  - Les tétraèdres doivent être modifiés par des opérations topologiques appelées flips.

### 3.4.2. A partir d'un nuage de points

L'algorithme suivant décrit le processus de la tétraédrisation à partir d'un nuage de points.

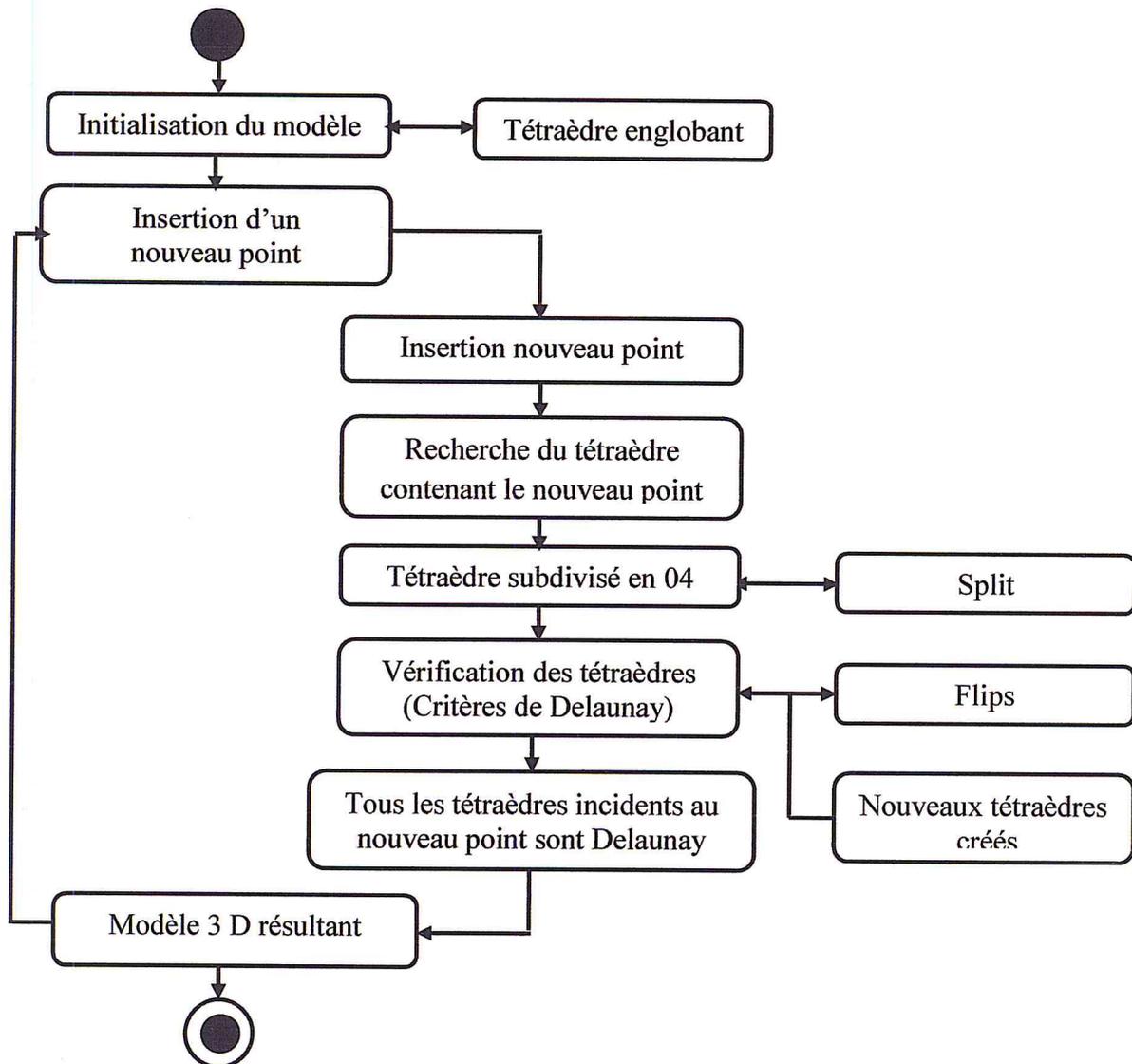


Figure 33 : Étapes de la triangulation Delaunay 3D par la méthode de Flip.

L'algorithme utilisé pour la construction de la triangulation Delaunay 3D par la méthode Flip comporte plusieurs étapes :

- Première étape (initialisation du modèle) : création d'un tétraèdre englobant qui doit être assez grand pour contenir tous les points de données à traiter (tétraèdre universel).
- Deuxième étape: insertion point par point dans le modèle.
- Troisième étape : Recherche du tétraèdre qui contient le nouveau point ajouté en utilisant les relations d'adjacence entre les tétraèdres du modèle. Une fois le point

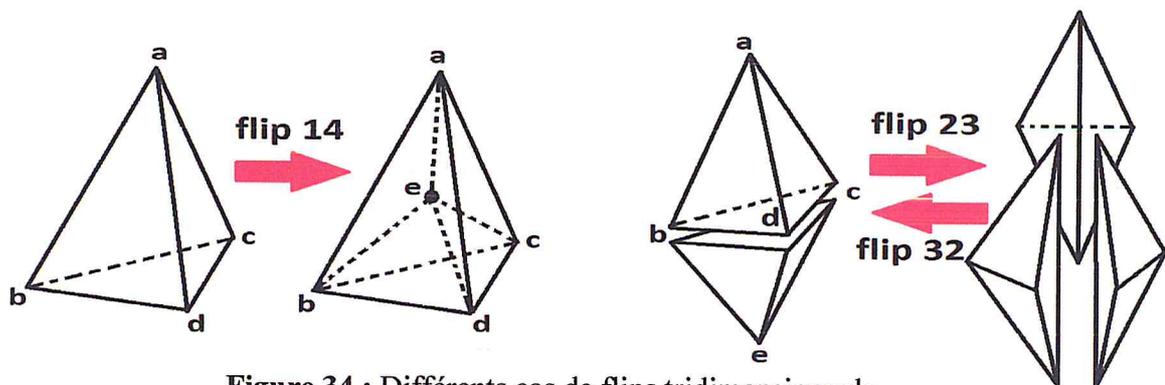
## Méthode de FLIP

inséré dans le tétraèdre, le « *split* » est effectué et les quatre nouveaux tétraèdres créés peuvent ne pas respecter le critère de Delaunay.

- Quatrième étape : vérifier la validité du critère de Delaunay de tous les nouveaux tétraèdres créés. Dans le cas contraire, modifier ces tétraèdres par des opérations topologiques appelées « *flips* » qui consiste à créer soit deux, trois ou encore quatre tétraèdres différents à partir des cinq points. Ainsi, comme le montre la Figure 28, divers flips tridimensionnels peuvent être définis.

Cette procédure est répétée jusqu'à l'insertion de tous les points de la liste.

La séquence des flips est contrôlée par une structure « pile » qui contient tous les tétraèdres qui n'ont pas encore été testés. La pile commence avec les quatre nouveaux tétraèdres créés par l'insertion du point et chaque fois qu'un flip est réalisé, les nouveaux tétraèdres sont ajoutés à la pile. Lorsque la pile est vide, cela signifie que tous les tétraèdres constituant la triangulation sont Delaunay.



### 3.4.3 Problème de légalité 3D :

Il y a deux façons de légaliser une face d'un tétraèdre : le flip convexe et le flip non convexe. Le choix entre ces deux méthodes dépend de la convexité de l'ensemble formé par les deux tétraèdres adjacents et le point contenu dans la sphère circonscrite. Pour cela, un test du « *cône test* » est effectué. Le test consiste à vérifier l'appartenance du point au cône engendré par un de ces deux tétraèdres et reposant sur leur face commune.

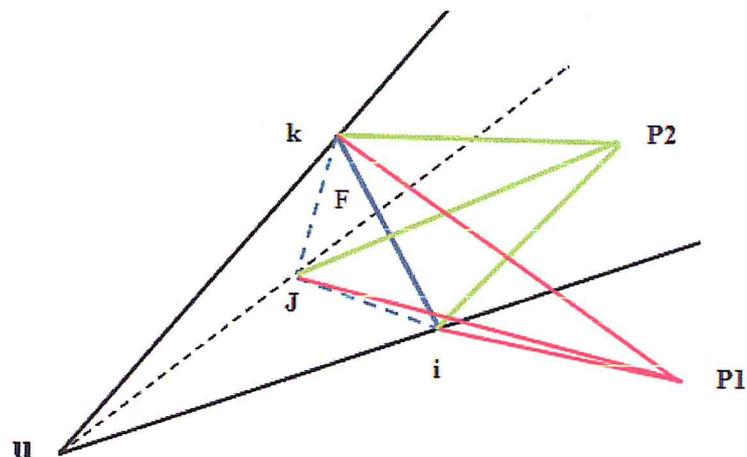


Figure 35 : « Cône test ».

Dans le cas où le point est à l'intérieur du cône, le couple est convexe et les deux tétraèdres sont remplacés par trois nouveaux autres.

Dans le cas où le point est à l'extérieur du cône, le couple de tétraèdres formé est non convexe et deux cas sont envisagés :

- Dans le cas d'existence d'un autre tétraèdre commun unique aux deux autres précédents, on applique le flip. Ainsi, les trois tétraèdres précédents sont remplacés par deux autres nouveaux tétraèdres.
- Dans le cas d'absence du tétraèdre commun ou l'existence de plusieurs tétraèdres, le traitement est effectué ultérieurement.

### Conclusion :

Diverses méthodes de reconstruction existent, chacune ayant ses artifices et ses limitations. Parmi les méthodes de segmentation existantes, il y a celles qui s'appuient sur la polyédrisation du nuage de points et qui a fait l'objet de notre travail. La construction du polyèdre, s'appuie sur l'adaptation de la triangulation de Delaunay à la reconstruction. Celle-ci étant classiquement réservée à la triangulation d'objets plans ou à la tétraédrisation d'objets 3D. Pour mieux exploiter ce modèle, la méthode de Flip qui modifie localement les triangles d'un maillage pour obtenir une triangulation de Delaunay. L'algorithme de flip a été présenté en détail. Dans un premier temps, l'accent a été mis sur les quatre types de structure de données utilisée (piles, pointeurs, DAG et structures vectorielles). L'application de la méthode à la triangulation 2D a été présentée par la suite les différentes étapes de la méthode ont été explicitées. Les étapes de la méthode appliquée au processus de la triangulation

## Méthode de FLIP

---

Delaunay 3D pour de type de triangulation ; triangulation à partir d'une triangulation quelconque et triangulation à partir d'un nuage de points ont achevé ce chapitre.

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

# Chapitre III

## Conception de l'application

### Introduction:

La reconstruction 3D permet l'obtention d'une représentation numérique en trois dimensions d'un objet à partir d'un ou de plusieurs nuages de points. Le but recherché est de produire en sortie une surface STL approchant au mieux la forme de l'objet physique. Pour cela, la méthode de FLIP en 3D a été appliquée pour générer une triangulation respectant les critères de Delaunay.

### 1. Problématique

La reconstruction 3D, désigne la technique qui permet d'obtenir une représentation en trois (03) dimensions d'un objet à partir d'un nuage de points dont les coordonnées 3D sont saisies par un dispositif de mesure prises sous différents points de vue de l'objet. Ces objets sont utilisés dans divers domaines tels que l'aéronautique, l'industrie de l'automobile, les moules, les matrices, etc.

On dispose en entrée du processus de reconstruction, d'un ensemble de points non organisés acquis à la surface d'un objet, le but recherché est de reproduire en sortie une surface approchant au mieux la forme de la surface physique de l'objet échantillonné. La reconstruction des modèles 3D avec une bonne précision est une tâche ardue, couteuse en temps, par conséquent, la solution à ce problème n'est pas une tâche triviale.

### 2. Architecture globale de l'application:

Le travail concerne l'approximation de surfaces en trois dimensions à partir d'un nuage de points non structuré. L'approximation consiste à générer la tétraédrisation de Delaunay 3D en utilisant l'algorithme de FLIP. L'objectif final est la génération d'un modèle STL exploitable par les applications de l'usinage de surfaces discrètes. L'architecture générale de l'application est divisée en quatre 04 parties illustrées par l'organigramme suivant (Figure 36) :

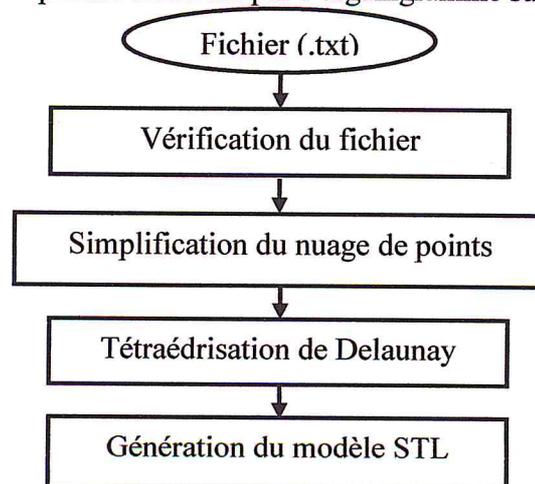


Figure 36 : Organigramme de l'Architecture globale.

### 3. Vérification du fichier

La vérification du fichier d'entrée consiste à vérifier la conformité syntaxique du fichier de base. Si le fichier n'est pas valide, le nuage de points n'est pas traité (Figure 36). La procédure de vérification est illustrée ci-dessous :

1. Sélection d'un fichier texte.
2. Vérification du fichier s'il contient des caractères non conformes (informations hors les coordonnées des points):
  - Si oui, le fichier est non valide.
  - Si non, lecture du fichier:
    - Récupérer les coordonnées des points (x, y, z)
    - Stocker chaque point dans une classe créée à cet effet « Point\_Nuage ».
    - Enregistrement du point dans un tableau « tableau\_nuage\_point » de type 'Point\_Nuage'.

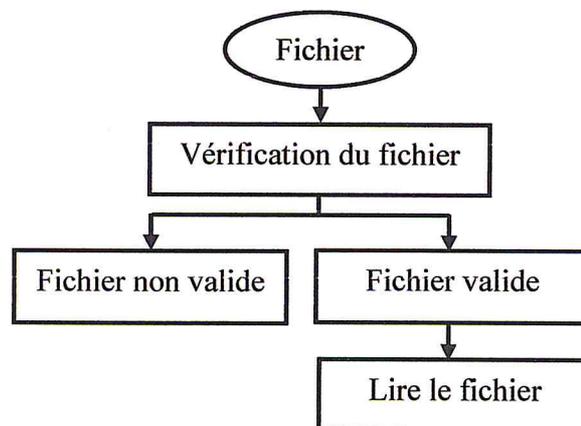


Figure 37 : Organigramme de vérification du fichier.

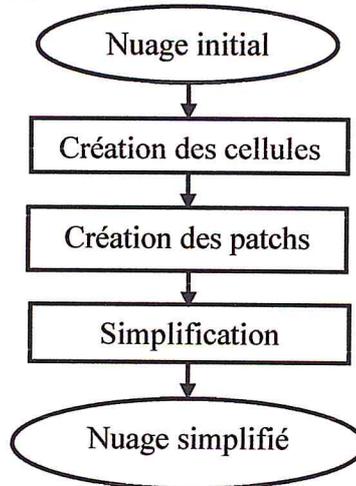
### 4. Simplification du nuage de points

Après la lecture du fichier, celui-ci est simplifié pour minimiser le nombre de points afin de faciliter sa manipulation et son traitement. L'objectif de cette étape est l'optimisation du temps de reconstruction des surfaces. Pour atteindre cet objectif, une solution en deux étapes est proposée:

- Première étape : création des cellules et des patches.
- Deuxième étape : suppression des points redondants et des points rapprochés d'une distance seuil.



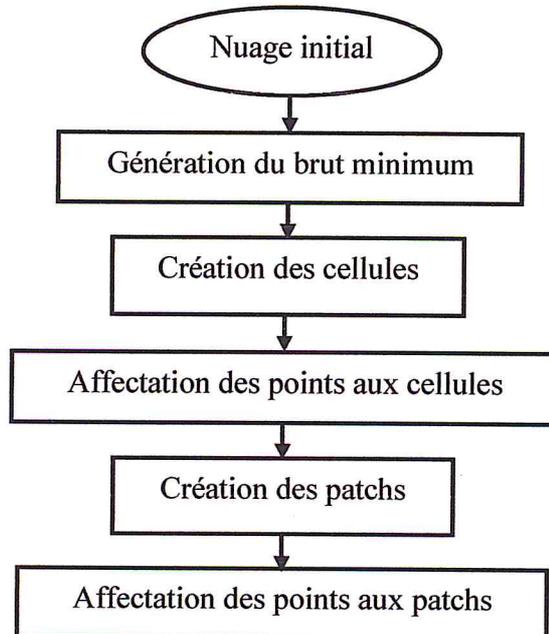
La Figure suivante décrit l'approche générale :



**Figure 38** : Organigramme de la simplification du nuage de points

### 4.1. Création des cellules et des patches

Cette partie consiste à subdiviser le brut englobant le nuage de points en cellules de même taille. Par la suite, chaque cellule est subdivisée en patches. L'objectif de cette subdivision est d'accélérer et de minimiser les temps de calcul global.

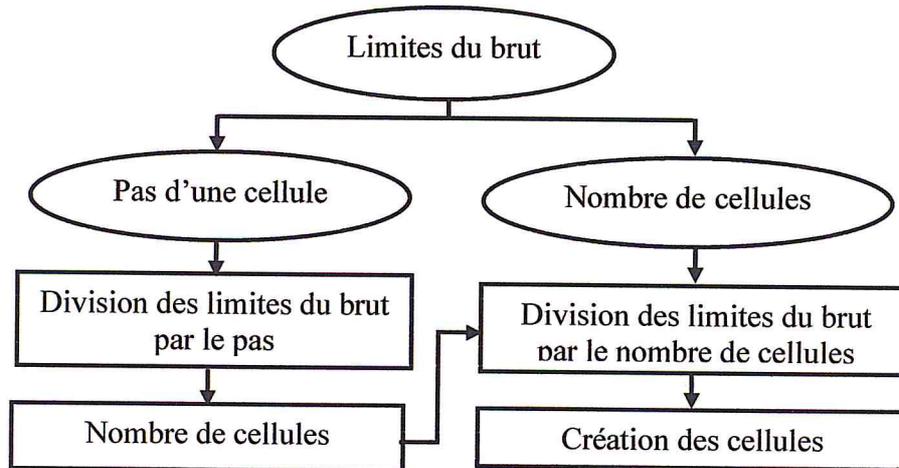


**Figure39** : Organigramme de création des cellules et des patches

## Conception de l'application

### 4.1.1. Création des cellules

La première étape dans le processus de simplification est la création de blocs parallélépipédiques appelés cellules. La figure suivante illustre ses principales étapes.

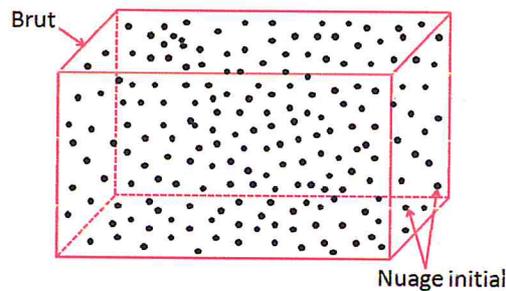


**Figure 40 :** Organigramme de création des cellules

Le processus de création des cellules est illustré ci-dessous :

#### 1. Limites du brut (Figure 41) :

- Calcul des coordonnées du brut minimum ( $X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{min}, Z_{max}$ ) à partir des coordonnées des points du nuage.
- Calcul des dimensions du brut à savoir la hauteur, la longueur et la largeur.



**Figure 41:** Calcul du brut

2. Subdivision de la hauteur, de la longueur et de la largeur par le nombre de cellules pour calculer les pas suivant les trois axes X, Y et Z.
  - Si non, subdiviser la hauteur, la longueur et la largeur par le pas respectif de la cellule pour avoir le nombre de cellules.
3. Création des cellules et leurs enregistrements dans un tableau dynamique à trois dimensions (Figure 42).

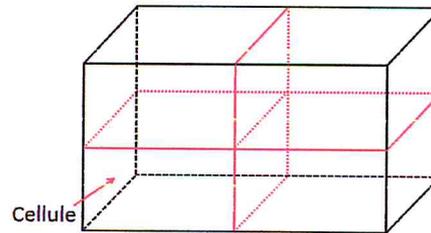


Figure 42: Création des cellules

### 4.1.2 Affectation des points aux cellules

L'affectation des points aux cellules consiste à déterminer pour chaque cellule les points appartenant à l'intérieur de son enveloppe en se basant sur des considérations purement géométriques (Figure 43).

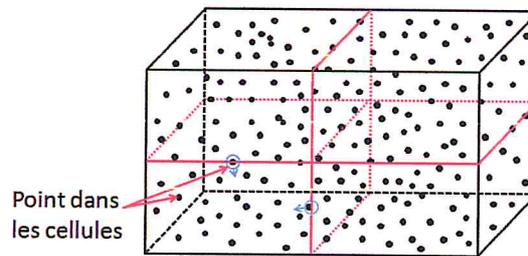


Figure 43: Affectation des points aux cellules

### 4.1.3. Création des patches

Cette opération permet de subdiviser chaque cellule en sous-cellules appelées patches pour faciliter l'opération de simplification. La création des patches et l'affectation des points à ces patches est similaire à celle des cellules. Pour la création des patches, trois modes sont proposés (Figure 44) :

Mode 1 : spécification du nombre de cellules.

Mode 2 : spécification des dimensions de la cellule.

Mode 3 : spécification d'un pourcentage des dimensions de la cellule.

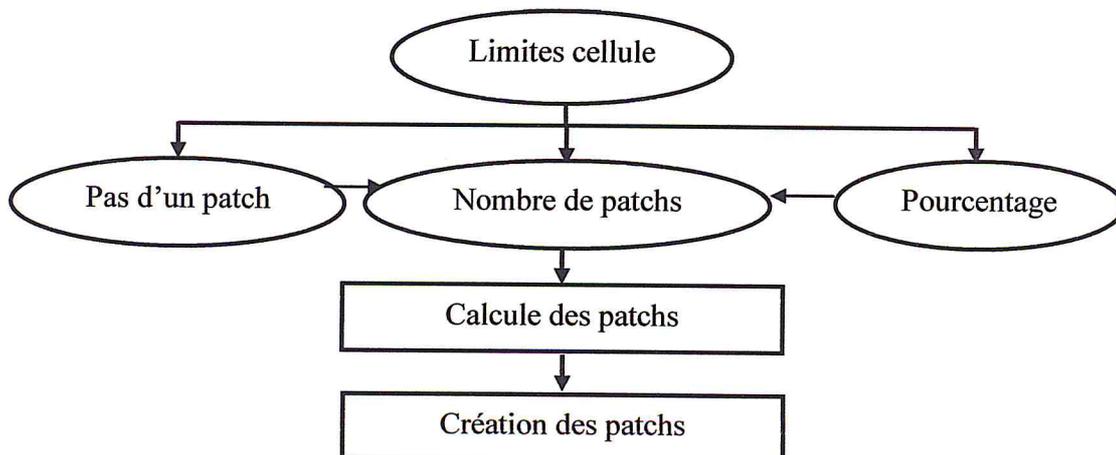
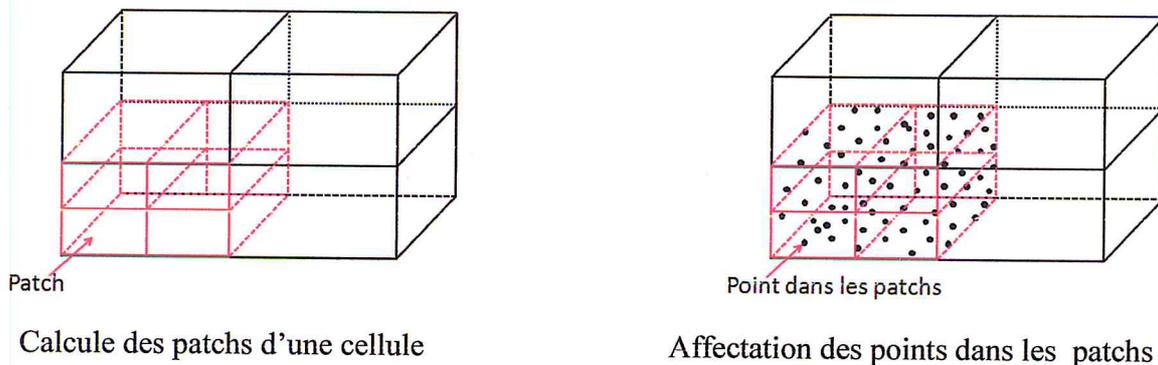


Figure 44 : Organigramme de la création des patches

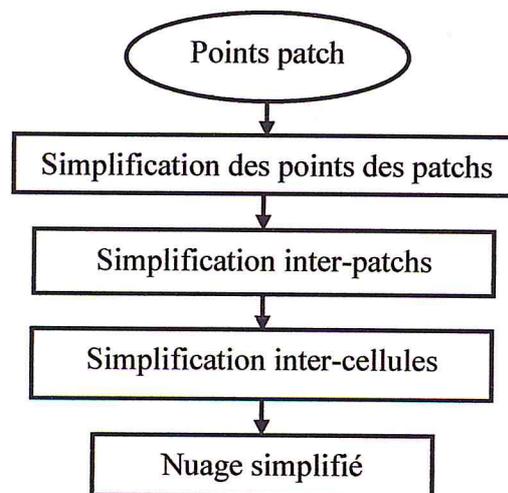
La Figure suivante montre la subdivision d'une cellule en patches et les points de chaque patch.



**Figure 45:** Processus de création des patches

### 4.2. Suppression des points redondants

Consiste à supprimer des patches les points doubles ainsi que les points dont la distance entre eux est inférieure à une distance maximale spécifiée par l'utilisateur. Cette opération est réalisée pour un patch d'une cellule donnée et sera généralisée sur toutes les cellules par la suite. Par la suite, des simplifications inter-patches et inter-cellules sont exécutées (Figure 46). Le résultat final, obtention d'un nuage de points simplifié.



**Figure 46 :** Organigramme de simplification du nuage de points

#### 4.2.1 Simplification des points dans un patch :

Le processus de simplification des points dans un patch est inspiré d'une méthode qui permet la simplification rapide d'un nuage de points avec préservation des caractéristiques vives.

L'approche est illustrée par l'algorithme ci-après (Figure 47) :

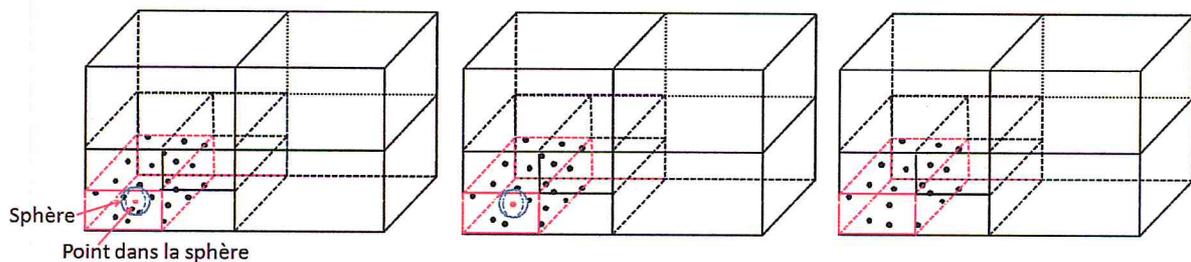
**Entrée :** distance pour la simplification

1. Calculer l'enveloppe d'un point appartenant à un patch.

## Conception de l'application

2. Calculer la sphère dont le centre est le point donné et le rayon est la distance entrée.
3. Vérifier l'appartenance des autres points à l'intérieur de l'enveloppe.
4. Si oui :
  - Calculer la distance entre ces points et le centre de la sphère :
    - Si la distance calculée est inférieure au rayon, les points sont supprimés.
    - Si non, ils ne sont pas supprimés.

Puis, l'opération de la création des patchs ainsi que celle de la simplification des points dans les patchs seront généralisées sur toutes les cellules.



a). Point dans la sphère

b). Suppression du point

c). Patch simplifié

**Figure 47:** Processus de simplification d'un patch

### 4.2.2. Simplification inter-patch:

L'algorithme de généralisation inter-patch est donné ci-dessous (Figure 48) :

1. Détermination de l'indice début et l'indice fin de l'ensemble des points de chaque patch.
2. Calcul de la distance de chaque point d'un patch par rapport à l'extrémité maximale de ce patch suivant l'axe X.
3. Si cette distance est inférieure à la distance spécifiée :
  - Calcul de la distance entre ce point et les points du patch suivant, avec un déplacement suivant l'axe X :
    - Si cette distance est inférieure à la distance spécifiée, alors suppression du point du deuxième patch.
4. Calcul de la distance de chaque point d'un patch par rapport à l'extrémité maximale de ce patch suivant l'axe Y.
5. Si cette distance est inférieure à la distance spécifiée :
  - Calcul de la distance de ce point avec les points du patch suivant avec un déplacement suivant l'axe Y :
    - Si cette distance est inférieure à la distance spécifiée, alors suppression du point du deuxième patch.

## Conception de l'application

6. Calcul de la distance de chaque point d'un patch par rapport à l'extrémité maximale de ce patch suivant l'axe Z :
7. Si cette distance est inférieure à la distance spécifiée :
  - Calcul de la distance de ce point avec les points du patch suivant avec un déplacement suivant l'axe Z :
  - Si cette distance est inférieure à la distance spécifiée, alors suppression du point du deuxième patch.

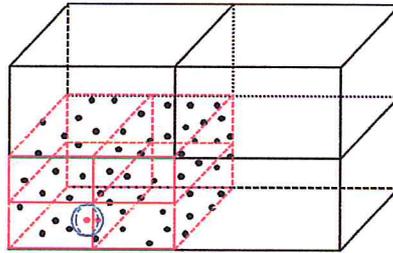
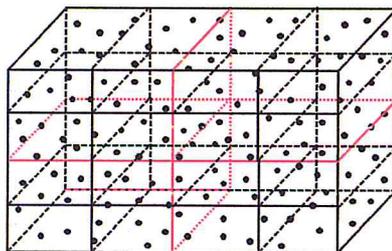


Figure48 : Simplification inter patches

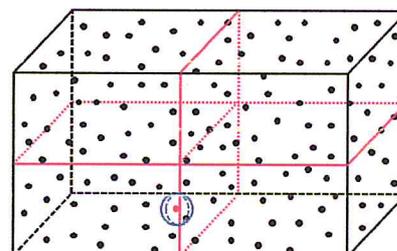
### 4.2.3. Simplification inter-cellules

La simplification inter-cellules est réalisée en deux étapes (Figure 49) :

1. Généraliser la création des patches, la simplification des points dans les patches et inter patches sur toutes les cellules.
2. Simplifier les points inter cellules de la même manière que la simplification inter patches.



a). Généralisation des patches



b). Simplification inter cellules

Figure 49: Processus de simplification inter cellules

## 5. Triangulation de Delaunay

La triangulation de Delaunay 3D, suit la segmentation du nuage de points initial. Elle consiste à générer des tétraèdres à partir du nuage de points non structuré pour pouvoir générer à la fin un fichier STL représentant la peau extérieure de l'objet digitalisé.

Dans ce travail, la tétraédrisation peut commencer soit avec un tétraèdre englobant soit avec douze (12) (Figure 50).

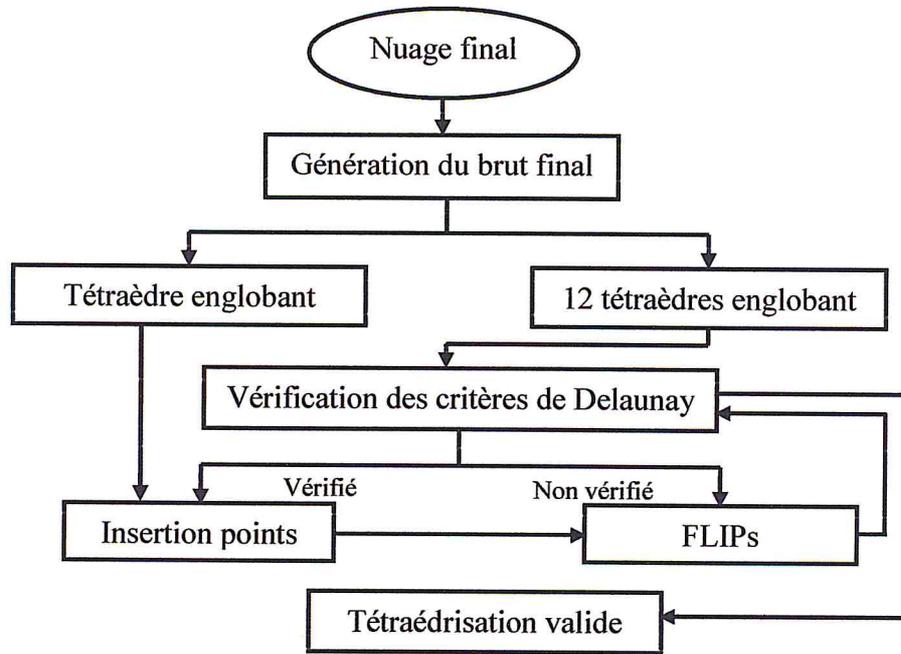


Figure 50 : Organigramme de la triangulation de Delaunay

### 5.1 Triangulation à partir de 12 tétraèdres :

La tétraédration à partir de douze (12) tétraèdres est réalisée en trois (03) étapes (Figure51) :

1. Création d'un point fictif qui est le centre du nouveau brut.
2. Génération des douze (12) tétraèdres à partir des neufs (09) points où huit (08) points sont les coins du brut et le dernier est le point centre du nouveau brut.
3. Stockage des tétraèdres créés dans un vecteur :

Chaque tétraèdre est caractérisé par ses quatre (04) sommets, quatre (04) normales, quatre (04) faces, quatre (04) voisins et les paramètres de sa sphère circonscrite (point centre et rayon).

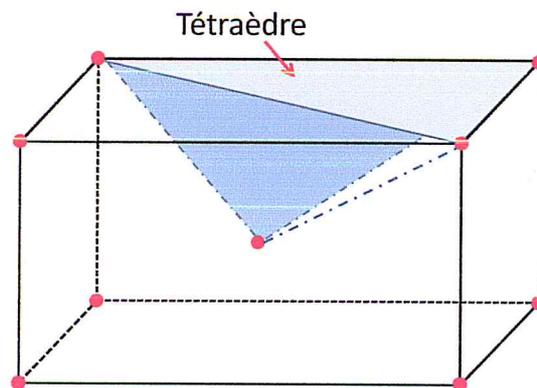


Figure51 : Création d'un des douze tétraèdres

### 5.2 Création d'un tétraèdre englobant :

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

1. Calcul du centre du brut.
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 ( $X_{min}$ ,  $X_{max}$ ,  $Y_{min}$ ,  $Y_{max}$ ,  $Z_{min}$ ,  $Z_{max}$ ).
3. Génération des quatre points sur la sphère qui permettent 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.

### 5.3 Caractéristiques d'un tétraèdre :

Chaque tétraèdre est caractérisé par :

1. Quatre (04) points.

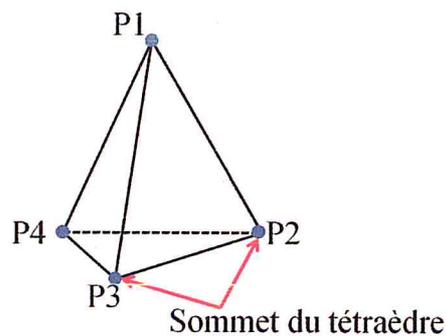


Figure 52 : Sommets d'un tétraèdre.

2. Quatre (04) normales dirigées vert l'extérieurs (normalisée).

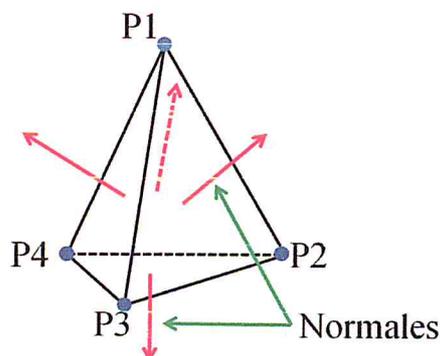
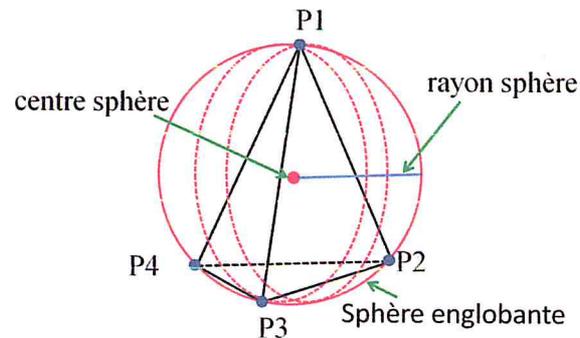


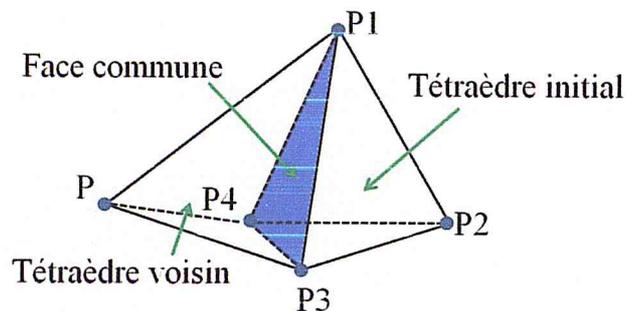
Figure 53 : Normales d'un tétraèdre.

3. Une sphère englobante.
4. Le centre et le rayon de la sphère



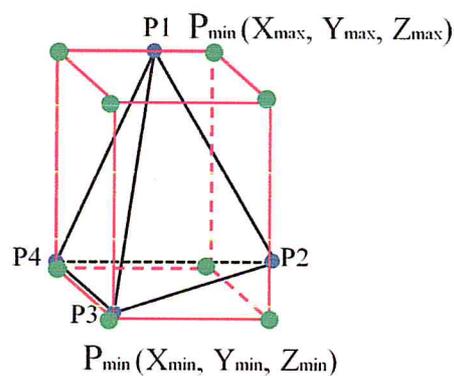
**Figure 54 :** Sphère circonscrite, rayon et centre de la sphère.

5. Entre un (01) et quatre (04) voisins (voisin 1, voisin 2, voisin 3, voisin 4)



**Figure 55 :** Représentation d'un tétraèdre voisin.

6. les coordonnées limites d'extrémités de sa boîte englobante ( $X_{min}$ ,  $X_{max}$ ,  $Y_{min}$ ,  $Y_{max}$ ,  $Z_{min}$ ,  $Z_{max}$ ).



**Figure 56 :** Représentation du brut englobant.

### 5.4 Insertion d'un point:

Le processus d'insertion d'un point est illustré par la Figure 53.

1. Insertion d'un point suivant le mode séquentiel ou suivant le mode aléatoire.
2. Vérification de l'appartenance du point inséré au tétraèdre.

3. Vérification de l'appartenance du point à l'une des faces du tétraèdre.
4. Substitution du tétraèdre actuel par quatre nouveaux tétraèdres (split).

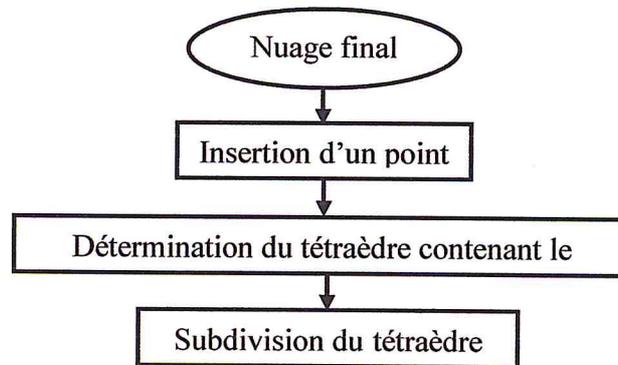


Figure 57 : Organigramme d'insertion d'un point

### 5.4.1 Identification du tétraèdre contenant le point inséré:

Ce traitement permet l'identification du tétraèdre contenant le point inséré. Pour cela, les opérations suivantes sont réalisées:

1. Calcul de la normale  $Normale (X_{normal}, Y_{normal}, Z_{normal})$  de chaque face d'un tétraèdre donné.
2. Détermination des paramètres de l'équation du plan de la face :  
 $Equation = A * X_{normal} + B * Y_{normal} + C * Z_{normal} + D.$
3. Calcul du centre de gravité du tétraèdre :

$$X_{centre\_gravité} = \frac{X_{point1} + X_{point2} + X_{point3} + X_{point4}}{4}$$

$$Y_{centre\_gravité} = \frac{Y_{point1} + Y_{point2} + Y_{point3} + Y_{point4}}{4}$$

$$Z_{centre\_gravité} = \frac{Z_{point1} + Z_{point2} + Z_{point3} + Z_{point4}}{4}$$

4. Remplacement des coordonnées du point inséré et du centre de gravité du tétraèdre courant dans l'équation du plan d'une face.
5. Si le produit des deux résultats est positif :
  - Vérification de la face suivante.
  - Si les résultats des quatre faces sont positifs, alors le point est à l'intérieur de ce tétraèdre.
6. Si non, vérification du tétraèdre commun à cette face (dont le résultat est négatif).
  - Refaire le même traitement pour les faces du tétraèdre courant, jusqu'à détermination du tétraèdre contenant le point.

## 5.5 Subdivision du tétraèdre (Split)

Lors de la subdivision du tétraèdre, quatre cas sont à considérer :

### 5.5.1 Point inséré est confondu avec un des sommets du tétraèdre

- Dans ce cas, le point n'est pas pris en considération et il est supprimé (Figure 54).

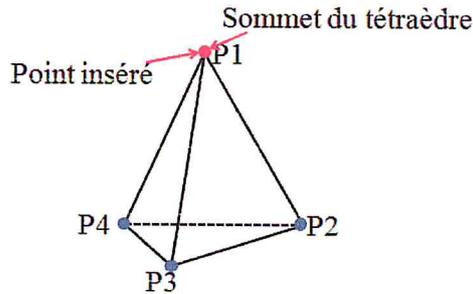
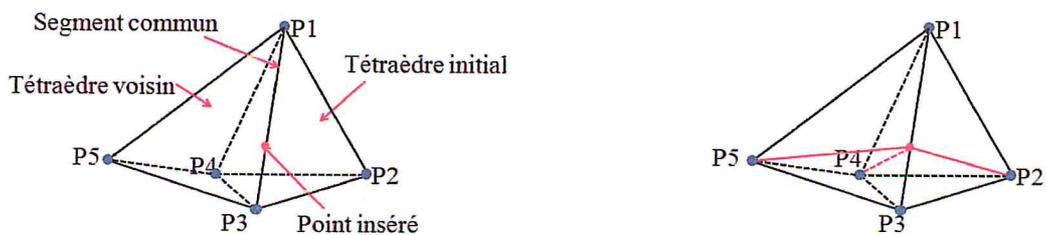


Figure 58: Point inséré sur le sommet

### 5.5.2 Point inséré est sur un des segments du tétraèdre

1. Le tétraèdre est subdivisé en créant deux nouveaux tétraèdres.
2. Supprimer le tétraèdre initial.
3. Chercher tous les tétraèdres ayant ce segment en commun
  - Subdiviser chaque tétraèdre en en créant deux nouveaux tétraèdres.
  - Supprimer chaque tétraèdre initial.
4. Déterminer les voisins pour tous les tétraèdres de la liste.

La Figure 55 montre le processus de subdivision d'un tétraèdre en deux tétraèdres.



a). Segment contenant le point

b). Subdivision des tétraèdres en deux

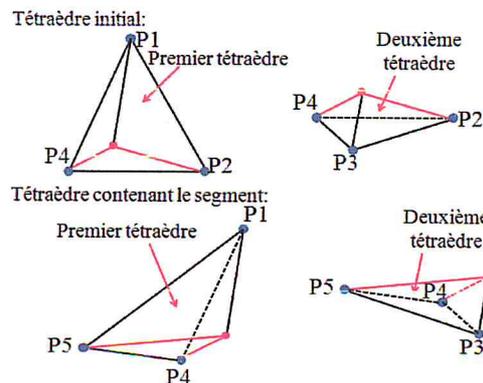


Figure 59 : Split pour un point inséré sur le segment d'un tétraèdre

### 5.5.3 Point inséré est sur une des faces du tétraèdre

1. Détermination du voisin commun à cette face (Figure 60a)
2. Subdivision du tétraèdre voisin en créant trois nouveaux tétraèdres (Figure 60b).
3. Suppression du tétraèdre voisin.
4. Subdivision du tétraèdre initial en créant trois nouveaux tétraèdres.
5. Suppression du tétraèdre initial.
6. Détermination des voisins pour tous les tétraèdres de la liste.

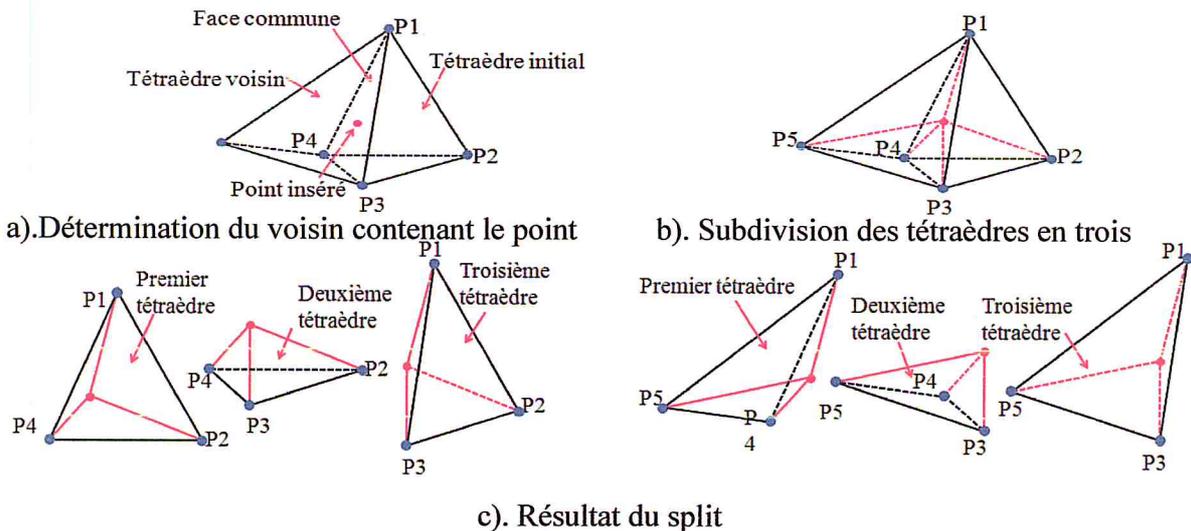


Figure 60: Split pour un point inséré sur la face d'un tétraèdre

### 5.5.4 Point inséré est à l'intérieur du tétraèdre

1. Subdivision du tétraèdre en quatre tétraèdres.
2. Suppression du tétraèdre initial.
3. Détermination des voisins pour tous les tétraèdres de la liste.

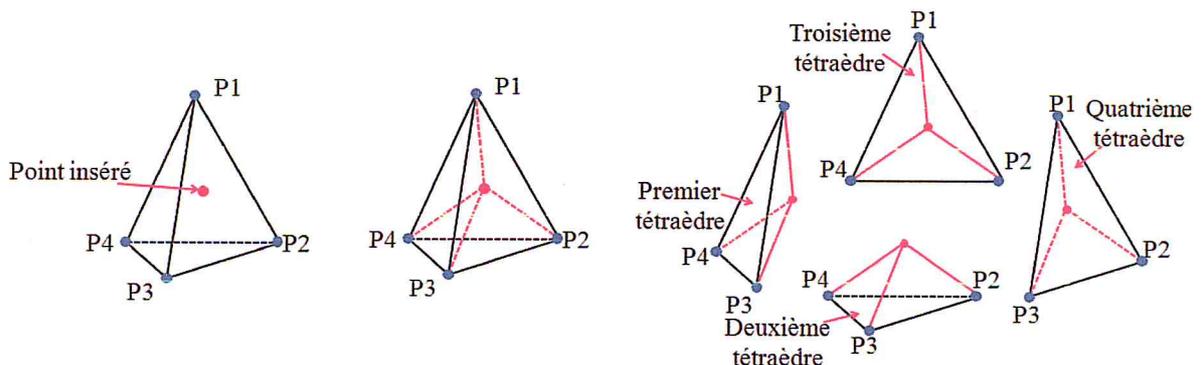


Figure 61 : Split pour un point à l'intérieur du tétraèdre

## 5.6 Vérification des critères de Delaunay :

La vérification des critères de Delaunay consiste à (Figure 62):

1. Pour chaque tétraèdre, détermination de sa sphère circonscrite.

2. Vérification de l'appartenance d'un point de ses voisins à la sphère circonscrite (par la suite on appellera ce point 'point 5').
3. Si oui:
  - Réalisation d'un « FLIP » entre ces deux (02) tétraèdres.
  - Stockage des tétraèdres résultant dans un tableau et suppression des anciens.
  - Mise à jour et stockage des voisins des tétraèdres résultant ainsi que leurs voisins.

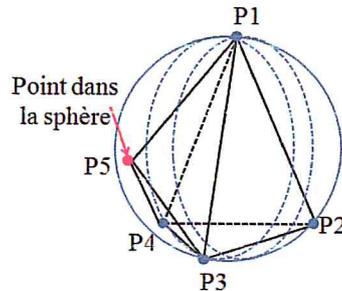


Figure 62 : Cas d'un point à l'intérieur de la sphère circonscrite

### 5.7 Réalisation des FLIPs :

Si le critère de Delaunay n'est pas vérifié, on doit effectuer des FLIPs pour corriger la triangulation 3D. On a trois (03) types de FLIPs : FLIP 2-3, FLIP 3-2 et le FLIP 4-4.

Pour déterminer quel FLIP utiliser on procède au cône test.

#### 5.7.1 Cône test

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

1. On remplace les coordonnées du point 5 dans les équations des plans des trois (03) faces constituant ce cône.

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

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

$$plan3 = X_{Normale3} * X_{point5} + Y_{Normale3} * Y_{point5} + Z_{Normale3} * Z_{point5} + D3$$

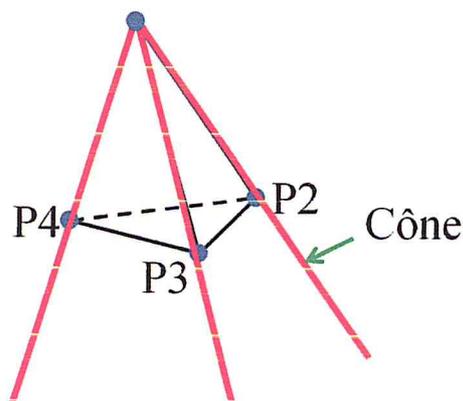


Figure 63 : Cône formé par trois faces du tétraèdre

## Conception de l'application

2. On calcule le produit des trois (03) équations :

$$\text{Resultat} = \text{plan1} * \text{plan2} * \text{plan3}.$$

- Si  $\text{Resultat} < 0$  : le point est à l'intérieur du cône donc on effectuera un « FLIP 2-3 » (Figure 60a).
- Si  $\text{Resultat} > 0$  : le point est à l'extérieur du cône donc on effectuera un « FLIP 3-2 » (Figure 60b).
- Si  $\text{Resultat} = 0$  : le point est sur une des faces du cône donc on effectuera un « FLIP 4-4 » (Figure 60c).

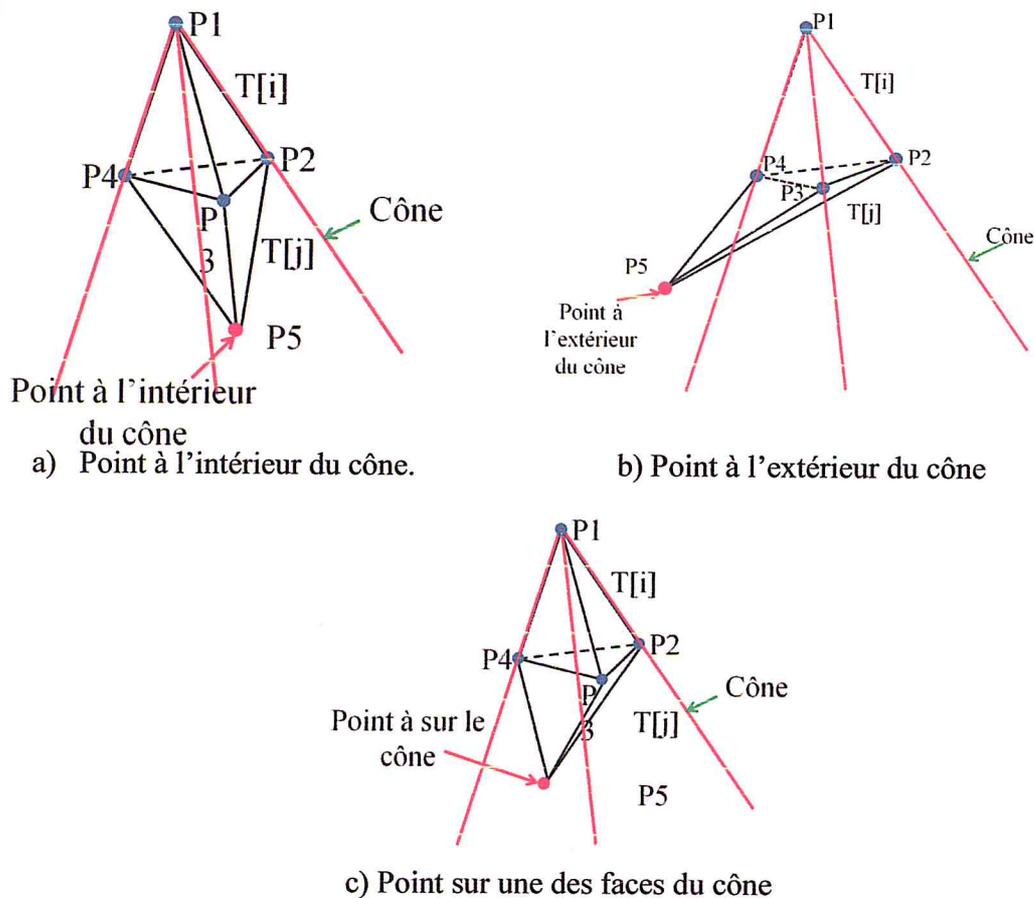


Figure 64 : Les trois cas de position du point 5 par rapport au cône.

### 5.7.1 FLIP 2-3

Ce FLIP permet de passer de deux tétraèdres à trois tétraèdres. Il est effectué suivant les étapes suivantes (Figure 65) :

1. Vérification de l'appartenance du point au cône (intérieur).
2. Si oui, création de trois tétraèdres à partir des deux initiaux.
3. Suppression des deux tétraèdres initiaux.

## 4. Détermination des voisins pour tous les tétraèdres de la liste.

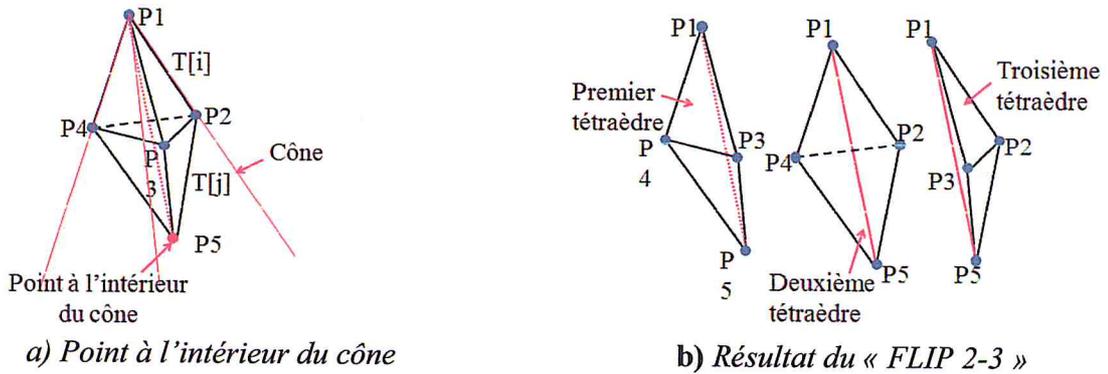


Figure 65 : FLIP convexe '2-3'

### 5.7.2 FLIP 3-2

Ce FLIP permet de passer de trois tétraèdres à deux tétraèdres (Figure 66) :

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 ( $T[i]$  et  $T[j]$ ).
3. Si oui :
  - Création de deux tétraèdres à partir des trois initiaux.
  - Suppression des trois tétraèdres initiaux.
  - Détermination des voisins pour tous les tétraèdres de la liste.
4. Si non, ce cas ne sera pas traité, d'autres FLIPs régulariseront ce problème par la suite.

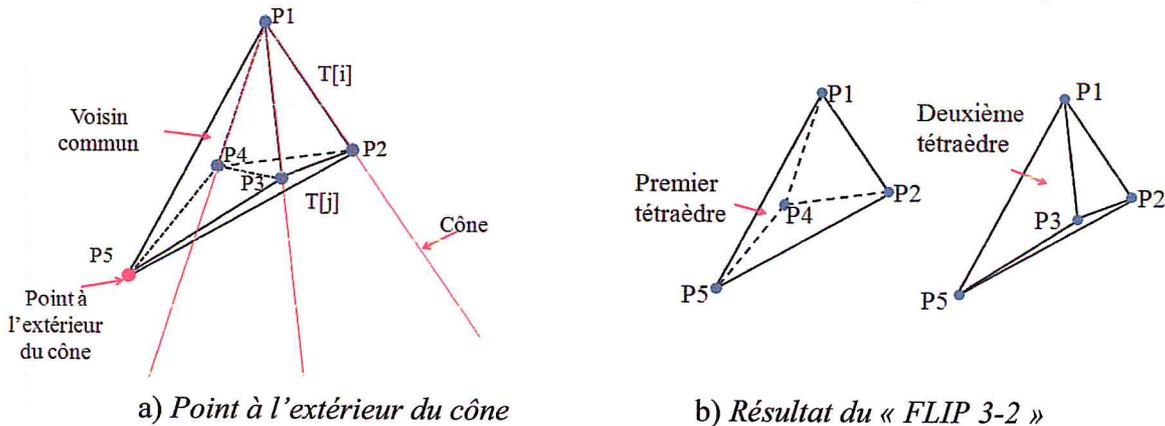


Figure 66 : FLIP non convexe '3-2'.

### 5.7.3 FLIP 4-4

1. Vérifier si le point est sur le plan d'une des faces du tétraèdre initial.
  - On obtient la face commune entre le tétraèdre initial et le tétraèdre voisin « j ».
2. Déterminer le voisin « j ».
3. Déterminer le voisin commun entre le 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.
  - Supprimer les deux tétraèdres initiaux.
  - Déterminer les voisins pour tous les tétraèdres de la liste.
5. Si non, ce cas ne sera pas traité, d'autres FLIPs régulariseront ce problème par la suite.

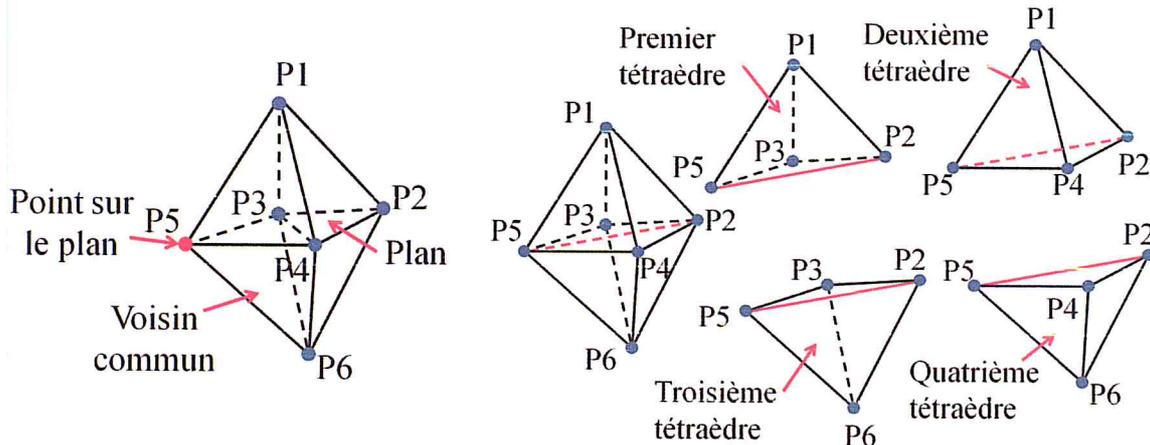


Figure 67 : FLIP point sur le plan d'une des faces du tétraèdre '4-4'.

### 5. Suppression des points fictifs

Une fois la triangulation terminée, on procède à la suppression des points fictifs :

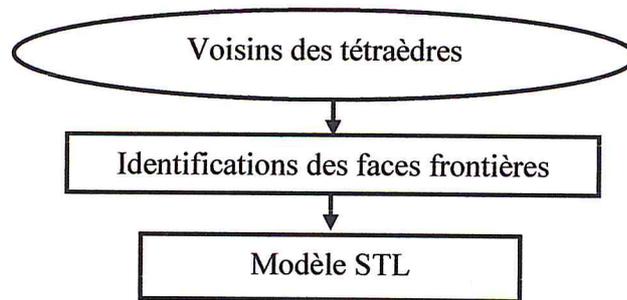
1. Parcourir la liste des tétraèdres résultant :
  - Si un des quatre points du tétraèdre est fictif alors on le supprime.
2. Après avoir tout parcouru on vérifie le critère de Delaunay.

### 6. Génération du fichier STL:

L'objectif final de l'application est la génération du modèle STL de l'objet digitalisé pour être exploité dans la phase de fabrication. La génération du modèle STL suit l'étape de validation de la triangulation de Delaunay 3D et consiste à approximer le nuage de points non structuré par des triangles qui représentent la peau externe de l'objet. Ceci est fait de la manière suivante (Figure 68):

1. Calcul des voisins de tous les tétraèdres.
2. Recherche des faces dont les voisins sont nuls (le tétraèdre ne partage cette face avec aucun tétraèdre).
3. Sauvegarde de la face identifiée.
4. Sauvegarde des faces en suivant la structure du fichier STL.

### 5. Visualisation du modèle STL.



**Figure 68 :** Organigramme de génération du fichier STL

## Conclusion

Dans ce chapitre on a expliqué en détaille la partie conception, citer tous les détailles 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 IV

**TEST**

**ET**

**REALISATION**

### **Introduction :**

La réalisation de notre travail s'achève par une présentation de l'application conçue, ainsi que l'environnement de développement et les outils utilisés pour sa mise en œuvre. Puis une présentation de l'interface utilisateur et une évaluation du système à travers un exemple de validation.

Ce chapitre a pour but de donner une vue à la fois intelligible et relativement concrète sur notre application en présentant les informations manipulées à travers les sessions d'utilisation appliquées à un exemple de validation. Certainement, un exemple ne peut refléter l'intérêt exacte de l'application, les performances de notre système, mais il peut donner une idée générale sur son intérêt surtout sur la complexité de la reconstruction d'objet 3D et les différentes techniques utilisées pour le respect de la contrainte de Delaunay.

Dans la première partie, le langage utilisé et les différentes fenêtres de notre application sont présentées. Quand à la deuxième partie, les scénarios les plus généraux appliqués sur une pièce donnée illustré par des captures d'écrans, commençant par un nuage de points jusqu'à la génération du model STL, constituent la partie validation.

### **1. langage utilisé :**

La majorité des systèmes de CFAO sont développés en langage C++. Afin de se conformer à cette règle, et afin d'intégrer notre application à l'environnement de reconstruction d'objet développée par l'équipe CFAO du CDTA, où tous les modules sont développés en C.

Le langage C++, inventé par Bjarne Stroustrup vers 1983, est une évolution orientée objets du langage C de Brian Kernighan et Denis Ritchie. Il s'est enrichi, au cours de la décennie 1980, parallèlement à la stabilisation et la normalisation de C.

Ce langage repose sur les mêmes mécanismes d'écriture et de génération, 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 Templates (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. Elle comporte notamment de nombreux patrons de classes et de fonctions permettant de mettre en œuvre les structures de données les plus importantes (vecteurs dynamiques, listes chaînées, chaînes...) et les algorithmes les plus usuels [19].

Parmi les environnements de développements nous citons: AnjutaDevStudio , C++ Builder , Code:: Blocks (open-source) , Dev-C++, Eclipse (open-source) , Microsoft Visual C++ , etc [20].

### **1.1. Builder C++ :**

C ++ Builder est un environnement de développement d'applications rapide (RAD) qui permet aux développeurs de logiciels de développer un code avec une vitesse et une productivité accrues. L'environnement est emballé avec un certain nombre de composants qui rendent le codage logiciel plus simple et plus rapide. C ++ Builder est destiné à la plupart des plates-formes modernes et des systèmes d'exploitation.[27]

On a utiliser Builder C++ version 6.

### **1.2 C++ Builder 6 :**

C'est la parfaite alliance entre la simplicité d'un RAD qui a fait ses preuves depuis des années, et l'utilisation du C++ sans concession. La richesse et la qualité des composants de C++Builder 6 vous permet de vous concentrer sur les parties de code vraiment utile et d'aller droit à l'essentiel. Par la diversité des technologies utilisées C++Builder 6 vous permet de concevoir des applications fiables et compatibles avec les standards les plus récents. C++Builder 6 est un outil précieux dans le monde du développement Professionnel où vitesse et fiabilité sont de rigueur. [28]

### **1.3 OpenGL**

Open Graphic Library ou OpenGL est une API graphique open-source, de bas niveau, dédiée pour les applications générant des images 2D ou 3D. OpenGL est sortie en 1992 par la SiliconGraphicsInc (ou SGI) et est actuellement géré par la société Khronos Group.

OpenGL est conçu 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, ouvert et portable, la bibliothèque graphique a pu se vendre dans le milieu scientifique et industrielle, et être utilisé dans des nombreuses applications artistiques ou de traitement d'images [23].

Du fait de ses performances, l'API OpenGL est aussi présent dans différents domaines jeux-vidéo ludiques, les applications web, les systèmes embarqués et les téléphones portables.

C'est une API très performante et simple d'utilisation, qui permet de réaliser des rendus et des scènes complexes, selon les besoins. Cette technologie est ouverte et est implémenté par de nombreuses bibliothèques, sous des langages de programmations différentes (dont des langages de script comme Python ou Ruby) [24].

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 [25]

## 2. Présentation des fenêtres

L'application principale est composée d'un environnement qui englobe dans ses onglets toutes les applications du projet.

### 2.1 Présentation de l'application développée

L'application développée permet la reconstruction d'objet en trois (03) D à partir d'un nuage de points quelconques, en utilisant la triangulation de Delaunay implémentée par la méthode de « FLIP ».

La fenêtre principale de notre projet est composée de 6 onglets, chaque onglet est spécialisé dans une phase du processus de génération de la triangulation Delaunay :

1. Lecture du fichier.
2. Subdivision.
3. Simplification.
4. Première tétraédrisation.
5. Deuxième tétraédrisation et génération du fichier STL.

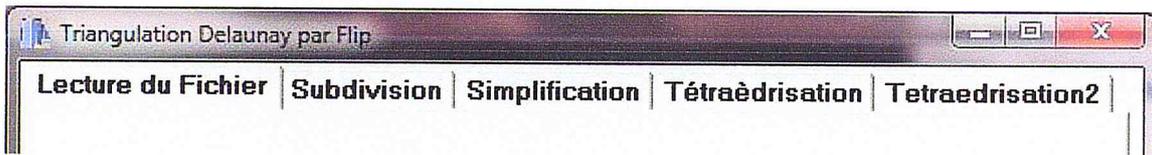


Figure 69 : Onglets de la fenêtre principale.

### 2.1.1 Lecture du fichier du nuage de points

Le premier onglet « Lecture du fichier » permet la lecture du fichier contenant le nuage de points par simple clic sur le bouton « Ouvrir le fichier » (Figure 70).

Après la lecture du fichier, les dimensions ( $X_{min}$ ,  $X_{max}$ ,  $Y_{min}$ ,  $Y_{max}$ ,  $Z_{min}$ ,  $Z_{max}$ ) du brut sont affichées, ainsi que le nombre de points.

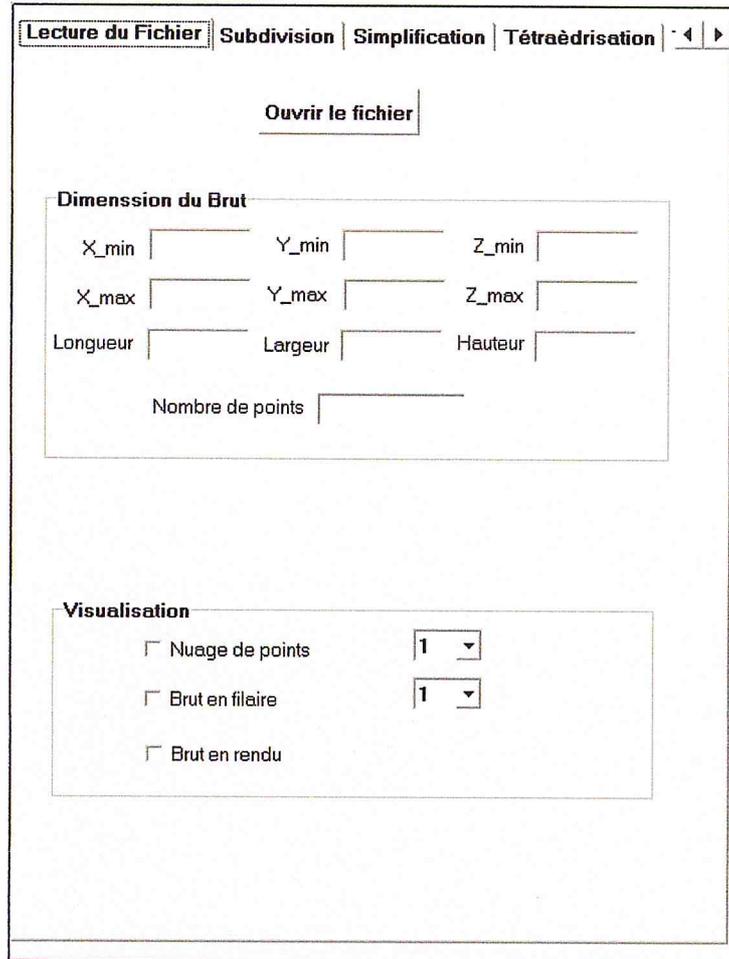


Figure 70 : Onglets de la fenêtre principale.

### 2.1.2 Subdivision du brut

Cet onglet est consacré à la segmentation. Il concerne la première partie de la simplification du nuage de points.

Comme cela a été expliqué dans le chapitre précédent, la première étape est la subdivision du brut généré dans l'onglet précédent (Figure 71) en cellules, et l'affectation des points du nuage aux cellules correspondantes.

Puis une cellule sélectionnée est subdivisée en patches (sous cellules) en effectuant les mêmes étapes que pour la création des cellules. Lors de la création des cellules l'utilisateur fixe soit : le nombre de cellules selon les directions x, y et z, soit le pas de la cellule.

Ces mêmes critères sont appliqué pour la creation des patches.

Lecture du Fichier | **Subdivision** | Simplification | Tétraèdrisation

**Paramètre cellules**

<input checked="" type="radio"/> Nombre de cellules		<input type="radio"/> Pas		<b>Nouveaux Pas</b>	
Selon X	2	Selon X	20	Selon X	
Selon Y	2	Selon Y	20	Selon Y	
Selon Z	2	Selon Z	20	Selon Z	

**Creation des cellules**

---

**Visualisation**

<input type="radio"/> Cellule globale		<input type="radio"/> Une Cellule	
<input type="checkbox"/> Cellules en filaire	1	Selon i	Selon j
<input type="checkbox"/> Cellules en rendu		0	0
<input type="checkbox"/> Affectation des points	1	Selon k	0
		<input type="checkbox"/> Brut globale	
		<input type="checkbox"/> Filaire	1
		<input type="checkbox"/> Rendu	
		<input type="checkbox"/> Affectation des Points	1

**Patch**

<input type="radio"/> Automatique		<input checked="" type="radio"/> Nombre de patches		<input type="radio"/> Pas		<b>Nouveaux Pas</b>	
Selon X	5	Selon X	2	Selon X	2	Selon X	
Selon Y	5	Selon Y	1	Selon Y	2	Selon Y	
Selon Z	5	Selon Z	1	Selon Z	2	Selon Z	

**Patches pour une cellule**

---

**Un Patch**

**Selectionner une cellule**

Selon x	Selon y	Selon z
0	0	0
<input type="checkbox"/> Filaire patches une cellule	1	
<input type="checkbox"/> Rendu		

**Selectionner un patch**

Selon x	Selon y	Selon z
0	0	0
<input type="checkbox"/> Filaire un patch	1	
<input type="checkbox"/> Affectation des Points	1	

Création des cellules

Création des patches pour une cellule

Figure 71 : Fenêtre première étape de simplification.

## 2.1.3 Simplification

La deuxième partie du processus de la simplification est présentée dans l'onglet simplification (Figure 72). Cette simplification permet de supprimer les points redondants ainsi que les points qui ne remplissent pas la contrainte de distance introduite par l'utilisateur.

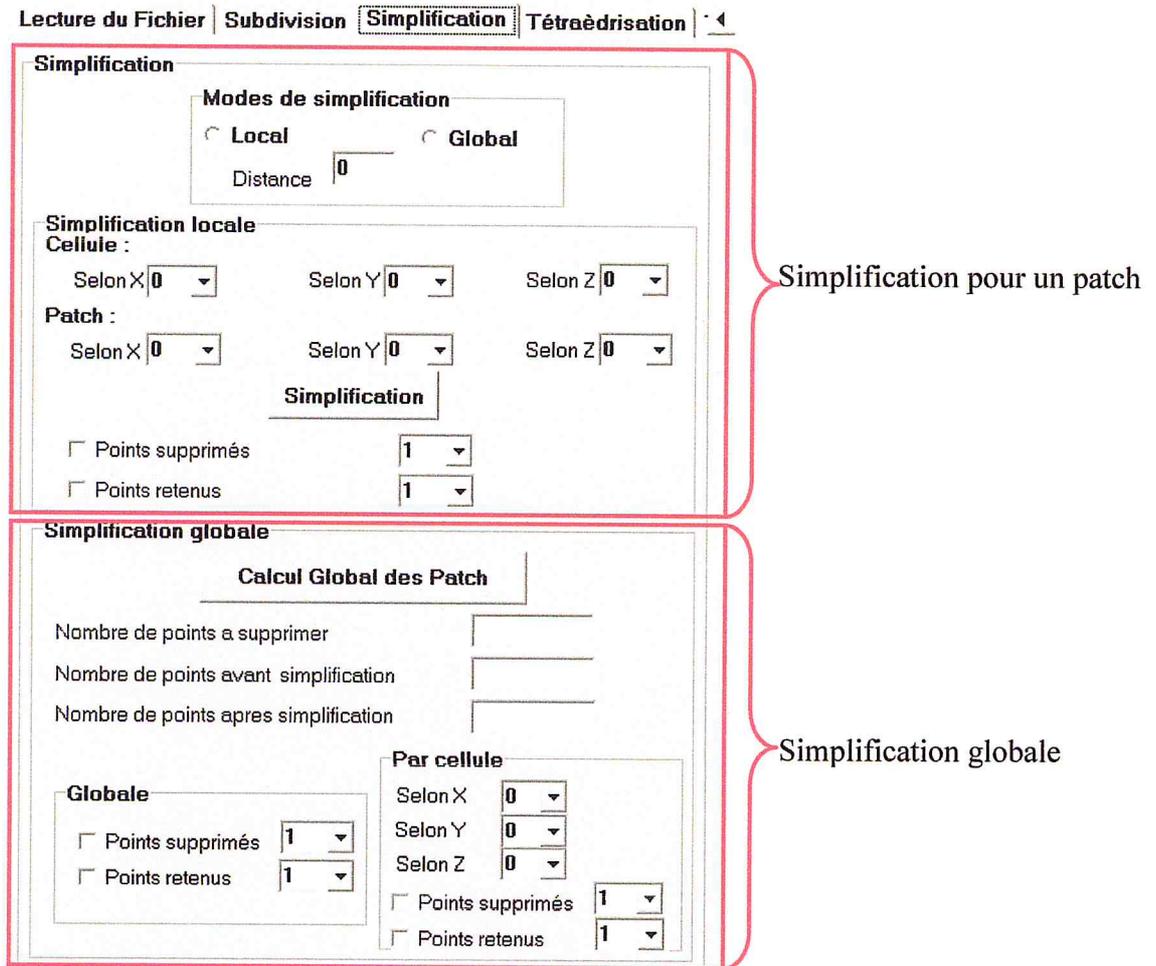


Figure 72 : Fenêtre de simplification.

## TEST ET REALISATION

### 2.1.4 Première tétraédrisation

Cet onglet est réservé à la première partie de la tétraédrisation de Delaunay. Celle-ci, peut se faire avec un seul tétraèdre englobant tout le nuage de points, ou avec douze (12) tétraèdres englobant. La présence de deux approches a été justifiée précédemment.

L'utilisation de douze (12) tétraèdres, permet la visualisation des tétraèdres, leurs voisins, leurs sphères circonscrite ainsi que les normales. Il permet en outre la vérification des critères de Delaunay.

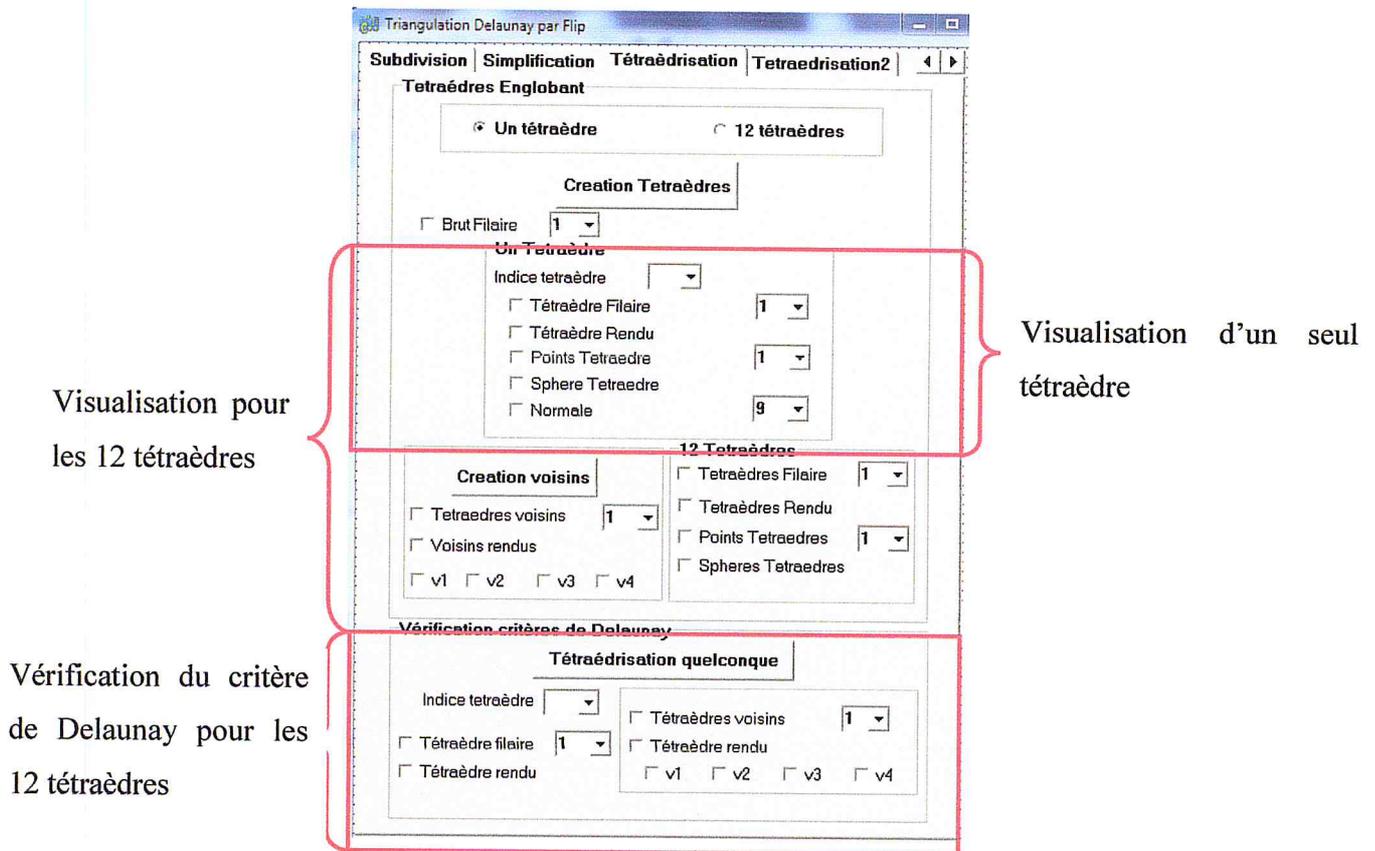


Figure 73 : Fenêtre de la première tétraédrisation.

### 2.1.5 Deuxième tétraédrisation

Deux modes d'insertion sont intégrés à cet onglet: « Manuel » ou « Automatique » Le mode manuel consiste à insérer point par point soit séquentiellement ou aléatoirement le nuage de points, subdiviser le tétraèdre contenant le point insérer et vérifier les critères de Delaunay.

Le mode automatique consiste à insérer les points de manière séquentielle, subdiviser et vérifier automatiquement les critères de Delaunay.

Après l'insertion, le modèle final est généré par suppression des points fictifs ajouté lors de la création des tétraèdres englobant.

Par la suite le modèle STL est généré.

Subdivision | Simplification | Tétradratisation | **Tétradratisation2** | < | >

Mode d'exécution  
 Manuel     Automatique

**Insertion d'un point**  
 Séquentiel     Aléatoire

**Insérer**

Indice point: \_\_\_\_\_    Indice tetraedre: \_\_\_\_\_  
 Dessiner point [1]     Tetraedre contenant point

**Critères Delaunay**  
**Subdivision**    **Flip**

indice tetraedres: \_\_\_\_\_

Tetraedres Filaire [1]    **Voisins**  
 Tetraedres Rendu     Voisin filaire [1]  
 Sphere tetraedres     Voisin rendu  
 v1     v2     v3     v4

---

**Automatique**  
**Tétradratisation de Delaunay**

0%

**finalisation**

Tétradrès filaire [1]  
 Tétradrèdre rendu  
 Normales tétradrès [9]

**Générer modèle STL**    **Vérifier modèle STL**

**Enregister modèle STL**

dessiner STL     Normales STL [9]

Mode d'exécution manuel

Mode d'exécution automatique

Figure 74 : Fenêtre de la deuxième tétraédration.

### 3. validation:

La validation de notre travail est effectuée sur l'exemple de pièce présenté ci-dessous. Le traitement effectué sur le nuage de point issus de la digitalisation de l'objet jusqu'à la génération d'un modèle STL représentant d'une manière plus ou moins fidèle a l'objet cible est présenté dans la suite.



Figure 75 : La pièce scannée.

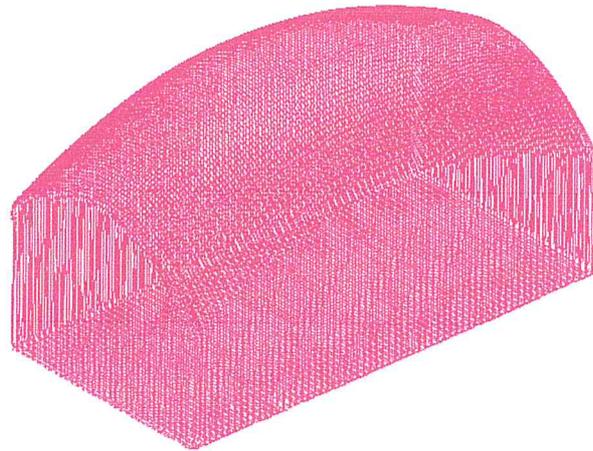


Figure 76 : Résultat après le scan.

## 3.1 Lecture nuage de point

La première étape consiste à ouvrir le fichier contenant le nuage points, calculer les limites du brut. Le nuage de points contient 322787 points.

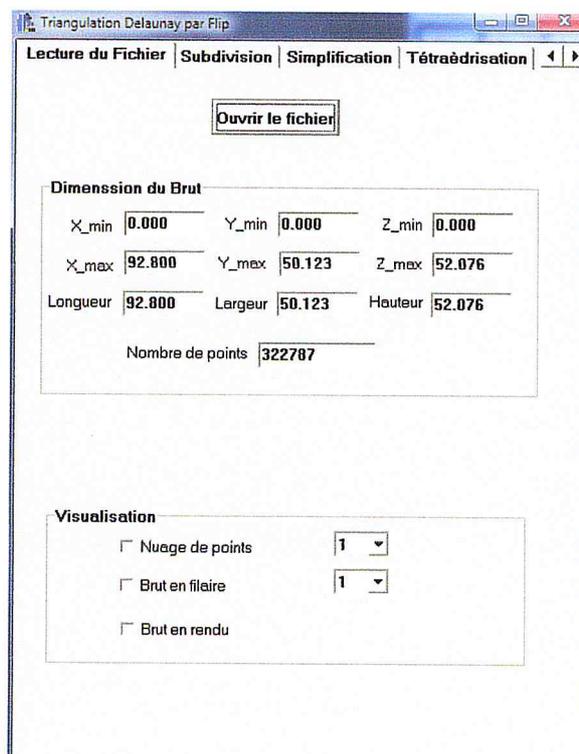
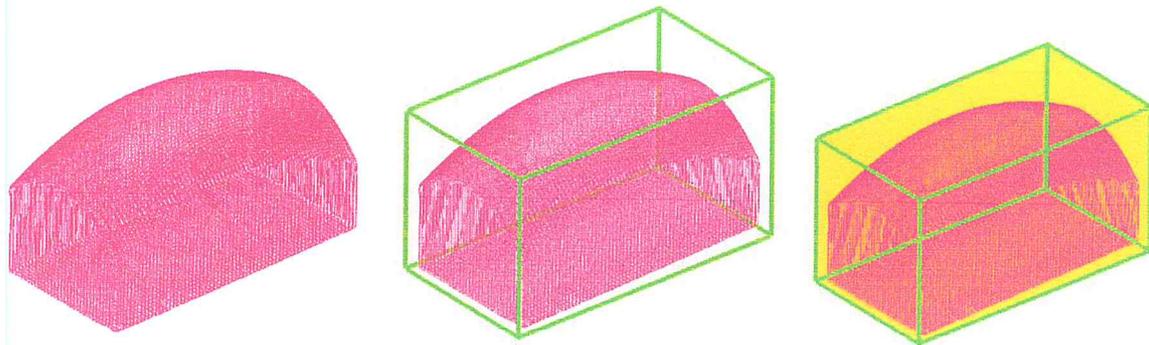


Figure 77 : Lecture du fichier de nuage de points



a. Nuage de point

b. Brut en filaire

c. Brut en rendu

Figure 78 : Visualisation du nuage.

### 3.2 Subdivision:

#### a. pour les cellules:

La subdivision est la première étape vers la simplification, la subdivision des cellules précède celle des patches. Le test a été effectué pour 3 cellules selon X, 3 cellules selon Y et 3 cellules selon Z.

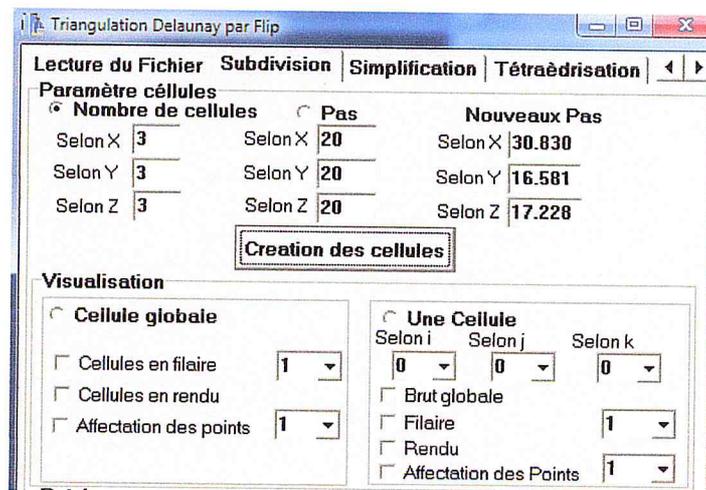
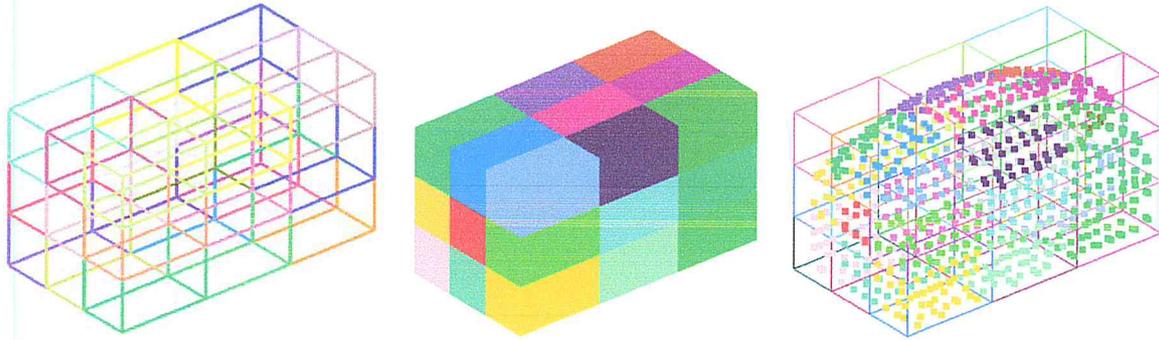


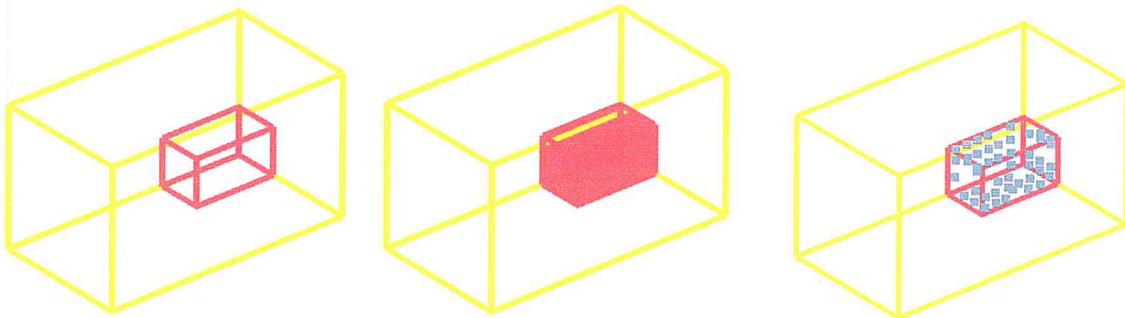
Figure 79 : Première subdivision spécification le nombre de cellules.

Le bouton création des cellules permet de générer des cellules en fonction des informations introduites par l'utilisateur. Par la suite le nouveau pas est calculé.



a. Cellules en filaire.                      b. Cellules en rendu.                      c. Points dans les cellules

**Figure 80 :** Visualisation de toutes les cellules.



a. Une cellule en filaire.                      b. Une cellule en rendu.                      c. Points dans la cellule.

**Figure 81 :** Visualisation d'une seule cellule.

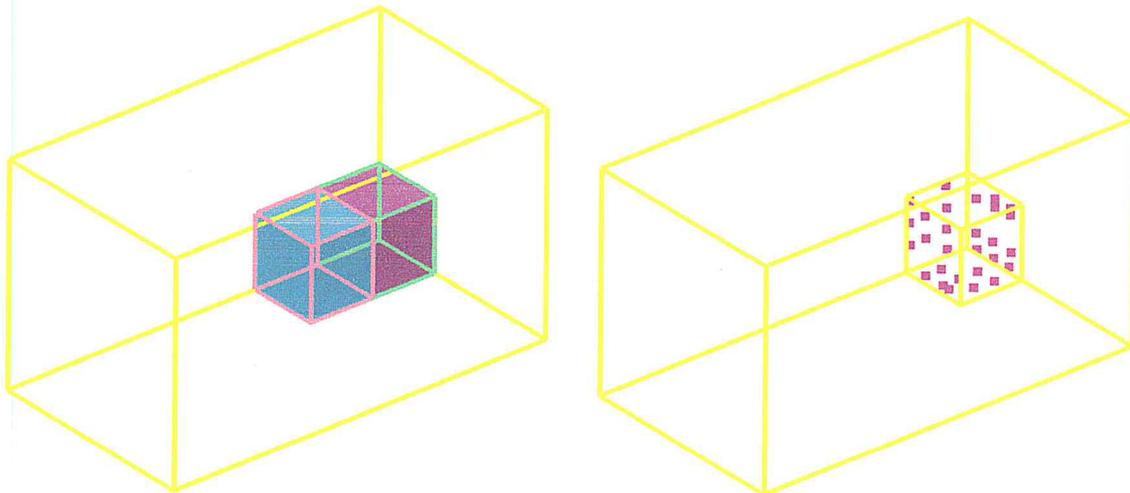
Plus le nombre de cellule est élevé plus le traitement est rapide.

**b. pour les patches:**

La création des patches se fait de la même manière de celle des cellules, sauf que dans ce cas l'utilisateur introduit le nombre de patches pour une seule cellule et puis une translation sera faite pour l'ensemble des cellules. Le nombre de patch utilisé dans ce test est de 2 patches selon X ,1 patch selon Y et 1 patch selon Z.

Patch			
<input type="radio"/> Automatique	<input checked="" type="radio"/> Nombre de patches	<input type="radio"/> Pas	Nouveaux Pas
Selon X   5	Selon X   2	Selon X   2	Selon X   15.415
Selon Y   5	Selon Y   1	Selon Y   2	Selon Y   16.581
Selon Z   5	Selon Z   1	Selon Z   2	Selon Z   17.228
<b>Un Patch</b> Sélectionner une cellule Selon x   0   Selon y   0   Selon z   0 <input type="checkbox"/> Filaire patches une cellule   5   <input type="checkbox"/> Rendu Sélectionner un patch Selon x   0   Selon y   0   Selon z   0 <input checked="" type="checkbox"/> Filaire un patch   5   <input checked="" type="checkbox"/> Affectation des Points   8.5			

**Figure 82 :** Fenêtre de création des patches.



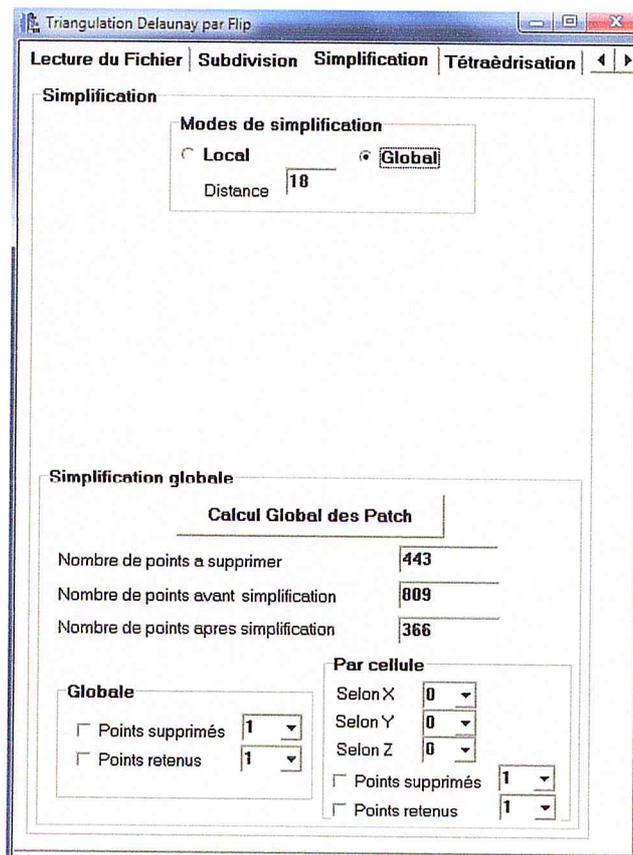
a. les patches dans une cellule.

b. points dans un seul patch.

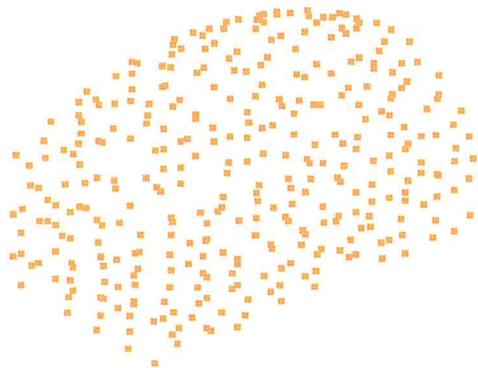
**Figure 83** : Visualisation des patches

### 3.3 Simplification:

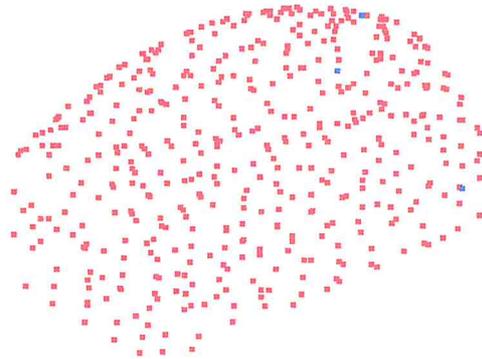
Le mode de simplification choisi est le mode global. Il permet de lancer la simplification de tout le nuage à la fois. Pour une distance de simplification égale à 18mm, le résultat obtenu est de 366 points.



**Figure84** : Test de simplification avec 18 mm.



a. points retenu.



b. points supprimé.

Figure 85 : Nuage après simplification.

Plus la distance de simplification est importante plus le temps d'exécution est rapide.

### 3.4 Première Tétraédrisation:

Le test est effectué en commençant par un seul tétraèdre englobant le nuage de points.

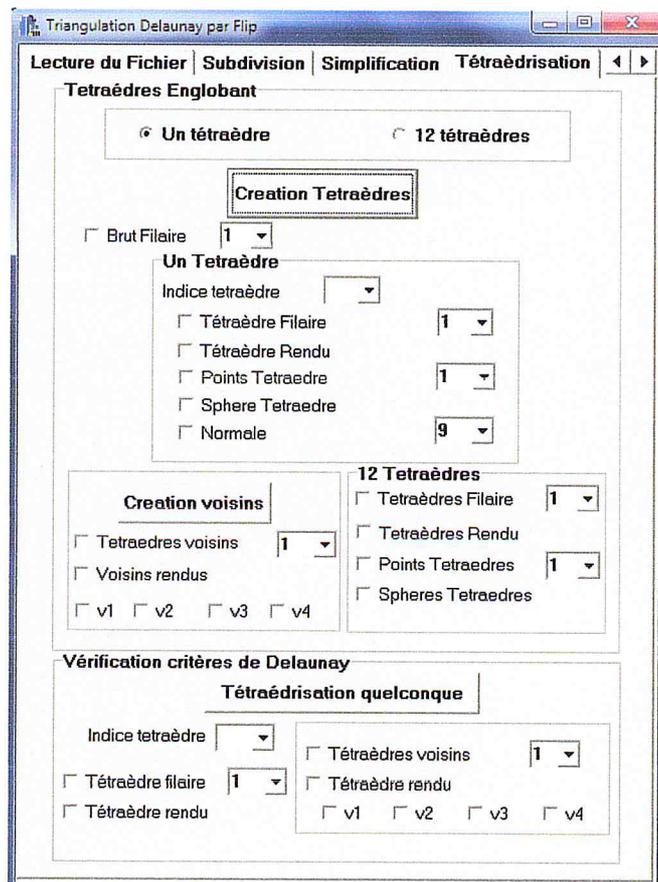


Figure 86 : Onglets « première tétraédration » choix d'un seul tétraèdre.

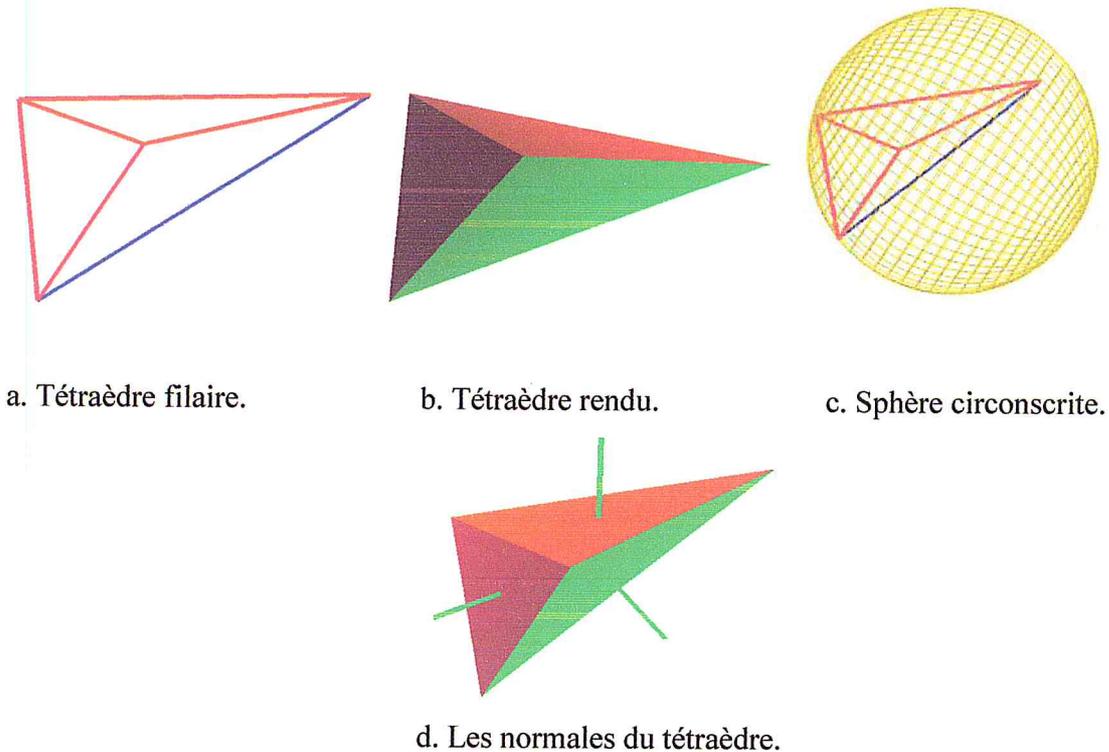


Figure 87 : Visualisation du tétraèdre englobant le nuage de points.

### 3.5 Deuxième Tetraedrisation:

Dans cette partie, le mode d'exécution automatique est adopté.

a. **Tétraédrisation finale** : le bouton Tétraédrisation de Delaunay permet de générer la tétraédrisation global pour tout le model.

Le bouton Finalisation sert à supprimer le tétraèdre englobant (fictif).

L'utilisateur peut visualiser l'ensemble des tétraèdres obtenu en filaire, en rendu et les normales des tétraèdres.

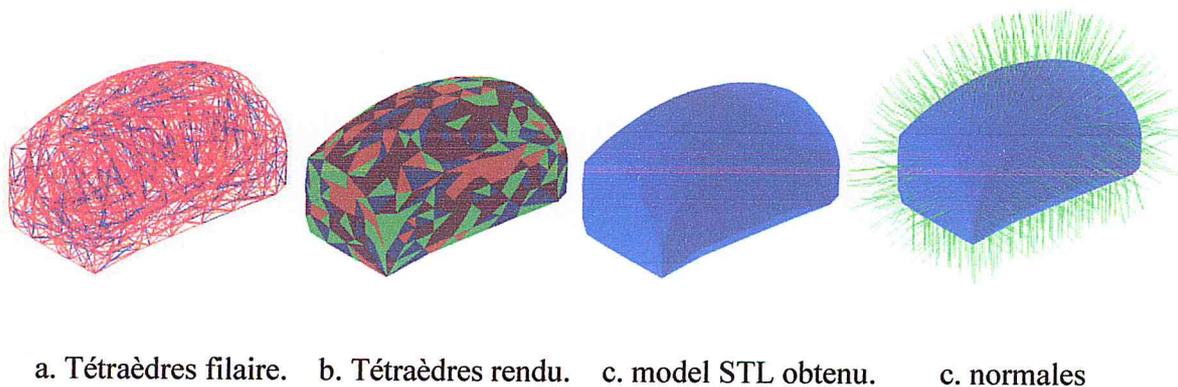


Figure 88 : Visualisation de la tétraédrisation finale.

## TEST ET REALISATION

**b. Générer le model STL :** le bouton Générer fichier STL permet de garder que les faces (triangles) extérieur du model.

Le bouton vérifier fichier STL permet de vérifier si le model STL obtenu est valide.

Le bouton Model STL sert à enregistrer le fichier STL.

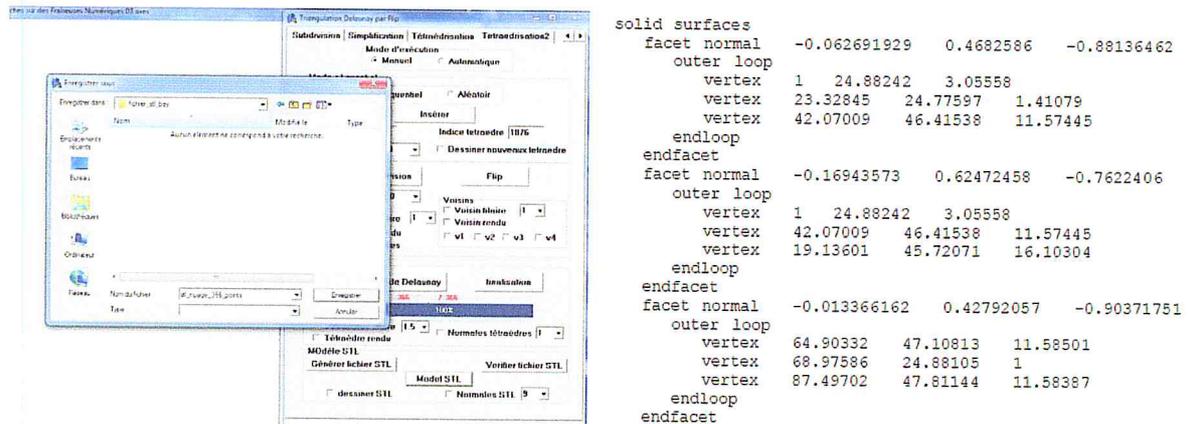


Figure 89 : Sauvegarde d'un fichier STL.

### 4. Etude comparative :

Dans cette partie une comparaison est faite entre les résultats obtenus, en prenant trois (03) distances de simplification appliquée sur un même nuage de points de 322787 points, ce test a été fait sur une machine contenant un processeur : Intel(R) Core(TM) i5-4200U CPU @ 0.60GHz, une mémoire vive (RAM) 4.00 Go.

Distance	16 mm	18 mm	22 mm
Nombre de points	621 points.	366 points.	195 points.
Model STL			
Temps d'exécution	1h et 3min.	15 min et 6s.	4 min.

Figure 90 : Tableau comparatif entre les différentes distances de simplification.

Dans ce tableau comparatif, une génération du model STL est faite en prenant différentes distances de simplification. La différence est remarquable entre les trois résultats, plus le nuage de points est dense plus le model obtenu est similaire au model voulu avec un temps

## TEST ET REALISATION

---

d'exécution élevé. Si le nombre de point supprimés est important, le taux de similarité entre le model généré et le model voulu est faible.

L'utilisateur doit prendre le choix de distance de simplification qui lui permet de gagné en terme de temps d'exécution (optimiser le temps), et garder la forme voulu.

### **Conclusion :**

Dans ce chapitre, les différentes étapes de notre application ont été présentées et le travail réalisé a été validé à travers un exemple concret de pièce de forme complexe. L'application a été déroulée depuis la lecture du fichier de nuage de points jusqu'à la génération du modèle STL.

La validation à travers un seul exemple ne peut refléter l'intérêt de l'application dans le domaine de la reconstruction d'objet surtout pour les applications du reverse engineering, mais elle constitue un nouveau maillon dans l'environnement de reconstruction d'objets développé par l'équipe de CFAO du CDTA.

## Conclusion générale:

---

### Conclusion générale:

Le travail réalisé au sein de l'équipe CFAO de la Division Productique et Robotique du CDTA s'articule autour de développement d'une application logicielle graphique et interactive pour la reconstruction des surfaces complexes à partir de nuage de points non structuré. Il s'agit précisément de la réalisation d'une application logicielle pour l'approximation de nuages de points par des éléments géométriques simples (triangle) en utilisant la méthode de flip et génération du modèle STL.

Le travail présenté dans ce mémoire a porté sur le processus du Reverse Engineering et plus précisément au niveau de la phase de reconstruction d'objets 3D. La méthode de Flip a été développée d'une part ; pour modéliser la tétraédrisation de Delaunay 3D, et d'autre part ; pour optimiser le processus de reconstruction.

Pour cela, nous avons tout d'abord introduit les notions de modélisation de surface, définit le Reverse Engineering et présenté les différentes méthodes concourant à la tétraédrisation de Delaunay.

Le deuxième chapitre, a été réservé à la description de la tétraédrisation de Delaunay par la méthode de FLIP et les troisièmes et quatrièmes chapitres ont porté sur la description explicite et l'implémentation de l'approche adoptée ainsi que les solutions proposées.

Le résultat de notre travail est une application logicielle qui prend en charge les points suivants:

- Simplification du nuage de points.
- Génération de la première tétraédrisation soit avec:
  - Un (01) super tétraèdre englobant.
  - Douze (12) tétraèdres englobant.
- Génération d'une triangulation 3D par la méthode de Flip pour générer une tétraédrisation Delaunay.
- Génération d'un modèle STL à partir de d'un ensemble de tétraèdres obtenu de la triangulation de Delaunay.

Arrivée au terme de ce travail, nous pouvons certifier que le respect des contraintes liées au problème de reconstruction d'objets 3D doit conduire à la réalisation d'un produit fidèle au modèle de référence de la maquette numérique, le modèle CAO. Pour parfaire, la qualité de ce travail, nous passons en revue les aspects liés aux améliorations éventuelles et les extensions à donner à ce travail :

## Conclusion générale:

---

1. Reconstruction des frontières de l'objet en utilisant la technique « alpha shape ».
2. Application de d'autres modèles d'optimisation dans la simplification du nuage de points qui prennent en considération les points vifs (fortes courbures).[28]
3. Développement de nouvelles méthodes de traitements pour améliorer la qualité de la triangulation.
4. Paralléliser le traitement des nuages de points volumineux.

## Références bibliographiques

- [1] Vanackère, P. (1992), Les modélisations géométriques utilisées dans les logiciels de dao/cao. In L'intégration de l'informatique dans l'enseignement et la formation des enseignants, pages 123–131. EPI-INRP.
- [2] AbdErahim, L & Zeghoudi, S. (2011). Décomposition d'objet 3d composé de mailles triangulaires par la courbe directe.
- [3] Bey, M. (2000, Avril). Modelisation des courbes et des surfaces, Laboratoire de Robotique et; Intelligence Artificielle Equipe Systèmes Robotisés de Production. P.21-27.
- [4] Brévilliers, M. (2008). Construction de la triangulation de Delaunay de segments par un algorithme de flip. PhD thesis, PhD thesis, Université de Haute Alsace, Mulhouse, France.
- [5] FARO. (2016). Le reverse engineering : principes, applications et solutions.  
<http://blog-fr.faro.com/2016/06/20/le-reverse-engineering-principes-applications-et-solutions>
- [6] <http://junior.universalis.fr/>
- [7] Marguin, O. (2003), C++ : LES BASES, p.1-2.
- [8] Autodesk. (2014, Aout). STL File. [www.docs.autodesk.com](http://www.docs.autodesk.com)
- [9] Bendifallah, M.E.H. (2011) « Approche pour la Finition des Surfaces Gauches par la Stratégie 'Gradient Conjugué' à Partir de Modèle STL ». Mémoire de Master, Faculté Des Sciences et Sciences de L'ingénieur- Université Amar Telidji Laghouat.
- [10] Challali, M. O., Belaidi, I., Mohammedi, K., Belaidi, A., & Ishiomin, G. Application et adaptation de la triangulation de Delaunay pour la reconstruction de surfaces. Surface reconstruction using and adapting Delaunay triangulation.
- [11] Bahi, A. (1997). Segmentation de surfaces représentées par des nuages de points non organisés (Doctoral dissertation).
- [12] 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.
- [13] Ait Mansour, M. Extraction d'éléments géométriques dans un nuage de points LiDAR terrestre : application aux relevés de façades.
- [14] Mansour, M. A. (2014). Extraction d'éléments géométriques dans un nuage de points LiDAR terrestre: application aux relevés de façades (Doctoral dissertation, G2S GeoSystem Surveying, 3 rue Marie Davy, 75014 Paris).

- [15] Céline ROUDET, Contribution de la reconstruction 3D à la compression de maillages surfaciques triangulaires, Rapport de stage – Master M2 Recherche Informatique, Université Claude Bernard Lyon 1, p. 13-14.
- [16] Gold, H. L. C. (2005). La modélisation de données océanographiques à l'aide du diagramme de Voronoï tridimensionnel. Revue internationale de géomatique. Volume, 1(1).
- [17] Hermeline, F. (1982). Triangulation automatique d'un polyèdre en dimension  $N$ . RAIRO-Analyse numérique, 16(3), 211-242.
- [18] Beyler, J C & Isenmann, B. (2006, Mai). Génération de terrain et triangulation de Delaunay. <http://fearyourself.developpez.com/tutoriel/jeu/Delaunay/>.
- [19] Schwartz, C. (2009). Contribution à l'élaboration d'un espace commun de représentation pour l'analyse morpho-fonctionnelle du membre supérieur: application à l'articulation glénohumérale (Doctoral dissertation, Ecole Nationale Supérieure des Télécommunications de Bretagne-ENSTB).
- [20] Valérie, C, (2008). Rapport amélioration de la représentation géométrique 2D et 3D des agrégations de poissons en support à l'étude de leur évolution spatio-temporelle, Université Laval
- [21] TAHHAN, H., & DOPCHIE, N. Rapport de stage ; Reconstitution de formes à partir d'un nuage de points.
- [22] Club des développeurs. (2014, Novembre). Incremental Delaunay Triangulation. <http://yahiko.developpez.com/tutoriels/triangulation-delaunay-incrementale/>.
- [23] Jiang, Z., Jaillet, F., & Zara, F. Reconstruction et complétion de maillages sous contraintes. Laboratoire d'Informatique en Image et Systèmes d'information, Université Claude Bernard Lyon 1
- [24] MOREAU, N. (2014, juin). Rapport de stage Génération de maillages volumiques et adaptables pour la simulation physique. Université Claude Bernard Lyon 1.
- [25] Introduction au Reverse Engineering Titre In [http://www-igm.univ\\_mlv.fr/~dr/XPOSE2013/reverse\\_engineering/generality.html](http://www-igm.univ_mlv.fr/~dr/XPOSE2013/reverse_engineering/generality.html).
- [26] Rajan, V.T. (1994). Optimality of the Delaunay triangulation in Rd. Discrete Computational Geometry 12.p. 189-202
- [27] <http://www.pcmag.com/encyclopedia/term/39117/c-builder>
- [28] BENHABILES, H. et AUBRETON, O. Simplification rapide de nuages de points avec préservation des caractéristiques vives.

