

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE D'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE  
HOUARI BOUMEDIENE  
FACULTE DE GENIE MECANIQUE ET GENIE DES PROCÉDES



## MEMOIRE

Présenté pour l'obtention de diplôme de **MAGISTER**

En : **Génie Mécanique**

Spécialité : **Mécanique et Fatigue des Matériaux**

par

**BOUCETTA Abdelkader**

Sujet

**SIMULATION PRE-CLINIQUE  
POUR OPTIMISATION DE LA PROTHESE FEMORALE  
EN APPLIQUANT LES MODELES DE LA CAO**

Soutenu publiquement le 17 avril 2005, devant le jury composé de :

Mr- T. BOUKHAROUBA	Professeur à l'USTHB	Président
Mr- S. RECHAK	Professeur à l'ENP	Directeur de thèse
Mr- N. OUALI	Professeur à l'USTHB	Examineur
Mr- A. NOUR	Professeur à l'U de Boumerdesse	Examineur
Mr- K. AZOUAOUI	Maître de conférence à l'USTHB	Examineur

## SOMMAIRE

<b>RESUME</b> .....	05
<b>INTRODUCTION GENERALE</b> .....	07

### *Chapitre 1*

#### MODELS MATHEMATIQUE DE LA CAO

<b>NOTIONS DE BASE</b> .....	13
<b>1.1 - MODEL DE BEZIER</b> .....	13
1.1.1 - Courbes de BEZIER .....	13
1.1.1.1 - Fonction de Bernstein .....	14
1.1.1.2 - Propriétés des courbes de BEZIER .....	15
1.1.2 - Courbe de BEZIER Rationnelle .....	16
1.1.2.1 - Propriétés des courbes rationnelles de BEZIER .....	16
1.1.3 - Surface de BEZIER .....	16
1.1.4 - Surface Rationnelle de BEZIER .....	17
<b>1.2 - MODELE B-SPLINE</b> .....	17
1.2.1 - Courbe B-Spline.....	17
1.2.2 - Courbes B-Splines rationnelles .....	20
1.2.2.1- Propriétés des courbes NURBS.....	20
1.2.3 - Surface B-Spline .....	21
1.2.4 - Surface B-Spline Rationnelle .....	22
Conclusion.....	22

### *Chapitre 2*

#### **OPERATIONS GEOMETRIQUES APPLIQUEES SUR LA SURFACE FEMORALE**

<b>2-1 LA METHODE DE SUBDIVISION DU VOLUME FEMORALE</b> .....	24
<b>2-2 DEVELOPPEMENT ALGORITHMIQUE DE LA METHODE</b> .....	24
2-2-1 Algorithme de DE-BOOR: .....	26
2-2-2-1 Application de l'algorithme de DE-CASTELJAU .....	27
2-2-2-2 Schéma d'algorithme général .....	29
2-2-3 Description de l'application de cet algorithme .....	30
2-2-3-1 Conversion des surfaces en carreaux de BEZIER .....	30
<b>2-3 Création des boite Min-Max des carreaux de surfaces</b> .....	31
2-3-1 Définitions des boites .....	31

2-3-2	l'intérêt d'utilisations des boites Min-Max .....	31
2-3-3	Détection de l'intersection avec la surface fémorale .....	33
2-3-4	Calcul des boites dans le repère local au réseau .....	34
2-3-5	Position du plan de coupe arbitraire.....	35
2-3-6	Test de pénétration dans des boites situées dans le repère local .....	35
<b>2-4</b>	<b>Vérification de la planéité de chaque carreau pendant la subdivision.....</b>	<b>36</b>
2-4-1	Vérification en fonction des tangentes et des normales .....	36
2-4-2	Vérification de la planéité en fonction de la linéarité des bords et la planéité de réseau de contrôle .....	37
2-4-3	Test de planéité par les boites Min-Max dans le repère local .....	38
2-4-4	Conclusion et choix d'un test optimal .....	38
<b>2-5</b>	<b>Calcul des points d'intersections .....</b>	<b>38</b>
2-5-1	Principales Méthodes d'intersection .....	40
<b>2-5-2</b>	<b>Technique NON-LINEAIRE .....</b>	<b>40</b>
2-5-2-1	Présentation de la méthode .....	41
2-5-2-2	Calcul du prochain point approché à la courbe d'intersection .....	43
2-5-2-3	Affichage du point approché .....	45
2-5-3	L'équation implicite du plan .....	46
2-5-4	Autre méthode .....	47
<b>2-6</b>	<b>Ordonnancements des points après calculs des intersections .....</b>	<b>49</b>
a)-	Ordonnement des points a critère de continuité géométrique .....	49
b)-	Ordonnement des points a critères des paramètres croissants .....	50

### *Chapitre3*

#### **APPLICATION DES ELEMENTS FINIS A LA BIOMECHANIQUE**

<b>3-1</b>	<b>Introduction .....</b>	<b>52</b>
	Elaboration et programmation .....	53
<b>3-2</b>	<b>Algorithmes Programmés pour l'application des calculs.....</b>	<b>55</b>
3-2-1	Reconnaissance de formes des coupes transversale du fémur .....	55
3-2-2	Création du solide représente l'Os fémoral par définition spécifique ....	57
3-2-3	Régénération de surface fémorale avec précision.....	58
3-2-4	Ordonnements des segments de droite qui construisent les courbes.	64
3-2-5	Habillage des nouvelles courbes calculé.....	65
<b>3-3</b>	<b>Insertion d'une prothèse à partir d'une base de données .....</b>	<b>68</b>
<b>3-4</b>	<b>Insertion de l'ensemble Os prothèse dans le logiciel d'analyse sys .....</b>	<b>69</b>
	<b>Conclusion.....</b>	<b>71</b>

## **CONCLUSIONS**

c-1	Couplage des relations biomécanique avec le système de CAO définie .....	73
c-2	L'analyse structurée de la Méthode de préparation à la simulation .....	75
c-3	Simulation pré clinique sur l'environnement de calculs développé.....	77
	<b>Conclusions</b> .....	78

## **ANNEXES**

### **Annexe A1**

#### **Rappel Mathématique**

A11	La normale et le plan tangent a une surface .....	79
A12	Courbure et rayon de courbure des surfaces .....	79
A13	Equation paramétrique d'une droite .....	80
A14	Intersection droite plan .....	80
A15	Intersection plan-plan.....	81

<b>BIBLIOGRAPHIE</b> .....	83
----------------------------	----

# INTRODUCTION ET RECHERCHE BIBLIOGRAPHIQUE

## Les différents chapitres essentiels de la thèse

Introduction et recherche bibliographique.

**Chap1** : Définition des modèles de la CAO (NURBS ; BEZIER).

**Chap2** : Description de la méthode de subdivision d'un volume englobé par une surface  
modéliser par les modèles précédents.

**Chap3** : Application a la Biomécanique.

Conclusion générale

**Bibliographie.**

## خلاصة

نموذج رقمي من أجل تعريف المساحات المعقدة الشكل المعروفة باسم (NURBS)، تمت برمجته تحت AUTOLIS ، و يقوم هذا الأخير بتنظيم عملية الدراسة الطبية بالكمبيوتر. و الطريقة تتجلى في أخذ مجموعة نقاط التي يتم أخذها من الصورة النووية للعظام . و باستعمال هذه النقاط نقوم بإنشاء مساحة تمر على هذه الأخيرة، و منه نستخرج من هذه المساحة بما يعرف بشبكة التحكم في المساحة المعقدة للعظام و بعد ذلك نستعمل برنامج قمنا به عن طريق C++ و يقوم هذا البرنامج بتعريف و رسم كل المساحة بدقة و بكثافة على شكل منحنيات (NURBS) مغلقة و التي تنشأ بحساب تقاطع شبكة التحكم في المساحة و المستويات الأفقية التي تقطع المساحة.

و بإنهاء ما سبق من عمليات، نقوم في برنامجنا الإلكتروني بعمليات غرس افتراضي ل Prothèse في العظام تحت مراقبة معطيات الصنع ل Prothèse و بعد إجراء الاختيار الأولي نقوم بحساب القوى و الضغوطات و تحركات الجزيئات للمادة الداخلية و الخارجية للعظام و لمادة الـ Prothèse و حسب المعطيات يتم حساب مجالات القدرة الصحية للمريض و القدرة الحركية لـ Prothèse .

يكمن عملنا في هذا المجال من البحث العلمي في تحويل هندسة البرامج الإلكترونية للدراسات الطبية إلى الاهتمام أكثر في إنشاء الشكل الحقيقي المعقد للعظام في الأبعاد الثلاثة 3D للكمبيوتر.

## Abstract

A numeric model for modelling of surfaces (NURBS) has been programmed under AutoCAD environment and AUTOLISP both system, that makes management of the pre-clinic simulation (SPC). The study way consist to take several points from the transversals cuts made by the scanner in order to creates control points that use to calculate network of the control femoral surface upon application programmed under C++, this geometrical part should go to be introduced in the second program under C++ language that go to running a new transverse cuts that define a real surface of femoral body as well precision, we proceeded then by dressing of the surface and begin the testes of a virtual implantation of total prosthesis hip inside the femoral body, for optimisation of geometrical position of implant under anatomical axis of the thighbone, and to finalise the SPC we transfer the femoral whole toward a mechanical analysis software for assessment of the primary stability of micro-distortions and constraints has the interface of a both bodies. Our consists has topple this architecture of the SPC to an conception that converges more toward the exact geometry of the volumes as complex form, and in addition mainly the decomposition of the femoral body in a whole of subfields created and limited by the Iso-Density then bounded between them by embeddings, in order to better converge towards a nonlinear modelling and to be able to apply the laws of isotropic viscoplastic mechanics and to eliminate almost assumptions from linearity of the behaviour.

### **Résumé**

Un modèle numérique pour la modélisation des surfaces (NURBS), a été programmé sous environnement AUTOCAD en utilisant le langage Autolisp qui fait la gestion de la simulation pré –clinique (SPC). La méthode consiste à prendre des points à partir des coupes transversales produites par l'imagerie scanner, pour créer un nuage de points, à partir de lequel on procède au calcul sous une application programmé en C++ afin de calculer les points du réseau de control de la surface fémorale. Ce dernier va être introduit dans le deuxième programme C++ qui calcul les nouvelles coupes transversales qui définissent la surface réelle du corps fémoral avec plus de précision. On procède par suite à l'habillage de la surface, et on lance les essais d'implantations virtuelle de la prothèse totale de hanche à partir d'une base de données de produit fabricants. Pour l'optimisation géométrique de la position de l'implant sur l'axe anatomique du fémur, et pour finaliser la SPC, on procède à l'envoi de l'ensemble fémoral vers un logiciel d'analyse mécanique pour l'évaluation de la stabilité primaire vis à vis des microdéformations et des contraintes à l'interface des deux corps (Os –Prothèse). Notre exploit consiste à basculer l'architecture de conception qui converge plus vers les géométries exactes des volumes à surface complexes, et d'autre part principalement la décomposition du corps fémoral en un ensemble de sous-corps créent et limités par des Iso-Densité et liés entre eux par des encastremets, afin de mieux converger vers une modélisation non linéaire et pouvoir appliquer les lois de la mécanique viscoplastique isotrope et éliminer les hypothèses de linéarité du comportement.

### **Mots clé**

Modélisation des surfaces (NURBS) ; Simulation pré clinique (SPC) ; Biomécanique ; Réseau de contrôle ; Corps fémoral ; L'axe anatomique ; Stabilité primaire ; Microdéformations ; Contraintes ; BEZIER ; Os Fémoral ; Subdivision.

### 1 -INTRODUCTION GENERALE

Avec le développement de la technologie de la Conception Assistée par Ordinateur, connue sous le nom de la CAO, les méthodes numériques se sont imposées comme des compléments indispensables aux méthodes expérimentales pour l'analyse du comportement ou même du mouvement des corps et des solides déformables surtout lorsque :

- les formes géométriques de ces corps sont compliquées (englobé par des surfaces gauches qui peuvent être modéliser),
- les déformations qu'ils subissent sont aléatoirement réparties, et les matériaux qui les constituent ont un comportement non linéaire,
- les charges appliquées sont dynamiques et variables dans le temps.

La technologie permet de créer des architectures pour les études de comportements des volumes complexes sur les quelles la conception des logiciels d'analyses des systèmes se base. Des modèles mathématiques permettant de bien respecter ces formes et qui donnent une meilleure approche ont été misent on place. Parmi ces modèles on cite : Surface Réglée, Surface de COONS, Surface de BEZIER,.etc., et plus récemment les surfaces dites NURBS (Non Uniforme Rational B-SPLINE). Ces modèles se basent sur la représentation paramétrique. Une fois le modèle du volume est conçu à l'aide des logiciels de la CAO, on passera à l'étape de la discrétisation volumétrique en vue d'une étude du comportement mécanique et l'optimisation des positions des implants dans le fémur d'une part et d'autre part la définition de l'alliage optimale qui constitue la prothèse.

Notre travail porte sur une l'utilisation d'une nouvelle architecture pour gérer une étude pré clinique, depuis les coupes scanner et jusqu'à la définition rigoureuse du comportement de l'ensemble Os-Prothèse on prenant en considération tous les paramètres biomécaniques tel que le remodelage de l'os et la différence de structure anisotropique dans les différentes zone de l'os, ce travail ce fait par la création d'une interface programmé par « AUTOLISP » et qui gère plusieurs programme fait par le C++ pour la génération de la surface fémorale et l'ordonnancement des courbes en plus à l'interface avec les logiciels d'analyses « ANSYS ;PATRAN ;COSMOS ».

Toute la conception géométrique ce fait dans le programme crée, puis le modèle géométrique est renvoyé vers un de ces logiciels, pour le calcul des contraintes et les déformations en utilisant la méthode des éléments finie.

En définie dans ce qui suit le Concept **ESOP** fémoral (La prothèse fémorale sans ciment). La base de cette étude était de démarrer à partir des problèmes trouvé dans les implants standards. Il s'agissait essentiellement de complications dues à un manque de respect de la physiologie et de l'élasticité fémorale. Les mauvaises transmissions des contraintes et le manque d'adaptation entre fémur et prothèse conduisait à des fractures et des tassements immédiats ou précoces, à des remaniements osseux importants (ostéolyses ou hyperostoses) se traduisant par des résultats cliniques décevants dans un bon nombre de cas. La douleur de cuisse très invalidante était présente dans un pourcentage inquiétant des observations.

La prothèse anatomique devait être modulaire avec un élément diaphysaire et un élément métaphysaire, pour prendre en compte la diversité fémorale. Elle devait être asymétrique, droite ou gauche. La jonction entre les deux pièces présentait théoriquement une zone fragile, et elle devait être non ou peu sollicitée pour éviter tout risque de rupture. Une étude par la méthode des éléments finis de l'ensemble fémur prothèse à du montrée que cette jonction était très peu sollicitée. Toutes les contraintes devaient passer par la métaphyse haute pour ne pas solliciter la jonction, mais aussi pour transmettre les charges à la partie haute du fémur sans contraindre directement la diaphyse. Il fallait le moins possible rigidifier le fémur et donc, ne pas chercher l'adaptation corticale parfaite en zone métaphysaire moyenne et inférieure. La pièce diaphysaire devait être lisse, cylindrique, légèrement sous dimensionnée par rapport à l'os. Elle devait être courte pour ne pas risquer de blocage distal.

Par traitement d'images scanner d'un fémur humain, un modèle numérique 3D de type Éléments finis (N éléments H8 par exemple, et M nœuds) se construisent, ce type de construction c'est effectué en plusieurs étapes

- a) le décodage et la visualisation des coupes scanner, détection des contours par seuillage des niveaux de gris, construction des points, lignes et surfaces de contour, importation de ces paramètres géométriques dans un logiciel de pré et post traitement Patran 5 (MSC Corps),
- b) maillage 3D du modèle dans Patran 5 (MSC Corps), et affectation des propriétés mécaniques de chaque constituant aux divers éléments,

Les paramètres mécaniques des parties prothétiques étaient issus de données de la littérature, celles de l'os cortical et de l'os spongieux ont été tiré de précédents travaux du laboratoire. L'os spongieux se considère comme un matériau isotrope hétérogène, l'os cortical a été choisi orthotrope. Puis la définition de tous les niveaux de contraintes et les microdéformations dans chaque élément et le transfert de contrainte depuis l'implant vers l'os et en fin la comparaison des résultats avec les limites des caractéristiques mécaniques de l'os et de la prothèse.

En a développer une méthode qui régit le problème d'intersection plan/surface gauche et trouve les courbes d'intersection sur la surface de l'os fémoral afin de définir le volume réel du corps fémoral avec les étapes suivantes pour pouvoir discrétisé le volume en vue de suivre le comportement mécanique (Micro déformations et contraintes a l'interface entre L'os et la prothèse). On doit d'abord présenter les outils de travail pour la modélisation des surfaces pour pouvoir appliquer les méthodes d'intersection. Et comme dans tous les cas pour faire une conception, on a besoin toujours des techniques mathématiques pour régler les problèmes de continuité de surfaces qui apparaissent au cours de la conception. En a comme ordre d'étapes chronologique

- 1) faire le scanner de coupes transversales sur la partie supérieur des corps fémoraux j'us qu'au niveau de l'extrémité inférieur de la prothèse,
- 2) sauvegarde des points de contours des coupes d'après l'ordinateur relier au scanner ou faire introduire les contours et établir une reconnaissance tridimensionnelle des points constitue les contours comme des courbes Spline (NURBS) et la définition de ces courbes sera faite dans le chapitre 1,
- 3) le programme établis génère un ordonnancement des différentes courbes sur l'axe anatomique et prépare un habillage de volume englober par les courbes, l'algorithme et la méthode d'ordonnancement seront définis dans le chapitre 3,
- 4) définir le réseau de contrôle (ou les réseaux de contrôles d'après la complexité de la surface) du volume fémoral qui représente une surface NURBS qu'on peut la considère Bi-Cubique et de séquence nodale déterminer a partire de la complexité de la surface étant model mathématique définie dans le chapitre 1,
- 5) faire une subdivision (en note que la subdivision se fait par l'interférence de la surface avec les plans virtuels de niveaux, le pa entre les plans et adéquat a la précision voulue pour les éléments finies en prévision de calcules des amorçages micro-fissurale) du réseau de contrôle par les algorithmes de DE-BOOR pour les surfaces NURBS et celui de DE-CASTELJO pour les surfaces de BEZIER pour avoir

- les nouvelles courbes de surface avec une haute précision et tout ce travail et opérations seront expliquer dans le chapitre 2,
- 6) refaire la génération des éléments finis d'après les nouveaux résultats, chapitre 3,
  - 7) dans un logiciel d'analyse des systèmes comme (ANSYS ; CATIA ...), après avoir faire la discrétisation en éléments finies on fait une simulation Pré-clinique avec un montage d'un certain types de prothèses et lancer un calcul de contraintes et de mouvements des surfaces de contacts, chapitre 5,
  - 8) avec un petit programme Fait sur Matlab en fait un balayage des fichiers de donnée crée par le stockage externe du logiciel avec une valeur limite de contrainte pour définir le niveau de la concentration de contraintes, et d'après le balayage on sélectionne tous les éléments qui représentent des pics de contraintes avec des géométries standards de concentration c a d le rapport limite de contrainte géométrie,
  - 9) on suppose des amorçages de fissures locales et on pose les paramètres biomécaniques de propagations et l'étude biomécanique sera prise par le chapitre 5,
  - 10) Les solutions proposées seront sur les formes géométriques et les alliages des prothèses pour l'optimisation des micro-mouvements et l'influence de ces paramètres sera montrée dans le chapitre 4.

On note que toutes les étapes sont nécessaires, car la géométrie fémorale est propre à chaque personne.

# **CHAPITRE 1**

## **MODELE MATHEMATIQUE DE LA CAO**

## 1-1- NOTIONS DE BASE

La modélisation mathématique a été toujours la seule solution pour régler le problème de définition des surfaces et des formes complexe. La technologie a prévue des fonctions pour la modélisation des surfaces gauche pour la C.A.O (Conception assistée par ordinateur).

Les principaux modèles traités dans la C.A.O sont :

Modèle de BEZIER,

Modèle B-Spline,

Modèle de COONS.

### Réseau caractéristique

Les concepts de polygone et de réseau caractéristique facilitent les manipulations des courbes et des surfaces, ou polygone caractéristique est définie dans un repère  $R(O,x,y,z)$  par un ensemble d'ordonné de  $(m+1)$  points :

$S_0, \dots, S_m$  ces points forment les sommets du polygones caractéristique qui est noté  $PG_m$ . Les côtés du polygone sont définis par les vecteurs.

$$(P_{i+1} - P_i) ; \quad i \in ( 0, 1, \dots, (m-1) ).$$

Un réseau caractéristique et on peut dire un réseau de contrôle est une généralisation du polygone permettant de définir une surface. Un réseau est construit à partir de  $(n+1)$  séquences ordonnées de  $(m+1)$  points.

## 1.1 - MODELE DE BEZIER

En présente dans cette partie les courbes et les surfaces de Bezier.

### 1.1.1 - Courbe de BEZIER

Pour faire une conception de formes, l'idée est de mettre à la disposition des opérateurs, un outil permettant de numériser la courbe et cela a partir de quelques points. Une des premières approches a été développée sous forme d'un polynôme à coefficients vectoriels :

$${}^I \hat{S}(u) = {}^r \hat{a}_0 + u {}^r \hat{a}_1 + u^2 {}^r \hat{a}_2 + u^3 {}^r \hat{a}_3 \quad \text{Ou} \quad u \in [0, 1]$$

La courbe  $S$  doit passer par à l'origine  $u = 0$  et par l'extrémité  $u = 1$ . Pour cela  ${}^I \hat{a}_0 = S(0)$  est la position de cette origine, et  $a_n = {}^I \hat{S}(n)$  est la position de l'extrémité.

La direction des tangentes se calcule par la dérivée de la fonction  $S$ , le principe a été repris par P. BEZIER.

Il a pris la forme polynomiale précédente, et il a utilisé des fonctions de BERNSTEIN, avec laquelle on peut approcher au mieux la forme voulue, et on peut introduire à l'aide de ces fonctions des contraintes géométriques.

De ce qui précède on définit une courbe de BEZIER par l'expression paramétrique suivant :

$$\overset{\mathbf{r}}{S}(u) = \sum_{i=0}^m F_{im}(u) \overset{\mathbf{r}}{a}_i \quad (1.1)$$

Ou  $a_i = S_i - S_{i-1}$  : Segment du polygone de contrôle.

$F_i$  : Fonction de contrainte géométrique imposée par BEZIER.

$u \in [0,1]$

$m$  : degré de la fonction

#### 1.1.1.1 - Fonction de BERNSTEIN

On a les vecteurs,  $\overset{\mathbf{r}}{a}_i$  noté par les composantes  $\overset{\mathbf{r}}{S}_i$  donnant les sommets du polygone.

$$\overset{\mathbf{r}}{a}_i = \overset{\mathbf{r}}{S}_i - \overset{\mathbf{r}}{S}_{i-1} \quad \text{Ou} \quad \overset{\mathbf{r}}{S}_{-1} = \overset{\mathbf{r}}{0}$$

L'expression 2.2 devient :

$$\overset{\mathbf{r}}{S}(u) = \sum_{i=0}^m F_{im}(u) [\overset{\mathbf{r}}{S}_i - \overset{\mathbf{r}}{S}_{i-1}] \quad (1.2)$$

et on trouve :

$$\overset{\mathbf{r}}{S}(u) = \sum_{i=0}^m [F_i(u) - F_{i+1}(u)] \overset{\mathbf{r}}{S}_i$$

avec :

$$F_{im}(u) - F_{i+1m}(u) = C_i^m \cdot u^i (1-u)^{m-1} = B_{im}(u)$$

#### Remarque

Les fonctions  $B_{im}(t)$  sont les fonctions de **Bernstein**,

avec  $C_i^m$  : Coefficient du binôme de Newton.

$$C_i^m = \frac{m!}{(m-i)! i!} \quad (1.3)$$

Une courbe de BEZIER est définie comme étant une forme polynomiale paramétrique de degré  $m$ , les points de cette courbe sont calculés par :

$$S(u) = \sum_{i=0}^m S_i B_{im}(u) \quad u \in [0,1] \quad (1.4)$$

avec :  $S_i$  : sommets du polygone de contrôle.

$B_{im}$  : fonction de BERNSTEIN

$$B_{im} = \frac{m!}{(m-i)! i!} u^i (1-u)^{m-1}$$

$m$  : degré de la fonction

### 1.1.1.2 - Propriétés des courbes de BEZIER

Le nombre de sommets du polygone impose le degré  $m$  de la courbe.

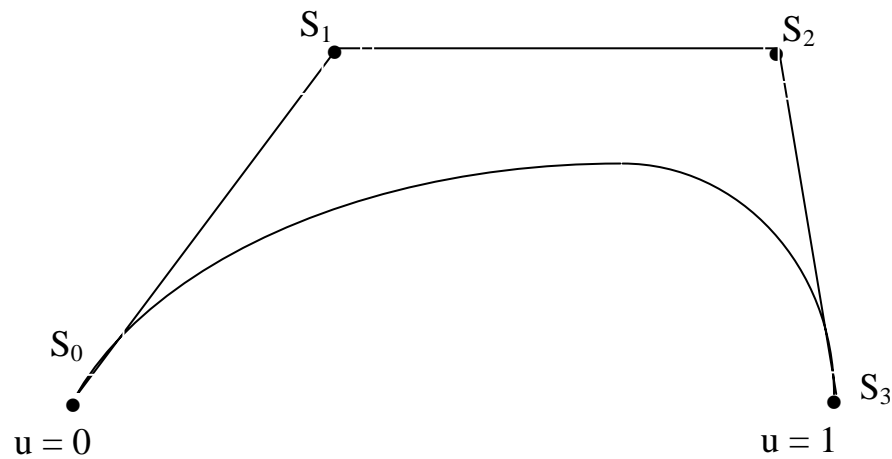
Une courbe de Bezier de degré  $m$  est construite à partir du polygone de contrôle (caractéristique)  $PG_m$  ne change pas de forme lorsqu'elle est décrite à partir du polygone caractéristique :

$S_m, S_{m-1}, \dots, S_0$  c-à-d interpolation réversible.

La courbe a pour origine  $S_0$  et pour extrémité  $S_m$  ( voir fig 2.1 ).

Le déplacement d'un point du polygone affecte toute la courbe exceptionellement les deux points extrêmes.

A toute courbe de BEZIER  $S(u)$  peut être associé un polygone convexe  $PC_m$  construit à partir du polygone caractéristique  $PG_m$  de  $S(u)$  voire l'exemple suivant.



**Fig 2.1** Courbe de Bezier - courbe degré 3 avec son polygone de contrôle [35]

### 1.1.2 - Courbe de BEZIER Rationnelle

La courbe de Bezier rationnelle a été utilisée pour son aptitude à représenter une classe plus importante de formes.

Pour imposer à la courbe une certaine déformation. Il faut introduire à l'expression de la courbe des coefficients qui fassent ce travail, et pour arriver à l'expression de la courbe déformée, il faut penser à l'analogie physique tel que les poids.

Les points de la courbe sont calculés par l'expression :

$$S(u) = \frac{\sum_{i=0}^m h_i s_i B_{im}(u)}{\sum_{j=0}^m h_j s_j B_{jm}(u)} \quad u \in [0, 1] \quad (1.5)$$

#### 1.1.2.1- Propriétés des courbes rationnelles de BEZIER

Elles sont différentes par rapport à celles de BEZIER dans la mesure où on note juste l'influence des poids sur la courbe et on suppose les centres des poids sont les sommets du polygone caractéristique.

- Influence de poids à valeurs voisines de l'infini : La coordonnée homogène  $h_i$  agit comme un poids, associé au sommet  $S_i$ , attire la courbe  $S$  vers ce sommet.
- Influence de coordonnées homogènes à valeurs voisines à zéro : la forme de la courbe évolue de façon à s'éloigner du sommet  $S_i$  associé à  $h_i$ .

### 1.1.3 - Surface de BEZIER

La surface de Bézier est définie comme le produit cartésien de courbes de Bézier

$$S(u,v) = \sum_{j=0}^m \sum_{i=0}^n P_{ij} B_{in}(u) \cdot B_{jm}(v) \quad u, v \in [0, 1] \quad (1.6)$$

La surface est de degré  $n \times m$ . Les sommets  $P_{ij}$  déterminent complètement une surface de Bézier et constituent le réseau caractéristique noté  $RS_{mn}$ .

Pour  $m = n = 3$ , nous avons la surface bi cubique de Bézier associée au réseau caractéristique  $RS_{3n}$ .

### 1.1.4 - Surface Rationnelle de BEZIER

L'élaboration de surfaces à partir de courbes de Bézier rationnelles relève de principes analogues aux étapes abordées aux surfaces de Bézier.

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n S_{ij} h_{ij} B_{jn}(v) B_{im}(u)}{\sum_{k=0}^m \sum_{l=0}^n h_{kl} B_{ln}(v) B_{km}(u)} \quad u, v \in [0,1] \text{ et } h_{ij} > 0 \quad (1.7)$$

## 1-2 Model B-SPLINE

En présente les courbes et les surfaces B-Spline afin de comprendre les NURBS.

### 1-2-1 Courbe B-Spline

Une courbe B-Spline est une forme polynomiale par morceaux de degré m, elle est formée à partir d'une succession de polynômes  $f_i(u)$  de degré m se raccordant entre eux à des abscisses  $u_i$ , tel que les polynômes  $f_i(u)$  ainsi que leurs dérivées d'ordre  $i, \dots, m-1$  sont partout continues et en particulier aux points de raccordement  $u_i$ .

$$0 \leq i \leq m$$

Ces points sont appelés nœuds, et l'ensemble de ces nœuds sera nommé séquence nodale. Une courbe B-Spline de degré m et de séquence nodale  $u_i, i \in \{0, \dots, (n+m+1)\}$  associé à un polygone caractéristique  $P_{gm}$  répond à l'équation : [ LEO 91]

$$P(u) = \sum_{i=0}^n S_i N_{im}(u) \quad n > m, \quad u \in [u_m, u_{n+m+1}] \quad (2.8)$$

Où  $N_{im}(u)$  sont des fonctions de base de degré m définie sur l'intervalle  $[u_{min}, u_{max}]$  par les formules de récurrence de D.Boor :

$$\begin{cases} N_{im}(u) = \frac{u - u_i}{u_{i+m} - u_i} N_{i,m-1}(u) & \text{si } u \leq u_{i+1} \\ N_{im}(u) = \frac{u - u_i}{u_{i+m} - u_i} N_{i,m-1}(u) + \frac{u_{i+m+1} - u}{u_{i+m+1} - u_{i+1}} N_{i+1,m-1}(u) & \text{si } u_{i+1} < u < u_{i+m} \\ N_{im}(u) = \frac{u_{i+m+1} - u}{u_{i+m+1} - u_{i+1}} N_{i+1,m-1}(u) & \text{si } u \geq u_{i+m} \end{cases} \quad (1.9)$$

avec :

$$N_{i,0}(u) = \begin{cases} 1 & \text{si } u_i \leq u \leq u_{i+1} \\ 0 & \text{partout ailleurs} \end{cases}$$



- Pour chaque multiplicité.

$L+n-1$

$\sum_{i=n-1} r_i = L + 1$  avec  $r_i$ : la multiplicité du nœud du domaine  $u_i$ .

$i=n-1$

Alors le calcul de la position d'un point  $s(u)$ , se déroule selon l'algorithme suivant :

On détermine l'intervalle des paramètres tel que :

$u \in ] u_i , u_{i+1} [$ .

Faire pour  $k = 1$  jusqu'à  $k = m$ .

Faire pour  $i = j$  jusqu'à  $i=(j-m+k)$ .

Calculer la position des  $(m-k+1)$  sommets pour la  $k_{ime}$  itération à l'aide de la relation.

$dk_i = [ (1-k_i) w_{i-1} k-1dk-1i-1 + k_i.w_i k-1.d_i k-1 ] .1/wk_i$

Fin faire.

Fin faire.

Le point converge vers la courbe.

Alors le calcul de la position d'un point  $S(u)$  se déroule selon l'algorithme suivant :

On détermine l'intervalle tel que  $u \in ] u_j, u_{j+1} [$ .

Faire pour  $k = 1$  jusqu'à  $k = m$ .

Faire pour  $i = j$  jusqu'à  $i = ( j - m + k )$ .

Calculer la position des  $( m+1-k )$  sommets

Pour la keme itération à l'aide de la relation

Fin faire

Fin faire

Le point converge vers la courbe d'intersection.

C'est l'algorithme qu'on utilise pour faire la conversion de NURBS vers BEZIER rationnel mais pour le choix des nœuds à insérer on choisi ceux de la séquence nodale car ces dernier représente les limites des nœuds de chaque traçons de BEZIER car une courbe b-spline est une succession de courbes de bezier avec un nombre = intervalle nodale.

### 1-2-1-1 Propriétés des courbes B-Spline

- Le degré  $m$  des fonctions de base  $N_{im}(u)$  n'est pas imposé par le nombre de sommets  $m+1$  du polygone caractéristique  $PG_m$ .
- L'intervalle de définition n'est pas restreint à  $[0, 1]$  comme pour les courbes de Bézier.
- Réversibilité d'une courbe B-Spline : la courbe ne change pas de forme lorsqu'elle est décrite à partir du polygone caractéristique  $P_m, P_{m-1}, \dots, P_1, P_0$ .
- Caractère local des courbes B-Spline : si l'un des sommets du polygone caractéristique est déplacé la déformation ne se fait que sur une partie de la courbe.

### 1-2-2 Courbes B-Splines rationnelles

Les formes rationnelles sont du même type que les modèles " traditionnels " (Bezier, B-Spline); Les seules différences notables sont la modélisation des quadriques et une certaine souplesse de définition de la forme. Le modèle NURBS (non uniforme rationnel B-Spline) est le modèle le plus utilisé en CFAO car il inclut les propriétés des modèles de Bezier, Bezier rationnels et des B-Splines. Les points de la courbe NURBS sont calculés par l'équation :

$$P(u) = \frac{\sum_{i=0}^n P_i h_i N_{im}(u)}{\sum_{i=0}^n h_j N_{jm}(u)} \quad (1.10)$$

### 1-2-2-1 Propriétés des courbes NURBS

- Influence de paramètre  $h$  sur le comportement d'une courbe B-Spline rationnelle : Elle est comparable à celle du modèle de Bezier rationnel, avec en plus, l'influence du comportement local des courbes B-Spline. Les NURBS sont la généralisation des modèles de Bezier, Bezier rationnel et B-Spline.

**Remarque :** On conclut et par analogie en tire les paramètres du modèle de surface B-spline.

### 1-2-3 Surface B-Spline

L'équation d'une surface B-Spline représente le lieu géométrique d'une courbe B-Spline lorsqu'elle subit simultanément un déplacement et une déformation. L'expression d'un point de la surface est :

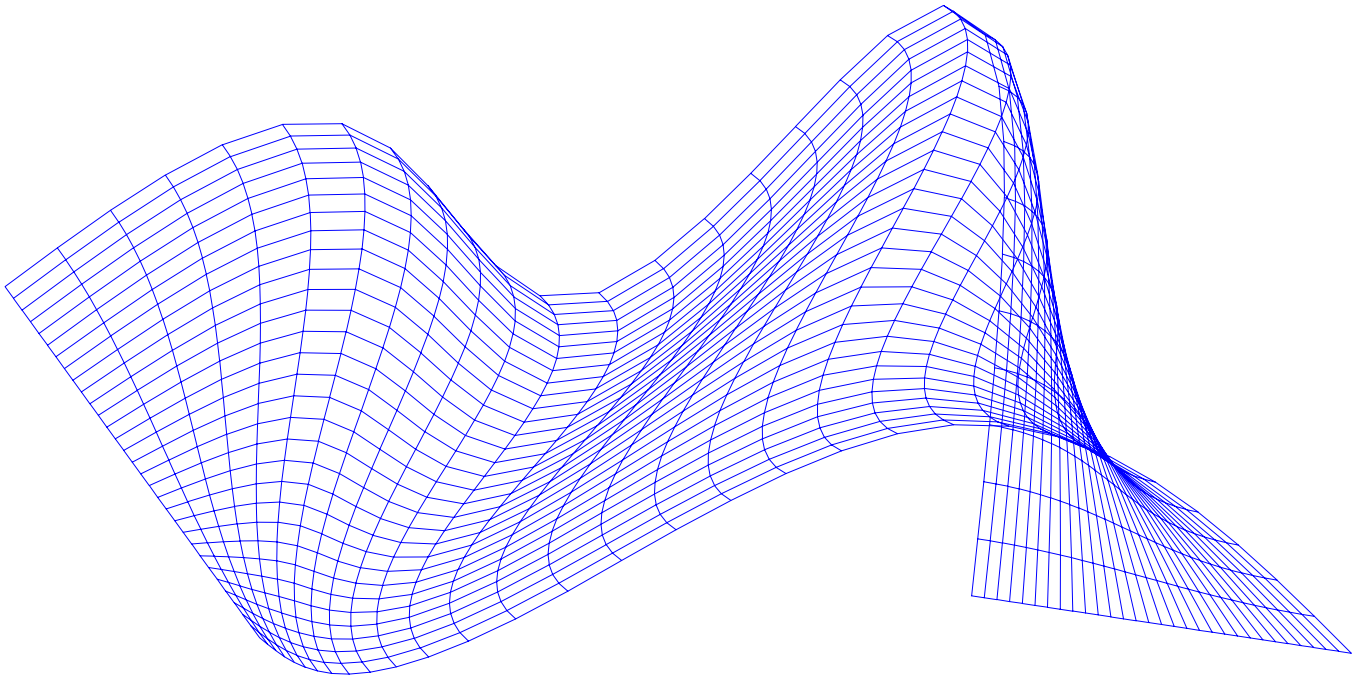
$$S(u,v) = \sum_{j=0}^q \sum_{i=0}^p S_{ij} N_{in}(u) N_{jm}(v); \quad (1.11)$$

Avec :  $u \in [u_0, u_{n+p+1}]$  et  $v \in [v_0, v_{m+q+1}]$

Les pôles  $S_{ij}$ ,  $j \in \{0, 1, \dots, m\}$ ;  $i \in \{0, 1, \dots, n\}$  constituent le réseau caractéristique de la surface;

avec :  $p+1$  : nombre de sommets du réseau suivant  $u$

$q+1$  : nombre de sommets du réseau suivant  $v$



**Fig. 1.2** : Exemple de Surface B-Spline  $m = n = 3$  [35]

#### 1-2-4 Surface B-Spline Rationnelle

Une surface B-Spline rationnelle est le lieu géométrique d'une courbe B-spline rationnelle à laquelle on impose un déplacement et une déformation.

Les séquences nodales  $u_i, i \in \{ 0, 1, \dots, (n+m+1) \}$ ,

$v_j, j \in \{ 0, 1, \dots, (t+r+1) \}$  et

le réseau caractéristique  $RS_{nr}^h$  définit complètement une surface N.U.R.B.S.

Les points de la surface sont calculés par :

$$P(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^t h_{ij} S_{ij} N_{im}(u) N_{jr}(v)}{\sum_{k=0}^n \sum_{l=0}^t h_{kl} N_{km} N_{lr}(v)} \quad \text{avec } h_{ij} > 0 \quad (1.12)$$

#### Conclusion

La détection des contours par seuillage des niveaux de gris induit des problèmes sur la précision géométrique dans les volumes fémorale, et puit la détection des contours n'est pas par modélisation directe en courbes Spline, ce qui leur induit le deuxième niveau d'erreur, et par la suite le maillage en éléments finis n'a pas de relation directe avec le model NURBS, ce qui résulte qu'ils auront le troisième niveau d'erreur dans le calcul, mais dans notre étude on assimile les segment de droites calculé à des constituons d'une courbe B-Spline qui donne une meilleur précision avec le lissage effectué et les l'ensembles des courbes NURBS qui sont on entrée de données sous forme de fichier SAT ou IGES, et ce pour le calcul des éléments finies dans ANSYS. Autrement dit ANSYS fait la discrétisation fine d'un volume en fonction du nombre de points en entrée, et dans le cas ou le nombre et minimise il fait l'interpolation linéaire c'est-à-dire le point qui situer entre deux extrémités de segment se trouve au milieu (c'est la différence dans le cas de faire une définition de volume par les forme NURBS, le point qui se trouve entre deux extrémité de segment de courbes se calcul après un lissage et une interpolation non linéaire. En note que l'affectation arbitraire des models de structures cristallographiques pour l'os cortical et l'os spongieux (isotrope et orthotrope respectivement) présente les même principes pour notre étude d'une part. Et d'autre part, leur absence de l'étude de comportement interactive vise à vis d'une base de donnée de différentes prothèses fémorales dans un même environnement, contrairement a notre cas, en a crée une base de données sous AUTOCAD qui permet l'insertion de tous type de prothèses dans l'environnement géométrique qui développe le volume fémoral.

## **CHAPITRE 2**

### **OPERATIONS GEOMETRIQUES APPLIQUEES SUR LA SURFACE FEMORALE**

## 2-1 La méthode de subdivision du volume fémorale

Ce chapitre consiste à définir l'astuce pour avoir une surface modélisée avec les modèles cités dans le chapitre précédent vers une discrétisation fine pour préparer le calcul des éléments finis. La méthode de subdivision est la méthode nécessaire pour arriver à une discrétisation surfacique, elle génère des courbes qui englobent toute la surface supérieure du fémur, les courbes d'intersection entre plans géométriques qui coupent la surface et la surface elle-même en joignant des points trouvés après un ordonnancement fait à partir d'un ensemble de données géométriques.

- On peut citer comme donnée le réseau caractéristique de la surface considérée (NURBS) surface non uniforme rationnelle B-SPLINE qui est le résultat de plusieurs étapes décrites dans l'introduction et que sont détaillées dans les chapitres suivants.
- par suite, le calcul des points d'intersection, il nous faut la position des plans anatomiques convertis en plans de coupes géométriques.

La méthode de subdivision de DE-BOOR, appliquée à un réseau caractéristique d'une surface B-SPLINE et compliquée à cause de l'insertion de nœuds qui est apportée à des calculs complexes, pour cela ce réseau sera d'abord converti en réseau de BEZIER afin d'utiliser une méthode plus simple dite, DE-CASTELJAU.

Dans le cas de la conversion l'algorithme de DE-BOOR consiste à faire une insertion de nœuds dans la séquence ou la suite nodale car ce modèle présente une suite polynomiale par morceau.

## 2-2 Développement algorithmique de la méthode

Afin d'évaluer une courbe de B-SPLINE de degré  $n$  donnée par les ordonnées de DE-BOOR et la suite des nœuds pour une valeur de  $u$  :

Insérer  $u$  dans la séquence nodale jusqu'à atteindre la multiplicité  $n$ . et en parallèle on calcule chaque fois les sommets des nouveaux polygones c.-à-d. le polygone résultat de chaque subdivision. Après  $n$  calculs de points et l'insertion du paramètre  $u$ ,  $m$  fois, le point qui a ce paramètre converge vers la courbe et sera une extrémité du nouveau polygone.

Si  $r$  était la multiplicité de  $u$  dans la séquence des nœuds avant le calcul, on aura  $(n-r)$  insertion de paramètre  $u$  dans la suite, et les points se calculent suivant la formule suivante.

$$d_i^k = [ (1-\alpha_i^k) w_{i-1}^{k-1} d_{i-1}^{k-1} + \alpha_i^k w_i^{k-1} d_i^{k-1} ] \cdot 1/w_i^k \quad (2.1)$$

Sachant que pour B-SPLINE rationnelle de degré  $n$

$$k \in \{1, \dots, n-r\} ;$$

$$i \in \{l-n+k+r, \dots, l+1\} ;$$

$$\alpha_i^k = (u - u_{i-1}) / (u_{i+n-k} - u_{i-1}) \quad (2.2)$$

$$w_i^k = (1 - \alpha_i^k) w_i^{k-1} + \alpha_i^{k-1} w_i^{k-1} \quad (2.3).$$

avec :  $u \in [u_l, u_{l+1}] \subset [u_{n-1}, u_{l+n-1}]$  conventionelle

$$d^0_i = d_i ; w^0_i = w_i ; r = 0 ;$$

$$s(u) = d_{L+1}^{n-r}(u) \quad (2.4)$$

### Remarque

L'algorithme n'est applicable qu'aux surfaces B-SPLINE. On considère le polygone caractéristique  $(d_0, d_1, d_2, d_3, d_4)$  d'une courbe B-SPLINE. Avant de passer aux études des surfaces on se trouve contraint d'étudier les courbes, car une surface et un ensemble de courbes ordonnées dans des directions données.

Pour avoir la surface B-SPLINE il faut que les courbes soient B-SPLINE.

- On suppose une courbe avec son polygone caractéristique (polygone de contrôle).

On avait au début une séquence des nœuds tel quel.

$[u_0, u_1, \dots, u_{n+p+1}]$  telle que :

$$u_0 = u_1 = \dots = u_n \quad \text{et} \quad u_{p+1} = u_{p+2} = \dots = u_{n+p+1}$$

Et après insertion de nœuds :

- Pour chaque multiplicité.

$L+n-1$

$$\sum_{i=n-1}^{L+n-1} r_i = L + 1 \quad \text{avec } r_i : \text{La multiplicité du nœud du domaine } u_i .$$

Alors le calcul de la position d'un point  $S(u)$ , se déroule selon l'algorithme suivant :

Afin de déterminer l'intervalle des paramètres tel quel

### 2-2-1 Algorithme de DE-BOOR

$u \in ] u_i, u_{i+1} [$ .

Faire pour  $k = 1$  jusqu'à  $k = m$ .

Faire pour  $i = j$  jusqu'à  $i = (j-m+k)$ .

Calculer la position des  $(m-k+1)$  sommets pour la  $k$  ième itération à l'aide de la relation.

$$d_i^k = [ (1-\alpha_i^k) w_{i-1}^{k-1} d_{i-1}^{k-1} + \alpha_i^k w_i^{k-1} d_i^{k-1} ] \cdot 1/w_i^k$$

Fin faire.

Fin faire.

Le point converge vers la courbe.

Alors le calcul de la position d'un point  $S(u)$  se déroule selon l'algorithme suivant :

On détermine l'intervalle tel que  $u \in ] u_j, u_{j+1} [$ .

Faire pour  $k = 1$  jusqu'à  $k = m$ .

Faire pour  $i = j$  jusqu'à  $i = (j - m + k)$

Calculer la position des  $(m+1-k)$  sommets

pour la  $k^{\text{ème}}$  itération à l'aide de la relation

$$d_i^k = d_i^{k-1} \left( \frac{u - u_i}{u_{i+m+1-k} - u_i} \right) + d_{i-1}^{k-1} \left( \frac{u_{i+m+1-k} - u}{u_{i+m+1-k} - u_i} \right) \quad (2.5)$$

Fin faire

Fin faire

Le point  $S_j^m$  converge vers la courbe d'intersection.

### 2-2-2 L'algorithme de DE-CASTELJAU

Cet algorithme est appliqué seulement aux surfaces de BEZIER [3].

A titre d'exemple, on suppose cette courbe avec son polygone de contrôle, la courbe est de degré 3, et le polygone de contrôle converge vers la courbe après la 3ème itération de calcul. La convergence sera sur le point  $d_0^3$ .

En démarre d'un polygone et après une subdivision de  $n$  fois on obtiendra deux sous réseaux ou polygones.

La formule qui calcule les sommets des nouveaux polygones est la suivante :

$$b_i^r = (1-t) \cdot b_i^{r-1} + t \cdot b_{i+1}^{r-1} \quad (2.6)$$

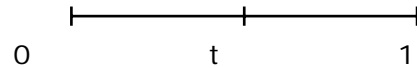
$r = 1, \dots, n$  ;

$i = 0, \dots, n-r$  ;

$b^0_i = b_i$  ;

Ce principe est plus simple que celui de DE-BOOR car on évite les études d'insertions de nœuds , mais le reste du calcul est le même principe que l'algorithme de DE-BOOR

L'intervalle paramétrique ;



Les découpages de l'intervalle paramétrique se fait au fur et à mesure avec la subdivision du polygone.

### 2-2-2-1 Application de l'algorithme de DE-CASTELJAU

Une surface gauche de BEZIER décrit la trajectoire d'une courbe qui se déplace dans l'espace en changeant de forme.

En conséquence, on arrive à :

$$S(u,v) = \sum_{i=0}^m \sum_{j=0}^n b_{i,j} B_i^m(u) B_j^n(v) \quad I=0 \quad j=0 \quad (2-7)$$

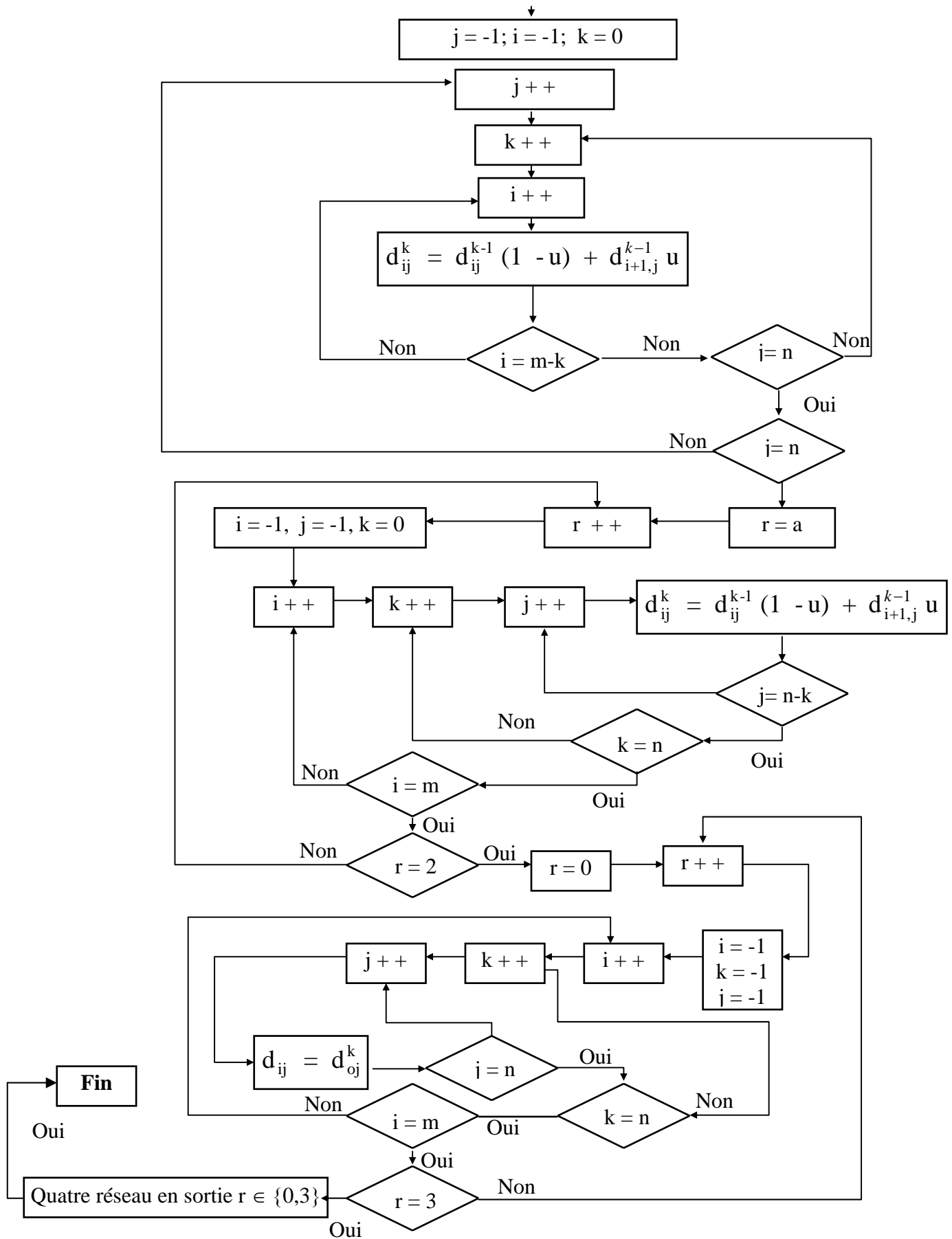
- Cette idée de la courbe nous a ramené à subdiviser une surface en nous basant sur le principe de subdivision d'une courbe.
- L'algorithme est similaire à celui décrit précédemment mais il tient compte des particularités suivantes des surfaces de BEZIER :

1. La subdivision double chaque fois le nombre de points de la ligne du polygone. Se procédé est extrêmement rapide surtout quand il s'agit de visualiser la surface. Cependant seuls les points extrêmes correspondent à des valeurs exactes des paramètres.
2. on ne fait pas appel à des évaluations de  $S(u,v)$  mais seulement un calcul itératif de moyenne sur les points du réseau.
3. la propriété d'enveloppe convexe de toute la surface permet d'envisager facilement le calcul de ce qu'on appelle boîte Min Max.

En montre sur la figure (3-1) suivante l'algorithme de subdivision.

(Fig 2-1) : Algorithme de De-Casteljau [35].

**Entrée du réseau caractéristique du BEZIER**



### 2.2.2.2 Schéma de l'algorithme général

Nous commençons à donner un schéma général de l'algorithme pour introduire les différents problèmes que nous allons rencontrer par la suite.

**Procédure** intersection surface – plan

**Début**

**Si** interpénétration ( s, plan ) **alors**

**Début si** planaire (s) **alors** intersection –( s, plan )

**Sinon si non**-planaire (s) **alors** subdiviser (s)  
( p a été subdivisé en  $s^{(1)}$  , ..... ,  $s^{(n)}$  )

**pour** i=1 à n **faire**

intersection ( plan-s)

**fin faire**

**fin si**

**fin**

**fin si**

**fin** intersection -plan – surface .

### 2-2-2-3 Schéma de l'algorithme des surfaces à pôles rationnelles

Surface de BEZIER rationnelle : c'est notre cas.

**Début**

**si** interpénétration ( SBEZIER , plan ) **alors**

**début**

**si** planaire (SB) **alors** intersection –( SB – plan )

**sinon**

**si non**-planaire (SB) **alors**

Subdiviser (SB) (s a été subdivisé en  $s^{(1)}$  , ..... ,  $s^{(n)}$  )

**faire pour** j = 0 **jusqu'à** n

**faire pour** k = 1 **jusqu'à** m

**faire pour** i = 1 **jusqu'à** (m-k)

$$h_{i,j}^k = h_{i,j}^{k-1} (1-u) + u h_{i+1,j}^{k-1}$$

$$d_{i,jk} = d_{i,j}^{k-1} (h_{i,j}^{k-1} / h_{i,j}^k ) (1-u) + d_{i+1,j}^{k-1} (h_{i+1,j}^{k-1} / h_{i+1,j}^k ) u$$

**fin faire**

**fin faire**

```

fin faire
faire pour l=0 jusqu'à m
faire pour k=1 jusqu'à n
faire pour j =0 jusqu'à (n-l)
 $d_{0,j}^{m,l} = ( h_{i,j}^{k-1} / h_{i,j}^k ) d_{0,j}^{m,l-1} (1-v) + v d_{0,j+1}^{m,l-1}$ 
 $( h_{i+1,j}^{k-1} / h_{i+1,j}^k )$ 
fin faire
fin faire
fin faire
fin si
faire ( calcul d'intersection surface de BEZIER – plan )
fin faire

```

**fin si**

**fin du traitement**

### 2-2-3 Description de l'application de cet algorithme

#### Conversion des surfaces en carreaux de BEZIER

Les surfaces du modèle sont convertit en carreau de BEZIER.

Pour les types de surfaces NURBS une conversion exacte par insertion de nœuds à l'aide de l'algorithme DE-BOOR est mise en œuvre.

Remarque :

On peut avant de commencer la méthode, entrer dans une autre procédure qui est la discrétisation de la surface.

La conversion du modèle commence par la conversion du format de données B-SPLINE vers la forme de BEZIER par morceau. peut être intéressante lorsque nous décidons de générer une surface B-SPLINE, la solution réside comme d'habitude pour les produits tensoriels, dans la composition du problème de surface en une série de problème de courbes, sinon on réécrit le modèle sous la forme de :

$$X ( u,v ) = \sum N_i^3(v) [ \sum d_{i,j} N_j^3 (u) ] \quad (2.8)$$

Nous voyons pour chaque indice i la somme entre crochet représente une courbe B-SPLINE de variable u, nous pouvons la convertir en courbe de BEZIER en utilisant les méthodes de conversion de courbes, ceci est équivalent à interpréter le réseau de

contrôle B-SPLINE, rangé par rangé comme un ensemble de polygone B-SPLINE, puis à les converties sous forme de BEZIER par morceau.

Les points de BEZIER ainsi obtenus peuvent être interprétés par des polygones B SPLINE, que nous pouvons encore transformer en forme de BEZIER pour affiner la subdivision, Cette famille finale de polygones de BEZIER constitue le réseau de contrôle de BEZIER de la surface.

## **2-3 Création des boîte Min-Max des carreaux de surfaces**

### **2-3-1 Définition des boîtes**

La boîte " Min – Max " est engendré par les six plans qui passent par:

- Le point qui a le Z min, situé sur la surface ou sur le réseau de contrôle et parallèle XOY.
- Le point qui a le Z max sur le réseau.
- Le point qui a le X min et le plan est vertical.
- Le point qui a le X max et le plan est vertical.
- Le point qui a le Y min et le plan est vertical.
- Le point qui a le Y max et le plan est vertical.

### **2-3-2 l'intérêt d'utilisation des boîtes MIN-MAX**

Généralement on trouve un problème pour détecter l'intersection entre deux surfaces gauche alors, la notion des boîte a résolu ce problème en vérifiant l'intersection entre les plans de la boîte du 1er carreau avec ceux de l'autre carreau car dans le cas de deux surfaces ce ne sert à rien de fixer le repère dans une position définie car la position des deux surfaces est quelconque.

Et pour le cas d'une intersection plan / surface et pour diminuer le temps de calcul des tests de planéité on peut faire un changement de repère et calculer la boîte Min-Max dans ce repère après calcul du 1er plan de la boîte qui passe par trois des quatre points extrêmes du réseau de contrôle et après ce calcul il nous suffira de tester l'épaisseur de boîte par rapport à la tolérance de planéité. et si on rencontre avec étude nous impose la position du repère qui n'est pas horizontal au plan du repère alors il nous faut un test d'interpénétration du plan dans la boîte.

Les directions  $u$  et  $v$  jouent un rôle pareil et peuvent donc être permutées sur la priorité de subdivision seulement. L'algorithme de DE-CASTELJAU appliqué successivement dans les directions  $u$  puis  $v$  comme l'indique son déroulement nécessite :

- $3(m^2n + mn + m^2 + n^2 + m + n)$  multiplication.
- $(3/2)(m^2n + m + n + m^2 + n^2 + m + n)$  addition.

Le nombre d'opérations dépend de l'ordre de traitement des directions  $u$  et  $v$ , on comprend l'intérêt de débiter le traitement suivant la direction dans laquelle le degré est plus faible.

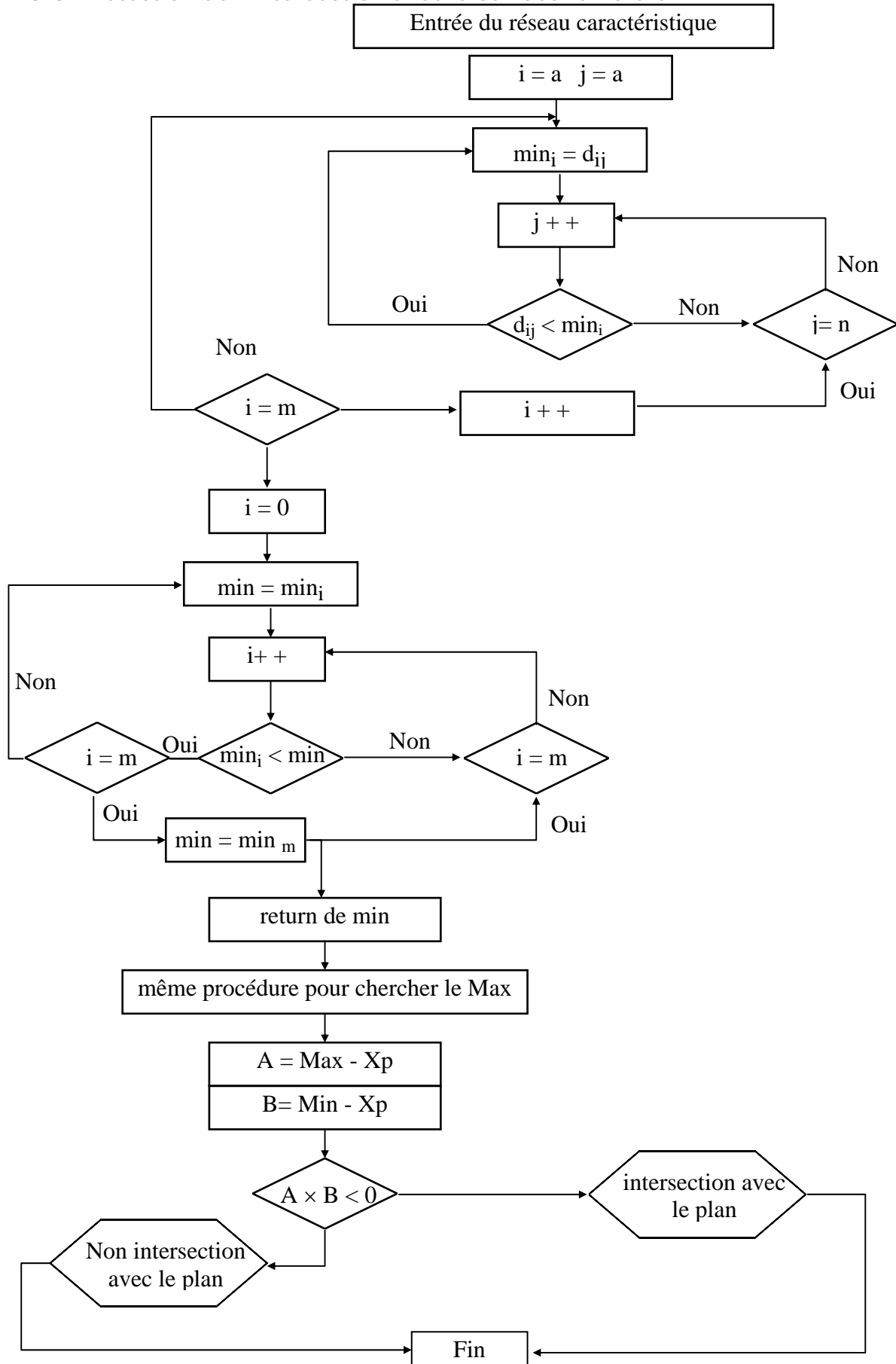
Pour simplifier les calculs on fixe le repère global de la surface tel que l'axe  $x$  ou  $y$  ou l'axe  $z$  soit normal au plan de coupe. Pour qu'on puisse tester la pénétration du plan à l'aide du  $X_{min}$  et  $X_{max}$  du réseau caractéristique.

Equation de plan du type.

$$\mathbf{X} = \mathbf{C} \quad \text{avec } C : \text{ est une constante.}$$

Dans le cas de l'intersection surface–surface on ne peut pas imposer le sens du repère global et son sens ne sert à rien, dans ce cas, il faut travailler avec des boîtes Min-Max pour traiter l'intersection.

2-3-3 Détection de l'intersection avec la surface fémorale



(Fig 2-2) : Test de l'intersection [35].

### 2-3-4 Calcul des boites dans le repère local au réseau

Généralement les boites sont hors des zones d'intersections, ce qui implique qu'il n'y a pas de calcul effectué sur ces boites par exemple, les frontières des sous réseaux caractéristiques provenant de l'opération subdivision effectuée à partir des procédures précédentes, elles sont utilisées pour générer des boites Min-Max.

- Le premier objectif est de s'assurer que les courbes limitent des sous carreaux appartiennent à la boite Min-Max, pour cela il faut utiliser des parallélépipèdes rectangles.
- Nous allons présenter la création et les tests concernant ces boites dans un repère local.

Une transformation locale de coordonnées est utilisée pour représenter les boites fermées.

Les coordonnées des points dans le repère global ou local sont données par les opérations suivantes :

$$P = T \times p \quad (2.9)$$

$$P = T^{-1} p \quad (2.10)$$

Où P et p respectivement les représentations globales et locales du même point.

- La transformation T est définie par les coordonnées locales des directions des axes k, l, m et de l'origine déplacée o'. Les vecteurs k, l et m représentent une matrice (3x3) tel que si ces vecteurs sont orthogonaux, le repère local est orthogonal, sinon il est oblique cela correspond respectivement à un parallélépipède rectangle ou quelconque. Les parallélépipèdes sont construits à partir des coins des réseaux de contrôle des surfaces et des tangentes correspondantes. Tel que la matrice T :

$$T = \begin{bmatrix} a & b & c \\ -b & \frac{1 - b^2}{1 + a} & \frac{-bc}{1 + a} \\ -c & \frac{-bc}{1 + a} & \frac{1 - c^2}{1 + a} \end{bmatrix}$$

**Avec** : a, b, c les Composantes du vecteur normal au plan passant par 3 des 4 coins du réseau.

Pour faciliter le découpage paramétrique pour la subdivision on met à la boîte une paramétrisation de ces plans à partir des limites des réseaux.

- On travaille dans l'espace de coordonnées cartésiennes.
- Les plans de la boîte sont parallèles au plan de l'espace  $(x,y)$  ,  $(x, z)$  et  $(y,z)$

### 2-3-5 Position du plan de coupe arbitraire

Dans ce cas, on vérifie l'intersection du plan de coupe avec les six plans de la boîte Min-Max.

Alors : Dans ce cas, la production est déterminée ou de rechercher l'intersection du plan de coupe avec les 6 plans de la boîte en se basant sur l'algorithme suivant :

*Faire plan de boîte de 1 jusqu'à 6*

*Poser le plan de coupe*

*Faire intersection plan-plan de coupe*

*Si intersection*

*Stockée le réseau pour le calcul .*

*Fin si*

*Sinon intersection*

*Faire sortie le réseau pour le calcul*

*Fin sinon*

*Fin faire*

*Fin faire*

### 2-3-6 Test de pénétration dans des boîtes situées dans le repère local

Le premier objectif consiste à créer des boîtes contenant les quatre coins de chaque carreau, ensuite, il faut agrandir ces boîtes de telle sorte que les frontières du carreau soient contenues à l'intérieur du parallélépipède.

Soient  $C_1$ ,  $C_2$ ,  $C_3$  et  $C_4$  les quatre coins du carreau.

$K$  et  $l$  sont donnés respectivement par  $(C_2 - C_1)$  et  $(C_3 - C_1)$ ,  $m$  est pris dans la direction de la normale à surface en  $c_1$  avec une norme imposée .

Il est à noter que dans cette construction le point  $c_4$  n'a pas été pris en compte et n'est donc pas obligatoirement à l'intérieur de la boîte.

Nous agrandissons donc la boîte en fonction des tangentes et des normales du carreau. la longueur de  $m$  sera égale à  $2 Hm$  qui est donnée par la relation suivante :

$$H_m = (L/2)(1-l)(1+k) \quad (2.11)$$

Où : L est le Max de  $c_3-c_1$  et  $c_4-c_2$

et :  $H_m$  est la valeur Max de l'angle entre les normales ou les tangentes au coin du carreau. Les longueurs des vecteurs l et k sont calculées similairement.

Pour tester si la boîte et le plan est en conflit, pour cela nous nous plaçons dans le repère d'une boîte (repère local), soit R le repère global,  $p_1$  les coordonnées du point p, dans le repère R1 donné par la transformation T1.

Nous avons donc : 
$$p_1 = T_1 \times p \quad (2.12)$$

Il faut noter que les coordonnées des coins du parallélépipède sont (0,0,0), ....., (1,1,1) dans le repère local. On a considéré la boîte comme un cube unitaire.

Les deux tests suivants effectués dans le repère local d'une des deux boîtes (Globale ou locale) détermine s'il y'a intersection.

#### Test n° 1

1. Il vérifie si tous les sommets des boîtes locales sont à l'extérieur du plan de coupe c à d  $p_1$  (plan).
2. Nous regardons si tous les X, tous les Y et tous les Z du plan à l'extérieur du X [0,1], y [0,1], z [0,1], dans ce cas il ni y'a pas d'intersection.

#### Test n° 2

il vérifie qu'un sommet de la boîte appartient au plan de coupe dans ce cas il y'a intersection .

### 2-4 Vérification de la planéité de chaque carreau pendant la subdivision

#### 2-4-1 Vérification en fonction des tangentes et des normales

On prévoit trois méthodes de vérification de la planéité on a pour la subdivision,

La structure de donnés de la subdivision est une structure arborescente et si les arbres sont non uniformes elle est dite adaptative.

Dans ces cas, la planéité et la linéarité des bords sont déterminées par les vecteurs tangents et les normales.

Il faut comparer :

- 1- Les vecteurs tangents unitaires des points formant les quatre coins du réseau dans la direction des courbes iso paramétriques.
- 2- Les vecteurs normaux unitaires adjacents des quatre coins.
- 3- Les vecteurs normaux unitaires des quatre coins avec la normale au centre du carreau.

Pour les vecteurs normés on aura comme suit.

$$(1-T_i) \cdot T_j < T_{lin} \quad (2-13)$$

$$(1-N_i) \cdot N_j < T_{pla} \quad (2-14)$$

Avec  $T_{lin}$  et  $T_{pla}$  étant respectivement les tolérances de linéarité et de planéité.

#### 2-4-2 Vérification de la planéité en fonction de la linéarité des bords et la planéité de réseau de contrôle

Comme on a dit ci dessus en cas de détection d'intersection on vérifie si la surface est plane à une tolérance voisine, sinon on continue la subdivision.

Pour satisfaire la linéarité il faut qu'on vérifie la relation :

$$d_k < \xi_l \quad (2-15)$$

Avec : donnée comme erreur permise de linéarité.

Les testes dans ce cas seront :

$$d_1(\delta_{i0}, \Delta_1) \leq \xi_l \quad (2-16)$$

$$d_2(\delta_{0j}, \Delta_2) \leq \xi_l \quad (2-17)$$

$$d_3(\delta_{i3}, \Delta_3) \leq \xi_l \quad (2-18)$$

$$d_4(\delta_{3j}, \Delta_4) \leq \xi_l \quad (2-19)$$

Si tous les quatre inéquations sont vérifiées, on passe au deuxième test qui est le test de planarité du réseau caractéristique, et si au moins une inéquation est non vérifiée alors tout le test de planéité est non vérifié, c à d qu'on continue la subdivision des sous réseau.

On a besoin des quatre droites,  $\Delta_1, \dots, \Delta_4$ , Pour le calcul des points d'intersection pour amener l'intersection plan-plan à une intersection plan-droites.

On fixe la valeur limite de l'erreur tolérée de planarité  $\xi_p$ , on choisi dans chaque réseau par défaut, les trois points  $S_{00}, S_3, S_{33}$ . Pour calculer un plan qui passe par eux

On vérifie la distance de chaque point du réseau a ce plan (P), il faut qu'on vérifie

$$D(S_{i,j}, P) \leq \xi_p \quad (2-20)$$

### 2-4-3 Test de planéité par les boites Min-Max dans le repère local

On peut vérifier la planéité d'une surface ou d'un réseau de contrôle par la méthode des valeurs Min-Max. Si on effectue un changement de repère tel que l'axe X sera normal au plan qui passe par trois points extrêmes du réseau caractéristique. Soit N la normale unitaire au plan de coordonnées (a,b,c) dans le repère global

Nous effectuons le changement de repère suivant :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} a & b & c \\ -b & \frac{1-b^2}{1+a} & \frac{-bc}{1+a} \\ -c & \frac{-bc}{1+a} & \frac{1-c^2}{1+a} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

avec  $a^2 + b^2 + c^2 = 1$

Après un changement de repère on va tester la planéité de la surface avec l'épaisseur de la boite Min-Max calculés dans le repère (X-Y-Z) M origine.

$$h < \text{Tolérance de planéité}$$

### 2-4-4 Conclusion et choix d'un test optimal

On a vu que le test des tangents et des normales n'est pas général car on trouve des surfaces où ça ne vérifie pas. On prend par exemple une surface qui a un pic au centre, dans ce cas de figure le critère de tangente n'est pas applicable. Dans ce cas on a toutes les tangentes et les normales parallèles entre elles. Et pour le cas de test à l'aide des boites Min-Max on a l'inconvénient de faire des calculs supplémentaire au dessus du calcul du test standard de linéarité et de planéité. Alors enfin on a décidé de choisir le test de linéarité-planéité car il est plus général.

### 2-5 Calcul des points d'intersections

Si le plan de coupe pénètre un carreau de surface, alors on a toujours le plan coupe exactement deux droites frontières, et ce critère ajoute de la précision dans nos calculs et on aura moins de calculs et ça nous évitera de calculer un quatrième point pour faire le plan de surface pour calculer l'intersection plan-plan et dans cette dernière on a un des deux points de segment d'intersection sera plus loin de la surface. C'est ce qui nous a permis de choisir l'autre méthode. On arrive au carreau supposé plan, On prend par exemple une surface de BEZIER bi cubique En commence par calculer les quatre droites (**Erreur ! Signet non défini.**) :

( $S_{00} S_{03}$ )( $S_{00} S_{30}$ )( $S_{03} S_{33}$ )( $S_{30} S_{33}$ ) :

Par l'équation d'une droite paramétrique dans l'espace.

$$\Delta_i : (\Delta_1) p_1(u) = (1-u) S_{00} + u S_{03} \quad (2.21)$$

$$(\Delta_2) p_2(u) = (1-u) S_{00} + u S_{30} \quad (2.22)$$

$$(\Delta_3) p_3(u) = (1-u) S_{03} + u S_{33} \quad (2.23)$$

$$(\Delta_4) p_4(u) = (1-u) S_{30} + u S_{33} \quad (2.24)$$

Et on a dans cet exemple, les plans de coupes, ou une position telle que l'axe x est normal à chaque plan. Cette méthode de calcul est la méthode optimale, avec un minimum temps de calcul. La conclusion finale et les avantages qu'on peut tirer sont que cette méthode est un remède à tous les problèmes au voisinage de point d'intersection dans les carreaux adjacents.

### Organigramme

L'organigramme qui calcule les points d'intersection est le suivant :

I=1

N=0

**Si** ( planéité du réseau )

**Faire** ( détection d'intersection avec le plan )

**Si** ( intersection )

**Tant que** (  $i < 4$  ) **faire**

**Tant que** (  $n < 2$  ) **faire**

Intersection ( $\Delta_i$  – plan de coupe )

$i=i+1$

**Si** (intersection )

$N=n+1$

**Fin si**

**Fin tant que**

**Fin tant que**

**Fin si**

**Fin faire**

**Fin si**

Avec : i est le nombre de droite frontière du carreau.

N est le nombre de points d'intersection calculés

Une fois le coté identifié, le calcul du point d'intersection est effectué et les valeurs des paramètres initiaux ( $u_1, v_1$ ) sont calculés en supposant une répartition uniforme des paramètres le long de l'iso paramétrique.

Si les deux tests sont vérifiés, alors on considère la surface comme ( S00, S03, S30, S33) et on calcule l'intersection plan-plan de coupe.

Alors on a optimiser la méthode de calcul d'intersection plan de coupe-quatre droites frontières, car elle nous donne plus de précision et moins de temps de calcul de test .

### 2-5-1 Principales Méthodes d'intersection

Vu l'importance de la recherche de l'intersection entre surface / surface ( dans notre cas surface / plan ) dans la CFAO et vu la difficulté et la complexité de résoudre le problème analytiquement beaucoup de méthodes numériques approchées ont été développées.

Actuellement, trois approches recouvrent la majorité des algorithmes couramment utilisés pour la recherche d'intersection.

- Technique non linéaire utilisant, en général une méthode numérique de type Newton-Raphson.

- Techniques algébriques faisant appel, pour l'une des entités, à une expression implicite et reposante ensuite sur la résolution d'une équation non linéaire.

- Techniques de subdivision utilisant les propriétés géométriques liantes une surface et son polygone caractéristique.

### 2-5-2 TECHNIQUE NON-LINEAIRE

Cette technique consiste à trouver un premier point de la courbe d'intersection entre le plan et la surface puis on parcourt toute la courbe en calculant à partir de chaque point trouvé le point qui le suit.

#### Présentation du problème

Soit une surface  $S_1(u,v)$  représentée sous forme paramétrique.

Soit un plan  $S_2(s,t)$  représenté paramétriquement et coupe la surface  $S_1(u,v)$ .

Pour trouver la courbe d'intersection on doit résoudre :

$$S_1(u,v) - S_2(s,t) = 0$$

$$(u,v) \in [u_{\min}, u_{\max}] \times [v_{\min}, v_{\max}]$$

$$(s,t) \in (\mathbb{R} \times \mathbb{R}) \tag{2.25}$$

On peut aussi écrire :

$$\begin{cases} X_1(u,v) - X_2(s,t) = 0 \\ Y_1(u,v) - Y_2(s,t) = 0 \\ Z_1(u,v) - Z_2(s,t) = 0 \end{cases} \tag{2.26}$$

Ou bien :

$$\begin{cases} X(u, v, s, t) = 0 \\ Y(u, v, s, t) = 0 \\ Z(u, v, s, t) = 0 \end{cases} \quad (2.27)$$

Où X, Y, Z sont des fonctions de u, v, s, t.

C'est un système de trois équations à quatre inconnues : u, v, s, t.

Dans le cas d'une surface NURBS qui est une surface polynomiale par morceaux on n'a pas une forme mathématique uniforme sur tout l'intervalle de définition, on doit subdiviser la surface en carreaux de BEZIER qui ont une forme polynomiale uniforme pour chaque carreau, puis trouver la courbe d'intersection avec le plan pour chaque carreau à part et enfin assembler les courbes d'intersection résultantes à l'aide d'un ordonnancement.

### 2-5-2-1 Présentation de la méthode

La résolution du problème revient à fixer un des paramètres et la résolution de la relation (3.26) qui devient un système de trois équations non linéaires à trois inconnues par une méthode numérique de type Newton-Raphson :

On fixe par exemple le paramètre "u", ( $u = u_i$ )

Soit le **développement de Taylor** concernant X, Y, Z :

$$\begin{aligned} X(v, s, t) &= X(v_r, s_r, t_r) + h \frac{\partial X(v_r, s_r, t_r)}{\partial t} + k \frac{\partial X(v_r, s_r, t_r)}{\partial s} \\ &\quad + l \frac{\partial X(v_r, s_r, t_r)}{\partial v} + \text{termes en } h^2, k^2, l^2, hk, hl, kl, \dots \\ Y(v, s, t) &= Y(v_r, s_r, t_r) + h \frac{\partial Y(v_r, s_r, t_r)}{\partial t} + k \frac{\partial Y(v_r, s_r, t_r)}{\partial s} \\ &\quad + l \frac{\partial Y(v_r, s_r, t_r)}{\partial v} + \text{termes en } h^2, k^2, l^2, hk, hl, kl, \dots \\ Z(v, s, t) &= Z(v_r, s_r, t_r) + h \frac{\partial Z(v_r, s_r, t_r)}{\partial t} + k \frac{\partial Z(v_r, s_r, t_r)}{\partial s} \\ &\quad + l \frac{\partial Z(v_r, s_r, t_r)}{\partial v} + \text{termes en } h^2, k^2, l^2, hk, hl, kl, \dots \end{aligned} \quad (2.28)$$

avec :

$$h = t - t_r$$

$$k = s - s_r$$

$$l = v - v_r$$

$t, s, v$  : supposées solutions exactes

$t_r, s_r, v_r$  : Supposées solutions approchées.

Nous négligeons les termes en  $h, k, l$  d'un degré supérieur à un et le système devient :

$$\begin{aligned} h \frac{\partial X(v_r, s_r, t_r)}{\partial t} + k \frac{\partial X(v_r, s_r, t_r)}{\partial s} + l \frac{\partial X(v_r, s_r, t_r)}{\partial v} &= -X(v_r, s_r, t_r) \\ h \frac{\partial Y(v_r, s_r, t_r)}{\partial t} + k \frac{\partial Y(v_r, s_r, t_r)}{\partial s} + l \frac{\partial Y(v_r, s_r, t_r)}{\partial v} &= -Y(v_r, s_r, t_r) \\ h \frac{\partial Z(v_r, s_r, t_r)}{\partial t} + k \frac{\partial Z(v_r, s_r, t_r)}{\partial s} + l \frac{\partial Z(v_r, s_r, t_r)}{\partial v} &= -Z(v_r, s_r, t_r) \end{aligned}$$

On peut l'écrire sous forme matricielle :

$$\begin{bmatrix} \frac{\partial X}{\partial t} & \frac{\partial X}{\partial s} & \frac{\partial X}{\partial v} \\ \frac{\partial Y}{\partial t} & \frac{\partial Y}{\partial s} & \frac{\partial Y}{\partial v} \\ \frac{\partial Z}{\partial t} & \frac{\partial Z}{\partial s} & \frac{\partial Z}{\partial v} \end{bmatrix} \begin{pmatrix} h \\ k \\ l \end{pmatrix} = - \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (2.29)$$

Notons  $J$  la matrice Jacobienne :

$$[J] = \begin{bmatrix} \frac{\partial X}{\partial t} & \frac{\partial X}{\partial s} & \frac{\partial X}{\partial v} \\ \frac{\partial Y}{\partial t} & \frac{\partial Y}{\partial s} & \frac{\partial Y}{\partial v} \\ \frac{\partial Z}{\partial t} & \frac{\partial Z}{\partial s} & \frac{\partial Z}{\partial v} \end{bmatrix}$$

En pré multipliant par la matrice inverse  $[J]^{-1}$  on obtient :

$$\begin{pmatrix} t - t_r \\ s - s_r \\ v - v_r \end{pmatrix} = - [J]^{-1} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (2.30)$$

ça revient à faire des itérations :

$$\begin{pmatrix} t_{n+1} \\ s_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} t_n \\ s_n \\ v_n \end{pmatrix} - [J_n]^{-1} \begin{pmatrix} X(t_n, s_n, v_n) \\ Y(t_n, s_n, v_n) \\ Z(t_n, s_n, v_n) \end{pmatrix} \quad (2.31)$$

Le premier point approché est difficile à trouver. On procède à la méthode de subdivision avec une tolérance de planéité plus large pour déterminer ce point.

On arrête les itérations quand :

$$|t_{n+1} - t_n| < \varepsilon$$

$$|s_{n+1} - s_n| < \varepsilon$$

$$|v_{n+1} - v_n| < \varepsilon$$

$\varepsilon$  Est déterminé suivant la tolérance voulue.

Pour simplifier les calculs, on peut procéder à un changement de repère tel que la normale au plan sera parallèle à l'axe des "X", et le critère d'arrêt des itérations sera:

$$|X(t_n, s_n, v_n) - X_{\text{Plan}}| < \varepsilon'$$

### 2-5-2-2 Calcul du prochain point approché à la courbe d'intersection

Soit  $P_0 = S_1(u_0, v_0) = S_2(s_0, t_0)$  un point appartenant à la courbe d'intersection cherchons un autre point  $P_1$  appartenant à la courbe d'intersection. Posons :

$$P_1' = P_0 + \delta V \quad (2.32)$$

avec :  $V$  est le vecteur tangent à la courbe d'intersection.

$P_1'$  : est une approximation de  $P_1$

$\delta$  : est la distance entre  $P_0$  et  $P_1'$

$V$  appartenant au plan tangent à la surface  $S_1$  donc :

$$V = a S_1^u(u_0, v_0) + b S_1^v(u_0, v_0).$$

$$\text{où : } S_1^u = \frac{\partial S_1}{\partial u}, \quad S_1^v = \frac{\partial S_1}{\partial v}$$

Soit :

$$V = a \begin{Bmatrix} X_1^u(u_0, v_0) \\ Y_1^u(u_0, v_0) \\ Z_1^u(u_0, v_0) \end{Bmatrix} + b \begin{Bmatrix} X_1^v(u_0, v_0) \\ Y_1^v(u_0, v_0) \\ Z_1^v(u_0, v_0) \end{Bmatrix} \quad (2.33)$$

Où :

$$X_1^u = \frac{\partial X_1}{\partial u}; \quad X_1^v = \frac{\partial X_1}{\partial v}$$

$$Y_1^u = \frac{\partial Y_1}{\partial u}; \quad Y_1^v = \frac{\partial Y_1}{\partial v}$$

$$Z_1^u = \frac{\partial Z_1}{\partial u}; \quad Z_1^v = \frac{\partial Z_1}{\partial v}$$

$a, b$  sont des réels inconnus à déterminer

$V$  appartenant au plan de coupe, puisque ce plan est parallèle à (YOZ) donc  $X$  un plan est constant. d'où :

$$a \times X_1^u(u_0, v_0) + b \times X_1^v(u_0, v_0) = 0 \quad (2.34)$$

En ajoutant une autre équation

$$\|V\| = 1 \quad \text{Donc}$$

$$\begin{aligned} & \left[ a X_1^u(u_0, v_0) + b X_1^v(u_0, v_0) \right]^2 + \left[ a Y_1^u(u_0, v_0) + b Y_1^v(u_0, v_0) \right]^2 \\ & + \left[ a Z_1^u(u_0, v_0) + b Z_1^v(u_0, v_0) \right]^2 = 1 \end{aligned} \quad (2.35)$$

On a deux équations à deux inconnues, donc on peut trouver les inconnues a et b.

Le parcours de la courbe d'intersection se fait en prenant l'incrément positif ou négatif de la direction V. La distance  $\delta$  est délicate à choisir. Si  $\delta$  est trop grand la courbe d'intersection n'est d'une part pas assez précis et d'autre part le calcul du point exact à partir du point approximatif risque de ne pas converger. Si  $\delta$  est trop petit, nous calculons plus de point, donc le temps de calcul sera très grand et les fichiers résultats seront énormes On a la relation suivante :

$$\delta = \rho_1 \cdot \theta$$

$\rho_1$  : est la grandeur du vecteur projection du vecteur rayon de courbure de la surface "p" au point  $P_0$  sur le plan "YOZ".

On a encore la relation :

$$\varepsilon_1 = \rho_1 \left( 1 - \cos\left(\frac{\theta}{2}\right) \right)$$

$\varepsilon_1$  : est une tolérance prise sur la courbe.

D'où on obtient :

$$\theta = 2 \text{ Arc cos} \left( 1 - \frac{\varepsilon_1}{\rho_1} \right)$$

On déduira ensuite  $\delta$  et le calcul de  $\rho$  se fait à partir des dérivées premières et secondes de l'équation de la surface.

### 2-5-2-3 Affichage du point approché

$P_1$  est un point de la surface  $S_1$  donc on peut écrire :

$$P_1 = S_1 ( u_0 + \Delta u , v_0 + \Delta v ) \quad (2.36)$$

Ou encore :

$$P_1 = S_1 (u_0 , v_0) + [J_1] \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} \quad (2.37)$$

$$\text{avec : } [J_1] = \begin{bmatrix} \frac{\partial X_1}{\partial u} & \frac{\partial X_1}{\partial v} \\ \frac{\partial Y_1}{\partial u} & \frac{\partial Y_1}{\partial v} \\ \frac{\partial Z_1}{\partial u} & \frac{\partial Z_1}{\partial v} \end{bmatrix}$$

La relation devient :

$$[J_1] \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = [P_1 - S_1(u_0 , v_0)] \quad (2.38)$$

$[J_1]$  Est une matrice  $3 \times 2$ . Cette matrice décrit le plan tangent à la surface  $S_1$  au point ayant pour paramètre  $(u_0, v_0)$ . Elle est parfaitement définie car la surface a au minimum une continuité  $C^1$ . ( Continuité de tangence)

La solution ne peut être trouvée qu'en inversant la matrice Jacobienne. Mais elle n'est pas une matrice carrée. Donc on doit utiliser la matrice pseudo-inverse  $[J^t \cdot J]^{-1}$ .

D'où :

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = [J_1^t \cdot J_1]^{-1} [J_1^t] [P_1 - S_1(u_0 , v_0)] \quad (2.39)$$

Cela revient à faire des itérations en prenant pour la première itération  $P_1 = P_1'$

La valeur trouvée de  $\begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}$  on l'injecte dans l'équation (2.38) on trouve une nouvelle valeur de  $P_1$  puis on l'injecte dans l'équation (2.39) et ainsi de suite.

Le critère d'arrêt des itérations est :

$$\|P_1 - P_0\| < \epsilon$$

Et puis, à partir du point  $P_1$  on calcule le point  $P_2$  de la courbe d'intersection et ainsi pour  $P_3$  jusqu'à parcourir toute la courbe. L'inconvénient de cette méthode est la difficulté de la détermination du premier point de la courbe ce qui oblige de passer par la subdivision de la surface pour le déterminer. Pour simplifier ce problème, on fait appel à une expression implicite de l'une des entités cela revient à la résolution d'une équation à deux inconnues.

### 2-5-3 L'équation implicite du plan

Soit le plan  $S_2(s,t)$  représenté sous forme paramétrique :

$$\begin{cases} x_2 = a_1 s + b_1 t + c_1 \\ y_2 = a_2 s + b_2 t + c_2 \\ z_2 = a_3 s + b_3 t + c_3 \end{cases}$$

Sous forme matricielle :

$$[A] \begin{Bmatrix} s \\ t \end{Bmatrix} + \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \end{Bmatrix} = \begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix}$$

$$\text{Où : } [A] = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}$$

La matrice  $[A]$  est une matrice  $3 \times 2$ .

Elle n'est pas inversible. On pré multiplie la relation vectoriel précédente par  $[A^t]$ , ( $[A^t]$  est la matrice transposée de  $[A]$  ).

$$[A^t] \cdot [A] \begin{Bmatrix} s \\ t \end{Bmatrix} = [A^t] \cdot \left( \begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix} - \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \end{Bmatrix} \right)$$

d'où on trouve  $s, t$  en fonction de  $x_2, y_2, z_2$  :

$$\begin{Bmatrix} s \\ t \end{Bmatrix} = [A^t \cdot A]^{-1} \cdot [A^t] \cdot \begin{pmatrix} x_2 - c_1 \\ y_2 - c_2 \\ z_2 - c_3 \end{pmatrix}$$

En remplaçant dans l'une des relations paramétriques précédentes on peut trouver l'équation implicite du plan :

$$a x + b y + c z + d = 0$$

où :  $a, b, c, d$  sont des coefficients réels constants.

Le problème d'intersection sera simplifié à une seule équation à deux inconnues:

$$a X_1(u, v) + b Y_1(u, v) + c Z_1(u, v) + d = 0$$

Soit :

$$F(u, v) = 0$$

On fixe un paramètre par exemple  $u = u_i$  et on résout l'équation non linéaire par rapport à  $v$  numériquement. On choisit la méthode de Newton-Raphson.

$$V_{n+1} = V_n - \frac{f(v_n)}{f'(v_n)} \quad (2.40)$$

On arrête les itérations quand :

$$|V_{n+1} - V_n| < \varepsilon$$

#### 2-5-4 Autre méthode

Le problème d'intersection plan / surface revient à résoudre le système :

$$S_1(u, v) - S_2(s, t) = 0 \quad (2.41)$$

##### 1er Cas

On fixe un des paramètres du plan S2 (s,t) par exemple "s" ( s= s<sub>i</sub>).

Le problème sera réduit à un problème d'intersection entre surface et droite.

P<sub>0</sub> : est le point d'intersection

S<sub>2</sub>(t) : Est le point approché du point d'intersection appartenant à la droite.

S<sub>1</sub>(u,v) : Est le point approché du point d'intersection appartenant à la surface.

r : est la différence entre S<sub>2</sub>(t) et S<sub>1</sub>(u,v).

$$r = S_2(t) - S_1(u, v)$$

On cherche à ramener S<sub>2</sub>(t) et S<sub>1</sub>(u,v) au point P<sub>0</sub> en faisant tendre r vers zéro.

En dérivant la relation on obtient :

$$dr = S_2^t dt - S_1^u du - S_1^v dv \quad (2.42)$$

S<sub>2</sub><sup>t</sup> : la première dérivée de S<sub>2</sub> par rapport à "t"

S<sub>1</sub><sup>u</sup> : la première dérivée de S<sub>1</sub> par rapport à "u"

S<sub>1</sub><sup>v</sup> : la première dérivée de S<sub>1</sub> par rapport à "v"

On isole chaque paramètre.

Par exemple on isole "v".

En faisant le produit vectoriel de chaque paramètre de l'équation (2.42) avec S<sub>1</sub><sup>u</sup>

$$S_1^u \wedge dr = (S_1^u \wedge S_2^t) \cdot dt - (S_1^u \wedge S_1^u) \cdot du - (S_1^u \wedge S_1^v) \cdot dv \quad (2.43)$$

Puisque (S<sub>1</sub><sup>u</sup> ∧ S<sub>1</sub><sup>u</sup>) = 0 On obtient :

$$S_1^u \wedge dr = (S_1^u \wedge S_2^t) \cdot dt - (S_1^u \wedge S_1^v) \cdot dv \quad (2.44)$$

On fait le produit scalaire de (3.32) avec S<sub>2</sub><sup>t</sup> :

$$S_2^t \cdot (S_1^u \wedge dr) = S_2^t \cdot (S_1^u \wedge S_2^t) \cdot dt - S_2^t \cdot (S_1^u \wedge S_1^v) \cdot dv \quad (2.45)$$

Or  $S_2^t$  est perpendiculaire à  $(S_1^u \wedge S_2^u)$  donc :  $S_2^t \cdot (S_1^u \wedge S_2^u) = 0$  alors :

$$S_2^t (S_1^u \wedge dr) = - S_2^t \cdot (S_1^u \wedge S_1^v) \cdot dv \quad (2.46)$$

En posant  $dr = r_{i+1} - r_i$  et  $dv = v_{i+1} - v_i$  et en mettant  $r_{i+1} = 0$

$$v_{i+1} = v_i + \frac{S_2^t \cdot (S_1^u \wedge r_i)}{S_2^t \cdot (S_1^u \wedge S_1^v)} \quad (2.47)$$

On fait les mêmes démarches pour trouver :

$$u_{i+1} = u_i + \frac{S_2^t \cdot (S_1^v \wedge r_i)}{S_2^t \cdot (S_1^v \wedge S_1^u)} \quad (2.48)$$

et

$$t_{i+1} = t_i - \frac{S_1^t \cdot (S_1^v \wedge r_i)}{S_1^u \cdot (S_1^v \wedge S_2^t)} \quad (2.49)$$

Ca revient à faire un calcul itératif.

Pour déterminer l'ensemble des points de la courbe d'intersection on fait la même procédure en faisant varier le paramètre "s" d'une valeur  $\Delta s$  qui est déterminée à partir de la tolérance sur la courbe :

### 2eme Cas

On fixe un des paramètres de la surface  $S_1(u,v)$  .

Par exemple "u" (  $u = u_i$  )

Le problème sera réduit à un problème d'intersection entre plan et courbe.

$k$  est le vecteur qui a comme origine le point "o" et comme extrémité un point du plan  $S$  et parallèle à la normale "n" au plan .  $k \wedge n = 0$ .

$S$  est le point d'intersection donc  $S$  appartient à la courbe  $S_1(v)$  et  $S$  appartient au plan  $S_2$ . Donc  $(S_2(v) - k)$  est un vecteur qui appartient au plan  $S_2$ , alors il est perpendiculaire à  $k$  :

$$(S_2(v) - k) \cdot k = 0 \quad (2.50)$$

$k$  étant un vecteur qui a une valeur constante, donc on a à résoudre numériquement une équation non linéaire à une seule inconnue "v".

Pour déterminer les autres points de la courbe d'intersection on fait la même procédure en faisant varier le paramètre  $u$  avec un pas " $\Delta u$ " qui est déterminé à partir de la tolérance sur la courbe.

## 2-6 Ordonnancements des points après calculs des intersections

### a)- Ordonnement des points a critère de continuité géométrique

L'ordonnement se fait sur la base que tous les points ou segments calculés on la relation de continuité et d'après la méthode de calcul que les points se répètent, alors on cherche à faire l'ordonnement des segments en commençant par le premier segment calculé par le compilateur.

L'algorithme appliqué pour le calcul, est le suivant :

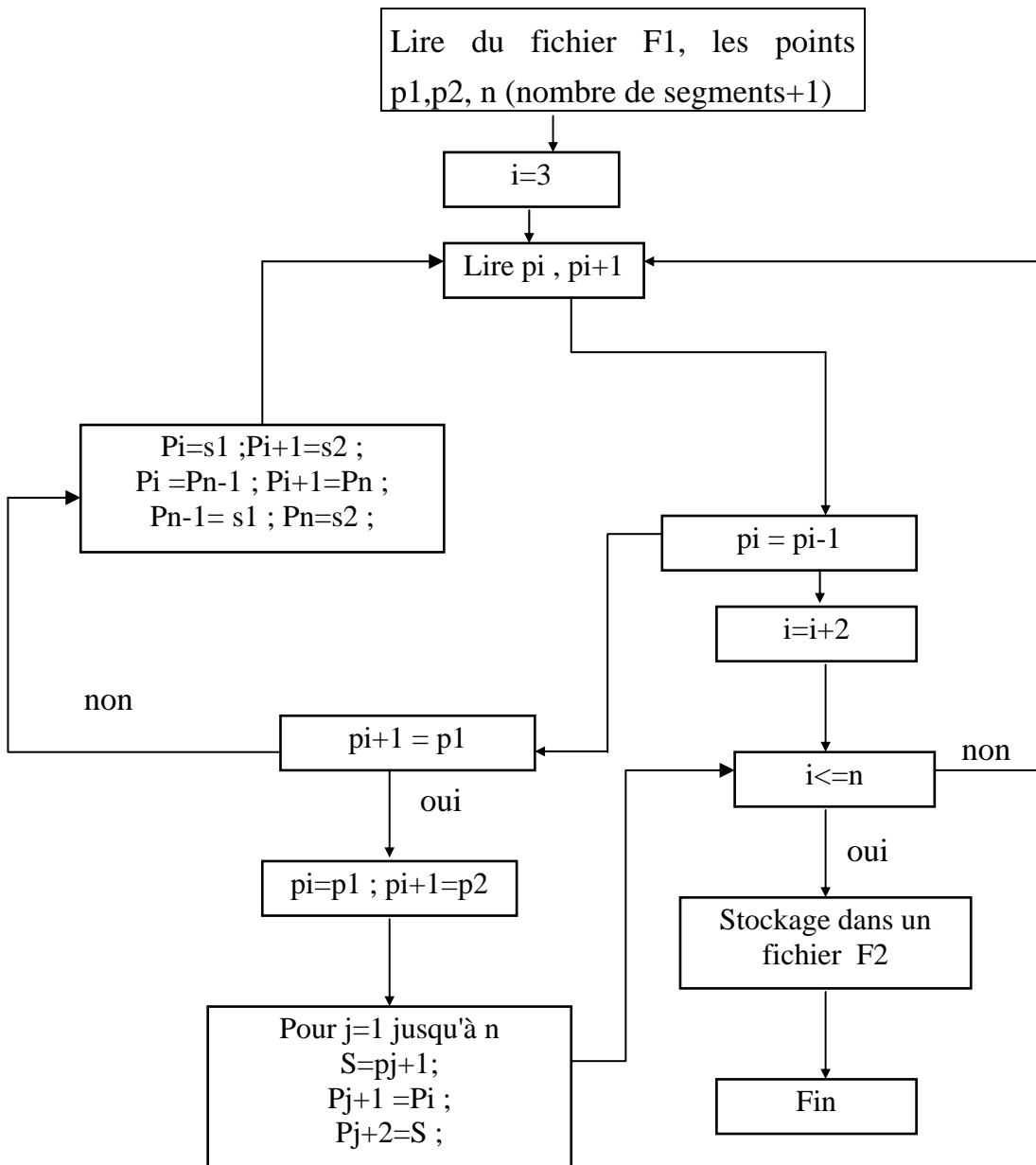


Fig (2-3) : Ordonnement de points des courbes transversales [35].

**b)- Ordonnement des points a critères des paramètres croissants**

Après avoir stocké tous les paramètres des points des segments d'intersection deux par deux on fait appel à un algorithme qui va faire le tri et relié les points pour avoir des courbes d'intersection.

Il commence à prendre les deux premiers points, il compare leurs paramètres aux paramètres des autres points, lorsqu'on aura les paramètres les plus proches aux paramètres de l'un des deux points alors on relie ces derniers points aux autres sur les points comparés identiques.

D'après la première opération on fait une détection de direction ensuite on continue la même opération pour comparer les paramètres extrêmes de la direction détectée précédemment.

Le programme s'arrête jusqu'à l'opération ou on ne trouvera pas de paramètre identique c.-à-d. : la courbe est terminée.

Autrement dit dans le cas des surfaces de BEZIER si on relie les deux points ou un de ces deux points à un paramètre égale à " 1 " alors la courbe d'intersection est terminée.

Le stockage des paramètres des points se fera tel que les points d'intersection de chaque plan de coupes stocké suivant une liste chaînée différente des autres pour qu'on passe d'une courbe à une autre facilement.

Le programme d'ordonnement termine son calcul jusqu'au dernier plan de coupes.

D'après tous les paramètres d'ordonnement en à adapté la méthode d'ordonnement à critère des continuités géométrique.

## **CHAPITRE 3**

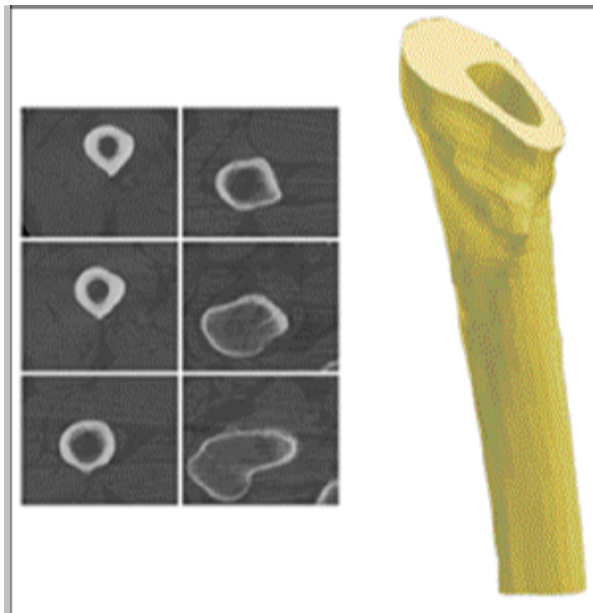
### **APPLICATION A LA BIOMECHANIQUE**

### 3-1 Introduction

La première étape est l'acquisition de l'information géométrique.

L'imagerie scanner nous fournit cette information sous forme d'un ensemble de coupes transversales. Après extraction par un logiciel de traitement d'image approprié des géométries fémorales interne et externe, le solide est reconstruit en 3-D avec le maillage correspondant.

On ajoute que le scanner produit des coupes avec un pas minimum qui n'est pas suffisant a la précision adéquate a la complexité de quelques zones dans l'os, pour cela on a reconstruit un logiciel qui se base sur le models de la CAO pour générer la surface externe et interne du fémur on plaçant des plans virtuels transversaux avec le pas qu'on choisi et qu'on juge suffisant. On présente dans le chapitre suivant toute la méthode de modélisation



**Fig.3.1:** Reconstruction tridimensionnelle [18]

Après avoir défini tout le réseau d'éléments cubique ou triangulaire constituant le volume, on va prévoir par une méthode géométrique qui va être définie après la répartition d'une charge (forces) Sur la partie extérieure de la surface de contact et on optimisant sa fonction de variation de temps, pour suivre le comportement mécanique qui va être conditionner par la variation géométrique de la surface de contact qui est définie par le model NURBS (Non Uniforme Rationnel B-Spline) et la variation géométrique de la surface va être calculer mathématiquement.

De cette étude on va optimiser la conception de l'implant fémoral pour maintenir la durée de vie de la stabilité du montage os-prothèse le plus longtemps possible.

Chaque tranche fémorale doit être parcourue dans une image informatisée qui traite le système. Sur ces images, les principales directions de raideur d'os cortical et spongieux ont été déterminées par l'introduction manuelle des directions anatomiques pour l'anisotropie.

Finalement, ces données géométriques définissent le modèle du fémur à trois dimensions avec la répartition de densité.

## Elaboration de méthode et programmation

Le modèle numérique pour la modélisation des surfaces (NURBS), a été programmé sous environnement *AUTOCAD* avec le langage *AUTOLISP* et qui fait la gestion de la simulation pré-clinique (**SPC**).

Prendre des points a partir des coupes transversales scanner pour crée le nuage de points a partir de lequel on lance le calcul des points du réseau de contrôle de la surface fémorale par une application programmé par le c++, ce dernier sera introduit dans le deuxième programme fait par le c++ qui calcul les nouvelles coupes transversales qui définissent la surface réelle du corps fémoral avec précision.

On procède par suite à l'habillage de la surface par la jonction des courbes coupes par des surfaces réglées.

Des essais d'implantation virtuelle de la prothèse totale de hanche a l'intérieur du corps fémoral en question sont nécessaire a l'optimisation géométrique de la position de l'implant sur l'axe anatomique du fémur, et pour finaliser la SPC en procède a l'envoi de l'ensemble fémoral vers un logiciel d'analyse mécanique pour l'Evaluation de la stabilité primaire vis a vis des micro-déformations et des contraintes à l'interface des deux corps (Os-Prothèse).

D'après nos observation des systèmes de calculs en éléments finis, et en matière d'analyse de contraintes et de déformation, y compris la fatigue et la fissuration, on à constaté que ANSYS est le système optimal pour nous permettre de bien gérer nos codes de calculs pour arrivé à l'optimisation de la prothèse.

Dans tous ce travail, le nôtre consiste à basculer l'architecture des logiciels de la SPC vers une architecture de conception qui converge plus vers les géométries exactes des volumes a surface complexes.

Par ailleurs en à développer une méthode qui régit le problème **d'intersection plan / surface gauche** et trouve les courbes d'intersection sur la surface de l'os fémoral avec les étapes suivantes pour pouvoir discrétisé le volume en vue de suivre le comportement mécanique (*Micro déformations et contraintes à l'interface entre L'os et la prothèse*) :

On doit d'abord présenter les outils de travail pour la modélisation des surfaces pour pouvoir appliquer les méthodes d'intersection.

Comme dans tous les cas, pour faire une conception, on a besoin toujours des techniques mathématiques pour régler les problèmes de continuité de surfaces qui apparaissent au cours des conceptions.

- 1- Faire un scanner de coupes transversales sur la partie supérieur des corps fémoraux j'us qu'au niveau de l'extrémité inférieur de la prothèse.
- 2- Sauvegarde des points de contours des coupes à partir de l'ordinateur relier au scanner, ou faire introduire les contours et établir une reconnaissance tridimensionnelle des points constituons les contours comme des courbes **Spline (NURBS)** cette dernière opération dépend du type du scanner.
- 3- Le programme établis génère un ordonnancement des différentes courbes sur **l'axe anatomique** et fait un habillage de volume englober par les courbes.
- 4- Définir le **réseau de contrôle (ou les réseaux de contrôles d'après la complexité de la surface)** du volume fémoral qui représente une surface **NURBS** qu'on peut la considère **Bi Cubique** et de **séquence nodale** déterminer à partir de la complexité de la surface.
- 5- Convertir le model NURBS vers un réseau de models de BEZIER en utilisant l'algorithme de de-boor par l'insertion de la séquence nodale.
- 6- Faire une **subdivision**, (*en note que la subdivision se fait par l'interférence de la surface avec des plans virtuels de niveaux*).
- 7- Refaire la génération des éléments finis d'après les nouveaux résultats en vue des calculs des **micro-mouvements**.

**Remarque**

A cette étape on prépare le terrain pour la recherche dans les axes qui concernent les modélisations des micro-fissures dans les milieux de complexités géométriques.

- 8- Dans un logiciel d'analyse des systèmes comme (**ANSYS ; FEMAP et CATIA**), après avoir faire la discrétisation en éléments finies on fait une **simulation Pré clinique** avec un montage d'un ensemble de types de prothèses et lance un calcul de contraintes et de mouvements des surfaces de contacts.
- 9- Pour l'optimisation du montage Os-Prothèse la modification sera faite aux niveaux des paramètres géométriques en un premier lieu en suite en fait varier les caractéristiques mécaniques de la prothèse.
- 10-Faire un balayage sur la base de donnée du logiciel à une valeur limite de contrainte pour définir le niveau de la concentration de contraintes, et d'après le balayage on sélectionne tous les éléments tétraédriques qui représentent des **pics de contraintes** avec des **géométries standards de concentration** c a d le rapport limite de **contrainte géométrie**.
- 11-Calculs et **comparaisons des paramètres biomécaniques à celles calculés par ANSYS**.
- 12-On suppose des **amorçages de fissures** locales et on pose les paramètres biomécaniques de **propagations**.
- 13-Les solutions proposer seront sur les formes géométriques, et les alliages des prothèses pour **l'optimisation** des micro-mouvements.
  - On montre par exemple les différentes opérations numériques effectuer sur le volume osseux du fémur dans la figure suivante avec le calcul des contraintes de **Van Mises**.

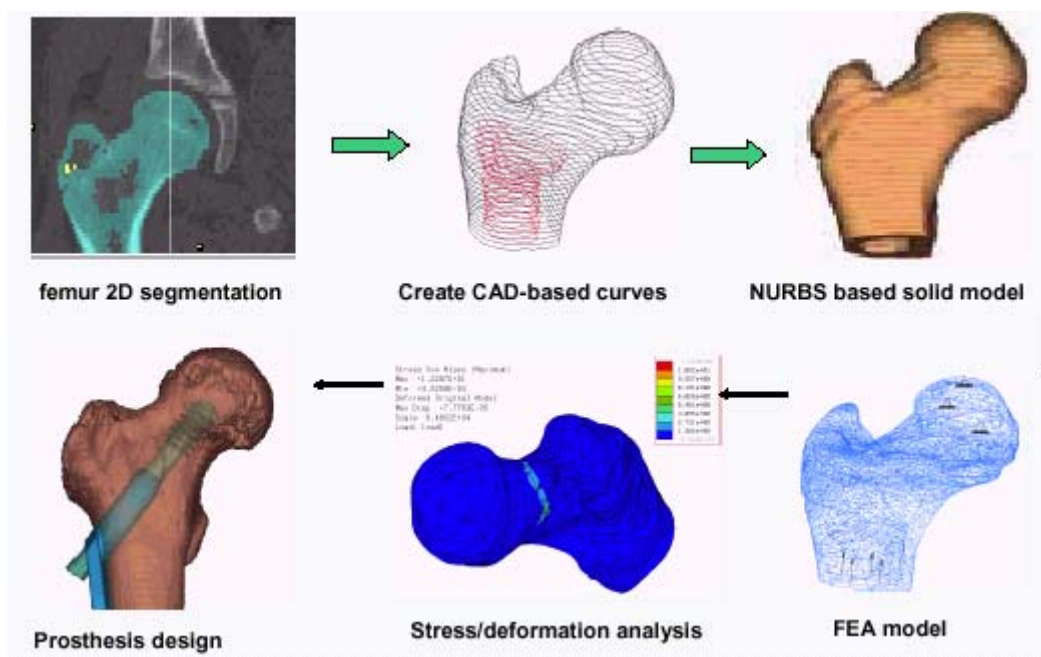


Fig. 3.2 : Opérations numériques [26]

### 3-2 Algorithmes Programmés pour l'application des calculs

#### 3-2-1 Reconnaissance de formes des coupes transversale du fémur

Dans le cas ou le scanner donne des fichiers numériques montrant l'organisation géométrique des coupent transversale du fémur, on fait le chargement des valeurs et on procède directement au calcul du réseau du contrôle. Le problème se montre dans le cas ou le scanner donne seulement des images. On a prévue une application dans notre programme d'AUTOLISP qui fait la reconnaissance de forme et convertie les images de coupes en courbes B-Spline dans le graphisme d'AUTOCAD. Pour plus de détails, en à pris un algorithme de reconnaissance de forme qui existe déjà, et on intégrant cette fonction sous environnement AUTOCAD, en à refait la construction des courbes en prenant les données géométriques des segments reconnus depuis le fichier DXF.

Le schéma suivant montre les étapes à suivre pour la reconnaissance de formes.

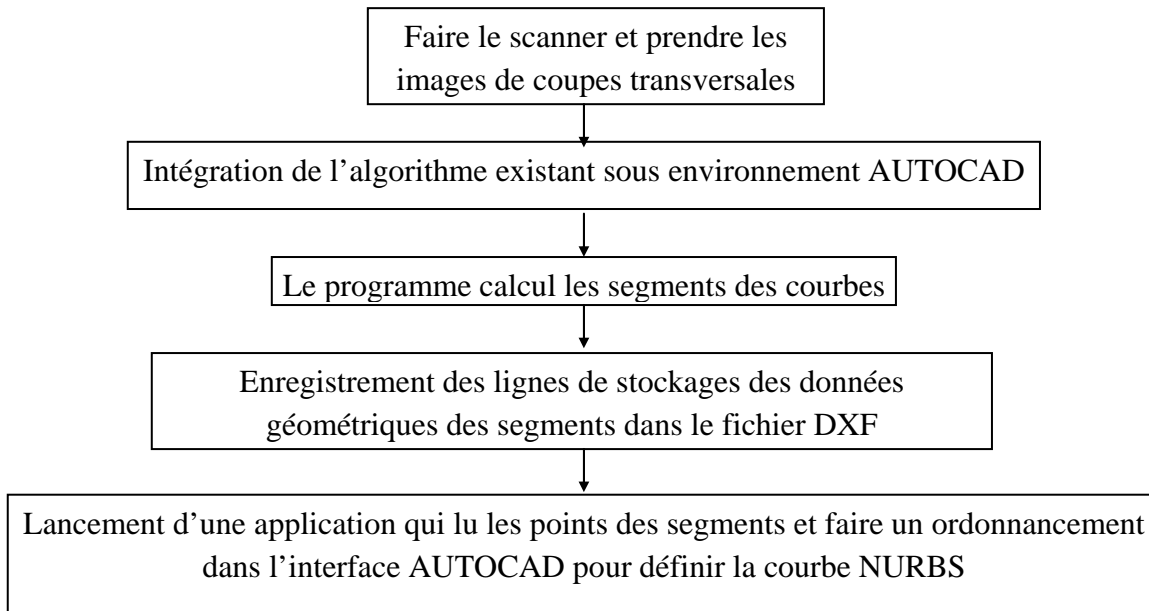


Fig. 3.3 : Opérations effectuer pour la reconnaissance de formes [35]

Les figures suivantes montre notre interface de programme l'ors du chargement d'une image de coupes scanner.

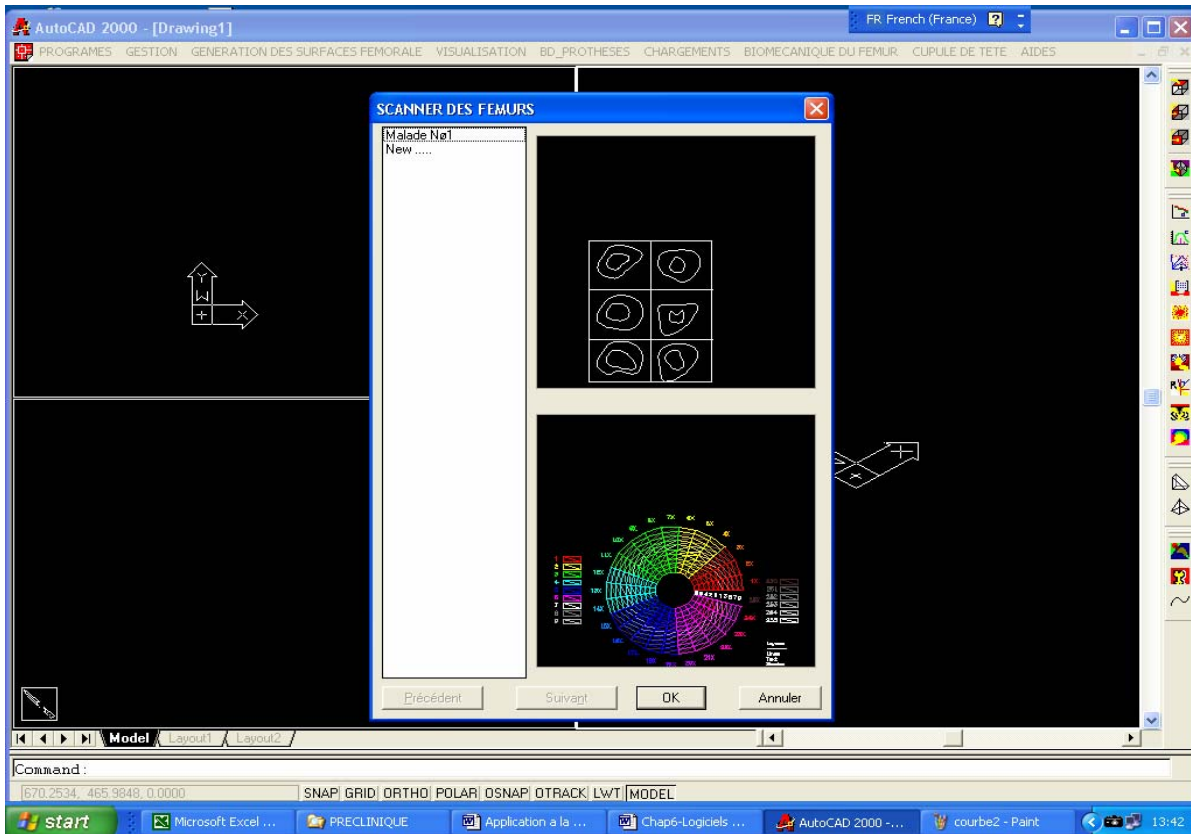


Fig. 3.4 : Insertion d'images [36]

La figure suivante montre l'intégration des courbes reconnues dans l'environnement 3Dimension de notre logiciel.

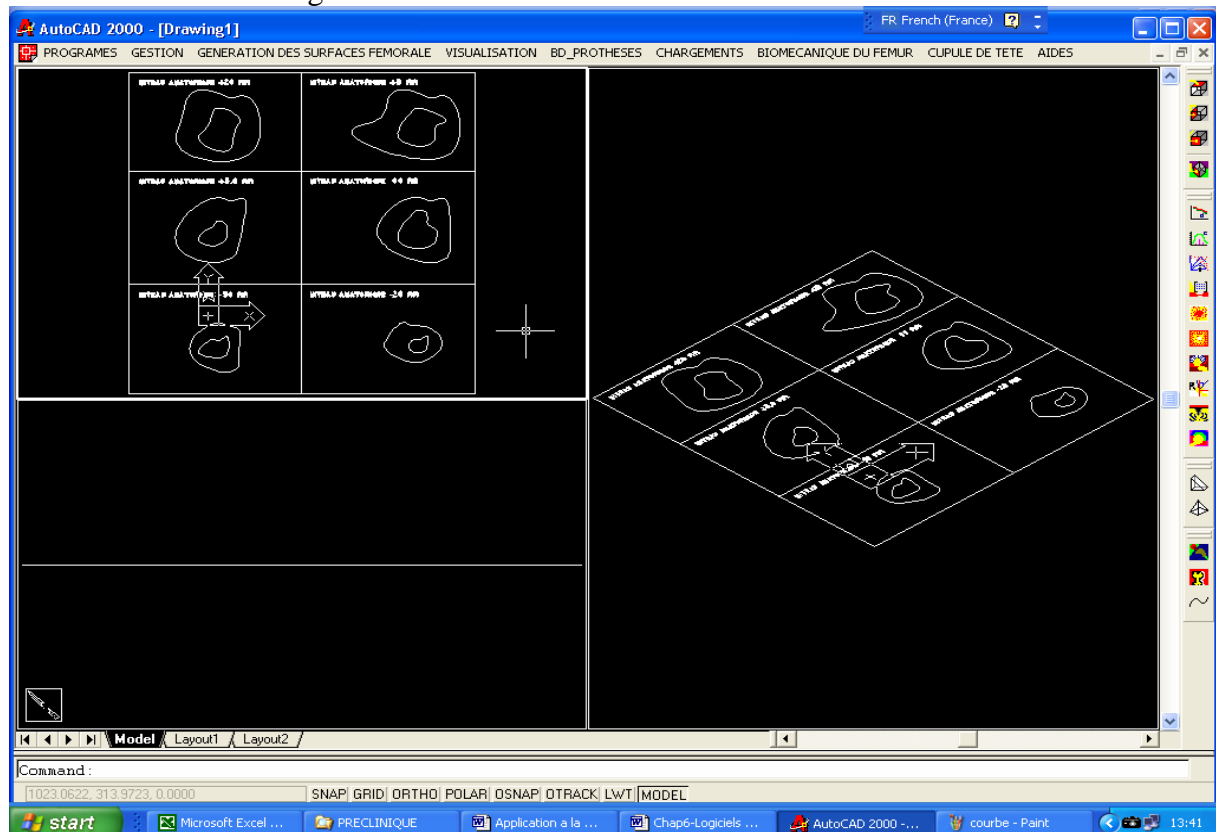


Fig. 3.5 : Vectorisation en courbes b-spline [36]

La vectorisation se fera par un algorithme de traitement d'image et qui fait une segmentation puis un ordonnancement, et puis une génération de courbe spline ou segment de droite ou un arc selon l'optimisation voulue.

La figure suivante montre l'ordonnancement des courbes calculées sur l'axe anatomique, représenté dans l'interface de notre logiciel.

Le programme élaboré nous permet d'enregistrer les coordonnées des segments de courbes, puis faire une succession de sélection des courbes par ordre pour les placer chacune une sur la position adéquate.

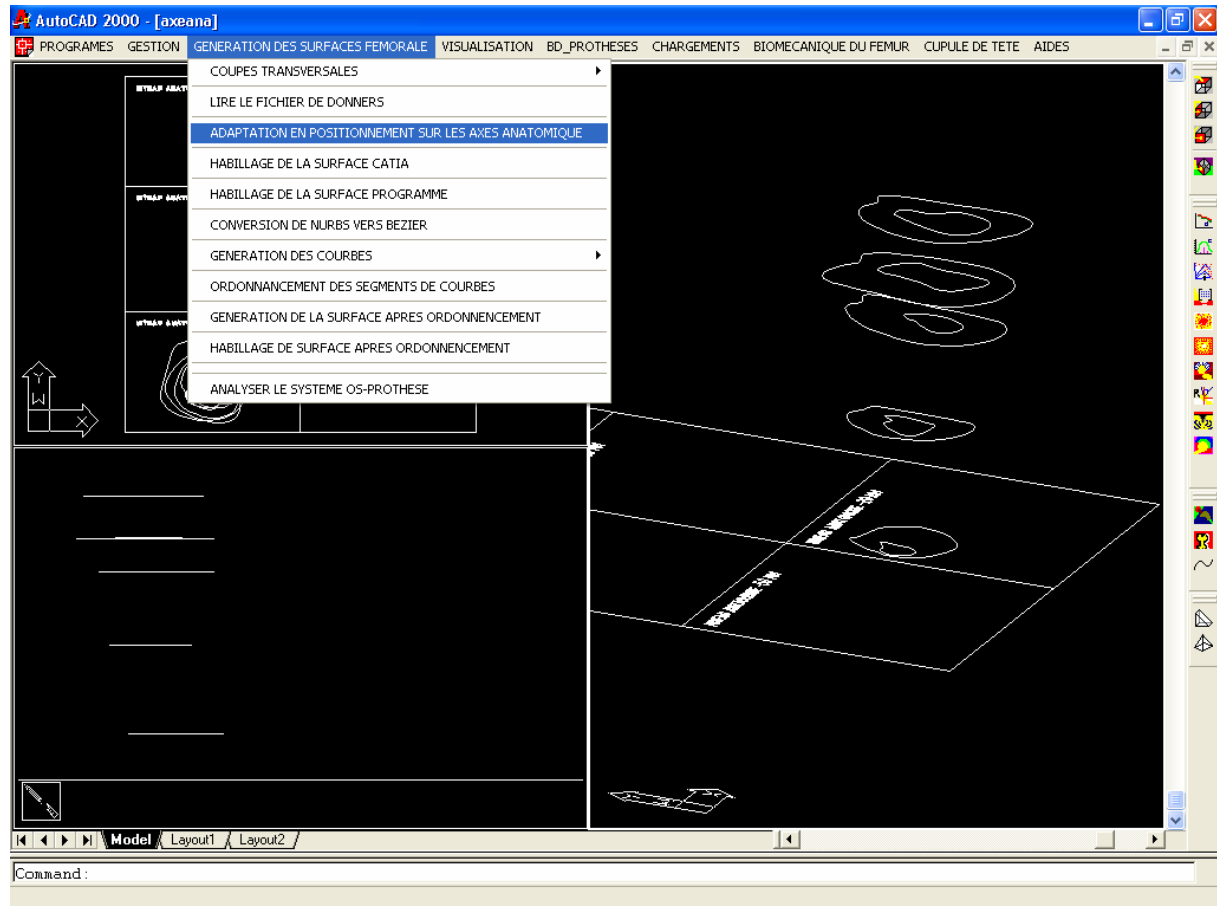


Fig. 3.6 : Adaptation des courbes en position sur l'axe anatomique [36]

### 3-2-2 Création du solide qui représente l'Os fémoral

On considérant les surfaces ente courbes comme des surfaces réglées.

De ce faite en à impliquer dans le programme une procédure d'habillage des coupes transversales.

- La surface réglée brute construite à partir des courbes prise par le scanner est envoyé sur un logiciel de CAO afin d'obtenir le réseau de contrôle d'une surface complexe.

Le calcul des points de contrôle sera fait par le logiciel a base d'un algorithme de De-Boor dit **interpolation inverse**, c a d, pour calculer un point sur la surface en utilise l'algorithme de De-Boor directe, et dans le cas ou les points sur la surface existe et en doit déterminer les sommées en est dans le cas inverse.

Le chargement ce fait de notre programme vers un format DXF, puis charger dans une esquisse d'une session du logiciel CATIA, comme est montré dans la figure (4-7). Le volume fémoral étant défini comme un volume de contrôle de la surface réel, le résultat sera transféré vers le logiciel CATIA pour ressortir le réseau de contrôle selon le model NURBS comme indiqué en lignes vert et points orange dans la figure suivante. En fera la lecture par chargement des points du réseau, notre programme lira les points et fasse la génération finale de la surface fémorale comme il va être préciser dans les parties suivante.

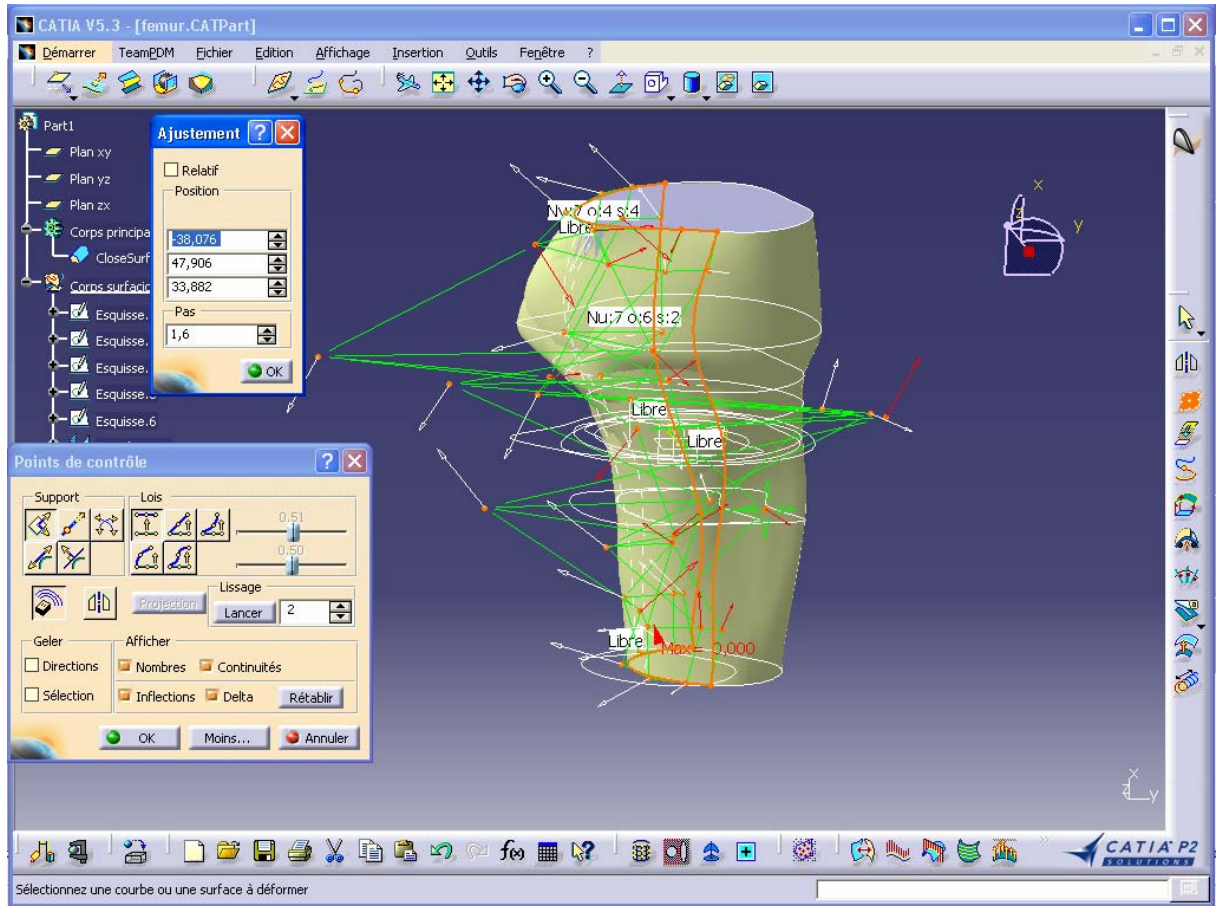


Fig. 3.7 : Réseau de contrôle de la surface dans CATIA system [21]

### 3-2-3 Régénération de surface fémorale avec précision

Cette étape est la plus importante car a partir du réseau de contrôle en calculs la géométrie fine par le modèle mathématique **NURBS**, ce model nous à permis de calculer n'importe quelle point sur la surface, c a d n'importe quelle courbe sur n'importe quelle niveaux anatomique du fémur.

Il a été présenté dans les chapitres précédents les descriptions de ce models géométrique des surfaces complexes.

On rappelle que :

$$P(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^t h_{ij} S_{ij} N_{im}(u) N_{jr}(v)}{\sum_{k=0}^n \sum_{l=0}^t h_{kl} N_{km}(u) N_{lr}(v)}$$

Avec  $h_{ij} > 0$

P : Est le point sur la surface.

u : Le paramètre suivant la première direction de la séquence nodale.

v : Le paramètre suivant la deuxième direction de la séquence nodale.

S : Les points du réseau caractéristique de la surface.

N : Les fonctions d'interpolation qui caractérise le models.

Pour le détail de la subdivision voir chapitre N°3.

### Structure de données de la méthode

Du point de vue algorithmique, la structure de données associées à une telle décomposition des carreaux est de type quadrette ou structure arborescente utilisée pour la décomposition cellulaire de surfaces.

Il existe plusieurs types d'arbres qui découlent d'une subdivision uniforme, non uniforme ou avec contraintes respectivement selon les figures (a), (b) et (c).

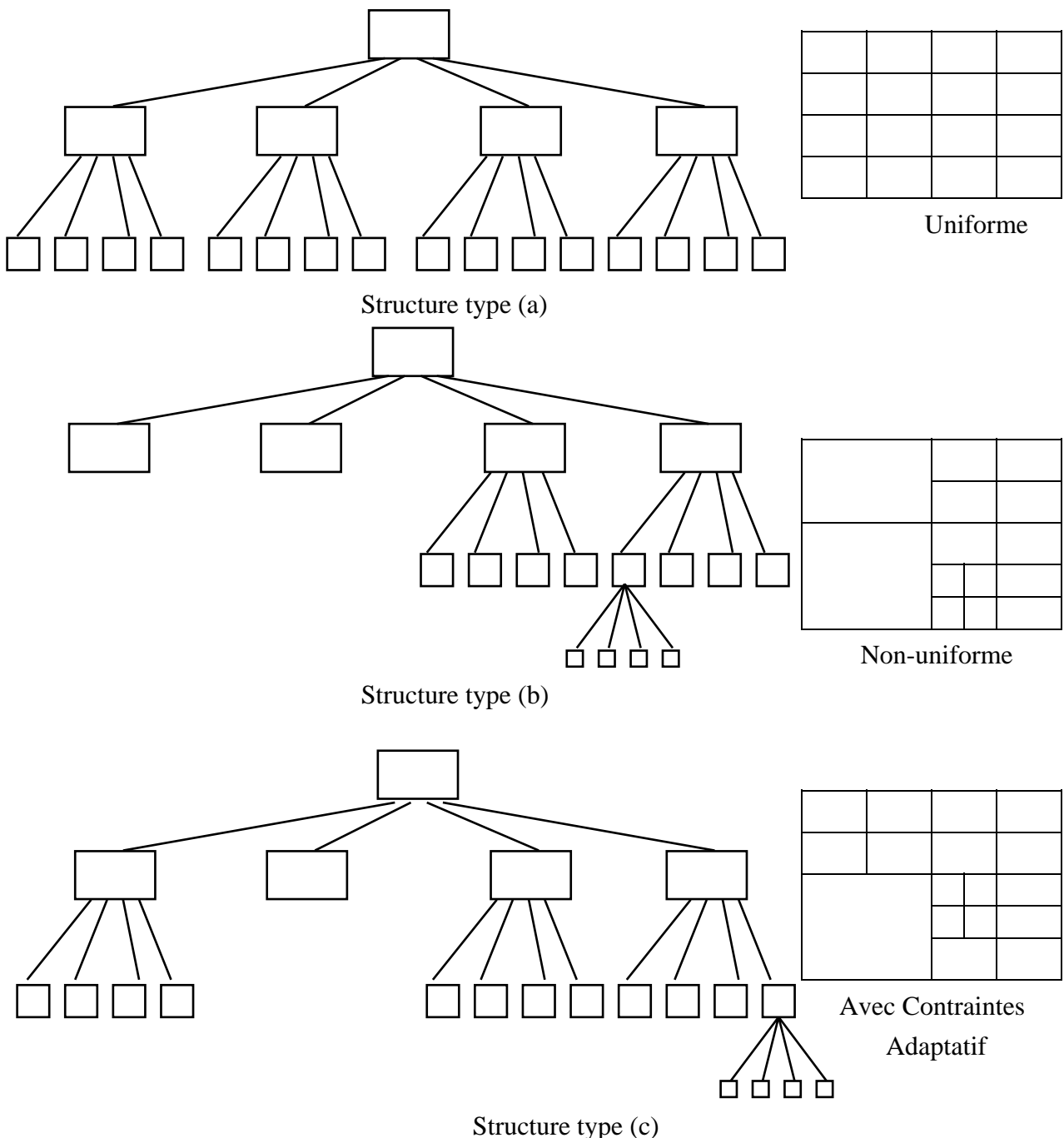


Fig. 3.8 : Structures de données [35]

En démarre initialement le calcul d'une surface NURBS pour reconstruire le réseau de carreaux de BEZIER, et ensuite chaque carreau subit la décomposition suivant la structure arborescente.

- Comme en a cité la subdivision suivant l'algorithme de De-Casteljau d'un carreau de surface modélisé par une fonction polynomiale et un nuage de points est la première principale opération, puis noté points de contrôle de la surface, vers quatre réseaux des mêmes caractéristiques mathématique (degré des fonction d'interpolation).
- En stock les quatre réseaux dans un vecteur de longueur (04) dont les éléments de ce vecteur sont des réseaux, puis en pose dans le programme les plans de coupe de 01 vers n, et se d'après la précision voulue et la distance entre les niveaux de coupes.
- En passe par le vecteur des réseaux un par un, et en vérifie les paramètres suivants : le plan de coupe traverse le réseau et si oui en fait le teste de la planité du réseau c a d si en peut considérer le réseau comme un plan, si oui en fait le calcul plan-plan pour avoir le segment de droite qui va construire la courbe de coupe du premier plan, et si le plan ne passe pas par le réseau en va passer à l'autre réseau dans le vecteur, mais dans le cas ou le plan coupe le réseau et ce dernier n'est pas considéré comme plan en va l'enregistrer j'us qu'a se quand termine le vecteur de réseau puis en va subdiviser les réseaux coupés par le plan de coupe mais ne satisfais pas la planité, alors l'algorithme est conçu selon cette philosophie de teste et de calculs.

En montre dans la figure suivante, un exemple de subdivision d'un carreau de surface avec suivant la structure précédente en vue de dessus.

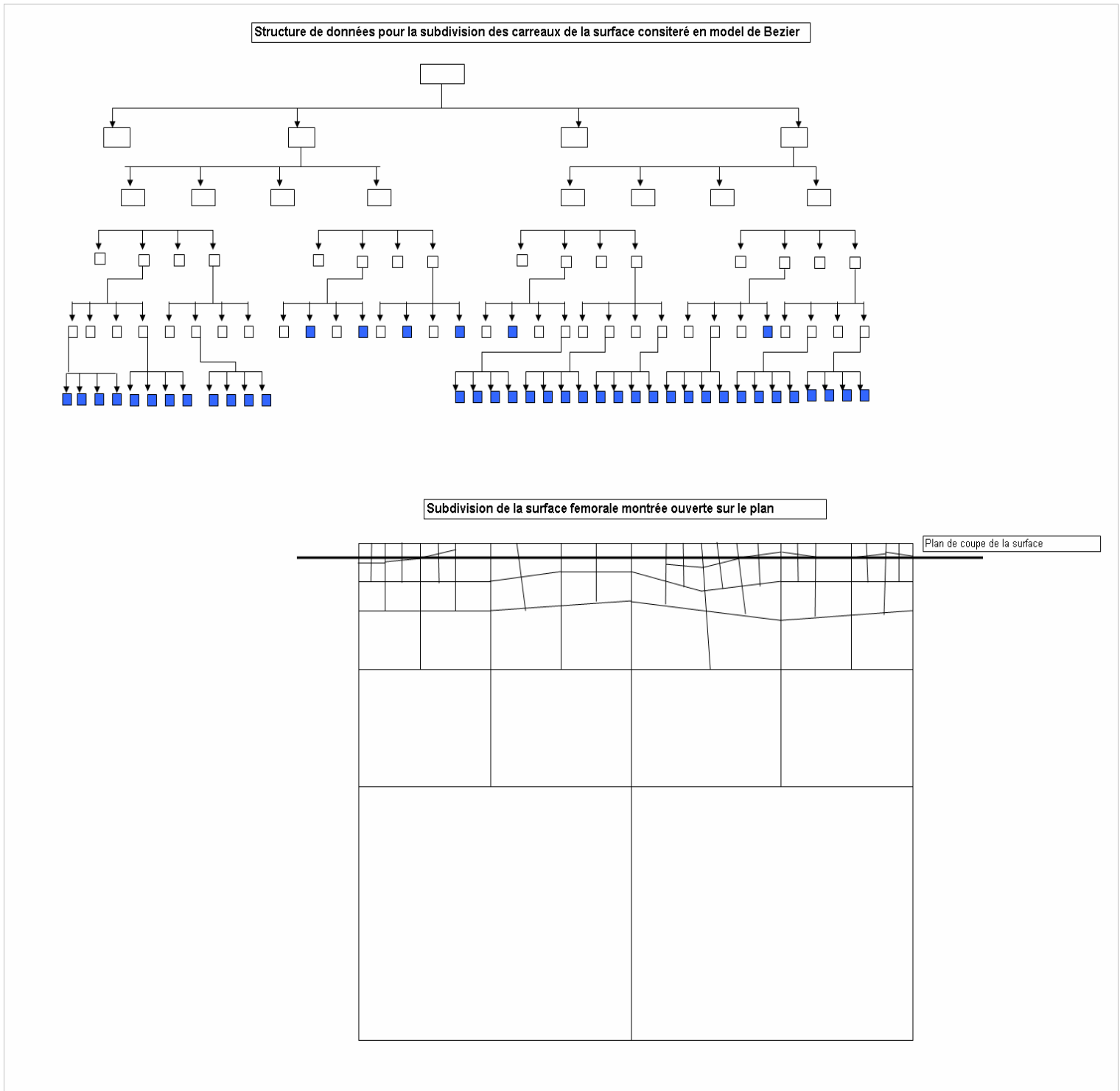
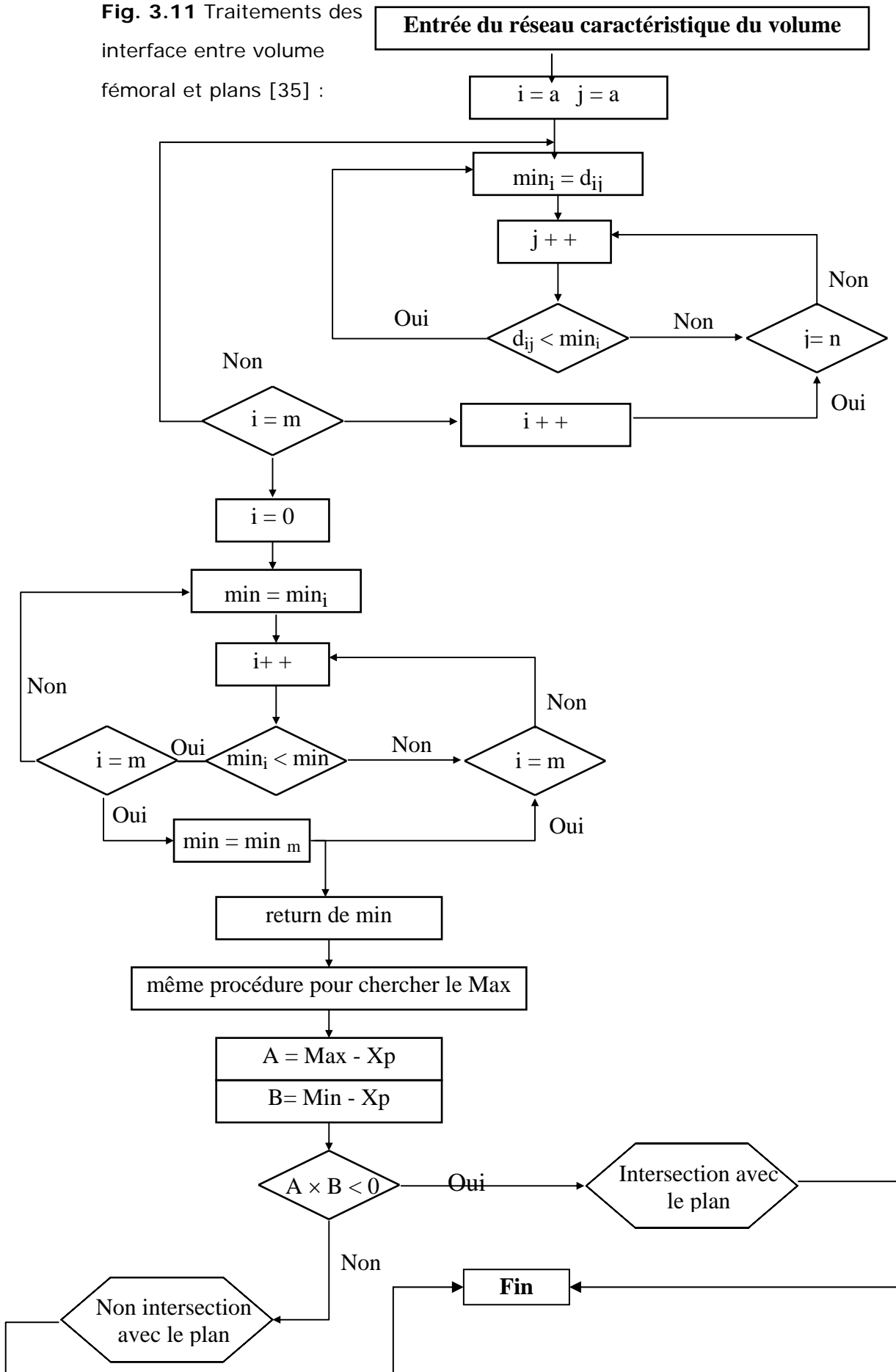


Fig. 3.9 : Structure de données d'un exemple [35].



**Fig. 3.11** Traitements des interface entre volume fémoral et plans [35] :



### 3-2-4 Ordonnements des segments de droite qui construisent les courbes

La méthode de subdivision et adaptative, en conséquence, les segments de droite calculés sont désordonnés, alors un ordonnancement des segments dans chaque courbe est indispensable pour que les courbes NURBS englobant la nouvelle surface et qui sont similaires aux courbes des segments, doivent avoir des points de constitution géométrique ordonnés.

En fait le programme qui fait l'ordonnement par MATLAB.

Les points seront stockés dans un fichier de données sous forme d'une matrice de  $N \times 3$ , soit les trois colonnes sont les coordonnées de l'espace.

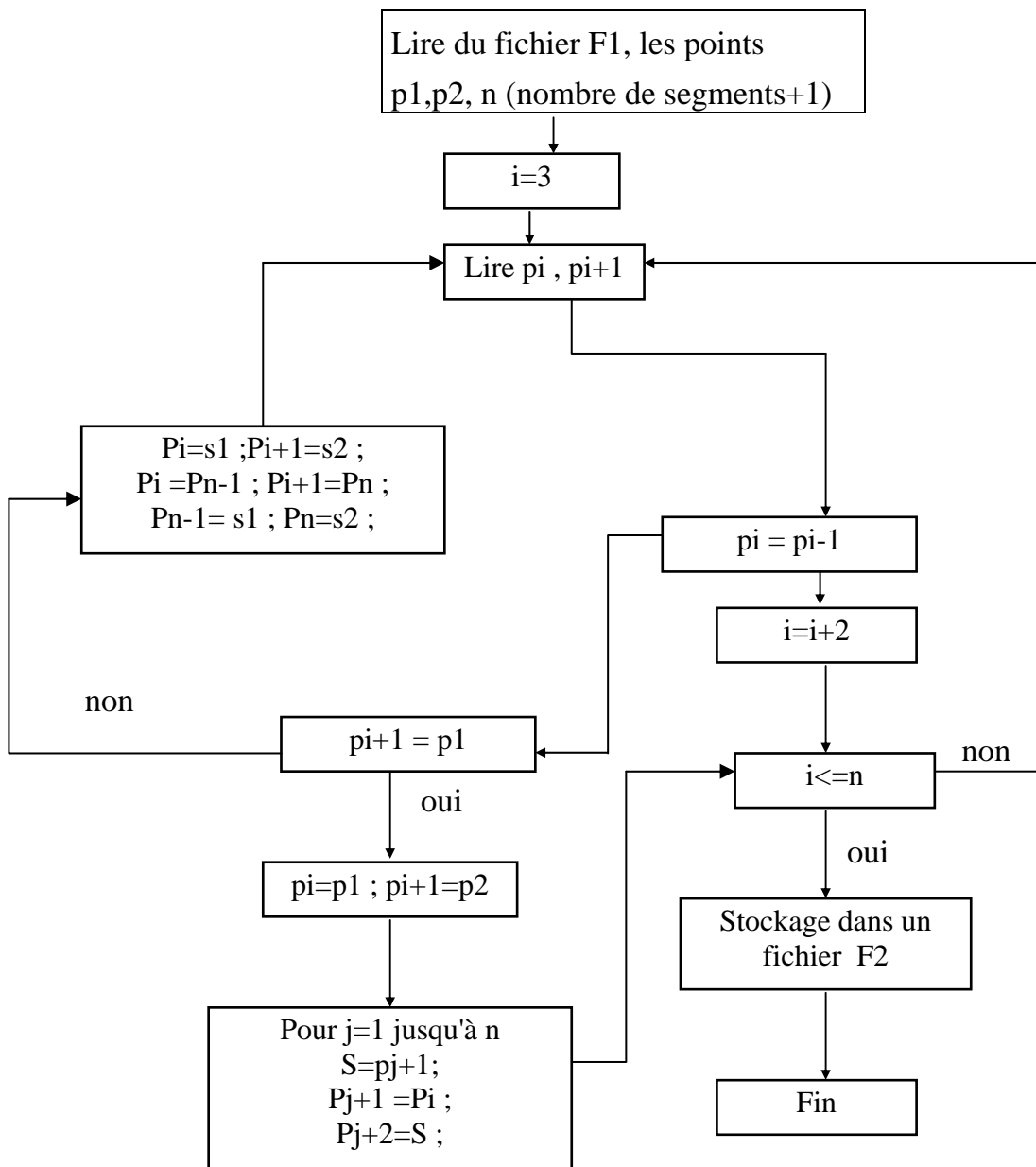


Fig. 3.12 : Organigramme d'Ordonnement [35]

La figure suivante est une image de notre logiciel qui montre les étapes pour faire l'ordonnancement des segments de droites.

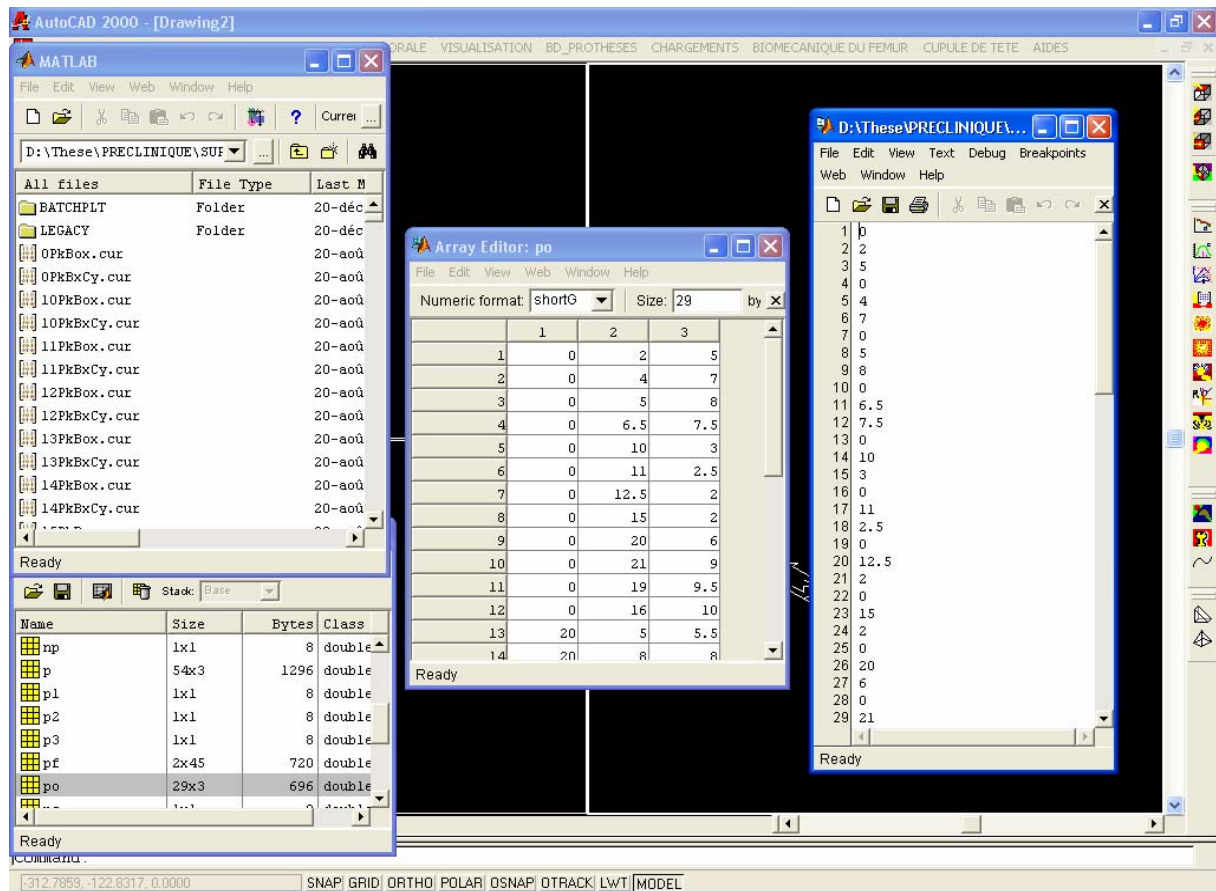


Fig. 3.13 : Programme d'ordonnement [36]

### 3-2-5 Habillage des nouvelles courbes calculées

Maintenant on arrive à l'étape de faire un habillage général sur ces courbes, ce qui est très délicat.

Dans notre interface on peut générer une surface réglée entre deux courbes successive, mais s'il y a des centaines de courbes, comment faire sélectionner une par une et crée des centaines de tranche de surfaces ?

Pour répondre à cette question on a automatisé la procédure par un programme fait par le LISP et l'algorithme est le suivant :

#### Programme d'habillage de la surface finale

##### Partie C++

L'or du calcul des courbes d'intersections par l'algorithme de de-casteljau

Pour plan de coupe n=1 j'us qu a np

Pour le nombre des resaux BEZIER r=1 :nr

Enregistrer le premier point calculé dans un fichier habillage.txt

cn=cn+1 (conteur de plans de coupes)

fin resaux

fin de-casteljau

## Partie AUTOLISP

Ouvrir le fichier habillage.txt

Lire nr

Pointer le début de fichier

Pour n=1 : np

    Lire pt(n) et pt(n+1)

    Sélectionné l'entité qui passe par pt(n)

    Sélectionné l'entité qui passe par pt(n+1)

    (setq ss(n) (ssget pt(n)))

    (setq ss(n+1) (ssget pt(n+1)))

    %% le programme sélectionne automatiquement les deux courbes

    Crée une surface réglée entre les deux courbes

Fin pour n

Fermé le fichier habillage.txt

Le problème de continuité tangente des surfaces ne se pose pas dans notre cas car les pas entre les tranches sont plus petite que la précision qui va être calculé en éléments finis.

En montre dans notre logiciel la Surface après habillage finale puis en envoi la surface vers CATIA pour faire le remplissage en solide et enfin faire un rappelle par format Dxf comme c'est montré dans la figure suivante.

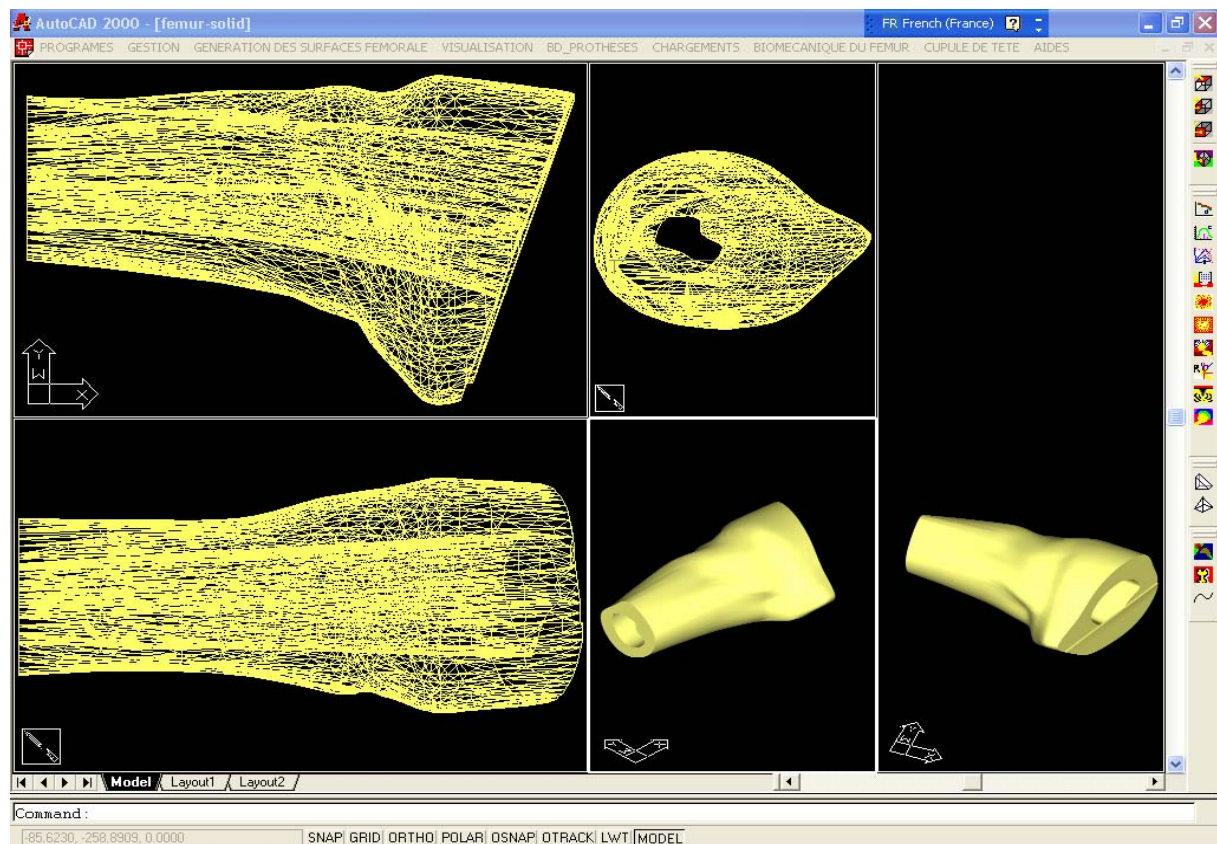
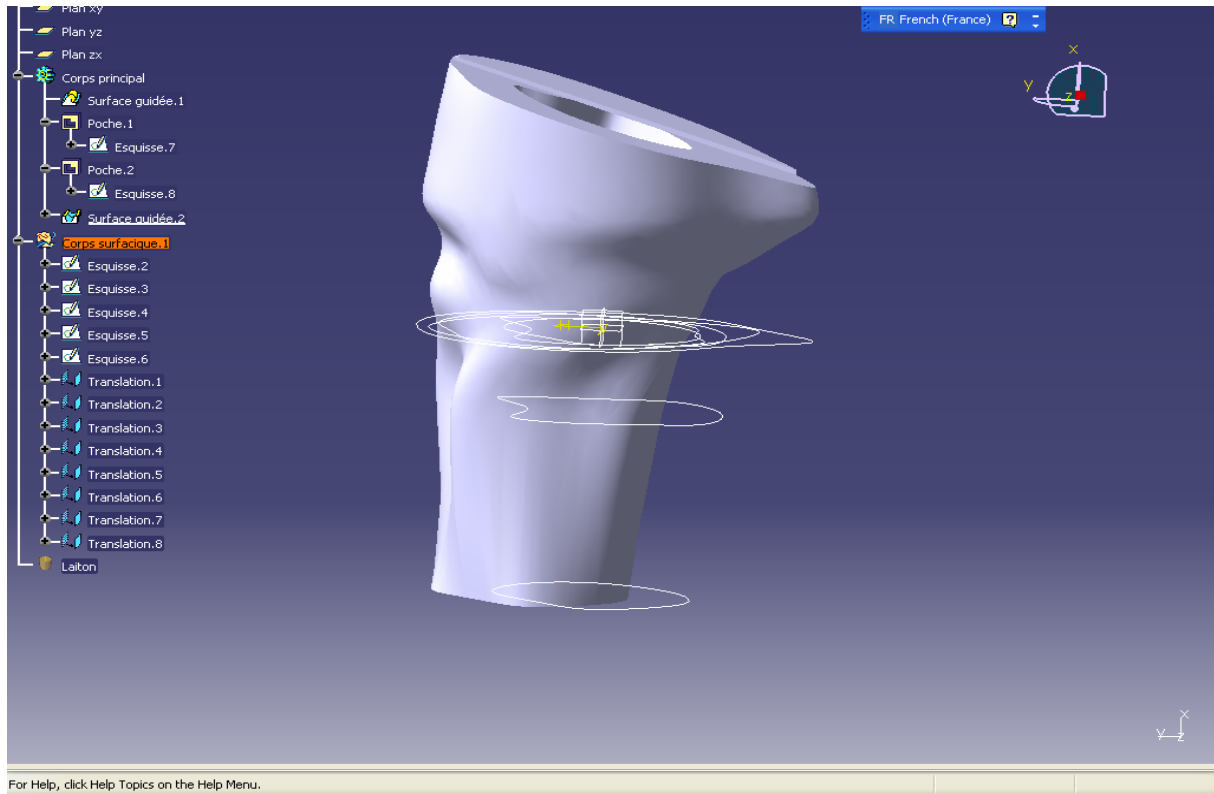


Fig. 3.14 Habillage final dans notre logiciel [36]

## Remplissage

Par suite en procède au remplissage de ces surfaces afin de permettre la discrétisation volumétrique de l'os fémoral.

La figure suivante montre le volume d'os déjà calculé, remplis de matire considéré solide et se dans le logiciel CATIA Solution qui facilite cette tâche.



**Fig. 3.15** : Remplissage de la surface en solide CATIA [36]

### 3-3 Insertion d'une prothèse a partir d'une base de données

Dans notre programme une base de données géométrique avec une liste de caractéristiques techniques est programmée d'une façon ouverte pour remplissage. Le remplissage se fait d'une façon automatique par le chargement des fichier DXF produit par les fabricants des prothèses, et le chargement se fait dans une base de données interne au logiciel comme étant une entité prédéfinie et le choix sera fait par le non du matériau et la forme sur le cliché.

La figure suivante montre notre logiciel l'or d'une insertion d'une prothèse fémoral.

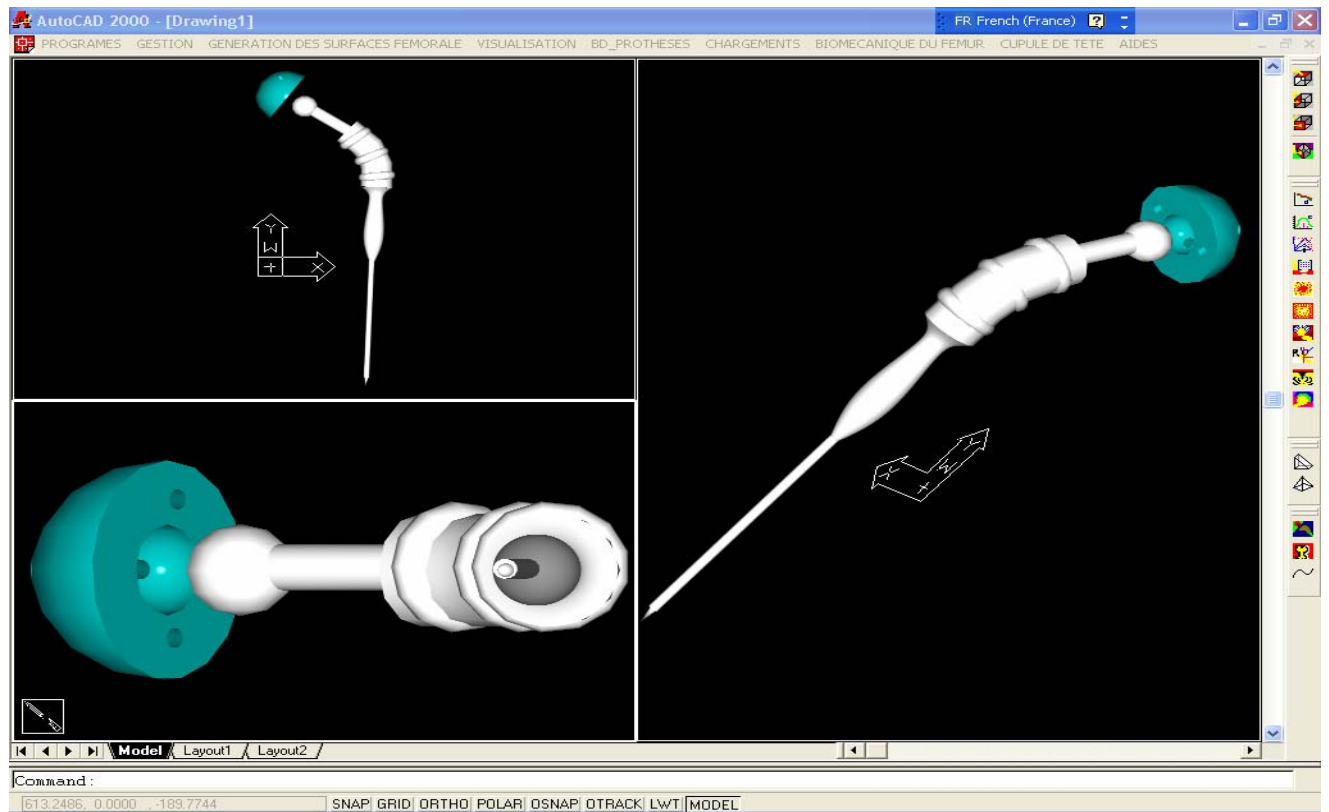


Fig. 3.16 : La base de données des prothèses [36]

La vérification géométrique se fait par le montage interactif de la prothèse sur la forme calculé de la surface fémorale, en respectant les normes d'adaptation et les intervalles de tolérances entre la surface extérieure du fémur et l'implant.

En montre dans cette figure, un montage entre l'os calculé et un prototype d'une prothèse de hanche, dans l'interface de notre logiciel.

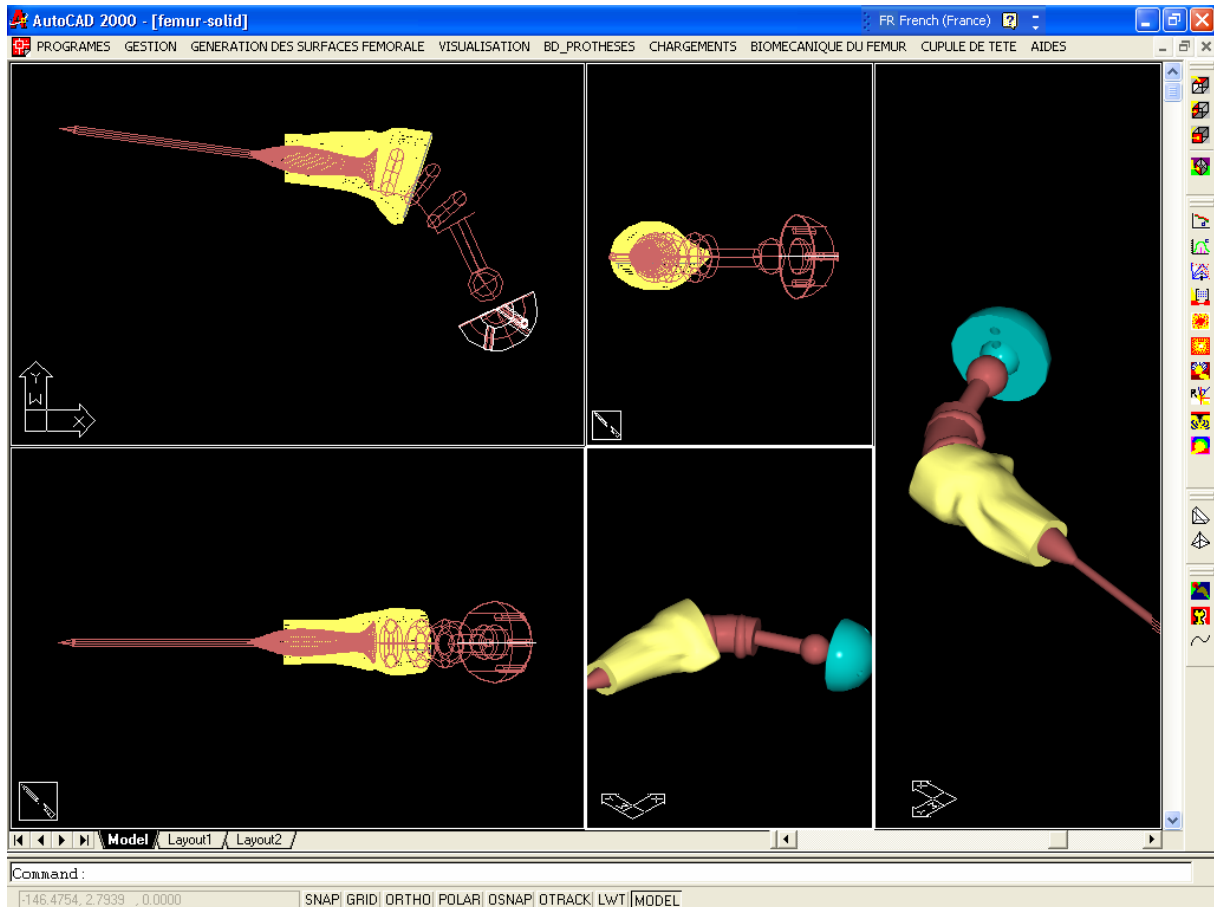


Fig. 3.17 : Ensembles Os-Prothese dans notre interface [36]

### 3-4 Insertion de l'ensemble Os-prothèse dans le logiciel d'analyse des systèmes

En envoi l'ensemble monté Os-Prothèse vers COSMOS ou FEMAP pour maillage, puis vers ANSYS pour analyse ou à l'intérieur de COSMOS, et se, par l'intermédiaire du format SAT ou IGES ou DXF, les entités seront reconnues comme étant des corps solides, et puis les opérations qui vont être suivies sont :

- Discrétisation en éléments finie (modèle tétraédrique) avec une haute précision.
- Définition des surfaces des contacts.
- Définir les paramètres de contacts, soit, Viscosité ; Etat de surface ; Coefficient d'adhérence ; Coefficient de glissement ; Précontrainte d'assemblage.
- Faire appel à ANSYS afin de créer le modèle analyse des contraintes et déformation.

La figure suivante montre dans notre logiciel, l'étape d'envoi de l'ensemble Os-Prothèse vers le logiciel d'analyse, puis le début du maillage dans le logiciel d'analyse dans les deux autre figures.

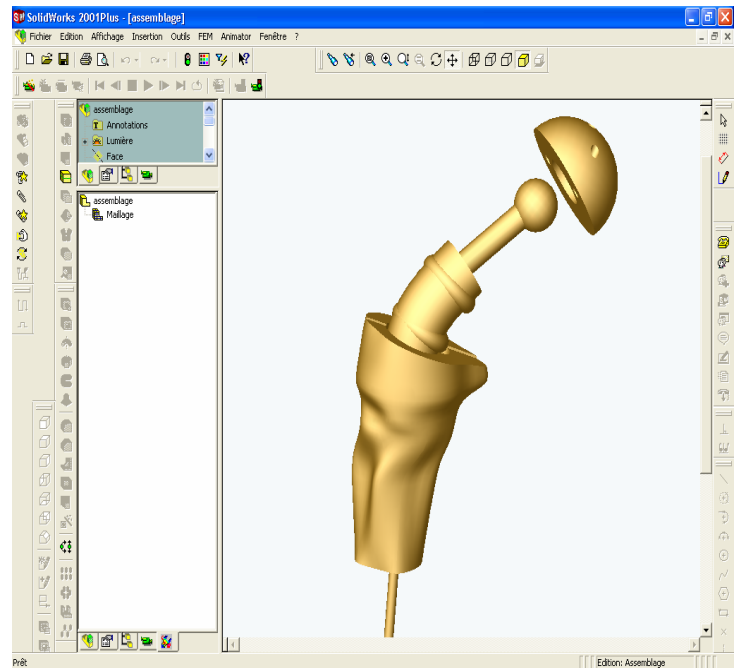
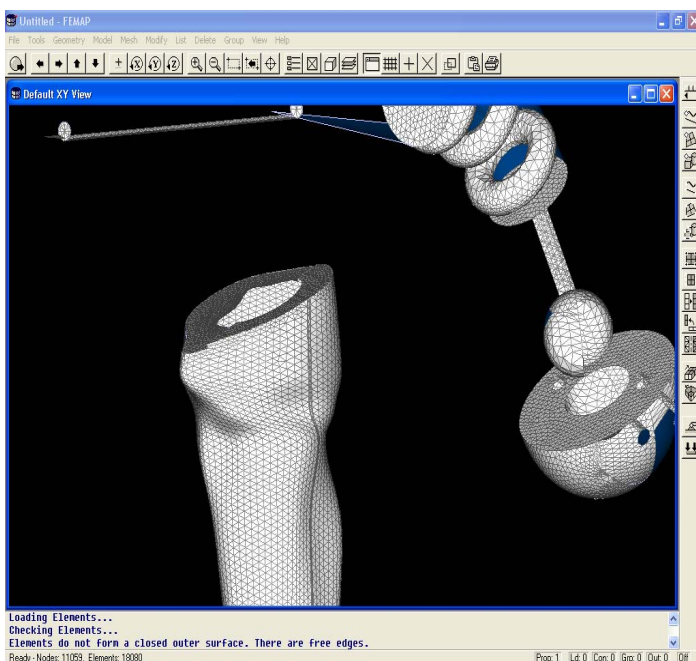
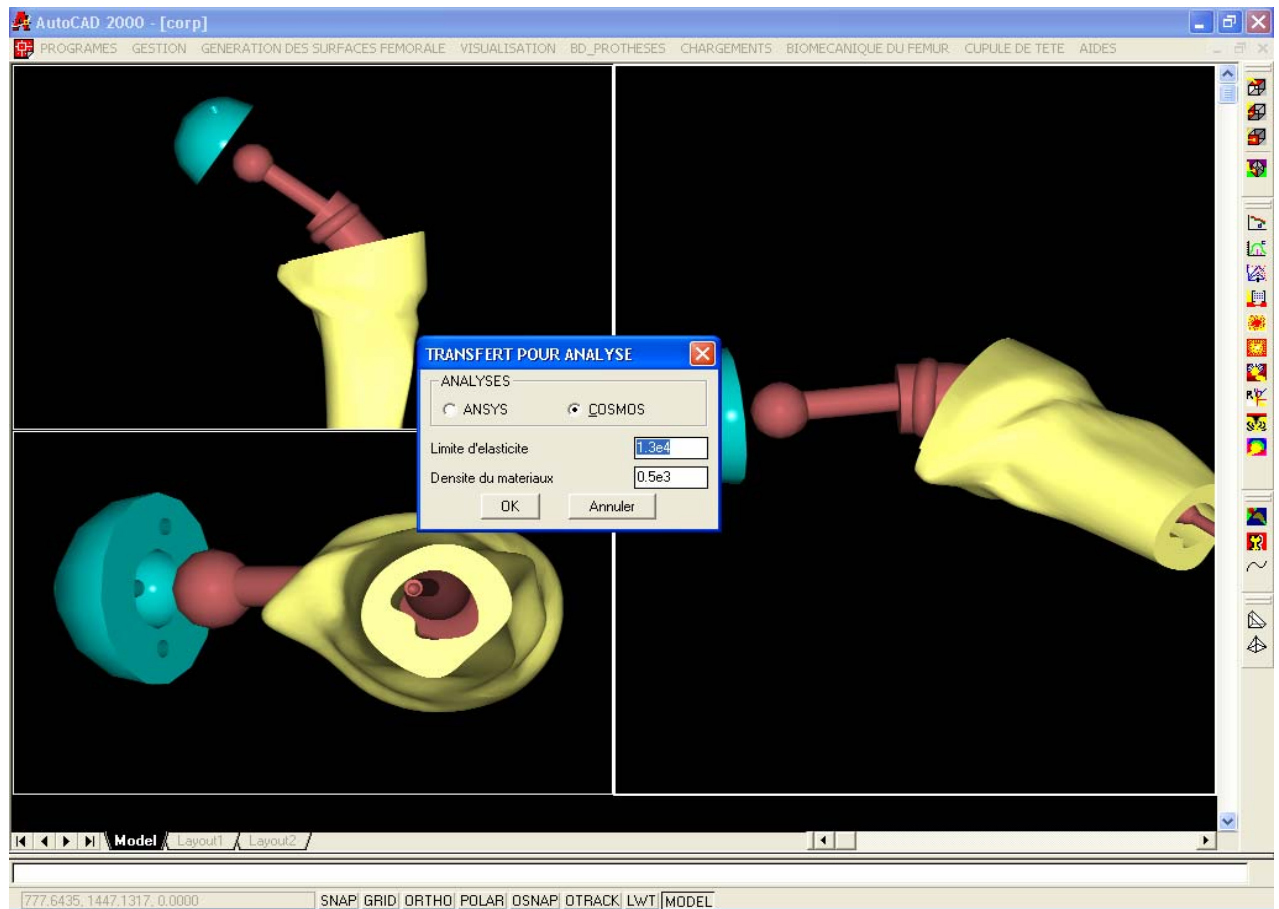


Fig. 3.18 : Maillage de l'ensemble fémur-prothèse dans le système FEMAP a gauche et COSMOS a droite [36]

## Conclusion

L'étude par éléments finis nous permis d'accéder à l'état des contraintes et des déformations à l'intérieur de l'os du fémur.

La modélisation par éléments finis demande de définir des caractéristiques mécaniques précises et de se rapprocher de la réalité. Elle permet de modifier facilement les paramètres, telle que la géométrie locale, les caractéristiques mécaniques des matériaux, le chargement et les conditions aux limites.

L'exploitation des résultats se trouve également facilitée puisqu'on peut obtenir une cartographie de tous les types de contrainte, déformation, déplacement, ainsi que la valeur en chaque nœud. Le modèle, réalisé par les ingénieurs des laboratoires de biomécanique, reflète la réalité anatomique de la hanche, car il s'inspire d'une coupe frontale, obtenue par tomodensitométrie passant à l'endroit où le diamètre cotyloïdien est maximal, avec ceci en compare notre approche qui calcule la géométrie par celle des coupes transversales.

Le but de notre étude étant de trouver une **méthode pré clinique optimale et qui génère l'exactitude de la géométrie osseuse** et l'influence de celle ci sur les contraintes que subit la hanche lors d'un appui, dans des zones bien définies, nous avons choisi une modélisation en trois dimensions, suffisamment précise dans sa géométrie pour nous permettre d'apprécier l'allure de la répartition des contraintes, et leur modification en fonction de la géométrie du fémur.

Les temps de calcul étaient rapides pour la génération de volume fémoral, et le temps qui reste c'est le temps de calculs des logiciels de CAO, cela nous a permis d'avoir un plus grand nombre d'éléments de la zone intéressante et, par conséquent, une plus grande précision dans les résultats des calculs, l'étude en 3D présente l'avantage d'avoir une géométrie plus proche de la réalité et de visualiser les contraintes sur l'ensemble du fémur et la prothèse.

## **CONCLUSIONS**

### **c-1 Couplage des relations de la biomécanique avec le système de CAO définie:**

Le phénomène du remodelage conduit a perturbé nos calculs si on le néglige, d'autre part en a introduit le phénomène par le calcul de la variation de densité osseuse a travers le temps en démarrent par le paramètre de la densité apparente qui est considéré la densité pris par les image scanner.

Par le calcul des variations sur plusieurs périodes, en introduit les résultats dans le logiciel d'analyse ANSYS afin de calculer les contraintes et les déformations dans plusieurs périodes pour définir la durée de vie et le comportement de l'os, et les paramètres à introduire sont.

#### **a- Variation de densité osseuse**

C'est la variation des caractéristiques et structure de l'os au cours du temps.

#### **b- Niveau de contraintes du tissu osseux**

C'est une variable qui nous donne les valeur réels des contraintes sur la surface osseuse du contacte avec la prothèse et celle qu'on va la calculer par l'introduction des contrainte calculer par le logiciel ANSYS et celle qu'on va prendre seulement les éléments tétraédriques extérieurs.

#### **c- L'énergie d'identification**

C'est l'énergie dépensée pour arrivée à un certain niveau de remodelage de l'os (voir détail chap5).

#### **d- Le tenseur d'élasticité de l'os**

C'est le calcul du tenseur d'élasticité osseuse dans les zones considéré orthotrope et le tenseur sera calculé sur les deux directions principales.

#### **e- Le tenseur d'endommagement de l'os**

C'est le tenseur qui calculé a partir du tenseur d'élasticité et celui qui définit le comportement mécanique de l'os l'ors qu'il dépasse les limites élastique pour introduit l'os dans un état de résorption (diminution de la densité osseuse) ou un état d'apposition (augmentation de la densité osseuse) pour la théorie.

#### **f- Direction de la résultante des forces d'applications**

C'est la partie qui calcul la valeur et la direction de la force résultante de tous types d'application simulés sur l'os (voir détail chap5).

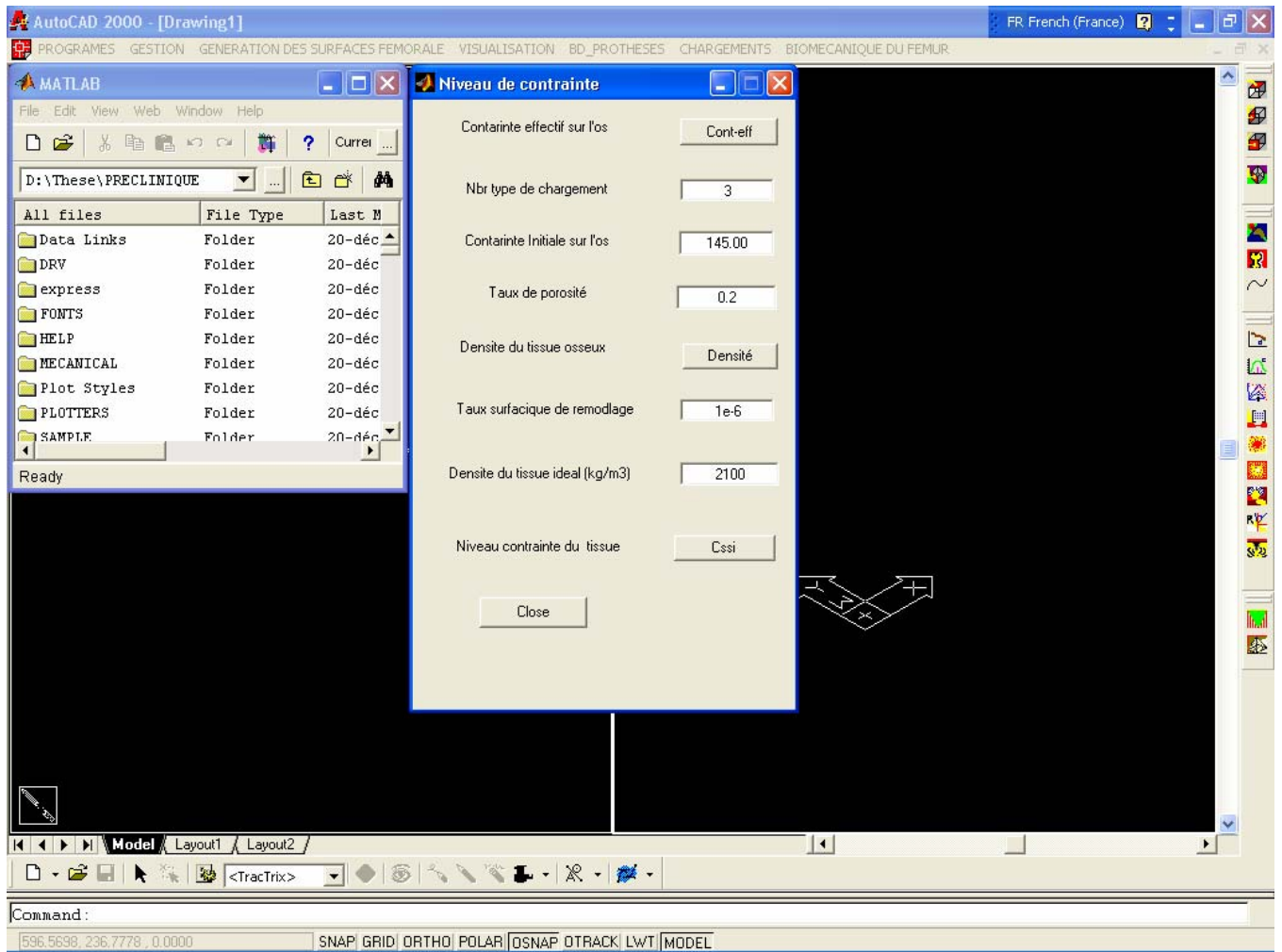
#### **g- Micro fissures sur la surface de l'os**

C'est le calcul du facteur d'intensité de contrainte et le facteur de cisaillement.

A partir de les quels en comparent les contrainte des éléments surfaciques aux nœuds calculés par ANSYS avec la limites de cisaillement calculé et prévoir les zones d'amorçages de microfissures a la surface de contacte os-prothèse.

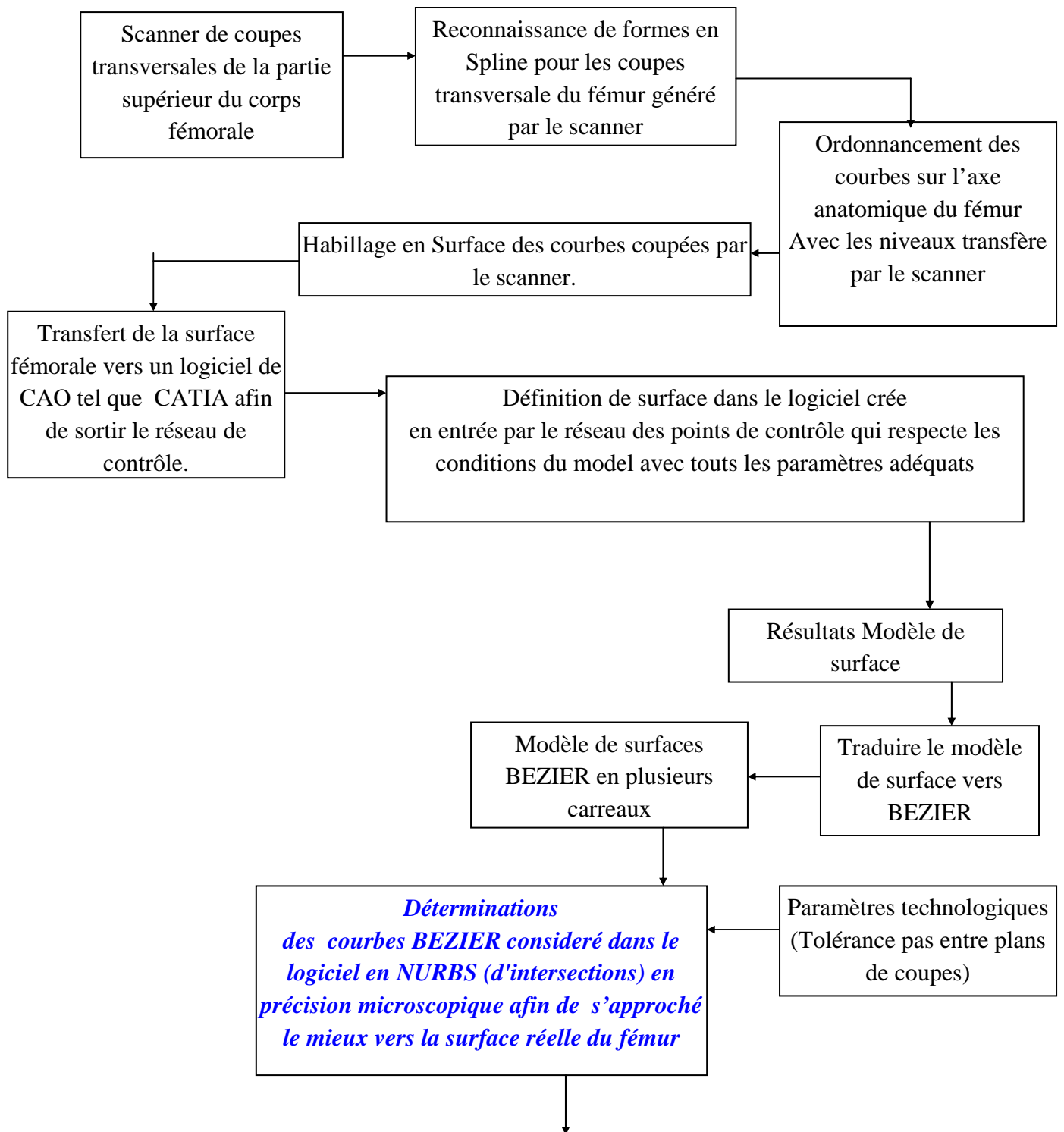
On a programmé tous ces paramètres et relations sur notre programme par MATLAB mais l'interface et les chargement des données seront traiter et gérer par notre logiciel sous environnement AUTOLISP.

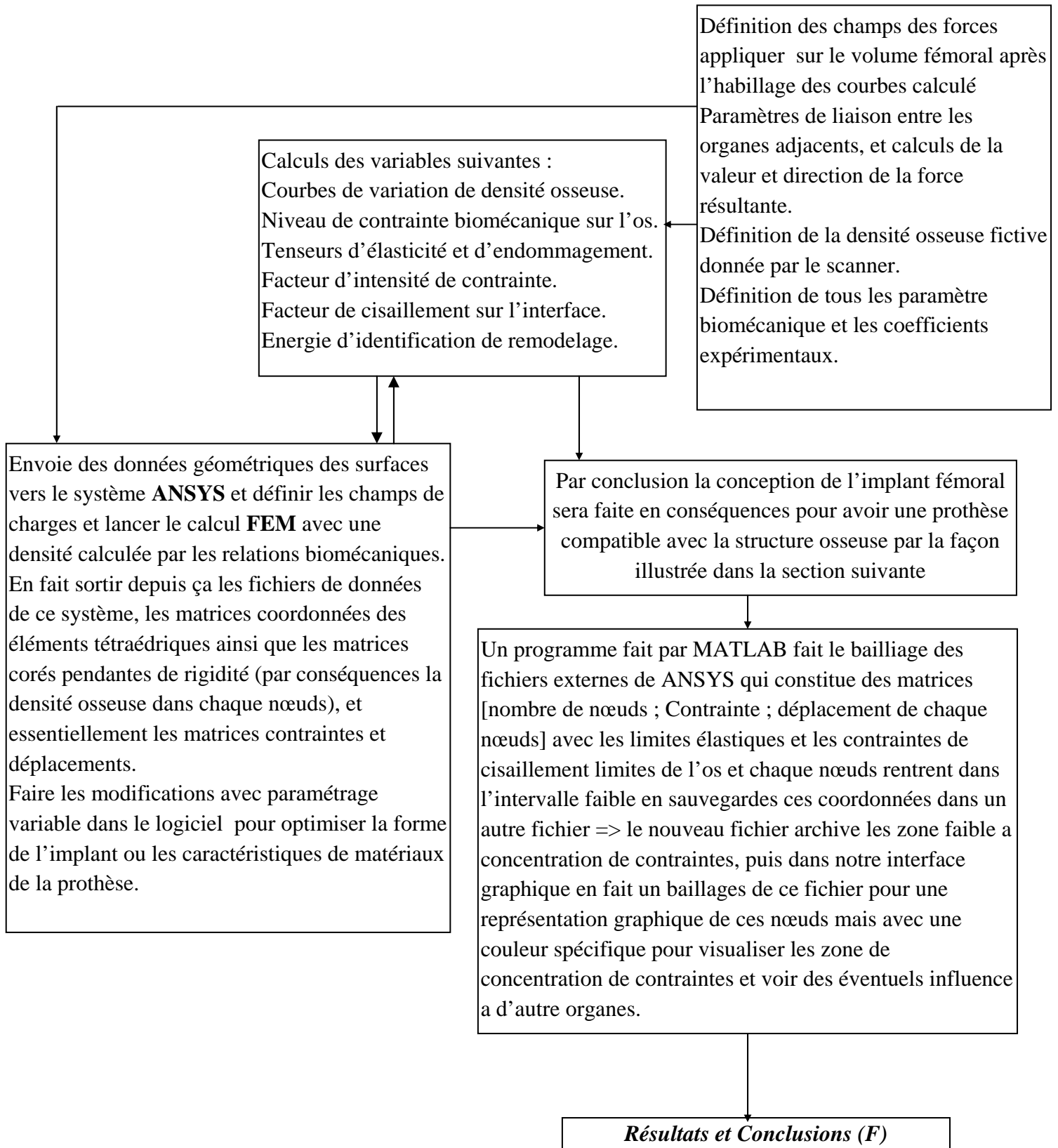
En a pris



**Fig. c-1** : Chargement d'un programme établi avec MATLAB en lien avec notre logiciel sous environnement AUTOLISP [36].

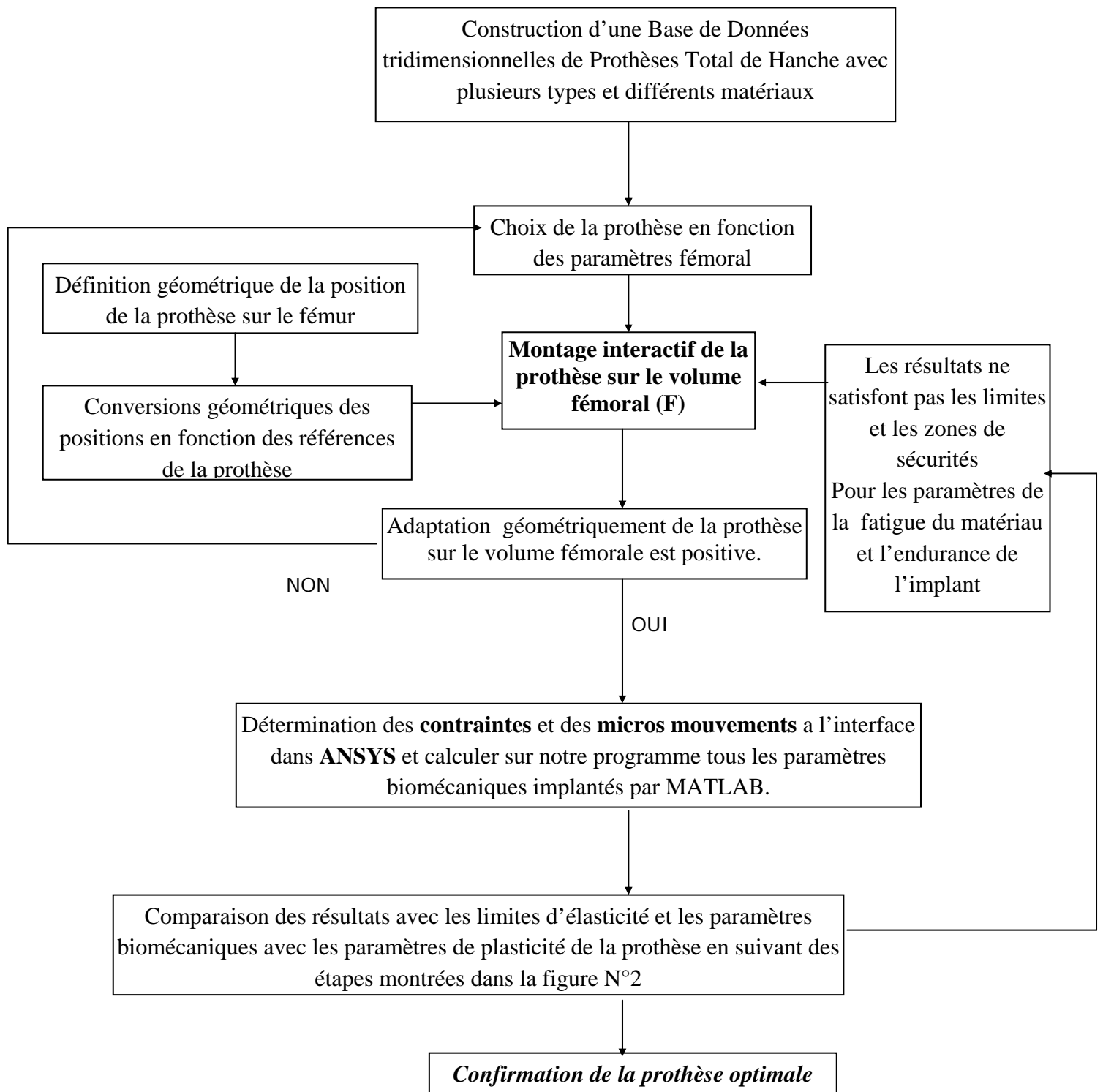
**c-2 L'analyse structurée de la Méthode de préparation à la simulation**





**Fig c.2 :** Diagramme de flots de données  
De la 1<sup>er</sup> phases de génération de surface fémorale  
Et préparation a la simulation pré-clinique [35]

**c-3 Simulation pré clinique sur l'environnement de calculs développé**



**Fig c.3 :** Organigramme de Simulation Pré clinique [35].

## **Conclusion**

Il reste donc encore une place pour le développement des composants prothétiques plus performants. L'outil informatique que nous présentons est un complément intéressant. Il permet de valider un choix d'options de dessin, de matériau ou de revêtement de surface de la prothèse en matière de géométrie avant toute implantation clinique avec une génération de surface en précision d'après les coupes transversales du scanner.

Dans le cadre d'une utilisation extensive, il peut permettre également d'évaluer lors de la planification préopératoire le positionnement des composants prothétiques afin d'une part d'optimiser la répartition des contraintes entre l'os et la prothèse et d'autre part d'augmenter les amplitudes de mouvements de la hanche prothétique.

- **Le logiciel élaborer permet de générer la géométrie tridimensionnelle de toutes les structures osseuses du corps humain en précision à partir d'un ensemble de coupes scanner.**

- Les différents programmes qui calculent les paramètres biomécaniques servent à élargir le diagnostic et définir l'état physique de l'assemblage en matière de pouvoir et compatibilité biomécanique, car il intervient le phénomène du remodelage en plus que l'os est une matière instable et n'accepte pas un comportement mécanique fixe, en précisant le phénomène du remodelage est incontrôlable et aléatoire car il est amorcé par les contraintes dues au mouvement physique de la personne qui n'est en aucun cas prédéfinie.

- La programmation du logiciel est faite par le langage AUTOLISP qui s'exécute sous environnement AUTOCAD. Le programme qui calcule la géométrie exacte du fémur est élaboré par le langage C++. Les programmes qui calculent les paramètres biomécaniques tels que l'énergie d'identification ou le taux de remodelage osseux sont programmés par MATLAB.

Le calcul du réseau de contrôle de la surface fémorale et qui caractérise la modélisation géométrique du fémur est calculé par la conversion de volume initiale calculé par la reconnaissance des courbes scanner en un fichier de format DXF et puis l'envoie sur CATIA Solution pour définir le réseau de contrôle, et après définition de la nouvelle géométrie par le programme C++ et insertion d'une prothèse d'après la base de données, en envoi l'ensemble vers ANSYS pour le calcul des contraintes et les déformations et faire changer à chaque fois les paramètres mécaniques d'après les résultats des programmes MATLAB et tous ces systèmes sont indispensables pour l'exécution de l'organigramme pré clinique pour une reprise totale de hanche.

Le modèle biomécanique devrait donc être un outil d'évaluation pré-clinique utilisé en pratique courante dans le développement de nouveaux dessins prothétiques.

## Annexe A1

### Rappelle Mathématiques

#### A1.1 – La normale et le plan tangent a une surface

Nous considérons ici des surfaces représentées par des fonctions à deux paramètres  $S(u,v)$ . la continuité des dérivées secondes est nécessaire pour que soient bien définies la normale à la courbe en chaque point de la surface. Si les dérivées premières

$$\frac{\partial S(u,v)}{\partial u} \quad \text{et} \quad \frac{\partial S(u,v)}{\partial v}$$

Notées respectivement  $S_u$  et  $S_v$  sont non nulles et non colinéaires dans le domaine considéré nous définissons alors le vecteur unitaire normal par :

$$\mathbf{n}(u,v) = \frac{S^u \wedge S^v}{|S^u \wedge S^v|} \quad (\text{A1-1})$$

Tout vecteur orthogonal au vecteur normal à la surface en un point donné  $S(u_i,v_i)$  est dit tangent à la surface en ce point. Ces vecteurs appartiennent au plan tangent à la surface en ce point. Nous définirons l'équation du plan tangent en  $S(u_i,v_i)$  par l'équation suivante :

$$(P - S(u_i, v_i)) \cdot \mathbf{n}(u_i, v_i) = 0 \quad (\text{A1-2})$$

#### A1.2 - Courbure et rayon de courbure des surfaces

Soit une surface représentée paramétriquement sous forme  $S(u,v)$ . On considère une courbe  $C(t)$  appartenant à la surface représentée par les paramètres  $(u,v)$  de la surface en fonction de  $t$   $\{u = u(t) ; v = v(t)\}$ .

$\mathbf{N}$  et  $\mathbf{n}(u,v)$  sont respectivement les normales à la courbe "c" et à la surface "s".

La courbure de la courbe "C" est donnée par [RUB 93]

$$K_c = K \cos(\varphi) = \frac{q \cdot D \cdot q^t}{q \cdot G \cdot q^t} \quad (\text{A1-3})$$

$$\text{où } q = \begin{bmatrix} du \\ dv \end{bmatrix}$$

$$D = \begin{bmatrix} \mathbf{n} \cdot S^{uu} & \mathbf{n} \cdot S^{uv} \\ \mathbf{n} \cdot S^{uv} & \mathbf{n} \cdot S^{vv} \end{bmatrix}$$

$$G = \begin{bmatrix} S^u \cdot S^u & S^u \cdot S^v \\ S^u \cdot S^v & S^v \cdot S^v \end{bmatrix}$$

Un intérêt spécial est porté lorsque  $\cos(\varphi) = \pm 1$ . Dans ce cas les courbes sont des sections normales à la surface c'est à dire qu'elles sont l'intersection entre la surface et un plan contenant la normale  $\mathbf{n}(u,v)$  au point considéré . La courbure d'une courbe appartenant à une section normale est appelée courbure normale de la surface en un point et est donnée par :

$$K_n = \frac{q \cdot D \cdot q^t}{q \cdot G \cdot q^t} \quad (\text{A1-4})$$

Par convention  $K_n$  sera positif lorsque le centre de courbure sera vers l'intérieur de la surface (l'extérieur de la surface est donné par le sens de la normale  $\mathbf{n}$ ).

Le rayon de courbure normale est donnée par :

$$\rho = \frac{1}{K_n} \quad (\text{A1-5})$$

On définit le vecteur rayon de courbure par :

$$\dot{\rho} = \rho \cdot n \quad (A1-6)$$

### A1.3 - Equation paramétrique d'une droite

Soient un point "Pu" de l'espace de coordonnées X0,Y0,Z0, et un vecteur n1 (a1,b1, c1).

L'équation paramétrique de la droite ( $\Delta$ ) passant par le point  $\rho_0$  et parallèle au vecteur n1 est donnée par :

$$\rho(s) = \rho_0 + n_1 S$$

Equation paramétrique d'un plan :

Soient un point  $\rho_0$  de l'espace de coordonnées X0, Y0, Z0, et deux vecteurs n1 (a1, b1, c1) et n2 (a2, b2, c2) tel que :

$$n_1 \wedge n_2 \neq 0 \quad \text{ou} \quad (n_1 \text{ n'est pas parallèle à } n_2)$$

L'équation paramétrique du plan (Q) passant par le point P0 et parallèle aux vecteurs n1 et n2 est donnée par :

$$Q(s,t) = P_0 + n_1 s + n_2 \cdot t \quad (A1-7)$$

### A1.4 - Intersection droite-Plan

Soit une droite définie paramétriquement :

$$P(u) = d + e \cdot t \quad (A1-8)$$

Soit un plan défini paramétriquement :

$$Q(u, w) = a + b u + c w \quad (A1-9)$$

$$a = \begin{cases} a_1 \\ a_2 \\ a_3 \end{cases}, \quad b = \begin{cases} b_1 \\ b_2 \\ b_3 \end{cases}, \quad c = \begin{cases} c_1 \\ c_2 \\ c_3 \end{cases}, \quad d = \begin{cases} d_1 \\ d_2 \\ d_3 \end{cases}$$

u, w, t sont des paramètres  $\in [0, 1)$

Pour qu'il y'ai une intersection il faut que les deux droites ne sont pas parallèle au plan .

Condition de non parallélisme

$$(b \wedge c) \cdot e \neq 0 \quad (A1-10)$$

Intersection droite plan :

$$P(t) - Q(u, w) = 0 \quad (A1-11)$$

$$d + e t = a + b u + c w \quad (A1-12)$$

La technique utilisée est d'éliminer deux paramètres en multipliant par le produit vectoriel de leurs coefficients et de déterminer la valeur du 3eme paramètre :

$$(1) \quad (b \wedge c) \cdot (d + e t) = (b \wedge c) \cdot (a + b u + c w)$$

$$\Rightarrow t_1 = \frac{(b \wedge c) \cdot a - (b \wedge c) \cdot d}{(b \wedge c) \cdot e}$$

$$(2) \quad (c \wedge e) \cdot (d + e t) = (c \wedge e) \cdot (a + b u + c w)$$

$$\Rightarrow u_1 = \frac{(c \wedge e) \cdot d - (c \wedge e) \cdot a}{(c \wedge e) \cdot b}$$

$$(3) \quad (b \wedge e) \cdot (d + e t) = (b \wedge e) \cdot (a + b u + c w)$$

$$\Rightarrow w_1 = \frac{(b \wedge e) \cdot d - (b \wedge e) \cdot a}{(b \wedge e) \cdot c}$$

d'où on tire le point d'intersection

$$\begin{cases} X_1 = d_1 + e_1 t_1 \\ Y_1 = d_2 + e_2 t_1 \\ Z_1 = d_3 + e_3 t_1 \end{cases} \quad (A2-16)$$

### A1.5 - Intersection Plan-Plan

Soit deux plans définis para métriquement :

$$P(u,v) = a + b \cdot u + c \cdot v$$

$$Q(s,t) = s + e \cdot s + f \cdot t$$

Où

$$a = \begin{cases} a_1 \\ a_2 \\ a_3 \end{cases}, \quad b = \begin{cases} b_1 \\ b_2 \\ b_3 \end{cases}, \quad c = \begin{cases} c_1 \\ c_2 \\ c_3 \end{cases}, \quad d = \begin{cases} d_1 \\ d_2 \\ d_3 \end{cases}, \quad e = \begin{cases} e_1 \\ e_2 \\ e_3 \end{cases}, \quad f = \begin{cases} f_1 \\ f_2 \\ f_3 \end{cases}$$

u, v, s, t sont des paramètres  $\in [0, 1)$

L'intersection existe si les deux plans ne sont pas parallèles donc les deux normales ne sont pas parallèle.

donc  $n_1 \wedge n_2 \neq 0$

où  $n_1 = b \wedge c$

$n_2 = e \wedge f$

(\*) si "c" n'est pas parallèle au plan "Q"

c-à-d :  $(e \wedge f) \cdot c \neq 0$

Alors on fixe le paramètre "n" et on le fait varier avec un pas " $\Delta u$ " et pour chaque n constant on calcule les valeurs des autres paramètres en faisant disparaître deux d'entre eux en multipliant par le produit vectoriel de leurs coefficients et on calcule la valeur du troisième (3eme) paramètre.

$$v = \frac{(e \wedge f) \cdot d - (e \wedge f) \cdot (a + b \cdot u)}{(e \wedge f) \cdot c}$$

$$s = \frac{(c \wedge f) \cdot (a + b \cdot u) - (c \wedge f) \cdot d}{(c \wedge f) \cdot e}$$

$$t = \frac{(c \wedge e) \cdot (a + b \cdot u) - (c \wedge e) \cdot d}{(c \wedge e) \cdot f}$$

En joignant les points trouvés on obtient le segment d'intersection.

### Algorithme

Introduction des données

les vecteurs  $a, b, c, d, e, f$

$$f(u,v) = a + bu + cv$$

$$a(s,t) = d + es + ft$$

si  $(b \wedge c) \wedge (e \wedge f) = 0$

Pas d'intersection

Sinon

si  $(e \wedge f) \cdot c \neq 0$  alors

on fixe la valeur de « u »

Faire pour « u » variant de 0 jusqu'à 1 avec un pas =  $\Delta u$

$$v = \frac{(e \wedge f) \cdot d - (e \wedge f)(a + bu)}{(e \wedge f) \cdot c}$$

$$s = \frac{(c \wedge f) \cdot (a + bu) - (c \wedge f)d}{(c \wedge f) \cdot e}$$

$$t = \frac{(c \wedge e) \cdot (a + bu) - (c \wedge e)d}{(c \wedge e) \cdot f}$$

si non

on fixe la valeur de "v"

Faire pour "v" variant de 0 jusqu'à 1 avec un pas =  $\Delta v$

$$v = \frac{(e \wedge f) \cdot d - (e \wedge f) \cdot (a + c.v)}{(e \wedge f) \cdot b}$$

$$s = \frac{(b \wedge f) \cdot (a + c.v) - (b \wedge f) \cdot d}{(b \wedge f) \cdot e}$$

$$t = \frac{(b \wedge e) \cdot (a + c.v) - (b \wedge e) \cdot d}{(b \wedge e) \cdot f}$$

Fin si

Si  $0 \leq u \leq 1$  et  $0 \leq v \leq 1$  et  $0 \leq s \leq 1$  et  $0 \leq t \leq 1$

Intersection plan plan

Sinon

Pas d'intersection

Fin si

Fin du programme

## Bibliographie

### **Modélisation en CAO:**

- [1] - La méthode des éléments finis: Jean Garrigues (Janvier 2002).
- [2] - Méthode Mathématique pour la modélisation des surfaces gauche 1992.
- [3] - Norbert Giambiasisi et Jean Claude Rault : Techniques graphiques interactives et C.A.O.
- [4] - Rubio 1992 Modélisation des surfaces.
- [5] - Jean Claude Léon : Modélisation et construction de surfaces pour la CFAO, Hermes
- [6] - MIE. Mortensen : Géométric modeling , 1985.
- [7] - Techniques Modernes de modélisation des surfaces en CAO.
- [8] - Gerald Farin: Department of computer science Arizona State University.  
Temps, Arizona, «Courbes et surfaces pour le CGAO».
- [9] - N.Piskounov: Calcul différentiel et Intégral  
Tome II, Mir Moscow, 1988
- [10] - Techniques de l'ingénieur juin 1998 (Méthodes de la CAO).

### **Biomécanique et Mécanique des Matériaux:**

- [11] - RDM
- [12] - Techniques de l'ingénieur juin 1998 (Résistance des matériaux).
- [13] - Techniques de l'ingénieur juin 1998 (Matériaux Métallique).
- [14] - Techniques de l'ingénieur juin 1998 (Méthodes des éléments finis).
- [15] - Techniques de l'ingénieur juin 1998 (Mécanique élastoplastique).
- [16] - Mécanique Elastoplastique.
- [17] - Médecine orthopédique
- [18] - Journal of Biomechanics: September 2002.

### **Programmation :**

- [19] - AUTO LISP : Manuel de programmation. Autodesk 1994.
- [20] - MATLAB : Référentiel de programmation.
- [21] - CATIA Solution et Références.
- [22] - ANSYS : pour définir les éléments et les paramètres pris en charge par ANSYS.
- [23] - COSMOS : Le système d'exécution de notre exemple d'application.
- [24] - FEMAP : pour définir la possibilité d'interface et de compatibilité entre logiciel avec  
notre programme.
- [25] - Gérard le Blanc : C++ 1990.

### **Publications:**

- [26] - Anisotropic bone remodelling model based on a continuum damage-repair theory  
M.Doblar et J.M.Garc. 2001.
- [27] - Modélisation expérimentale du champ élastoplastique en bout d'une fissure de fatigue.
- [28] - An elasto-plastic finite element model for polyethylene wears in total hip arthroplasty  
S.H.Teohn et W.H.Chan, R.Thampuran (octobre 2001).
- [29] - Slide track analysis of the relative motion between femoral head and  
acetabular cup in walking and in hip simulators  
Vesa Saikko\*, Olof Calonius (November 2001).
- [30] - Microstructural elasticity and regional heterogeneity in human femoral bone of various  
ages examined by nano-indentation  
J.Y.Rhoa, P.Ziouposb et J.D.Currey , G.M.Pharrd (November 2001).
- [31] - Residual stresses at the stem –cement interface of an idealized  
cemented hip stem, G.Avanzolinib (février 2002).
- [32] - Functional adaptation of cancellous bone in human proximal femur predicted by  
trabecular surface remodelling simulation toward uniform stress state,  
Ken-ichi Tsubota, Taiji Adachi et Yoshihiro Tomita.
- [33] - Determination of orthotropic bone elastic constants using FEA and modal analysis  
(février 2002).
- [34] - Equine cortical bone exhibits rising R-curve fracture mechanics (October 2002).

**[35] – Propre développement de la présente thèse.**

**[36] – Logiciel développé par nos soins.**