

## Chapitre 4

### La méthode de quadtree modifiée

#### 4-1 Introduction :

Pour la manipulation de données, l'informatique graphique fait souvent appel à des structures hiérarchiques. Celles-ci sont essentiellement de deux types [49] : "*hiérarchie dans l'espace objet*" et "*hiérarchie dans l'espace image*".

La première consiste en une décomposition des objets en leurs éléments constitutifs. C'est une représentation particulièrement efficace pour résoudre des problèmes tels que le positionnement d'objets dans l'espace, ou l'étude des mouvements d'objets les uns par rapport aux autres. En outre, elle peut permettre des implantations particulièrement peu gourmandes en place mémoire dans le cas où plusieurs objets identiques seraient présents dans la même scène. En effet, dans ce cas il suffit de définir les objets avec leurs composants une seule fois. Puis, pour chaque occurrence d'un objet donné dans la scène, on ne précise que les transformations l'ayant amené dans la position qu'il occupe.

La deuxième structure (i.e. "hiérarchie dans l'espace image") s'obtient en décomposant non plus l'objet, mais l'image dans laquelle est enfermé cet objet. Cette décomposition peut s'effectuer de diverses façons : en 2D, on peut par exemple partir d'une image carrée, et effectuer des décompositions successives en régions carrées. Cette organisation est la base de la méthode quadtree. Une telle organisation est généralement plus gourmande en mémoire et elle est moins appropriée que la précédente pour la présentation d'un objet ; mais dans notre cas, parce que nous voulons décomposer l'objet original en éléments et que chaque élément est un objet, à la fin, nous allons arriver à la première présentation.

En utilisant cette méthode pure, la frontière exacte de l'objet ne peut être représentée qu'approximativement. Plus l'objet étudié sera complexe, plus il faudra aller loin dans la décomposition pour obtenir une représentation correcte de celui-ci. Toutefois, cette représentation ne restera jamais qu'une approximation de l'objet réel.

Pour résoudre ce problème, qui est le plus important défaut de cette méthode, il existe une variante de la méthode originale appelée, "la méthode de quadtree modifiée" [2, 3, 20-26]. Dans les pluparts de ces méthodes, on trouve les quadrants coupés par les courbes des frontières. Donc, on arrive aux nombreuses formes d'objets sur les frontières [2-3, 20-26] qui sont difficiles à manipuler et à mailler en éléments finis.

Dans cette thèse, on utilise une autre approche qui consiste en la génération indépendante des quadrants frontières entre les quadrants intérieurs et les courbes de frontières. De cette façon, on résout non seulement le problème d'approximation des courbes de frontière, mais aussi, on réduit les différentes formes d'objet sur les frontières à une simple forme qui est un quadrant, facile à mailler et à manipuler.

Parmi les avantages de cette méthode citons, le fait que les calculs de propriétés géométriques de l'objet (i.e. surface, volume, centre de gravité, ...) sont facilement calculables. Une autre qualité, beaucoup plus importante dans le cas de notre étude, est la possibilité, dans une organisation hiérarchique, de faire varier la résolution (i.e. la profondeur de la décomposition) d'une région à une autre.

## 4-2 Modélisation géométrique de l'objet :

Un bon mailleur d'éléments finis a besoin d'une bonne modélisation géométrique de l'objet, aussi bien au niveau géométrique qu'au niveau topologique. "Le niveau géométrique" veut dire que la géométrie (formes, positions et dimensions) du modèle doit être similaire à celle de l'objet origine. "Le niveau topologique" veut dire que les relations entre les entités définissant le modèle doivent être bien représentées par le modèle.

La modélisation par les courbes B-splines rationnelles non-uniformes composites fermées "NURBS" (voir section 3-3-3-2 et 3-5 du chapitre 3) peut répondre à nos besoins. On peut modéliser n'importe quelle forme en deux dimensions par un nombre limité de courbes NURBS [46,50].

Supposé que les  $V_i = (x_i, y_i)$  soient les points de contrôle dans l'espace à 2 dimensions, alors le  $i^{\text{ème}}$  point de contrôle homogène est défini par :

$$V_i^h = (h_i x_i, h_i y_i, h_i) \quad \text{avec } h_i \neq 0 \quad (4.1)$$

$h_i$  est le poids du point  $i$ .

Alors, la courbe NURBS de degré "d" peut s'écrire sous la forme matricielle suivante :

$$R_i^d(u) = UNH \equiv (x(u), y(u), h(u)) \quad (4.2)$$

avec :

$$U = [1, u, u^2, \dots, u^d]$$

$$N : (d+1) * (d+1)$$

matrice de coefficients

$$H = [V_i^h, V_{i+1}^h, \dots, V_{i+d}^h]^T$$

points homogènes

La matrice  $N$  (matrice de coefficients de la courbe NURBS) pour le cas cubique est donnée à l'équation (3.73).

Normalement, la courbe cubique B-spline composite présente la continuité d'ordre deux ( $C^2$ ) entre les segments adjacents d'une courbe lisse. Toutefois, si on utilise la même coordonnée pour deux points adjacents, on arrive à la continuité d'ordre ( $C^1$ ) sur ce point. Et, si on utilise la même coordonnée pour trois points successifs, on va arriver à la continuité de l'ordre ( $C^0$ ), la continuité de la fonction seulement, i.e. on peut introduire une discontinuité de la dérivée en ce point [51, 52]. De cette façon, on peut modéliser n'importe quelle courbe de frontière lisse par morceaux (piecewise smooth).

La courbe cubique rationnelle composite B-spline fermée est un bon choix pour la modélisation géométrique dans le cas de génération de maillage en deux dimensions. En effet une courbe de frontière de complexité quelconque est représentable par un ou par un nombre limité de segments de courbe cubique rationnelle composite B-spline. En utilisant ce modèle, on a l'avantage d'une représentation unifiée, pour les lignes droites, les courbes quadratiques et les courbes cubiques. Et on peut bénéficier de travailler seulement avec un polynôme de degré trois. De cette façon le problème de la modélisation des frontières mentionné en [16, 28], ne présente pas de problème.

On peut décrire l'objet à modéliser en donnant trois types de points et des listes de numéros des points pour chaque courbe fermée de la frontière. Les trois types de points sont :

1. Les points de la première classe sont les points simples qui sont sur les segments lisses de la courbe de frontière. Il n'y a pas de discontinuité de la dérivée.
2. Les points de la deuxième classe sont les points aux coins. La courbe de frontière présente une discontinuité de la première dérivée, la continuité d'ordre

( $C^0$ ). Ces points doivent rester fixes à leurs positions aussi.

3. Les points de la troisième classe sont les points à l'intérieur de l'objet. Ces points sont utilisés pour définir le paramètre de contrôle de maillage en ces positions. Si, on veut un raffinage local dans un endroit de l'objet, on utilise ce type de points pour un contrôle local de la finesse du maillage.

Les points de deuxième classe doivent être triplés avant leur utilisation dans la représentation B-spline cubique. Comme la troisième classe, la première et la deuxième classe doivent avoir le paramètre de contrôle de maillage. On peut trouver les paramètres de contrôle de maillage pour chaque point sur la frontière, en utilisant l'approximation B-spline ou une autre méthode d'approximation des points de contrôle sur les frontières.

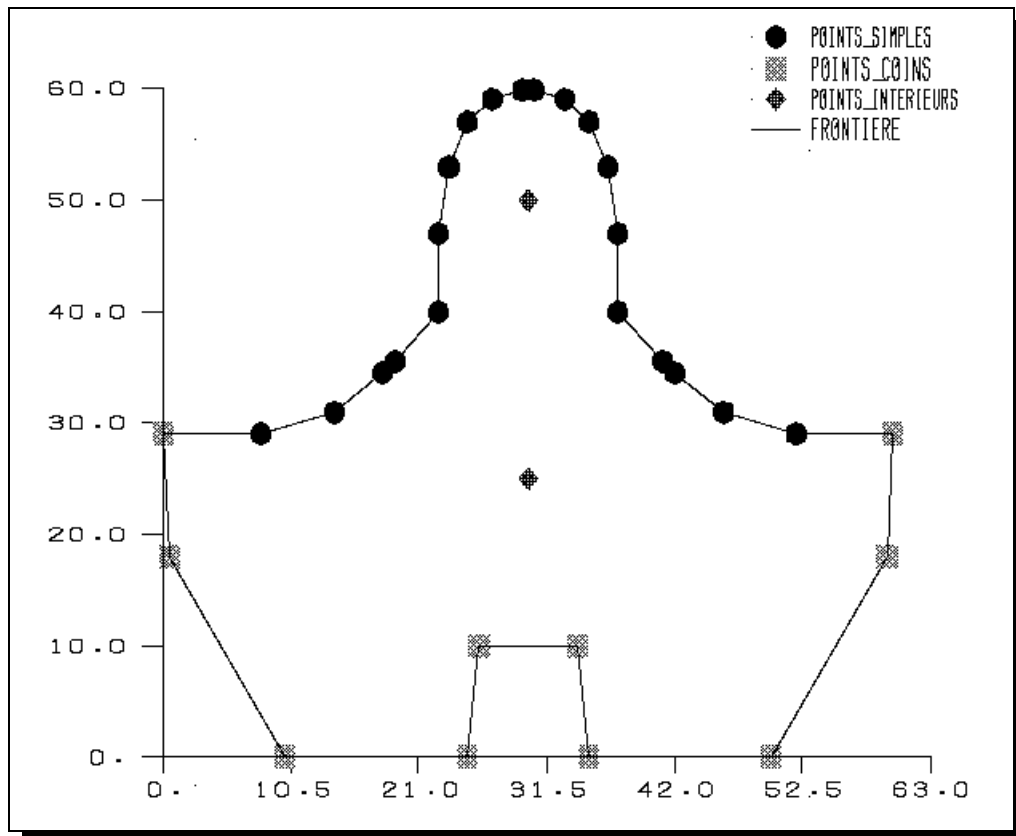


Figure 4-1 : Points donnés pour l'exemple d'une dent d'engrenage.

La figure (4-1) présente les points utilisés dans l'exemple d'une dent d'engrenage. Cet exemple est utilisé dans toutes les sections de ce chapitre. L'algorithme suivant présente les étapes à suivre pour arriver à la représentation par la courbe cubique B-spline composite fermée :

### 1- Tripler les points aux coins.

### 2- Trouver les espacements nodaux $\{\nabla_i\}$ :

- a. *Intervalles de support* : Les intervalles nodaux supportant la courbe peuvent être approximés raisonnablement par la méthode des longueurs des cordes entre points successifs i.e. :

$$\nabla_i = | P_{i+1} - P_i | , i = 0, 1, \dots, n-1.$$

- b. *Intervalles auxiliaires* : Les intervalles auxiliaires doivent être choisis de la façon suivante :

$$\nabla_{-1} = \nabla_{n-2}, \nabla_{-2} = \nabla_{n-3}, \nabla_{n+1} = \nabla_2, \nabla_n = \nabla_1$$

- c. *Intervalles entre les points triples* : Les intervalles entre les points triples sont nuls mais pour éviter la division par zéro dans les calculs, on choisit une valeur petite mais assez grande pour éviter la division par zéro.

### 3- Construction des systèmes linéaires pour calculer les points de contrôle inconnus :

(voir la section 3-3-3-2 du chapitre 3)

### 4- Construction de la matrice de système (équation 3.33) :

(voir la section 3-3-3-2 du chapitre 3)

### 5- Résolution du système d'équations pour $\{V_i\}$ :

Le système tri-diagonal cyclique doit être résolu par un algorithme qui

permet l'échange entre les colonnes et les rangées car la matrice contient des valeurs nulles sur la diagonale.

La Figure (4-2) représente la courbe frontière obtenue avec une représentation B-spline de la dent d'engrenage.

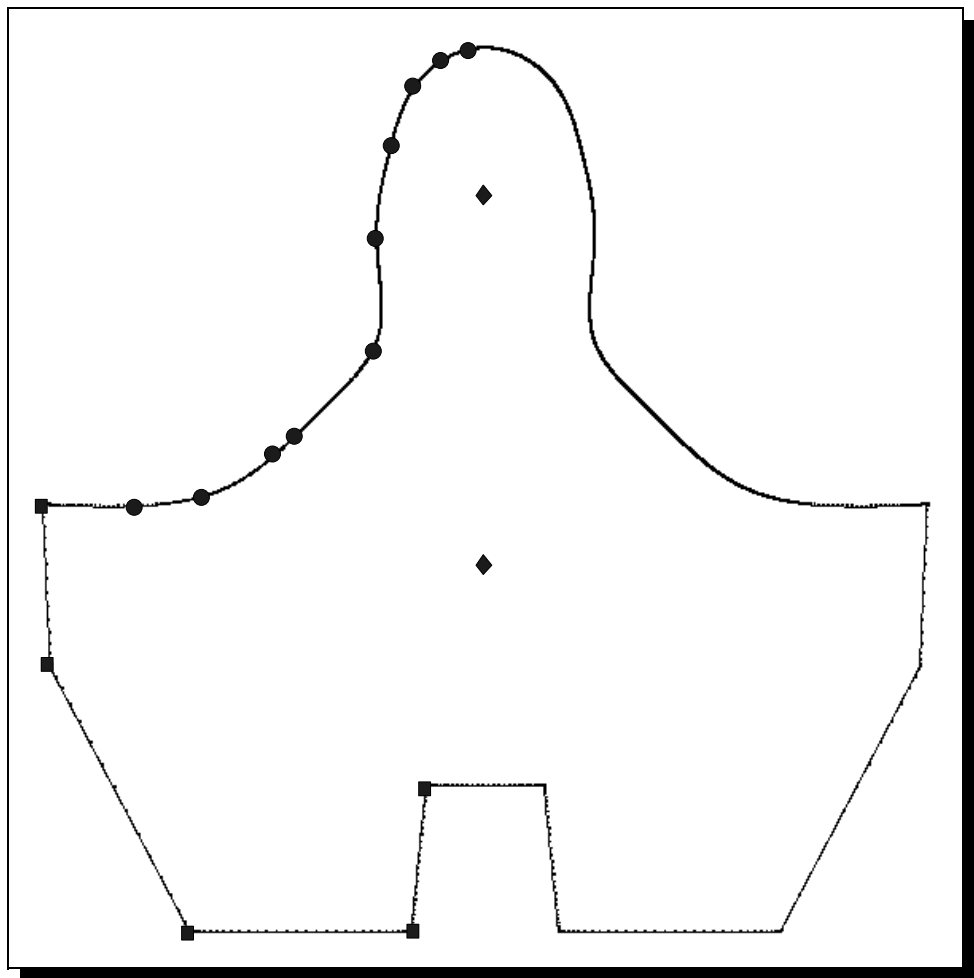


Figure 4-2 : représentation B-spline de la dent d'engrenage.

### 4-3 Création des quadrants et de l'arbre quadtree :

Il s'agit d'un cas particulier d'un modèle hiérarchique dans l'espace image. La méthode des quadtree permet donc de représenter des domaines 2D par une structure arborescente.

L'objet à décomposer est d'abord enfermé dans un carré qui est immédiatement subdivisé en quatre quadrants. Chacun de ces quadrants est alors testé pour voir s'il se trouve à l'intérieur de l'objet, à l'extérieur de celui-ci, ou bien s'il est partiellement à l'intérieur et partiellement à l'extérieur. Seuls ces derniers seront de nouveau subdivisés en quatre sous-quadrants. Le processus précédent est répété jusqu'à ce qu'on atteigne les niveaux de décomposition désirés (figure 4-3).

Puisque la méthode des quadtree effectue la discrétisation d'un domaine à partir d'un ensemble de carrés, on pourrait être tenté d'utiliser directement cette décomposition pour mener l'analyse par éléments finis. Toutefois, si on se reporte aux conditions d'un bon maillage (voir la section 1-4-1 propriétés essentielles d'un maillage), on constate que le schéma obtenu est susceptible de présenter les défauts suivants :

1. L'intérieur du domaine pourrait être représenté par un petit nombre de gros éléments [2].
2. Des quadrants voisins peuvent se trouver à des niveaux de subdivision différents [2].

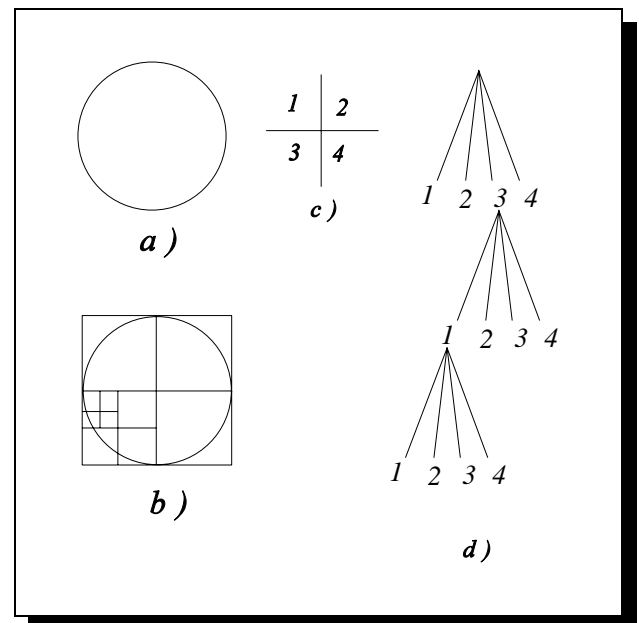


Figure 4-3 : Représentation d'un cercle par quadtree. a) L'objet à décomposer, b) Décomposition effectuée, c) numérotation des sous-quadrants, d) Arborescence correspondante.

Des traitements particuliers devront être envisagés pour les coins de quadrants se trouvant le long d'un côté d'un quadrant voisin afin d'assurer la continuité entre éléments finis.

3. Le suivi de la frontière réelle est mauvais en particulier là où la frontière n'est ni verticale, ni horizontale ; dans ces zones, la frontière sera représentée par des marches d'escalier susceptibles de fausser les résultats de l'analyse [3] (figure 4-4).

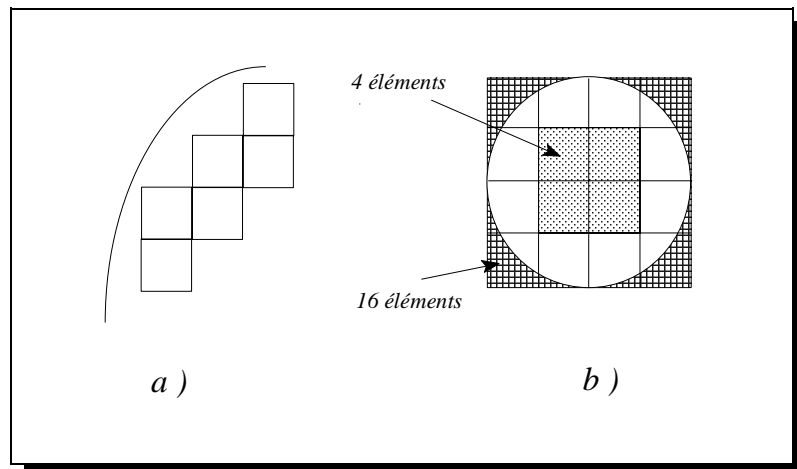


Figure 4-4 : a) Problèmes d'approximation de frontières par les quadtree, b) Avec ce niveau de décomposition, le cercle est représenté soit par 4 éléments ou par 16 éléments.

Les deux premiers points pourraient être réglés en imposant des contraintes sur les niveaux minimaux de différence de maillage entre les quadrants voisins, et en créant des éléments finis capables d'assurer la transition entre éléments de niveaux de subdivision différents. Le troisième point exige, lui, une refonte très importante du principe de traitement des éléments frontières. Dans cette thèse, nous générons les quadrants frontières entre les côtés libres des quadrants intérieurs et les courbes frontières.

Dans le cas de la génération des quadrants frontières entre les côtés libres des quadrants intérieurs et les courbes frontières, on doit avoir les quadrants intérieurs voisins des quadrants partiels presque avec le même niveau de subdivision que les quadrants coupés. Dans cette situation le processus qui est présenté ci-haut n'est pas assez efficace. Donc, on utilise une autre méthode laquelle est une modification de l'approche précédente.

Dans cette nouvelle approche, le premier quadrant est construit en trouvant d'abord un carré qui renferme l'objet. Ce carré est testé et subdivisé en quatre quadrants de la façon suivante :

- a. Si le paramètre «maximum de contrôle du maillage» dans le quadrant est plus grand que le niveau actuel du quadrant plus un, ce quadrant sera subdivisé en quatre quadrants et chacun de ses quadrants est testé de la même façon.
- b. Si le paramètre «maximum de contrôle du maillage» dans le quadrant est égal ou inférieur au niveau actuel du quadrant plus un, le quadrant sera subdivisé et le processus se terminera.

On peut présenter cette procédure sous forme d'algorithme comme suit :

**Algorithme : Tester\_subdiviser\_quadrant (quadrant, niveau)**

**Début**

**Si** Max\_paramètre\_de\_contrôle > ( niveau + 1 ) **alors**

**Diviser\_quadrant\_en\_quatre ;**

**Pour** i = 1 **à** 4 **faire**

**Tester\_subdiviser\_quadrant (quadrant(i), niveau+1)**

**Finfaire;**

**Sinon**

**Diviser\_quadrant\_en\_quatre;**

**Finsi;**

**Fin;**

Les résultats de cet algorithme sont similaires à ceux de l'ancienne méthode parce que les paramètres de contrôle du maillage se trouvent normalement sur les frontières, mais cette méthode génère des quadrants plus uniformes proches des courbes frontières, c'est-à-dire les quadrants voisins des quadrants partiels sont au même niveau de subdivision que les points de contrôle de maillage sur les frontières.

La figure (4-5) présente la génération des quadrants pour l'exemple d'une dent d'engrenage.

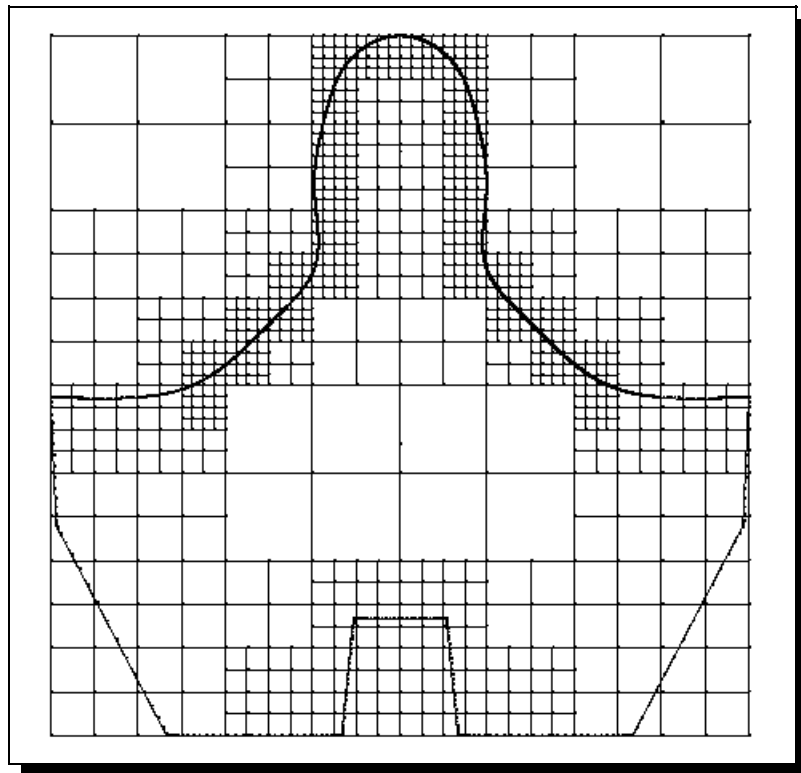


Figure 4-5 : Génération des quadrants pour l'exemple d'une dent d'engrenage.

#### 4-4 Un niveau de différence :

Dans la section précédente, on a parlé du problème de génération d'un petit nombre de gros éléments à l'intérieur du domaine et aussi de la possibilité d'avoir des quadrants voisins ayant des différences de niveaux de subdivision élevées. Afin de régler ces problèmes et d'assurer la continuité entre éléments finis, on peut imposer la contrainte d'*un seul niveau de différence* sur les niveaux minimaux de différence de maillage entre les quadrants voisins. Donc, on s'assure que deux quadrants voisins n'ont pas plus d'un niveau de subdivision de différence ; si nécessaire, on effectue des subdivisions supplémentaires pour que cette condition soit vérifiée. De cette manière, on a le passage lisse entre les quadrants voisins et on peut créer des éléments finis capables d'assurer la transition entre des niveaux de subdivision différents.

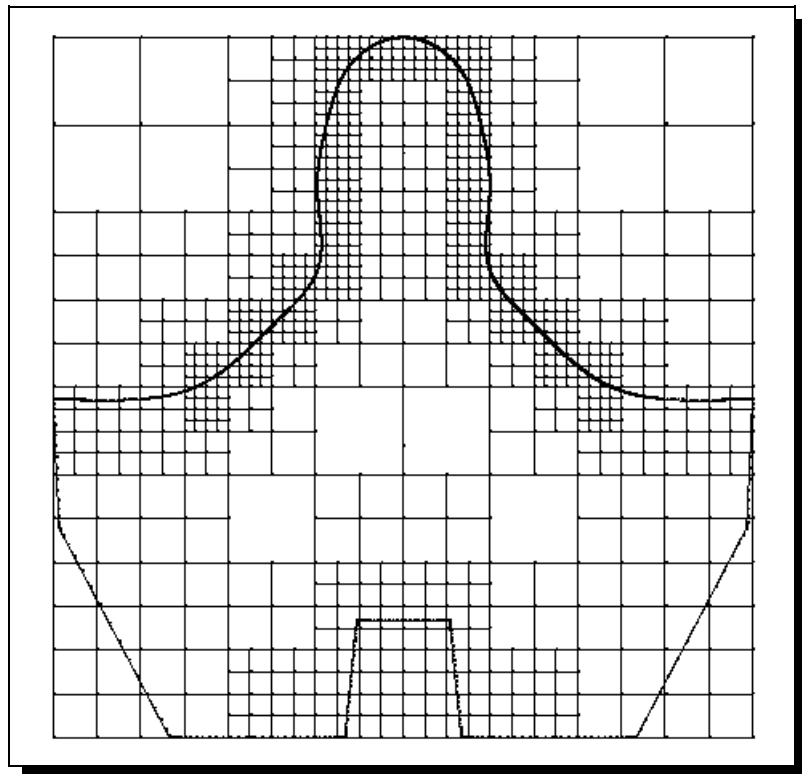


Figure 4-6 : Quadrants après l'étape d'un niveau de subdivision de différence.

L'algorithme suivant présente les étapes principales d'une subdivision avec un seul niveau de différence :

- 1. Construire la liste chaînée des quadrants feuilles de l'arbre quadtree.**
- 2. Trouver les voisinages de chaque quadrant.**
- 3. Pour tous les quadrants dans la liste chaînée :**
  - a. Si le niveau de subdivision du quadrant est plus petit que le niveau des quadrants voisins plus un alors :**
    - i. Subdiviser le quadrant en quatre quadrants**
    - ii. Modifier la liste chaînée des quadrants feuilles**
      - (1) Enlever le quadrant père**
      - (2) Ajouter les quadrants fils dans la liste chaînée**
    - iii. Mise à jour des voisinages**

La figure (4-6) présente les quadrants générés après l'étape d'imposition d'un niveau de subdivision de différence pour l'exemple d'une dent d'engrenage.

#### **4-5 Classification des quadrants :**

On peut trouver les quadrants partiels soit par la méthode présentée dans le chapitre 3 section (3-6-3), ou soit par une méthode de vérification de l'intersection des quadrants avec les courbes frontière. Tous les quadrants ayant une intersection avec les courbes frontières sont nommés quadrants partiels. On peut ensuite distinguer les quadrants faisant partie de l'intérieur du contour de ceux qui sont à l'extérieur. La méthode que nous avons développée à cette fin a été poétiquement baptisée "méthode du ballon gonflable".

Partant d'un quadrant quelconque mais non partiel, on peut progresser à la façon d'une souris dans un labyrinthe, si possible à gauche, sinon en avant, autrement

à droite ou en arrière. Les obstacles sont les quadrants partiels de contours extérieurs ou intérieurs, et les limites du carré racine du quadtree original. On utilisera pendant cette phase les relations de voisinage calculées auparavant. Lorsqu'on ne peut plus avancer (la progression s'est faite récursivement), on regarde la nature des différents obstacles rencontrés pendant la progression. Ceci nous permet de déterminer le type des quadrants par lesquels nous venons de passer. Par exemple, si on n'a rencontré que des quadrants partiels de contour intérieur, les quadrants sont extérieurs à l'objet : on vient de "gonfler le ballon" dans un des trous de l'objet. Par contre, si nous avons rencontré à la fois des quadrants partiels de contours intérieur et extérieur, nos quadrants parcourus sont intérieurs. Et si on a rencontré des quadrants partiels de contour extérieur et les limites du carré racine du quadtree original, les quadrants sont extérieurs à l'objet.

On peut présenter cette étape sous la forme de l'algorithme suivant :

### **Algorithme de classification des quadrants :**

#### **1. Marquer tous les quadrants partiels.**

#### **2. Pour tous les quadrants qui ne sont pas marqués faire :**

**a. Liste\_quadrants\_inconnus = NUL;**

**b. test\_int = 0;**

**c. test\_ext = 0;**

**d. test\_limit = 0;**

**e. Tester Quadrant**

**f. Si test\_int == 1 et test\_ext == 0 marquer tous les quadrants dans la liste "Liste\_quadrants\_inconnus" à quadrant extérieur;**

**g. Si test\_int == 1 et test\_ext == 1 marquer tous les quadrants dans la liste "Liste\_quadrants\_inconnus" à quadrant intérieur;**

**h. Si test\_ext == 1 et test\_limit == 1 marquer tous les quadrants dans la liste "Liste\_quadrants\_inconnus" à quadrant extérieur;**

**i. Autrement arrêter avec un message erreur.**

**Tester Quadrant :**

- 1. Marquer le quadrant inconnu et ajouter dans la liste "Liste\_quadrants\_inconnus" .**
- 2. Si le quadrant a un quadrant voisin partiel de contour intérieur test\_int = 1;**
- 3. Si le quadrant a un quadrant voisin partiel de contour extérieur test\_ext = 1;**
- 4. Si le quadrant a un voisin de type limite test\_limit = 1;**
- 5. Pour tous les quadrants voisins de ce quadrant qui ne sont pas marqués faire:**
  - a. Tester Quadrant**

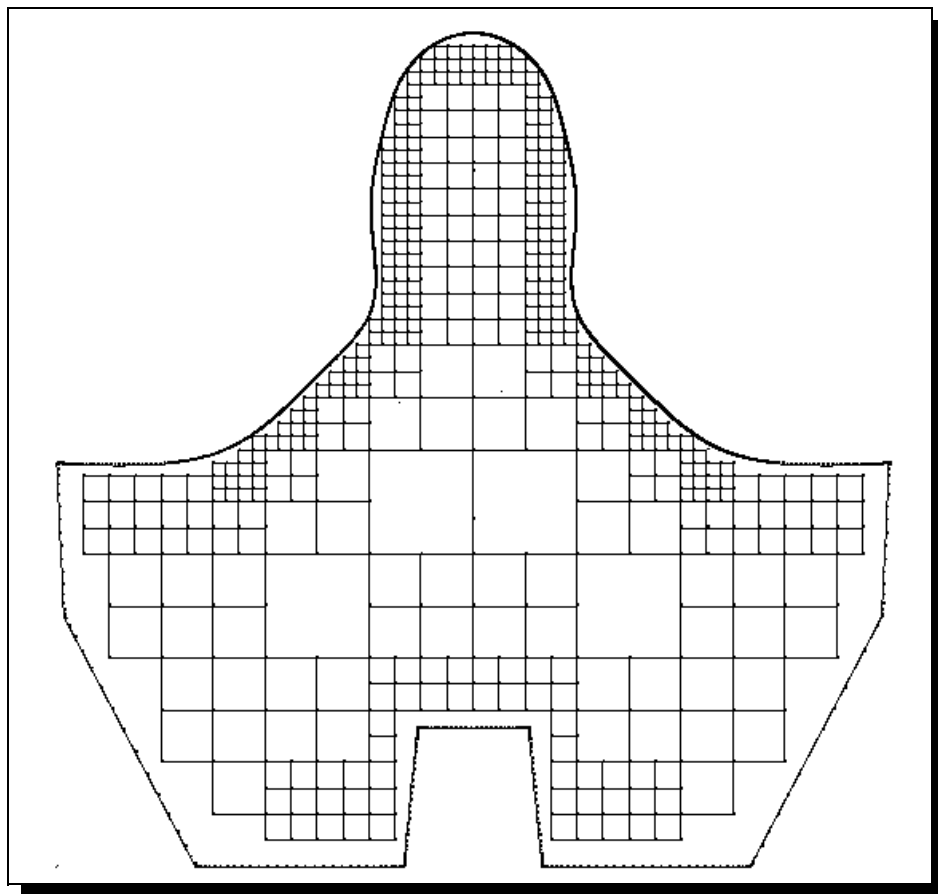


Figure 4-7 : Quadrants intérieurs pour l'exemple d'une dent d'engrenage.

Dans les étapes suivantes, on n'a pas besoin de quadrants partiels et extérieurs, donc on peut les éliminer de l'arbre quadtree. La figure (4-7) présente les quadrants intérieurs pour l'exemple d'une dent d'engrenage.

#### 4-6 Génération des quadrants frontières :

Une des étapes la plus délicate dans le processus de décomposition de l'objet est la représentation de la frontière. Dans l'ancienne méthode [2,3], les quadrants coupés sont construits à partir des intersections entre les courbes frontières et les quadrants partiels (figure 4-8).

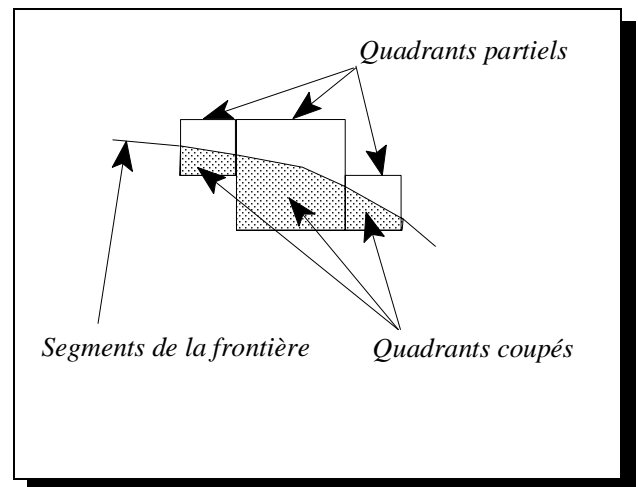


Figure 4-8 : Formation des quadrants coupés.

Toutefois, ceci introduit les difficultés suivantes :

- ◆ Certains éléments frontières peuvent désormais être constitués de segments démesurément petits.

Ceci peut par exemple se produire dans le cas où le contour coupe un quadrant frontière à proximité d'un de ces coins (figure 4-9 a). Ces petits segments sont généralement indésirables car ils sont souvent à l'origine d'éléments particulièrement déformés, ou

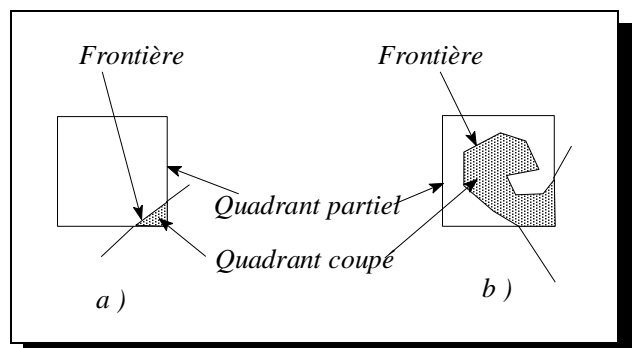


Figure 4-9 : Problèmes dus au suivi du contour réel.

beaucoup plus petits que leurs voisins.

- ◆ Certains éléments frontières peuvent être rendus géométriquement complexes (figure 4-9 b).

De ce fait, on ne pourra pas utiliser des gabarits pré-définis pour introduire les éléments finis dans les quadrants frontières (contrairement à ce qui sera fait pour les quadrants intérieurs). Il faudra développer des algorithmes spécifiques.

- ◆ Il existe de nombreux cas de quadrants coupés (coupé à un, deux, trois ou quatre côté(s) et par un, deux, ... segment(s) et etc.).

Pour résoudre ces problèmes, nous présentons une nouvelle approche de formation des quadrants frontières.

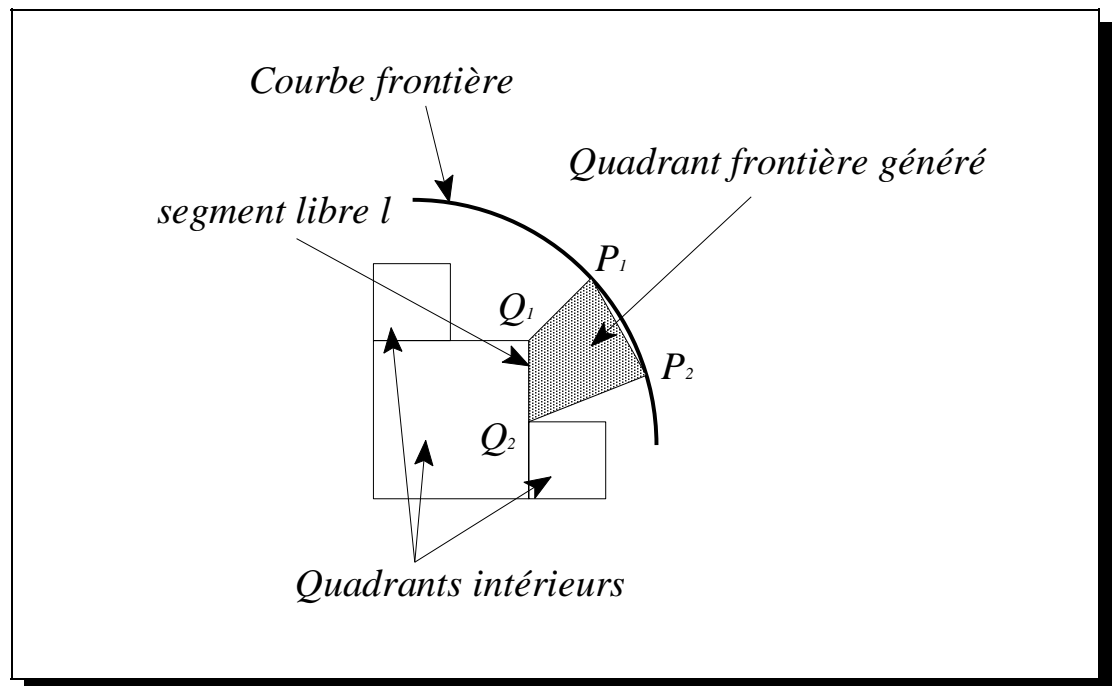


Figure 4-10 : Génération des quadrants frontières entre le segment libre de quadrant intérieur et la courbe frontière.

Au lieu de modifier les quadrants partiels pour arriver aux quadrants frontières, on peut générer les quadrants frontières en utilisant les informations emmagasinées

dans les quadrants intérieurs et les courbes frontières. Dans la première étape, on génère les quadrants frontières entre les côtés libres des quadrants intérieurs et les courbes frontières.

Supposons que les points  $Q_1$  et  $Q_2$  soient des points (sommets) sur le segment libre "l" d'un quadrant (figure 4-10). Correspondant aux points  $Q_1$  et  $Q_2$  on trouve sur la courbe frontière des points  $P_1$  et  $P_2$  qui sont à une distance minimum des points  $Q_1$  et  $Q_2$  respectivement. Les segments  $(Q_1, P_1)$  et  $(Q_2, P_2)$  n'intersectent aucun quadrant, le nouveau quadrant frontière est défini par les segments : l,  $(Q_1, P_1)$ ,  $(P_1, P_2)$  et  $(Q_2, P_2)$ .

Pour trouver le point sur une courbe, situé à une distance minimum d'un point donné, on peut faire la projection du point sur la courbe (voir la section 3-6-1 du chapitre 3). La figure (4-11) présente les quadrants générés après la première étape de génération des quadrants frontières pour l'exemple de la dent d'engrenage.

Il est possible que le segment définissant la distance minimum coupe un des quadrants existants, si c'est le cas, on doit sélectionner un autre vecteur normal. Si tous les vecteurs normaux passent aux quadrants existants, on doit trouver le point situé à une distance minimum par une autre méthode, par exemple, on peut commencer par un point trouvé en haut et on redéplace d'un petit pas sur la courbe de frontière, étape par étape en essayant de trouver un point qui ne coupe pas les autres quadrants. Il est aussi possible que les points  $Q_1$ ,  $Q_2$ ,  $P_1$  et  $P_2$  soient colinéaires ; si c'est le cas, nous avons des quadrants avec l'aire égale à zéro. Nous pouvons éliminer ce problème dans cette étape en bougeant un peu le point  $P_1$  ou  $P_2$ , ou, nous pouvons laisser cette situation se résoudre automatiquement dans l'étape de lissage intérieur qui déplacera les points  $Q_1$  et  $Q_2$ . Si le point  $P_1$  coïncide avec le point  $P_2$ , on a un triangle à la place d'un quadrant. Cette situation doit être considérée au moment de la génération du maillage à partir des quadrants.

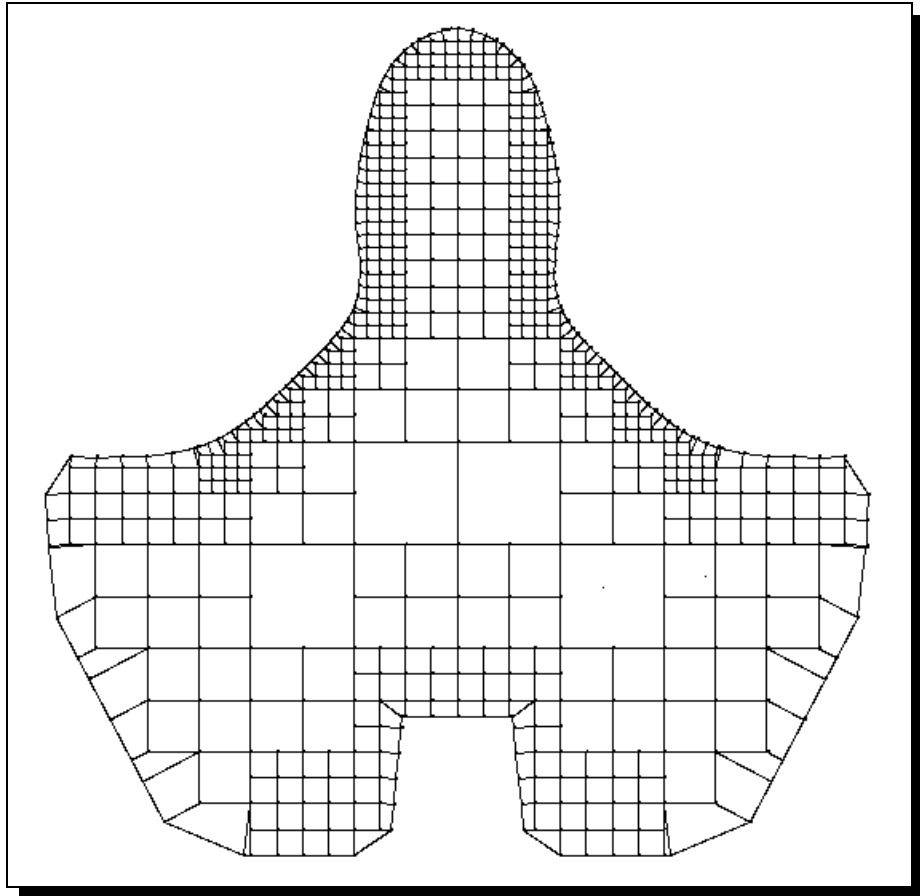


Figure 4-11 : Quadrants intérieurs et frontières après la première étape de génération des quadrants frontières pour l'exemple de la dent d'engrenage.

Il est évident, à partir de la figure (4-11), que les coins de l'objet ne sont pas bien représentés dans cette forme intermédiaire. Pour résoudre le problème de similarité géométrique de l'objet, il est nécessaire que l'on passe à la deuxième étape ou à l'étape d'ajustement des coins de l'objet. Dans cette étape, les points des quadrants générés sur les frontières qui sont proches des coins de l'objet sont tirés vers ces coins. La figure (4-12) présente les quadrants générés après la deuxième étape de génération des quadrants frontières pour l'exemple de la dent d'engrenage.

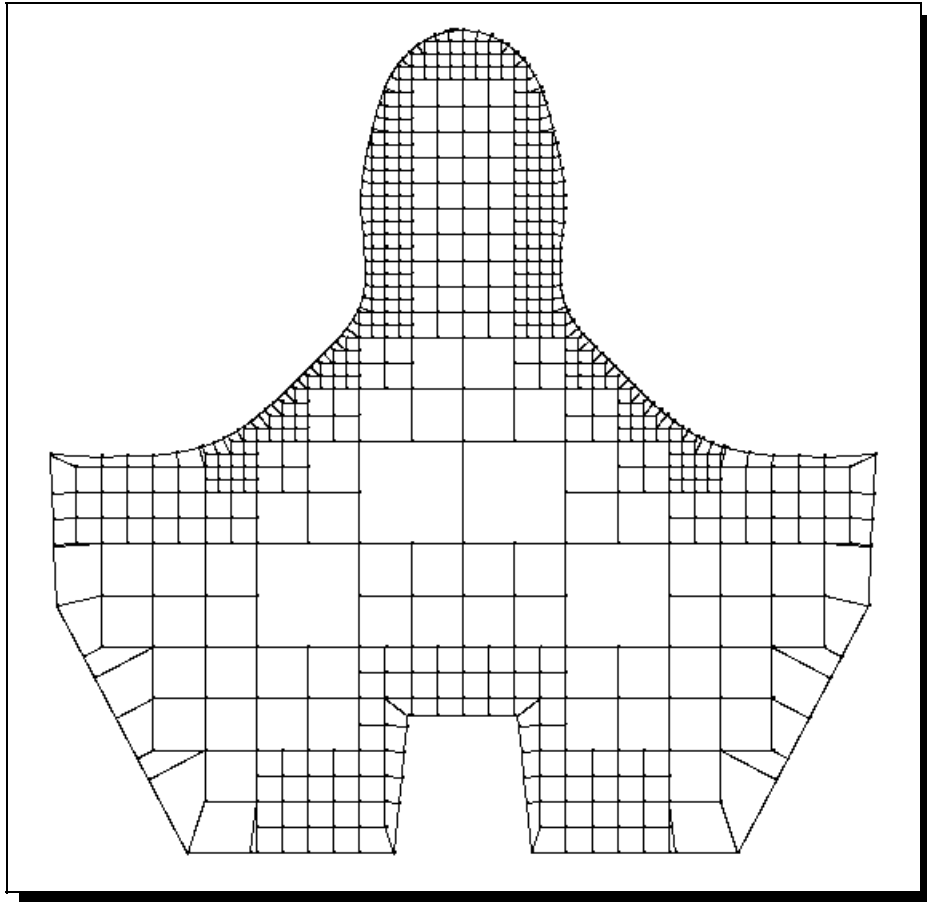


Figure 4-12 : Quadrants intérieurs et frontières après ajustement des coins pour l'exemple de la dent d'engrenage.

On peut présenter cette étape sous la forme de l'algorithme suivant :

**Algorithme de génération des quadrants frontières :**

**1. Pour tous les quadrants intérieurs à côté des quadrants partiels faire :**

**a. Pour tous les segments libres (I) du quadrant faire :**

**i. Pour les deux points d'extrémité du segment (Q) faire :**

**(1) Trouver le point (P) sur les courbes frontières de distance minimum avec le point Q.**

(2) Vérifier si le segment QP ne coupe pas les quadrants existants.

(3) Si segment QP coupe un quadrant et il existe un autre point de projection, essayer un autre. Aller à l'étape (2).

(4) S'il n'existe pas un autre point de projection, prendre le point le plus proche, se déplacer pas à pas sur la courbe frontière pour trouver le point (P) qui ne coupe pas les autres quadrants.

ii. Générer le nouveau quadrant à partir des segments I,  $(Q_1, P_1)$ ,  $(P_1, P_2)$  et  $(Q_2, P_2)$ .

## 2. Ajuster les coins.

### 4-7 Lissage intérieur :

Afin d'optimiser la forme des éléments qui vont être introduits, on va également effectuer un lissage des éléments intérieurs. Chacun des noeuds intérieurs va être déplacé conformément à un opérateur de lissage laplacien pondéré. Pour un noeud donné A, on va le placer au centre de gravité des noeuds des quadrants en contact avec A. L'opérateur est dit pondéré car les noeuds directement reliés à A reçoivent le poids relatif 2, alors que les autres n'ont qu'un poids relatif de 1. A sera donc déplacé en B comme suit :

$$B = \frac{\sum_{i=1}^N P_i + 2 \sum_{j=1}^M Q_j}{N + 2M} \quad (4.3)$$

où :

$P_i$  sont les coordonnées des noeuds  $i$  qui se trouvent dans un des quadrants en

contact avec A mais  $P_i$  ne sont pas directement reliés à A.

$Q_i$  sont les coordonnées des noeuds  $i$  qui se trouvent dans un des quadrants en contact avec A et  $Q_i$  directement reliés à A.

N est le nombre de sommets des quadrants en contact avec A, mais non directement reliés à A.

M est le nombre de sommets des quadrants en contact avec A, et directement reliés à A.

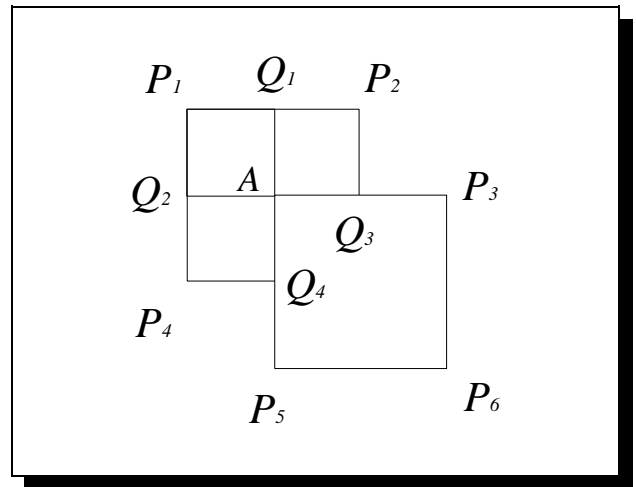


Figure 4-13 : Point A et les points reliés à A.

Par exemple, dans le cas de la figure (4-13), nous avons  $N=6$  et  $M=4$ .

D'autres expressions d'opérateurs sont proposées par Hermann [53].

Ce schéma est évidemment itératif. Baehmann [18] indique que quatre itérations permettent d'obtenir une convergence acceptable. La figure (4-14) présente les quadrants générés après la phase de lissage intérieur pour l'exemple de la dent d'engrenage.

On peut présenter cette étape sous la forme de l'algorithme suivant :

### Algorithme : lissage intérieur

1. Pour  $i = 1$  à 4 faire :

a. pour tous les quadrants intérieurs faire :

i. Pour tous les points (A) de quadrant qui ne sont pas encore modifiés dans le cycle  $i$  faire :

(1) Trouver les points directement reliés à A.

(2) Trouver les points reliés, mais pas directement reliés à A.

(3) Utiliser l'équation (4.3) et modifier les coordonnées du point.

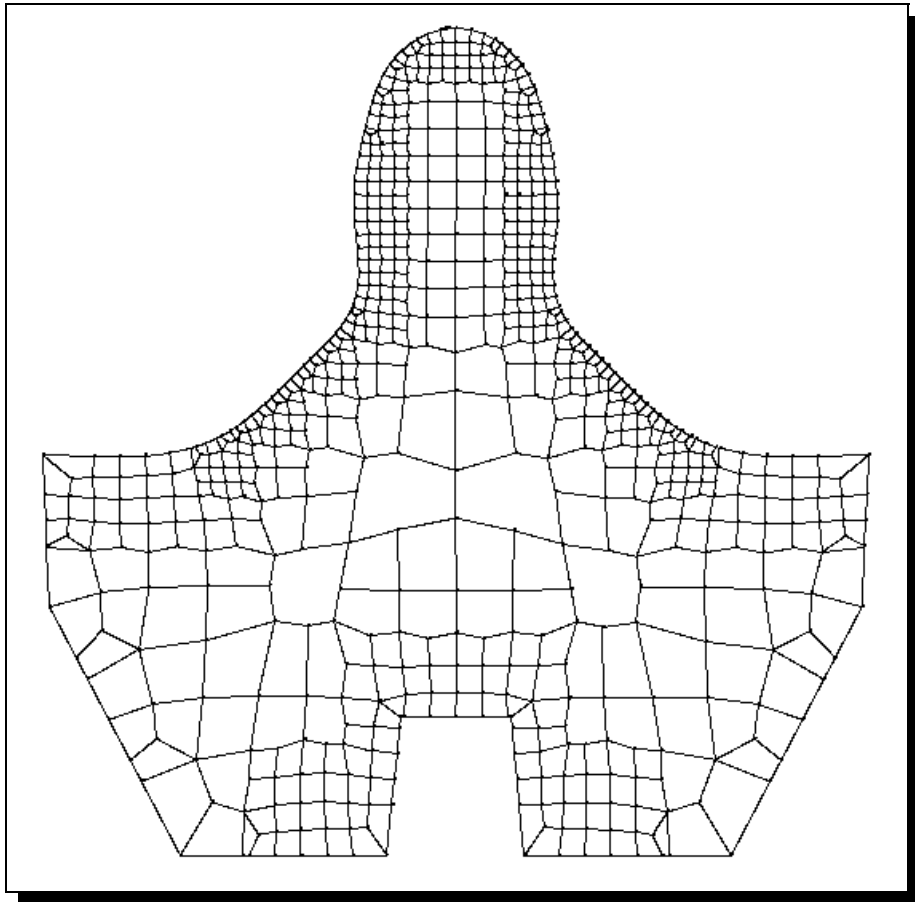


Figure 4-14 : Quadrants générés après la phase de lissage intérieur pour l'exemple de la dent d'engrenage.

#### **4- 8 Génération du maillage en éléments finis :**

On peut passer maintenant à l'introduction des éléments finis dans les quadrants. Par rapport aux différents logiciels d'éléments finis et à leurs capacités d'analyser des éléments conformes ou non-conformes (conforming, nonconforming) et à la possibilité d'utilisation d'estimateurs d'erreur, différents maillages d'éléments finis doivent être générés.

Par exemple, les résultats de l'étape précédente peuvent être utilisés comme

un maillage valide d'éléments finis. Mais en général, on peut utiliser une banque de patrons de maillage ("template") sur un quadrant de référence pour construire les éléments finis désirés. Les différents patrons de maillage doivent prendre en compte les différentes configurations des quadrants relativement au nombre de noeuds qui peuvent se trouver sur leur contour ; en plus des quatre coins d'origine, il faudra peut-être, lors du lissage des éléments intérieurs, introduire des noeuds le long des côtés à cause du voisinage avec des quadrants plus subdivisés. Nous allons parcourir le contour de chaque quadrant afin de relever la configuration de ses coins. Par exemple, celle-ci peut être codée par une suite de 0 et de 1, les 0 caractérisant des coins correspondant aux coins "naturels" du quadrant initial, les 1 désignant les coins intermédiaires rajoutés pendant le lissage à cause du voisinage avec des quadrants plus subdivisés.

La séquence obtenue va nous permettre de choisir le patron de maillage correct parmi les patrons de référence.

La figure (4-15) présente les six patrons de maillage utilisés pour construire les éléments triangulaires et la figure (4-16) présente les patrons de maillage utilisés pour construire les éléments quadrilatéral et triangulaire.

Ces patrons de maillage sont ensuite déformés afin d'être projetés sur chacun des quadrants.

Dans le cas où le quadrant frontière est dégénéré et représente un triangle, celui-ci peut être considéré comme un élément fini, sans besoin d'autres divisions.

La figure (4-17a) présente le maillage triangulaire final généré pour l'exemple de la dent d'engrenage et la figure (4-17b) présente le maillage quadrilatère/triangulaire final généré pour l'exemple de la dent d'engrenage.

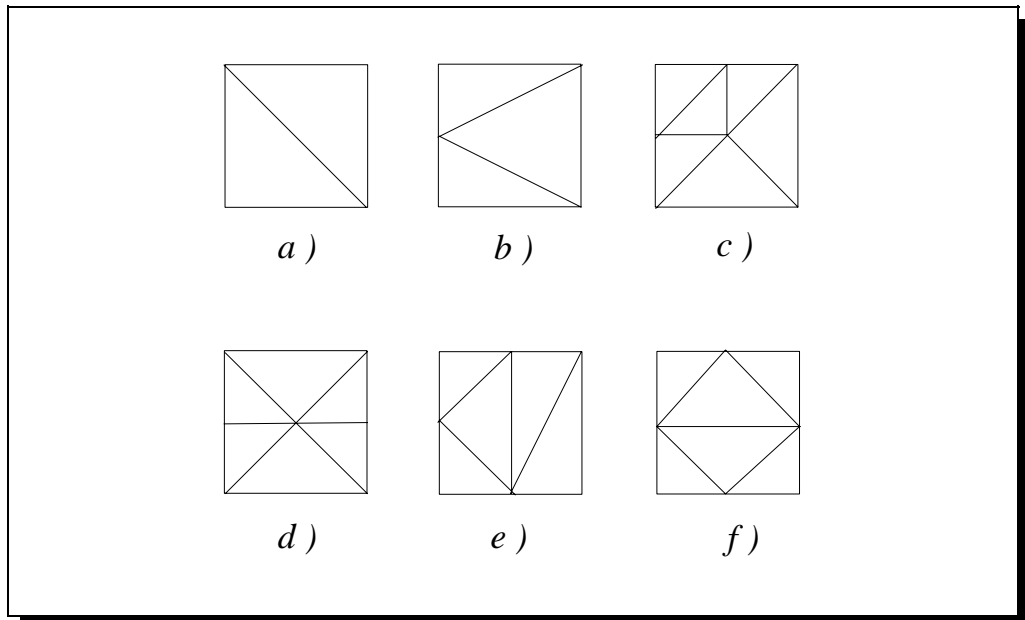


Figure 4-15 : Différents patrons de maillage en éléments triangulaires.  
 a) séquence 0000, b) séquence 10000, c) séquence 100010 ou 101000, d) séquence 100100, e) séquence 1010010 ou 1001010 ou 1010100, f) séquence 10101010.

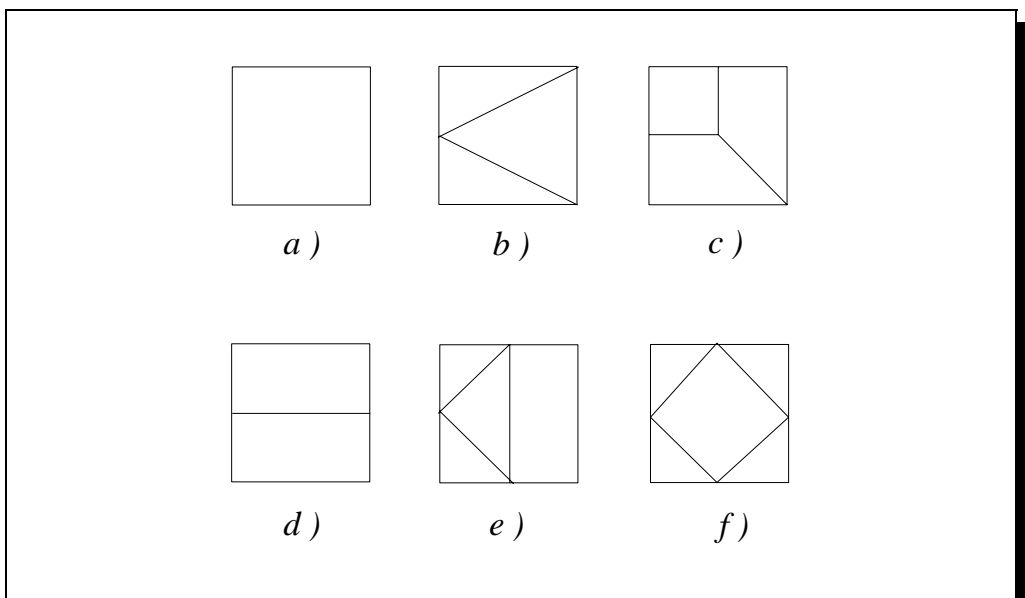


Figure 4-16 : Différents patrons de maillage en éléments quadrilatéraux et triangulaires. a) séquence 0000, b) séquence 10000, c) séquence 100010 ou 101000, d) séquence 100100, e) séquence 1010010 ou 1001010 ou 1010100, f) séquence 10101010.

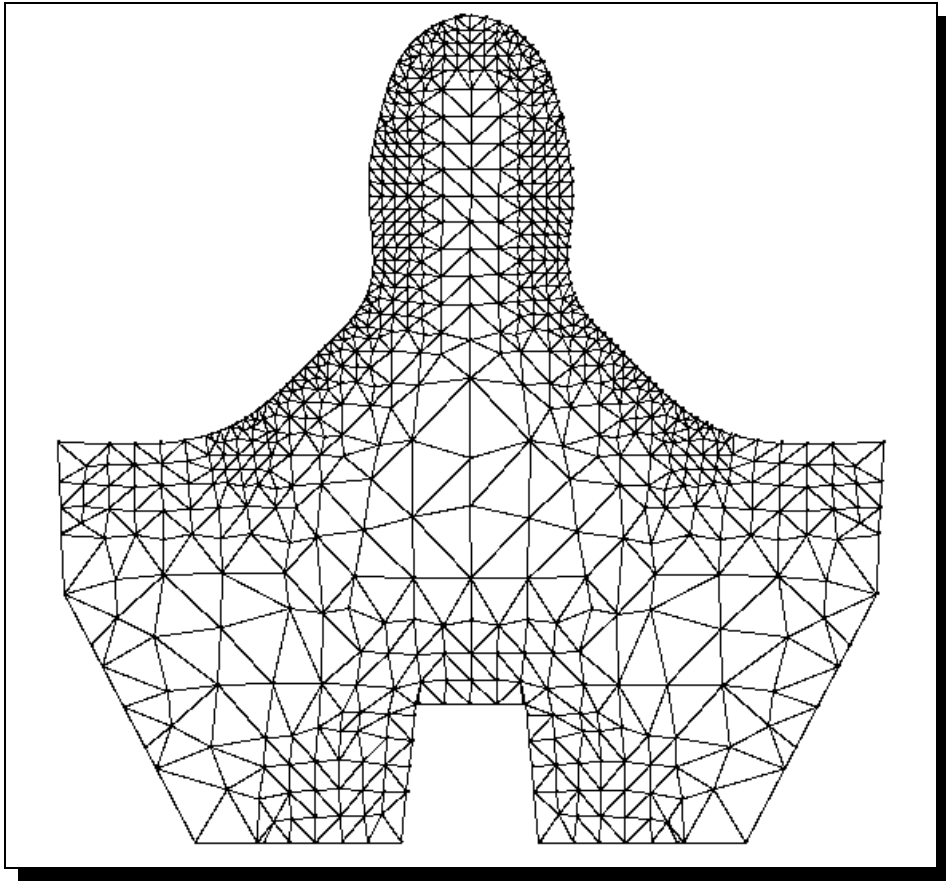


Figure 4-17a : Maillage triangulaire final pour l'exemple d'une dent d'engrenage.

On peut présenter l'algorithme général à cette étape sous la forme suivante :

### **Algorithme de génération de maillage à partir des quadrants générés**

**1. Pour Tous les quadrants faire :**

**a. Si le quadrant est dégénéré et correspond à un triangle**

**i. Le triangle est un élément fini.**

**Sinon**

- i. Trouver la séquence de quadrant**
- ii. Sélectionner le patron de maillage.**
- iii. Projeter le patron sur le quadrant et trouver les éléments finis.**

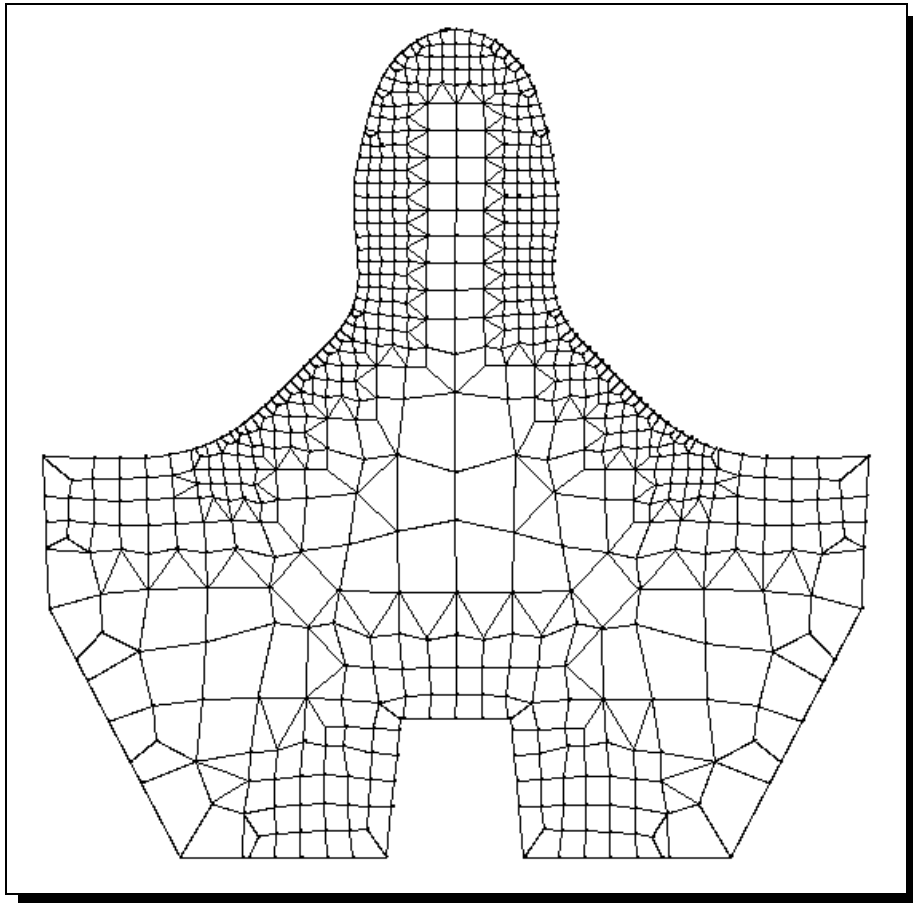


Figure 4-17b : Maillage quadrilatère/triangulaire final pour l'exemple d'une dent d'engrenage.

À partir du modèle géométrique, on peut générer les éléments finis de frontière. Ici on subdivise chaque segment des courbes B-spline de frontière par rapport aux paramètres de maillage des points sur le segment. Les éléments finis de frontière sont construits en utilisant les points successifs trouvés sur le segment.

#### **4-9 Sollicitations et conditions aux limites :**

Après la génération de maillage de chaque quadrant, on doit construire la liste de tous les éléments générés et la liste de tous les points ou noeuds identifiant les éléments. Chaque point dans l'espace à deux dimensions doit avoir un numéro unique pour éviter la discontinuité entre les éléments. La numérotation des points (noeuds) est importante dans la phase d'analyse par éléments finis ; cette numérotation permet d'identifier la largeur de bande de la matrice d'assemblage. Pour améliorer la numérotation des points, on peut utiliser un algorithme d'optimisation des numéros des noeuds tel que présenté dans [33, 34, 59].

Dans la phase de génération des quadrants frontières, on crée les quadrants frontières ; donc on peut distinguer les éléments frontières des autres éléments générés, en faisant la référence aux quadrants père. À partir du quadrant frontière, on peut trouver les courbes frontières à côté de ces quadrants et donc, on peut transférer les sollicitations et les conditions sur la courbe aux éléments finis générés sur les frontières. Seulement, on doit réserver l'espace dans les structures de données pour chaque élément afin d'emmagasiner les pressions, températures, forces ou autres conditions aux limites.

#### **4-10 Programmation de MESH2D :**

À partir des algorithmes décrits dans les sections précédentes, nous avons développé un système informatique mettant en oeuvre ces idées. Le logiciel obtenu a été baptisé MESH2D.

#### 4-10-1 Algorithme général :

L'algorithme général du programme peut se présenter comme suit :

1. **Entrée des données géométriques et les paramètres de maillage.**
2. **Génération du modèle géométrique unifié via le modèle NURBS.**  
 (Si on a besoin d'éléments finis de frontière de type CAO "géométrique boundary elements", on va à l'étape 5)
3. **Génération des quadrants :**
  - a. **Subdivision récursive et construction de l'arbre quadtree.**
  - b. **Imposition de la différence d'un niveau de subdivision.**
  - c. **Classification et identification des quadrants intérieurs.**
  - d. **Génération des quadrants frontières**
4. **Application du lissage intérieur.**
5. **Génération du maillage en éléments finis.**  
 (Classique ou éléments finis de frontière)
6. **Sortie et construction les fichiers résultats.**

#### 4-10-2 : L'arbre de construction du programme :

Pour implanter l'algorithme ci-haut, on peut structurer le programme d'une façon modulaire (voir la section 1-5 du chapitre 1) comme présenté à la figure (4-18). Cette figure présente l'arbre de construction du programme. Le programme est composé de quatre modules principaux : modélisation géométrique, mesh2d, opérations numériques globales et sortie. Le module de modélisation géométrique utilise les modules "lire entrées" et "lissage composite fermé" pour construire le modèle géométrique unifié par la représentation NURBS. Le module "mesh2d" utilise les modules : "crée l'arbre quadtree", imposer un niveau de différence, classification des

quadrants et quadrants intérieurs, génération des quadrants frontières et génération de maillage. Le module de génération de maillage est capable de générer les éléments finis classiques à partir des quadrants intérieurs et extérieurs ou les éléments finis de frontière à partir du modèle géométrique. Le module d'opérations numériques globales est composé de toutes les opérations numériques qui sont utilisées dans les autres modules. Le module "sortie" prépare les sorties conviviales pour fin d'utilisation dans des codes E.F., effectue l'affichage des entrées et des différentes étapes de génération de maillage et fait l'interface avec les autres logiciels comme PATRAN, IDEAS, etc. Le fichier de sortie contient le numéro et les coordonnées de chaque noeud ainsi que le numéro, le type et les numéros des noeuds pour chaque élément généré. Chaque module au niveau feuille peut aussi utiliser d'autres petits modules. Chaque module encapsule les primitives (procédures) et les attributs (données) nécessaires pour faire son travail.

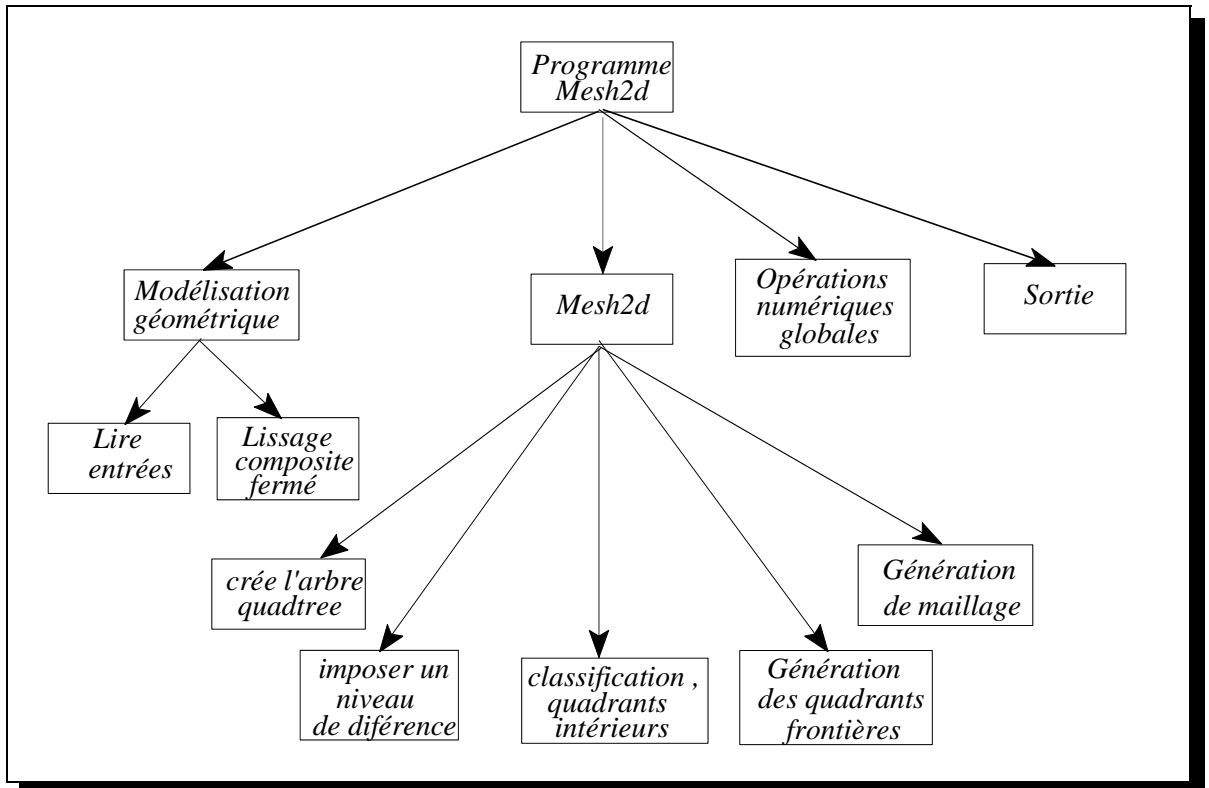


Figure 4-18 : Arbre de la construction du programme mesh2d.

#### 4-10-3 Représentation des données :

Pour la représentation des données, nous réservons l'espace pour les champs importants suivants :

**Sommets** : Les sommets des contours intérieurs et extérieurs peuvent être réunis dans des listes simplement chaînées. Il y aura une liste par contour, les sommets étant rangés dans l'ordre où on les rencontre lorsque l'on parcourt le contour.

Un sommet sera défini par :

- i. numéro de sommet,
- ii. statut du sommet : par exemple, 0 : le sommet sur la courbe ; 1 : le sommet sur le coin,
- iii. coordonnées x et y,
- iv. paramètre de profondeur du maillage.

**Segment de courbe B-spline** : Après le lissage, on doit emmagasiner les coefficients de la courbe paramétrique B-spline dans ce champ avec comme paramètres:

- i. numéro de segment,
- ii. paramètre de profondeur de maillage,
- iii. ordre de la courbe B-spline,
- iv. nombre de points de contrôle,
- v. points de contrôle,
- vi. coefficient homogène et
- vii. la matrice de coefficients de la courbe paramétrique (B\_spline).

**Quadrant** : Au niveau de quadrants et de l'arbre quadtree, on a besoin de stocker les informations suivantes :

- i. numéro du quadrant,
- ii. statut du quadrant : indéterminé, extérieur, intérieur, partiel, frontière, etc.,
- iii. niveau de subdivision du quadrant,

- iv. coordonnées du coin inférieur gauche,
- v. hauteur du quadrant,
- vi. les quatre fils de quadrant,
- vii. les quatre voisinages de quadrants,
- viii. contour de quadrants après lissage intérieur et
- ix. numéro de patron de maillage pour générer le maillage éléments finis.

**noeud** : Pour les noeuds de maillage générés après la génération de maillage, on emmagasine :

- i. le numéro et
- ii. les coordonnées de noeud.

**élément** : Pour les éléments générés après la phase de génération de maillage, on stocke :

- i. le numéro d'élément,
- ii. le type d'élément,
- iii. les numéros de noeuds définissant l'élément (connectivité) et
- iv. pointeur vers le quadrant père.
- v. les conditions aux limites et les sollicitations.
- vi. type de matériaux.

#### **4-11 Exemple de validation :**

Pour bien démontrer la capacité de cette méthode, nous présentons ici un autre exemple de génération de maillage pour une pièce mécanique. Dans les figures 4-19 nous présentons les différentes étapes de génération du maillage. La figure 4-19a présente les données entrées. À la figure 4-19b on a la représentation B-spline de la géométrie de la pièce. La figure 4-19c présente les quadrants intérieurs et frontières

généérés. Les quadrants déformés après le lissage intérieur sont présentés dans la figure 4-19d. Enfin on présente le maillage éléments finis généré aux figures 4-19e et 4-19f.

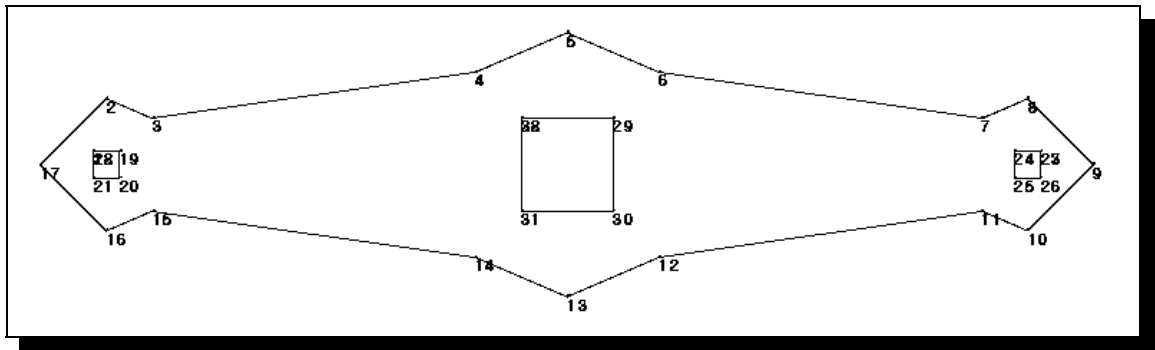


Figure 4-19a : Points donnés.

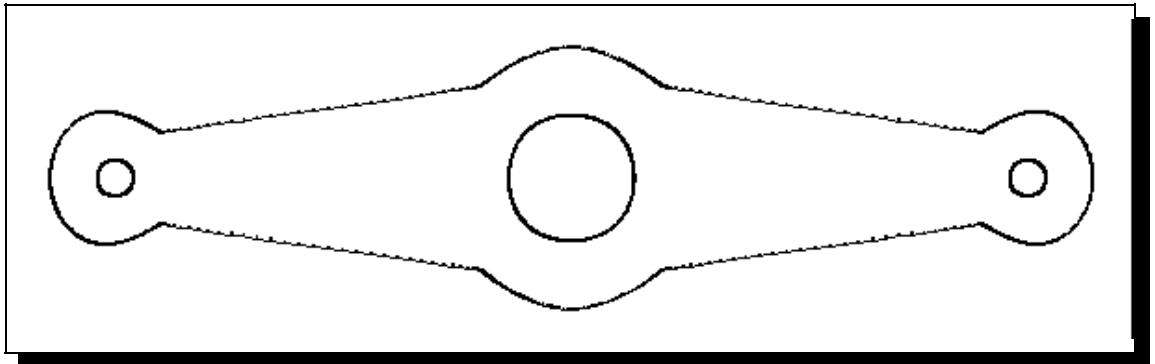


Figure 4-19b : Représentation B-spline de la pièce.

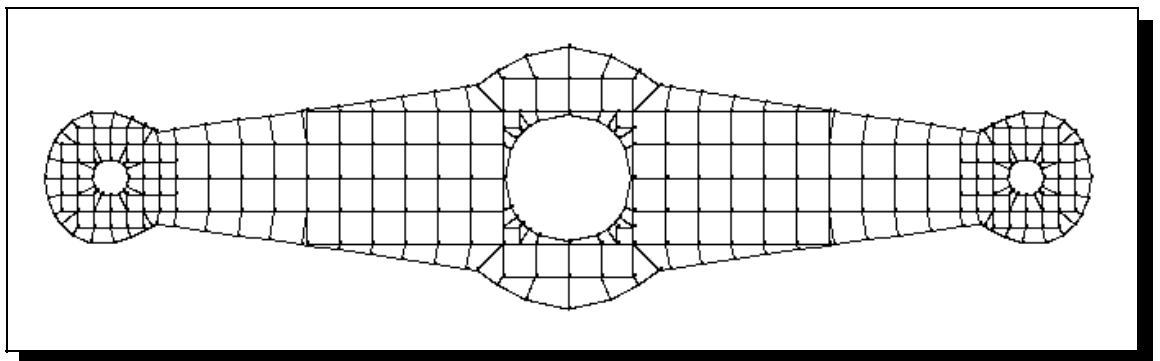


Figure 4-19c : Quadrants intérieurs et frontières générés.

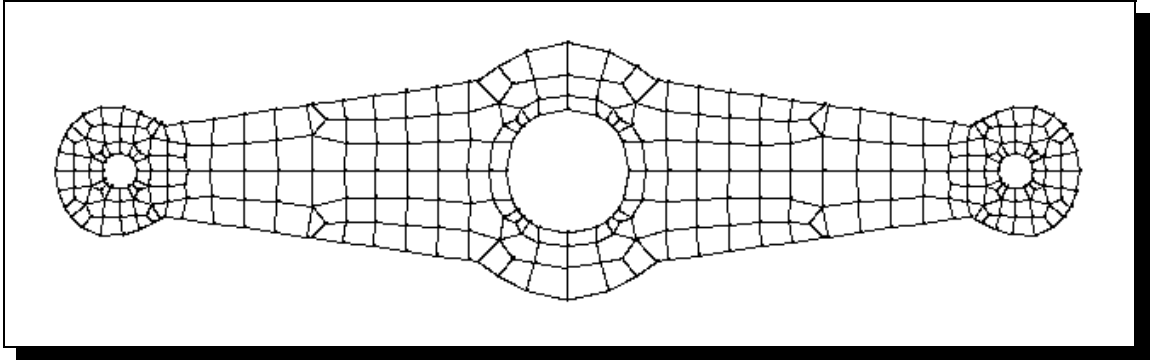


Figure 4-19d : Quadrants générés après le lissage intérieur.

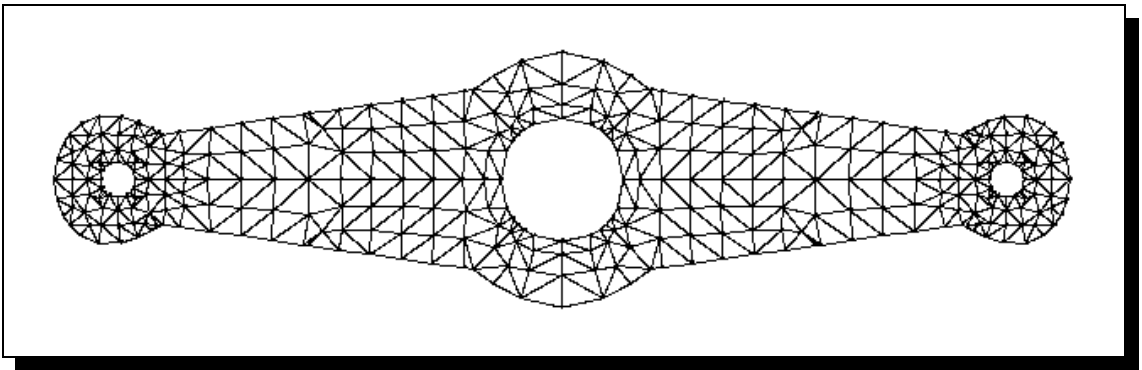


Figure 4-19e : Maillage triangulaire final de la pièce.

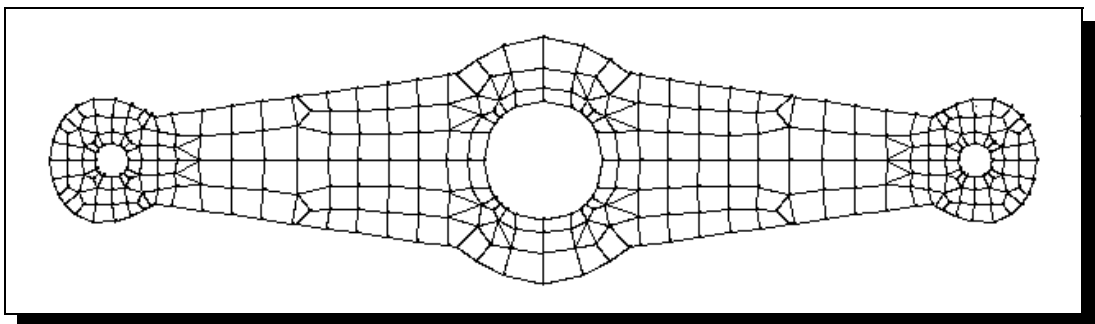


Figure 4-19f : Maillage quadrilatère/triangulaire final de la pièce.

## 4-12 Conclusions et rétrospectives :

La méthode de quadtree modifiée est une méthode bien intéressante pour automatiser entièrement le processus de génération de maillage en éléments finis en deux dimensions. Dans ce chapitre, nous avons vu que nous pouvons améliorer cette méthode en prêtant une attention particulière aux quadrants frontières. En utilisant une modélisation géométrique de l'objet par un modèle de courbe cubique B-spline rationnelle non uniforme composite fermée, nous arrivons à une représentation unifiée du modèle géométrique. Les quadrants intérieurs sont générés de la même façon que la méthode quadtree originale mais avec une petite modification sur le niveau des quadrants générés à côté des quadrants partiels, ces quadrants ayant ici le même niveau de subdivision que les quadrants partiels voisins. En éliminant les quadrants partiels et extérieurs, nous générons les quadrants frontières, entre les côtés libres des quadrants intérieurs et les courbes frontières. De cette manière, on élimine tous les cas particuliers de quadrants coupés. Donc nous éliminons tous les traitements particuliers de génération de maillage à partir des quadrants coupés. Les quadrants frontières sont générés, en utilisant un algorithme simple basé sur le principe de projection d'un point sur une courbe B-spline paramétrique. De cette façon, la similarité géométrique entre les quadrants générés et l'objet à mailler est garantie. La génération de maillage à partir des quadrants frontières est très simple, et on peut utiliser les mêmes patrons de maillage qui sont utilisés pour la génération du maillage à partir des quadrants intérieurs.

Cette méthode peut être généralisée dans l'espace à trois dimensions ; cela fera l'objet du prochain chapitre et permet d'éliminer les 4096 cas différents d'octants coupés déjà inventoriés [20-22, 24-26].