



**HAL**  
open science

# Maillages de volumes bornés par des surfaces lisses par morceaux

Laurent Rineau

► **To cite this version:**

Laurent Rineau. Maillages de volumes bornés par des surfaces lisses par morceaux. Informatique [cs]. Université Paris-Diderot - Paris VII, 2007. Français. NNT : . tel-00410864

**HAL Id: tel-00410864**

**<https://theses.hal.science/tel-00410864>**

Submitted on 24 Aug 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Meshing Volumes Bounded by Piecewise Smooth Surfaces

Laurent Rineau

3 novembre 2008



# Table des matières

<b>Table des matières</b>	<b>3</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Les qualités attendues d'un maillage simplicial . . . . .	7
1.2 Méthodes de génération automatique de maillages simpliciaux .	9
1.3 Contribution . . . . .	14
<b>2 Préliminaires mathématiques</b>	<b>17</b>
2.1 Notations élémentaires . . . . .	17
2.2 Distances . . . . .	17
2.3 Fonctions Lipschitz, et fonctions de classe $C^{1,\alpha}$ . . . . .	18
2.4 Topologie élémentaire . . . . .	18
2.5 Diagrammes de Voronoï et triangulations de Delaunay . . . . .	19
2.6 Triangulations de Delaunay restreintes . . . . .	20
<b>3 Maillage de surfaces lisses</b>	<b>21</b>
3.1 Introduction . . . . .	21
3.2 Définitions préliminaires . . . . .	22
3.3 Description de l'algorithme . . . . .	23
3.4 Précision de l'approximation . . . . .	25
3.5 Terminaison et taille du résultat . . . . .	29
3.6 Utilisation pratique . . . . .	33
3.7 Résultats . . . . .	34
<b>4 Surfaces lisses par morceaux</b>	<b>39</b>
4.1 Introduction . . . . .	39
4.2 Description de l'algorithme . . . . .	42
4.3 Précision de l'approximation . . . . .	44
4.4 Terminaison . . . . .	57
4.5 Résultats . . . . .	68
<b>5 Architecture générique pour la génération de maillages par raffinement de Delaunay</b>	<b>73</b>
5.1 Introduction . . . . .	73
5.2 Exemple typique . . . . .	74
5.3 Une architecture logicielle générique . . . . .	75
5.4 Applications et exemples . . . . .	80
<b>Bibliographie</b>	<b>85</b>



# Chapitre 1

## Introduction

La représentation d'objets par des techniques informatiques est nécessaire, à des fins d'études, dans beaucoup de domaines de la science et d'ingénierie. Le choix du type de représentation d'un objet est déterminé par l'utilisation que l'on veut faire de cette représentation. Ce peut être pour des besoins de visualisation d'un objet, pour faire des calculs ayant trait à cet objet, ou pour les deux à la fois.

Par exemple, dans le domaine médical, les appareils récents, comme les scanners d'imagerie à résonance magnétique (IRM), permettent d'acquérir des données représentant une partie d'un corps humain. Ces données prennent la forme d'une série d'images en noir et blanc ou en couleurs, représentant des coupes successives du corps étudié, où la couleur associée à un point de l'image dépend de caractéristiques physiques et physiologiques du tissu représenté en ce point de l'image. Dans la version la plus simple de l'IRM, l'image est en niveaux de gris, et l'intensité d'un pixel dépend de la densité d'eau dans le tissu. Dans des variantes plus compliquées, comme l'IRM fonctionnelle, le signal codé dans un point de l'image correspond à la quantité d'oxygène consommé par unité de temps par le tissu en ce point de l'image. Ces séries d'images successives, en coupe, forment une image tridimensionnelle, et éventuellement une image évoluant en fonction du temps. Les données fournies par l'imagerie médicale doivent être analysées par les médecins. Les besoins de ces derniers sont variés. Cela peut être la simple visualisation d'un tissu particulier du corps, en trois dimensions, des calculs comme le volume d'une tumeur présente dans le corps et visible sur les données d'IRM, ou même une simulation informatique d'une opération chirurgicale d'ablation de cette tumeur.

Dans d'autres domaines, plus liés à l'ingénierie ou à l'industrie, les objets sont créés par ordinateurs, grâce à des logiciels dits de *conception assistée par ordinateur* (CAO). Leur représentation informatique permet de visualiser ces objets, mais aussi de faire des calculs sur les objets ainsi définis. Dans l'industrie automobile, la représentation informatique d'un nouveau modèle de voiture permet de simuler la déformation de la carrosserie lors d'un choc, avant même la conception du premier prototype. Dans l'aviation, la représentation informatique d'un nouveau modèle d'avion permet de simuler l'écoulement de l'air autour de cet avion, ce qui permet d'obtenir et d'optimiser des caractéristiques de vol de l'avion pendant sa phase de conception dans les bureaux d'étude.

La représentation informatique de l'objet que l'on veut étudier doit être la

plus fidèle possible, étant données les contraintes liées à l'occupation en mémoire de cette représentation, mais elle doit aussi être la plus commode d'utilisation, pour les calculs qui sont faits sur l'objet ainsi représenté. Les *maillages* sont un type de représentation très utilisé, dans de très nombreux domaines, et notamment dans ceux déjà cités. Un maillage est une représentation d'un objet (un volume, ou une forme dans le plan) par un nombre fini d'éléments simples. Et parmi les maillages, les *maillages simpliciaux* sont ceux où tous les éléments sont des *simplexes*.

Un simplexe de dimension  $d$  (ou  $d$ -simplexe) est l'enveloppe convexe de  $d + 1$  points affinement indépendants, c'est-à-dire qui forment un repère affine dans l'espace euclidien de dimension  $d$ . Les 0-simplexes sont les points (ou sommets), les 1-simplexes sont les segments, les 2-simplexes sont les triangles, les 3-simplexes sont les tétraèdres.

Un maillage simplicial est une collection de simplexes, d'intérieurs disjoints, telle que l'intersection de deux simplexes est soit vide soit un simplexe de dimension inférieure, et telle que si un simplexe est présent dans la collection, alors ses sous-simplexes de dimension inférieure le sont aussi. Dans cette thèse, nous rencontrerons

- des maillages de surfaces, composés de simplexes de dimension inférieure ou égale à 2, dans l'espace de dimension 3,
- et des maillages volumiques, composés de simplexes de dimension inférieurs ou égale à 3, dans l'espace de dimension 3.

Les maillages que l'on considérera seront *purs*, au sens où tout simplexe du maillage est inclus dans un simplexe de dimension maximale du maillage. Ainsi, un maillage de surface pur ne contient pas de sommet ou d'arête qui ne soit pas incidente à un triangle, et un maillage volumique pur ne contient pas de sommet, d'arête, ou de triangle qui ne soit pas incident à un tétraèdre.

Les maillages simpliciaux sont très populaires pour représenter des surfaces, des volumes, éventuellement avec des données attachées, tels des champs scalaires ou des champs de vecteurs ou de tenseurs. Ils sont en effet très adaptés aux applications de simulation numérique, ou de visualisation. Par exemple, la *méthode des éléments finis* permet la résolution approchée de systèmes d'équations aux dérivées partielles, en cherchant une solution approchée du problème qui s'exprime comme combinaison linéaire d'un ensemble de fonctions élémentaires. Le choix de l'ensemble des fonctions élémentaires est très important pour la convergence de cette méthode, et est en général guidé par un maillage du domaine dans lequel on cherche à résoudre le système d'équation. À chaque élément du maillage est associé un petit ensemble de fonctions élémentaires adapté au problème, et la solution approchée est exprimé comme combinaison linéaire des fonctions élémentaires de tous les éléments du maillage. Pour calculer la solution approchée, il suffit alors d'inverser un système d'équations algébriques linéaires.

Les maillages simpliciaux sont un outil de choix pour la méthode des éléments finis, et plus généralement pour beaucoup des méthodes de calculs modélisant des systèmes physiques. Cependant, la création de tels maillages est parfois problématique. En effet, la génération d'un maillage représentant une forme continue, et approprié pour une simulation numérique ou une visualisation, peut consommer beaucoup de ressources de temps, ou de mémoire. Dans les cas compliqués, il peut être impossible de générer sans intervention ma-

nuelle un maillage répondant aux besoins du calcul. Ceci explique pourquoi la génération automatique de maillages a donné lieu à une littérature fournie. L'objet de cette thèse est de proposer de nouvelles méthodes de génération automatique de maillages, pour approcher une classe de surfaces et de volumes la plus étendue possible.

## 1.1 Les qualités attendues d'un maillage simplicial

Les applications (visualisation ou calculs) nécessitant l'utilisation de maillages ont des besoins divers. Résumons ici les différents types de besoins, et les moyens de qualifier ou quantifier la qualité d'un maillage.

### Approximation de l'objet

Le maillage doit premièrement constituer une bonne approximation de l'objet. Quand l'objet est défini par des bords polygonaux (en dimension 2) ou polyédriques (en dimension 3), l'objet peut être représenté de manière exacte par une union de simplexes. En dimension 2, un algorithme de génération de maillages doit représenter le bord d'un objet polygonal par une union de segments, et son intérieur par une union de triangles. Les arêtes polygonales peuvent être elles-même décomposées en plusieurs arêtes du maillage. En dimension 3, le volume d'un objet polyédrique est décomposé en une union de tétraèdres, son bord est décomposé en une union de triangles, et les arêtes du bord sont décomposées en arêtes du maillage.

Quand l'objet à mailler n'est pas défini par des bords polygonaux, ou polyédriques, par exemple si l'objet est défini par une union de morceaux de surfaces lisses, alors le maillage simplicial ne peut pas représenter l'objet de manière exacte. Le maillage constitue alors nécessairement une approximation de l'objet. Dans ce cas, plusieurs critères peuvent être pris en compte, pour mesurer la qualité d'approximation.

La première considération concerne l'approximation topologique. Il est souhaitable que la topologie du maillage et celle de l'objet de départ soit semblable. En de rares occasions, quand l'objet de départ a une topologie compliquée, on peut vouloir que le maillage ait une topologie plus simple. Par exemple, quand les données d'entrée sont bruitées, certaines caractéristiques topologiques, des tunnels ou des anses, sont petites, et ne correspondent pas à de vraies caractéristiques topologiques de l'objet que l'on veut représenter. Mais, le plus souvent, la topologie de l'objet de départ est bien définie, et la méthode de génération de maillages est tenue de générer un maillage *homéomorphe*<sup>1</sup> à l'objet de départ. Le bord d'un objet volumique doit correspondre à un sous-ensemble de triangles du maillage, homéomorphe lui-même au bord de l'objet. De plus, si l'objet a des cloisons intérieures, pour séparer différentes zones, le maillage doit représenter ces cloisons. Enfin, si le bord de l'objet ou les cloisons de l'objet ont des composantes unidimensionnelles singulières, comme une courbe distinguée sur une surface, le maillage doit aussi représenter ces composantes unidimensionnelles, par un sous-ensemble de segments du maillage. Idéalement, pour chaque composante de l'objet (composante connexe du bord, cloison, composante unidimensionnelle, zone distinguée dans le volume de l'objet), l'homéomorphisme

---

<sup>1</sup>Voir les rappels de topologie élémentaires, dans la section 2.1 page 17.



du maillage vers l'objet doit induire un homéomorphisme d'un sous-ensemble du maillage, vers cette composante de l'objet.

En plus de l'approximation topologique, un deuxième critère de qualité d'un maillage est l'approximation des caractéristiques géométriques de l'objet.

- Les maillages produits par la méthode de génération de maillages doivent être proches, en distance, des objets de départ. La distance de Hausdorff (voir définition 2.2.1 page 17) donne un moyen de quantifier la qualité de l'approximation géométrique.
- Quand l'objet étudié est volumique, le volume du maillage doit être proche du volume de l'objet.
- Quand l'objet est une surface, le maillage qui l'approche est constitué de triangles. Les droites normales des triangles du maillage doivent être proches des normales de l'objet approché. C'est particulièrement important pour les applications de visualisation. En effet, les méthodes de visualisation de maillages surfaciques utilisent des modèles d'éclairages dont le rendu dépend très fortement des normales aux triangles du maillage.

### Taille des éléments du maillage

Dans les applications de visualisation, la taille des éléments doit être corrélée à la taille des détails de l'objet que l'on souhaite représenter. Notamment, la taille des triangles d'un maillage de surface doit être proportionnelle localement au rayon de courbure de la surface. C'est un élément important pour la qualité d'approximation des normales de l'objet. Si la visualisation représente un champ scalaire ou un champ de vecteur sur une surface, ou dans un volume, il est important aussi d'avoir une taille inversement proportionnelle à la variation locale du champ, afin d'améliorer la précision de la visualisation.

Dans les applications de calcul, du type éléments finis, plus le maillage est *dense*, c'est-à-dire avec des éléments de petites tailles, plus la précision du calcul augmente. Cependant, la quantité de temps nécessaire à l'accomplissement d'un calcul basé sur un maillage est souvent proportionnelle au nombre d'éléments du maillage. Un compromis doit donc être trouvé entre la précision et le temps de calcul. La meilleure façon d'améliorer ce compromis est de créer un maillage avec des tailles d'éléments localement adaptés à la précision nécessaire. Dans les zones où les grandeurs étudiées par le calcul évoluent rapidement, comme par exemple au niveau des zones de turbulence dans un calcul d'écoulement de fluides, la précision du calcul doit être améliorée, et donc les éléments du maillage doivent localement être plus petits. Cependant, dans les zones où les grandeurs évoluent peu, les éléments du maillage peuvent être plus grands. Une méthode de génération de maillages est bien meilleure si elle est capable de créer un maillage où la taille des éléments peut rapidement changer, spatialement.

Étant donné un maillage *clairsemé*, avec des éléments plus grands que nécessaire, il n'est pas difficile de rajouter des éléments, pour en faire un maillage plus dense. Le contraire, qui s'appelle la *décimation* d'un maillage, est beaucoup plus difficile.

### Forme des éléments du maillage

La forme des éléments d'un maillage simplicial n'est pas très importante, pour les applications de visualisation. Cependant, elle est capitale, pour les

applications de calcul.

La forme d'un triangle est gouvernée par ses angles aux sommets. La forme d'un tétraèdre dépend des angles dièdres (les angles entre les plans supports des faces du tétraèdre) et des angles des faces du tétraèdre.

Dans les méthodes de résolution approchée d'équations différentielles, à la fois les grands angles et les petits angles doivent être évités [She02b, She]. L'erreur d'approximation de la solution dépend principalement de la taille des éléments. Cependant, pour une taille d'éléments donnée, la présence de grands angles dans le maillage augmente l'erreur d'approximation du gradient de la solution. Si certains angles sont proches de  $180^\circ$ , il se peut aussi que la méthode de calcul ne converge pas vers la solution [BA76]. Les petits angles, ainsi que les grands angles, dégradent le conditionnement du problème numérique, c'est-à-dire la stabilité de la solution par rapport aux erreurs d'arrondis du calcul. Si le problème numérique est résolu par une approche directe, un mauvais conditionnement dégrade la précision de la solution. Si le problème numérique est résolu par une méthode itérative, le mauvais conditionnement nécessite plus d'étapes de calculs, ce qui ralentit la méthode de résolution.

## 1.2 Méthodes de génération automatique de maillages simpliciaux

La génération de maillages simpliciaux, par des méthodes automatisées, est une tâche ardue, et ce sujet a donné à une importante littérature. Pour de larges références bibliographiques, le lecteur pourra se référer aux ouvrages [GF99] et [Geo01], aux articles de revue [Owe98] et [DW06], et à la thèse de Jonathan Richard Shewchuk [She97]. Cette section constitue un tour d'horizon des différentes méthodes de génération automatisée de maillages simpliciaux. Ces méthodes se classent principalement en trois catégories : les méthodes qui utilisent des grilles, celles qui fonctionnent par l'avancée d'un front, et celles qui utilisent les triangulations de Delaunay. Naturellement, cette classification n'est pas exacte, et l'on verra que des méthodes hybrides ont aussi été développées.

### Génération de maillages à l'aide de grilles régulières ou hiérarchiques

La première catégorie est la génération de maillages basés sur une grille (quadtree en dimension 2, octree en dimension 3). Dans ce genre de méthode, des cubes contenant l'objet ou le domaine à mailler sont récursivement subdivisés, jusqu'à ce qu'une taille requise soit atteinte. Les cubes sont ensuite subdivisés en simplexes. Aux alentours du bord de l'objet, cette méthode nécessite de nombreux calculs d'intersection, et crée des simplexes irréguliers. Mark Shephard et son équipe ont étudié la génération de maillage par l'utilisation de grilles, tout du long des années 1980, en dimension 2 et 3 [YS83, YS84, BWS<sup>+</sup>87]. Même si les résultats sont concluants, ces algorithmes ne sont pas accompagnés de preuve de qualité des maillages produits. Le premier algorithme à offrir des garanties théoriques est un algorithme de maillage d'un polygone, en dimension 2, par Brenda Baker, Eric Grosse, et Cristina Rafferty, en 1988 [BGR88] : en utilisant une simple grille uniforme, leur algorithme crée un maillage triangulaire, dont tous les éléments ont des angles compris entre

13° et 90° (sauf aux endroits où le polygone d'entrée a des angles plus petits que 13°). Leur algorithme est cependant coûteux, en terme de nombre d'éléments, et les auteurs suggèrent d'améliorer leur algorithme par l'utilisation de quadrees. En 1990, Marshall Bern, David Eppstein et John Gilbert proposent un algorithme pour mailler un polygone, avec des garanties à la fois sur la forme et sur la taille des triangles du maillage [BEG90]. Leur algorithme garantit de produire un maillage où tous les triangles ont leurs angles plus grands que 18,4° (à l'exception des angles du polygone initial qui seraient plus petits que cette borne), et optimal en terme de taille : le rapport entre le nombre de triangles dans le maillage produit par leur algorithme et le nombre de triangles du meilleur maillage possible, avec les mêmes garanties d'angles et de taille, est inférieur à une constante. En 1992, Mitchell et Vavasis étendent cette idée d'un algorithme de maillage avec garanties à la dimension 3, à l'aide d'un octree [MV92], puis en dimension quelconque [MV00] en 2000. Cependant, les angles garantis par l'algorithme de Mitchell et Vavasis, en dimension 3, ne sont pas satisfaisants en pratique. Un algorithme proposé en 1991 par Mark Shephard et Marcel Georges, basé lui aussi sur l'utilisation d'un octree, donne de meilleurs résultats en pratique, sans pour autant offrir de garanties théoriques [SG91].

L'utilisation de grilles hiérarchiques pour la génération de maillage a donné lieu à un algorithme populaire de maillage de surfaces implicites, l'algorithme *marching cubes* [LC87]. Cet algorithme analyse la configuration des valeurs de la fonction en les sommets de chacun des éléments d'une grille régulière, en les comparant à une valeur seuil, et crée le maillage correspondant à la surface dans chacun des éléments de la grille, selon un nombre limité de cas. Cet algorithme s'est avéré très efficace pour la génération de maillage d'isosurface extraite d'une image en trois dimensions.

## Génération de maillages par avancée de front

La deuxième catégorie d'algorithme est constituée par les méthodes d'avancée de front. Dans un premier temps, un maillage du bord de l'objet est calculé, à partir d'une carte de taille. Ce maillage du bord constitue le front initial. Ensuite, l'objet est rempli par des simplexes ajoutés un à un, et constitués à chaque fois par un élément du front qui est relié soit à un sommet déjà présent dans le maillage, soit à un sommet ajouté dans l'intérieur de la zone délimitée par le front. À chaque étape de l'algorithme, le front évolue dynamiquement, et sépare la partie maillée de l'objet de la partie non maillée. Quand un simplexe est ajouté, le front est mis à jour, et l'algorithme continue, jusqu'à ce que tout l'intérieur de l'objet soit rempli de simplexes. Ce genre de méthodes a été introduit en dimension 2 par la thèse de John Alan George, en 1971 [Geo71], puis longuement étudié par, entre autre, S.H.Lo [Lo85, Lo89, Lo91a, Lo91b], Rainald Löhner [LP88, Löh96, Löh97], Paul-Louis George et Éric Seveno [GS94], pour les dimension 2 et 3. Les méthodes par avancée de front nécessitent un choix adapté du maillage du bord de l'objet, afin d'assurer que le maillage du bord permet d'une part d'obtenir un maillage de l'objet qui satisfait les critères de taille, et d'autre part afin d'éviter que deux fronts avec des éléments de tailles différentes ne se rencontrent au centre de l'objet (car alors la méthode ne peut faire autrement que de créer des éléments de mauvaise qualité). Les méthodes par avancée de front permettent un contrôle important sur la qualité

des éléments : une fois qu'un élément du front a été sélectionné, le sommet qui sert à former un nouveau simplexe est choisi précisément pour former un simplexe avec la meilleure qualité possible. Si ce sommet optimal est trop proche d'un sommet déjà existant, le simplexe sera formé en utilisant le sommet le plus adapté, parmi les sommets proches. Ceci permet d'obtenir des simplexes de très bonne qualité, sauf éventuellement au niveau de la zone où les fronts se rejoignent. À l'inverse des méthodes de générations de maillages par octree, les méthodes frontales forment de très bons éléments près des bord de l'objet maillé. Les méthodes frontales ont été adaptées, en dimension 2, pour générer des maillages anisotropes, où la qualité des éléments ne dépend plus d'un champ de taille, mais d'un champ de tenseurs de métriques [Lee99, GF99]. Cela permet de générer des maillages particulièrement adaptés aux calculs d'écoulement de fluides. Cependant, en dimension 3, les méthodes d'avancée de front ne permettent pas toujours de générer un maillage, quand la géométrie de l'objet est compliquée.

La technique de maillage par avancée de front a été déclinée par Lau et Lo pour mailler des surfaces [LL96, LL98]. À chaque étape, le front progresse dans un plan tangent, et les nouveaux sommets sont projetés sur la surface. Ce genre de méthode nécessite cependant beaucoup de calculs d'intersection triangle-triangle, pour vérifier que les éléments du maillage de surface générés ne s'intersectent pas.

## Génération de maillages par raffinement de triangulations de Delaunay

La troisième catégorie d'algorithmes de génération de maillage est constituée par tous les algorithmes qui raffinent une triangulation de Delaunay. Dans ce genre de méthode, un maillage initial est construit, en calculant simplement la triangulation Delaunay d'un ensemble de points. Ce maillage dit « grossier » est ensuite modifié itérativement, en ajoutant un sommet à la fois. L'intérêt de la communauté des maillages pour la triangulation de Delaunay tient à deux faits :

- premièrement, la triangulation de Delaunay donne une manière canonique de connecter les sommets d'un maillage, d'après le critère de la sphère vide (voir section 2.5 où les propriétés des triangulations de Delaunay sont rappelées),
- deuxièmement, l'algorithme de Bowyer-Watson, qui tient son nom des auteurs de deux articles simultanés présentant le même algorithme [Bow81, Wat81], fournit une méthode efficace pour calculer des triangulations de Delaunay, en dimension quelconque et de manière incrémentale. Cette algorithme décrit comment mettre à jour une triangulation de Delaunay en dimension  $d$ , lors de l'insertion d'un nouveau sommet  $v$  : chacun des  $d$ -simplexes de la triangulation dont la sphère circonscrite contient le sommet  $v$  est retiré de la triangulation, ce qui crée une cavité polygonale (en dimension 2) ou polyédrique (en dimension 3 ou plus) dans la triangulation. Celle-ci est mise à jour en connectant chacun des sommets de la cavité au nouveau sommet  $v$ . L'algorithme de Bowyer-Watson décrit une méthode pour calculer cette cavité de manière efficace.

Les premiers algorithmes de maillage utilisant la triangulation de Delaunay génèrent les sommets du maillage dans un premier temps, puis construisent le maillage dans un deuxième temps, en calculant la triangulation de Delaunay de cet ensemble de sommets. Par exemple, dans [CFF85], des coupes d'un objet en dimension 3 sont utilisées pour générer une grille de sommets, et le maillage qui en résulte est un sous-complexe de la triangulation de Delaunay de ces sommets. L'idée d'utiliser la triangulation de Delaunay elle-même pour guider le choix du placement des sommets vient de François Hermeline [Her82], et William Frey [Fre87]. Ces deux articles posent les bases des méthodes de *raffinements de Delaunay*. Dans un premier temps, des sommets sont insérés sur le bord de l'objet à mailler, suffisamment rapprochés pour que les éléments du bord de l'objet soient représentés dans la triangulation de Delaunay par une union de simplexes. Puis, dans un deuxième temps, les simplexes de l'intérieur de l'objet qui ne satisfont pas les critères de taille ou de forme sont détruits, par l'insertion d'un point, à leur barycentre ou au centre de leur sphère circonscrite.

Le premier algorithme par raffinement de Delaunay présentant des garanties théoriques est celui de Paul Chew, en 1989, dans un rapport de recherche qui n'a pas été publié [Che89]. L'algorithme de Chew permet mailler un polygone en dimension 2, avec une triangulation de Delaunay, et de sorte que tous les angles de tous les triangles à l'intérieur du polygone soient entre  $30^\circ$  et  $120^\circ$ . Le polygone d'entrée ne peut cependant pas avoir d'angles internes plus petits que  $90^\circ$ . Cet algorithme produit un maillage uniforme (dont tous les triangles ont à peu près la même taille). En 1993, Jim Ruppert étend l'algorithme de Chew [Rup93, Rup95] et propose un algorithme pour mailler en dimension 2 des graphes planaires linéaires (c'est à dire des polygones avec des trous et des arêtes pendantes), avec une triangulation de Delaunay où la taille des triangles peut varier en fonction d'un critère de taille, de sorte que tous les angles des triangles soient plus grands que  $20,7^\circ$  environ. Il montre de plus que les maillages produit par son algorithme sont optimaux en terme de taille. Il obtient donc des garanties similaires à l'algorithme de Bern, Eppstein et Gilbert [BEG90] (algorithme de type octree, cité plus haut), mais son algorithme est plus simple, et produit moins de triangles en pratique. Cependant, comme l'algorithme de Chew, celui de Ruppert ne permet pas de mailler des graphes ayant des angles entre arêtes plus petits que  $90^\circ$ . Cette limitation sera levée en 1990 par l'algorithme *corner looping* [BEG90, BEG94], qui remplace le graphe planaire de l'entrée par un graphe modifié, qui ne comporte pas de petits angles. En 1996, Jonathan Shewchuk propose l'algorithme *terminator* [She96, She02a], capable de mailler des graphes planaires linéaires comportant des petits angles. Son algorithme assure que tous les angles des triangles du maillage sont plus grands que  $20,7^\circ$ , sauf ceux des triangles qui sont aux alentours de petits angles déjà présents dans le graphe à mailler. Par rapport à l'algorithme *corner looping*, l'algorithme de Shewchuk insère moins systématiquement de sommets aux alentours des petits angles. Shewchuk étend ensuite son algorithme en dimension 3, et propose un algorithme de maillage pour des objets dont le bord est linéaire par morceau (c'est à dire des emboîtement de polyèdres, avec d'éventuelles faces pendantes). Mais son algorithme en dimension 3 ne permet pas de mailler des objets avec un bord ayant des angles plus petits que  $90^\circ$ . Le problème des *petits angles* en dimension 3 est plus compliqué qu'en dimension 2. Sur une idée de David Cohen Steiner, Éric Colin de Verdière et Mariette Yvinec [CCY02, CCY04], Siu-Wing Cheng et Sheung-

Hung Poon [CP03] proposent en 2003 un algorithme utilisant des sphères de protection sur les arêtes incidentes à des facettes formant un angle plus petit que  $90^\circ$ . L'année suivante, Cheng, Tamal Dey, Edgar A. Ramos et Tathagata Ray modifient cet algorithme pour le rendre plus pratique [CDRR04].

Les algorithmes utilisant une triangulation de Delaunay sont très sensibles aux erreurs d'arrondis. L'utilisation d'une arithmétique à précision finie peut conduire à des résultats incorrects dans les prédicats utilisés pour construire la triangulation de Delaunay, et même à des choix inconsistants, ce qui peut conduire au calcul d'une triangulation invalide, avec par exemple des simplexes qui se chevauchent. Voilà pourquoi les implantations de ces algorithmes utilisent le plus souvent une arithmétique exacte, qui peut s'avérer coûteuse en temps de calcul. Paul-Louis George et Houman Borouchaki ont proposé une méthode alternative [GB98], qui consiste en l'utilisation de la méthode incrémentale de Bowyer et Watson [Bow81, Wat81], dans lequel la cavité calculée est corrigée, afin d'assurer qu'elle est étoilée par rapport au nouveau sommet, et permet de calculer une nouvelle triangulation sans simplexes qui s'intersectent. La triangulation ainsi calculée ne vérifie pas strictement le critère de Delaunay. Cette construction permet à Paul-Louis George, Frédéric Hecht et Éric Saltel de concevoir un algorithme de maillage, utilisant une telle triangulation que l'on qualifie de « presque Delaunay », où la représentation du bord du domaine est assurée par des modifications locales [GHS90, GHS91]. Le raffinement de la triangulation se déroule par l'insertion de nouveaux sommets sur les arêtes de la triangulation qui sont trop longues par rapport au critère de taille. Cette méthode sera modifiée ensuite par Bourouchaki, George, Hecht, Saltel, Patrick Laug et Bijan Mohammadi afin de pouvoir créer un maillage anisotrope en dimension 2, où la taille et la forme des triangles est guidée par une métrique riemannienne [BGH<sup>+</sup>97, BGM97].

Le maillage d'une surface par raffinement d'une triangulation de Delaunay a été étudié par Paul Chew, dans son article de 1993 [Che93], où il propose un algorithme pour mailler des surfaces lisses, mais sans étude théorique de l'algorithme. Plus récemment, Jean-Daniel Boissonnat et Steve Oudot [BO05] ont repris l'algorithme de Chew, et ont étudié sa terminaison et les garanties d'approximation obtenues par cet algorithme.

En dimension 3, le raffinement d'une triangulation de Delaunay ne permet pas de contrôler entièrement la forme des tétraèdres, mais seulement le rapport *rayon-arête*, qui est le rapport entre la longueur de la plus petite arête du tétraèdre et le rayon de la sphère circonscrite au tétraèdre. Le contrôle de ce rapport permet d'éliminer du maillage tous les tétraèdres mal formés, sauf une catégorie d'entre eux, que l'on appelle des *slivers*. Un *sliver* est un tétraèdre ayant ses quatre sommets proches d'un cercle équatorial de sa sphère circonscrite, et espacés de manière uniforme le long de cercle équatorial. Un *sliver* a quatre angles dièdres très petits, et ses deux autres angles dièdres sont très grands. Siu-Wing Cheng et Tamal Dey ont proposés des algorithmes [CDE<sup>+</sup>00, CD02] qui permettent de retirer les *slivers* du maillage, dans une phase de post-traitement, en changeant la triangulation de Delaunay en une triangulation régulière, et en assignant des petits poids bien choisis au sommets. Li et Teng [LT01] ont proposé un algorithme de raffinement de Delaunay modifié, afin d'éviter la création de *slivers* pendant le raffinement de la triangulation de Delaunay.

## Méthodes hybrides

Les triangulations de Delaunay sont devenues populaires, dans la communauté du calcul numérique, et de nombreuses méthodes hybrides ont vu le jour, combinant des méthodes frontales, ou à base de grilles hiérarchiques, à l'utilisation d'une triangulation de Delaunay pour connecter les sommets et ainsi former les simplexes.

De nombreuses publications utilisent une méthode d'avancée de front pour générer des sommets, qui sont ensuite connectés par une triangulation de Delaunay [Mer91, MW95, FBG96, FBG98]. George propose dans [Geo97] de gouverner le choix des sommets insérés par la triangulation de Delaunay, en insérant les nouveaux sommets le long des arêtes de la triangulation qui sont trop grandes. La méthode de la marche unité, proposée par Houman Borouchaki combine l'utilisation d'une méthode d'avancée de front, avec une triangulation de Delaunay modifiée pour prendre en compte des métriques riemanniennes. La méthode insère des sommets le long des arêtes de la triangulation, afin que les arêtes aient une longueur s'approchant de l'unité, dans le champ de métriques [Geo01].

Dans [Ras98], Alain Rassineux propose une méthode de maillage en dimension 3, qui utilise un octree pour générer un ensemble de sommets suffisamment dense dans le domaine à mailler, puis cette ensemble de sommets est utilisé par une méthode d'avancée de front qui génère des tétraèdres de bonne qualité, en insérant parfois des sommets supplémentaires, et en détruisant des tétraèdres quand l'avancée de front ne converge pas. En 2001, une équipe brésilienne présente une méthode similaire [CNWC<sup>+</sup>01].

### 1.3 Contribution

L'enjeu de cette thèse est la génération automatique de maillages tétraédriques, en dimension 3 donc, de domaines bornés par des surfaces lisses ou lisses par morceaux. Les techniques utilisées seront uniquement du type raffinement de Delaunay. Contrairement aux méthodes précédemment citées, les méthodes mises en jeu dans cette thèse ne se basent pas sur un maillage pré-existant du bord du domaine. Les algorithmes décrits dans cette thèse tirent partie du travail de Steve Oudot et Jean-Daniel Boissonnat [BO05] sur l'échantillonnage et l'approximation de surfaces lisses.

Dans le chapitre 3, l'algorithme proposé combine un mailleur tétraédrique insérant des centres de sphères circonscrites aux tétraèdres, au mailleur surfacique décrit dans [BO05]. Ce travail a été publié à la conférence IMRT [ORY05]. Il a aussi donné lieu à une publication dans une conférence d'imagerie médicale [PSB<sup>+</sup>07].

Le chapitre 4 est une extension de cet algorithme, pour pouvoir prendre en compte des domaines bornés par surfaces lisses par morceaux. L'extension consiste en l'ajout de nouvelles règles, dans le processus de raffinement de Delaunay, afin de prendre en compte les courbes distinguées du bord du domaine. Ce travail a été soumis et accepté à la conférence IMRT [RY07c]. Un algorithme très semblable a été publié parallèlement par Siu-Wing Cheng, Tamal Dey, Edgar Ramos et Tathagata Ray [CDR07]. Leur algorithme diffère notamment du nôtre par l'ensemble des critères utilisées par le raffinement de Delaunay, qui

sont plus difficiles à calculer. Néanmoins, leur algorithme est capable de mailler une classe plus large de domaines que le nôtre.

Les algorithmes décrits dans les chapitres 3 et 4 sont accompagnés de preuves de terminaison, et de preuve de la qualité de l'approximation du domaine.

Le chapitre 5 décrit une architecture générique, permettant l'implémentation des algorithmes de maillage par raffinement de Delaunay, et notamment des algorithmes décrits dans les chapitres 3 et 4. Cette architecture a été décrite dans un article de journal [RY07b].

Le chapitre 2 présente des rappels de notions élémentaires de géométrie et de topologie, qui seront utilisées tout du long de cette thèse. Le lecteur averti peut se contenter de s'y référer uniquement quand une notation ou une notion utilisée dans les chapitres suivantes comporte une ambiguïté.





## Chapitre 2

# Préliminaires mathématiques

### 2.1 Notations élémentaires

Tout du long de cette thèse, sauf mention contraire, l'espace ambiant sera  $\mathbb{R}^3$ , l'espace affine de dimension 3.

Les points de  $\mathbb{R}^3$  seront notés par des lettres latines, en minuscules, ainsi :  $p, q, r$ , etc. Les paramètres scalaires, dans  $\mathbb{R}$ , seront désignés par des lettres grecques, en minuscules, ainsi :  $\lambda, \gamma, \mu$ , etc. Les lettres  $d, r$  et  $l$  seront de rares exceptions, pour désigner des scalaires représentant des distances, des rayons, et des longueurs.

Les ensembles de points seront désignés par des lettres latines majuscules, ainsi :  $X, F, L$ , etc.

Les ensembles d'ensembles seront notés avec des lettres latines, en majuscules cursives, ainsi :  $\mathcal{P}, \mathcal{F}$ , etc.

### 2.2 Distances

Étant donnés deux points  $p$  et  $q$  de  $\mathbb{R}^3$ , on notera  $d(p, q)$  la distance euclidienne entre  $p$  et  $q$ .

Nous aurons aussi besoin de la distance entre un point et un ensemble de points. On notera ainsi

$$d(p, X) = \inf\{d(p, q), q \in X\}.$$

Enfin, nous utiliserons la notion suivante, pour quantifier la distance entre deux ensembles de points.

**Définition 2.2.1.** *Soient  $X$  et  $Y$  deux sous-ensembles de  $\mathbb{R}^3$ . Alors  $d(X, Y)$ , la distance de Hausdorff entre  $X$  et  $Y$  est*

$$d(X, Y) = \max\{\sup_{p \in X} d(p, Y), \sup_{q \in Y} d(q, X)\}.$$

### Angles de vecteurs, angles de droites

Dans  $\mathbb{R}^3$ , l'angle entre deux vecteurs ne peut pas être modulo  $2\pi$  comme dans  $\mathbb{R}^2$ , car un triplet de points n'est pas naturellement orienté dans  $\mathbb{R}^3$ . Les angles de vecteurs, dans  $\mathbb{R}^3$ , se définissent, ainsi, à partir du produit scalaire.

**Définition 2.2.2** (Angle de deux vecteurs). Soient  $\mathbf{u}$  et  $\mathbf{v}$  deux vecteurs de  $\mathbb{R}^3$ . L'angle entre  $\mathbf{u}$  et  $\mathbf{v}$ , noté  $(\mathbf{u}, \mathbf{v})$ , est tout angle  $\theta$  tel que :

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos \theta$$

où  $\|\mathbf{u}\|$  désigne la norme euclidienne du vecteur  $\mathbf{u}$ . Ainsi l'angle de deux vecteurs est non-signé. On peut choisir son représentant principal dans l'intervalle  $[0, \pi]$ .

À partir de cela, on définit l'angle entre deux droites.

**Définition 2.2.3** (Angle deux droites). Soient  $l_1$  et  $l_2$  deux droites affines de  $\mathbb{R}^3$ , de vecteurs directeurs respectifs  $\mathbf{u}_1$  et  $\mathbf{u}_2$ . Alors l'angle entre ces deux droites, noté  $(l_1, l_2)$  est soit  $(\mathbf{u}_1, \mathbf{u}_2)$  soit  $(\mathbf{u}_1, -\mathbf{u}_2)$ .  $(l_1, l_2)$  est celui des deux qui a le plus petit représentant dans l'intervalle  $[0, \pi]$ .

Un angle de droites est donc dans l'intervalle  $[0, \frac{\pi}{2}]$ .

Par abus de notation, je considérerai parfois des angles entre une droite  $l$  et un vecteur  $\mathbf{u}$ . Cela désignera l'angle entre la droite et la droite  $l_{\mathbf{u}}$  support du vecteur :

$$(l, \mathbf{u}) = (l, l_{\mathbf{u}}).$$

## 2.3 Fonctions Lipschitz, et fonctions de classe $C^{1,\alpha}$

**Définition 2.3.1** (fonction Lipschitz). Soit  $\lambda$  un nombre réel positif, et  $f : X \rightarrow Y$  une fonction, où  $X$  et  $Y$  sont des sous-ensembles de  $\mathbb{R}^3$ . On dit que  $f$  est  $\lambda$ -Lipschitz si pour tout  $(x, y) \in X^2$ ,

$$d(f(x), f(y)) \leq \lambda d(x, y)$$

Dans le cas où  $Y = \mathbb{R}$ , la définition est bien entendu  $d(f(x), f(y)) \leq \lambda|x-y|$ .

**Définition 2.3.2** (classe  $C^{1,\alpha}$ ). Soient  $f : X \rightarrow Y$  une fonction, où  $X$  et  $Y$  sont des sous-ensembles de  $\mathbb{R}^3$ , et  $\alpha$  un réel positif.  $f$  est dit de classe  $C^{1,\alpha}$  si  $f$  est dérivable sur  $X$ , et qu'il existe une constante  $\lambda > 0$ , telle que :

$$\forall (x, y) \in X^2, d(\nabla f(x), \nabla f(y)) \leq \lambda d(x, y)^\alpha.$$

Ainsi, une fonction  $f$  est de classe  $C^{1,1}$  si elle est dérivable et sa dérivée satisfait une condition Lipschitz.

## 2.4 Topologie élémentaire

Dans  $\mathbb{R}^3$ , muni de la distance euclidienne, on notera  $B(c, r)$  la boule fermée de centre  $c$  et de rayon  $r$  :

$$B(c, r) = \{p \in \mathbb{R}^3, d(c, p) \leq r\}.$$

En générale, les boules utilisées dans cette thèse seront fermées. À de rares occasions, il sera précisé quand les boules notées ainsi seront ouvertes.

On considérera que l'espace euclidien  $\mathbb{R}^3$  est muni de sa topologie naturelle, induite par l'ensemble des boules ouvertes. Étant donné un sous-ensemble  $X$  de  $\mathbb{R}^3$ , on notera  $\bar{X}$  la clôture de  $X$ ,  $\text{int } X$  l'intérieur de  $X$ , et  $\text{bd } X$  le bord de  $X$ .

Nous utiliserons deux notions d'équivalence topologique : l'*homéomorphisme*, et l'*isotopie*.

**Définition 2.4.1** (homéomorphisme). *Deux espaces topologiques  $X$  et  $Y$  sont homéomorphes s'il existe une application bijective, et continue,  $f : X \rightarrow Y$ , telle que  $f^{-1}$  est continue. On dit alors que  $f$  est un homéomorphisme de  $X$  vers  $Y$ .*

**Définition 2.4.2** (isotopie). *Deux espaces topologiques  $X$  et  $Y$  sont isotopes s'il existe une application continue  $f : [0, 1] \times X \rightarrow \mathbb{R}^3$ , telle que  $f(0, \cdot)$  est l'identité sur  $X$ ,  $f(1, X) = Y$ , et pour tout  $t \in [0, 1]$ ,  $f(t, \cdot)$  est un homéomorphisme de  $X$  sur  $f(t, X)$ . On dit alors que  $f$  est une isotopie de  $X$  vers  $Y$ .*

**Définition 2.4.3** (isotopie ambiante). *Si  $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  est une isotopie de  $\mathbb{R}^3$  dans lui-même, telle que  $f(1, X) = Y$ , alors on dit que  $f$  est une isotopie ambiante de  $X$  vers  $Y$ .*

Ces notions ont été présentées dans un ordre croissant : une isotopie ambiante de  $X$  vers  $Y$  induit une isotopie de  $X$  vers  $Y$ , et une isotopie de  $X$  vers  $Y$  induit un homéomorphisme de  $X$  vers  $Y$ .

## 2.5 Diagrammes de Voronoï et triangulations de Delaunay

**Définition 2.5.1** (Diagramme de Voronoï). *Soit  $\mathcal{P}$  un ensemble de points.*

- $V(p)$ , la cellule de Voronoï de  $p \in \mathcal{P}$  est l'ensemble des points de  $\mathbb{R}^3$  qui sont plus proches de  $p$  que de tout autre point  $p' \in \mathcal{P}$ .
- Pour tout sous-ensemble  $\mathcal{T} \subset \mathcal{P}$  on note  $V(\mathcal{T})$  l'intersection  $\bigcap_{p \in \mathcal{T}} V(p)$ .
- Le diagramme de Voronoï de  $\mathcal{P}$ , noté  $\mathcal{V}(\mathcal{P})$ , est le complexe cellulaire formé par l'ensemble des  $V(\mathcal{T})$  non vides. Ces ensembles  $V(\mathcal{T})$  non vides sont appelés faces du diagramme de Voronoï.

**Définition 2.5.2** (Triangulation de Delaunay). *Soit  $\mathcal{P}$  un ensemble de points, tels que cinq points distincts de  $\mathcal{P}$  ne soient jamais co-sphériques. Alors le complexe dual de  $\mathcal{V}(\mathcal{P})$  est une tétraédrisation de l'enveloppe convexe de  $\mathcal{P}$ , appelée triangulation de Delaunay de  $\mathcal{P}$ , et notée  $\mathcal{D}(\mathcal{P})$ . Pour tout sous-ensemble  $\mathcal{T} \subset \mathcal{P}$ , l'enveloppe convexe de  $\mathcal{T}$  appartient à la triangulation de Delaunay  $\mathcal{D}(\mathcal{P})$  si et seulement si  $V(\mathcal{T})$  est non vide.*

Une propriété fondamentale des triangulations de Delaunay est le *critère de la sphère vide* : l'enveloppe convexe d'un sous-ensemble  $\mathcal{T} \subset \mathcal{P}$  appartient à  $\mathcal{D}(\mathcal{P})$  si et seulement si il existe une sphère circonscrite à  $\mathcal{T}$  qui ne contient aucun point de  $\mathcal{P} \setminus \mathcal{T}$ . La cellule de Voronoï  $V(\mathcal{T})$  est en fait le lieu des centres des sphères circonscrites à  $\mathcal{T}$  ne contenant pas d'autres points de  $\mathcal{P}$ .

Par soucis de simplification, on écrira  $V(p)$ ,  $V(p, q)$ ,  $V(p, q, r)$  au lieu de  $V(\{p\})$ ,  $V(\{p, q\})$  et  $V(\{p, q, r\})$ . De même, pour dénommer des simplexes de  $\mathcal{D}(\mathcal{P})$ , on désignera par « segment  $pq$  », « triangle  $pqr$  », et « tétraèdre  $pqrs$  » respectivement les enveloppes convexes de  $\{p, q\}$ ,  $\{p, q, r\}$  et  $\{p, q, r, s\}$ .

Une cellule de Voronoï  $V(p)$  est un polytope : pour chacune des facettes  $V(p, q)$  de  $V(p)$ , on considère le demi-espace dont le bord est le plan affine de  $V(p, q)$ , et qui contient  $p$ , et  $V(p)$  est alors l'intersection de tous ces demi-espaces. Notamment, une cellule de Voronoï est une boule topologique, et est convexe.

## 2.6 Triangulations de Delaunay restreintes

Les algorithmes développés dans cette thèse reposent sur l'utilisation d'un sous-complexe de  $\mathcal{D}(\mathcal{P})$  défini ainsi :

**Définition 2.6.1** (Triangulation de Delaunay restreinte). *Soient  $\mathcal{P}$  un ensemble de points, et  $X$  un sous-ensemble de  $\mathbb{R}^3$ . On appelle triangulation de Delaunay restreinte à  $X$ , que l'on note  $\mathcal{D}_{|X}(\mathcal{P})$ , le sous-complexe de  $\mathcal{D}(\mathcal{P})$  formé par les simplexes de  $\mathcal{D}(\mathcal{P})$  dont la face de Voronoï duale intersecte  $X$ .*

Ainsi :

- un tétraèdre  $pqrs$  de  $\mathcal{D}(\mathcal{P})$  appartient à  $\mathcal{D}_{|X}(\mathcal{P})$  si et seulement si le sommet de Voronoï  $V(p, q, r, s)$  appartient à  $X$ ,
- un triangle  $pqr$  de  $\mathcal{D}(\mathcal{P})$  appartient à  $\mathcal{D}_{|X}(\mathcal{P})$  si et seulement si l'arête de Voronoï  $V(p, q, r)$  intersecte  $X$ ,
- une arête  $pq$  de  $\mathcal{D}(\mathcal{P})$  appartient à  $\mathcal{D}_{|X}(\mathcal{P})$  si et seulement si la facette de Voronoï duale  $V(p, q)$  intersecte  $X$ ,
- un sommet de  $p \in \mathcal{P}$  appartient à  $\mathcal{D}_{|X}(\mathcal{P})$  si et seulement si  $V(p)$  intersecte  $X$ .

Pour tout simplexe de  $\mathcal{D}_{|X}(\mathcal{P})$ , on appelle *boule de Delaunay restreinte*, la boule circonscrite au simplexe, centrée sur l'intersection de son dual avec  $X$ . Cette boule, ayant son centre sur le dual du simplexe, est d'intérieur vide de points de  $\mathcal{P}$ , d'après les propriétés des triangulations de Delaunay. Si  $X$  est une variété de dimension 1, cette boule est aussi appelée *boule de Delaunay linéique*. Si  $X$  est une variété de dimension 2, cette boule est appelée *boule de Delaunay surfacique*.

## Chapitre 3

# Maillage de surfaces lisses

### 3.1 Introduction

La plupart des techniques de maillage 3D maillent des domaines définis par des frontières qui sont linéaires par morceaux. Quand un objet  $\mathcal{O}$  avec des frontières non linéaires doit être maillé, son bord  $\text{bd } \mathcal{O}$  est discrétisé en un polyèdre simplicial  $P$ , avant que son intérieur puisse être maillé. D'un côté, la plupart des algorithmes de génération de maillages produisent un maillage qui est conforme à l'approximation polyédrique  $P$  du bord  $\text{bd } \mathcal{O}$ . D'un autre côté, les méthodes de maillage « à la Ruppert », comme [She98], raffinent le bord du maillage : quand un point doit être inséré sur le bord du maillage, il est inséré non pas sur  $\text{bd } \mathcal{O}$ , mais sur sa version discrétisée,  $P$ . Dans les deux cas, la qualité du maillage produit dépend fortement de la discrétisation  $P$  qui a été produite à partir de  $\text{bd } \mathcal{O}$ .

Ce chapitre décrit et analyse un nouvel algorithme de génération de maillages volumiques. Cet algorithme tire parti des résultats récents sur le maillage de surfaces lisses par raffinement de Delaunay, par Jean-Daniel Boissonnat et Steve Oudot [BO05], et permet de mailler, de manière complètement automatique, un domaine 3D borné par des surfaces lisses. Cet algorithme combine l'algorithme de maillage de surface de [BO05] et un algorithme de maillage volumique « à la Ruppert », et constitue un algorithme de maillage par raffinement de Delaunay, qui échantillonne en même temps la frontière d'un domaine volumique et son intérieur. L'algorithme n'a besoin de connaître le bord du domaine qu'à travers un oracle, qui répond à deux types de requêtes simples.

- Est-ce qu'un point donné est à l'intérieur du domaine ?
- Est-ce qu'un segment donné intersecte le bord du domaine ?

Ce faible requis sur la connaissance de la surface permet à cet algorithme d'être générique, et de mailler des objets dont le bord fait partie d'une grande variété de classes de surfaces, telles que les surfaces implicites, les iso-niveaux d'une image 3D, les surfaces définies par un ensemble de points, etc.

Concernant les garanties, il est prouvé que cet algorithme se termine, et construit des maillages de bonne qualité, pour des domaines dont le bord est une surface lisse (connexe ou non). La taille des éléments du maillage peut être contrôlée par un champ de taille défini par l'utilisateur. De plus, la précision de l'approximation du bord du domaine est garantie, et la taille du maillage produit est bornée.

Ce chapitre est organisé comme suit. La section 3.2 rappelle un certain nombre de faits et notations, concernant l'échantillonnage de surfaces. La section 3.3 décrit l'algorithme. La section 3.4 montre les garanties d'approximation de l'objet par le maillage produit. La section 3.5 présente une preuve de la terminaison de l'algorithme, et borne le nombre de sommets du maillage en sortie. La section 3.6 donne des détails pratiques, sur la façon de choisir le champ de taille, et explique comment les *slivers* sont retirés du maillage. Enfin, la section 3.7 fournit des exemples d'applications et des résultats expérimentaux.

## 3.2 Définitions préliminaires

Dans la suite de ce chapitre,  $\mathcal{O}$  désigne un sous-ensemble ouvert de l'espace euclidien  $\mathbb{R}^3$ , et  $\bar{\mathcal{O}}$  la fermeture topologique de  $\mathcal{O}$ . On appelle  $\text{bd } \mathcal{O}$  le bord de  $\mathcal{O}$ , et on suppose que  $\text{bd } \mathcal{O}$  est une surface de classe  $C^{1,1}$ , c'est-à-dire qu'en tout point le vecteur normal est défini, et le champ de vecteur normal est 1-Lipschitz.

**Définitions 3.2.1** (Axe médian et distance caractéristique locale).

- L'axe médian  $M$  de  $\text{bd } \mathcal{O}$  est la fermeture topologique de l'ensemble des points de  $\mathbb{R}^3$  qui ont au moins deux plus proches voisins sur  $\text{bd } \mathcal{O}$ . Chaque point de  $M$  est le centre d'une boule ouverte qui est maximale pour l'ordre d'inclusion dans l'ensemble des boules ouvertes incluses dans  $\mathbb{R}^3 \setminus \text{bd } \mathcal{O}$ . Une telle boule est appelée boule maximale.
- Étant donné un point  $x \in \mathbb{R}^3$ , on appelle distance caractéristique locale en  $x$ , que l'on note  $\text{lfs}(x)$ , la distance euclidienne de  $x$  à  $M$ .

Il est bien connu [Fed69] que, comme  $\text{bd } \mathcal{O}$  est  $C^{1,1}$ , l'infimum de  $\text{lfs}$  sur  $\text{bd } \mathcal{O}$  est positif. Cet infimum est appelé le *reach* de  $\text{bd } \mathcal{O}$ . La classe de surfaces dont le *reach* est strictement positif a été longuement étudiée. Sur cette classe de surface, la distance caractéristique locale a été utilisée pour définir une notion de *bon* échantillon de points, appelés  $\varepsilon$ -échantillons [AB99] :

**Définition 3.2.2** ( $\varepsilon$ -échantillon). Soit  $\mathcal{P}$  un ensemble fini de points.  $\mathcal{P}$  est un  $\varepsilon$ -échantillon de  $\text{bd } \mathcal{O}$  si il est inclu dans  $\text{bd } \mathcal{O}$  et si pour tout point  $x$  de  $\text{bd } \mathcal{O}$ ,  $d(x, \mathcal{P}) \leq \varepsilon \text{lfs}(x)$ .

On rappelle la définition suivante (voir les préliminaires mathématiques du chapitre 2) :

**Définition 3.2.3** (Triangulation de Delaunay restreinte). Soit  $\mathcal{P}$  un ensemble fini de points.

- La triangulation de Delaunay de  $\mathcal{P}$  restreinte à  $\mathcal{O}$ , notée  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ , est le sous-complexe de  $\mathcal{D}(\mathcal{P})$  formé par les tétraèdres dont les sommets de Voronoï duaux sont dans  $\mathcal{O}$ .
- La triangulation de Delaunay de  $\mathcal{P}$  restreinte à  $\text{bd } \mathcal{O}$ , notée  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$ , est le sous-complexe de  $\mathcal{D}(\mathcal{P})$  formée par les triangles dont l'arête de Voronoï duale intersecte  $\text{bd } \mathcal{O}$ .

Soit une facette  $f$  de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  et son arête de Voronoï dual  $e$ , tout point de  $e \cap \text{bd } \mathcal{O}$  est le centre d'une plus petite boule ouverte vide de point de  $\mathcal{P}$  et

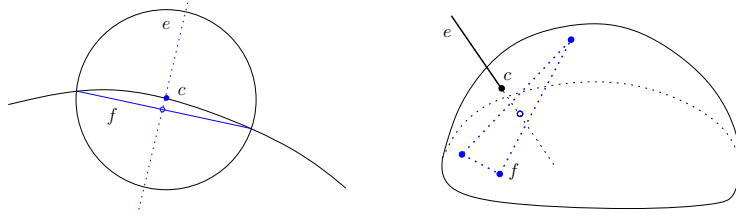


FIG. 3.1: À gauche : l'équivalent d'une boule de Delaunay surfacique, pour une courbe en dimension 2. À droite : le centre d'une boule de Delaunay surfacique d'une facette  $f \in \mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  est l'intersection de l'arête Voronoï  $e$ , duale de  $f$ , avec la surface  $\text{bd } \mathcal{O}$ .

englobant  $f$ . Cette boule est appelée *boule de Delaunay surfacique de  $\mathcal{P}$* . Voir la figure 3.1.

L'idée principale de notre algorithme est d'échantillonner  $\mathcal{O}$  et  $\text{bd } \mathcal{O}$  de manière incrémentale et simultanément, en utilisant  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  et  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  pour diriger le choix du prochain point à insérer. La sortie de l'algorithme est un ensemble de point, dont la restriction à  $\text{bd } \mathcal{O}$  est un  $\varepsilon$ -échantillon lâche de  $\text{bd } \mathcal{O}$  [BO05] :

**Définition 3.2.4.** Soit  $\mathcal{P}$  un ensemble fini de points, et  $\varepsilon$  un réel positif.  $\mathcal{P}$  est un  $\varepsilon$ -échantillon lâche de  $\text{bd } \mathcal{O}$  si les conditions suivantes sont vérifiées :

- L1**  $\mathcal{P} \subset \text{bd } \mathcal{O}$ ,
- L2**  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  a des sommets sur chaque composante connexe de  $\text{bd } \mathcal{O}$ ,
- L3** le centre  $c$  de toute boule surfacique de  $\mathcal{P}$  est à distance de  $\mathcal{P}$  inférieure à  $\varepsilon \text{ lfs}(c)$ .

Tout  $\varepsilon$ -échantillon vérifie les conditions L1 et L3. De plus, si  $\varepsilon < 0,1$ , alors L2 est vérifiée aussi, d'après le théorème 2 de [AB99]. Ainsi tout  $\varepsilon$ -échantillon est un  $\varepsilon$ -échantillon lâche, si  $\varepsilon < 0,1$ . Les propriétés des  $\varepsilon$ -échantillons lâches ont été étudiées par Jean-Daniel Boissonnat et Steve Oudot [BO05]. Elles sont résumées dans le théorème ci-dessous.

**Théorème 3.2.5.** Si  $\mathcal{P}$  est un  $\varepsilon$ -échantillon lâche de  $\text{bd } \mathcal{O}$ , avec  $\varepsilon \leq 0,09$ , alors  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  est une variété de dimension 2, fermée, image par une isotopie ambiante de  $\text{bd } \mathcal{O}$ , à distance d'Hausdorff  $O(\varepsilon^2)$  de  $\text{bd } \mathcal{O}$ , et dont les normales approchent les normales de  $\text{bd } \mathcal{O}$  avec une erreur de l'ordre de  $O(\varepsilon)$ . De plus,  $\text{bd } \mathcal{O}$  est recouverte par les boules de Delaunay surfaciques de  $\mathcal{P}$ , et  $\mathcal{P}$  est un  $\varepsilon(1 + 8,5\varepsilon)$ -échantillon de  $\text{bd } \mathcal{O}$ .

Notez que, dans la définition 3.2.4, on considère le graphe de Voronoï de  $\mathcal{P} \cap \text{bd } \mathcal{O}$ , et non celui de tout l'ensemble  $\mathcal{P}$ . De la même façon, dans le théorème 3.2.5, on considère la triangulation de Delaunay de  $\mathcal{P} \cap \text{bd } \mathcal{O}$ . Il est clair que  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  peut être différente de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P} \cap \text{bd } \mathcal{O})$ . Un des objectifs de l'algorithme sera d'assurer que ces deux objets soient identiques.

### 3.3 Description de l'algorithme

L'algorithme prend comme entrée le domaine  $\mathcal{O}$  à mailler, un champ scalaire de tailles  $\sigma$ , et deux paramètres réels  $\alpha$  and  $\varrho$ . Le domaine n'est connu que par



l'intermédiaire d'un oracle. Cet oracle répond à deux types de requêtes :

- il peut dire si un point est à l'intérieur ou à l'extérieur du domaine  $\mathcal{O}$ ,
- il peut aussi détecter si un segment intersecte  $\text{bd } \mathcal{O}$  et, le cas échéant, retourner tous les points d'intersection (qui sont, génériquement, en nombre fini).

Le champ de tailles est une fonction positive  $\sigma : \bar{\mathcal{O}} \rightarrow \mathbb{R}^+$  définie sur  $\bar{\mathcal{O}}$ . Elle est supposée 1-Lipschitz.

L'algorithme construit itérativement un ensemble de points  $\mathcal{P}$ , dont il maintient les triangulations de Delaunay restreintes  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  et  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  grâce à l'oracle, jusqu'à ce que des critères de taille et de forme sur les éléments de ces complexes soient respectés. Les critères de taille s'appliquent aux tétraèdres de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  et aux facettes de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$ . Étant donné un tétraèdre  $t$  de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ , le critère de taille requière que le rayon du cercle circonscrit à  $t$  soit inférieur à  $\sigma(c)$ , où  $\sigma(c)$  est la valeur du champ de taille évalué au centre  $c$  de la sphère circonscrite à  $t$ . Étant donnée une facette de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$ , le critère de taille requière que le rayon de toute sphère de Delaunay surfacique de  $f$  soit inférieur à  $\alpha \sigma(c)$ , où  $\sigma(c)$  est la valeur du champ de taille évalué au centre  $c$  de la sphère surfacique, et  $\alpha$  est un des paramètres de l'algorithme.

Le critère de forme est paramétré par un scalaire  $\varrho$  qui est une borne uniforme sur le *rapport rayon-arête* de tous les tétraèdres du maillage. Le rapport rayon-arête d'un tétraèdre  $t$  est  $\frac{r}{l}$ , où  $r$  est le rayon de la sphère circonscrite à  $t$ , et  $l$  est la longueur de la plus petite arête de  $t$ .

L'algorithme commence par construire un ensemble de points initial  $\mathcal{P}_0 \subset \text{bd } \mathcal{O}$ . Cet ensemble est construit en intersectant un certain nombre de rayons avec la surface  $\text{bd } \mathcal{O}$ , de sorte que  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}_0)$  ait au moins un triangle par composante connexe de  $\text{bd } \mathcal{O}$ . La construction d'un tel ensemble de points a été décrit extensivement dans [BO05], et ne sera pas précisée ici. Une fois que  $\mathcal{P}_0$  est construit, l'algorithme construit  $\mathcal{P}$  itérativement, en commençant avec  $\mathcal{P} = \mathcal{P}_0$  et en insérant un point dans  $\mathcal{P}$  à chaque itération. Les triangulations de Delaunay restreintes  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  et  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$ , sont maintenues tout de long de l'exécution de l'algorithme, à l'aide de l'oracle.

À chaque itération, un élément du maillage (une facette de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  ou un tétraèdre de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ ) est raffiné. Pour raffiner un tétraèdre, l'algorithme insère le centre de sa sphère circonscrite dans  $\mathcal{P}$ . Une facette  $f$  de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  peut avoir plusieurs boules surfaciques circonscrites. Ainsi, pour raffiner  $f$ , l'algorithme insère dans  $\mathcal{P}$  le centre de la boule de Delaunay surfacique  $B(c, r)$  qui a le plus grand rapport  $r/\sigma(c)$ . À chaque itération, le choix de l'élément suivant à raffiner est déterminé par les règles décrites ci-dessous, à considérer dans cet ordre :

- R1** si une facette  $f$  de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  n'a pas ses trois sommets sur  $\text{bd } \mathcal{O}$ , alors raffiner  $f$ ,
- R2** si une facette  $f$  de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  a une boule de Delaunay surfacique  $B(c, r)$  avec un rapport  $r/\sigma(c) > \alpha$ , alors raffiner  $f$ ,
- R3** si un tétraèdre  $t$  de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  a un rayon de sphère circonscrite plus grand que  $\sigma(c)$ , où  $c$  est le centre de la sphère, ou si  $t$  a un rapport rayon-arête plus grand que  $\varrho$ , alors on considère  $c$  le centre de la sphère circonscrite à  $t$  :

**R3.1** si  $c$  n'est inclus dans aucune sphère de Delaunay surfacique, alors insérer  $c$  dans  $\mathcal{P}$ ,

**R3.2** sinon, insérer dans  $\mathcal{P}$  le centre d'une sphère de Delaunay surfacique qui contient  $c$ .

Le déroulement de l'algorithme décrit ci-dessus peut être divisé en deux phases. La première phase commence avec  $\mathcal{P} = \mathcal{P}_0$  et continue tant que la règle R3 n'est pas déclenchée. Pendant cette phase, seule la règle R2 est appliquée, car aucun point n'a encore été inséré ailleurs que sur  $\text{bd } \mathcal{O}$ . La seconde phase commence quand la règle R3 est déclenchée pour la première fois. Pendant cette seconde phase, les trois règles sont appliquées alternativement.

L'algorithme est terminé quand les conditions déclenchant les règles R1, R2 et R3 ne sont plus vérifiées. À la fin de l'algorithme, toute facette de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  a ses trois sommets sur  $\text{bd } \mathcal{O}$  (règle R1) et toute boule de Delaunay surfacique  $B(c, r)$  a un rayon  $r \leq \alpha \sigma(c)$  (règle R2). De plus (règle R3), tout tétraèdre  $t$  de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  a un rayon de sphère circonscrite  $r \leq \min\{\sigma(c), \varrho l_{\min}\}$ , où  $c$  est le centre de la sphère et  $l_{\min}$  est la longueur de la plus petite arête de  $t$ .

### 3.4 Précision de l'approximation

Dans cette section, on suppose que l'algorithme s'est terminé. La terminaison de l'algorithme est étudiée dans la section 3.5, qui utilise quelques résultats obtenus ici.

La première phase de l'algorithme itère seulement la règle R2, tant que celle-ci peut être appliquée. Cette phase correspond à l'algorithme étudié dans [BO05]. On note  $\mathcal{P}_1$  l'ensemble de points  $\mathcal{P}$  obtenu à l'issue de cette première phase de l'algorithme. Si l'on suppose

$$\forall x \in \text{bd } \mathcal{O}, \alpha \sigma(x) \leq 0,08 \text{ lfs}(x)$$

alors, d'après l'analyse de l'algorithme faite dans [BO05], à l'issue de cette phase,  $\mathcal{P}_1$  est un 0,09-échantillon  $\frac{1}{3}$ -clairsemé de  $\text{bd } \mathcal{O}$ , c'est-à-dire :

- $\forall x \in \text{bd } \mathcal{O}, d(x, \mathcal{P}_1) \leq 0,09 \text{ lfs}(x)$  ;
- $\forall p \in \mathcal{P}_1, d(p, \mathcal{P}_1 \setminus \{p\}) \geq 0,03 \text{ lfs}(p)$ .

On note  $\mathcal{P}_{|\text{bd } \mathcal{O}} = \mathcal{P} \cap \text{bd } \mathcal{O}$ . Comme  $\mathcal{P}_1$  est un 0,09-échantillon de  $\text{bd } \mathcal{O}$ ,  $\mathcal{P}_{|\text{bd } \mathcal{O}}$  est aussi un 0,09-échantillon de  $\text{bd } \mathcal{O}$  pendant toute la deuxième phase de l'algorithme, puisqu'aucun point n'est retiré pendant son déroulement. Ainsi,  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}_{|\text{bd } \mathcal{O}})$  est une variété fermée de dimension 2, avec la même topologie que  $\text{bd } \mathcal{O}$ , d'après le théorème 3.2.5. Dès lors, pour avoir des garanties topologiques sur la sortie de l'algorithme, il suffit de prouver que le bord de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  est égal à  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}_{|\text{bd } \mathcal{O}})$ .

À partir de maintenant,  $\mathcal{P}$  désigne dans toute cette section 3.4 l'ensemble de points produit par l'algorithme.

Il existe une relation forte entre le bord  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  et  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  :

**Lemme 3.4.1.** *Le bord de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  est un sous-complexe de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$ . De plus, si toute arête du diagramme de Voronoï  $\mathcal{V}(\mathcal{P})$  intersecte  $\text{bd } \mathcal{O}$  au plus une fois, et de manière transverse, alors le bord de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  est égal à  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$ .*

*Démonstration.* Comme  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  est une union de tétraèdres de Delaunay, son bord est une union de facettes de Delaunay. Soit  $f$  une facette du bord de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ . Par définition, elle appartient à deux tétraèdres de Delaunay, l'un ayant son sommet de Voronoï dual dans  $\mathcal{O}$ , et l'autre ayant son sommet de Voronoï dual à l'extérieur de  $\mathcal{O}$ . Alors, l'arête de Voronoï duale de  $f$  intersecte  $\text{bd } \mathcal{O}$ , ce qui signifie que  $f$  appartient à  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$ . Cela prouve la première proposition du lemme.

Supposons maintenant que toute arête de  $\mathcal{V}(\mathcal{P})$  intersecte  $\text{bd } \mathcal{O}$  au plus une fois, et de manière transverse. Soit  $f$  une facette de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$ . Par définition, l'arête de Voronoï duale de  $f$  intersecte  $\text{bd } \mathcal{O}$ . Comme cette arête intersecte  $\text{bd } \mathcal{O}$  seulement une fois, et transversalement, l'un de ses sommets est à l'intérieur de  $\mathcal{O}$  alors que le second (qui peut être situé à l'infini) en est à l'extérieur. Alors, par définition de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ , l'un des deux tétraèdres incidents à  $f$  appartient à  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ , et pas l'autre. Cela signifie que  $f$  appartient au bord de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ .  $\square$

Dans notre cas,  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  est précisément le bord de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ , grâce au résultat suivant :

**Lemme 3.4.2.** *Toute arête de  $\mathcal{V}(\mathcal{P})$  intersecte  $\text{bd } \mathcal{O}$  au plus une fois, et de manière transverse.*

*Démonstration.* Parmi les arêtes de  $\mathcal{V}(\mathcal{P})$ , seulement celles dont la facette de Delaunay duale ont leurs trois sommets sur  $\text{bd } \mathcal{O}$  peuvent intersecter  $\text{bd } \mathcal{O}$ , de part la règle R1. Soit  $e$  une telle arête. Elle est incluse dans une arête  $e'$  de  $\mathcal{V}(\mathcal{P}_{|\text{bd } \mathcal{O}})$ . Comme  $\mathcal{P}_{|\text{bd } \mathcal{O}}$  est un 0,09-échantillon de  $\text{bd } \mathcal{O}$ , le lemme 3.6 de [BO05] dit que  $e'$  intersecte  $\text{bd } \mathcal{O}$  au plus une fois, et transversalement, ce qui montre le lemme.  $\square$

**Corollaire 3.4.3.** *Le bord de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  est  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$ .*

Suite au corollaire 3.4.3, si on peut prouver que  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}) = \mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}_{|\text{bd } \mathcal{O}})$ , alors le bord de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  sera égal à  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}_{|\text{bd } \mathcal{O}})$ , qui approche  $\text{bd } \mathcal{O}$  topologiquement. Nous avons encore besoin d'un résultat intermédiaire.

**Lemme 3.4.4.**  *$\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  a des sommets sur toutes les composantes connexes de  $\text{bd } \mathcal{O}$ .*

*Démonstration.* D'après la règle R1, pour toute arête  $e$  de  $\mathcal{V}(\mathcal{P})$  qui intersecte  $\text{bd } \mathcal{O}$ , sa facette de Delaunay duale  $f$  a ses trois sommets dans  $\mathcal{P}_{|\text{bd } \mathcal{O}}$ . Comme  $\mathcal{P}_{|\text{bd } \mathcal{O}}$  est un 0,09-sample de  $\text{bd } \mathcal{O}$ , le point  $c = e \cap \text{bd } \mathcal{O}$  est à distance au plus  $0,09 \text{ lfs}(c)$  des sommets de  $f$ . D'après le lemme 8 de [AB99],  $c$  et les sommets de  $f$  sont alors sur la même composante connexe de  $\text{bd } \mathcal{O}$ . Par conséquence, pour prouver le lemme, il suffit de montrer que toute composante connexe de  $\text{bd } \mathcal{O}$  est intersectée par au moins une arête de Voronoï.

Notez que toute composante connexe  $\mathcal{C}$  de  $\text{bd } \mathcal{O}$  est la frontière entre deux composantes  $\Omega_1$  et  $\Omega_2$  de  $\mathbb{R}^3 \setminus \text{bd } \mathcal{O}$ , telles que tout chemin allant de  $\Omega_1$  à  $\Omega_2$  intersecte  $\mathcal{C}$ . Dès lors, pour prouver que  $\mathcal{C}$  est intersecté par une arête de Voronoï, il suffit de prouver que  $\Omega_1$  et  $\Omega_2$  contiennent tous les deux des sommets de Voronoï, car le graphe composé des arêtes de Voronoï est connecté.

Supposons, par contradiction, qu'une composante  $\Omega$  de  $\mathbb{R}^3 \setminus \text{bd } \mathcal{O}$  ne contienne pas de sommet de Voronoï. Comme les boules de Delaunay, centrées aux sommets de Voronoï (en incluant ceux à l'infini) couvrent  $\mathbb{R}^3$ , au moins une telle

boule (disons  $B(c, r)$ ) contient un point  $x$  de  $M \cap \Omega$ . Comme  $x$  est à l'intérieur de  $\Omega$ , alors que  $c$  en est à l'extérieur, le segment  $[c, x]$  intersecte le bord de  $\Omega$  (qui est inclus dans  $\text{bd } \mathcal{O}$ ). Soit  $y$  un point de l'intersection. La boule centrée en  $y$ , de rayon  $d(x, y)$ , est contenue dans l'intérieur de  $B(c, r)$  (voir figure 3.2). Elle ne contient donc aucun point de  $\mathcal{P}$ . Son rayon est  $d(x, y)$ , qui est au moins la distance de  $y$  à  $M$  comme  $x \in M \cap \Omega$ . Par conséquent,  $y$  est plus loin de  $\mathcal{P}$  que  $\text{lfs}(y)$ , ce qui contredit le fait que  $\mathcal{P}|_{\text{bd } \mathcal{O}}$  soit un 0,09-échantillon de  $\text{bd } \mathcal{O}$ . Par conséquent,  $\Omega$  contient au moins un sommet de Voronoï, ce qui termine la preuve du lemme 3.4.4.  $\square$

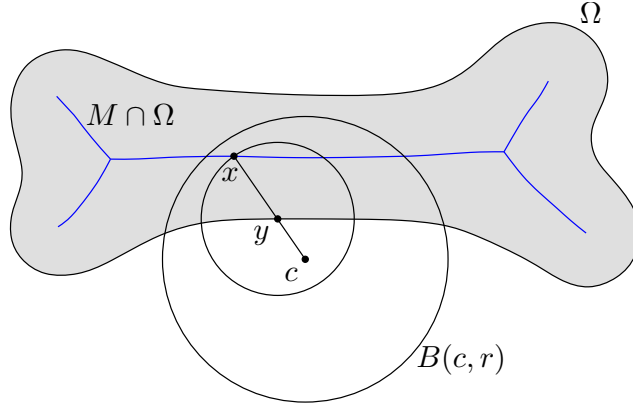


FIG. 3.2: Pour la preuve du lemme 3.4.4

Nous pouvons maintenant prouver que  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}) = \mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}|_{\text{bd } \mathcal{O}})$ , en utilisant le fait que  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  est le bord d'un objet tridimensionnel, à savoir  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  (corollaire 3.4.3).

**Lemme 3.4.5.**  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}) = \mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}|_{\text{bd } \mathcal{O}})$ .

*Démonstration.* Grâce à la règle R1, toutes les facettes de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  ont leur trois sommets dans  $\mathcal{P}|_{\text{bd } \mathcal{O}}$ , et donc leurs arêtes de Voronoï duales sont incluses dans les arêtes de  $\mathcal{V}(\mathcal{P}|_{\text{bd } \mathcal{O}})$ . Par conséquent,  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  est un sous-complexe de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}|_{\text{bd } \mathcal{O}})$ .

De plus, d'après le corollaire 3.4.3,  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  est le bord d'un objet tridimensionnel, à savoir  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ , et c'est donc l'union de composantes connexes, qui sont des variétés sans bord. Soit  $C$  une telle composante connexe. Comme  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  est un sous-complexe de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}|_{\text{bd } \mathcal{O}})$ ,  $C$  est incluse dans une composante connexe  $C'$  de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}|_{\text{bd } \mathcal{O}})$ .  $C'$  est elle-même une variété sans bord, et donc nécessairement  $C$  et  $C'$  sont identiques.

De plus, d'après le lemme 3.4.4,  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  a autant de composantes connexes que  $\text{bd } \mathcal{O}$ , et donc autant que de composantes connexes de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}|_{\text{bd } \mathcal{O}})$ , qui a la même topologie que  $\text{bd } \mathcal{O}$ . Ainsi,  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  est identique à  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}|_{\text{bd } \mathcal{O}})$ .  $\square$

Les résultats précédents impliquent que le bord de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  est égal à  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}|_{\text{bd } \mathcal{O}})$ , qui est équivalent par isotopie ambiante à  $\text{bd } \mathcal{O}$ , d'après le théorème 3.2.5.

En plus de ce résultat topologique, il serait souhaitable d'obtenir une borne sur la distance d'Hausdorff entre  $\text{bd } \mathcal{O}$  et le bord de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ , en fonction du

champ de tailles  $\sigma$ . Soit  $\varepsilon = \min\{0,09, \sup_{x \in \text{bd } \mathcal{O}} \frac{\alpha \sigma(x)}{\text{fs}(x)}\}$ . Notre borne dépendra de  $\varepsilon$ . Pour le moment, nous savons que  $\mathcal{P}_{|\text{bd } \mathcal{O}}$  est un 0,09-sample de  $\text{bd } \mathcal{O}$ . Nous allons prouver que, en contrôlant les rayons des boules de Delaunay surfaciques de  $\mathcal{P}$  (règle R2), l'algorithme contrôle en fait les rayons des boules de Delaunay surfaciques de  $\mathcal{P}_{|\text{bd } \mathcal{O}}$ .

**Lemme 3.4.6.** *Les boules de Delaunay surfaciques de  $\mathcal{P}$  et celles de  $\mathcal{P}_{|\text{bd } \mathcal{O}}$  sont les mêmes.*

*Démonstration.* Comme toute arête de  $\mathcal{V}(\mathcal{P})$  qui intersecte  $\text{bd } \mathcal{O}$  est incluse dans une arête de  $\mathcal{V}(\mathcal{P}_{|\text{bd } \mathcal{O}})$ , toutes les boules de Delaunay surfaciques de  $\mathcal{P}$  sont aussi des boules de Delaunay surfaciques de  $\mathcal{P}_{|\text{bd } \mathcal{O}}$ . Montrons que l'inverse est aussi vraie. Soit  $e$  une arête de  $\mathcal{V}(\mathcal{P}_{|\text{bd } \mathcal{O}})$ . Si  $e$  intersecte  $\text{bd } \mathcal{O}$ , alors il n'y a qu'une seule intersection :  $|e \cap \text{bd } \mathcal{O}| = 1$ , d'après le lemme 3.6 de [BO05]. De plus, la facette de Delaunay duale de  $e$  appartient à  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$ , d'après le lemme 3.4.5, elle contient donc une arête  $e'$  de  $\mathcal{V}(\mathcal{P})$ , qui intersecte  $\text{bd } \mathcal{O}$ . Comme  $|e \cap \text{bd } \mathcal{O}| = 1$ ,  $e \cap \text{bd } \mathcal{O} = e' \cap \text{bd } \mathcal{O}$ .  $\square$

D'après le lemme 3.4.6, la règle R2 contrôle les rayons des boules de Delaunay surfaciques de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}_{|\text{bd } \mathcal{O}})$ , ce qui implique que, après la fin de l'algorithme,  $\mathcal{P}_{|\text{bd } \mathcal{O}}$  is a  $\varepsilon$ -échantillon lâche de  $\text{bd } \mathcal{O}$ . Ainsi,  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}_{|\text{bd } \mathcal{O}})$  (et donc le bord de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ ) approche  $\text{bd } \mathcal{O}$  à la fois topologiquement et géométriquement, comme l'énonce le théorème suivant.

**Théorème 3.4.7.**  *$\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  est une variété de dimension 3, équivalente à  $\bar{\mathcal{O}}$  par une isotopie ambiante, à distance d'Hausdorff  $O(\varepsilon^2)$  de  $\bar{\mathcal{O}}$ , avec  $\varepsilon = \min\{0,09, \sup_{x \in \text{bd } \mathcal{O}} \frac{\alpha \sigma(x)}{\text{fs}(x)}\}$ . De plus, les boules de Delaunay surfaciques de  $\mathcal{P}$  couvrent  $\text{bd } \mathcal{O}$ .*

*Démonstration.* Par le corollaire 3.4.3 et le lemme 3.4.5, le bord de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  est égal à  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}_{|\text{bd } \mathcal{O}})$ . Comme  $\mathcal{P}_{|\text{bd } \mathcal{O}}$  est un  $\varepsilon$ -échantillon lâche de  $\text{bd } \mathcal{O}$ , on sait, par le théorème 3.2.5 qu'il existe une isotopie ambiante  $h : [0, 1] \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$  qui envoie  $\text{bd } \mathcal{O}$  sur  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}_{|\text{bd } \mathcal{O}})$ . L'application  $h(1, \cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  est un homéomorphisme ambiant qui envoie la variété compacte  $\bar{\mathcal{O}}$  sur la 3-variété compacte bornée par  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}_{|\text{bd } \mathcal{O}})$ . La seule 3-variété compacte bornée par  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}_{|\text{bd } \mathcal{O}})$  est  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  elle-même<sup>1</sup>. Ainsi,  $h(1, \bar{\mathcal{O}}) = \mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ , ce qui signifie que  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  est équivalent par une isotopie ambiante à  $\bar{\mathcal{O}}$ .

Comme  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  et  $\bar{\mathcal{O}}$  sont tous deux compacts, leur distance d'Hausdorff est atteinte pour une paire de points de leurs bords. Ainsi  $d_H(\mathcal{D}_{|\mathcal{O}}(\mathcal{P}), \bar{\mathcal{O}}) = d_H(\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P}_{|\text{bd } \mathcal{O}}), \text{bd } \mathcal{O})$ , qui est  $O(\varepsilon^2)$  car  $\mathcal{P}_{|\text{bd } \mathcal{O}}$  est un  $\varepsilon$ -échantillon de  $\text{bd } \mathcal{O}$ . Le fait que les boules de Delaunay surfaciques de  $\mathcal{P}$  couvrent  $\text{bd } \mathcal{O}$  est une conséquence directe des théorèmes 3.2.5 et lemme 3.4.6.  $\square$

Les résultats de cette section ne se basent pas sur la règle R3. Ainsi, ils sont ne sont pas seulement vrais à la fin de l'algorithme, mais aussi pendant l'exécution de celui-ci, à chaque fois que les règles R1 et R2 ne peuvent être appliquées. En particulier, le théorème 3.4.7 est vérifié à chaque fois que la règle R3 est déclenchée. Cette remarque sera utile pour prouver le lemme 3.5.2 de la section 3.5.

<sup>1</sup> $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  est compacte car c'est une union finie de tétraèdres.

### 3.5 Terminaison et taille du résultat

Dans cette section, nous donnerons des conditions sur les paramètres  $\alpha$  et  $\varrho$  pour assurer que l'algorithme termine. Nous supposons que le champ de taille  $\sigma$  est 1-Lipschitz sur  $\bar{\mathcal{O}}$ .  $\mathcal{P}$  désigne maintenant l'ensemble de points au cours du déroulement de l'algorithme.

Notre stratégie est de prouver une borne supérieure sur la taille de l'échantillon de point construit par l'algorithme. La terminaison de l'algorithme sera une conséquence de cette borne.

**Définition 3.5.1.** *Étant donné un point  $p$  inséré dans  $\mathcal{P}$  par l'algorithme, le rayon d'insertion de  $p$ , noté  $r(p)$ , est la distance euclidienne de  $p$  à  $\mathcal{P}$  juste avant son insertion<sup>2</sup>. Le rayon d'insertion d'un point  $p$  de l'ensemble de point initial  $\mathcal{P}_0$  est la distance euclidienne de  $p$  à  $\mathcal{P}_0 \setminus \{p\}$ .*

Dans un premier temps nous montrerons une borne inférieure sur le rayon d'insertion des points de  $\mathcal{P}$ . Pour cela, nous montrerons un résultat plus fort, énoncé dans le lemme 3.5.2.

La fonction lfs est définie sur  $\text{bd } \mathcal{O}$ . Nous pouvons définir un champ de taille  $\sigma_0$  sur  $\bar{\mathcal{O}}$ , en étendant lfs de cette façon :

$$\forall x \in \bar{\mathcal{O}}, \sigma_0(x) = \inf \{d(x, x') + \text{lfs}(x') \mid x' \in \text{bd } \mathcal{O}\}$$

Définie ainsi,  $\sigma_0$  est une fonction 1-Lipschitz [MTT99, TW00, ACSYD05]), identique à lfs( $x$ ) sur  $\text{bd } \mathcal{O}$ . En fait,  $\sigma_0$  est la fonction 1-Lipschitz maximale, point par point, qui vaut lfs sur  $\text{bd } \mathcal{O}$ .

Soit  $\sigma'(p) = \min\{\alpha \sigma(p), 0,03 \sigma_0(p)\}$ ,  $\forall p \in \bar{\mathcal{O}}$ . Comme  $\sigma$  et  $\sigma_0$  are 1-Lipschitz,  $\sigma'$  est  $\gamma$ -Lipschitz, avec  $\gamma = \max\{\alpha, 0,03\}$ .

**Lemme 3.5.2.** *Si  $\alpha < \frac{1}{5}$  et  $\varrho \geq \frac{4}{1-5\gamma}$ , alors les conditions suivantes sont vérifiées :*

- C1  $\forall p \in \mathcal{P}, r(p) \geq \sigma'(p)$  ;
- C2  $\forall p \in \mathcal{P} \setminus \mathcal{P}_{|\text{bd } \mathcal{O}}, \delta(p) \geq \frac{1}{1-\gamma} \sigma'(p)$ , où  $\delta(p)$  est la distance euclidienne entre  $p$  et  $\text{bd } \mathcal{O}$ .

*Démonstration.* On prouve ce lemme par induction. Au début de l'algorithme,  $\mathcal{P} = \mathcal{P}_0$  et tout point de  $\mathcal{P}_0$  vérifie C1. De plus, les points de  $\mathcal{P}_0$  appartiennent à  $\text{bd } \mathcal{O}$ , et donc la condition C2 est vérifiée trivialement par tout point de  $\mathcal{P}_0$ . Supposons maintenant que C1 et C2 sont vérifiés pour tout point de  $\mathcal{P}$ , à une certaine étape de l'algorithme. Si à l'étape suivant un point  $c$  est inséré dans  $\mathcal{P}$ . Montrons que  $c$  vérifie les conditions C1 et C2.

- Si  $c$  est inséré en appliquant la règle R1, alors  $c$  est le centre d'une boule de Delaunay surfacique de  $\mathcal{P}$  dont la sphère limite passe par un point  $p \in \mathcal{P} \setminus \mathcal{P}_{|\text{bd } \mathcal{O}}$ . Le rayon d'insertion de  $c$  est le rayon de cette sphère, c'est-à-dire  $r(c) = d(c, p)$ . De plus,  $d(c, p)$  est au moins la distance  $\delta(p)$  de  $p$  à  $\text{bd } \mathcal{O}$ . Par hypothèse d'induction,  $p$  vérifie la condition C2, et donc  $\delta(p) \geq \frac{1}{1-\gamma} \sigma'(p)$ . Comme  $\sigma'$  est  $\gamma$ -Lipschitz,  $\sigma'(p) \geq (\sigma'(c) - \gamma d(c, p))$ , et ainsi :

$$d(c, p) \geq \frac{1}{1-\gamma} (\sigma'(c) - \gamma d(c, p)) \Rightarrow r(c) = d(c, p) \geq \sigma'(c)$$

---

<sup>2</sup>Notez que c'est aussi la longueur de la plus petite arête qui est créée quand  $p$  est inséré.

Ainsi C1 est vérifié par le point  $c$ . Comme, de plus,  $c$  appartient à  $\text{bd } \mathcal{O}$ , C2 est aussi vérifiée trivialement par le point  $c$ .

- Si  $c$  est inséré par la règle R2, alors  $c$  est le centre d'une boule de Delaunay surfacique de rayon plus grand que  $\alpha \sigma(c) \geq \sigma'(c)$ . Le rayon d'insertion de  $c$  est donc au moins  $\sigma'(c)$ , ce qui satisfait la condition C1. De plus, encore une fois, comme  $c$  appartient à  $\text{bd } \mathcal{O}$ , la condition C2 est trivialement vérifiée par  $c$ .
- Si  $c$  est inséré par la règle R3.1, alors  $c$  est le centre d'un tétraèdre  $t$ , et son rayon d'insertion  $r(c)$  est le rayon  $r$  de la sphère circonscrite à  $t$ . Dans la règle R3.1,  $r$  est soit plus grand que  $\sigma(c)$  soit plus grand que  $\varrho l_{min}$ , où  $l_{min}$  est la longueur de la plus petite arête de  $t$ . Dans le premier cas, on a  $r > \sigma(c) > \alpha \sigma(c) \geq \sigma'(c)$ , car  $\alpha < 1$ , et donc  $c$  vérifie la condition C1. Dans le second cas,  $r > \varrho l_{min}$ . Parmi les deux sommets de la plus petite arête de  $t$ , appelons  $p$  celui qui a été inséré le dernier. Alors  $r(p) \leq l_{min}$ , et donc  $r > \varrho r(p)$ . De plus,  $p$  vérifie la règle C1, par hypothèse d'induction, et donc  $r(p) \geq \sigma'(p)$ . Ainsi,  $r \geq \varrho \sigma'(p)$ . Comme  $\sigma'$  est  $\gamma$ -Lipschitz,  $\varrho \sigma'(p)$  est au moins  $\varrho (\sigma'(c) - \gamma d(c, p)) \geq \varrho (\sigma'(c) - \gamma r)$ . Par conséquent,  $r \geq \frac{\varrho}{1 + \varrho \gamma} \sigma'(c)$ , ce qui montre que  $c$  vérifie la condition C1 dès lors que  $\varrho$  vérifie :

$$\varrho \geq \frac{1}{1 - \gamma} \quad (3.1)$$

Maintenant, pour montrer que  $c$  vérifie la condition C2, notons que, dans les deux cas ( $r > \sigma(c)$  et  $r > \varrho l_{min}$ ),  $r(c)$  est borné inférieurement par  $\frac{\varrho}{1 + \varrho \gamma} \sigma'(c)$ . Soit  $q$  un point de  $\text{bd } \mathcal{O}$  qui réalise la plus petite distance à  $c$ , de sorte que  $\delta(c) = d(c, q)$ . Alors :

$$d(c, q) \geq d(c, \mathcal{P}_{|\text{bd } \mathcal{O}}) - d(q, \mathcal{P}_{|\text{bd } \mathcal{O}}),$$

où  $d(c, \mathcal{P}_{|\text{bd } \mathcal{O}}) \geq r(c) \geq \frac{\varrho}{1 + \varrho \gamma} \sigma'(c)$  (car  $c$  est le centre d'une sphère de Delaunay).

Comme la règle R3 est appliquée uniquement quand les règles R1 et R2 ne peuvent plus s'appliquer, le théorème 3.4.7 est vrai, juste avant que  $c$  ne soit inséré. Ainsi, les boules de Delaunay surfaciques de  $\mathcal{P}$  recouvrent  $\text{bd } \mathcal{O}$ . Soit  $B''$ , une boule de Delaunay surfacique à laquelle  $q$  appartient, de centre  $c''$  et de rayon  $r''$ . Soit  $p \in \mathcal{P}$  un sommet de la face de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  qui est circonscrite par  $B''$ .  $d(q, p)$  est au plus  $2r''$  car  $p$  et  $q$  appartiennent tous deux à  $B''$ . Grâce à la règle R2,  $r''$  est inférieur à  $\alpha \sigma(c'')$ , ce qui est inférieur à  $\alpha(\sigma(q) + r'')$  comme  $\sigma$  est 1-Lipschitz. Par conséquent,  $d(q, \mathcal{P}_{|\text{bd } \mathcal{O}}) \leq d(q, p) \leq \frac{2\alpha}{1 - \alpha} \sigma(q)$ , ce qui est inférieur à  $3\alpha \sigma(q)$  car  $\alpha < \frac{1}{3}$ . De plus, comme  $\mathcal{P}_{|\text{bd } \mathcal{O}}$  est un 0,09-échantillon de  $\text{bd } \mathcal{O}$ ,  $d(q, \mathcal{P}_{|\text{bd } \mathcal{O}})$  est aussi borné par 0,09 lfs( $q$ ) = 3(0,03  $\sigma_0(q)$ ). Et donc,  $d(q, \mathcal{P}_{|\text{bd } \mathcal{O}}) \leq 3 \sigma'(q)$ .

Ainsi,  $\delta(c) = d(c, q) \geq \frac{\varrho}{1 + \varrho \gamma} \sigma'(c) - 3 \sigma'(q)$ . Comme  $\sigma'$  est  $\gamma$ -Lipschitz, cela implique  $\sigma'(q) \leq \sigma'(c) + \gamma d(c, q)$ , et donc

$$\delta(c) = d(c, q) \geq \frac{1}{1 + 3\gamma} \left( \frac{\varrho}{1 + \varrho \gamma} - 3 \right) \sigma'(c).$$

Par conséquent, la condition C2 est vérifiée par  $c$  si  $\varrho$  est assujéti à la relation

$$\frac{1}{1+3\gamma} \left( \frac{\varrho}{1+\varrho\gamma} - 3 \right) \geq \frac{1}{1-\gamma} \quad (3.2)$$

c'est-à-dire

$$\varrho \geq \frac{4}{1-5\gamma} \quad (3.3)$$

- Si  $c$  est inséré par la règle R3.2, alors  $c$  est le centre d'une boule de Delaunay surfacique  $B$ , de rayon  $r = r(c)$ , contenant le centre  $c'$  de la boule  $B'$  circonscrite à un tétraèdre  $t'$ . Comme dans le cas de R3.1, le rayon  $r'$  de  $B'$  est supérieur à  $\frac{\varrho}{1+\varrho\gamma} \sigma'(c')$ . Comme  $\sigma'$  est  $\gamma$ -Lipschitz, on a  $\sigma'(c') \geq \sigma'(c) - \gamma r(c)$ . Soit  $p$  un point de la facette de  $\mathcal{D}_{|\text{bd } \mathcal{O}}(\mathcal{P})$  à laquelle  $B$  est circonscrite.  $B'$  est une sphère vide (car de Delaunay), donc elle ne contient pas  $p$ , et donc  $r' \leq d(c', p)$ . Comme  $B$  contient à la fois  $p$  et  $c'$ ,  $d(c', p)$  est au plus  $2r(c)$ . Ainsi,

$$2r(c) \geq d(c', p) \geq r' \geq \frac{\varrho}{1+\varrho\gamma} (\sigma'(c) - \gamma r(c)),$$

c'est-à-dire

$$r(c) \geq \frac{\varrho}{2+3\varrho\gamma} \sigma'(c)$$

Par conséquent, la condition C1 est vérifiée par  $c$  si  $\varrho$  vérifie :

$$\varrho \geq \frac{2}{1-3\gamma} \quad (3.4)$$

De plus, comme  $c \in \text{bd } \mathcal{O}$ , la condition C2 est trivialement vérifiée par  $c$ .

En résumé, les conditions C1 et C2 sont vérifiées par le point  $c$ , et donc, par induction, tout du long de l'algorithme, si  $\varrho$  et  $\gamma$  satisfont les relations (3.1), (3.3) et (3.4). Ces relations sont assurées si on choisit  $\gamma < \frac{1}{5}$  (et donc  $\alpha < \frac{1}{5}$ ) et  $\varrho \geq \frac{4}{1-5\gamma}$ .  $\square$

À partir de maintenant, on suppose que  $\alpha < \frac{1}{5}$  et  $\varrho \geq \frac{4}{1-5\gamma}$ , où  $\gamma = \max\{\alpha, 0,03\}$ . Étant donné un point  $p \in \mathcal{P}$ , on définit  $B(p)$  comme la boule centrée en  $p$ , de rayon  $\varrho(p) = \frac{1}{2(1+\gamma)} \sigma'(p)$ .

**Lemme 3.5.3.** *Les boules  $(B(p))_{p \in \mathcal{P}}$  sont deux à deux disjointes.*

*Démonstration.* Étant donnés deux points  $p, q$  de  $\mathcal{P}$ , on suppose, sans perte de généralité, que  $q$  a été inséré dans  $\mathcal{P}$  avant  $p$ . La distance de  $p$  à  $q$  est au moins  $r(p)$ . D'après le lemme 3.5.2 (condition C1), on a  $r(p) \geq \sigma'(p)$ , ce qui est au moins  $\frac{1}{1+\gamma} \sigma'(q)$ , car  $\sigma'$  est  $\gamma$ -Lipschitz. Ainsi,  $d(p, q) \geq \frac{1}{1+\gamma} \max\{\sigma'(p), \sigma'(q)\}$ .

Par conséquent,  $\frac{1}{2(1+\gamma)} \sigma'(p) + \frac{1}{2(1+\gamma)} \sigma'(q) \leq d(p, q)$ , ce qui signifie que  $B(p)$  et  $B(q)$  sont disjointes.  $\square$

Pour calculer une borne supérieure sur la taille de l'échantillon de points, on a besoin d'un autre résultat, qui établie que chaque boule  $B(p)$  est au moins partiellement à l'intérieur de  $\mathcal{O}$ , et que le volume la partie incluse dans  $\mathcal{O}$  peut être borné intérieurement.



**Lemme 3.5.4.** *n* Pour tout point  $p \in \mathcal{P}$ ,  $B(p) \cap \mathcal{O}$  contient une boule de rayon  $\frac{1}{2} \varrho(p)$ .

*Démonstration.* On distingue deux cas :

- Si  $p$  est à l'intérieur de  $\mathcal{O}$ , alors, d'après le lemme 3.5.2 (condition C2), la distance  $\delta(p)$  de  $p$  à  $\text{bd } \mathcal{O}$  est au moins  $\frac{1}{1-\gamma} \sigma'(p)$ , qui est plus grand que  $\frac{1}{2} \varrho(p)$ . Ainsi, la boule centrée en  $p$ , de rayon  $\frac{1}{2} \varrho(p)$ , est incluse dans  $\mathcal{O}$ .
- Sinon,  $p$  est sur  $\text{bd } \mathcal{O}$ . Il existe deux boules maximales  $B_i$  et  $B_o$  tangente à  $\text{bd } \mathcal{O}$  en  $p$ . L'une des deux (disons  $B_i$ ) est incluse dans  $\mathcal{O}$ , alors que l'autre est incluse dans  $\mathbb{R}^3 \setminus \mathcal{O}$ . Comme  $B_i$  est une boule maximale, son rayon est au moins  $\text{lfs}(p) > \sigma'(p)$ . De plus, le rayon de  $B(p)$  est  $\varrho(p) < \sigma'(p)$ . Ainsi, l'intersection de  $B(p)$  avec  $B_i$  contient une boule de rayon  $\frac{1}{2} \varrho(p)$ .

□

**Théorème 3.5.5.** Si  $\alpha < \frac{1}{5}$  et  $\varrho \geq \frac{4}{1-5\gamma}$  (où  $\gamma = \max\{\alpha, 0,03\}$ ), alors l'échantillon de point  $\mathcal{P}$  produit par l'algorithme vérifie :

$$|\mathcal{P}| = O\left(\iiint_{\mathcal{O}} \frac{dx}{\sigma_0^3(x)} + \frac{1}{\alpha^3} \iiint_{\mathcal{O}} \frac{dx}{\sigma^3(x)}\right)$$

où  $\sigma_0$  dépend seulement de  $\mathcal{O}$  (et pas de  $\sigma$ ).

*Démonstration.* On utilise un schémas standard [BO05] : on borne une intégrale de  $1/\sigma'^3$  sur  $\mathcal{O}$ , où  $\sigma'$  est le minimum de  $\alpha \sigma$  et de  $0,03 \sigma_0$ . Comme  $B(p) \cap \mathcal{O} \subseteq \mathcal{O}$  pour tout point  $p \in \mathcal{P}$ , on a :

$$\iiint_{\mathcal{O}} \frac{dx}{\sigma'^3(x)} \geq \iiint_{\bigcup_{p \in \mathcal{P}} (B(p) \cap \mathcal{O})} \frac{dx}{\sigma'^3(x)}.$$

Les boules  $B(p)$  sont deux à deux disjointes, par le lemme 3.5.3. Ainsi :

$$\iiint_{\bigcup_{p \in \mathcal{P}} (B(p) \cap \mathcal{O})} \frac{dx}{\sigma'^3(x)} = \sum_{p \in \mathcal{P}} \iiint_{(B(p) \cap \mathcal{O})} \frac{dx}{\sigma'^3(x)}.$$

De plus, comme  $\sigma'$  is  $\gamma$ -Lipschitz, on a :

$$\forall x \in B(p), \sigma'(x) \leq \sigma'(p) + \gamma d(x, p) \leq \sigma'(p) + \gamma \varrho(p) = \left(1 + \frac{\gamma}{2(1+\gamma)}\right) \sigma'(p)$$

Par conséquent,

$$\sum_{p \in \mathcal{P}} \iiint_{(B(p) \cap \mathcal{O})} \frac{dx}{\sigma'^3(x)} \geq \sum_{p \in \mathcal{P}} \frac{\text{Vol}(B(p) \cap \mathcal{O})}{\left(1 + \frac{\gamma}{2(1+\gamma)}\right)^3 \sigma'^3(p)}.$$

Par le lemme 3.5.4, le volume de  $B(p) \cap \mathcal{O}$  est au plus  $\frac{4}{3} \pi \frac{1}{64(1+\gamma)^3} \sigma'^3(p)$ , ce qui donne :

$$\sum_{p \in \mathcal{P}} \frac{\text{Vol}(B(p) \cap \mathcal{O})}{\left(1 + \frac{\gamma}{2(1+\gamma)}\right)^3 \sigma'^3(p)} \geq \sum_{p \in \mathcal{P}} \frac{\frac{\pi}{48(1+\gamma)^3} \sigma'^3(p)}{\left(1 + \frac{\gamma}{2(1+\gamma)}\right)^3 \sigma'^3(p)} = \frac{\pi}{6(2+3\gamma)^3} |\mathcal{P}|,$$

qui vaut au moins  $\frac{1}{34} |\mathcal{P}|$  car  $\gamma < \frac{1}{5}$ . ainsi,  $|\mathcal{P}|$  est au moins  $34 \iiint_{\mathcal{O}} \frac{dx}{\sigma^3(x)}$ . Maintenant,  $\sigma'(x)$  est défini comme le minimum de  $\alpha \sigma(p)$  et de  $0,03 \sigma_0(p)$ , qui sont deux fonctions positives. Par conséquent,

$$\iiint_{\mathcal{O}} \frac{dx}{\sigma'^3(x)} \leq \iiint_{\mathcal{O}} \frac{dx}{\alpha^3 \sigma^3(x)} + \iiint_{\mathcal{O}} \frac{dx}{0,03^3 \sigma_0^3(x)},$$

ce qui termine la preuve du théorème.  $\square$

Comme  $\sigma_0$  et  $\sigma$  sont des fonctions positives et continue sur  $\bar{\mathcal{O}}$ , qui est compact, la borne donnée par le théorème 3.5.5 est finie. Ceci montre que l'algorithme se termine, car il insère un point dans  $\mathcal{P}$  à chaque itération, et ne retire aucun point de  $\mathcal{P}$ . Ce qui prouve le résultat suivant.

**Corollaire 3.5.6.** *L'algorithme se termine.*

## 3.6 Utilisation pratique

### Choix du champ de taille

L'algorithme de maillage présenté dans les sections précédentes prend comme entrée un champ de taille  $\sigma : \mathcal{O} \rightarrow \mathbb{R}^+$ . Ce champ de taille est supposé 1-Lipschitz, pour les besoins de l'analyse de la section 3.5. Cette section explique comment gérer un champ de taille, donné par l'utilisateur, qui ne serait pas Lipschitz, ou pas défini partout sur  $\bar{\mathcal{O}}$ .

Supposons que l'utilisateur donne un champ de taille  $\sigma_u$  qui n'est pas 1-Lipschitz. Alors, on peut utiliser la technique de Miller, Talmor et Teng [MTT99] transformer  $\sigma_u$  en un nouveau champ de taille  $\sigma'_u$  qui n'est pas 1-Lipschitz :

$$\forall x \in \bar{\mathcal{O}}, \sigma'_u(x) = \inf \{d(x, x') + \sigma_u(x') \mid x' \in \bar{\mathcal{O}}\}$$

En tout point  $p \in \bar{\mathcal{O}}$ ,  $\sigma'_u(p) \leq \sigma_u(p)$ . Le champ  $\sigma'_u$  est la meilleure approximation 1-Lipschitz de  $\sigma_u$  [ACSYD05], car toute fonction 1-Lipschitz inférieure à  $\sigma_u$  point par point, est aussi inférieure à  $\sigma'_u$  point par point.

L'algorithme de maillage peut être lancé en utilisant le champ  $\sigma'_u$ . Cependant, il n'est pas nécessaire de calculer  $\sigma'_u$  à l'intérieur de  $\mathcal{O}$ . En effet, l'algorithme ne requière l'évaluation du champ de taille uniquement aux points considérés par la règle R3, pour déclencher le raffinement d'un tétraèdre. Le raffinement d'un tétraèdre  $t$  est déclenché soit par un critère de taille (c'est-à-dire  $r \geq \sigma(c)$ , où  $c$  et  $r$  sont les centres et rayons de la sphère circonscrite à  $t$ , respectivement) ou par un critère de forme (c'est-à-dire quand le rapport rayon-arête de  $t$  est plus grand que  $\varrho$ ). Un examen précis de la preuve de terminaison de la section 3.5 montre qu'une fonction 1-Lipschitz bornant intérieurement les rayons de sphères circonscrites aux tétraèdres raffinés est suffisante pour la preuve. Ainsi, comme  $\sigma'_u(p) \leq \sigma_u(p)$  pour tout  $p \in \mathcal{P}$ , la preuve sont toujours vraies si la règle R3 est déclenchée par la condition  $r \geq \sigma_u(p)$ . En plus d'éviter l'évaluation du champ  $\sigma'_u$ , cette variante de l'algorithme construit un maillage moins dense, dont la densité est contrôlée par le champ de taille (non forcément Lipschitz) de l'utilisateur, et donc la gradation est contrôlée par la borne  $\varrho$  sur le rapport rayon-arête.

Dans le cas où l'utilisateur ne requière aucun critère de taille particulier, le champ de taille 1-Lipschitz utilisé dans l'analyse de la terminaison est le

champ  $\sigma_0$  introduit au début de la section 3.5. Encore une fois, l’algorithme n’a pas besoin d’évaluer  $\sigma_0$  à l’intérieur de  $\mathcal{O}$ . Il peut simplement éviter le test du critère de taille des tétraèdres, et ne considérer que les tétraèdres avec un mauvais rapport rayon-arête. Comme les raffinements de tétraèdres sont moins nombreux dans cette nouvelle variante de l’algorithme, il est clair que celle-ci termine aussi. La sortie de l’algorithme est alors un maillage avec des tailles de tétraèdres de l’ordre de  $\varepsilon$  lfs sur  $\text{bd } \mathcal{O}$ , et qui grossissent aussi vite que possible (en étant contraint par le critère du rapport rayon-arête) vers l’axe médian.

Dans tous les cas, l’algorithme a besoin de calculer  $\text{lfs}(x)$  sur le bord  $\text{bd } \mathcal{O}$  de l’objet, pour tester les conditions de la règle R2. Ce sujet a été étudié dans [BO05], où les auteurs proposent le calcul d’une approximation de  $\text{lfs}(x)$  basée sur la notion de  $\lambda$ -axe médian, introduit par [CL04].

### Traitement des slivers

L’optimisation du rapport rayon-arête retire du maillage tous les mauvais tétraèdres, exceptés les slivers. Afin de retirer les slivers des maillages produit par l’algorithme, on utilise l’algorithme de *pompage* de [CDE<sup>+</sup>00] en post-traitement. Cet algorithme assigne des poids choisis aux sommets du maillage, et calcule la triangulation de Delaunay régulière de l’ensemble des sommets avec leur poids. Ce procédé simule un ensemble d’inversion de faces sur la triangulation produite par l’algorithme de maillage. L’algorithme de pompage choisit les poids pour chaque sommet itérativement, qui est l’étape de pompage du sommet. Le pompage d’un sommet consiste à assigner un petit point à celui-ci, assez petit pour assurer que sa zone dans le diagramme de puissance ne recouvrira un sommet voisin, et choisit de sorte à maximiser le rapport entre le rayon de la sphère inscrite et le rayon de la sphère circonscrite des tétraèdres incidents à ce sommet. Ce rapport, appelé *rapport rayon-rayon*, est une bonne mesure de la qualité d’un tétraèdre. Bien que les garanties théoriques de l’algorithme de pompage soit très faibles [CDE<sup>+</sup>00], cette méthode est efficace en pratique, et génère des maillages pratiquement exempts de slivers [EG02].

### 3.7 Résultats

L’algorithme a été implanté en C++, en utilisant la bibliothèque CGAL [CGAL], qui fournit une implantation efficace, flexible, et robuste d’algorithme de construction de triangulation de Delaunay, en dimension 3.

Les figures 3.3 et 3.4 montrent deux maillages produit par l’algorithme présenté dans ce chapitre, après la phase d’élimination des slivers décrite dans la section 3.6. Chaque figure est composé de deux vues du maillage de sortie : une première vue (en haut à gauche) montre le bord du maillage, l’autre vue montre un zoom de l’intérieur du maillage, coupé par un plan (à droite). Toutes les figures ont été produites par le logiciel *Medit*. Dans le coin à gauche en bas de chaque figure, il est affiché une distribution des rapports rayon-rayon des tétraèdres du maillage, avec une échelle horizontale linéaire, entre 0 et  $\frac{1}{3}$  (ce qui correspond au rapport rayon-rayon d’un tétraèdre régulier). Les histogrammes sont normalisés par rapport à l’aire contenue sous l’histogramme, afin de pouvoir comparer les histogrammes de maillages ayant des nombres de sommets différents.

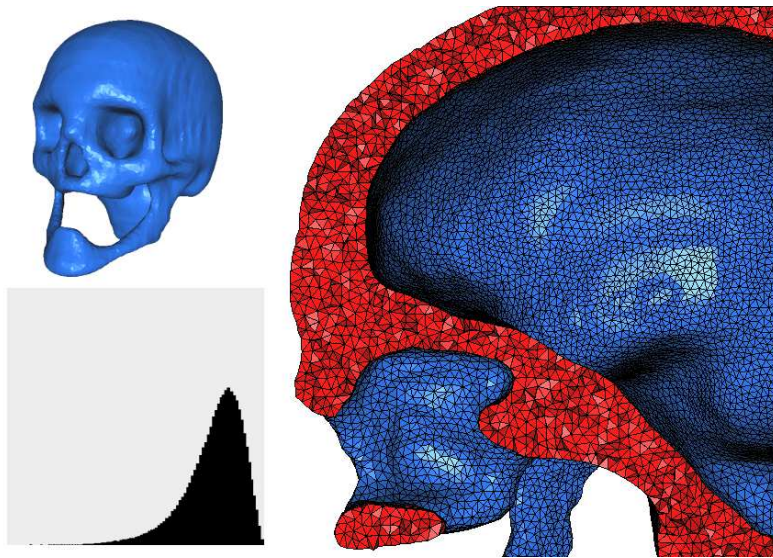


FIG. 3.3: Modèle **skull** : 89 245 sommets, 442 542 tétraèdres.

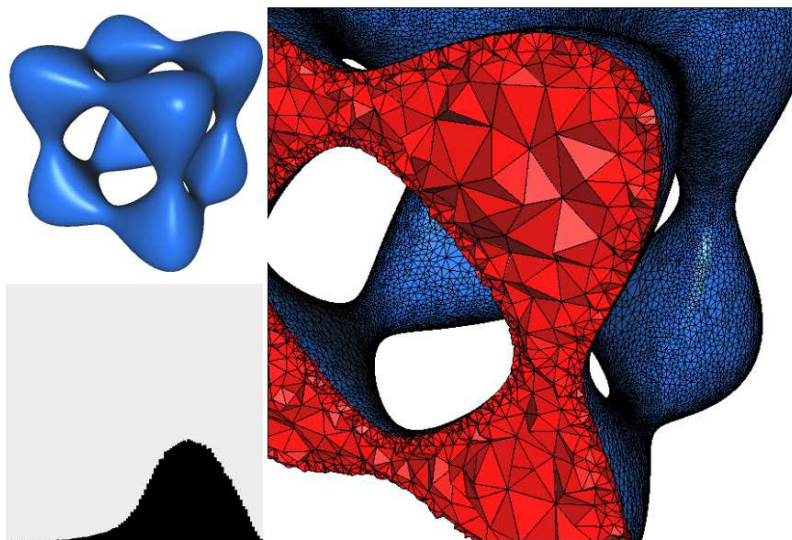


FIG. 3.4: Modèle **tanglecube** : 57 293 sommets, 226 010 tétraèdres.

Dans la figure 3.3, le bord du domaine est un iso-niveau d'un image 3D en niveaux de gris. Son diamètre est de l'ordre de 280 mm, et son *reach* est approximativement 1 mm. Bien que les résultats théoriques des sections 3.4 et 3.5 dépendent de conditions strictes sur les paramètres  $\sigma$ ,  $\alpha$  et  $\varrho$ , en pratique l'algorithme fonctionne bien avec des conditions plus lâches. Dans cet exemple, le champ de taille est uniformément de 2 mm, avec  $\alpha = 1$  et  $\varrho = 2$ , ce qui est très loin des limites théoriques. La topologie du domaine a néanmoins été capturée. L'angle dièdre minimal du maillage produit est  $5^\circ$ .

Sur la figure 3.4, la frontière du domaine est une surface algébrique de degré 4 et de genre 5, le *tanglecube*. Aucun champ de taille n'a été imposé dans l'intérieur du domaine, et le critère de taille imposé sur le bord du domaine était  $\sigma_0 = 0,09$  lfs (voir section 3.6). La borne  $\rho$  sur le rapport rayon-arête était 2, ce qui limite la gradation des tailles d'éléments dans le maillage de sortie. Bien que la distribution des rapports rayon-rayon soit détériorée par la non-uniformité du champ de taille, la qualité du maillage est globalement acceptable. L'angle dièdre minimal du maillage produit est  $1,1^\circ$ .

L'algorithme est aussi capable de prendre en entrée une surface maillée. Au lieu de réutiliser les sommets du maillage d'entrée, le déroulement normal de notre algorithme recalcule un nouveau maillage de surface qui respecte les critères de taille et de forme, et calcule un maillage volume de l'intérieur de la surface. La figure 3.5 montre un maillage de surface, et la figure 3.6 montre le résultat de notre algorithme sur cette surface. Le bord du maillage volumique produit a la même topologie que le maillage en entrée, et est à distance d'Hausdorff bornée, mais avec moins de sommets. Les critères choisis pour cette exécution de l'algorithme étaient un champs de taille uniforme, et  $\alpha = 1$ . L'angle dièdre minimal du maillage produit est  $3,2^\circ$ .

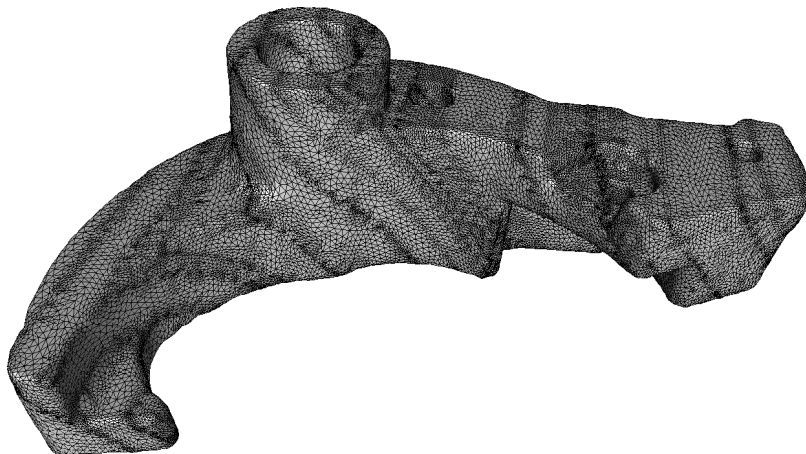


FIG. 3.5: Modèle **mechanical** : maillage de surface d'entrée (37 787 sommets).

Cet algorithme a été utilisé sur des images médicales issues de scanner, traitées par des algorithmes de segmentation 3D, puis « labelisées » par des médecins, de sorte que chaque voxel de l'image 3D contient l'identifiant qui désigne le tissu représenté par ce voxel. Pour chaque identifiant  $i$  représentant un tissu dans l'image, on considère  $\mathcal{O}_i$  le volume correspondant à l'union des voxels d'indice  $i$ . Notre algorithme a été appliqué sur ce genre d'image, pour mailler en même temps tous les volumes  $\mathcal{O}_i$ . L'oracle a été pour cela légèrement modifié. On note  $\mathcal{S} = \bigcup_i \text{bd } \mathcal{O}_i$ . On considère qu'un segment intersecte  $\mathcal{S}$  dès lors que si ses deux extrémités sont situés dans deux voxels de l'image avec des indices différents. Cette version modifiée de l'algorithme calcule un ensemble de sommets  $\mathcal{P}$ , et construit un maillage volumique composé des tétraèdres de  $\mathcal{D}_{\bigcup_i \mathcal{O}_i}(\mathcal{P})$ , ainsi qu'un maillage de surface, qui est l'ensemble des triangles de  $\mathcal{D}_{\mathcal{S}}(\mathcal{P})$ . La figure 3.7 montre des résultats ainsi obtenus. Comme la surface  $\mathcal{S}$  n'est pas à priori une variété, les résultats théoriques développés dans ce

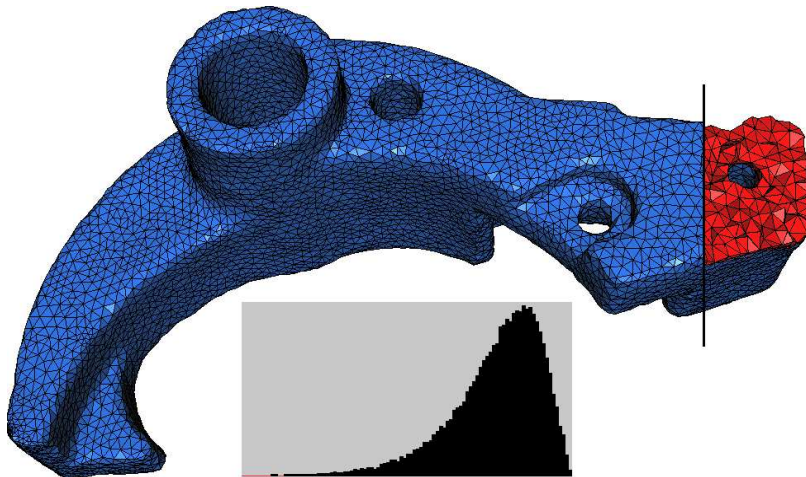


FIG. 3.6: Modèle **mechanical** : 15 687 sommets et 71 757 tétraèdres.

chapitre ne peuvent pas être appliqués. Néanmoins, les maillages obtenus sont bons. Cet algorithme adapté a été publié dans une conférence d'imagerie médicale [PSB<sup>+</sup>07] et a été bien reçu par cette communauté.

L'algorithme présenté dans ce chapitre ne peut mailler que des volumes dont le bord est composé de variétés lisses, est c'est une limitation importante. Le chapitre suivant traite d'une évolution de l'algorithme, capable de mailler des volumes bornés par des surfaces qui ne sont pas des variétés, composées d'une union de morceaux de surfaces lisses, joints en des courbes lisses, avec d'éventuelles « jonctions en T ».

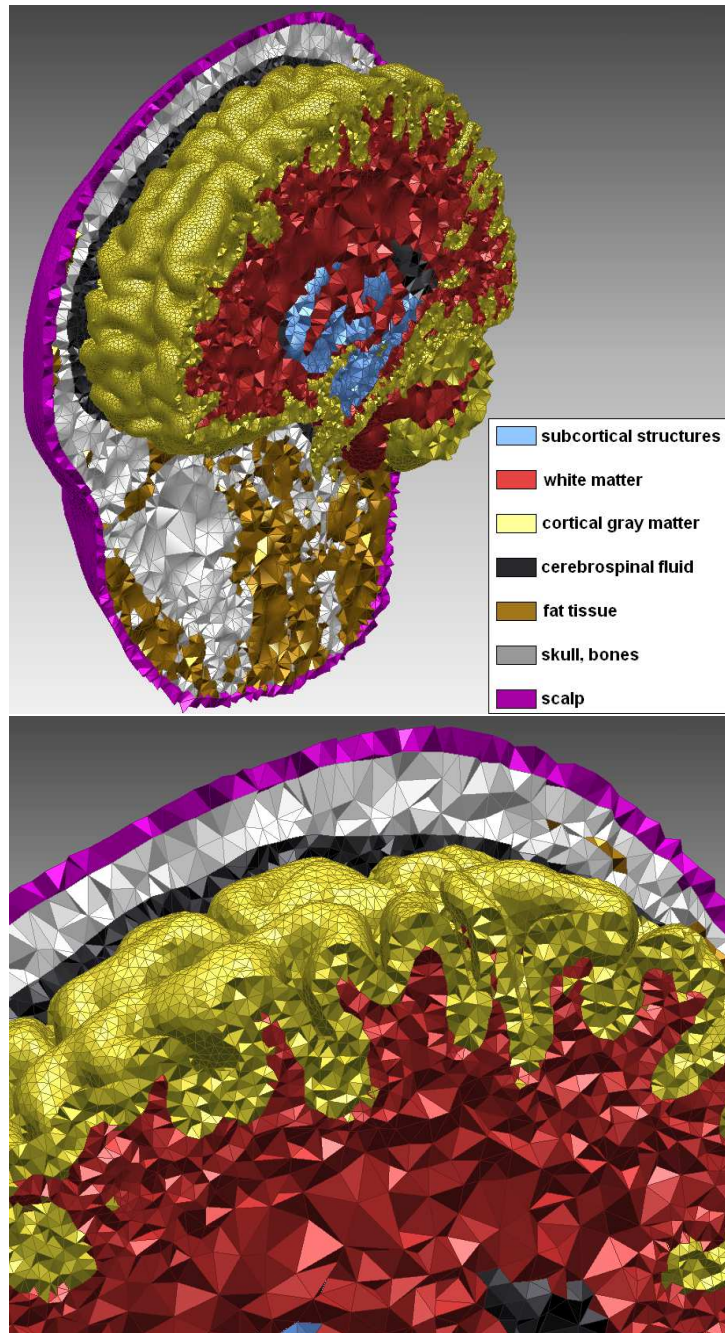


FIG. 3.7: Maillage obtenu à partir d'une image médicale 3D labélisée de la tête. Les tailles demandées des éléments dépendent du tissu. En haut, le maillage dans son ensemble, coupé par différents plans de coupe, selon le tissu. En bas, un zoom sur une partie du maillage. (Merci à Jean-Philippe Pons pour ces images.)

## Chapitre 4

# Algorithme de maillage de volumes bornés par des surfaces lisses par morceaux

### 4.1 Introduction

L'algorithme présenté dans le chapitre 3 maille des volumes bornés par des surfaces lisses. Ce chapitre 4 présente une extension de cet algorithme, afin de pouvoir mailler des volumes dont le bord est une surface lisse par morceaux. Une surface lisse par morceaux est une union de morceaux de surfaces lisses, qui sont joints par des segments de courbes lisses. L'algorithme du chapitre 3 maintient l'ensemble des triangles de la triangulation de Delaunay restreinte à une surface, et utilise un certain nombre de règles, pour assurer que la surface est bien représentée par les cet ensemble de triangles. L'algorithme de ce chapitre maintient de plus l'ensemble des arêtes de la triangulation de Delaunay restreinte aux segments de courbes. L'algorithme du chapitre 3 est amendé de règles supplémentaires qui assurent que les segments de courbes sont représentés dans le maillage de sortie par l'ensemble des arêtes de la triangulation de Delaunay restreinte aux segments de courbes.

Cet algorithme est suffisamment souple pour pouvoir mailler des volumes bornés par des surfaces qui ne sont pas des variétés, avec potentiellement des cloisons intérieures, et pouvant avoir plusieurs composantes connexes disjointes. Néanmoins, l'algorithme souffre actuellement d'une limitation importante : sa terminaison et sa correction ne sont pas démontrés si la surface bornant le volume comporte des angles plus petits que  $90^\circ$ .

La suite de cette introduction précise le type de domaine pris en charge par le nouvel algorithme, ainsi que les définitions et notations nécessaires à la lecture du chapitre. Ensuite, la section 4.2 décrit le nouvel algorithme de maillage. L'analyse de l'algorithme est faite dans les sections 4.3 et 4.4, qui analyse respectivement la précision d'approximation du maillage de sortie, et la terminaison de l'algorithme.



## Classe de domaines considérés en entrée

On supposera par la suite que le domaine  $\mathcal{O}$  à mailler est une union de cellules de dimension 3, dont les bords sont des surfaces lisses par morceaux, c'est-à-dire formés par des morceaux de surfaces lisses joints le long de segments de courbes lisses.

Plus précisément, on appellera *complexe régulier*, tout ensemble de variétés (fermées), appelées *faces*, qui vérifie les conditions suivantes :

- les faces du complexe sont d'intérieurs disjoints,
- le bord d'une face du complexe est une union de faces du complexe de dimensions inférieures.

On considère alors un complexe régulier  $\mathcal{F}$  de dimension 3, tel que :

- les faces de  $\mathcal{F}$  de dimension 2 sont des morceaux de surfaces lisses, de classe  $C^{1,1}$ , c'est-à-dire différentiables en tout point, et telles que leurs champs de vecteurs normaux unitaires satisfassent une condition Lipschitz,
- les faces de  $\mathcal{F}$  de dimension 1, qui forment le *1-squelette*, sont des segments de courbes lisses fermées de classes  $C^{1,1}$ , c'est-à-dire différentiables en tout point, et telles que leur champ de vecteur tangent unitaires satisfassent une condition Lipschitz.

On supposera que toute arête de dimension 1 est un sous-ensemble compact connecté d'une courbe lisse et fermée, et que toute face de dimension 2 est incluse dans une surface lisse sans bord, de classe  $C^{1,1}$ .

Parmi les faces de  $\mathcal{F}$ , on notera :

$\mathcal{Q}$  l'ensemble des sommets,

$\mathcal{L}$  l'ensemble des segments de courbes lisses, appelées aussi *arêtes* de  $\mathcal{F}$ ,

$\mathcal{S}$  l'ensemble des morceaux de surfaces lisses, appelées aussi *facettes* de  $\mathcal{F}$ ,

$\mathcal{C}$  l'ensemble des faces de dimension 3, appelées aussi *cellules* de  $\mathcal{F}$ ,

de sorte que  $\mathcal{F} = \mathcal{Q} \cup \mathcal{L} \cup \mathcal{S} \cup \mathcal{C}$ . Le domaine  $\mathcal{O}$  à mailler est l'union  $\mathcal{O} = \bigcup_{F \in \mathcal{F}} F$  des faces de  $\mathcal{F}$ .

On utilisera souvent la notation  $\bigcup \mathcal{F}$  pour désigner  $\bigcup_{F \in \mathcal{F}} F$ .

On notera  $\mathcal{F}_2$  l'ensemble des faces de dimension au plus 2 :  $\mathcal{F}_2 = \mathcal{Q} \cup \mathcal{L} \cup \mathcal{S}$ , et on supposera deux éléments de  $\mathcal{F}_2$  qui ne sont ni disjoints ni incidents ne forment pas un angle aigu. Plus précisément, on supposera la proposition suivante :

**Définition 4.1.1** (Hypothèse angulaire). *Il existe une distance  $\lambda_0$  telle que, pour toute paire  $(F, G)$  de faces qui ne sont ni disjointes ni incidentes de  $\mathcal{F}_2$ , s'il existe une paire de points  $(x, y) \in F \times G$ , et un point  $z$  sur  $F \cap G$ , tel que  $d(x, z) \leq \lambda_0$  et  $d(y, z) \leq \lambda_0$ , alors l'inégalité suivante est vérifiée :*

$$d(x, y)^2 \geq d(x, F \cap G)^2 + d(y, F \cap G)^2.$$

Dans le cas où  $\mathcal{F}_2$  est un complexe linéaire par morceaux (c'est-à-dire un complexe régulier avec des facettes polygonales), la *condition de projection*, introduite par Jonathan Shewchuk [She98], implique l'hypothèse angulaire ci-dessus. La condition de projection énonce que, si  $F$  et  $G$  sont deux éléments de  $\mathcal{F}_2$  qui ne sont ni disjoints ni incidents, la projection orthogonale de  $G$

sur l'espace affine engendré par  $F$  n'intersecte pas l'intérieur de  $F$ . Pour deux facettes polygonales adjacentes, cela signifie que l'angle dièdre doit être d'au moins  $90^\circ$ .

### Définition de la distance caractéristique locale

L'algorithme est contrôlé par une fonction de taille définie par l'utilisateur. Afin de pouvoir offrir des garanties sur le résultat de l'algorithme, cette fonction de taille doit être inférieure sur  $\bigcup \mathcal{F}_2$  à une fonction de taille particulière, qui dépend de la surface  $\bigcup \mathcal{F}_2$ . La fonction de taille lfs définie ci-dessous est une extension à fois de la distance caractéristique locale (*least feature size*) définie pour les polyèdres [Rup95, She98], et de la distance caractéristique locale définie pour les surfaces lisses [AB99].

Pour étendre la distance caractéristique locale définie pour les polyèdres tout en tenant compte de la courbure des éléments de  $\mathcal{F}_2$ , nous avons défini la notion de *points corrélés*, analogue à la notion de *points entrelacés* (*intertwined points*) définie dans [LS03].

**Définition 4.1.2** (Points corrélés). *Deux points  $x$  et  $y$  de  $\mathcal{S}$  sont dits corrélés si :*

- ou bien ils appartiennent à une face commune  $F \in \mathcal{F}$ ,
- ou bien ils appartiennent à deux faces qui sont non incidentes et non disjointes, respectivement  $F$  et  $G$ , et il existe un point  $w$  de l'intersection  $F \cap G$  tels que  $d(x, w) \leq \lambda_0$  et  $d(y, w) \leq \lambda_0$ .

La longueur  $\lambda_0$  provient de l'hypothèse angulaire (définition 4.1.1).

On peut alors définir une distance caractéristique locale  $\text{lfs}^P$ , analogue à la distance caractéristique locale d'un polyèdre : pour tout point  $x \in \mathbb{R}^3$ ,  $\text{lfs}^P(x)$  est le rayon de la plus petite boule centrée en  $x$  qui intersecte deux points *non corrélés* de  $\bigcup \mathcal{F}_2$ . La fonction  $\text{lfs}^P$  est une fonction 1-Lipschitz définie sur  $\mathbb{R}^3$ .

Ensuite, on définit la distance caractéristique locale  $\text{lfs}_{F_i}$ , pour chaque face de  $\mathcal{L} \cup \mathcal{S}$ . Soit  $F_i$  un segment de courbe de  $\mathcal{L}$  ou un morceau de surface de  $\mathcal{S}$ . On appelle  $F_i^*$  la variété lisse, c'est à dire de classe  $C^{1,1}$ , de même dimension que  $F_i$ , sans bord, et qui contient  $F_i$ . L'existence d'une telle variété  $F_i^*$  pour tout  $F_i \in \mathcal{L} \cup \mathcal{S}$  fait partie des hypothèses sur le complexe  $\mathcal{C}$  (voir section 4.1). Pour tout point  $p$  de  $F_i$ , on définit  $\text{lfs}_{F_i}(p)$  comme la distance entre  $p$  et l'axe médian de  $F_i^*$ . Ainsi définie, la fonction  $\text{lfs}_{F_i}$  est une fonction 1-Lipschitz sur  $F_i$ . Une technique de Miller, Talmor et Teng [MTT99] permet d'étendre cette fonction en une fonction 1-Lipschitz sur  $\mathbb{R}^3$  :

$$\forall x \in \mathbb{R}^3, \text{lfs}_{F_i}(x) = \inf_{x' \in F_i} \{d(x, x') + \text{lfs}_{F_i}(x')\}$$

Alors, pour tout point  $x \in \mathbb{R}^3$ , on définit :

$$\text{lfs}(x) = \min \left( \text{lfs}^P(x), \min_{F_i \in \mathcal{F}} \text{lfs}_{F_i}(x) \right).$$

Cette fonction lfs étant le minimum point à point d'un ensemble fini de fonction 1-Lipschitz, elle est elle-même 1-Lipschitz.

## Notations

Les notations utilisées dans tout le chapitre sont récapitulées à la figure 4.1.

$\mathcal{Q}$	l'ensemble des sommets de $\mathcal{F}$
$\mathcal{L}$	l'ensemble des arêtes de $\mathcal{F}$
$\mathcal{S}$	l'ensemble des facettes de $\mathcal{F}$
$\mathcal{C}$	l'ensemble des cellules de $\mathcal{F}$
$\mathcal{F}_2$	l'ensemble $\mathcal{Q} \cup \mathcal{L} \cup \mathcal{S}$
$\mathcal{F}_2(x)$	l'ensemble des faces de $\mathcal{F}_2$ qui contiennent le point $x$
$\text{lfs}(x)$	la distance caractéristique locale en $x$ (voir section 4.1)
$\delta(x)$	la distance entre $x$ et $\mathcal{F}_2 \setminus \mathcal{F}_2(x)$
$\mathcal{P}$	un ensemble fini de points
$\mathcal{D}(\mathcal{P})$	la triangulation de Delaunay de $\mathcal{P}$
$\mathcal{D}^1(\mathcal{P})$	l'ensemble des arêtes de la triangulation de Delaunay de $\mathcal{P}$
$\mathcal{D}^2(\mathcal{P})$	l'ensemble des triangles de la triangulation de Delaunay de $\mathcal{P}$
$\mathcal{D}^3(\mathcal{P})$	l'ensemble des tétraèdres de la triangulation de Delaunay de $\mathcal{P}$
$\mathcal{V}(\mathcal{P})$	le diagramme de Voronoï de $\mathcal{P}$
$V(p)$	la cellule de Voronoï d'un point $p$ , dans $\mathcal{V}(\mathcal{P})$
$V(p, q)$	la facette de Voronoï $V(p) \cap V(q)$ , dans $\mathcal{V}(\mathcal{P})$
$V(\mathcal{T})$	plus généralement, pour $\mathcal{T} \subset \mathcal{P}$ , la face de Voronoï $\bigcap_{p \in \mathcal{T}} V(p)$
$\mathcal{D}_{ X}(\mathcal{P})$	la triangulation de Delaunay de $\mathcal{P}$ restreinte à un espace $X \subset \mathbb{R}^3$
$\mathbf{n}_x(S_k)$	un vecteur normal à $S_k$ en $x$ , unitaire (si la surface $S_k$ n'a pas besoin d'être précisée, la notation raccourcie $\mathbf{n}_x$ sera utilisée)
$\mathbf{n}(f)$	une normale unitaire à une facette $f$ de $\mathcal{D}(\mathcal{P})$
$H_x(S_k)$	le plan tangent à $S_k$ en $x$ , qui s'abrèvie en $H_x$ si $S_k$ n'a pas besoin d'être précisée

FIG. 4.1: Notations utilisées dans le chapitre

## 4.2 Description de l'algorithme

L'algorithme est un algorithme de raffinement de Delaunay, qui construit itérativement un ensemble de points  $\mathcal{P}$ , et maintient tout du long de son exécution

- la triangulation de Delaunay de  $\mathcal{P}$ , notée  $\mathcal{D}(\mathcal{P})$ ,
- pour toute arête  $L_j \in \mathcal{L}$ , l'ensemble des arêtes de la triangulation de Delaunay de  $\mathcal{P}$  restreinte à  $L_j$ , noté  $\mathcal{D}_{|L_j}^1(\mathcal{P})$ ,

- pour toute facette  $S_k \in \mathcal{S}$ , l'ensemble des facettes de la triangulation de Delaunay de  $\mathcal{P}$  restreinte à  $S_k$ , noté  $\mathcal{D}_{|S_k}^2(\mathcal{P})$ ,
- l'ensemble des tétraèdres de la triangulation de Delaunay restreinte à  $\mathcal{O}$  de  $\mathcal{P}$ , noté  $\mathcal{D}_{|\mathcal{O}}^3(\mathcal{P})$ .

À la fin de l'algorithme, l'ensemble des tétraèdres de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  forme le maillage final, et les sous-complexes  $\mathcal{D}_{|S_k}(\mathcal{P})$  et  $\mathcal{D}_{|L_j}(\mathcal{P})$  sont respectivement de bonnes approximations de  $S_k$  et  $L_j$ .

L'algorithme prend en entrée :

- un domaine à mailler,  $\mathcal{O}$ , décrit par un complexe régulier  $\mathcal{C}$ ,
- une fonction de taille  $\sigma : \mathcal{O} \rightarrow \mathbb{R}$ ,
- des critères de formes, qui sont donnés par deux paramètres réels  $\beta_2, \beta_3$ . Ces deux constantes sont respectivement les bornes supérieures sur les rapports rayon-arête des facettes de  $\mathcal{D}_{|\cup \mathcal{S}}(\mathcal{P})$  et des tétraèdres de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ .

L'algorithme commence avec un ensemble de points  $\mathcal{P}_0$ , qui contient  $\mathcal{Q}$ , au moins deux points sur chaque arête de  $\mathcal{L}$ , et au moins trois points sur chaque morceau de surface de  $\mathcal{S}$ .

L'ensemble de points  $\mathcal{P}$  est augmenté, étape par étape. Lors d'une étape, un nouveau point est ajouté, selon les règles qui suivent. Une règle  $R_i$  n'est appliquée que si aucune règle  $R_j$  avec  $j < i$  ne peut l'être. Les paramètres  $\alpha_1$  et  $\alpha_2$  sont de petites constantes,  $0 < \alpha_1 \leq \alpha_2 \leq 1$  qui seront fixées ultérieurement.

- R1** S'il existe une arête  $L_j \in \mathcal{L}$  et une arête  $e \in \mathcal{D}_{|L_j}(\mathcal{P})$ , dont l'une des extrémités n'est pas sur  $L_j$ , appeler **refine-edge**( $e$ ).
- R2** S'il existe une arête  $L_j \in \mathcal{L}$  et une arête  $e \in \mathcal{D}_{|L_j}(\mathcal{P})$ , dont une boule de Delaunay restreinte  $B(c_e, r_e)$  ne satisfait pas  $r_e \leq \alpha_1 \sigma(c_e)$ , appeler **refine-edge**( $e$ ).
- R3** S'il existe une facette  $S_k \in \mathcal{S}$  et une facette  $f \in \mathcal{D}_{|S_k}(\mathcal{P})$ , dont l'un des sommets n'appartient pas à  $S_k$ , appeler **refine-facet-or-edge**( $f$ ).
- R4** S'il existe une facette  $S_k \in \mathcal{S}$  et une facette  $f \in \mathcal{D}_{|S_k}(\mathcal{P})$ , telle que :
  - R4.1** ou bien une boule de Delaunay restreinte  $B(c_f, r_f)$  circonscrite à  $f$  ne vérifie pas le critère de taille  $r_f \leq \alpha_2 \sigma(c_f)$ ,
  - R4.2** ou bien le rapport rayon-arête  $\rho_f$  ne vérifie pas le critère de forme  $\rho_f \leq \beta_2$ ,
appeler **refine-facet-or-edge**( $f$ ).
- R5** S'il existe un tétraèdre  $t \in \mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ , dont la boule de Delaunay  $B(c_t, r_t)$  le rapport rayon-arête  $\rho_t$  violent
  - R5.1** ou bien le critère de taille  $r_t \leq \sigma(c_t)$ ,
  - R5.2** ou bien le critère de forme  $\rho_t \leq \beta_3$ ,
appeler **refine-tet-facet-or-edge**( $t$ ).

Les procédures auxiliaires **refine-edge**, **refine-facet-or-edge** et **refine-tet-facet-or-edge** sont définies ci-dessous.

**refine-edge** La procédure **refine-edge**( $e$ ) est appelée pour une arête  $e$  de la triangulation de Delaunay  $\mathcal{D}_{|L_j}(\mathcal{P})$  restreinte à une arête  $L_j \in \mathcal{L}$ . La procédure ajoute à  $\mathcal{P}$  le centre  $c_e$  de la boule de Delaunay restreinte  $B(c_e, r_e)$  circonscrite à  $e$  qui a le plus grand rapport rayon.

**refine-facet-or-edge** La procédure **refine-facet-or-edge**( $f$ ) est appelée pour une facette de la triangulation de Delaunay  $\mathcal{D}_{|S_k}(\mathcal{P})$  restreinte à une facette  $S_k \in \mathcal{S}$ . La procédure considère le centre  $c_f$  de la boule de Delaunay restreinte  $B(c_f, r_f)$  circonscrite à  $f$  avec le plus grand rayon,

- si  $c_f$  accroche une arête  $e \in \bigcup_{L_j \in \mathcal{L}} \mathcal{D}_{|L_j}^1(\mathcal{P})$ , appeler **refine-edge**( $e$ ),
- sinon, insérer  $c_f$  dans  $\mathcal{P}$ .

**refine-tet-facet-or-edge** La procédure **refine-tet-facet-or-edge**( $t$ ) est appelée pour un tétraèdre du Delaunay restreint à  $\mathcal{O}$ . Elle considère le centre  $c_t$  de la sphère circonscrite à  $t$  :

- si  $c_t$  accroche une arête  $e \in \bigcup_{L_j \in \mathcal{L}} \mathcal{D}_{|L_j}^1(\mathcal{P})$ , appeler **refine-edge**( $e$ ),
- sinon, si  $c_t$  accroche une facette  $f \in \bigcup_{S_k \in \mathcal{S}} \mathcal{D}_{|S_k}^2(\mathcal{P})$ , appeler **refine-edge**( $f$ ),
- sinon, insérer  $c_t$  dans  $\mathcal{P}$ .

### 4.3 Précision de l'approximation

Dans cette section, il sera admis dans un premier temps que l'algorithme termine. On supposera de plus que la fonction de taille  $\sigma$  vérifie  $\sigma(x) \leq \text{lfs}(x)$  pour tout point  $x \in \mathcal{F}_2$ .

L'objectif de cette partie est de démontrer que, lors que l'algorithme termine, la triangulation de Delaunay restreinte à  $\mathcal{O}$ ,  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ , est un maillage de l'objet  $\mathcal{O}$ , formé de bons éléments, et qui réalise une bonne approximation de  $\mathcal{O}$ .

**Théorème 4.3.1.** *Si l'algorithme termine, et si  $\alpha_2$  est suffisamment petit, alors l'ensemble final de points  $\mathcal{P}$  a les propriétés suivantes :*

- (a) **Forme et taille.** *Les tétraèdres  $t$  de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  vérifient le critère de taille  $r_t \leq \sigma(c_t)$  (où  $B(c_t, r_t)$  est la sphère circonscrite à un tétraèdre  $t$ ), et leurs rapports rayon-arête sont bornés par  $\beta_3$ .*
- (b) **Homéomorphisme.** *L'union des tétraèdres de  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ , notée  $\mathcal{D}_{|\mathcal{O}}^3(\mathcal{P})$ , est homéomorphe à  $\mathcal{O}$ . De plus, il existe un homéomorphisme  $h : \mathcal{O} \rightarrow \bigcup \mathcal{D}_{|\mathcal{O}}^3(\mathcal{P})$ , qui associe toute face de  $\mathcal{F}$  au sous-ensemble de la triangulation de Delaunay qui lui correspond. C'est-à-dire que :*
  - pour tout sommet  $p \in \mathcal{P}$ ,  $h(p) = p$ ,
  - pour toute arête  $L_j \in \mathcal{L}$ ,  $h_{|L_j}$  est un homéomorphisme de  $L_j$  dans  $\bigcup \mathcal{D}_{|L_j}^1(\mathcal{P})$ ,
  - pour toute facette  $S_k \in \mathcal{S}$ ,  $h_{|S_k}$  est un homéomorphisme de  $S_k$  dans  $\bigcup \mathcal{D}_{|S_k}^2(\mathcal{P})$ ,
  - pour toute cellule  $C_l \in \mathcal{C}$ ,  $h_{|C_l}$  est un homéomorphisme de  $C_l$  dans  $\bigcup \mathcal{D}_{|C_l}(\mathcal{P})$ .
- (c) **Distance de Hausdorff.** *Pour toute face  $F \in \mathcal{F}$ , la distance de Hausdorff entre  $\bigcup \mathcal{D}_{|F}(\mathcal{P})$  et  $F$  est  $O(\alpha_2)$ .*

*Démonstration.* La proposition 4.3.1 a est une conséquence directe des règles R5.1 et R5.2. Le reste de cette section 4.3 a pour objectif de démontrer les propositions 4.3.1 b sur l'homéomorphisme (ci-dessous) et 4.3.1 c sur la distance de Hausdorff (voir page 56).  $\square$

## Homéomorphisme

Montrons maintenant la proposition 4.3.1 b. Pour montrer l'homéomorphisme entre  $\mathcal{O}$  et  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ , nous allons utiliser le théorème d'Edelsbrunner et Shah [ES97], dans sa version qui traite le cas des ensembles qui ne sont pas des variétés.

Ce théorème montre un homéomorphisme entre  $\mathcal{O}$  et  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  si  $\mathcal{P}$  vérifie la *propriété de la boule topologique étendue* et la *propriété de l'intersection générique étendue*, définies ci-dessous.

On appelle *boule topologique* de dimension  $k$  tout sous-ensemble de  $\mathbb{R}^3$  homéomorphe à une boule fermée de dimension  $k$ ,  $0 \leq k \leq 3$ .

**Définition 4.3.2** (Propriété de la boule topologique étendue). *Un complexe CW est un complexe régulier dont toutes les faces sont des boules topologiques. Soit  $X \subset \mathbb{R}^d$  un espace topologique. Un ensemble de point  $\mathcal{P}$  a la propriété de la boule topologique étendue par rapport à  $X$  s'il existe un complexe CW  $\mathcal{R}$ , avec  $X = \bigcup \mathcal{R}$ , qui vérifie les propriétés suivantes pour tout sous-ensemble  $T$  de  $\mathcal{P}$  tel que  $V(T) \cap X \neq \emptyset$  :*

- (E1) *il existe un sous-complexe CW  $\mathcal{R}_T \subset \mathcal{R}$ , tel que  $\bigcup \mathcal{R}_T = V(T) \cap X$ ,*
- (E2) *l'ensemble  $\mathcal{R}_T^\circ = \{\gamma \in \mathcal{R} \mid \text{int } \gamma \subset \text{int } V(T)\}$  des faces de  $\mathcal{R}$  dont l'intérieur est inclus dans l'intérieur de  $V(T)$  contient une unique face  $F_T$  qui est contenue dans toutes les faces de  $\mathcal{R}_T^\circ$ ,*
- (E3) *si  $F_T$  est une face de dimension  $j$ , alors  $F_T \cap \text{bd } V(T)$  est une sphère topologique de dimension  $j - 1$ , et*
- (E4) *pour tout entier  $k$  et pour toute face  $F \in \mathcal{R}_T^\circ \setminus \{F_T\}$  de dimension  $k$ ,  $F \cap \text{bd } V(T)$  est une boule topologique de dimension  $k - 1$ .*

On dit de plus que  $\mathcal{P}$  a la propriété de l'intersection générique étendue par rapport à  $X$  si pour tout sous-ensemble  $T$  de  $\mathcal{P}$ , et tout face  $F \in \mathcal{R}_T \setminus \mathcal{R}_T^\circ$ , il existe une face  $G \in \mathcal{R}_T^\circ$ , telle que  $F \subset G$ .

**Théorème 4.3.3** (Théorème 3 de [ES97]). *Soient  $X \subset \mathbb{R}^d$  un espace topologique, et  $\mathcal{P} \subset \mathbb{R}^3$  un ensemble fini de points qui a la propriété de l'intersection générique étendue par rapport à  $X$ . Si  $\mathcal{P}$  a la propriété de la boule topologique étendue par rapport à  $X$ , alors  $\bigcup \mathcal{D}_{|X}(\mathcal{P})$  est homéomorphe à  $X$ .*

La suite de la section 4.3 montre que l'ensemble de points  $\mathcal{P}$  créé par l'algorithme de la section 4.2 est tel que  $\mathcal{P}' = \mathcal{P} \cap \bigcup \mathcal{S}$  a la propriété de la boule topologique étendue pour  $\mathcal{O}$ . Il n'est possible, pour des raisons techniques, de montrer la propriété de la boule topologique étendue pour  $\mathcal{P}$ . Néanmoins, la propriété de la boule topologique étendue pour  $\mathcal{P}'$  va s'avérer suffisant pour montrer les propriétés d'homéomorphie qui nous intéressent.

Pour cela, on définit  $\mathcal{R} = \{V(T) \cap F \mid T \subset \mathcal{P}', F \in \mathcal{F}\}$ . Nous allons montrer que le complexe  $\mathcal{R}$  ainsi défini est un complexe CW ayant toutes les propriétés de la définition de la propriété de la boule topologique étendue. Nous supposons, de plus, que  $\mathcal{P}'$  a la propriété de l'intersection générique étendue avec  $\mathcal{O}$ . Cette propriété de généricité peut être assurée par une petite perturbation de la surface qui ne modifie pas la triangulation  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P}')$ . Le théorème 4.3.3 montrera alors un homéomorphisme entre  $\mathcal{O}$  et  $\bigcup \mathcal{D}_{|\mathcal{O}}(\mathcal{P}')$ . Pour montrer la proposition 4.3.1 b, il suffira alors de montrer que  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P}')$  et  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$  ont exactement les mêmes triangles, et que  $\bigcup \mathcal{D}_{|\mathcal{O}}(\mathcal{P}')$  est le même objet que  $\bigcup \mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ . La première étape de cette démarche est de montrer que chaque face du complexe  $\mathcal{R}$  défini ci-dessus est une boule topologique.

## Échantillonnage de surface

Pour commencer, on rappelle quelques lemmes de la théorie de l'échantillonnage de surface [AB99, BO05]. Ces lemmes sont ici adaptés à notre situation, où les facettes  $S_k \in \mathcal{S}$  sont des morceaux de surfaces lisses fermées. On rappelle que les angles entre des droites et des vecteurs sont en fait des angles de droites (voir les précisions section 2.2 page 17).

**Lemme 4.3.4** (Lemme topologique [AB99]). *Soit  $x$  un point de  $\bigcup \mathcal{S}$ , toute boule  $B(x, r)$  centrée en  $x$  et de rayon  $r \leq \text{lfs}(x)$  intersecte toute face  $F$  de  $\mathcal{F}_2$  contenant  $x$  selon un disque topologique de même dimension que  $F$ .*

**Lemme 4.3.5** (Distance longue [Dey06]). *Soit  $x$  un point d'une facette  $S_k \in \mathcal{S}$ . Si une ligne droite  $l$  passant par  $x$  forme un angle  $(l, \mathbf{n}_x) \leq \eta$  avec une normale  $\mathbf{n}_x$  à  $S_k$  en  $x$ , et intersecte  $S_k$  en un deuxième point  $y$  distinct de  $x$ , alors  $d(x, y) \geq 2 \text{lfs}(x) \cos(\eta)$ .*

**Lemme 4.3.6** (Angle avec la corde [AB99]). *Soit  $S_k \in \mathcal{S}$ . Pour tout point  $x$  et  $y$  de  $S_k$ , tels que  $d(x, y) \leq \eta \text{lfs}(x)$ ,  $\eta \leq 2$ , l'angle entre  $xy$  et  $H_x$ , le plan tangent à  $S_k$  en  $x$ , est au plus  $\arcsin \frac{\eta}{2}$ .*

**Lemme 4.3.7** (Variation de la normale [AB99]). *Soient  $x$  et  $y$  deux points de  $S_k$  tels que  $d(x, y) \leq \eta \min(\text{lfs}(x), \text{lfs}(y))$ , avec  $\eta < \frac{1}{3}$ , et  $\mathbf{n}_x$  et  $\mathbf{n}_y$  deux vecteurs unitaires respectivement normaux à  $S_k$  en  $x$  et  $y$ , orientés selon l'orientation de la surface fermée qui contient  $S_k$ . Alors l'angle  $(\mathbf{n}_x, \mathbf{n}_y)$  est au plus  $\frac{\eta}{1-3\eta}$ .*

**Lemme 4.3.8** (Normale à une facette [AB99]). *Soit  $\mathcal{P}$  un ensemble de points, et  $pqr$  un triangle de  $\mathcal{D}_{|S_k}(\mathcal{P})$ , avec une boule de Delaunay surfacique  $B(c, \rho)$ , telle que  $\rho \leq \eta \text{lfs}(c)$ . Si  $p$  est le sommet de  $pqr$  avec le plus grand angle au sommet, la direction  $n(pqr)$  normale à  $pqr$ , et les vecteurs unitaires  $\mathbf{n}_p$ ,  $\mathbf{n}_q$  et  $\mathbf{n}_r$  normaux à  $S_k$  respectivement en  $p$ ,  $q$  et  $r$  sont tels que  $(n(pqr), \mathbf{n}_p) \leq \arcsin(\frac{\sqrt{3}\eta}{1-\eta})$  et  $\max((n(pqr), \mathbf{n}_q), (n(pqr), \mathbf{n}_r)) \leq \arcsin(\frac{\sqrt{3}\eta}{1-\eta}) + \frac{2\eta}{1-7\eta}$ .*

Les résultats de Nina Amenta et Marshall Bern [AB99] sur l'échantillonnage de surfaces lisses s'étendent à nos morceaux de surfaces lisses  $S_k \in \mathcal{S}$ , et notamment nous avons le lemme suivant :

**Lemme 4.3.9** ( $\varepsilon$ -échantillon). *Si  $\mathcal{P}$  est un  $\varepsilon$ -échantillon d'un morceau de surface  $S_k \in \mathcal{S}$  alors, pour tout point  $p \in \mathcal{P}$ , l'intersection  $V(p) \cap S_k$  de sa cellule de Voronoï avec  $S_k$  est incluse dans la boule  $B(p, \frac{\varepsilon}{1-\varepsilon} \text{lfs}(p))$ , de centre  $p$  et de rayon  $\frac{\varepsilon}{1-\varepsilon} \text{lfs}(p)$ .*

À la fin de l'algorithme, l'ensemble de point  $\mathcal{P}$  est tel que pour tout morceau de surface  $S_k \in \mathcal{S}$ ,  $\mathcal{P} \cap S_k$  est un  $\alpha_2$ -échantillon lâche (voir la définition 3.2.4 page 23). Steve Oudot et Jean-Daniel Boissonnat [BO05, BO06] ont étudié les  $\varepsilon$ -échantillons lâches de surfaces lisses, et l'un de leurs résultats principaux réside dans le fait qu'un  $\varepsilon$ -échantillon lâche est  $\varepsilon'$ -échantillon, avec  $\varepsilon' = \varepsilon(1 + O(\varepsilon))$ . La définition de notre fonction  $\text{lfs}$  permet d'adapter ce résultat aux morceaux de surfaces lisses  $S_k \in \mathcal{S}$ . La plupart des lemmes intermédiaires de leur théorie sont aussi applicables à notre cas. Les lemmes suivants en font partie.

**Lemme 4.3.10** (Lemme de projection [Boi06]). *Si  $\mathcal{P}$  est un  $\varepsilon$ -échantillon du morceau de surface  $S_k$ , avec  $\varepsilon < 0,24$ , alors pour toute paire de facettes  $f$  et  $f'$  de  $\mathcal{D}_{|S_k}(\mathcal{P})$  partageant un sommet  $p$ , elles ont des projections orthogonales sur le plan tangent  $H_p(S_k)$  en  $p$  dont les intérieurs relatifs ne s'intersectent pas.*

**Lemme 4.3.11** (Lemme des  $\varepsilon$ -échantillons lâches [BO05, Boi06]). *Si  $\varepsilon < 0,09$ , tout  $\varepsilon$ -échantillon lâche d'un morceau de surface  $S_k \in \mathcal{S}$  est un  $\varepsilon'$ -échantillon de  $S_k$ , avec  $\varepsilon' = \varepsilon(1 + O(\varepsilon))$ .*

Les lemmes 4.3.10 et 4.3.11 sont encore applicables si la fonction  $\text{lfs}$  est remplacée par toute fonction 1-Lipschitz  $\sigma$ , telle que  $\sigma(x) \leq \text{lfs}(x)$  en tous points de  $S_k$ .

Dès que les règles de R1 à R4 ne peuvent plus être appliquées (parce que leurs critères sont satisfaits), et notamment à la fin de l'algorithme, l'ensemble de points  $\mathcal{P}$  est tel que, pour tout morceau de surface  $S_k$ ,  $\mathcal{P} \cap S_k$  est un  $\alpha_2$ -échantillon de  $S_k$ . Le lemme précédent permet ainsi de dire que  $\mathcal{P} \cap S_k$  est en fait un  $\varepsilon$ -échantillon pour  $\varepsilon = \alpha_2(1 + O(\alpha_2))$ . Cette propriété sera utilisée dans les démonstrations présentes dans la suite de ce chapitre. On suppose par la suite que  $\alpha_2$  est suffisamment petit pour que  $\varepsilon < 0,09$ , de sorte que les lemmes 4.3.9 et 4.3.10 peuvent être appliqués.

### Vérification des propriétés topologiques

Les lemmes suivants s'appliquent à l'ensemble de point  $\mathcal{P}' = \mathcal{P} \cap \bigcup \mathcal{S}$ . Ils supposent que le champs de taille  $\sigma$  est tel que  $\sigma(x) \leq \text{lfs}(x)$  pour tout point  $x$  de  $\bigcup \mathcal{F}$ , et que les constantes  $\alpha_1$  et  $\alpha_2$  sont suffisamment petites.

**Lemme 4.3.12** (Segments de courbes). *Soit un segment de courbe  $L_j \in \mathcal{L}$ .*

- (a) *Soit  $V(p, q)$  une facette de  $\mathcal{V}(\mathcal{P}')$ . Alors l'intersection  $V(p, q) \cap L_j$  est non vide si et seulement si  $p$  et  $q$  sont deux sommets consécutifs sur  $L_j$ . Par conséquent, une facette de  $\mathcal{V}(\mathcal{P}')$  intersecte au plus un segment de courbe de  $\mathcal{L}$ .*  
*De plus, si l'intersection  $V(p, q) \cap L_j$  est non vide, alors elle est réduite à un seul point, c'est-à-dire à une boule topologique de dimension 0.*
- (b) *Soit  $V(p)$  une cellule de Voronoï.  $L_j$  intersecte  $V(p)$  si et seulement si le sommet  $p$  appartient à  $L_j$ . De plus, si  $V(p) \cap L_j$  est non vide, alors c'est une boule topologique de dimension 1.*

*Démonstration de la proposition 4.3.12 a.* Soient  $p$  et  $q$  deux sommets de  $\mathcal{P}'$  tels que  $L_j$  intersecte la facette  $V(p, q)$ . On note  $e$  l'arête  $pq$ . Soit  $c_e = V(p, q) \cap L_j$ .  $c_e$  est le centre d'une boule linéique  $B(c_e, r_e)$  circonscrite à  $e$  (voir figure 4.2). D'après la règle R2, le rayon  $r_e$  est inférieur à  $\text{lfs}(c_e)$ . Ainsi, selon le lemme topologique 4.3.4,  $B(c_e, r_e) \cap L_j$  est un disque topologique. Ce disque topologique est la portion de courbe  $L_j(p, q)$  incluse dans  $L_j$  qui joint  $p$  à  $q$ . Par conséquent,  $L_j(p, q)$  est inclus dans  $B(c_e, r_e)$ , qui est une boule d'intérieur vide de sommets de  $\mathcal{P}'$ , et  $L_j(p, q)$  ne contient pas d'autres sommets que ses extrémités  $p$  et  $q$ . Cela veut dire que  $p$  et  $q$  sont consécutifs sur  $L_j$ . Inversement, si  $p$  et  $q$  sont deux sommets de  $\mathcal{P}'$  consécutifs sur  $L_j$ , la portion  $L_j(p, q)$  de  $L_j$  entre  $p$  et  $q$  ne peut intersecter d'autre facette de Voronoï que  $V(p, q)$ . Par conséquent,  $L_j(p, q)$  est incluse dans  $V(p) \cup V(q)$ , et doit intersecter  $V(p, q)$ .

Montrons maintenant que, si l'intersection  $V(p, q) \cap L_j$  est non vide, alors cette intersection est réduite à un point. Si l'intersection est non vide, alors seule la portion  $L_j(p, q)$  de  $L_j$  comprise entre  $p$  et  $q$  peut intersecter  $V(p, q)$ . Le segment de courbe  $L_j(p, q)$  est inclus dans la boule de Delaunay linéique  $B(c_e, r_e)$ . Selon la règle R2, le rayon  $r_e$  est inférieur à  $\alpha_1 \text{lfs}(c_e)$ . Si le segment de courbe  $L_j(p, q)$  intersecte  $V(p, q)$  plus d'une fois, il doit exister un point  $x$  de



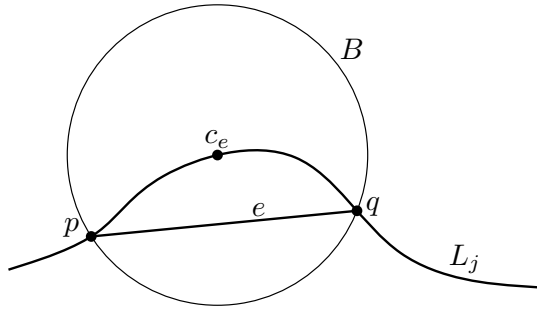


FIG. 4.2: L'intersection d'une boule linéique avec  $L_j$ .

$L_j(p, q)$  avec un rayon de courbure plus petit que  $r_e \leq \alpha_1 \text{lfs}(c_e)$ . Comme  $\text{lfs}$  est une fonction 1-Lipschitz, pour tout point  $x$  de  $L_j(p, q)$ ,  $\text{lfs}(x) \geq \text{lfs}(c_e)(1 - \alpha_1)$ . En tout point  $x$  de  $L_j$ , le rayon de courbure est nécessairement plus grand que  $\text{lfs}(x)$ . Nous avons donc une contradiction, si  $\alpha_1 \leq 1/2$ .  $\square$

*Démonstration de la proposition 4.3.12 b.* Tout segment de courbe  $L_j \in \mathcal{L}$  intersecte au minimum les cellules de Voronoï de ses deux extrémités, et ne peut donc être entièrement inclus dans une cellule de Voronoï. Ainsi, si  $L_j$  intersecte une cellule de Voronoï  $V(p)$ , alors il doit intersecter une facette  $V(p, q)$  du bord de  $V(p)$ . Selon la proposition 4.3.12 a,  $p$  doit nécessairement être sur  $L_j$ .

Si un sommet  $p \in \mathcal{P}'$  appartient à  $L_j$ , alors  $L_j$  intersecte la cellule de Voronoï  $V(p)$ . La proposition 4.3.12 a implique que  $L_j \cap \text{bd } V(p)$  est soit restreint à un point, si  $p$  est une extrémité de  $L_j$ , soit exactement deux points, si  $p$  n'est pas une extrémité de  $L_j$ . Dans les deux cas,  $L_j \cap V(p)$  est un segment de courbe, c'est-à-dire une boule topologique de dimension 1.  $\square$

**Lemme 4.3.13** (Morceaux de surfaces). *Soit un morceau de surface  $S_k \in \mathcal{S}$ .*

- (a) *Soit  $V(p, q, r)$  une arête de  $\mathcal{V}(\mathcal{P}')$ . Si  $V(p, q, r)$  intersecte  $S_k$ , alors les sommets  $p, q$ , et  $r$  appartiennent à  $S_k$ . Une arête de  $\mathcal{V}(\mathcal{P}')$  intersecte au plus un morceau de surface de  $\mathcal{S}$ . Si elle est non vide, l'intersection  $S_k \cap V(p, q, r)$  est réduite à un point, c'est donc une boule topologique de dimension 0.*
- (b) *Soit  $V(p, q)$  une facette de  $\mathcal{V}(\mathcal{P}')$ . Si  $V(p, q)$  intersecte  $S_k$ , alors les sommets  $p$  et  $q$  appartiennent à  $S_k$ . Si elle est non vide, l'intersection  $S_k \cap V(p, q)$  est un segment de courbe, c'est-à-dire une boule topologique de dimension 1.*
- (c) *Soit  $V(p)$  une cellule de  $\mathcal{V}(\mathcal{P}')$ . L'intersection  $S_k \cap V(p)$  est non vide si et seulement si  $p$  appartient à  $S_k$ . De plus, si  $S_k \cap V(p)$  est non vide, alors c'est une boule topologique de dimension 2.*

*Démonstration de la proposition 4.3.13 a.* Si une arête  $V(p, q, r)$  de  $\mathcal{V}(\mathcal{P}')$  intersecte un morceau de surface  $S_k$ , alors le triangle  $pqr$  est une facette  $f$  de  $\mathcal{D}_{|S_k}(\mathcal{P}')$ . D'après la règle R3, les sommets  $p, q$ , et  $r$  appartiennent donc à  $S_k$ .

Pour montrer que  $V(p, q, r)$  n'intersecte qu'un seul morceau de surface, supposons, pour trouver une contradiction, que  $V(p, q, r)$  intersecte plus d'une facette de  $\mathcal{S}$ . Soient alors  $S_k$  et  $S_j$  deux facettes de  $\mathcal{S}$ , et  $c_j$  et  $c_k$  deux points

d'intersection consécutifs sur  $V(p, q, r)$ , tels que  $c_j \in S_j$  et  $c_k \in S_k$ . La règle R3 implique que  $p, q$  et  $r$  sont à la fois sur  $S_j$  et sur  $S_k$ , qui sont donc des facettes adjacentes dans  $\mathcal{F}_2$  (voir figure 4.3). Supposons, sans perdre la généralité, que  $p$

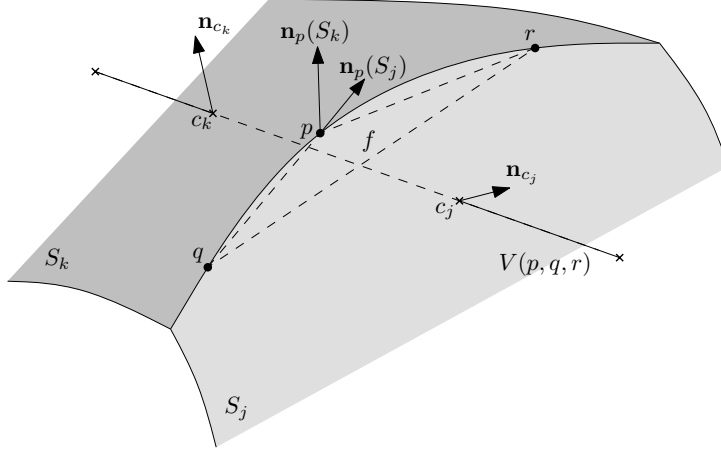


FIG. 4.3: Pour la démonstration de la proposition 4.3.13 a.

est le sommet du triangle  $pqr$  avec le plus grand angle au sommet. On considère les vecteurs unitaires  $\mathbf{n}_{c_j}$  et  $\mathbf{n}_p(S_j)$  respectivement normaux à  $S_j$  en  $c_j$  et  $p$ . Cela suppose avoir choisi une orientation pour  $S_j$ . On choisit une orientation pour  $S_k$ , cohérente avec celle choisie pour  $S_j$ , et on considère les vecteurs unitaires  $\mathbf{n}_{c_k}$  et  $\mathbf{n}_p(S_k)$  respectivement normaux à  $S_k$  en  $c_k$  et  $p$ . Les orientations de  $S_j$  et  $S_k$  sont cohérentes si l'angle entre  $\mathbf{n}_p(S_j)$  et  $\mathbf{n}_p(S_k)$  est inférieur à  $\pi/2$ . Enfin, on notera  $n(f)$  la *direction* normale à la facette  $f = pqr$ . (On ne considère pas un vecteur normal à  $f$  car on ne peut pas orienter la normale à  $f$  de manière cohérente avec les orientations de  $S_j$  et  $S_k$ .) La règle R4.1 implique que le rayon de la plus petite sphère circonscrite à  $f$  est inférieur à  $\alpha_2 \min(\text{lfs}(c_j), \text{lfs}(c_k))$ , et alors, selon le lemme de la normale à une face 4.3.8, les angles  $(n(f), \mathbf{n}_p(S_j))$  et  $(n(f), \mathbf{n}_p(S_k))$  sont inférieurs à  $\arcsin(\frac{\sqrt{3}\alpha_2}{1-\alpha_2})$ . Par conséquent, l'angle  $(\mathbf{n}_p(S_j), \mathbf{n}_p(S_k))$  est soit  $O(\alpha_2)$  soit  $\pi - O(\alpha_2)$ . La seconde possibilité contredirait l'hypothèse angulaire, donc  $(\mathbf{n}_p(S_j), \mathbf{n}_p(S_k)) = O(\alpha_2)$ . La règle R4.1 implique aussi que  $d(p, c_j) \leq \alpha_2 \text{lfs}(c_j)$  et  $d(p, c_k) \leq \alpha_2 \text{lfs}(c_k)$ , et alors le lemme de variation de la normale 4.3.7 implique que les angles  $(\mathbf{n}_{c_j}, \mathbf{n}_p(S_j))$  et  $(\mathbf{n}_{c_k}, \mathbf{n}_p(S_k))$  sont des  $O(\alpha_2)$ . On a :

$$(\mathbf{n}_{c_j}, \mathbf{n}_{c_k}) \leq (\mathbf{n}_{c_j}, \mathbf{n}_p(S_j)) + (\mathbf{n}_p(S_j), \mathbf{n}_p(S_k)) + (\mathbf{n}_{c_k}, \mathbf{n}_p(S_k)).$$

Ainsi  $(\mathbf{n}_{c_j}, \mathbf{n}_{c_k})$  est un  $O(\alpha_2)$ . Or on a aussi

$$(\mathbf{n}_{c_j}, n(f)) \leq (\mathbf{n}_{c_j}, \mathbf{n}_p(S_j)) + (\mathbf{n}_p(S_j), n(f)) = O(\alpha_2),$$

et de la même façon  $(\mathbf{n}_{c_k}, n(f)) = O(\alpha_2)$ . Ainsi,  $\mathbf{n}_{c_j}$  et  $\mathbf{n}_{c_k}$  forment chacun un angle  $O(\alpha_2)$  avec la direction  $n(f)$ . Comme  $n(f)$  est la direction de  $V(p, q, r)$ , et  $c_j$  et  $c_k$  sont deux points d'intersection consécutifs le long de  $V(p, q, r)$ , cela implique que ces deux vecteurs normaux sont de direction proche, et de sens opposés. Ceci est en contradiction avec la propriété  $(\mathbf{n}_{c_j}, \mathbf{n}_{c_k}) = O(\alpha_2)$  démontrée ci-dessus. Ainsi,  $V(p, q)$  ne peut pas intersecter deux facettes différentes de  $\mathcal{S}$ .

Le raisonnement précédent est valide aussi si  $c_j$  et  $c_k$  sont deux points d'intersection consécutifs distincts avec la même facette  $S_j = S_k$ . Cela montre qu'une arête  $V(p, q, r)$  ne peut pas intersecter deux fois la même facette de  $\mathcal{S}$ . Ainsi, l'intersection entre une arête  $V(p, q, r)$  et un morceau de surface de  $\mathcal{S}$  est soit vide soit restreinte à un point.  $\square$

La figure 4.4 montre différentes configurations possibles, pour l'intersection entre une facette de Voronoï  $V(p, q)$  et une facette  $S_k$  de  $\mathcal{S}$ . Afin de démontrer la proposition 4.3.13 b, il faut au préalable démontrer le lemme suivant, qui affirme que la configuration de gauche est impossible.

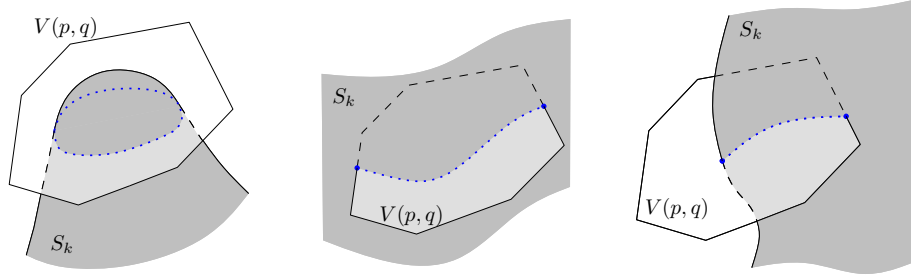


FIG. 4.4: Pour la démonstration de la proposition 4.3.13 b. Différentes configurations d'intersection entre une facette de Voronoï  $V(p, q)$  et un morceau de surface  $S_k$ , si  $p$  et  $q$  appartiennent tous deux à  $S_k$ . Dans les différentes configurations, l'intersection est représentée en bleu, par des pointillés fins. À gauche, l'intersection est une courbe fermée. Au milieu et à droite, l'intersection est un segment de courbe.

**Lemme 4.3.14.** *Soit  $V(p, q)$  une facette de  $\mathcal{V}(\mathcal{P}')$ , et  $S_k$  une facette de  $\mathcal{S}$ . Si  $p$  et  $q$  appartiennent à  $S_k$ , alors l'intersection  $V(p, q) \cap S_k$  n'inclut aucune courbe fermée.*

*Démonstration.* Supposons, par contradiction, que  $p$  et  $q$  appartiennent à  $S_k$ , et que  $V(p, q) \cap S_k$  contient une courbe fermée  $\gamma$ . Soit  $x$  un point de  $\gamma$ , et  $l$  la ligne qui appartient au plan  $h$  support de  $V(p, q)$ , et normal à  $\gamma$  en  $x$  (voir figure 4.5). La ligne  $l$  est la projection de  $\mathbf{n}_x$ , un vecteur normal à  $S_k$  en  $x$ , sur le plan  $h$ . Ainsi  $(\mathbf{n}_x, l) \leq (\mathbf{n}_x, l')$  pour toute ligne  $l'$  de  $h$ . Soit alors  $l'$  la ligne passant par  $x$ , et parallèle à la projection de  $\mathbf{n}_p$ , un vecteur normal à  $p$  en  $S_k$ , sur  $h$ . Comme  $\mathcal{P}'$  est un  $\alpha_2$ -échantillon lâche de  $S_k$ , c'est un  $O(\alpha_2)$ -échantillon de  $S_k$ , d'après le lemme 4.3.11, et donc  $d(x, p) = O(\alpha_2)\text{lfs}(p)$ , et  $d(p, q) \leq 2d(p, x) = O(\alpha_2)\text{lfs}(p)$ . Alors, d'après le lemme 4.3.7 de la variation de la normale,  $(n_x, n_p) = O(\alpha_2)$ , et d'après le lemme 4.3.6 de l'angle avec la corde,  $(n_p, pq)$  est au moins  $\frac{\pi}{2} - O(\alpha_2)$ , et

$$(\mathbf{n}_p, l') = \frac{\pi}{2} - (n_p, pq) = O(\alpha_2)$$

. Alors,

$$(\mathbf{n}_x, l) \leq (\mathbf{n}_x, l') \leq (n_x, n_p) + (n_p, l') = O(\alpha_2).$$

La ligne  $l$  intersecte nécessairement  $\gamma$  en un deuxième point  $y$ , distinct de  $x$ . alors le lemme 4.3.5 de la distance longue implique que  $d(x, y)$  est au moins

$(1 - O(\alpha_2))\text{lfs}(p)$ , ce qui contredit le résultat du lemme d'échantillonnage 4.3.9, qui implique que  $x$  et  $y$  sont tous deux contenus dans une boule de rayon  $O(\alpha_2)$ .

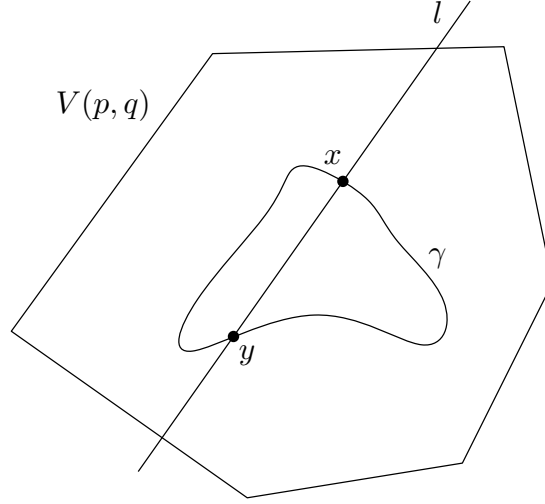


FIG. 4.5: Pour la preuve du lemme 4.3.14.

□

*Démonstration de la proposition 4.3.13 b.* Soit  $x$  un point de l'intersection  $V(p, q) \cap S_k$ . Si  $p$  n'appartient pas à  $S_k$  et n'est pas corrélé avec  $x$ , alors la distance  $d(p, x)$  est supérieure à  $\text{lfs}(p)$ , ce qui contredit le lemme d'échantillonnage `lem :volmesh2-epsilon-sample`. Si  $p$  n'appartient pas à  $S_k$ , mais est corrélé avec  $x$ , alors  $p$  appartient à un morceau de surface  $S_i$ , adjacent à  $S_k$ . Par hypothèse angulaire,  $d(p, S_i \cap S_k) < d(p, x)$ . Or, les arêtes de  $S_i \cap S_k$  sont finement échantillonnées. La boule de centre  $p$  et de rayon  $d(x, p)$  contient donc un point de l'échantillon  $\mathcal{P}' \cap S_j \cap S_k$ . Cela contredit le fait que  $p$  est le point de  $\mathcal{P}'$  le plus proche de  $x$ . Finalement,  $p$  est nécessairement sur  $S_k$ .

Montrons maintenant que, si elle est non vide, l'intersection  $V(p, q) \cap S_k$  est une boule topologique de dimension 1. Comme cette intersection ne contient aucune courbe fermée (lemme 4.3.14), elle est composée de segments de courbes, avec donc deux extrémités chacun, et  $\text{bd}(V(p, q) \cap S_k)$  est un ensemble de points de cardinal pair. Notons que

$$\text{bd}(V(p, q) \cap S_k) = (\text{bd} V(p, q) \cap S_k) \cup (V(p, q) \cap \text{bd} S_k).$$

D'après la proposition 4.3.13 a, chaque arête de Voronoï de  $\text{bd} V(p, q)$  intersecte  $S_k$  en au plus un point. Le lemme de projection 4.3.10 implique qu'il y a au plus deux facettes de  $\mathcal{D}_{|S_k}(\mathcal{P}')$  incidente à une arête donnée, et par conséquent  $\text{bd} V(p, q) \cap S_k$  inclut au plus deux points. De plus, d'après la proposition 4.3.12 a sur les segments de courbes,  $V(p, q) \cap \text{bd} S_k$  contient au plus un point. Ainsi, le nombre de points d'intersections dans  $\text{bd}(V(p, q) \cap S_k)$  est au plus trois. Comme ce nombre est nécessairement pair, il vaut zéro ou deux. Finalement, on peut conclure que l'intersection est soit vide, soit un segment de courbe, c'est-à-dire une boule topologique de dimension 2. □

*Démonstration de la proposition 4.3.13 c.* Soit  $p \in \mathcal{P}'$ , tel que  $V(p)$  intersecte  $S_k$ . Montrons que  $p$  appartient à  $S_k$ . Le morceau de surface  $S_k$  contient au minimum trois sommets de  $\mathcal{Q}$ , et ne peut donc être inclus entièrement dans l'intérieur de  $V(p)$ . Donc  $S_k$  intersecte au moins une facette  $V(p, q)$  du bord de  $V(p)$ . D'après la proposition 4.3.13 b,  $p$  appartient à  $S_k$ .

Maintenant, montrons que, si  $p \in S_k$ , alors  $V(p) \cap S_k$  est une boule topologique de dimension 2. Il suffit pour cela de montrer que son bord  $\text{bd}(V(p) \cap S_k)$  est une sphère topologique. En effet, comme  $\mathcal{P}' \cap S_k$  est un  $\varepsilon$ -échantillon de  $S_k$ , les lemmes 4.3.9 et 4.3.7 montrent que la normale  $\mathbf{n}_x$  à  $S_k$  en un point  $x \in V(p) \cap S_k$  est proche de la normale  $\mathbf{n}_p$  à  $S_k$  en  $p$ . Ainsi,  $V(p) \cap S_k$  est un terrain au dessus du plan tangent  $H_p$  à  $S_k$  en  $p$ , et une sphère topologique de dimension 1, sur un tel terrain, borne une boule topologique de dimension 2.

Supposons, pour commencer, que  $p$  appartient à l'intérieur de  $S_k$ . Alors, d'après le lemme 4.3.12,  $V(p)$  n'intersecte aucune arête de  $\mathcal{L}$ , et donc l'intersection  $V(p) \cap \text{bd} S_k$  est vide. D'après la proposition 4.3.13 b, l'intersection de  $S_k$  avec une facette  $V(p, q)$  de  $\text{bd} V(p)$  est soit vide soit une boule topologique de dimension 1 (c'est à dire un segment de courbe) dont les deux extrémités sont sur des arêtes  $V(p, q, r)$  de  $V(p)$ . Ainsi, l'intersection  $S_k \cap \text{bd} V(p)$  est l'union d'un ensemble de segments de courbes, qui forment des cycles sur  $\text{bd} V(p)$ . Chacun de ces cycles correspond, dans  $\mathcal{D}_{|S_k}(\mathcal{P}')$ , à un cycle de triangles adjacents, et incidents au sommet  $p$ , qui forment une boule topologique autour de  $p$ . Alors, le lemme de projection 4.3.10 implique qu'il existe dans  $\mathcal{D}_{|S_k}(\mathcal{P}')$  au plus un tel cycle de triangles adjacents, et incidents à  $p$ , et donc un seul cycle dans  $S_k \cap \text{bd} V(p)$ . Par conséquent,  $\text{bd}(V(p) \cap S_k) = S_k \cap \text{bd} V(p)$  est une sphère topologique de dimension 1. Cela implique que  $V(p) \cap S_k$  est une boule topologique de dimension 2.

Si maintenant on suppose que  $p$  appartient à l'intérieur d'une arête  $L_j$  du bord de  $S_k$ , on sait, d'après le lemme 4.3.12, que la cellule  $V(p)$  intersecte  $L_j$  mais aucun autre segment de courbe de  $\mathcal{L}$ . Cela signifie que  $V(p) \cap \text{bd} S_k$  est réduit à  $V(p) \cap L_j$ , qui est une boule topologique de dimension 1. Chacune des deux facettes de  $V(p)$  qui intersectent  $L_j$  intersecte  $S_k$  selon une boule topologique de dimension 1, avec une extrémité sur  $L_j$ , et une extrémité sur une arête de  $V(p)$ . Tout autre intersection non vide  $V(p, q) \cap S_k$  de  $S_k$  avec une facette de  $V(p)$  est une boule topologique de dimension 1, avec les deux extrémités sur des arêtes de  $V(p)$ . Parmi les intersections non vide  $V(p, q) \cap S_k$  entre les facettes de  $V(p)$  et  $S_k$ , certaines de ces intersections forment ainsi une chaîne de segments de courbes, cette chaîne ayant ses deux extrémités sur  $L_j \cap \text{bd} V(p)$ . Le reste des intersections non vide de  $S_k \cap \text{bd} V(p)$  forment nécessairement des cycles. Le lemme de projection 4.3.10 montre qu'il n'y a, en fait, aucun cycle dans  $S_k \cap S_k$ . Par conséquent,  $S_k \cap \text{bd} V(p)$  est restreint à une chaîne de segments de courbes. Comme, de plus,  $V(p) \cap \text{bd} S_k = V(p) \cap L_j$  est un segment de courbe.  $\text{bd}(S_k \cap V(p)) = (S_k \cap \text{bd} V(p)) \cup (V(p) \cap \text{bd} S_k)$  est une sphère topologique de dimension 1. Ceci implique, comme on l'a vu, que  $S_k \cap V(p)$  est une boule topologique de dimension 2.

Enfin, si  $p$  est un sommet de  $S_k$ , alors  $p$  est l'extrémité commune de deux arêtes  $L_i$  et  $L_j$  de  $\text{bd} S_k$ . Dans cette situation,  $V(p) \cap L_i$  et  $V(p) \cap L_j$  sont des boules topologiques de dimension 1, avec  $p$  comme extrémité commune, et  $V(p) \cap \text{bd} L_i$  (ou respectivement  $V(p) \cap \text{bd} L_j$ ) comme deuxième extrémité. De plus,  $\text{bd} V(p) \cap S_k$  est une chaîne de segments de courbes (comme dans les cas précédents) joignant  $V(p) \cap \text{bd} L_i$  à  $V(p) \cap \text{bd} L_j$ , avec éventuellement des cycles

de segments de courbes en plus. Encore une fois, le lemme de projection 4.3.10 indique qu'il n'y a pas de cycles dans  $\text{bd } V(p) \cap S_k$ . Ainsi,  $\text{bd } (V(p) \cap S_k)$  est une boule topologique de dimension 1, obtenue comme la concaténation de  $V(p) \cap L_i$ ,  $V(p) \cap L_j$  et la chaîne des intersections non vides  $V(p, q) \cap S_k$  entre  $S_k$  et les facettes de  $\text{bd } V(p)$ .  $\square$

**Lemme 4.3.15** (Cellules). *Soit  $C_l$  une cellule de  $\mathcal{C}$ .*

- (a) *Soit  $V(p)$  une cellule de  $\mathcal{V}(\mathcal{P}')$ . L'intersection  $V(p) \cap C_l$  est non vide si et seulement si  $p \in C_l$ .*
- (b) *L'intersection de  $C_l$  avec toute face de Voronoï (arête, facette ou cellule) de  $\mathcal{V}(\mathcal{P}')$  est une boule topologique.*

*Démonstration de la proposition 4.3.15 a.* D'après le lemme 4.3.13, la cellule  $V(p)$  intersecte le bord de  $C_l$  si et seulement si  $p$  est sur le bord de  $C_l$ . Ainsi, si  $p$  appartient à l'intérieur de  $C_l$ , la cellule  $V(p)$  est tout entière incluse dans l'intérieur de  $C_l$ , et si  $p$  n'appartient pas à  $C_l$ , alors  $V(p)$  et  $C_l$  sont disjoints.  $\square$

*Démonstration de la proposition 4.3.15 b.* – Soit  $V(p, q, r)$  une arête de Voronoï. La proposition 4.3.13 a assure que  $V(p, q, r)$  intersecte le bord de  $C_l$  en au plus un point. Par conséquent,  $V(p, q, r) \cap C_l$  est soit vide, soit  $V(p, q, r)$  tout entier, soit un segment inclus dans  $V(p, q, r)$ , avec une extrémité sur  $\text{bd } C_l$ . Ainsi,  $V(p, q, r) \cap C_l$  est soit vide soit une boule topologique de dimension 1.

- Soit  $V(p, q)$  une facette de Voronoï. Considérons l'intersection  $V(p, q) \cap \text{bd } C_l$ . Selon les lemmes 4.3.12 et 4.3.13, plusieurs cas sont possibles, en fonction de la location de  $p$  et  $q$  par rapport au bord  $\text{bd } C_l$ .
  - Si  $p$  et  $q$  n'appartiennent au même morceau de surface  $S_k$ ,  $V(p, q) \cap \text{bd } C_l$  est vide.
  - Si  $p$  et  $q$  appartiennent au même morceau de surface  $S_k$ , mais pas à un même segment de courbe de  $L_j \subset \text{bd } S_k$ , alors l'intersection  $V(p, q) \cap \text{bd } C_l$  est restreinte à  $V(p, q) \cap S_k$ . C'est donc une boule topologique de dimension 1, dont les deux extrémités sont incluses dans  $\text{bd } V(p, q)$ .
  - Enfin, si  $p$  et  $q$  appartiennent tout deux à un segment de courbe  $L_j$  du bord de  $C_l$ . Soient  $S_i$  et  $S_k$  les deux surfaces du bord de  $C_l$ , qui sont incidentes à  $L_j$ . L'intersection  $V(p, q) \cap \text{bd } C_l$  est alors l'union  $(V(p, q) \cap S_i) \cup (V(p, q) \cap S_k)$  de deux boules topologiques de dimension 1, d'intérieurs disjoints, qui partagent l'extrémité  $L_j \cap V(p, q)$  et qui ont leur autre extrémité sur  $\text{bd } V(p, q)$ .  $V(p, q) \cap \text{bd } C_l$  est donc une boule topologique de dimension 1, avec ses deux extrémités sur  $\text{bd } V(p, q)$ .

L'intersection  $V(p, q) \cap C_l$  est soit vide, soit égale à  $V(p, q)$  toute entière, qui est une boule topologique de dimension 2, soit une portion de  $V(p, q)$ , telle que  $V(p, q) \cap \text{bd } C_l$  est une boule topologique de dimension 1, avec ses extrémités sur  $\text{bd } V(p, q)$ . Dans ce dernier cas,  $V(p, q) \cap \text{bd } C_l$  sépare  $V(p, q)$  en deux boules topologiques de dimension 2, et  $V(p, q) \cap C_l$  est l'une d'elle.

- Pour finir, considérons une cellule de Voronoï  $V(p)$ . Si  $p$  n'appartient pas à  $C_l$ , alors  $V(p) \cap C_l$  est vide. Si  $p$  est inclus dans l'intérieur de  $C_l$ , alors  $V(p) \cap C_l = V(p)$  est une boule topologique de dimension 3. Dans les autres cas,  $p$  appartient à  $\text{bd } C_l$  et est soit dans l'intérieur d'un segment de courbe de  $C_l$ , soit dans l'intérieur d'un morceau de surface de  $C_l$ , soit un sommet de  $C_l$ . Dans les trois cas,  $V(p) \cap \text{bd } C_l$  est l'union des  $V(p) \cap S_k$ , pour toutes les facettes  $S_k$  de  $\text{bd } C_l$  qui contiennent  $p$ . Chaque  $V(p) \cap S_k$  est une boule topologique de dimension 2, d'après le lemme 4.3.13, et leur union est une boule topologique de dimension 2, dont le bord est une sphère topologique de dimension 1 (voir la démonstration de la proposition 4.3.13 c). Ainsi,  $V(p) \cap \text{bd } C_l$  sépare  $V(p)$  en deux boules topologiques de dimension 3, et  $V(p) \cap C_l$  est l'une d'elles. □

Parmi tous les démonstrations des lemmes de cette section, seul la démonstration du lemme 4.3.14 utilise le fait que  $\mathcal{P}'$  n'a des sommets que sur la surface  $\bigcup \mathcal{S}$ . En fait, le lemme 4.3.12 est même vrai si on l'applique avec l'ensemble de sommets  $\mathcal{P}$  tout entier, car il n'utilise pas le lemme 4.3.14. La section suivante démontre que  $\mathcal{P}'$  vérifie les propriétés de la propriété de la boule topologique étendue pour  $\mathcal{F}$ , alors que ça n'est pas nécessairement vrai pour  $\mathcal{P}$ .

### Démonstration des propriétés d'homéomorphisme

D'après les lemmes 4.3.12, 4.3.13 et 4.3.15, pour tout sous-ensemble  $\mathcal{T} \subset \mathcal{P}'$  et pour toute face  $F \in \mathcal{F}$ , alors l'intersection  $V(\mathcal{T}) \cap F$  est soit vide soit une boule topologique de la bonne dimension. Dans ce qui suit, on supposera que  $\mathcal{P}'$  a la propriété de l'intersection générique étendue avec  $\mathcal{F}$ . Plus précisément on suppose que, toute face de Voronoï non vide  $V(\mathcal{T})$ , pour un sous-ensemble  $\mathcal{T} \in \mathcal{P}'$  de cardinal  $k$ , a la dimension  $k' = 4 - k$ , et que toutes ses intersections non vides  $V(\mathcal{T}) \cap F$  avec une face de  $\mathcal{F}$  de dimension  $j$  a la dimension  $k' + j - 3$ . Cette hypothèse de généralité peut être assurée par une petite perturbation.

Montrons maintenant le lemme suivant.

**Lemme 4.3.16.** *Si  $\mathcal{P}$  est l'ensemble de point final produit par l'algorithme de la section 4.2, on note  $\mathcal{P}' = \mathcal{P} \cap \bigcup \mathcal{S}$ . alors l'ensemble  $\mathcal{R} = \{V(\mathcal{T}) \cap F \mid \mathcal{T} \subset \mathcal{P}', F \in \mathcal{F}\}$  est un complexe CW et, sous l'hypothèse de généralité,  $\mathcal{P}'$  a la propriété de la boule topologique étendue, et la propriété de l'intersection générique étendue pour  $\bigcup \mathcal{F}$ .*

*Démonstration.* Les lemmes 4.3.12, 4.3.13 et 4.3.15 montrent que tous les éléments de  $\mathcal{R}$  sont des boules topologiques. Les faces de  $\mathcal{R}$  sont par définitions d'intérieurs disjoints, et le bord de chacune des faces de  $\mathcal{R}$  est une union de faces de  $\mathcal{R}$ .  $\mathcal{R}$  est ainsi un complexe CW. Pour tout sous-ensemble  $\mathcal{T} \subset \mathcal{P}'$ , on définit  $\mathcal{R}_{\mathcal{T}} = \{V(\mathcal{T}) \cap F \mid F \in \mathcal{F}\}$ .

L'hypothèse de généralité implique la propriété de l'intersection générique étendue par rapport à  $\mathcal{R}$ .

Dans [CDR07], Siu-Swing Cheng, Tamal Dey, et Edgar Ramos montrent que les conditions 1-4 de l'propriété de la boule topologique étendue sont satisfaites par  $\mathcal{P}'$  si les deux propriétés suivantes sont satisfaites.

**P1 :** Si  $V(\mathcal{T})$  est une face de dimension  $k$  de  $\mathcal{V}(\mathcal{P})$  et  $F$  est une face de dimension  $j$  de  $\mathcal{F}$ , alors l'intersection  $V(\mathcal{T}) \cap F$  est soit vide soit une boule topologique de dimension  $k + j - 3$ .

**P2 :** Pour tout sous-ensemble  $\mathcal{T} \subset \mathcal{P}'$ , il existe un unique élément de dimension minimal  $F_{\mathcal{T}} \in \mathcal{F}$ , tel que  $F_{\mathcal{T}}$  intersecte  $V(p)$  et  $F_{\mathcal{T}}$  est inclus dans toute face de  $\mathcal{F}$  qui intersecte  $V(\mathcal{T})$ .

La propriété P1 est une conséquence des lemmes 4.3.12, 4.3.13 et 4.3.15. Montrons que ces trois lemmes impliquent aussi la propriété P2.

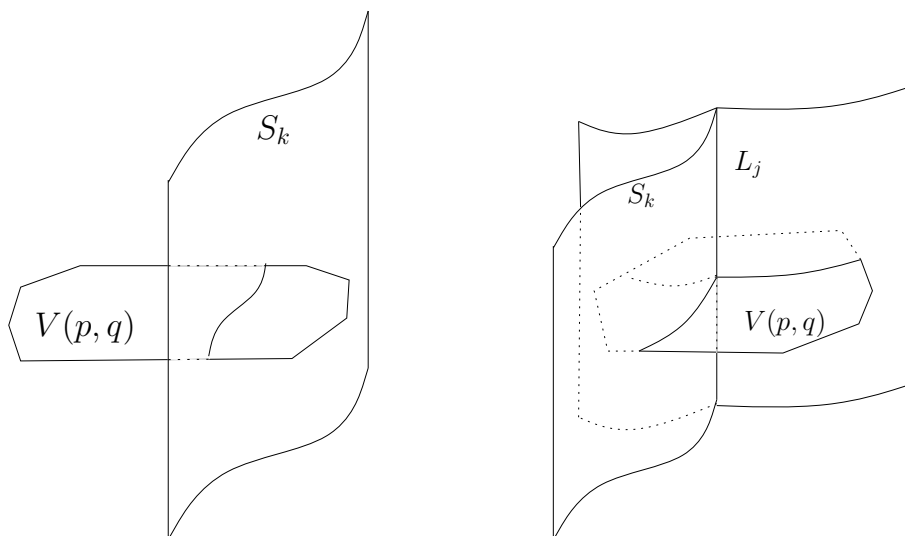


FIG. 4.6: Pour la démonstration du lemme. Intersection entre une facette  $V(p, q)$  et  $\mathcal{V}(\mathcal{P}')$ . À gauche,  $V(p, q)$  n'intersecte aucun segment de courbe. À droite,  $V(p, q)$  intersecte un unique segment de courbe  $L_j$ .

Soit  $V(p, q, r)$  une arête de  $\mathcal{V}(\mathcal{P}')$ .

- Soit  $V(p, q, r)$  est incluse dans une cellule  $C_i$  de  $\mathcal{C}$  (auquel cas  $F_{\mathcal{T}} = C_i$  vérifie trivialement la propriété P2),
- soit elle intersecte un unique morceau de surface  $S_k$  de  $\mathcal{S}$ , et les cellules de  $\mathcal{C}$  incidentes à  $S_k$  et alors  $F_{\mathcal{T}} = S_k$  vérifie P2.

Soit  $V(p, q)$  une facette de  $\mathcal{V}(\mathcal{P}')$ .

- Soit  $V(p, q)$  est incluse dans une cellule  $C_i$  de  $\mathcal{C}$  (auquel cas  $F_{\mathcal{T}} = C_i$  vérifie P2),
- soit elle intersecte un unique morceau de surface  $S_k$ , et les cellules incidentes à  $S_k$  (voir figure 4.6) alors  $F_{\mathcal{T}} = S_k$  vérifie P2,
- soit elle intersecte un unique segment de courbe  $L_j$ , et les cellules et morceaux de surface incidents à  $L_j$ , auquel cas  $F_{\mathcal{T}} = L_j$  vérifie P2.

Soit  $V(p)$  une cellule de  $\mathcal{V}(\mathcal{P}')$ .

- Soit  $V(p)$  est incluse dans une cellule  $C_i$  ( $F_{\mathcal{T}} = C_i$ ),
- soit  $V(p)$  intersecte un unique morceau de surface  $S_k$ , et les cellules incidentes à  $S_k$  ( $hubrt = S_k$ ),



- soit  $V(p)$  intersecte un unique segment de courbe  $L_j$ , et les cellules et morceaux de surface incident à  $L_j$  ( $F_{\mathcal{T}} = L_j$ ).

□

Le lemme précédent montre que  $\mathcal{P}'$  a la propriété de la boule topologique étendue par rapport à  $\mathcal{O}$ . On peut conclure alors, par le théorème 4.3.3 que  $\mathcal{D}_{|\mathcal{O}}(\mathcal{P}')$  et  $\mathcal{O}$  sont homéomorphes. De plus, comme la preuve du théorème 4.3.3 construit l'homéomorphisme étape par étape entre chaque face  $V(\mathcal{T}) \cap F \in \mathcal{R}$  et  $\bigcup \mathcal{D}_{|F}(\mathcal{P}')$ , par dimensions croissantes, l'homéomorphisme construit est tel que l'image de chaque face  $F \in \mathcal{F}$  est  $\bigcup \mathcal{D}_{|F}(\mathcal{P}')$ .

Pour conclure et montrer la propriété 4.3.1 b, il suffit maintenant de constater la propriété suivante.

**Lemme 4.3.17.** *Pour tout segment de courbe  $L_j \in \mathcal{L}$ , pour tout morceau de surface  $S_k \in \mathcal{F}$ , pour toute cellule  $C_l \in \mathcal{C}$ , on a  $\mathcal{D}_{|L_j}^1(\mathcal{P}) = \mathcal{D}_{|L_j}^1(\mathcal{P}')$ ,  $\mathcal{D}_{|S_k}^2(\mathcal{P}) = \mathcal{D}_{|S_k}^2(\mathcal{P}')$ , et  $\bigcup \mathcal{D}_{|C_l}(\mathcal{P}) = \bigcup \mathcal{D}_{|C_l}(\mathcal{P}')$ .*

*Démonstration.* Le lemme 4.3.12 est applicable à l'ensemble de points  $\mathcal{P}$  entier. Par conséquent, les arêtes de  $\mathcal{D}_{|L_j}(\mathcal{P})$  et  $\mathcal{D}_{|L_j}(\mathcal{P}')$  sont les arêtes  $pq$  telles que  $p$  et  $q$  soient consécutifs sur  $L_j$ .

Selon la règle R3, pour toute surface  $S_k$ , tout triangle  $pqr$  de  $\mathcal{D}_{|S_k}(\mathcal{P})$  est un triangle de  $\mathcal{D}_{|S_k}(\mathcal{P}')$ , car sa boule surfacique est toujours vide, avec un ensemble de points plus petit. Et réciproquement, comme la règle R5 assure qu'aucun point de  $\mathcal{P} \setminus \mathcal{P}'$  n'accroche de facette de  $\mathcal{D}_{|S_k}(\mathcal{P}')$ , un triangle  $pqr$  de  $\mathcal{D}_{|S_k}(\mathcal{P}')$  a sa boule surfacique d'intérieur vide de points de  $\mathcal{P}$ , donc  $pqr$  est un triangle de  $\mathcal{D}_{|S_k}(\mathcal{P})$ .

Enfin, pour toute cellule  $C_l \in \mathcal{C}$ ,  $\mathcal{D}_{|C_l}(\mathcal{P}')$  et  $\mathcal{D}_{|C_l}(\mathcal{P})$  ont le même bord. Donc les volumes bornés par ce bord sont égaux :  $\bigcup \mathcal{D}_{|C_l}(\mathcal{P}')$  =  $\bigcup \mathcal{D}_{|C_l}(\mathcal{P})$ . □

Les deux lemmes précédents achèvent ainsi la démonstration de la propriété 4.3.1 b.

## Distance d'Hausdorff

Montrons maintenant la proposition 4.3.1 c du théorème.

Commençons par une arête  $L_j \in \mathcal{L}$ . Pour toute arête  $e = pq$  in  $\mathcal{D}_{|L_j}(\mathcal{P})$ , à la fois  $e$  et le segment  $L_j(p, q)$  de la courbe  $L_j$ , joignant  $p$  à  $q$  sont contenus dans la boule linéique  $B(c_e, r_e)$  circonscrite à  $e$ . La distance d'Hausdorff entre  $e$  et  $L_j(p, q)$  est donc plus petite que  $2r_e$ . D'après la règle R2, cette quantité est plus petite que  $2\alpha_1\sigma(c_e)$ , et la distance d'Hausdorff entre  $L_j$  et  $\bigcup \mathcal{D}_{|L_j}(\mathcal{P})$  est plus petite que  $2\alpha_1 \max_{x \in L_j} \sigma(x)$ .

Considérons maintenant un morceau de surface  $S_k \in \mathcal{S}$ . Tout triangle  $pqr$  de  $\mathcal{D}_{|S_k}(\mathcal{P})$  est inclus dans sa boule de Delaunay surfacique  $B(c, r)$ , de rayon  $r \leq \alpha_2\sigma(c)$ , et ainsi tout point de  $pqr$  est à distance inférieure à  $\alpha_2\sigma(c)$  de  $S_k$ . D'après le lemme d'échantillonnage 4.3.9, on sait que tout point  $x$  de  $S_k$  est à distance  $O(\alpha_2)\sigma(p)$  du sommet  $p \in \mathcal{P}$  le plus proche. La distance d'Hausdorff entre  $S_k$  et  $\mathcal{D}_{|S_k}(\mathcal{P})$  est donc inférieure à  $O(\alpha_2) \max_{x \in S_k} \sigma(x)$ .

Enfin, si  $C_l$  est une cellule de  $\mathcal{C}$ . Tout tétraèdre  $pqrs$  de  $\mathcal{D}_{|C_l}(\mathcal{P})$  est inclus dans sa boule de Delaunay  $B(c, r)$ , de rayon inférieur à  $\sigma(c)$ . Et inversement, pour tout point  $x$  de  $C_l$ ,  $x$  est dans la cellule de Voronoï  $V(p)$  d'un

sommet de l'échantillon  $\mathcal{P}$ . Si  $x$  n'est pas dans  $\bigcup \mathcal{D}_{C_l}(\mathcal{P})$ , alors  $p$  est nécessairement un sommet de  $\text{bd } C_l$  d'après l'analyse de la section 4.3. Selon le lemme d'échantillonnage 4.3.9, la distance de  $x$  à  $\mathcal{P} \cap \text{bd } S_k$  est inférieure à  $O(\alpha_2) \max_{x \in S_k} \sigma(x)$ . La distance de Hausdorff entre  $C_l$  et  $\bigcup DCI(\mathcal{P})$  est inférieure à  $O(\alpha_2) \max_{x \in S_k} \sigma(x)$ .

#### 4.4 Terminaison

Dans cette section, nous montrerons que l'algorithme termine, si certaines hypothèses sur les paramètres  $\alpha_1$ ,  $\alpha_2$ ,  $\beta_2$ , et  $\beta_3$ , et sur la fonction de taille  $\sigma$  sont vérifiées. Cette preuve de terminaison procède de manière identique à celle du chapitre 3 (voir section 3.5 page 29), en bornant inférieurement le rayon d'insertion des sommets insérés dans le maillage par l'algorithme.

On note  $\sigma_0$  le minimum de  $\sigma$  sur  $\mathcal{O}$  et  $\mu_0$  le maximum de  $\sigma$  sur  $\bigcup \mathcal{S}$  :

$$\begin{aligned}\sigma_0 &= \min_{\mathcal{O}}(\sigma) \\ \mu_0 &= \max_{\bigcup \mathcal{S}}(\sigma).\end{aligned}$$

Naturellement, le rapport  $\frac{\mu_0}{\sigma_0}$  est supérieur à 1.

Pour chaque sommet  $p$  inséré par l'algorithme, on note  $r(p)$  le rayon d'insertion de  $p$  (voir la définition 3.5.1 page 29). Afin de montrer que le rayon d'insertion de tous les sommets insérés dans le maillage par l'algorithme est borné inférieurement, il est nécessaire de montrer que les sommets insérés sont suffisamment éloignés des faces de  $\mathcal{F}$  qui ne les contiennent pas. Pour tout sommet  $p$ , on appelle  $\mathcal{F}_2(p)$  l'ensemble des faces de  $\mathcal{F}_2$  qui contiennent  $p$ , et on note  $\delta(p)$  la plus petite distance entre  $p$  et les faces de  $\mathcal{F}_2$  qui ne le contiennent pas :

$$\delta(p) = \min_{F \in \mathcal{F}_2 \setminus \mathcal{F}_2(p)} d(x, F).$$

Afin de montrer que l'algorithme termine, nous allons prouver qu'il existe deux constantes  $\eta_2$  et  $\eta_3$ , comprises entre 0 et 1, qui vérifient  $\alpha_1 \leq \eta_3 \leq \eta_2 \leq 1$ , telles que les relations suivantes sont satisfaites tout du long de l'exécution de l'algorithme :

$$\forall p \in \mathcal{P}, \quad r(p) \geq \alpha_1 \sigma_0 \tag{4.1}$$

$$\delta(p) \geq \begin{cases} \alpha_1 \sigma_0 & \text{si } p \in \bigcup \mathcal{L} \\ \frac{\alpha_1}{\eta_2} \sigma_0 & \text{si } p \in \bigcup \mathcal{S} \setminus \bigcup \mathcal{L}, \\ \frac{\alpha_1}{\eta_3} \sigma_0 & \text{si } p \in \mathcal{O} \setminus \bigcup \mathcal{S} \end{cases} \tag{4.2}$$

Dans la suite de cette section 3.5, les relations (4.1) et (4.2) seront appelées *invariants*.

L'invariant (4.1) du rayon d'insertion assure à la fois la terminaison de l'algorithme, et une majoration du nombre de points insérés par l'algorithme.

Ces relations (4.1) et (4.2) sont vérifiées par l'ensemble des points de départ  $\mathcal{P}_0$ . Pour montrer que chaque insertion de point, on montre d'abord un lemme élémentaire, avant d'examiner la conservation des invariants à l'insertion d'un point, en fonction la règle de l'algorithme qui insère ce point.

**Lemme 4.4.1.** *Pour tout point  $x$  d'une face  $F \in \mathcal{F}_2$ , on a la minoration suivante :*

$$\delta(x) \geq \min(d(x, \text{bd } F), \text{lfs}(x)).$$

*Démonstration.* Soit  $G \in \mathcal{F}_2$  une face qui ne contient pas  $x$ , et  $y$  un point de  $G$ .

- Si  $x$  et  $y$  ne sont pas corrélés, alors  $d(x, y) \geq \text{lfs}(x)$ .
- Si  $x$  et  $y$  sont corrélés, alors  $F$  et  $G$  s'intersectent, et il existe  $w \in F \cap G$  tel que  $d(x, w) \leq \lambda_0$  et  $d(y, w) \leq \lambda_0$ . Dans ce cas, l'inégalité de l'hypothèse angulaire est vérifiée, et  $d(x, y) \geq d(x, F \cap G) \geq d(x, \text{bd } F)$ .

Dans les deux cas,  $d(x, y) \geq d(x, \text{bd } F)$  ou  $d(x, y) \geq \text{lfs}(x)$ , d'où le lemme.  $\square$

Examinons maintenant la conservation des invariants (4.1) et (4.2) lors de l'insertion d'un point par l'un des règles de l'algorithme.

### Règle R1

Quand la règle R1 est appliquée, le nouveau sommet est le centre  $c_e$  d'une boule de Delaunay restreinte à une arête  $L_j \in \mathcal{L}$ , circonscrite à une arête  $e = pq \in \mathcal{D}_{|L_j}$ , et dont l'un des sommets,  $q$  par exemple, n'appartient pas à  $L_j$ . Le rayon d'insertion de  $c_e$  est alors :

$$r(c_e) = \|c_e q\| \geq d(q, L_j) \geq \delta(q) \geq \alpha_1 \sigma_0$$

la dernière inégalité étant l'hypothèse d'induction. L'invariant (4.1) est donc satisfait après l'insertion de  $c_e$ .

Dans un second temps, remarquons que la boule de Delaunay  $B(c_e, r(c_e))$  est vide de sommets. Ainsi :

$$d(c_e, \mathcal{P}) \geq r(c_e) \geq \alpha_1 \sigma_0.$$

Cela signifie que  $c_e$  est au moins à distance  $\alpha_1 \sigma_0$  des extrémités de  $L_j$ . Le lemme 4.4.1 permet d'en conclure que l'invariant (4.2) est préservé par l'insertion de  $c_e$ .

### Règle R2

Quand la règle R2 est appliquée, le nouveau sommet  $c_e$  est le centre d'une boule de Delaunay  $B(c_e, r_e)$  circonscrite à une arête  $e$  de la triangulation de Delaunay restreinte  $\mathcal{D}_{|L_j}(\mathcal{P})$ , et telle que  $r_e \geq \alpha_1 \sigma(c_e)$ . Le rayon d'insertion  $r(c_e)$  de  $c_e$  est  $r_e$ , et satisfait :

$$r_e \geq \alpha_1 \sigma(c_e) \geq \alpha_1 \sigma_0.$$

Ainsi l'invariant (4.1) est conservé par l'insertion de  $c_e$ .

L'invariant (4.2) est prouvé de la même façon que pour la règle R1, par application du lemme 4.4.1.

### Règle R3

Quand la règle R3 est appliquée,  $c_f \in S_k$  est le centre d'une boule de Delaunay, circonscrite à une facette  $f \in \mathcal{D}_{|S_k}$ . Au moins l'un de ses sommets n'appartient pas à  $S_k$ . Appelons  $p$  un tel sommet de  $f$  n'appartenant pas à  $S_k$ . Alors

$$r(c_f) = \|c_f p\| \geq d(p, S_k) \geq \delta(p),$$

ce qui est supérieur à  $\alpha_1 \sigma_0$  par hypothèse d'induction. L'invariant (4.1) est donc préservé par l'insertion de  $c_f$ .

Pour montrer que l'invariant (4.2) est conservé, grâce au lemme 4.4.1, il suffit de montrer que la distance  $d(c_f, \text{bd } S_k)$  à tout point du bord de  $S_k$  est au moins  $\frac{\alpha_1 \sigma_0}{\eta_2}$ .

Soient  $y$  le point de  $\text{bd } S_k$  le plus proche de  $c_f$  et  $L_i$  une arête de  $\mathcal{L}$  qui contient  $y$ . Soit alors  $q$  le sommet de  $\mathcal{P} \cap L_i$  le plus proche de  $y$ . Alors

$$d(c_f, \text{bd } S_k) = d(c_f, y) \geq d(c_f, q) - d(y, q).$$

Comme la boule de Delaunay centrée en  $c_f$  et de rayon  $r(c_f)$  est vide de sommets,

$$d(c_f, q) \geq r(c_f),$$

et donc

$$d(c_f, \text{bd } S_k) \geq r(c_f) - d(y, q).$$

Comme la règle R2 est satisfaite quand la règle R3 est appliquée, on sait, d'après la preuve du lemme 4.3.12, que  $L_i$  est couverte par l'union des boules de Delaunay linéiques centrées sur  $L_i$ . Ainsi, il existe une boule de Delaunay linéique  $B(c_{e_1}, r_{e_1})$ , circonscrite à une arête  $e_1$  et contenant  $y$ . Soit  $p_1$  l'un des sommets de  $e_1$ . Alors, comme  $q$  est le point de  $\mathcal{P} \cap L_i$  le plus proche de  $y$ ,

$$d(y, q) \leq d(y, p_1) \leq 2r_{e_1}.$$

Donc,

$$d(y, q) \leq 2\alpha_1 \mu_0,$$

et ainsi

$$d(c_f, \text{bd } S_k) \geq r(c_f) - 2\alpha_1 \mu_0. \quad (4.3)$$

– Si  $p \in \mathcal{P} \setminus \bigcup \mathcal{S}$ , par hypothèse d'induction,

$$r(c_f) = d(c_f, p) \geq \frac{\alpha_1 \sigma_0}{\eta_3},$$

et ainsi

$$d(c_f, \text{bd } S_k) \geq \frac{\alpha_1 \sigma_0}{\eta_3} - 2\alpha_1 \mu_0.$$

La relation  $d(c_f, \text{bd } S_k) \geq \frac{\alpha_1 \sigma_0}{\eta_2}$ , et donc l'invariant (4.2), est satisfaite si

$$\frac{\alpha_1}{\eta_3} \sigma_0 - 2\alpha_1 \mu_0 \geq \frac{\alpha_1 \sigma_0}{\eta_2}$$

c'est-à-dire

$$\frac{1}{\eta_3} - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0}. \quad (4.4)$$

– Si  $p \in \bigcup \mathcal{S}$ , et  $p$  et  $c_f$  ne sont pas corrélés, alors

$$r(c_f) = d(c_f, p) \geq \text{lfs}(c_f),$$

donc, comme  $\text{lfs}(c) \geq \sigma_0$ , l'équation (4.3) donne :

$$d(c_f, \text{bd } S_k) \geq \sigma_0 - 2\alpha_1\mu_0.$$

La relation  $d(c_f, \text{bd } S_k) \geq \frac{\alpha_1\sigma_0}{\eta_2}$ , et donc l'invariant (4.2) est satisfaite si

$$\sigma_0 - 2\alpha_1\mu_0 \geq \frac{\alpha_1\sigma_0}{\eta_2}$$

c'est-à-dire

$$\frac{1}{\alpha_1} - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0}. \quad (4.5)$$

– Si  $p \in \bigcup \mathcal{S}$ , et est corrélés avec  $c_f$ , soit  $F_i$  la face de  $\mathcal{S} \cup \mathcal{L}$  qui contient  $p$  dans son intérieur. Alors  $F_i$  et  $S_k$  sont non disjointes, et il existe  $w \in F_i \cap S_k$ , tel que

$$\begin{aligned} d(p, w) &\leq \lambda_0 \\ \text{et } d(c_f, w) &\leq \lambda_0. \end{aligned}$$

Alors, par hypothèse angulaire,

$$r(c_f)^2 = d(c_f, p)^2 \geq d(c_f, F_i \cap S_k)^2 + d(p, F_i \cap S_k)^2. \quad (4.6)$$

Cette équation implique  $d(c_f, F_i \cap S_k) \leq r(c_f)$ , ce qui veut dire que  $F_i \cap S_k$  intersecte la boule de Delaunay surfacique centrée en  $c_f$ . Ceci implique que  $F_i$  ne peut être une arête de  $\mathcal{L}$ , car alors l'intersection  $F_i \cap S_k$  serait restreinte à un sommet de  $\mathcal{Q}$ , ce qui conduit à une contradiction, car  $F_i \cap S_k$  serait à la fois un sommet de  $\mathcal{Q}$  et dans une boule de Delaunay. Ainsi  $F_i \in \mathcal{S}$ . Alors, par hypothèse d'induction,

$$\delta(p) \geq \frac{\alpha_1\sigma_0}{\eta_2},$$

et donc

$$d(p, w) \geq d(p, S_k \cap F_i) \geq \frac{\alpha_1\sigma_0}{\eta_2}.$$

L'équation (4.6) implique maintenant

$$r(c_f)^2 \geq d(c_f, \text{bd } S_k)^2 + \left( \frac{\alpha_1\sigma_0}{\eta_2} \right)^2. \quad (4.7)$$

Rappelons l'équation suivante :

$$d(c_f, \text{bd } S_k) \geq r(c_f) - 2\alpha_1\mu_0. \quad (4.3)$$

Si  $d(c_f, \text{bd } S_k) \geq \frac{\alpha_1\sigma_0}{\eta_2}$ , alors l'invariant (4.2) est préservé. On va maintenant supposer  $d(c_f, \text{bd } S_k) < \frac{\alpha_1\sigma_0}{\eta_2}$ , et chercher une condition sur  $\eta_2$  et  $\alpha_1$ , par rapport au rapport  $\frac{\mu_0}{\sigma_0}$ , pour que les équations (4.3) et (4.7)

conduisent à une contradiction. Cela montrera que, si la condition est satisfaite, l'invariant (4.2) est nécessairement préservé.

La combinaison des deux équations (4.3) et (4.7) donne

$$d(c_f, \text{bd } S_k)^2 + \left(\frac{\alpha_1 \sigma_0}{\eta_2}\right)^2 \leq r(c_f)^2 \leq (d(c_f, \text{bd } S_k) + 2\alpha_1 \mu_0)^2.$$

Cet relation s'écrit autrement :

$$\left(\frac{\alpha_1 \sigma_0}{\eta_2}\right)^2 \leq 2\alpha_1 \mu_0 (2\alpha_1 \mu_0 + 2d(c_f, \text{bd } S_k)).$$

Si on suppose  $d(c_f, \text{bd } S_k) < \frac{\alpha_1 \sigma_0}{\eta_2}$ , alors

$$\left(\frac{\alpha_1 \sigma_0}{\eta_2}\right)^2 < 2\alpha_1 \mu_0 \left(2\alpha_1 \mu_0 + 2\frac{\alpha_1 \sigma_0}{\eta_2}\right),$$

ce qui peut s'écrire

$$\left(\frac{1}{\eta_2}\right)^2 - 4\frac{\mu_0}{\sigma_0} \left(\frac{\mu_0}{\sigma_0} + \frac{1}{\eta_2}\right) < 0.$$

Ce qui conduit à une contradiction si

$$\frac{1}{\eta_2} \geq 2(\sqrt{2} + 1)\frac{\mu_0}{\sigma_0}. \quad (4.8)$$

#### Règle R4

Quand la règle R4 est appliquée, on considère une boule de Delaunay surfacique  $B(c_f, r_f)$ , circonscrite à une facette  $f \in \mathcal{D}_{|S_k} \mathcal{P}$ , telle que

$$r_f \geq \alpha_2 \sigma(c_f)$$

ou

$$r_f \geq \beta_2 l_{\min}(f),$$

où  $l_{\min}(f)$  est la longueur de la plus petite arête de  $f$ .

- Dans le premier cas,  $r_f \geq \alpha_2 \sigma(c) \geq \alpha_2 \sigma_0$ .
- Dans le second cas, par hypothèse d'induction,  $l_{\min}(f) \geq \alpha_1 \sigma_0$ , et ainsi  $r_f \geq \beta_2 \alpha_1 \sigma_0$ .

La procédure **refine-facet-or-edge** est appelée sur  $f$ . Le centre  $c_f$  est alors inséré, ou rejeté.

**Premier cas :  $c_f$  est inséré.** Supposons que **refine-facet-or-edge** insère  $c_f$ . Alors, son rayon d'insertion est  $r(c_f) = r_f$ .

L'invariant (4.1) est conservé si  $r_f \geq \alpha_1 \sigma_0$ , c'est-à-dire si on a à la fois

$$\alpha_2 \geq \alpha_1 \quad (4.9)$$

et

$$\beta_2 \geq 1. \quad (4.10)$$

Il reste à montrer la conservation de l'invariant (4.2). Comme dans le cas de la règle R3, il est suffisant de montrer  $d(c_f, L_i) \geq \frac{\alpha_1 \sigma_0}{\eta_2}$  pour tout  $L_i \in \mathcal{L}$  du bord de  $S_k$ . De plus, on a aussi

$$d(c_f, L_i) \geq r(c_f) - 2\alpha_1 \mu_0$$

comme dans le cas de la règle R3

L'invariant (4.2) est donc satisfait si

$$\alpha_2 \sigma_0 - 2\alpha_1 \mu_0 \geq \frac{\alpha_1 \sigma_0}{\eta_2}$$

et

$$\beta_2 \alpha_1 \sigma_0 - 2\alpha_1 \mu_0 \geq \frac{\alpha_1 \sigma_0}{\eta_2}$$

c'est-à-dire

$$\frac{\alpha_2}{\alpha_1} - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0} \quad (4.11)$$

$$\beta_2 - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0}. \quad (4.12)$$

**Second case :  $c_f$  est rejeté.** Supposons maintenant que *refine-facet-or-edge* insère le centre  $c_e \in L_i$  d'une boule de Delaunay linéique  $B(c_e, r_e)$  accrochée par  $c_f$ . Le rayon d'insertion de  $c_e$  est  $r_e$ . Soit  $p$  un sommet de  $e$ . Comme  $p$  n'est pas dans  $B(c_f, r_f)$  est que  $c_f$  et  $p$  appartiennent à  $B(c_e, r_e)$ ,

$$r_f \leq d(c_f, p) \leq 2r_e.$$

Alors,

$$r_e \geq \frac{r_f}{2} \geq \begin{cases} \frac{\alpha_2 \sigma_0}{2} \\ \text{ou} \\ \frac{\beta_2 \alpha_1 \sigma_0}{2}. \end{cases}$$

L'invariant (4.1) est conservé par l'insertion de  $c_e$  si :

$$\alpha_2 \geq 2\alpha_1 \quad (4.13)$$

et

$$\beta_2 \geq 2. \quad (4.14)$$

Comme  $B(c_e, r_e)$  est une boule vide, si les conditions (4.13) et (4.14) sont vérifiées,  $c_e$  est au moins à distance  $r_e \geq \alpha_1 \sigma_0$  des deux extrémités de  $L_i$ . Selon le lemme 4.4.1, l'invariant (4.2) est alors conservé.

## Règle R5

Quand la règle R5 est appliquée, on considère la boule de Delaunay  $B(c_t, r_t)$  d'un tétraèdre  $t$ , qui viole soit le critère de taille (règle R5.1) soit le critère de forme (règle R5.2).

Le rayon d'insertion  $r(c_t)$  du centre  $c_t$  est le rayon de la boule,  $r_t$ .

- Si la règle R5.1 est appliquée,  $r_t \geq \sigma(c_t) \geq \sigma_0$ .
- Si la règle R5.2 est appliquée,  $r_t \geq \beta_3 l_{min}(t)$  où  $l_{min}(t)$  est la longueur de la plus petite arête de  $t$ . Par hypothèse d'induction,  $l_{min}(t) \geq \alpha_1 \sigma_0$ . Ainsi  $r_t \geq \beta_3 \alpha_1 \sigma_0$ .

**Règle R5. Premier sous-cas 5.1** Supposons que la procédure **refine-tet-facet-or-edge** insère  $c_t$  comme nouveau sommet. Le rayon d'insertion de  $c_t$  est  $r(c_t) = r_t$ . L'invariant (4.1) est préservé si  $\sigma_0 \geq \alpha_1 \sigma_0$  et  $\beta_3 \alpha_1 \sigma_0 \geq \alpha_1 \sigma_0$ , c'est-à-dire si

$$1 \geq \alpha_1 \quad (4.15)$$

et

$$\beta_3 \geq 1. \quad (4.16)$$

Pour montrer la conservation de l'invariant (4.2), prenons  $y$  le point de  $\bigcup \mathcal{S}$  le plus proche de  $c_t$ . On a :

$$d(c_t, \bigcup \mathcal{S}) = d(c_t, y) \geq d(c_t, \mathcal{P}') - d(y, \mathcal{P}')$$

où  $\mathcal{P}' = \mathcal{P} \cap \bigcup \mathcal{S}$  est l'ensemble courant des sommets sur  $\bigcup \mathcal{S}$ . Soit  $S_k$  le morceau de surface qui contient  $y$ , et  $q$  le sommet de  $\mathcal{P} \cap S_k$  le plus proche de  $y$ . Quand la règle R5 est appliquée, la règle R4 est satisfaite et ainsi toute boule de Delaunay surfacique  $B(c, r)$  centrée sur  $S_k$  a un rayon  $r$  inférieur à  $\alpha_2 \sigma(c)$ . D'après le lemme 4.3.11, on sait que  $d(y, q)$  est au plus  $\alpha_2 (1 + O(\alpha_2))$ . Pour  $\alpha_2$  suffisamment petit, la constante  $\alpha_2 (1 + O(\alpha_2))$  est inférieure à  $2\alpha_2$ , et  $d(y, q) \leq 2\alpha_2$ . Ainsi

$$d(y, \mathcal{P}') \leq 2\alpha_2 \sigma(y) \leq 2\alpha_2 \mu_0$$

et

$$\begin{aligned} d(c_t, \mathcal{S}) = d(c_t, y) &\geq r(c_t) - d(y, p) \\ &\geq r(c_t) - 2\alpha_2 \mu_0 \\ &\geq \begin{cases} \sigma_0 - 2\alpha_2 \mu_0 \\ \text{ou} \\ \beta_3 \alpha_1 \sigma_0 - 2\alpha_2 \mu_0 \end{cases} \end{aligned}$$

Donc l'invariant (4.2) est vérifié si

$$\begin{aligned} \sigma_0 - 2\alpha_2 \mu_0 &\geq \frac{\alpha_1}{\eta_2} \sigma_0 \\ \beta_3 \alpha_1 \sigma_0 - 2\alpha_2 \mu_0 &\geq \frac{\alpha_1}{\eta_2} \sigma_0, \end{aligned}$$

c'est-à-dire

$$1 - \frac{\alpha_1}{\eta_2} \geq \frac{2\alpha_2 \mu_0}{\sigma_0} \quad (4.17)$$

$$\beta_3 \alpha_1 - \frac{\alpha_1}{\eta_2} \geq \frac{2\alpha_2 \mu_0}{\sigma_0}. \quad (4.18)$$



**Règle R5. Deuxième sous-cas 5.2** Supposons maintenant que la procédure `refine-tet-facet-or-edge` insère le centre  $c_f$  d'une boule de Delaunay surfacique  $B(c_f, r_f)$  de  $\mathcal{D}_{|S_k}$ , accrochée par  $c_t$ . Le rayon d'insertion  $r(c_f)$  de  $c_f$  est  $r_f$ . Soit  $p$  un sommet de la facette  $f$  circonscrite par  $B(c_f, r_f)$ . Alors, comme la boule de Delaunay  $B(c_t, r_t)$  circonscrite au tétraèdre  $t$  ne contient pas le sommet  $p$ , et  $c_t$  et  $p$  appartiennent tous les deux à la boule  $B(c_f, r_f)$ , on a :

$$r_t \leq d(c_t, p) \leq 2r_f.$$

Alors, selon le critère qui a déclenché la règle R5, on a :

$$r_f \geq \frac{\sigma_0}{2}$$

ou

$$r_f \geq \frac{\beta_3 \alpha_1 \sigma_0}{2}.$$

Ainsi, l'invariant (4.1) est conservé si les deux inégalités sont vérifiées :

$$\beta_3 \geq 2 \tag{4.19}$$

$$\frac{1}{2} \geq \alpha_1. \tag{4.20}$$

Pour montrer la conservation de l'invariant (4.2), il suffit de montrer que  $c_f$  est au moins à distance  $\frac{\alpha_1 \sigma_0}{\eta_2}$  de toute arête de  $\text{bd } S_k$  (lemme 4.4.1).

Soient  $L_i \in \mathcal{L}$  une arête du bord de  $S_k$  et  $y$  le point de  $L_i$  le plus proche de  $c_f$  sur  $L_i$ . On  $L_i$ . De la même façon que dans le cas de la règle R4, on a :

$$d(c_f, L_i) \geq r(c_f) - 2\alpha_1 \mu_0.$$

Alors l'invariant (4.2) est satisfait si les deux inégalités suivantes sont vérifiées :

$$\begin{aligned} \frac{\sigma_0}{2} - 2\alpha_1 \mu_0 &\geq \frac{\alpha_1 \sigma_0}{\eta_3} \\ \frac{\beta_3 \alpha_1 \sigma_0}{2} - 2\alpha_1 \mu_0 &\geq \frac{\alpha_1 \sigma_0}{\eta_3} \end{aligned}$$

c'est-à-dire, si

$$\frac{1}{2} - \frac{\alpha_1}{\eta_3} \geq \frac{2\alpha_1 \mu_0}{\sigma_0} \tag{4.21}$$

$$\frac{\beta_3}{2} - \frac{1}{\eta_3} \geq \frac{2\mu_0}{\sigma_0}. \tag{4.22}$$

**Règle R5. Troisième sous-cas 5.3** Pour finir, supposons que `refine-tet-facet-or-edge` insère le centre  $c_e$  d'une boule de Delaunay linéique  $B(c_e, r_e)$ , soit parce que  $B(c_e, r_e)$  est accrochée par  $c_t$  (sous-cas 5.3.1) soit parce que  $B(c_e, r_e)$  est accrochée par le centre  $c_f$  d'une boule de Delaunay surfacique qui est elle-même accrochée par  $c_t$  (sous-cas 5.3.2). Le rayon d'insertion de  $c_e$  est  $r_e$ .

Dans le sous-cas 5.3.1, on a

$$r_e \geq \frac{\sigma_0}{2}$$

ou

$$r_e \geq \frac{\beta_3 \alpha_1 \sigma_0}{2}.$$

La preuve de l'invariant (4.1) est la même que dans le sous-cas 5.2, en remplaçant  $B(c_f, r_f)$  par  $B(c_e, r_e)$  et en considérant un sommet  $p$  de l'arête  $e$  circonscrite par  $B(c_e, r_e)$ . Ainsi, l'invariant (4.1) est conservé si les inégalités (4.19) et (4.20) sont vérifiées.

Dans le sous-cas 5.3.2, on a, comme dans le sous-cas 5.2,

$$r_f \geq \frac{\sigma_0}{2}$$

ou

$$r_f \geq \frac{\beta_3 \alpha_1 \sigma_0}{2}.$$

Pour tout sommet  $p$  de  $e$ ,  $2r_e \geq d(p, c_f) \geq r_f$ , ainsi :

$$r_f \geq \frac{\sigma_0}{4}$$

ou

$$r_f \geq \frac{\beta_3 \alpha_1 \sigma_0}{4}.$$

L'invariant (4.1) est conservé si les deux inégalités suivantes sont vérifiées :

$$\beta_3 \geq 4 \tag{4.23}$$

$$\frac{1}{4} \geq \alpha_1. \tag{4.24}$$

Pour montrer la conservation de l'invariant (4.2), il suffit de remarquer que  $B(c_e, r_e)$  est vide de sommets, et donc que  $d(c_e, \mathcal{Q}) \geq r_e$ . D'après le lemme 4.4.1, l'invariant (4.2) est donc conservé si les conditions énoncées ci-dessus sont vérifiées.

### Résumé des conditions, et conclusion de la preuve

Pour résumer, les invariants (4.1) et (4.2) sont maintenus durant l'exécution de l'algorithme, si l'ensemble suivant d'inégalités est vérifié :

$$0 < \alpha_1 \leq \alpha_2 \leq 1 \tag{4.25}$$

$$\alpha_1 \leq \eta_3 \leq \eta_2 \leq 1 \tag{4.26}$$

#### (règle R3)

(cas  $p \in \mathcal{P} \setminus \bigcup \mathcal{S}$ )

$$\frac{1}{\eta_3} - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0} \tag{4.4}$$

(cas  $p \in \bigcup \mathcal{S}$ , pas corrélé avec  $c_f$ )

$$\frac{1}{\alpha_1} - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0} \tag{4.5}$$

(cas  $p \in \bigcup \mathcal{S}$ , corrélé avec  $c_f$ )

$$\frac{1}{\eta_2} \geq 2(\sqrt{2} + 1) \frac{\mu_0}{\sigma_0} \tag{4.8}$$

#### (règle R4)

(cas  $c_f$  inséré)

$$\alpha_2 \geq \alpha_1 \quad (4.9)$$

$$\beta_2 \geq 1 \quad (4.10)$$

$$\frac{\alpha_2}{\alpha_1} - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0} \quad (4.11)$$

$$\beta_2 - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0} \quad (4.12)$$

(cas  $c_f$  rejeté, accroche  $c_e$ )

$$\alpha_2 \geq 2\alpha_1 \quad (4.13)$$

$$\beta_2 \geq 2 \quad (4.14)$$

**(règle R5)**

(premier sous-cas,  $c_t$  inséré)

$$1 \geq \alpha_1 \quad (4.15)$$

$$\beta_3 \geq 1 \quad (4.16)$$

$$1 - \frac{\alpha_1}{\eta_3} \geq \frac{2\alpha_2\mu_0}{\sigma_0} \quad (4.17)$$

$$\beta_3\alpha_1 - \frac{\alpha_1}{\eta_3} \geq \frac{2\alpha_2\mu_0}{\sigma_0} \quad (4.18)$$

(deuxième sous-cas,  $c_t$  accroche  $f$ ,  $c_f$  est inséré)

$$\beta_3 \geq 2 \quad (4.19)$$

$$\frac{1}{2} \geq \alpha_1 \quad (4.20)$$

$$\frac{1}{2} - \frac{\alpha_1}{\eta_2} \geq \frac{2\alpha_1\mu_0}{\sigma_0} \quad (4.21)$$

$$\frac{\beta_3}{2} - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0} \quad (4.22)$$

(troisième sous-cas,  $c_e$  qui est inséré)

$$\beta_3 \geq 4 \quad (4.23)$$

$$\frac{1}{4} \geq \alpha_1. \quad (4.24)$$

Les relations (4.25) et (4.26) viennent des définitions des paramètres  $\alpha_1$ ,  $\alpha_2$ ,  $\eta_2$  et  $\eta_3$ .

Cet ensemble de conditions peut être réduite. En effet, les relations (4.25) et (4.26) sont impliquées par les conditions (4.4), (4.9) et (4.17), et les conditions (4.9), (4.10), (4.15), (4.16), (4.19) et (4.20) sont clairement plus faibles que l'ensemble des conditions (4.14), (4.13), (4.23) et (4.24). De plus, comme  $\alpha_2 \leq 1$  et  $\mu_0 \geq \sigma_0$ , les conditions (4.5) et (4.13) sont plus faibles que la condition (4.11), la condition (4.14) est plus faible que la condition (4.12), la condition (4.23) est plus faible que la condition (4.22), et la condition (4.24) est plus faible que la condition (4.21). Ainsi, l'ensemble de conditions peut se

réduire à :

$$\frac{1}{\eta_3} - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0} \quad (4.4)$$

$$\frac{1}{\eta_2} \geq 2(\sqrt{2} + 1) \frac{\mu_0}{\sigma_0} \quad (4.8)$$

$$\frac{\alpha_2}{\alpha_1} - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0} \quad (4.11)$$

$$\beta_2 - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0} \quad (4.12)$$

$$1 - \frac{\alpha_1}{\eta_3} \geq \frac{2\alpha_2\mu_0}{\sigma_0} \quad (4.17)$$

$$\beta_3\alpha_1 - \frac{\alpha_1}{\eta_3} \geq \frac{2\alpha_2\mu_0}{\sigma_0} \quad (4.18)$$

$$\frac{1}{2} - \frac{\alpha_1}{\eta_2} \geq \frac{2\alpha_1\mu_0}{\sigma_0} \quad (4.21)$$

$$\frac{\beta_3}{2} - \frac{1}{\eta_2} \geq \frac{2\mu_0}{\sigma_0}. \quad (4.22)$$

On peut choisir  $\eta_2$  et  $\eta_3$  tels que les inégalités (4.4) et (4.8) soient des égalités :

$$\begin{aligned} \frac{1}{\eta_2} &= 2(\sqrt{2} + 1) \frac{\mu_0}{\sigma_0} \\ \frac{1}{\eta_3} &= 2(\sqrt{2} + 2) \frac{\mu_0}{\sigma_0}. \end{aligned}$$

La seule condition sur  $\beta_2$  est l'inégalité (4.12). D'après le choix précédent pour  $\eta_2$  et  $\eta_3$ , cette condition s'exprime maintenant ainsi :

$$\beta_2 \geq 2(\sqrt{2} + 2) \frac{\mu_0}{\sigma_0}. \quad (4.27)$$

L'ensemble des conditions restantes est équivalent à :

$$\frac{\alpha_2}{\alpha_1} \geq \frac{1}{\eta_3} \quad (4.11 \text{ bis})$$

$$1 \geq \alpha_1 \left( \frac{1}{\eta_3} + \frac{\alpha_2}{\alpha_1} \times \frac{2\mu_0}{\sigma_0} \right) \quad (4.17 \text{ bis})$$

$$\frac{1}{2} \geq \frac{\alpha_1}{\eta_3} \quad (4.21 \text{ bis})$$

$$\beta_3 \geq \frac{1}{\eta_3} + \frac{\alpha_2}{\alpha_1} \times \frac{2\mu_0}{\sigma_0} \quad (4.18 \text{ bis})$$

$$\beta_3 \geq \frac{2}{\eta_3}. \quad (4.22 \text{ bis})$$

D'après la relation (4.11 bis), les conditions (4.21 bis) et (4.22 bis) sont plus faibles que les conditions (4.17 bis) et (4.18 bis), respectivement. L'ensemble

des conditions se réduit donc à :

$$\frac{\alpha_2}{\alpha_1} \geq \frac{1}{\eta_3} \quad (4.11 \text{ bis})$$

$$1 \geq \alpha_1 \left( \frac{1}{\eta_3} + \frac{\alpha_2}{\alpha_1} \times \frac{2\mu_0}{\sigma_0} \right) \quad (4.17 \text{ bis})$$

$$\beta_3 \geq \frac{1}{\eta_3} + \frac{\alpha_2}{\alpha_1} \times \frac{2\mu_0}{\sigma_0}. \quad (4.18 \text{ bis})$$

La première équation fixe le ratio  $\frac{\alpha_2}{\alpha_1}$ , et la deuxième est satisfaite si  $\alpha_2$  et  $\alpha_1$  sont suffisamment petits. La troisième équation impose un minimum, sur  $\beta_3$ , en fonction du ratio  $\frac{\alpha_2}{\alpha_1}$ . On choisit de minimiser le ratio  $\frac{\alpha_2}{\alpha_1}$ . Les conditions sur  $\alpha_1$ ,  $\alpha_2$ , et  $\beta_3$  sont alors :

$$\alpha_1 \leq \frac{1}{\frac{1}{\eta_3} \left( 2\frac{\mu_0}{\sigma_0} + 1 \right)} = \frac{1}{2(\sqrt{2} + 2)\frac{\mu_0}{\sigma_0} \left( 2\frac{\mu_0}{\sigma_0} + 1 \right)}, \quad (4.28)$$

$$\alpha_2 = \frac{\alpha_1}{2\frac{\mu_0}{\sigma_0} + 1} \leq \frac{1}{2(\sqrt{2} + 2)\frac{\mu_0}{\sigma_0} \left( 2\frac{\mu_0}{\sigma_0} + 1 \right)^2}, \quad (4.29)$$

et

$$\beta_3 \geq 2(\sqrt{2} + 2)\frac{\mu_0}{\sigma_0} \left( 2\frac{\mu_0}{\sigma_0} + 1 \right). \quad (4.30)$$

Si les paramètres  $\alpha_1$ ,  $\alpha_2$ ,  $\beta_2$ , et  $\beta_3$  de l'algorithme satisfont les équations (4.27), (4.29), (4.28) et (4.30), alors l'invariant (4.1) est conservé tout du long du déroulement de l'algorithme. Par un argument standard volumique standard (qui a été développé dans la section 3.5 page 29), cet invariant montre que l'algorithme se termine.

Les conditions (4.29) et (4.28) sur  $\alpha_1$  et  $\alpha_2$  permettent de choisir ces deux paramètres aussi petits que l'on veut, notamment pour satisfaire les conditions d'homéomorphie de la section 4.3. Si  $\alpha_1$  et  $\alpha_2$  sont trop différents, cela dégrade la borne inférieure sur  $\beta_3$ , selon l'équation (4.18 bis).

Le rapport  $\frac{\mu_0}{\sigma_0}$  qui apparaît dans les équations est dû au caractère non-local de cette preuve de terminaison. Dans le chapitre 3, la preuve de terminaison utilisait un invariant qui dépendait de  $\sigma(x)$ , en chaque point du volume. Dans ce chapitre, les règles supplémentaires de raffinement ne permettent pas d'établir des invariants qui dépendent du point, pour la preuve de terminaison. Les invariants (4.1) et (4.2) dépendent du *minimum* de  $\sigma$  sur  $\bigcup \mathcal{S}$ , ce qui ne permet pas d'utiliser le caractère Lipschitz de  $\sigma$ . Le rapport  $\frac{\mu_0}{\sigma_0}$  qui intervient dans les équations est une mesure de la variation de  $\sigma$ .

La condition (4.30) sur  $\beta_3$  est relativement mauvaise. En effet, même si le rapport  $\frac{\mu_0}{\sigma_0}$  est proche de 1,  $\beta_3$  vaut au minimum 20,4. Cependant, en pratique l'algorithme termine avec des valeurs de  $\beta_3$  de l'ordre de 4, 3, et même 2.

## 4.5 Résultats

L'algorithme de ce chapitre est en cours d'implantation dans la bibliothèque CGAL. Une phase d'élimination des slivers est appliquée, à la fin de l'algorithme (voir section 3.6).

L'algorithme interagit avec l'objet  $\mathcal{O}$  en entrée par l'intermédiaire d'un oracle, qui détecte les intersections entre les segments et les morceaux de surfaces et entre les triangles et les segments de courbes. Pour l'instant, une seule implantation de cet oracle a été faite, qui ne gère que les segments de courbes et les morceaux de surfaces qui sont décrits respectivement par des polygones et des maillages triangulaires. Avec cet oracle, notre algorithme agit, au niveau de la surface, comme un algorithme de remaillage, qui préserve les arêtes vives de l'objet.

Les figures 4.7 et 4.8 présentent deux résultats préliminaires de l'algorithme. Le modèle « ITER » est un morceau du futur réacteur thermonucléaire expérimental ITER. Le modèle « CAD » est un pièce mécanique d'avion.

Pour le modèle « ITER », les paramètres choisis étaient  $\alpha_1 = \alpha_2 = \frac{1}{2}$ , et  $\beta_2 = 1$  et  $\beta_3 = 5$ . Le plus petit angle dièdre du maillage est  $4,5^\circ$ . Pour le modèle « CAD », les paramètres étaient  $\alpha_1 = \frac{1}{3}$ ,  $\alpha_2 = \frac{2}{3}$ , et  $\beta_2 = 1$  et  $\beta_3 = 3$ , et le plus petit angle dièdre du maillage est  $4,4^\circ$ . On voit sur ces deux exemples que le choix des paramètres de l'algorithme peut se faire sans respecter les limitations théoriques imposées par la preuve de terminaison section 4.4. En pratique, l'algorithme termine quelques que soient les paramètres, tant que  $\beta_2$  est supérieur à 1, et  $\beta_3$  est supérieur à 2.

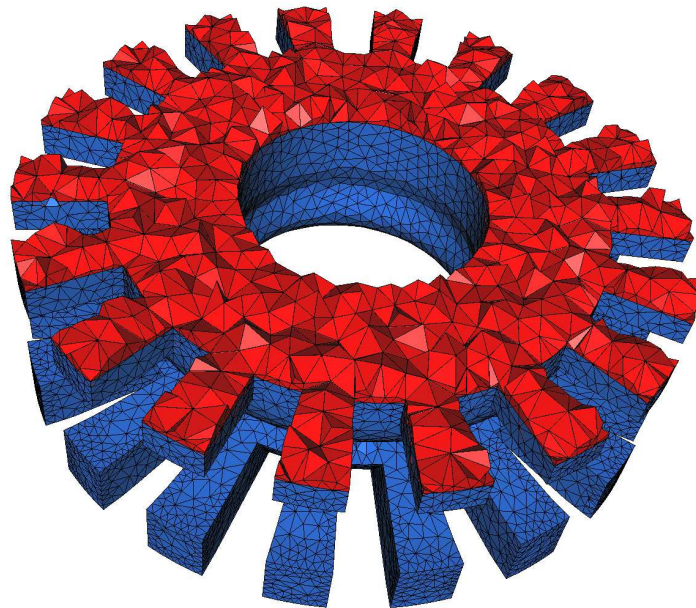
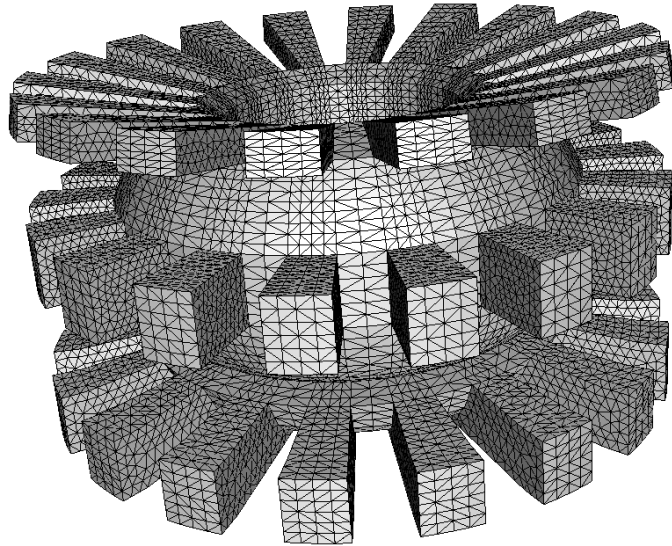


FIG. 4.7: Modèle **ITER**. En haut, le maillage de surface d'origine. En bas, le maillage volumique produit par l'algorithme de ce chapitre. 21 779 sommets et 69 756 tétraèdres.

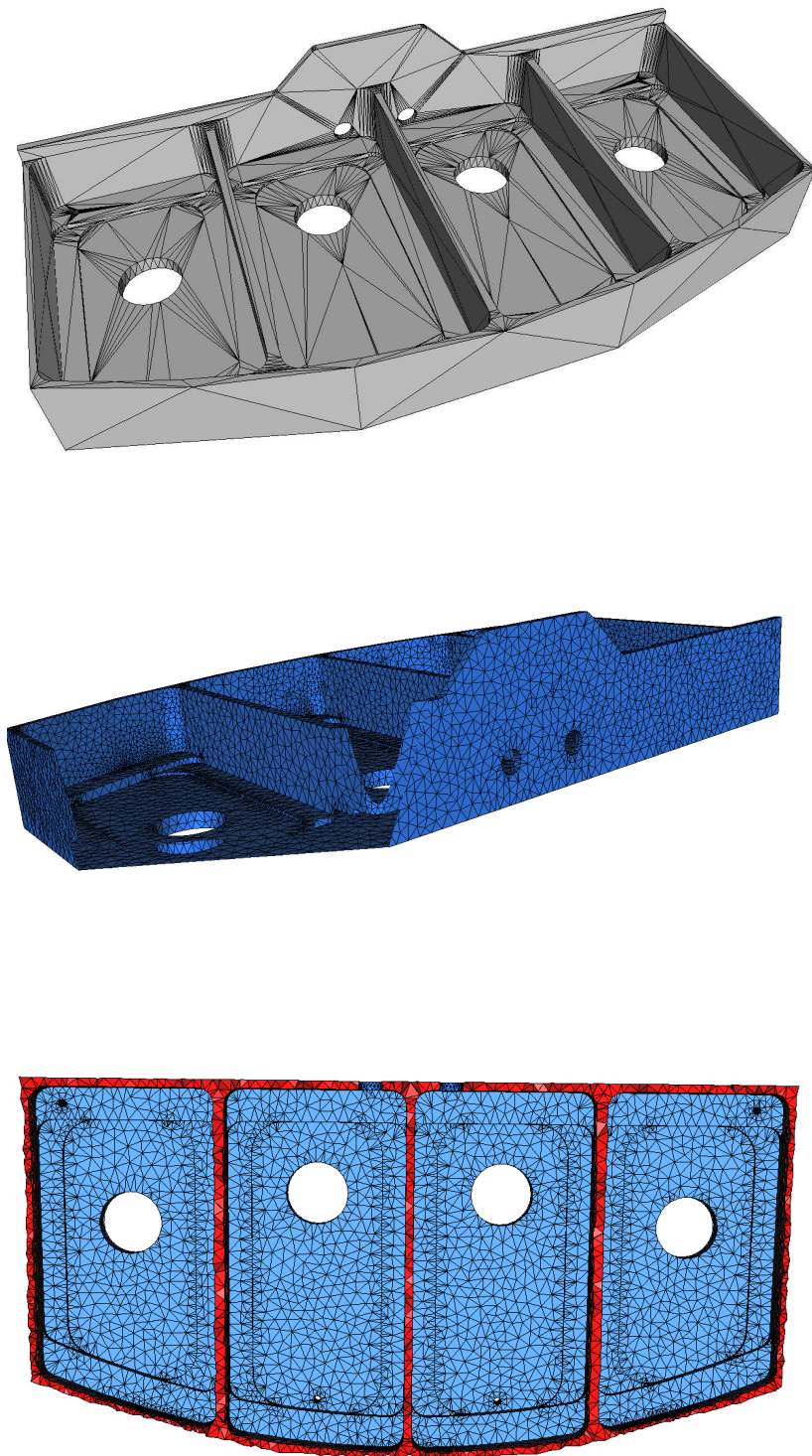


FIG. 4.8: Modèle **CAD**. En haut, le maillage de surface d'origine. Au milieu, un zoom sur une partie du maillage de surface produit par l'algorithme. En bas, une vue de dessus, qui montre le maillage volumique produit par l'algorithme, vu en coupe. 85 483 sommets et 280 006 tétraèdres.





## Chapitre 5

# Architecture générique pour l'implantation

### 5.1 Introduction

Les algorithmes de maillage présentés dans cette thèse (voir sections 3.3 et 4.2) procèdent par raffinement de Delaunay, en appliquant une liste de règles, ordonnées par priorité. Dans ce chapitre, on présente une architecture générique pour implanter un algorithme de ce genre. L'élément central de cette architecture est une classe générique, appelée *niveau de raffinement*, qui est associée à une des règles du processus de raffinement, et prend en charge un type d'élément du maillage (arêtes, triangles, ou tétraèdres). Plusieurs instances de *niveau de raffinement* peuvent être empilées, et reliées ensemble, afin d'implanter un processus de raffinement complet. L'architecture est suffisamment flexible pour implanter tous les algorithmes de maillages qui raffinent une triangulation de Delaunay et itèrent un ensemble de règles en suivant un ordre de priorité.

Cette architecture a été utilisée dans la bibliothèque CGAL [CGA], pour implanter plusieurs algorithmes [Rin07, RY07a], ainsi que les algorithmes décrits dans cette thèse. L'architecture générique décrite dans ce chapitre utilise des paradigmes de programmation du langage C++. Cependant, cette architecture pourrait être implémentée à d'autres langages de programmation.

Afin d'être plus concret dans la description de cette architecture générique, la section 5.2 présente un cas typique d'algorithme de génération de maillage qui procède par raffinement de Delaunay en itérant une série de règles : l'algorithme de Shewchuk [She02a] pour mailler un volume polyédrique. La section 5.3 présente la classe générique et explique comment plusieurs instances de cette classe peuvent être utilisées ensemble. Enfin, la section 5.4 montre comment cette architecture permet d'implanter une grande variété d'algorithmes par raffinement de Delaunay.

### 5.2 Exemple typique

Dans cette section, on présente l'algorithme proposé par Shewchuk [She02a] pour mailler un domaine volumique dont le bord est un polyèdre. Cet algorithme est un exemple typique d'algorithme procédant au raffinement d'une

triangulation de Delaunay en itérant un ensemble de règles ordonnées par ordre de priorité.

Dans cet algorithme, le bord du domaine à mailler est décrit par un complexe linéaire par morceau (abrégé en « PLC », pour *piecewise linear complex*). En dimension 3, un PLC est une collection de faces de dimensions 0, 1, et 2, respectivement appelé sommets, arêtes et facettes. Les sommets sont des points, les arêtes sont des segments de droite, et les facettes sont des régions polygonales. Les facettes ne sont pas forcément convexes ou même simplement connexes.

Les faces d'un PLC satisfont les conditions suivantes :

- le bord de toute face du PLC est l'union de faces du PLC,
- l'intersection de toute paire de face du PLC est soit vide, soit l'union de faces du PLC.

Étant donné un PLC  $\mathcal{C}$ , le domaine  $D$  à mailler est décrit comme l'union de composantes connexes bornées de  $\mathbb{R}^3 \setminus \mathcal{C}$ . Ainsi le domaine est borné, et son bord est l'union de faces du PLC.

Le but de l'algorithme est de construire un maillage simplicial du domaine, c'est à dire une triangulation  $\mathcal{T}$  telle que les sommets de  $\mathcal{C}$  soient des sommets de  $\mathcal{T}$ , et telle que chaque facette de  $\mathcal{C}$  soit une union de facettes de  $\mathcal{T}$ . De plus, les tétraèdres de  $\mathcal{T}$  qui sont inclus dans le domaine  $D$  doivent satisfaire des critères de taille et de forme.

Le processus de raffinement maintient une triangulation de Delaunay. Il démarre avec la triangulation de Delaunay des sommets de  $\mathcal{C}$ , et ajoute itérativement de nouveaux sommets, afin d'assurer que les facettes de  $\mathcal{C}$  sont représentées dans  $\mathcal{T}$  par une union de facettes, et pour éliminer les mauvais éléments. Dans le cas présent, les mauvais éléments sont les tétraèdres qui ne satisfont pas les critères de taille ou de forme.

Rappelons quelques définitions données par Shewchuk [She02a]. Les sommets insérés sur les arêtes de  $\mathcal{C}$  partitionnent ces segments en segments plus petits, appelés *sous-segments*. L'un des objectifs de l'algorithme est d'assurer que tous les sous-segments soient ultimement des arêtes de la triangulation de Delaunay. Une fois cela assuré, chaque facette  $F$  de  $\mathcal{C}$  apparaît comme une union de facettes dans la triangulation de Delaunay bi-dimensionnelle  $\mathcal{T}_2(F)$  des sommets de  $\mathcal{T}$  situés dans l'hyperplan support de  $F$ . On appelle *sous-facettes* de  $F$  les facettes de  $\mathcal{T}_2(F)$  incluses dans  $F$ . Dans ce qui suit, les faces (sommets, arêtes et facettes) de  $\mathcal{C}$  sont appelées *contraintes d'entrée*, et les sous-segments et sous-facettes sont appelés *éléments contraints*.

On dit qu'un point *accroche* un sous-segment ou une sous-facette si il est situé dans sa plus petite sphère circonscrite. Un sous-segment ou une sous-facette est dit *accroché* si un sommet du maillage courant l'accroche. Un élément qui n'est accroché par aucun sommet de  $\mathcal{T}$  est nécessairement un élément de la triangulation  $\mathcal{T}$  (selon le critère de la sphère vide). L'un des objectifs de l'algorithme est d'assurer que les sous-segments et sous-facettes ne sont pas accrochés.

L'algorithme raffine les sous-segments accrochés en ajoutant leur milieu dans les sommets du maillage. Les sous-facettes accrochées et les mauvais tétraèdres sont raffinés en ajoutant le centre de leur sphère circonscrite. Les sous-segments accrochés ont priorité sur les sous-facettes accrochées, et sur les mauvais tétraèdres, et les sous-facettes accrochées ont priorité sur les mauvais

tétraèdres. De plus, l'algorithme assure qu'aucun centre de plus petite sphère circonscrite de sous-facette n'est ajouté aux sommets du maillage s'il accroche un sous-segment, et qu'aucun centre de sphère circonscrite de tétraèdre n'est ajouté si il accroche un sous-segment ou une sous-facette.

Plus précisément, l'algorithme appelle trois procédures `refine_edge`, `refine_facet_or_edge` et `refine_tetrahedron_or_facet_or_edge` à des moments choisis, en appliquant itérativement les trois règles suivantes. Les règles sont appliquées selon un ordre de priorité, de sorte que la règle  $R_i$  est appliquée seulement si aucun règle  $R_j$  avec  $j < i$  ne peut être appliquée.

**R1** S'il existe un sous-segment accroché  $e$ , appeler `refine_edge(e)`.

**R2** S'il existe une sous-facette accrochée  $f$ , appeler `refine_facet_or_edge(f)`.

**R3** S'il existe un mauvais tétraèdre  $t$ , appeler `refine_tetrahedron_facet_or_edge(t)`.

Les procédures auxiliaires `refine_edge`, `refine_facet_or_edge` et `refine_tetrahedron_facet_or_edge` sont définies ci-dessous.

`refine_edge(e)` : Calculer le milieu  $p$  de  $e$ . Insérer  $p$  dans les sommets de la triangulation.

`refine_facet_or_edge(f)` : Calculer le centre  $p$  de la sphère circonscrite à  $f$ .

- Si  $p$  accroche une arête  $e$ , appeler `refine_edge(e)`,
- sinon insérer  $p$  comme sommet de la triangulation.

`refine_tetrahedron_facet_or_edge(t)` : Calculer le centre  $p$  de la sphère circonscrite à  $t$ .

- Si  $p$  accroche une arête  $e$ , appeler `refine_edge(e)`,
- Si  $p$  accroche une facette  $f$ , appeler `refine_facet_or_edge(f)`,
- sinon insérer  $p$  comme sommet de la triangulation.

Avec des critères de taille et de forme appropriés, il est prouvé que cet algorithme termine si les contraintes d'entrée ne forment pas d'angles plus petits que  $90^\circ$ . Les angles considérés ici sont les angles dièdres, les angles arête-facette, et les angles arête-arête, formés par deux faces de  $\mathcal{C}$  qui ne sont ni incidentes ni disjointes.

### 5.3 Une architecture logicielle générique

#### La classe générique *niveau de raffinement*

Dans tous les algorithmes de raffinement de Delaunay de notre connaissance, l'insertion de points est guidée par un ensemble de règles. De manière générale, chaque règle est liée au raffinement d'un type d'éléments du maillage. Dans le cas de l'algorithme présenté à la section 5.2 page précédente, ces éléments sont les sous-segments, les sous-facettes et les tétraèdres. Les règles sont appliquées selon un ordre de priorité précis.

L'élément central de notre architecture est une classe nommée *niveau de raffinement*. Chaque instance de cette classe est associée à une règle du processus de raffinement. Plusieurs instances de niveaux de raffinement sont combinés pour composer un algorithme de raffinement de Delaunay. Par exemple, l'implantation de l'algorithme de la section 5.2 page 74 met en jeu trois niveaux, c'est-à-dire un pour chaque règle.

Dans notre architecture, les rôles d'un niveau de raffinement sont de maintenir un ensemble d'éléments à raffiner, appelé `to_be_refined` dans la suite de ce chapitre, et d'assurer, si possible, le raffinement de ces éléments. Un élément à raffiner peut être soit un mauvais élément qui ne satisfait pas les critères de taille ou de forme, soit un élément contraint, qui doit être raffiné pour s'assurer que la triangulation de Delaunay contient une représentation des contraintes de l'entrée.

Pour raffiner un élément donné, le niveau de raffinement doit, dans un premier temps, calculer un point de raffinement, puis vérifier l'acceptation de ce point par les autres niveau qui composent l'implantation de l'algorithme et, si le point est accepté, finalement l'insérer comme nouveau sommet de la triangulation. Un point de raffinement est accepté s'il n'accroche aucun élément contraint géré par les niveaux de plus grande priorité. Enfin, l'ensemble `to_be_refined` d'un niveau de raffinement doit être mise à jour :

- à chaque insertion d'un point de raffinement, quelque soit le niveau responsable de l'insertion de ce point,
- à chaque rejet d'un point de raffinement qui accroche un élément contraint géré par ce niveau.

Pour accomplir ces différentes tâches, un niveau de raffinement fournit plusieurs méthodes. Ces méthodes peuvent être classées en deux groupes :

- les méthodes du premier groupe, appelées *méthodes primitives* (ou plus simplement primitives) sont toutes les méthodes qui doivent être spécialisées dans chaque instance de niveau de raffinement, car leur code dépend de la règle implantée. En particulier, ce groupe inclut les méthodes qui gèrent l'ensemble `to_be_refined`, qui peut être ordonné ou non, en fonction du niveau. Ainsi la méthode `is_to_be_refined_empty()` informe si l'ensemble est vide, la méthode `get_next_element()` retourne un élément choisi dans l'ensemble, et la méthode `pop_next_element()` retire cet élément de l'ensemble. En plus de ces méthodes, ce groupe des méthodes primitives contient les méthodes suivantes :
  - `scan_triangulation()` parcourt toute la triangulation, et initialise l'ensemble `to_be_refined`.
  - `refinement_point(e)` calcule un point de raffinement pour l'élément  $e$ .
  - `check_refinement_point(p)` vérifie si le point de raffinement  $p$ , calculé par ce niveau ou par un niveau de priorité plus basse, peut être accepté. Si  $p$  a été calculé par un niveau de priorité plus basse, et accroche un élément contraint  $e$  géré par le niveau courant, alors  $p$  est rejeté, et  $e$  est inséré dans la l'ensemble `to_be_refined`.
  - `after_insertion(p)` met à jour l'ensemble `to_be_refined` après l'insertion du point  $p$ .
- Les méthodes du second groupe, appelées *fonctions squelettes*, sont partagées par tous les niveaux de raffinement. Ces méthodes forment le squelette du processus de raffinement. Elles font appel aux primitives, et s'occupent de la communication entre les différents niveaux mis en jeu dans le même processus de raffinement de Delaunay. Parmi ces méthodes, on trouve les méthodes suivantes :

- `previous()` retourne un pointeur vers le niveau précédent dans le processus de raffinement. Les niveaux de raffinement formant le processus de raffinement sont ordonnés par ordre de priorité décroissante des règles associées. La méthode `previous()` renvoie donc un pointeur vers le niveau de priorité immédiatement supérieure.
- `check_refinement_point_by_all_levels(p)` fait vérifier l’acceptation du point  $p$  par tous les niveaux de priorité supérieure et par le niveau courant.
- `process_one_element()` s’occupe d’un élément de l’ensemble `to_be_refined` : Il trouve le prochain élément  $e$  à raffiner, calcule le point de raffinement  $p$  de  $e$ , appelle la méthode `check_refinement_point_by_all_levels(p)` et éventuellement, si le point  $p$  est accepté, insère  $p$  dans la triangulation puis appelle `after_insertion(p)`.
- `refine()` commande le processus de raffinement. Il appelle la méthode `refine()` du niveau précédent, puis appelle `process_one_element()` jusqu’à ce que les ensembles `to_be_refined` du niveau courant et des niveaux de plus grande priorité soient vides. Finalement, le processus de raffinement de Delaunay formé par l’ensemble des niveaux de raffinement impliqués est déclenché par un appel à la méthode `refine()` du niveau de plus basse priorité.

Comme tous les niveaux de raffinement partagent un ensemble de fonctionnalités (les méthodes squelettes), il est naturel de les implémenter comme des classes dérivées d’une classe de base commune, appelée `Meshier_level_base`. Chaque niveau de raffinement dérivé doit alors fournir des versions spécialisées des méthodes primitives. Cependant, comme les méthodes primitives sont appelées par les méthodes squelettes, elles doivent être définies comme des méthodes virtuelles. Pour éviter le coût à l’exécution de l’utilisation de méthodes virtuelles, `Meshier_level_base` peut être modèle de classe, paramétrée par une classe qui fournit les méthodes primitives. Par exemple, le niveau de plus basse priorité de l’algorithme présenté à la section 5.2 page 74, nommé `Tetrahedra_level`, serait alors défini ainsi :

```
class Tetrahedra_level
    : public Meshier_level_base<Tetrahedra_level_policy> {
    ...
}
```

Dans ce choix d’implantation, `Tetrahedra_level_policy` fournit les méthodes primitives dont a besoin `Meshier_level_base`, et la classe `Tetrahedra_level` hérite des méthodes squelettes de `Meshier_level_base`. Cette construction a le défaut de nécessiter une classe supplémentaire, ce qui rend plus difficile la maintenance de l’ensemble, si les méthodes de `Tetrahedra_level` et `Tetrahedra_level_policy` s’appellent. Il est possible de pour ainsi dire « fusionner » les deux classes `Tetrahedra_level` et `Tetrahedra_level_policy`. Pour cela, on a utilisé le paradigme de programmation (*design pattern*) nommé *Curiously Recurring Template Pattern* (CRTP), décrit par James O. Coplien dans [Cop96]. Selon ce paradigme, la classe de base `Meshier_level_base` est un modèle de classe, et la classe dérivée est fourni comme paramètre du modèle de la classe de base. Le niveau `Tetrahedra_level` est défini ainsi :

```
class Tetrahedra_level
```

```

    : public Mesher_level_base<Tetrahedra_level> {
    ...
}

```

De cette façon, `Tetrahedra_level` a la fois dérive de `Mesher_level_base`, et ainsi hérite des méthodes squelettes, et en même temps fournit les méthodes primitives dont les méthodes squelettes ont besoin. En fait, le modèle de classe `Mesher_level_base` n'est pas seulement paramétré par le type de le niveau dérivé, mais aussi par trois autres types : le type `Tr` de la classe qui implante la triangulation de Delaunay, le type d'élément géré par ce niveau de raffinement, nommé `Element`, et le type du niveau précédent, nommé `Previous`. Un schéma de la classe `Mesher_level_base` est donné dans le listing 5.1 page ci-contre.

### Lier les niveaux de raffinement

Un processus de raffinement de Delaunay met en jeu en général plusieurs niveaux de raffinement, qui doivent être liées les uns aux autres.

Par exemple, l'algorithme de la section 5.2 met en jeu trois niveaux, appelés `Edges_level`, `Facets_level` et `Tetrahedra_level`, chacun correspondant à une règle de raffinement différente. Le paramètre `Previous` de `Edges_level` est instancié par une classe représentant un niveau de raffinement trivial, dont les méthodes ne font rien. Les paramètres `Previous` de `Facets_level` et `Tetrahedra_level` sont instanciés respectivement par `Edges_level` et `Facets_level`. Les niveaux de raffinement sont instanciés et liés par ordre de priorité décroissante. La liaison est effectuée via le paramètre template `Previous`, et le pointeur retourné par la méthode `previous()`. Ainsi, un niveau de raffinement peut appeler les méthodes des niveaux de plus grande priorité. Cependant, la communication dans l'autre sens entre les niveaux est aussi nécessaire. En particulier, quand un niveau insère un point de raffinement, l'ensemble `to_be_refined` de toutes les autres niveaux doit être mis à jour. Ceci est permis par le paradigme des *visiteurs*.

Le paradigme des visiteurs est utilisé pour modifier le comportement d'un algorithme [SLL02]. Un visiteur est similaire à un foncteur mais comportent plusieurs méthodes, plutôt que simplement un `operator()`. Dans notre architecture, les méthodes `refine()` et `process_one_element()` du modèle `Mesher_level_base` sont paramétrées par un type `Visitor`, et ont un argument supplémentaire de type `Visitor`. La classe de visiteur passée comme argument doit nécessairement spécifier plusieurs méthodes, parmi lesquelles une méthode `after_insertion(p)`, qui a pour rôle de mettre à jour les ensembles `to_be_refined` de tous les niveaux après l'insertion d'un point de raffinement. Dans la classe de base `Mesher_level_base`, et donc dans tous les niveaux de raffinement, la méthode `visitor.after_insertion()` est appelée après chaque insertion d'un point de raffinement (les insertions sont effectuées dans la méthode `process_one_element()`). Le code de `refine()` et `process_one_element()` est donné de façon schématique dans le listing 5.1 page précédente. Les visiteurs sont instanciés et créés après l'instanciation et la création des niveaux de raffinement. De cette façon, un visiteur peut stocker une référence vers chaque niveau mise en jeu dans le processus de raffinement. La méthode `after_insertion(p)` d'un visiteur peut alors appeler les méthodes primitives `after_insertion(p)` de chaque niveau. Pour des raisons pratiques, les visiteurs des différents niveaux

Listing 5.1: Schéma de la classe `Meshes_level_base`

---

```

template <class Triangulation ,
           class Element ,
           class CRTP_Derived ,
           class Previous_level = Null_mesher_level>
class Meshes_level_base {

    Previous_level& previous; // reference to the previous mesher level
    Triangulation& tr;       // reference to the triangulation

public:
    CRTP_Derived& derived() {
        return static_cast<CRTP_Derived&>(*this);
    }

    bool are_lists_empty() {
        return previous().is_list_empty() &&
            derived().are_lists_empty();
    }

    Encroaching_status check_refinement_point_by_all_levels(Point p) {
        Encroaching_status status =
            previous.check_refinement_point_by_all_levels(p);
        if( status == NO_ENCROACHMENT )
            return derived().check_refinement_point(p);
        else
            return status;
    }

    template <class Visitor>
    void process_one_element(Visitor visitor = Null_visitor) {
        Element e = derived().get_next_element();
        Triangulation::Point p = derived().refinement_point(e);
        Encroachment_status point_status =
            check_refinement_point_by_all_levels(p)
        if( point_status == ACCEPTED ) {
            visitor.before_insertion(p);
            Triangulation::Vertex_handle vh = tr.insert(p);
            derived().after_insertion();
            visitor.after_insertion(vh);
            pop_next_element();
        } else if( point_status == REJECTED_FOREVER ) {
            pop_next_element();
        }
    }

    template <class Visitor>
    void refine(Visitor visitor = Null_visitor) {
        previous().refine(visitor.previous());
        while( ! derived().is_list_empty() ) {
            process_one_element(visitor);
            previous().refine(visitor.previous());
        }
    }
}

```

---



sont liés ensemble par des pointeurs, accessibles par les méthodes `previous()`, exactement comme les niveaux le sont. L'image 5.1 montre les dépendances et les connexion entre les différents niveaux et les visiteurs, sur l'exemple de l'algorithme spécifié en section 5.2 page 74.

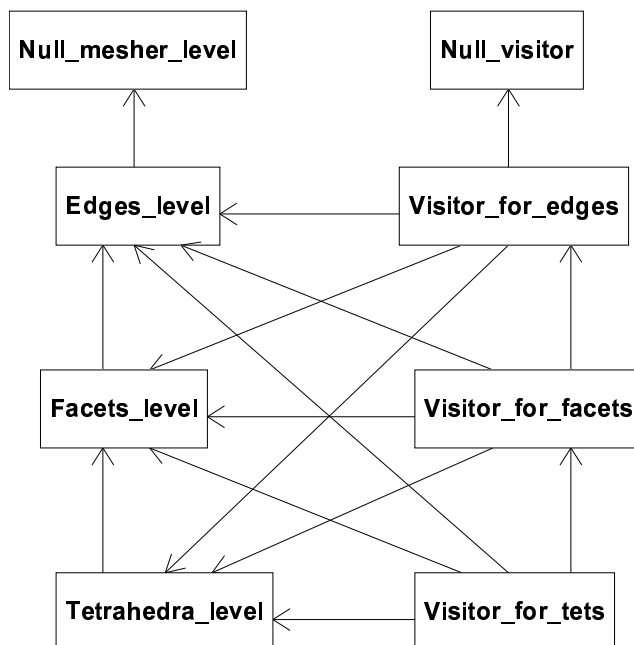


FIG. 5.1: Diagramme de classes UML des niveaux et visiteurs impliqués dans l'implantation de l'algorithme de la section 5.2 page 74. Un lien d'une classe  $A$  à une classe  $B$  signifie que  $A$  peut avoir besoin d'appeler des méthodes de  $B$ , et doit donc être instancié après  $B$ . Les classes `Null_mesher_level` et `Null_visitor` sont respectivement un niveau de raffinement trivial et un visiteur trivial, dont les méthodes ne font rien.

## 5.4 Applications et exemples

Cette section montre comment l'architecture décrite dans la section précédente peut être utilisée pour implanter plusieurs algorithmes de raffinement de Delaunay.

### Mailleur 2D de graphes planaires linéaires

Dans [She00], Shewchuk présente un algorithme pour mailler des domaines plans dont les frontières et contraintes sont décrites par un graphe linéaire (un graphe d'arêtes rectilignes, plongé dans le plan). C'est un algorithme de raffinement de Delaunay composé de deux règles. La première raffine les arêtes contraintes, et la deuxième raffine les mauvaises facettes. Cet algorithme s'implante naturellement, en suivant notre architecture, avec deux niveaux, respectivement appelés `Edges_level_2` et `Facets_level_2`.

L’algorithme présenté dans [She00] est capable de gérer les petits angles, c’est-à-dire le cas où deux segments de contraintes de l’entrée, partageant un sommet, forment un angle plus petit que  $90^\circ$ . L’algorithme a deux particularités pour éviter que le processus de raffinement n’entre dans une boucle infinie d’insertion de points, aux alentours des petits angles.

Premièrement, les arêtes contraintes partageant un sommet sont regroupées dans un même paquet, appelé *cluster* par Shewchuk, de telle sorte que deux arêtes autour d’un même sommet formant un angle plus petit que  $60^\circ$  sont placées dans le même paquet. Les arêtes d’un paquet ne sont pas raffinées en leur milieu, mais en un point choisi parmi les intersections de l’arête avec un ensemble approprié de cercles concentriques, centrés sur le sommet commun du paquet d’arêtes. De cette façon, après quelques insertions, les arêtes d’un même paquet ont la même longueur, ce qui arrête une éventuelle chaîne de raffinement en cascade. Cette particularité de l’algorithme est codée dans la méthode primitive `compute_refinement_point()` de `Edges_level_2`.

La seconde particularité de cet algorithme concerne le raffinement des mauvaises facettes. Quand le point de raffinement d’une mauvaise facette accroche une arête contrainte qui fait parti d’un paquet, l’algorithme teste si le raffinement des arêtes du paquet risque de compromettre la terminaison de l’algorithme. Dans un tel cas, le point de raffinement de la mauvaise facette est rejeté, mais le raffinement des arêtes accrochées du paquet est refusé : la mauvaise facette est définitivement retirée de l’ensemble `to_be_refined`, de sorte qu’elle ne sera plus jamais reconsidérée et restera éventuellement présente dans le maillage final. Pour pouvoir implanter ce comportement, la méthode `check_refinement_point()` a un type de retour avec trois valeurs possibles : `PERMITTED` (accepté), `REJECTED` (rejeté) et `REJECTED_FOREVER` (rejeté définitivement). Quand le point est accepté, il est inséré dans la triangulation. Quand le point est rejeté, il n’est pas inséré dans la triangulation, mais une arête accrochée est placée dans l’ensemble `to_be_refined` du niveau `Edges_level_2`, et donc le paquet sera raffiné. Quand le point est rejeté définitivement, il n’est pas inséré dans la triangulation, et la mauvaise facette est retirée de l’ensemble `to_be_refined` du niveau `Facets_level_2`. Elle reste dans le maillage, bien qu’elle soit mauvaise, et le paquet ne sera pas raffiné.

## Mailleur 3D de domaines polyédriques, avec petits angles

L’algorithme de Cheng et al. [CDRR04] est différent de celui décrit à la section 5.2, par sa capacité à gérer les petits angles. Les arêtes qui sont à l’intersection de deux facettes formant un angle dièdre plus petit que  $90^\circ$  sont appelées *arêtes vives*, et leurs sous-segments sont appelés *segments vifs*. Les sommets qui sont aux extrémités des segments vifs, ou aux sommets de petits angles sont appelés *sommets vifs*. L’idée de l’algorithme est de protéger les sommets vifs et les arêtes vives par une union de boules, à l’intérieur desquelles l’insertion de centres de tétraèdres est interdite.

L’initialisation de l’algorithme consiste à calculer des boules de protection centrées sur les sommets vifs, et à insérer dans la triangulation les intersections entre ces boules et les arêtes du complexe.

Ensuite, la première phase de l’algorithme, appelée `CONFORM`, assure la représentation des contraintes du maillage, dans la triangulation de Delaunay, et raffine les sous-segments et les sous-facettes jusqu’à ce que les sous-segments

des arêtes vives aient une taille reliée à la distance caractéristique locale du PLC, ce qui permettra de calculer des boules de protection de taille adéquate autour des arêtes vives. Dans [CDRR04], cette première phase est décrite par trois règles. Cependant, il est pratique de scinder la troisième règle (qui gère à la fois les arêtes et les facettes contraintes) en deux sous-règles. On peut implanter cette phase avec quatre niveaux. Les deux premières règles sont similaires à celles de la section 5.2 page 74, et les niveaux de raffinement associés, `Edges_level` et `Facets_level`, sont pratiquement identiques à ceux évoqués à la section 5.3 page 78. Il y a néanmoins deux différences. La première concerne le point de raffinement de certaines arêtes à l'intérieur des boules de protection des sommets. Pour ces arêtes, appelées *arêtes bouclier*, le point de raffinement n'est pas le milieu de l'arête, mais le milieu d'un arc sphérique ayant les mêmes extrémités (selon la stratégie SOS définie dans [CCY04]). Ce détail est pris en compte dans la méthode `compute_refinement_point` de `Edges_level`. La deuxième différence est que la boule diamétrale d'une sous-facette peut contenir des points (c'est-à-dire que ces points accrochent la sous-facette). Les sous-facettes ont seulement à être présentes dans la triangulation de Delaunay. Seules les sous-facettes de la boîte englobantes doivent avoir une sphère diamétrale vide. Les méthodes `scan_triangulation()`, `check_refinement_point()` et `after_insertion()` de `Facets_level` sont spécialisées pour prendre cette différence en compte. Les deux dernières règles de la phase CONFORM peuvent être écrites ainsi :

- Règle 3** Si le milieu d'un sous-segment vif  $s$  accroche un sous-segment  $e$   
et si  $s$  et  $e$  sont contenus dans des éléments disjoints du PLC,  
appeler `refine_edge(e)`
- Règle 3'** Si le milieu d'un sous-segment vif  $s$  accroche une sous-facette  $f$   
et si  $s$  et  $f$  sont contenus dans deux éléments disjoints du PLC,  
appeler `refine_facet(f)`

Ces deux règles sont implantées par deux niveaux de raffinement respectivement similaires à `Edges_level` et `Facets_level`. Les seules différences sont la sélection des éléments à raffiner dans ces deux niveaux. Ceci est pris en compte dans les méthodes `scan_triangulation` et `after_insertion`.

Quand la phase CONFORM est terminée, l'ensemble des boules de protection des arêtes vives est calculé, en doublant les rayons des plus boules englobant les sous-segments des arêtes vives. L'algorithme entre alors dans une phase appelée REFINE. Cette phase est un raffinement de Delaunay analogue à celui présenté dans la section 5.2 page 74, si ce n'est qu'aucun point de raffinement n'est accepté dans l'union des boules de protection. La phase REFINE est implantée par trois niveaux. Les niveaux `Edges_level` et `Facets_level`, utilisés pendant la phase CONFORM, sont réutilisés à l'identique, en conjonction avec un nouveau niveau `Tetrahedra_level`. Ce `Tetrahedra_level` a une version spécialisée de la méthode `check_refinement_point`, qui retourne la valeur `REJECTED_FOREVER` si le point de raffinement candidat est situé à l'intérieur d'une boule de protection.

Dans l'article [CDRR05], Cheng et al. étendent leur algorithme de maillage, en lui ajoutant une phase de *pompage*, nommée PUMPING, pour se débarrasser des tétraèdres slivers, le seul type de tétraèdre dégénéré qu'un processus de raffinement de Delaunay n'enlève pas du maillage. Cette phase n'ajoute aucun point, et ne déplace aucun point. Le *pompage* consiste à changer la triangulation

de Delaunay, pour en faire une triangulation régulière, en assignant des poids adaptés aux sommets de la triangulation. De façon évidente, cette phase de pompage n'est pas un processus de raffinement de Delaunay, et l'architecture décrite dans ce chapitre n'est pas adaptée pour l'implanter. Cependant, la phase `REFINE` de l'algorithme doit être légèrement modifiée, afin que la phase `PUMPING` ne détruise pas la représentation dans la triangulation des éléments contraints. Une nouvelle règle anticipant la phase `PUMPING` doit être ajoutée. Cette règle concerne à nouveau soit des sous-segments, soit des sous-facettes, et elle est implantée par deux nouveaux niveaux, similaires à `Edges_level` et `Facets_level`, sauf pour la sélection des éléments à raffiner. Le critère de sélection prend en compte le poids maximal qui pourra être affecté aux sommets pendant la phase `PUMPING`. Ainsi, pour implanter la nouvelle phase `CONFORM` de l'algorithme de [CDRR05], six niveaux sont liés. La phase *Pumping* elle-même est implantée sans niveau de raffinement.

### Mailleur de volumes bornés par des surfaces lisses, et lisses par morceaux

L'article [BO05] de Steve Oudot et Jean-Daniel Boissonnat détaille un processus de raffinement de Delaunay pour calculer un maillage 2D approchant une surface lisse. L'algorithme maintient une triangulation de Delaunay 3D, et sa restriction à la surface, obtenue en sélectionnant dans la triangulation de Delaunay les facettes dont l'arête de Voronoï duale intersecte la surface. Cet algorithme peut être implanté en utilisant un seul niveau, appelé `Surface_level`. Les critères pour sélectionner les mauvais triangles sont la forme de ceux-ci, et leur taille. Le point de raffinement d'une facette est un point d'intersection entre la surface et l'arête duale de la facette. La connaissance de la surface dont a besoin l'algorithme se réduit à un oracle, capable de détecter les intersections entre la surface et des segments requêtes. Cela permet à l'algorithme d'être appliqué pour tout type de surfaces pour lesquelles on peut définir un tel oracle, et notamment pour les surfaces définies par une équation implicite, pour les ensembles de niveau d'images 3D noir et blanc, et pour les surfaces définies implicitement par un ensemble de points.

Dans le chapitre 3, le mailleur de surface de [BO05] est étendu en un mailleur volumique, capable de mailler un domaine borné par des surfaces lisses. Par rapport à l'article [BO05], la différence est l'ajout des règles R1 et R3 (voir la description de l'algorithme à la section 3.3 page 23). La première donne une condition supplémentaire pour raffiner une facette : si une facette de la triangulation de Delaunay restreinte à la surface n'a pas ses trois sommets sur la surface, alors elle doit être raffinée. Cette règle est juste un critère supplémentaire pour raffiner les facettes. L'implantation des deux règles R1 et R2 a été faite en réutilisant le niveau de raffinement `Surface_level` décrit ci-dessus, et en lui faisant utiliser un nouveau jeu de critère, pour gérer la liste des facettes à raffiner dans les fonctions `scan_triangulation` et `after_insertion`. La règle R3, qui gère les tétraèdres qui ne sont pas conformes aux critères de taille et de forme a été implantée en utilisant le niveau de raffinement `Tetrahedra_level` déjà utilisé ci-dessus. Finalement, l'implantation de l'algorithme décrit à la section 3.3 a été réalisé en liant ces deux niveaux `Surface_level` et `Tetrahedra_level` ensemble.

Dans le chapitre 4, cet algorithme est à étendu pour pouvoir mailler des volumes dont le bord est composés de surfaces lisses par morceaux. Dans ce dernier algorithme, le bord du domaine à mailler est un complexe, dans lequel les facettes sont des morceaux de surfaces lisses, qui se recollent le long de segments de courbes, qui sont les arêtes du complexe. L'ensemble des arêtes du complexe forment le 1-squelette. La différence avec l'algorithme de la section 3.3 est l'ajout de deux règles supplémentaires, prioritaires sur les précédentes, pour raffiner les arêtes qui font partie de la triangulation de Delaunay restreinte aux segments de courbes et qui ne satisfont pas à certains critères topologiques ou géométriques. Ces deux règles supplémentaires ont été implanté dans un niveau de raffinement nommé `Curved_edges_level`. L'implantation de l'algorithme a ainsi été facilement réalisé par la liaison des trois niveaux de raffinement `Curved_edges_level`, `Surface_level`, et `Tetrahedra_level`. La figure 5.2 montre l'empilement des niveaux de raffinement utilisés par l'implantation de cet algorithme.

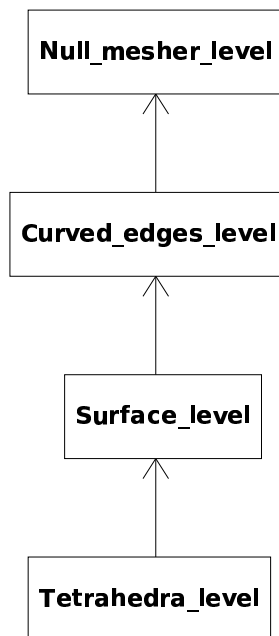


FIG. 5.2: Diagramme de classes UML des niveaux impliqués dans l'implantation de l'algorithme de la section 3.3. Un lien d'une classe  $A$  à une classe  $B$  signifie que  $A$  peut avoir besoin d'appeler des méthodes de  $B$ , et doit donc être instancié après  $B$ . La classes `Null_mesher_level` est un niveau de raffinement trivial, dont les méthodes ne font rien.

## Chapitre 6

# Conclusion

Cette thèse s’appuie sur le travail théorique de Steve Oudot et Jean-Daniel Boissonnat [BO05], qui étudie l’algorithme de Chew appliqué aux surfaces lisses. Cet algorithme construit un  $\varepsilon$ -échantillon de la surface, par un processus de raffinement d’une triangulation de Delaunay à une seule règle, et extrait de la triangulation un maillage de surface.

Dans un premier temps, cet algorithme a été étendu en un algorithme de génération de maillages volumiques. Pour cela, deux règles ont été ajoutées, afin d’une part de séparer les sommets insérés dans le volume des sommets insérés sur la surface, et d’autre part de contrôler la taille et la forme des tétraèdres du volume.

Dans un deuxième temps, l’algorithme a de nouveau été étendu, afin de pouvoir mailler des volumes dont le bord est une surface lisse par morceau. Deux nouvelles règles ont été ajoutées au processus de raffinement. Cette fois, l’étude de la précision de l’approximation ne s’appuie plus que sur certains lemmes de la théorie développée par Steve Oudot et Jean-Daniel Boissonnat, et montre que l’ensemble de sommets produit par l’algorithme satisfait les conditions d’Edelsbrunner et Shah. Les résultats de [BO05] ne peuvent en effet plus être appliqués tels quels, car ils ne sont valables que pour des surfaces lisses. Cet algorithme souffre pour le moment d’une limitation importante, car son résultat n’est pas garanti si le bord du domaine comporte des angles plus petits que  $90^\circ$ .

Toutes les différentes déclinaisons de l’algorithme décrites dans cette thèse ont été implantées en utilisant une architecture logicielle commune, qui permet d’organiser le code, en factorisant en un même endroit un certain nombre de fonctionnalités utilisées par chaque règle d’un processus de raffinement de Delaunay. L’implantation de ces algorithmes est en cours de transfert dans la bibliothèque CGAL. Pour le moment, certaines parties du code sont encore au stade expérimental.

### 6.1 Travaux futurs

Au terme de ce travail de thèse, un certain nombre de problèmes théoriques ou pratiques restent à étudier afin d’obtenir un algorithme de maillage et une implantation qui correspondent aux besoins de l’industrie.

## Gérer les petits angles

L'algorithme décrit et étudié dans le chapitre 4 souffre d'une restriction importante sur la classe de volume pouvant être considérés en entrée. En effet, si le bord du volume à mailler comporte des morceaux de surface adjacents formant un angle plus petit que  $90^\circ$ , alors l'algorithme n'est pas garanti de produire un maillage approchant correctement l'objet à mailler, ni même de terminer.

Un article récent de Tamal Dey et al. [CDR07] présente un algorithme semblable à celui présenté dans cette thèse. L'algorithme de Tamal Dey ne souffre pas de la limitation ci-dessus. Cependant, il dépend d'un oracle difficile à implanter. Plus récemment encore, un article publié à l'International Meshing Roundtable d'octobre 2007 [?] présente une version plus facilement implantable du même algorithme.

L'une des premières tâches à l'issue de cette thèse sera d'étudier la solution de Tamal Dey, afin de concevoir un algorithme hybride entre le sien et celui présenté dans cette thèse.

## Concevoir de nouveaux oracles

Pour le moment, le seul oracle implanté capable de gérer des surfaces lisses par morceaux travaille à partir d'un maillage surfacique de l'objet à mailler. Pour pouvoir toucher le plus d'applications possibles, il est nécessaire d'enrichir la classe de surfaces pouvant borner les domaines maillés par l'algorithme. Cela revient à implanter de nouveaux oracles. Un certain nombre d'oracles sont d'ores et déjà envisagés.

**Maillages de subdivision** Un oracle est actuellement en cours d'implantation, pour traiter des volumes bornés par des maillages obtenus à partir d'une subdivision lisse par morceau d'un maillage de contrôle. Dans ce genre d'application, l'entrée de l'algorithme serait alors un maillage de surface, triangulaire, avec un certain nombre d'arêtes marquées. Le bord du volume à mailler serait alors défini comme la subdivision lisse par morceaux de ce maillage de contrôle.

**Surfaces algébriques** Il est envisagé d'implanter prochainement un oracle qui permettrait de mailler directement des surfaces définies algébriquement. Cela permettrait de mailler des données issues de logiciels de CAO. Les arêtes de  $\mathcal{L}$  et les faces de  $\mathcal{S}$  seraient par exemple définies par des B-splines, ou des NURBS.

**Images 3D** Dans le domaine des images 3D médicales, l'oracle implanté ne peut être utilisé que par l'algorithme du chapitre 3. L'algorithme du chapitre 4 serait pourtant très intéressant. En effet, une image médicale peut être labélisée : à chaque voxel de l'image est associé un tissu. Une telle image médicale, une fois lissée, comporte naturellement des jonctions « en T », c'est à dire un segment de courbe lisse qui est l'intersection du bord d'au moins trois tissus différents. L'implantation d'un oracle qui permettrait de mailler ce genre d'image médicale est envisagée. La difficulté sera d'arriver à détecter les segments de courbes de manière fiable dans l'image lissée, sans pour autant faire

une recherche exhaustive au niveau des voxels de l'image. Cet oracle pourrait aussi être appliqué à des images 3D issues d'études sismiques.

## Optimisation de maillage

Actuellement, la seule phase d'optimisation de maillage utilisée est l'exudation des slivers. Cette optimisation exécute uniquement des bascules d'arêtes et de facettes. Après le travail de Pierre Alliez, Olivier Devillers et Jane Tournois sur l'optimisation de maillages en dimension 2 [?], l'optimisation de maillages surfaciques et volumiques est le prochain sujet d'étude. L'article [?] présenté par Bryan M. Klingner et Jonathan R. Shewchuk présente différentes stratégies d'optimisation de maillage, et décrit un algorithme d'optimisation qui les combine. Cet algorithme est coûteux, mais ses résultats sont prometteurs. Ce sujet est l'objet de la thèse de doctorat commencée par Jane Tournois.

## Travaux sur l'architecture logicielle

Du côté de l'architecture logicielle, trois tâches restent à faire.

- Au cours de l'implantation des différentes déclinaisons de l'algorithme présenté dans cette thèse, il est apparu que les objets *visiteurs* utilisés pour lier les différents niveaux de raffinement font pratiquement toujours un travail similaire, qui consiste à mettre à jour les listes d'éléments à raffiner au cours de l'insertion d'un nouveau sommet par l'un des niveaux. Il est envisagé prochainement d'essayer d'organiser le code des visiteurs de manière à factoriser encore une fois l'essentiel des fonctionnalités communes à tous les visiteurs. Cela devrait permettre d'implanter encore plus facilement de nouveaux algorithmes de raffinement de Delaunay.
- Pour le moment, notre algorithme, qui procède par raffinement de Delaunay, ne fait qu'*insérer* de nouveaux sommets dans le maillage. Pour pouvoir implanter une fonctionnalité de dé-raffinement d'un maillage, l'architecture logicielle doit être amendée de nouvelles fonctions, qui auront la charge des opérations de suppression de sommets.
- Enfin, il serait nécessaire de pouvoir utiliser plus efficacement la puissance des processeurs actuels, dotés de plusieurs cœurs. Pour cela, il faut envisager d'implanter une version parallélisable de l'algorithme. Cela nécessite un travail important sur les algorithmes de construction de triangulation de Delaunay, dans la bibliothèque CGAL. Ce travail est en cours de réalisation, et permettra bientôt d'envisager d'étudier la parallélisation des algorithmes de maillages.





# Bibliographie

- [AB99] Nina Amenta et Marshall Bern: *Surface Reconstruction by Voronoi Filtering*. Discrete and Computational Geometry, 22(4) :481–504, 1999. 22, 23, 26, 41, 46
- [ACSYD05] Pierre Alliez, David Cohen-Steiner, Mariette Yvinec et Mathieu Desbrun: *Variational Tetrahedral Meshing*. Dans *Proceedings SIGGRAPH*, 2005. 29, 33
- [BA76] I. Babuska et A. K. Aziz: *On the Angle Condition in the Finite Element Method*. SIAM Journal on Numerical Analysis, 13(2) :214–226, 1976. <http://link.aip.org/link/?SNA/13/214/1>. 9
- [BEG90] Marshall Bern, David Eppstein et John Gilbert: *Provably good mesh generation*. Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on, pages 231–241, 1990. 10, 12
- [BEG94] Marshall Bern, David Eppstein et John Gilbert: *Provably good mesh generation*. J. Comput. Syst. Sci., 48(3) :384–409, 1994. 12
- [BGH<sup>+</sup>97] Houman Borouchaki, Paul Louis George, Frédéric Hecht, Patrick Laug et Éric Saltel: *Delaunay mesh generation governed by metric specifications. Part I. Algorithms*. Finite Elements in Analysis and Design, 25(1-2) :61–83, 1997. 13
- [BGM97] Houman Borouchaki, Paul Louis George et Bijan Mohammadi: *Delaunay mesh generation governed by metric specifications. Part II. applications*. Finite Elements in Analysis and Design, 25(1-2) :85–109, 1997. 13
- [BGR88] B.S. Baker, E. Grosse et C.S. Rafferty: *Nonobtuse triangulation of polygons*. Discrete and Computational Geometry, 3(1) :147–168, 1988. 9
- [BO05] Jean Daniel Boissonnat et Steve Oudot: *Provably good sampling and meshing of surfaces*. Graphical Models, 67(5) :405–451, 2005. 13, 14, 21, 23, 24, 25, 26, 28, 32, 34, 46, 47, 83
- [BO06] Jean Daniel Boissonnat et S. Oudot: *Provably Good Sampling and Meshing of Lipschitz Surfaces*. Dans *Proc. 22nd Annu. ACM Sympos. Comput. Geom.*, 2006. Preprint available

at [ftp://ftp-sop.inria.fr/geometrica/soudot/preprints/boiss\\_oudot\\_SoCG\\_06.pdf](ftp://ftp-sop.inria.fr/geometrica/soudot/preprints/boiss_oudot_SoCG_06.pdf). 46

- [Boi06] Jean Daniel Boissonnat: *Vornoi Diagrams, Triangulations and Surfaces*, chapitre 5. Inria, 2006. 46, 47
- [Bow81] A. Bowyer: *Computing Dirichlet tessellations*. The Computer Journal, 24(2) :162, 1981. 11, 13
- [BWS<sup>+</sup>87] P. Baehmann, S. Wittchen, M. Shephard, K. Grice et M. Yerry: *Robust, geometrically based, automatic two-dimensional mesh generation*. International Journal for Numerical Methods in Engineering, 24(6) :1043–1078, 1987. 9
- [CCY02] David Cohen-Steiner, Éric Colin de Verdière et Mariette Yvinec: *Conforming Delaunay triangulations in 3D*. The 18th Annual Symposium on Computational Geometry (SCG '02), pages 199–208, 2002. 12
- [CCY04] David Cohen-Steiner, Éric Colin de Verdière et Mariette Yvinec: *Conforming Delaunay triangulations in 3D*. Computational Geometry : Theory and Applications, 28(2-3) :217–233, 2004. 12, 82
- [CD02] Siu Wing Cheng et Tamal K. Dey: *Quality meshing with weighted Delaunay refinement*. Dans *SODA '02 : Proceedings of the thirteenth annual ACM-SIAM symposium on discrete algorithms*, pages 137–146, 2002, ISBN 0-89871-513-X. 13
- [CDE<sup>+</sup>00] Siu Wing Cheng, Tamal K. Dey, H. Edelsbrunner, M. A. Facello et S. H. Teng: *Silver exudation*. J. ACM, 47(5) :883–904, 2000, ISSN 0004-5411. 13, 34
- [CDR07] Siu Wing Cheng, Tamal K. Dey et Edgar A. Ramos: *Delaunay Refinement for Piecewise Smooth Complexes*. Proc. 18th Annual ACM-SIAM Symposium Discrete Algorithms (SODA'07), pages 1096–1105, 2007. 14, 54
- [CDRR04] Siu Wing Cheng, Tamal K. Dey, Edgar A. Ramos et Tathagata Ray: *Quality meshing for polyhedra with small angles*. Dans *SCG '04 : Proceedings of the twentieth annual symposium on Computational geometry*, pages 290–299. ACM Press, 2004, ISBN 1-58113-885-7. 13, 81, 82
- [CDRR05] Siu Wing Cheng, Tamal K. Dey, Edgar A. Ramos et Tathagata Ray: *Weighted Delaunay refinement for polyhedra with small angles*. Dans *Proceedings 14th International Meshing Roundtable, IMR2005*, 2005. 83
- [CFF85] James C. Cavendish, David A. Field et William H. Frey: *An approach to automatic three-dimensional finite element mesh generation*. International journal for numerical methods in engineering, 21(2) :329–347, 1985. 12

- [CGA] CGAL, *Computational Geometry Algorithms Library*. <http://www.cgal.org>. 73
- [CGAL] *The CGAL Library*. <http://www.cgal.org>, Release 3.1 (<http://www.cgal.org>). 34
- [Che89] L. Paul Chew: *Guaranteed-quality triangular meshes*. *Department of Computer Science*. rapport technique, Cornell University, TR-89-983, 1989. 12
- [Che93] L. Paul Chew: *Guaranteed-quality mesh generation for curved surfaces*. Dans *SCG '93 : Proceedings of the ninth annual symposium on Computational geometry*, pages 274–280. ACM Press, 1993, ISBN 0-89791-582-8. 13
- [CL04] Frédéric Chazal et André Lieutier: *Stability and homotopy of a subset of the medial axis*. Dans *SM '04 : Proceedings of the 9th ACM symposium on Solid modeling and applications*, 2004. 34
- [CNWC<sup>+</sup>01] JB Cavalcante Neto, PA Wawrzynek, MTM Carvalho, LF Martha et AR Ingraffea: *An Algorithm for Three-Dimensional Mesh Generation for Arbitrary Regions with Cracks*. *Engineering with Computers*, 17(1) :75–91, 2001. 14
- [Cop96] James O. Coplien: *Curiously recurring template patterns*. Dans Stanley B. Lippman (rédacteur) : *C++ gems*, tome 5, pages 135–144. SIGS Publications, Inc., 1996. 78
- [CP03] Siu Wing Cheng et Sheung Hung Poon: *Graded conforming Delaunay tetrahedralization with bounded radius-edge ratio*. Dans *SODA '03 : Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 295–304. Society for Industrial and Applied Mathematics, 2003, ISBN 0-89871-538-5. 13
- [Dey06] Tamal K. Dey: *Curve and Surface Reconstruction : Algorithms with Mathematical Analysis*. Cambridge University Press, 2006. 46
- [DW06] Qiang Du et Desheng Wang: *Recent progress in robust and quality Delaunay mesh generation*. *Journal of Computational and Applied Mathematics*, 195(1-2) :8–23, 2006. 9
- [EG02] H. Edelsbrunner et D. Guoy: *An Experimental Study of Sliver Exudation*. *Engineering With Computers*, Special Issue on ‘Mesh Generation’ (10th IMR 2001), 18(3) :229–240, 2002. 34
- [ES97] H. Edelsbrunner et N. R. Shah: *Triangulating Topological Spaces*. *International Journal on Computational Geometry*, 7 :365–378, 1997. 45
- [FBG96] Pascal J. Frey, Houman Borouchaki et Paul Louis George: *Delaunay tetrahedralization using an advancing-front approach*. 5th International Meshing Roundtable, pages 31–48, 1996. 14

- [FBG98] Pascal J. Frey, Houman Borouchaki et Paul Louis George: *3D Delaunay mesh generation coupled with an advancing-front approach*. Computer methods in applied mechanics and engineering, 157(1) :115–131, 1998. 14
- [Fed69] H. Federer: *Geometric Measure Theory*. Springer-Verlag, 1969. 22
- [Fre87] William .H. Frey: *Selective Refinement : a New Strategy For Automatic Node Placement in graded triangular meshes*. International Journal for Numerical Methods in Engineering, 24(11) :2183–2200, 1987. 12
- [GB98] Paul Louis George et Houman Borouchaki: *Delauney Triangulation and Meshing : Application to Finite Elements*. Hermes Science Publications, Paris, 1998. 13
- [Geo71] J.A. George: *Computer implementation of the finite element method*. Thèse de doctorat, Dept. of Computer Science, Stanford University., 1971. 10
- [Geo97] Paul Louis George: *Improvements on Delaunay-based three-dimensional automatic mesh generator*. Finite Elements in Analysis and Design, 25(3) :297–317, 1997. 14
- [Geo01] Paul Louis George (rédacteur): *Maillage et adaptation (Traité Mécanique et Ingénierie des matériaux, Série Méthodes numériques)*. Lavoisier, Paris, 2001. 9, 14
- [GF99] Paul Louis George et Pascal Frey: *Maillages : Applications aux éléments finis*. Hermes Sciences, 1999. 9, 11
- [GHS90] Paul Louis George, Frédéric Hecht et Éric Saltel: *Fully automatic mesh generator for 3D domains of any shape*. IMPACT of Computing in Science and Engineering, 2(3) :187–218, 1990. 13
- [GHS91] Paul Louis George, Frédéric Hecht et Éric Saltel: *Automatic mesh generator with specified boundary*. Computer Methods in Applied Mechanics and Engineering, 92(3) :269–288, 1991. 13
- [GS94] Paul Louis George et Éric Seveno: *The advancing-front mesh generation method revisited*. International Journal for Numerical Methods in Engineering, 37(21) :3605–3619, 1994. 10
- [Her82] François Hermeline: *Triangulation automatique d'un polyèdre en dimension  $N$* . RAIRO numerical analysis, 16(3), 1982. 12
- [LC87] W.E. Lorensen et H.E. Cline: *Marching cubes : A high resolution 3D surface construction algorithm*. Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pages 163–169, 1987. 10
- [Lee99] CK Lee: *Automatic adaptive mesh generation using metric advancing front approach*. Engineering Computations, 16(2) :230–263, 1999. 11

- [LL96] T. S. Lau et S. H. Lo: *Finite element mesh generation over analytical curved surfaces*. Computers and Structures, 59(2) :301–309, 1996. 11
- [LL98] S. H. Lo et T. S. Lau: *Mesh generation over curved surfaces with explicit control on discretization error*. Engineering Computations, 15(3) :357–373, 1998. 11
- [Lo85] S. H. Lo: *A new mesh generation scheme for arbitrary planar domains*. International Journal for Numerical Methods in Engineering, 21(8) :1403–1426, 1985. 10
- [Lo89] S. H. Lo: *Delaunay triangulation of non-convex planar domains*. International Journal for Numerical Methods in Engineering, 28(11) :2695–2707, 1989. 10
- [Lo91a] S. H. Lo: *Volume discretization into tetrahedra-I. Verification and orientation of boundary surfaces*. Computers and Structures, 39(5) :493–500, 1991. 10
- [Lo91b] S. H. Lo: *Volume Discretization into Tetrahedra-II. 3D Triangulation by Advancing Front Approach*. Computers and Structures, 39(5) :501–511, 1991. 10
- [Löh96] Rainald Löhner: *Progress in grid generation via the advancing front technique*. Engineering with Computers, 12(3) :186–210, 1996. 10
- [Löh97] Rainald Löhner: *Automatic unstructured grid generators*. Finite Elem. Anal. Des., 25(1-2) :111–134, 1997, ISSN 0168-874X. 10
- [LP88] Rainald Löhner et Paresh Parikh: *Generation of three-dimensional unstructured grids by the advancing-front method*. International Journal for Numerical Methods in Fluids, 8 :1135–1149, 1988. 10
- [LS03] Francois Labelle et Jonathan R. Shewchuk: *Anisotropic voronoi diagrams and guaranteed-quality anisotropic mesh generation*. Dans *SCG '03 : Proceedings of the nineteenth annual symposium on Computational geometry*, pages 191–200, New York, NY, USA, 2003. ACM Press, ISBN 1-58113-663-3. 41
- [LT01] X. Y. Li et S. H. Teng: *Generating well-shaped Delaunay meshed in 3D*. Dans *SODA '01 : Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 28–37. Society for Industrial and Applied Mathematics, 2001, ISBN 0-89871-490-7. 13
- [Mer91] Marshal L. Merriam: *An efficient advancing front algorithm for Delaunay triangulation*. AIAA, Aerospace Sciences Meeting, 1991. 14

- [MTT99] G. L. Miller, D. Talmor et S. H. Teng: *Data generation for geometric algorithms on non uniform distributions*. International Journal of Computational Geometry and Applications, 9(6) :577–599, 1999. 29, 33, 41
- [MV92] SA Mitchell et SA Vavasis: *Quality mesh generation in three dimensions*. 8th ACM Symp. Comp. Geom, pages 212–221, 1992. 10
- [MV00] Scott A. Mitchell et Stephen A. Vavasis: *Quality Mesh Generation in Higher Dimensions*. SIAM J. Comput., 29(4) :1334–1370, 2000, ISSN 0097-5397. 10
- [MW95] David L. Marcum et Nigel P. Weatherill: *Unstructured grid generation using iterative point insertion and local reconnection*. AIAA Journal, 33(9) :1619–1625, 1995. 14
- [ORY05] Steve Oudot, Laurent Rineau et Mariette Yvinec: *Meshing Volumes Bounded by Smooth Surfaces*. Dans *Proc. 14th International Meshing Roundtable*, pages 203–219, 2005. 14
- [Owe98] S.J. Owen: *A survey of unstructured mesh generation technology*. 7th International Meshing Roundtable, 3(6), 1998. 9
- [PSB<sup>+</sup>07] Jean Philippe Pons, Florent Ségonne, Jean Daniel Boissonnat, Laurent Rineau, Mariette Yvinec et Renaud Keriven: *High-quality consistent meshing of multi-label datasets*. Dans *Information Processing in Medical Imaging*, pages 198–210, Jul 2007. 14, 37
- [Ras98] Alain Rassinoux: *Generation and optimization of tetrahedral meshes by advancing front technique*. International Journal for Numerical Methods in Engineering, 41(4) :651–674, 1998. 14
- [Rin07] Laurent Rineau: *2D Conforming Triangulations and Meshes*. Dans CGAL Editorial Board (rédacteur) : *CGAL User and Reference Manual*. 3.3 édition, 2007. [http://www.cgal.org/Manual/3.3/doc\\_html/cgal\\_manual/packages.html#Pkg:Mesh2](http://www.cgal.org/Manual/3.3/doc_html/cgal_manual/packages.html#Pkg:Mesh2). 73
- [Rup93] Jim Ruppert: *A New and Simple Algorithm for Quality 2-D Mesh Generation*. Proc. ACM Symp. on Disc. Alg, pages 83–92, 1993. 12
- [Rup95] Jim Ruppert: *A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation*. Journal of Algorithms, 18(3) :548–585, 1995. 12, 41
- [RY07a] Laurent Rineau et Mariette Yvinec: *3D Surface Mesh Generation*. Dans CGAL Editorial Board (rédacteur) : *CGAL User and Reference Manual*. 3.3 édition, 2007. [http://www.cgal.org/Manual/3.3/doc\\_html/cgal\\_manual/packages.html#Pkg:SurfaceMesher3](http://www.cgal.org/Manual/3.3/doc_html/cgal_manual/packages.html#Pkg:SurfaceMesher3). 73
- [RY07b] Laurent Rineau et Mariette Yvinec: *A generic software design for Delaunay refinement meshing*. Computational Geometry : Theory and Applications, 38(1-2) :100–110, 2007. 15

- [RY07c] Laurent Rineau et Mariette Yvinec: *Meshing Volumes Bounded by Piecewise Smooth Surfaces*. Dans *Proc. 16th International Meshing Roundtable (Á paraÁ@tre)*, 2007. 14
- [SG91] M. Shephard et M. Georges: *Automatic three-dimensional mesh generation by the finite octree technique*. *International Journal for Numerical Methods in Engineering*, 32(4) :709–749, 1991. 10
- [She] Jonathan R. Shewchuk: *What Is a Good Linear Finite Element ? Interpolation, Conditioning, Anisotropy, and Quality Measures*. <http://www.cs.berkeley.edu/~jrs/papers/elemj.pdf>. 9
- [She96] Jonathan R. Shewchuk: *Triangle : Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*. *Applied Computational Geometry : Towards Geometric Engineering*, 1148 :203–222, 1996. 12
- [She97] Jonathan R. Shewchuk: *Delaunay Refinement Mesh Generation*. Thèse de doctorat, School of Computer Science, Carnegie Mellon University, 1997. 9
- [She98] Jonathan R. Shewchuk: *Tetrahedral mesh generation by Delaunay refinement*. Dans *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 86–95, 1998. 21, 40, 41
- [She00] Jonathan R. Shewchuk: *Mesh Generation for Domains with Small Angles*. Dans *Proc. 16th Annual ACM Symposium Comput. Geom.*, pages 1–10, 2000. 80, 81
- [She02a] Jonathan R. Shewchuk: *Delaunay Refinement Algorithms for Triangular Mesh Generation*. *Computational Geometry : Theory and Applications*, 22 :21–74, 2002. 12, 73, 74
- [She02b] Jonathan R. Shewchuk: *What is a good linear element ? interpolation, conditioning, and quality measures*. 11th International Meshing Roundtable, pages 115–126, 2002. 9
- [SLL02] Jeremy G. Siek, Lie Quan Lee et Andrew Lumsdaine: *The boost graph library : user guide and reference manual*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002. 78
- [TW00] S. H. Teng et C. W. Wong: *Unstructured Mesh Generation : Theory, Practice, and Perspectives*. *Int. J. Comput. Geometry Appl.*, 10(3) :227–266, 2000. 29
- [Wat81] DF Watson: *Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes*. *The Computer Journal*, 24(2) :167, 1981. 11, 13
- [YS83] MA Yerry et MS Shephard: *A Modified Quadtree Approach To Finite Element Mesh Generation*. *Computer Graphics and Applications, IEEE*, 3(1) :39–46, 1983. 9



- [YS84] M.A. Yerry et M.S. Shephard: *Automatic three-dimensional mesh generation by the modified-octree technique*. International Journal for Numerical Methods in Engineering, 20(11) :1965–1990, 1984.  
9