



HAL
open science

Algorithmes parallèles de manipulation de maillages

Guillaume Vialaneix

► **To cite this version:**

Guillaume Vialaneix. Algorithmes parallèles de manipulation de maillages. Autre [cs.OH]. Télécom ParisTech, 2012. Français. NNT : 2012ENST0068 . tel-01081682

HAL Id: tel-01081682

<https://pastel.hal.science/tel-01081682>

Submitted on 10 Nov 2014

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.



EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Signal – Images »

présentée et soutenue publiquement par

Guillaume VIALANEIX

le 26 novembre 2012

**Algorithmes parallèles de
manipulation de maillages**

Directeur de thèse : **Tamy BOUBEKEUR**
Co-encadrement de la thèse : **Yvan FOURNIER**

Jury

Mme Raphaëlle CHAINE, LIRIS, Université de Lyon
M. Jean-François REMACLE, Université Catholique de Louvain
Mme Isabelle BLOCH, Telecom ParisTech, CNRS-LTCl
M. Houman BOROUCHEKI, Projet GAMMA, INRIA
M. Tamy BOUBEKEUR, Telecom ParisTech, CNRS-LTCl
M. Yvan FOURNIER, MFEE, EDF R&D

Rapporteur
Rapporteur
Examineur
Examineur
Directeur
Encadrant

TELECOM ParisTech

École de l'Institut Télécom - membre de ParisTech

**T
H
È
S
E**

Résumé

Nous traitons dans cette thèse des différents aspects de la manipulation de maillages, et de la façon dont ces opérations peuvent être effectuées en parallèle, ou en mémoire distribuée, à l'heure où les GPUs comme les super-calculateurs deviennent de plus en plus utilisés. Ainsi, nous présentons des algorithmes de lissage surfacique et volumique, inspirés d'algorithmes de traitement d'images (filtre bilatéral, histogrammes locaux). Après ces manipulations de géométrie, nous abordons des problématiques topologiques comme le remaillage local, dans le but de générer, à partir d'un maillage tétraédrique, une couche limite de cellules prismatiques et hexaédriques de bonne qualité, permettant des simulations de mécanique des fluides dans ces zones proches de la surface. Enfin, nous décrivons une technique de maillage basée sur des interactions particulières, permettant de générer des maillages à dominante quadrangulaire.

Abstract

This thesis deals with different aspects of mesh processing, and the way those operations can be done in parallel, or using distributed memory, when GPUs and supercomputers are more and more commonly used. We present surfacic and volumetric mesh smoothing algorithms, based upon image processing techniques (bilateral filter, local histograms). After those geometric considerations, we talk about topologic methods, as local remeshing, enabling one to generate, from a tetrahedral mesh, one layer of good quality prisms and hexahedron, allowing fluid mechanics simulations in those near-surface areas. Finally, we present a meshing technique based upon particular interactions, in order to construct quad-dominant meshes.

Table des matières

Table des matières	iii
1 Introduction	1
1.1 Contexte	1
1.2 La simulation numérique	3
1.3 Le traitement de maillages	4
1.3.1 Maillage polygonal	4
1.3.2 Méthodes de traitement	9
1.3.3 Objectifs	9
1.4 Le calcul parallèle	10
1.4.1 Introduction	10
1.4.2 Définitions	11
1.4.3 Exemples	11
1.4.4 Parallélisation avec OpenMP	12
1.4.5 Parallélisation avec MPI	13
1.4.6 Déterminisme	13
1.5 <i>Code_Saturne</i>	13
1.6 Contributions de la thèse	14
1.6.1 Vue d'ensemble	14
1.6.2 Publications	15
2 Vers la parallélisation	17
2.1 Voisinage distribué	17
2.1.1 Problématique	17
2.1.2 Moyennes locales	18
2.1.3 Adaptation parallèle	19
2.2 Approximation	21
2.2.1 Matrice jacobienne	21
2.2.2 Adaptation parallèle	24
2.2.3 Résultats	25
2.3 Discussion	25
3 Filtrage surfacique	27

3.1	Contexte	27
3.1.1	Bruit et filtres passe-bas	27
3.1.2	Travaux précédents	28
3.2	Filtre bilatéral séparable	33
3.2.1	Accélérations	33
3.2.2	Notre approche	34
3.2.3	Résultats	35
3.2.4	Discussion	40
3.3	Histogrammes locaux lissés	42
3.3.1	Présentation	42
3.3.2	Médiane géométrique	44
3.3.3	Extension aux maillages	45
3.3.4	Résultats	47
4	Lissage volumique	49
4.1	Contexte	50
4.1.1	Maillages volumiques	50
4.1.2	Notion de dual	50
4.1.3	Travaux existants	50
4.2	Dégauchissement	52
4.2.1	Problématique	52
4.2.2	Notre approche	52
4.2.3	Résultats	53
4.3	Filtre bilatéral volumique	55
4.3.1	Problématique	55
4.3.2	Qualité de maillage	56
4.3.3	Notre approche	57
4.3.4	Résultats	58
5	Remaillage : Prismes	63
5.1	Introduction	63
5.2	Création de la couche limite	66
5.2.1	Partition de la surface	67
5.2.2	Suppression	69
5.2.3	Projections	69
5.2.4	Cas d'un plan	70
5.2.5	Arêtes vives	70
5.2.6	Concavités	71
5.2.7	Relaxation	71
5.2.8	Découpe en couches limites	72
5.3	Résultats	72
6	Maillages cristallins	75

6.1	Travaux précédents	75
6.2	Notre approche	78
6.2.1	Principe	78
6.2.2	Application en 2D	80
6.3	Notre méthode	81
6.3.1	Maillage du contour	82
6.3.2	Initialisation	83
6.3.3	Mouvement des points	84
6.3.4	Triangulation et quadrangulation	85
6.4	Résultats	85
6.4.1	Résultats dans le plan	85
6.4.2	Analyse	89
6.4.3	Résultats en 3D	90
6.4.4	Perspectives	93
7	Conclusion	95
7.1	Rappel des contributions	95
7.1.1	Parallélisation d'algorithmes	95
7.1.2	Filtrage surfacique	95
7.1.3	Optimisation hexaédrique	96
7.1.4	Création de couches limites	96
7.1.5	Maillages cristallins	97
7.2	Notions émergentes	97
7.3	Perspectives	97
7.3.1	Structures de données	97
7.3.2	Communication	98
	Bibliographie	99

Chapitre 1

Introduction

1.1 Contexte

Les simulations numériques constituent aujourd'hui l'un des piliers de la recherche scientifique et de certains domaines de l'ingénierie, permettant d'une part une réduction drastique des coûts de fabrication, en évitant la construction de nombreuses maquettes et prototypes, et d'autre part une meilleure observation et compréhension de phénomènes se déroulant à des échelles trop grandes ou trop petites pour pouvoir être réalisées en pratique.

C'est au cœur de ces simulations que se situe le maillage, une discrétisation la plus juste possible de la géométrie du problème, permettant une estimation suffisamment précise des quantités continues de la physique. Les équations aux dérivées partielles régissant l'évolution des variables du problème peuvent alors, sous leur forme discrète, révéler le mouvement des particules, les changements de températures ou les variations de pression.

Mais pour garantir la justesse de cette discrétisation, le maillage doit répondre à des critères de qualité, dépendant des méthodes numériques de résolution utilisées. Ces critères impliquent un certain agencement spatial et topologique des différents éléments du maillage, ainsi que, par exemple, leur nombre. De ce fait, une géométrie complexe, sur laquelle un calcul de haute précision doit être effectué, requerrait un très grand nombre de sommets, faces et cellules.

Or, un maillage de grande taille implique de longs temps de calcul, et demande un espace de stockage important en mémoire, et il arrive que l'un de ces paramètres

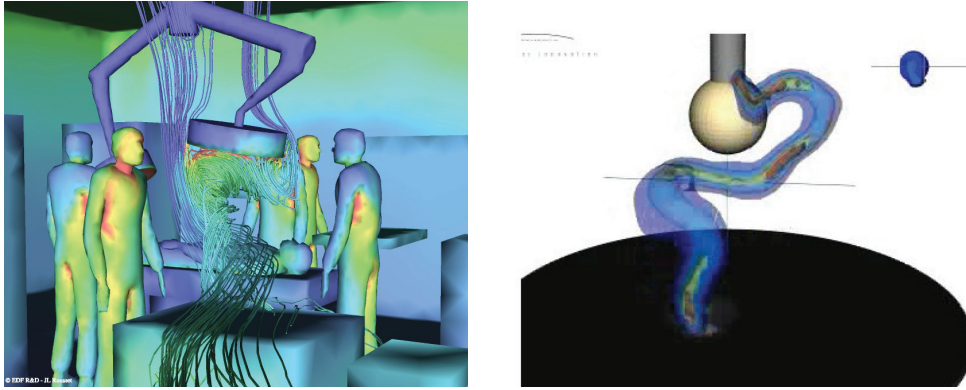


FIGURE 1.1: Exemple d'utilisation des logiciels de simulation. Gauche : calcul de flux d'air dans une salle d'opération. Droite : simulation d'un arc électrique. *source : EDF R&D*

soit une limitation pour la simulation. Il est donc nécessaire d'optimiser le maillage en vue de la simulation, et de nombreux algorithmes de lissage, remaillages, raffinement abordent cette problématique.

La solution la plus classique pour réduire le temps d'exécution d'une tâche est de la paralléliser : si deux opérations n'ont aucun lien entre elles, alors elles peuvent être effectuées en même temps. En suivant cette logique, la parallélisation de la mémoire, ou mémoire *distribuée*, permet aussi (et surtout) de traiter de grands volumes de données en séparant spatialement le problème sur différents processeurs.

Afin de repousser encore les limites des possibilités de calcul qui s'offrent à nous, il est intéressant de chercher à combiner les deux types d'optimisation, autrement dit : comment optimiser le maillage lorsque celui-ci se trouve dans une architecture parallèle ?

Dans la suite de cette introduction, nous proposons une vue d'ensemble sur le traitement de maillages, et sur la parallélisation. Nous commencerons par décrire brièvement le processus d'une simulation numérique (1.2), avant d'aborder la définition d'un maillage (1.3), puis les grandes familles de traitement de maillage, et enfin les objectifs recherchés. Nous décrirons ensuite (1.4) le calcul parallèle, au travers de quelques exemples. Nous clôturerons ce chapitre avec une présentation des grandes lignes de ce manuscrit (1.6).

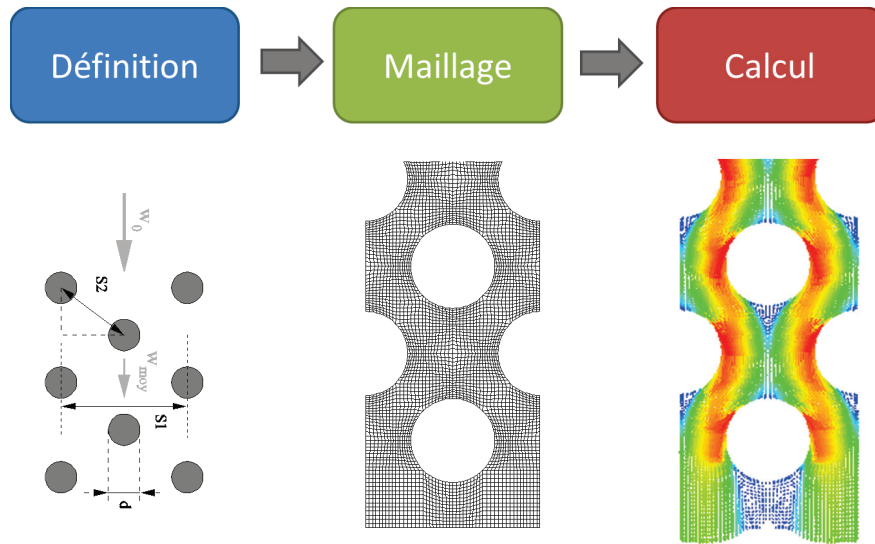


FIGURE 1.2: Présentation du processus général d'une simulation numérique. De gauche à droite : la conception du modèle, la réalisation du maillage, l'application du code de calcul.

1.2 La simulation numérique

La réalisation complète d'une simulation numérique est en général constituée des éléments suivants :

- Conception du modèle
- Maillage de la géométrie
- Application du code de calcul

Nous avons représenté ces étapes sur la figure 1.2, à partir d'un des maillages utilisés comme cas-test pour un logiciel industriel.

Les trois étapes de la simulation sont parfaitement indépendantes et il est de fait toujours possible d'utiliser un premier logiciel pour définir le modèle, un second pour mailler celui-ci, et enfin un code de calcul pour effectuer la simulation. En pratique, cette indépendance est rapidement limitée par les différents formats de fichiers et les informations transmises d'un logiciel au suivant.

Ainsi, en regroupant deux ou trois de ces étapes au sein d'un même logiciel, on accroît d'une part les capacités intrinsèques de chaque étape, mais également leur inter-opérabilité. En combinant par exemple la conception du modèle et son maillage, il est possible pour la méthode de maillage de s'appuyer sur les

définitions géométriques du modèle, sans avoir à détecter les arêtes vives ou autres caractéristiques du maillage.

De même, si les deux dernières étapes sont regroupées, il est possible d'imaginer un maillage initial qui serait modifié (raffiné, par exemple) en fonction de la sortie du code de calcul, et ainsi optimiser le maillage à l'utilisation qui en est faite.

1.3 Le traitement de maillages

1.3.1 Maillage polygonal

Géométrie et topologie

Un maillage polygonal surfacique est la donnée d'un ensemble de polygones qui approximent une surface. Ils peuvent être représentés par une série de d'éléments désorganisés (souples de polygones), ou bien être donnés sous la forme de maillage indexé, représentation basée sur des graphes. Nous n'allons, pour la suite de ce document, considérer que des maillages indexés, dont la structure permet bien plus aisément d'en manipuler les éléments.

Ainsi, un maillage porte essentiellement deux informations. D'une part, une information géométrique, avec, le plus souvent, la liste des positions dans l'espace des sommets, en général sous forme d'un triplet de coordonnées (x, y, z) pour chaque sommet (*entrelacées*), mais peuvent aussi être données sous la forme de trois listes des coordonnées dans chaque dimension (toutes les abscisses, puis toutes les ordonnées, ...). D'autre part, l'information topologique, faisant intervenir des simplexes de dimension supérieure : arête (1-simplexe), face (2-simplexe) et cellule (3-simplexe) dans le cas d'un maillage volumique.

La topologie d'un maillage va ainsi être constituée de tableaux n -uplets reliant les simplexes de deux ordres différents. On parlera de connectivité *ascendante* lorsque un k -simplexe est décrit par les j -simplexes auxquels il appartient, avec $j > k$. À l'inverse, la connectivité est dite *descendante* lorsque un k -simplexe est décrit par les j -simplexes qui le constituent, avec $j < k$. Il existe également les représentations *nodales*, dans lesquelles chaque simplexe est décrit à l'aide des sommets qui le compose.

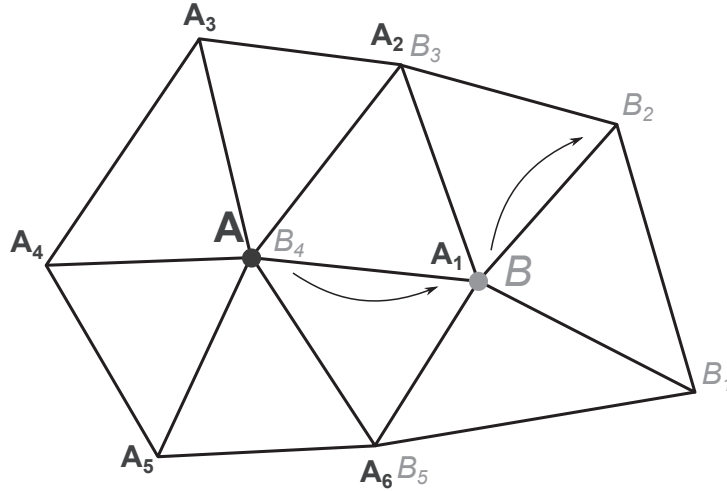


FIGURE 1.3: Illustration des définitions. Les sommets A_i (resp. B_i) sont les membres du 1-voisinage de A (resp. B). On a donc que la valence de A est de 6, il s'agit d'un sommet régulier. B n'a que 5 voisins, il est donc dit irrégulier. On note que le sommet B_2 , par exemple, appartient au 2-voisinage de A , car relié à A par au moins 2 arêtes.

Exemple

Voici un exemple de représentation permettant de stocker les informations d'un maillage polygonal volumique. Les positions des sommets sont donnés par l'ensemble P , les faces par l'ensemble F et les cellules par C , décrits ci-dessous :

$$P = (x_1, y_1, z_1, \dots, x_N, y_N, z_N) \quad (1.1)$$

$$F_P = (p_1, p_4, p_5), \dots, (p_3, p_7, p_9, p_2) \quad (1.2)$$

$$F_C = (c_3, c_1, c_7, c_8), \dots, (c_2, c_4, c_3) \quad (1.3)$$

On note ici que les coordonnées sont fournies de façon *entrelacées*, que la connectivité faces-sommets est descendante nodale, tandis que la connectivité faces-cellules est ascendante. Les arêtes ne sont *a priori* pas présentes, sauf si le tableau F_P est conçu de telle sorte que les n -uplets soient ordonnées, le triplet (4,8,9) indiquant les trois arêtes (4,8), (8,9) et (9,4).

Attributs

Ces informations sont suffisantes pour décrire les propriétés élémentaires du maillage, mais il est parfois utile de stocker d'autres données sur les simplexes. Dans le cas du rendu, par exemple, on pourra stocker les normales des sommets, ou les couleurs des faces. Si l'on souhaite appliquer une texture, on stockera les coordonnées (u, v) de paramétrisation en chaque sommet.

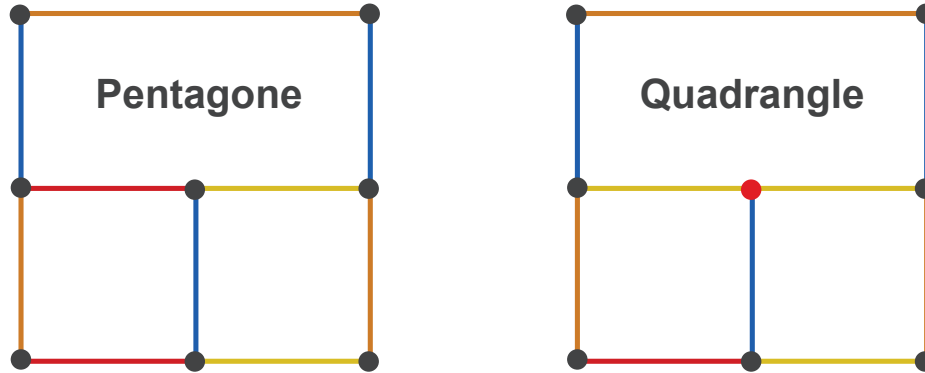


FIGURE 1.4: Illustration de la conformité d'un maillage. À gauche, le polygone du haut est un pentagone, puisque constitué de 5 sommets, mais à droite, le sommet central (rouge) ne fait pas partie du polygone, puisqu'il ne repose pas sur l'arête centrale (jaune). Ce sommet crée une non-conformité dans le maillage.

Tous ces attributs vont se présenter sous forme de tableaux de données, et viennent enrichir le maillage. Il est parfois utile de se baser sur ces attributs pour des processus de filtrage. Par exemple, un utilisateur peut avoir *colorié* le maillage en vue de distinguer des ensembles sur les lesquels il souhaite voir certaines opérations effectuées (raffinement) sans affecter le reste du maillage.

Vocabulaire

- **Voisinage** : le k -voisinage d'un sommet v_0 est défini comme l'ensemble v_1, \dots, v_n de points tels que $\forall i \in [1; n]$ v_0, v_i sont reliés par au moins k arêtes.
- **Valence** : le nombre de 1-voisins d'un sommet.
- **Élément de bord** : un élément de bord est un élément ont le voisinage est homéomorphe à une demi-boule.
- **Sommet régulier ou extraordinaire** : un sommet à l'intérieur d'un maillage est dit *regulier* lorsqu'il a une valence de 6 pour un maillage triangulaire, 4 pour un maillage quadrangulaire (4 et 3 respectivement pour un sommet de bord). Tous les autres sommets sont dits *extraordinaires*.
- **Maillage régulier** : un maillage est dit *regulier* lorsque tous les sommets de ce maillage sont réguliers. Un maillage est *semi-régulier* si la majorité de ses sommets sont réguliers.
- **Maillage isotrope ou anisotrope** : un maillage est dit *isotrope* si la forme des polygones est uniforme sur le maillage. Au contraire, un maillage dont la distribution de la forme des faces est variable dans l'espace est dit *anisotrope*.
- **Maillage structuré** : un maillage est dit *structuré* si tous les éléments ont la même valence (hors éléments de bord).

- **Maillage conforme** : un maillage est *conforme* si aucun de ses sommets ne se trouve au milieu d’une arête. La figure 1.4 illustre cette notion.

Représentations alternatives

Si les structures topologiques présentées ci-dessus sont les plus couramment utilisées, d’autres représentations peuvent s’avérer extrêmement utiles pour appliquer efficacement certains algorithmes. C’est le cas par exemple des *demi-arêtes*, introduites par Weiler dans [Wei88]. Chaque demi-arête est orientée, et pointe sur exactement un sommet (celui terminant la demi-arête), une arête, et une face (à gauche de la demi-arête), comme décrit sur la figure 1.5. Cette structure est particulièrement efficace lorsqu’il s’agit d’obtenir le voisinage d’un sommet, qui n’est pas directement accessible en utilisant la structure d’index classique. Une limitation de cette représentation est sa restriction aux surfaces, puisque la face « à gauche » n’est plus définie lorsque la face est intérieur à un volume.

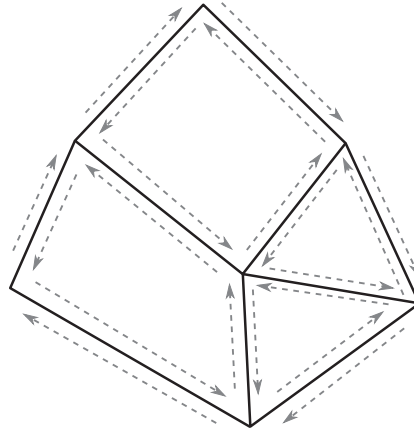


FIGURE 1.5: Exemple d’une structure de demi-arêtes.

Il existe également des représentations de la géométrie permettant d’optimiser certaines opérations. Dans [Sor05], Sorkine présente un nouveau moyen de représenter les sommets, les *coordonnées différentielles*. Pour un sommet $\mathbf{v}_i(x_i, y_i, z_i)$, les coordonnées différentielles sont données par la différence entre les coordonnées de ce sommet et celles du centre de gravité de son 1-voisinage

$$\delta_i = \mathbf{v}_i - \frac{1}{d_i} \sum_{j \in N(i)} \mathbf{v}_j \quad (1.4)$$

Cette représentation s’avère particulièrement efficace pour des algorithmes de type déformation variationnelle linéaire.

Obtention

Il existe plusieurs familles de processus permettant de générer des maillages, et l'on trouve parmi elles des méthodes basées capture, permettant de générer une version numérique d'éléments tangibles, et les méthodes de type Conception Assistée par Ordinateur (CAO), autorisant la création d'objets purement virtuels. Dans le cadre d'une acquisition laser, par exemple, la sortie de l'acquisition est un nuage de points, possédant chacun un attribut de position (auquel viennent éventuellement s'ajouter un attribut de normale et/ou de couleur). L'étape suivante consiste à créer une connectivité sommets-arêtes, ou sommets-faces, à l'aide d'une triangulation de Delaunay, par exemple. Ces méthodes ont le désavantage de se baser sur des mesures physiques, bruitées par nature, et un traitement géométrique du maillage sera très souvent nécessaire pour obtenir rendu des objets de qualité.

Les logiciels de CAO constituent l'autre grande famille des procédés de génération, permettant par exemple à l'utilisateur de créer la géométrie à l'aide d'une grammaire de courbes, surfaces et de relations logiques (intersection, union,...). Puis, à l'aide d'une paramétrisation, la composante topologique du maillage est créée en accord avec la géométrie.

Parmi les autres méthodes de génération numériques de géométries, il nous semble intéressant de citer le cas des surfaces de subdivision. Dans ce cas bien particulier, la géométrie est définie comme la limite d'une suite de raffinements lisses. Une présentation complète de cette technique pourra être trouvée dans [ZS00b]. Citons également le cas des splines, permettant la génération de surfaces complexes à partir d'un réseau clairsemé (plus particulièrement les NURBS, une généralisation des B-splines).

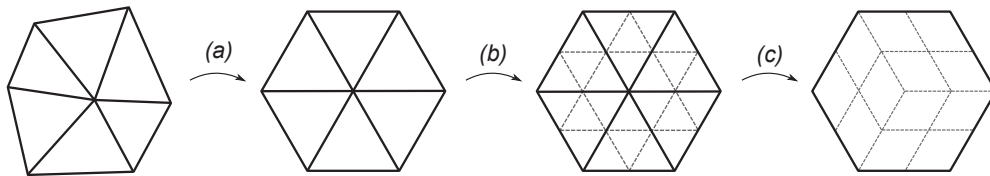


FIGURE 1.6: Exemple de manipulations de maillage. Ici, l'étape (a) est un lissage : la géométrie est modifiée, mais pas la topologie. L'étape (b) décrit un raffinement : les arêtes existantes sont conservées, d'autres sont ajoutées afin d'augmenter la densité de points du maillage. La troisième étape (c) présente un remaillage local : l'enveloppe du maillage est conservée, mais les arêtes et sommets internes sont complètement nouveaux.

1.3.2 Méthodes de traitement

Plusieurs types d'opérations peuvent être réalisées sur un maillage, selon les besoins de l'utilisateur.

Filtrage/Lissage

Une opération de filtrage, ou de lissage d'un maillage consiste à modifier la géométrie de celui-ci sans en altérer la topologie. Ainsi, dans le cadre d'un maillage indexé, seuls les attributs de position et/ou de normale verront leurs valeurs affectées, les ensembles topologiques conservant leurs propriétés.

On parle en général de filtrage lorsque le but de l'opération est la suppression de bruit ou d'artefacts, tandis que le terme de lissage s'applique à la régularisation d'une surface.

Raffinement/Simplification

On peut souhaiter, pour des besoins de précision par exemple, augmenter la résolution de certaines zones du maillage, en ajoutant des points sur la surface, et en modifiant la topologie de manière adéquate. Ce processus est appelé raffinement. L'opération inverse, dite simplification, a pour but d'alléger la taille du maillage, en simplifiant la topologie dans des zones où la géométrie varie très peu.

Remaillage

Cette dernière famille d'algorithmes regroupe les modifications topologiques qui ne sont ni des raffinements, ni des simplifications. Le but n'est pas de modifier la précision du maillage, mais d'optimiser la topologie de celui-ci, par exemple en essayant de régulariser le nombre de sommet de chaque face, le nombre de voisins de chaque sommet.

1.3.3 Objectifs

Les objectifs de ces opérations sont très divers, et sont liés à l'utilisation finale du maillage. Par exemple, un modèle qui serait utilisé dans l'animation aurait besoin d'avoir une géométrie très lisse, pour un rendu sans artefacts à l'écran, et une topologie raffinée, pour pouvoir être déformé afin de donner du mouvement à l'objet. A l'inverse, un maillage créé par CAO pour une simulation physique doit conserver une géométrie parfaitement définie, et être raffiné à l'endroit où les calculs seront les plus critiques. Par exemple, aux abords de la surface dans un calcul de mécanique des fluides, ou, plus généralement, lorsque de forts gradients des variables simulées sont attendus.

1.4 Le calcul parallèle

1.4.1 Introduction



FIGURE 1.7: Exemple de supercalculateur utilisé dans les calculs haute performance (HPC). Ici, le modèle *Blue Gene/P* d'IBM. *source : Argonne National Laboratory*

Les algorithmes de calculs ont tout d'abord été élaborés en vue d'une application séquentielle. Exécuté sur un ordinateur possédant un seul cœur, le problème est découpé en une série d'instructions discrètes, qui seront exécutées l'une après l'autre, sans chevauchement temporel.

Dans son approche la plus simple, le calcul parallèle implique seulement l'utilisation simultanée de ressources pour résoudre le problème. Celui-ci est d'abord découpé en parties indépendantes, pouvant être résolues de façon indépendante, puis chacune de ces parties est exécutée sur un cœur comme un problème séquentiel.

La parallélisation peut se faire sur une machine possédant plusieurs cœurs, ou bien sur plusieurs machines en réseau, ou bien encore grâce à une combinaison de ces deux méthodes. Le problème doit pouvoir être découpé en parties indépendantes et pouvant être exécutée simultanément. De fait, tous les algorithmes ne peuvent être parallélisés.

Le calcul parallèle permet de résoudre plusieurs défis. Le gain de temps, bien sûr, mais également la possibilité de traiter des problèmes de dimensions supérieures,

très gourmands en ressources. La parallélisation permet également de faire collaborer à travers le monde de nombreux ordinateurs sur le même projet.

1.4.2 Définitions

On peut classer les systèmes de calcul en quatre grands types, selon la taxinomie de Flynn [Fly72], présentant les architectures selon le nombre d'instructions (une ou plusieurs) et le nombre de mémoires (une ou plusieurs).

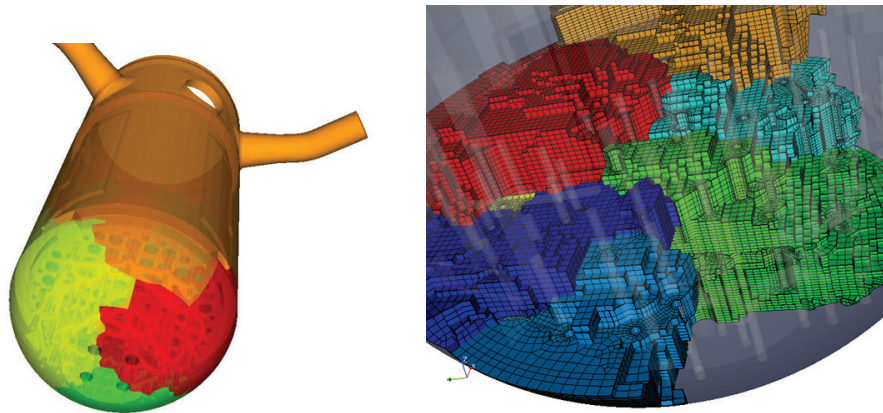


FIGURE 1.8: Exemple de maillage partitionné. *source : EDF R&D.*

- **SISD** : *Single Instruction, Single Data*. C'est l'architecture séquentielle simple, un seul processeur exécutant les tâches à partir d'une seule mémoire.
- **SIMD** : *Single Instruction, Multiple Data*. Tous les processeurs effectuent exactement la même opération, mais sur des zones de mémoire différente.
- **MISD** : *Multiple Instructions, Single Data*. Dans ce cas, une même zone de mémoire est utilisée par plusieurs processeurs, qui effectuent des opérations différentes.
- **MIMD** : *Multiple Instructions, Multiple Data*. Cette configuration est la plus courante dans le cadre du calcul parallèle. La mémoire peut être partagée, c'est-à-dire que chacune des unités de calcul a accès aux mêmes zones mémoire, ou bien *distribuée*, autrement dit chaque processeur n'a accès qu'à sa zone de mémoire propre, et la synchronisation et la communication doivent être prise en charge.

1.4.3 Exemples

Prenons par exemple le problème du calcul des termes de la suite de Fibonacci. La définition de cette suite est la suivante : les deux premiers termes sont $u_0 = 1$ et $u_1 = 1$. Les termes suivants sont définis comme $u_n = u_{n-1} + u_{n-2}$. À partir de ces définitions, il apparaît que chaque nouveau terme de la série dépend des deux

précédents et, par conséquent, la seule solution pour obtenir le n -ième terme de la série est de calculer un à un les précédents. Ce problème ne peut pas être paralléliser, il est de type SISD.

Imaginons à présent le problème de la somme de deux vecteurs de \mathbb{R}^n , $\mathbf{w} = \mathbf{u} + \mathbf{v}$. La i -ème composante de \mathbf{w} est alors donnée par $w_i = u_i + v_i$, et est donc totalement indépendante des autres composantes de \mathbf{u} ou \mathbf{v} . Il est par conséquent possible de séparer le calcul de chaque composante sur un processeur différent, et l'architecture obtenue ici est SIMD. C'est le mode de fonctionnement principal des GPUs.

Le traitement du signal peut nous fournir un exemple d'architecture MISD, dans le cas où un même signal serait filtré avec des paramètres différents par les processeurs. Il est à noter que cet agencement est toutefois plutôt rare.

Un calcul de mécanique des fluides, en mémoire distribuée, représente un bon exemple de configuration MIMD. La mémoire associée à chaque processeur contient une partie du maillage, et chaque processeur effectuera différentes opérations, en fonction du nombre de sommets, faces ou cellules qu'il comporte, et qui ne sont pas nécessairement identiques à ceux de son voisin.

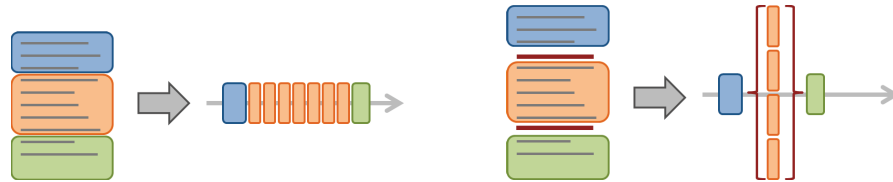


FIGURE 1.9: Principe de l'utilisation d'OpenMP. Nous prenons ici l'exemple d'un code, composé de trois parties, dont la seconde comporte beaucoup d'opérations identiques, pouvant être parallélisées. En haut, le code initial et son exécution. En bas, le code, avec l'insertion des balises OpenMP adéquates autour de la zone que l'on souhaite voir exécutée en multi-tâches, ainsi que son exécution.

1.4.4 Parallélisation avec OpenMP

Nous venons d'exposer les différentes sortes d'architectures parallèles offrant des possibilités d'accélération pour un programme. OpenMP [Bar07] est une API permettant d'optimiser un processus à l'origine SISD en le transformant en SIMD (mémoire partagée), en créant différentes tâches simultanément.

L'avantage de l'utilisation d'OpenMP est que celle-ci nécessite très peu de modification du code : il suffit en effet de placer des *balises* autour des zones que l'on souhaite voir s'exécuter en parallèle, et celles-ci seront exécutées comme des tâches parallèles, ainsi que l'illustre la figure 1.9.

Tous les algorithmes présentés dans cette thèse qui ne sont pas explicitement présentés comme parallèles ont vu leur implémentation accélérée à l'aide de cette API. En effet, il s'agit d'algorithmes dont les processus sont locaux, et chaque élément du maillage peut par conséquent être traité indépendamment des autres.

1.4.5 Parallélisation avec MPI

Si l'API OpenMP permet d'exécuter une partie des opérations d'un programme en mémoire partagée, la bibliothèque MPI [Mes12] (pour *Message Passing Interface*) permet quant à elle l'exécution en MIMD, dans le cadre d'une mémoire distribuée. L'intégration dans le code de calcul est bien plus complexe que dans le cadre d'OpenMP.

L'architecture d'un algorithme utilisant MPI peut être résumée de la sorte : pour chaque tâche, un processeur effectue les opérations suivantes

1. Réalisation des instructions
2. Échange de données
3. Synchronisation

Les algorithmes qui seront présentés dans la suite de cette thèse comme étant implémentés en parallèle ont été réalisés à l'aide de cette librairie.

1.4.6 Déterminisme

Un aspect extrêmement important lors de l'exécution de codes de calcul dans une architecture parallèle est la notion de *déterminisme*, c'est-à-dire que le comportement du système doit être parfaitement indépendant du nombre de processeurs qui composent l'architecture, ou de l'ordre de parcours du maillage lors d'un processus. En effet, les étapes de validation et de debug sont considérablement facilitées et accélérées par ce déterminisme. Par conséquent, certains outils se basant, par exemple, sur une numérotation des éléments ne peuvent être utilisés dans les méthodes que l'on souhaite mettre en œuvre en vue d'obtenir des résultats précis (aux erreurs de troncatures près).

1.5 Code_Saturne

Les travaux effectués dans cette thèse ont eu pour objet de fournir au logiciel libre de mécanique des fluides numériques *Code_Saturne* un module de traitement de maillage. Il est composé d'environ 350 000 lignes de code, dont 45% en C, 35% en

Fortran et 13% en Python, et son architecture peut être décrite par la figure 1.10. Ce logiciel est développé depuis 2000 au sein d'EDF R& D, et fournit de nombreux outils pour la modélisation de phénomènes physiques : flux laminaires, turbulents, transferts radiatifs, combustion,... Il est portable (Unix, Linux, MacOS X), équipé d'une interface graphique, peut être couplé avec d'autres codes EDF, et supporte de nombreux maillages créés par des logiciels externes.

Au sein du logiciel, les maillages sont non-structurés, conformes, et la connectivité est celle présentée dans l'exemple de ce chapitre (face-sommet descendante nodale, face-cellule ascendante). L'architecture est MIMD, la communication entre processeurs étant réalisée à l'aide de la librairie MPI, et le découpage du maillage à l'aide de bibliothèques comme Metis ou Scotch.

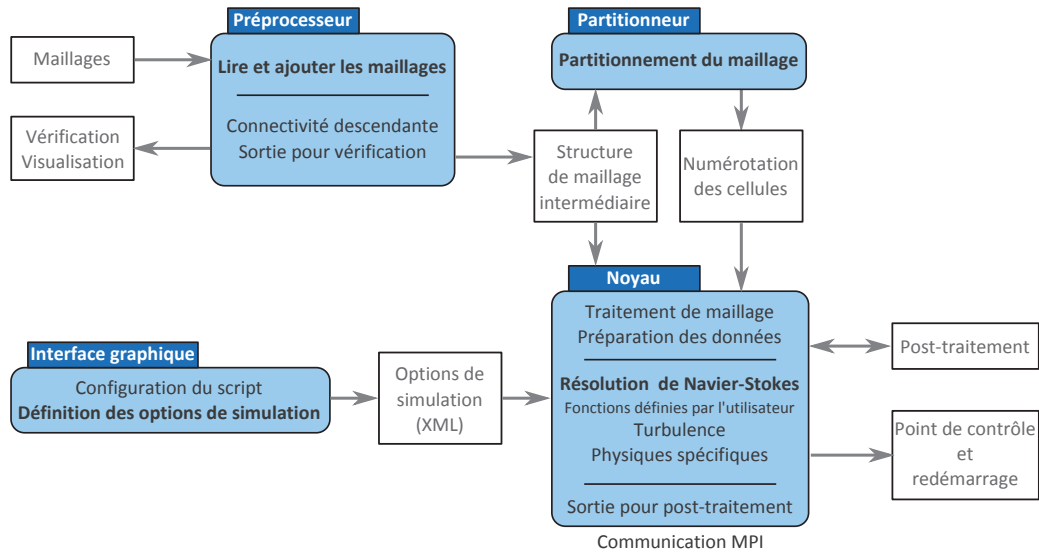


FIGURE 1.10: Principe de fonctionnement du logiciel *Code_Saturne*.

1.6 Contributions de la thèse

1.6.1 Vue d'ensemble

Le chapitre 2 met en avant deux algorithmes de filtrage et de lissage de maillage, et la façon dont nous avons dû les adapter au calcul parallèle distribué. Au travers de ce chapitre, nous souhaitons mettre en évidence les difficultés de ce procédé d'adaptation, qui devront être gardées à l'esprit lors de l'étude des algorithmes suivants.

Dans le chapitre 3, nous présentons les techniques que nous avons développé en matière de filtrage surfacique de maillages : l'accélération du filtrage bilatéral, l'adaptation des *histogrammes locaux* aux maillages, l'introduction de la médiane géométrique.

Nous développons dans le chapitre 4 l'extension de certaines de ces techniques aux maillages volumiques, et plus précisément aux maillages hexaédriques.

Le chapitre 5 est quant à lui consacré au remaillage local de la surface d'un maillage volumique, en vue de l'utilisation dans une simulation de mécanique des fluides numérique (MFN).

Le dernier chapitre traite de la génération de maillage à partir d'une simple forme, en se basant sur des interactions particulières, afin de générer des surfaces de quadrangles.

1.6.2 Publications

SBL : A Separable Approximation of Bilateral Mesh Filtering, *Guillaume Vialaneix et Tamy Boubekour*, SIGGRAPH 2011, Vancouver - Talk Program

SBL : A Separable Approximation of Bilateral Mesh Filtering, *Guillaume Vialaneix et Tamy Boubekour*, VMV 2011, Berlin

Hybrid Viscous Layer Insertion in a Tetrahedral Mesh, *Guillaume Vialaneix, Yvan Fournier et Tamy Boubekour*, IMR 2012, San José - Research Note Program

Chapitre 2

Vers la parallélisation

Nous venons de présenter ce qu'est un maillage, les différents types d'algorithmes que l'on peut appliquer à celui-ci, et les résultats que l'on peut escompter de ces opérations. Nous avons également passé rapidement en revue ce qu'est un algorithme parallèle, ainsi que les objectifs de cette parallélisation. Nous allons dans le présent chapitre, poser les bases de la réflexion suivante : quelles différences peuvent exister entre un algorithme séquentiel et sa version parallèle dans le cadre du traitement de maillages ? Autrement dit, si un algorithme n'est pas trivialement parallélisable, quels ajustements pourraient permettre de changer cet état, et quel serait l'impact de ces ajustements sur le fonctionnement de la méthode.

Nous allons baser notre approche de ces questions sur deux algorithmes. Le premier, un filtrage local explicité plus avant, cherche à optimiser la position des sommets du maillage en fonction du voisinage proche. Le second, basé sur le conditionnement de la matrice jacobienne, tente de créer, par le mouvement des sommets sur la surface du maillage, les polygones les plus réguliers possible.

2.1 Voisinage distribué

2.1.1 Problématique

Nous considérons ici le cas d'un maillage volumique distribué en mémoire sur différents processeurs, ainsi qu'illustré sur la figure 2.1. Nous considérons le partage suivant : chaque cellule ne peut appartenir qu'à un seul processeur, tandis que les faces et sommets peuvent être partagés entre plusieurs unités de calcul.

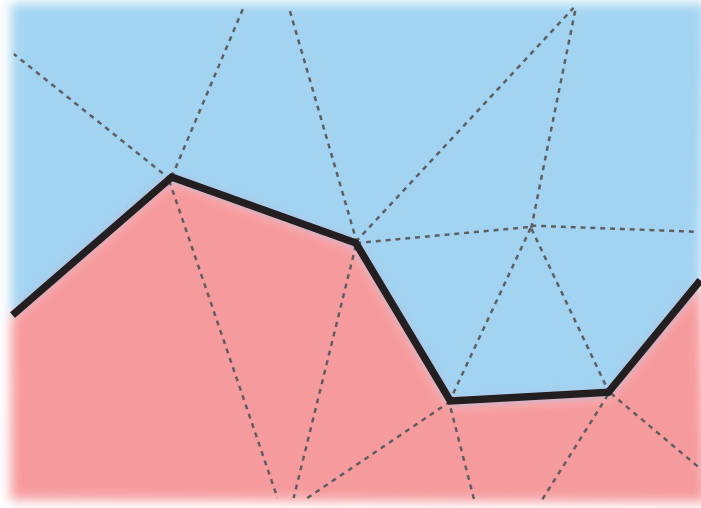


FIGURE 2.1: Exemple d'une zone de maillage dans laquelle deux processeurs possèdent des faces différentes. Ici, les faces et sommets de la partie supérieure (en bleu) appartiennent à un processeur, ceux de la partie inférieure (en rouge) appartiennent à un second processeur. Les sommets se trouvant sur la ligne forte entre les deux zones appartiennent aux deux processeurs.

2.1.2 Moyennes locales

Nous verrons plus en détail (notamment chap. 3) que de nombreux filtres sont basés sur des moyennes locales. Pour chaque sommet \mathbf{p}_i du maillage, un estimateur $\hat{\mathbf{p}}_i$, obtenu par une moyenne, parfois pondérée, de ses voisins, est calculé. Ensuite, chaque sommet est déplacé vers cet estimateur (lissage) ou dans la direction opposée (rehaussement). Nous allons dans la suite utiliser la formulation générale

$$\hat{\mathbf{p}}_i = \sum_{j \in N_i} w_j \mathbf{p}_j \quad (2.1)$$

$$\mathbf{p}_i \leftarrow \mathbf{p}_i + \alpha \hat{\mathbf{p}}_i \quad (2.2)$$

où α est un réel compris dans $\in [0;1]$, et w_j désigne le poids du voisin j , indépendamment de la méthode de calcul de ce poids. Si l'implémentation mono-processeur de cet algorithme est relativement simple, son adaptation en calcul parallèle distribué nécessite quelques précautions.

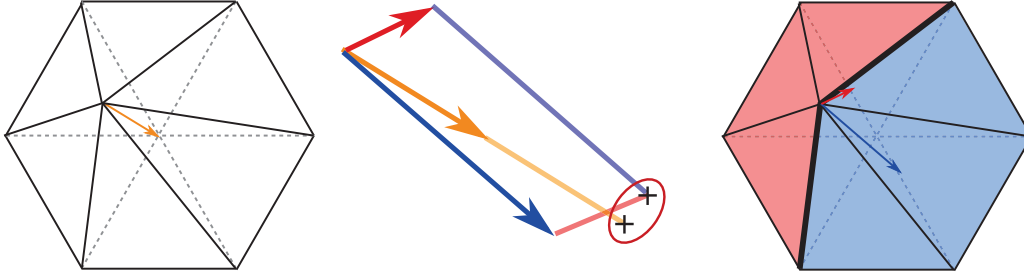


FIGURE 2.2: Illustration du problème posé par la distribution des données pour un filtrage laplacien. A gauche, le filtre appliqué en mono-processeur. A droite, le même filtre appliqué en mémoire distribuée. Au centre, un gros plan sur les vecteurs de déplacement de chaque mode : orange pour le filtrage mono-processeur, rouge et bleu pour chacune des projection en mémoire distribuée. On note que la somme directe des deux projections ne correspond pas à la projection obtenue en cas de calcul en mémoire partagée.

2.1.3 Adaptation parallèle

Transposition simple

Prenons l'exemple d'un maillage partagé entre deux processeurs, les points situés à la frontière étant communs aux deux processeurs (afin de pouvoir définir toutes les faces et cellules composées de ces points). Soient A et B les ensembles des points appartenant aux deux processeurs. La somme précédente peut s'écrire

$$\hat{\mathbf{p}}_j = \sum_{A \setminus B} w_i \mathbf{p}_i + \sum_{B \setminus A} w_i \mathbf{p}_i + \sum_{A \cap B} w_i \mathbf{p}_i \quad (2.3)$$

Si l'on applique basiquement un filtrage Laplacien sur chacun des processeurs séparément, on obtient :

$$\hat{\mathbf{p}}_j = \sum_A w_i \mathbf{p}_i + \sum_B w_i \mathbf{p}_i \quad (2.4)$$

$$\hat{\mathbf{p}}_j = \sum_{A \setminus B} w_i \mathbf{p}_i + \sum_{A \cap B} w_i \mathbf{p}_i + \sum_{B \setminus A} w_i \mathbf{p}_i + \sum_{B \cap A} w_i \mathbf{p}_i \quad (2.5)$$

$$\hat{\mathbf{p}}_j = \sum_{A \setminus B} w_i \mathbf{p}_i + \sum_{B \setminus A} w_i \mathbf{p}_i + 2 \sum_{A \cap B} w_i \mathbf{p}_i \quad (2.6)$$

On remarque que les sommets situés à l'interface sont comptabilisés deux fois. La figure 2.2 illustre ce phénomène. Une autre difficulté peut provenir de la définition même du 1-voisinage. En effet, il pourrait arriver que deux sommets, appartenant chacun à deux processeurs, soient reliés par une arête définie uniquement sur l'un des deux. Alors, lors de la requête générant le 1-voisinage, ces deux points ne seraient pas systématiquement considérés comme voisin, selon le processeur qui générerait cet appel.

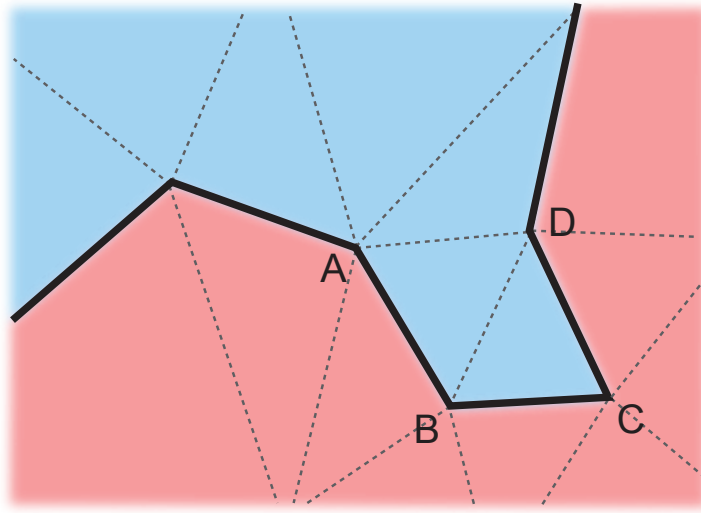


FIGURE 2.3: Illustration du problème posé par la distribution des données pour une requête de voisinage. Ici, les sommets A et D appartiennent aux deux processeurs, mais ne sont considérés comme voisin que pour le processeur bleu. De fait, si la convention choisie pour la répartition du poids des sommets entraînait que le poids de D pour A (et inversement) soit calculé à partir du processeur rouge, ce poids serait nul, ce qui ne correspond pas au maillage. D'où la nécessité de construire avant toute opération une connectivité cohérente sur tous les processeurs.

Solutions

Les difficultés énoncées ci-dessus peuvent être facilement surmontées, à la condition de générer correctement la connectivité sommet-sommet, et de fixer une convention pour les sommets appartenant à de multiples processeurs. La première étape est donc la construction de la « vraie » connectivité sommet-sommet. Autrement dit : si deux sommets appartiennent à un processeur, et qu'il existe, dans la connectivité des autres processeurs possédant ces sommets, une arête les reliant, alors tous les processeurs possédant ces sommets doivent contenir cette arête dans leur connectivité.

La deuxième condition à remplir pour pouvoir appliquer correctement un filtrage est de fixer une convention de répartition des poids au sein des différents processeurs. Par exemple, décréter que le processeur commun aux sommets possédant le plus petit numéro portera le poids w_i , tandis que les autres processeurs apporteront une contribution nulle. En reprenant l'exemple précédent, on aura alors

$$\hat{\mathbf{p}}_j = \sum_{A \setminus B} w_i \mathbf{p}_i + \sum_{A \cap B} w_i \mathbf{p}_i + \sum_{B \setminus A} w_i \mathbf{p}_i + \sum_{B \cap A} w_i \mathbf{p}_i \quad (2.7)$$

$$\hat{\mathbf{p}}_j = \sum_{A \setminus B} w_i \mathbf{p}_i + \sum_{A \cap B} (w_{i,A} + w_{i,B}) \mathbf{p}_i + \sum_{B \setminus A} w_i \mathbf{p}_i \quad (2.8)$$

$$\hat{\mathbf{p}}_j = \sum_{A \setminus B} w_i \mathbf{p}_i + \sum_{A \cap B} (w_i + 0) \mathbf{p}_i + \sum_{B \setminus A} w_i \mathbf{p}_i \quad (2.9)$$

$$\hat{\mathbf{p}}_j = \sum_{A \setminus B} w_i \mathbf{p}_i + \sum_{A \cap B} w_i \mathbf{p}_i + \sum_{B \setminus A} w_i \mathbf{p}_i \quad (2.10)$$

Il suffit alors de constater que, par définition, $A \setminus B + B \setminus A + A \cap B = A \cup B$, et donc, grâce à cette méthode, chaque élément n'est comptabilisé qu'une et une seule fois.

Cette problématique n'est qu'un exemple parmi d'autres, illustrant les précautions nécessaires à la mise en place d'un algorithme en parallèle. Bien souvent, les opérations à réaliser ne sont pas extrêmement complexes, mais nécessitent une observation minutieuse

2.2 Approximation

2.2.1 Matrice jacobienne

Fonction de coût

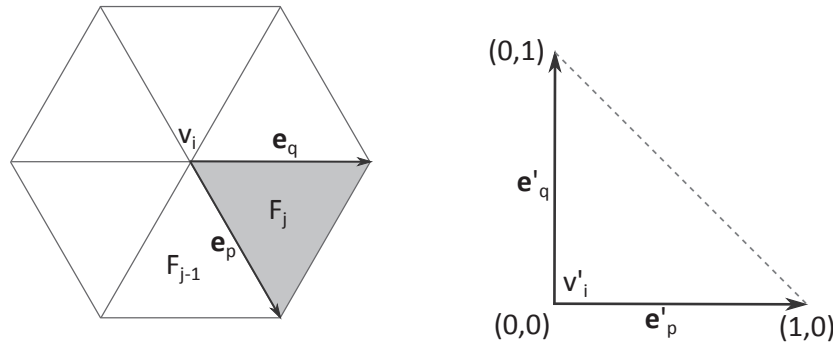


FIGURE 2.4: Définition de \mathbf{e}_p et \mathbf{e}_q

Dans cette approche, on considère l'application qui transforme un triangle quelconque en un triangle isocèle rectangle de côté 1 (cf fig. 2.4). On considère ainsi une face F_j dont V_i est le sommet transformé en sommet principal du nouveau triangle. La matrice jacobienne de cette application est donnée par $\mathbf{J}_{ij} = [\mathbf{e}_p \ \mathbf{e}_q]$, où \mathbf{e}_p et \mathbf{e}_q sont les vecteurs représentant les arêtes de F_j reliant V_i aux autres sommets.

Conditionnement

Le conditionnement d'une matrice M de taille $m \times n$ (avec $m \leq n$) est donné par le rapport de sa plus grande et de sa plus petite valeur singulière (racines carrées des valeurs propres de MM^T).

On peut alors écrire le conditionnement de la matrice jacobienne comme suit :

$$\kappa(\mathbf{J}_{ij}) = \frac{l_p^2 + l_q^2}{2A_j} \quad (2.11)$$

avec l_p et l_q les longueurs respectives de \mathbf{e}_p et \mathbf{e}_q , et A_j l'aire du triangle formé par \mathbf{e}_p et \mathbf{e}_q . On va chercher à minimiser la fonctionnelle définie par

$$\psi_i^c(\mathbf{x}_i) = \sum_j \kappa(\mathbf{J}_{ij}(\mathbf{x}_i)) = \sum_j \frac{l_p^2(\mathbf{x}_i) + l_q^2(\mathbf{x}_i)}{A_j(\mathbf{x}_i)}, \quad j \in \{j | F_j \in F(V_i)\} \quad (2.12)$$

avec \mathbf{x}_i la position de V_i . L'idéal serait de pouvoir minimiser directement la somme des ψ_i^c , mais cette opération n'est pas réalisable en pratique. Par conséquent, la stratégie consiste à chercher à minimiser en chaque sommet la fonctionnelle

$$\widetilde{\psi}_i^c = \sum_j \sum_k \kappa(\mathbf{J}_{jk}), \quad j \in \{j | F_j \in F(V_i)\}, k \in \{k | V_k \in V(F_j) \cap V(E(V_i))\} \quad (2.13)$$

autrement dit la somme des conditionnements calculés pour chacun des sommets représentés par un cercle plein sur la figure 2.4. De ce fait, la minimisation effectuée reste locale, conformément aux exigences du problème.

La matrice jacobienne considérée n'est pas celle qui transforme en un triangle équilatéral, mais isocèle rectangle, ce qui peut sembler déroutant puisque l'on cherche à obtenir des triangles équilatéraux. Cependant, il ne faut pas perdre de vue deux choses : tout d'abord, un triangle équilatéral minimise la fonction définie ci-dessus lorsqu'elle est appliquée en chacun de ses sommets. Ensuite, lorsque les faces ne sont pas des triangles, mais des polygones d'ordre supérieur, on cherche à obtenir des polygones réguliers, donc avec des angles de 90° pour des quadrangles, 108° pour des pentagones, etc. Or, de tels polygones ne pourraient pas être obtenus en minimisant une fonction cherchant à rendre des triangles équilatéraux (les quadrangles seraient sans doute transformés en losanges, par exemple).

Procédure de minimisation

Pour s'assurer de ne pas trop modifier la surface définie par le maillage, la procédure consiste à faire bouger les sommets sur les faces initiales. Pour cela, un sommet ne sera plus repéré par ses coordonnées globales, mais par deux *coordonnées locales* (ou plus si les faces ont plus de 4 arêtes) et la face à laquelle il appartient. La

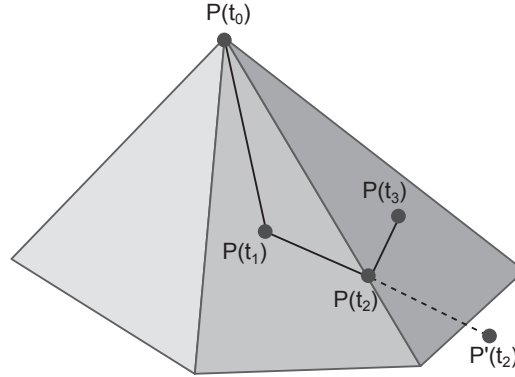


FIGURE 2.5: Procédure de minimisation de l'algorithme de conditionnement

première opération est donc de repérer chaque sommet avec ce nouveau système. Les coordonnées utilisées dans notre implémentation sont les *coordonnées isobarycentriques* [GS04]. Considérant un triangle $V_0V_1V_2$ d'aire $A = \text{aire}(V_0V_1V_2)$, et M un point intérieur au triangle, nous définissons les trois aires $A_0 = \text{aire}(MV_1V_2)$, $A_1 = \text{aire}(V_0MV_2)$ et $A_2 = \text{aire}(V_0V_1M)$. En notant \mathbf{x}_i les coordonnées du point I , nous avons alors

$$s_i = \frac{A_i}{A}, \quad i = \{1, 2, 3\} \quad (2.14)$$

$$s_0 + s_1 + s_2 = 1 \quad (2.15)$$

$$\mathbf{x}_M = s_0\mathbf{x}_0 + s_1\mathbf{x}_1 + s_2\mathbf{x}_2 \quad (2.16)$$

Une fois que les sommets sont repérés, on calcule le gradient de $\widetilde{\psi}_i^c$ dans le voisinage du point dans le nouveau repère. On détecte la direction de plus grande pente. On va alors effectuer une descente de gradient, en veillant à conserver la structure du maillage : si le point est amené à quitter la face pour progresser, alors la procédure se termine sur l'arête, et reprendra à la passe suivante.

La figure 2.5 illustre cette procédure : au départ de l'algorithme, le point P est encore à sa position d'origine $P(t_0)$. Après une première passe, le point a migré sur la face qui lui permettait de diminuer au maximum sa fonction de coût, et il se trouve alors en $P(t_1)$. Lors de la deuxième passe, le point cherche à aller en $P'(t_2)$. Seulement, le point sortirait alors de la face dans laquelle il se trouve, et son appartenance à la surface définie par le maillage initial est alors compromise. Par conséquent, le point s'immobilise sur l'arête de la face en $P(t_2)$. Lors de la boucle suivante, le point reprend sa progression dans la nouvelle face, et va se placer en $P(t_3)$.

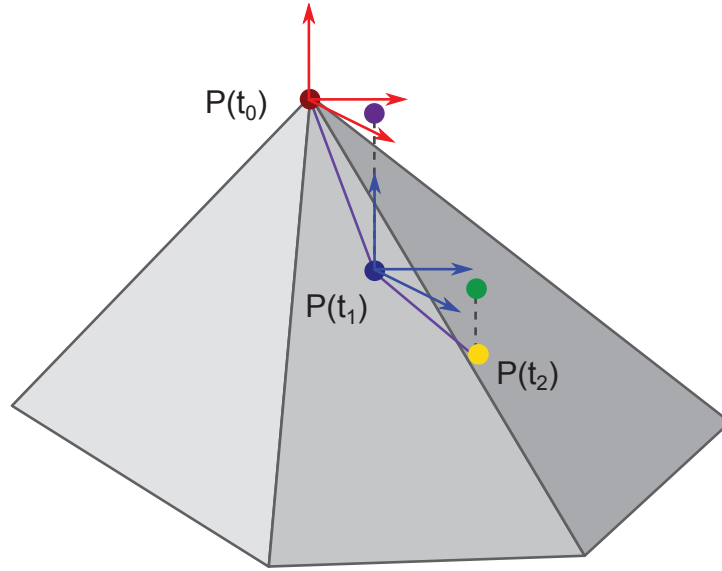


FIGURE 2.6: Approximation réalisée pour l'implémentation en mémoire distribuée. Le point initial $P(t_0)$ (rouge) sert d'origine pour un repère local dans lequel est calculée la nouvelle position optimale (violet), qui est alors projetée sur la surface (bleu) pour former le point suivant $P(t_1)$. Puis le processus se poursuit, avec une nouvelle valeur optimale (vert), suivie d'une projection (jaune), et ainsi de suite jusqu'à convergence.

Lorsque la somme des mouvements effectués par les points est inférieure à un seuil fixé au préalable, on estime que l'algorithme a convergé, et le processus s'arrête.

2.2.2 Adaptation parallèle

Cette procédure n'est pas applicable dans le cadre d'une simulation multi-processeur. En effet, pour chaque calcul de $\tilde{\psi}_i^c$ ou de son gradient, il est nécessaire de connaître au minimum tout le 1-voisinage de tous les points du maillage, ce qui n'est pas le cas immédiatement dans le cadre d'un maillage découpé entre plusieurs processeurs. Deux solutions pourraient permettre de contourner cette difficulté. La première nécessiterait de modifier la structure de données du maillage, afin de permettre la requête de 1-voisinage complets. Nous avons choisi une autre approche, consistant à trouver une approximation de la formulation précédente compatible avec les structures de données actuellement en place.

En chaque sommet, le gradient de la fonction définie ci-dessus est calculé dans les 3 directions globales, puis sommé sur les différents processeurs. Le point est

alors très légèrement déplacé ($\frac{1}{20}$ de la plus petite arête connectée au sommet considéré). Puisque nous ne disposons plus des coordonnées locales, le déplacement est alors projeté sur le plan tangent à la surface au point considéré. Le processus est répété jusqu'à convergence (ou atteinte du nombre limite de boucles, fixé au préalable).

Les deux différences essentielles avec l'algorithme mono-processeur sont le remplacement des coordonnées locales par la projection du déplacement sur le plan tangent à la surface, et la substitution de la procédure de recherche linéaire par un déplacement pas à pas des points, qui rappelle le mode de fonctionnement de nombreux solveurs de physique.

2.2.3 Résultats

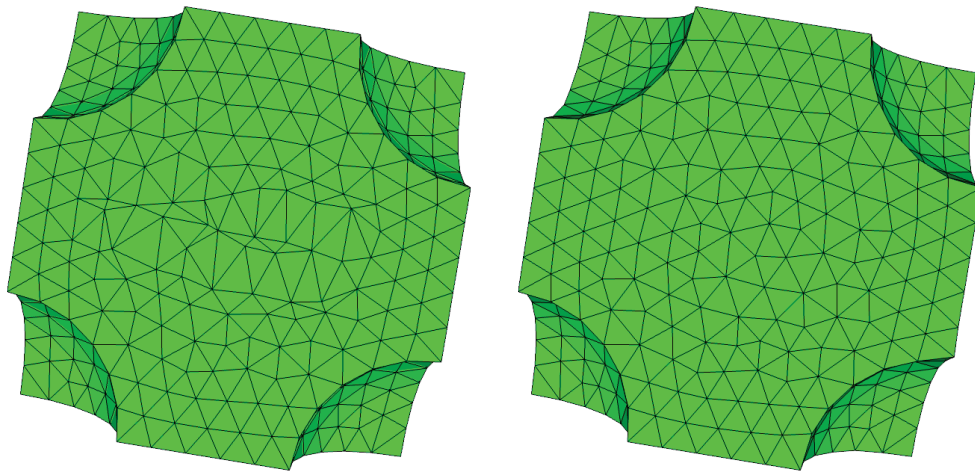


FIGURE 2.7: Résultat de l'application du filtre sur un maillage surfacique. À gauche, le modèle initial, dont la position des sommets n'est pas optimisée. À droite, le résultat à la sortie du lisseur. On remarque la disparition de l'anisotropie initiale.

La figure 2.7 présente le type de résultats obtenus avec cet algorithme. Comme attendu, les zones isotropes sont conservées en l'état, tandis que les sommets répartis de façon moins optimale sont déplacés sur la surface du maillage.

2.3 Discussion

Nous venons, au travers de ces deux exemples, de développer quelques éléments permettant de mieux appréhender ce qui fait la spécificité des algorithmes paral-

lèles à mémoire distribuée : la nécessité de disposer d'une structure de données adaptée, et l'utilité de la re-formulation d'une technique, parfois au détriment de son efficacité, afin de disposer d'une version parallèle.

Dans les chapitres qui vont suivre, nous nous sommes efforcés, si les méthodes n'ont pas été réalisées et testées à l'aide d'une architecture parallèle, de garder à l'esprit qu'elle devraient l'être, à terme, et donc que le cœur même des techniques se devait d'être parallélisable sans approximation.

Chapitre 3

Filtrage surfacique

3.1 Contexte

3.1.1 Bruit et filtres passe-bas

Lorsqu'un maillage est obtenu par la mesure physique d'une surface, à l'aide d'un scanner, par exemple, les composantes géométriques du maillage (positions, normales) sont entachées de bruit, l'intensité et la nature de celui-ci dépendant du mode d'acquisition. Il existe toutefois quelques caractéristiques communes à ces bruits, comme leur positionnement dans les hautes fréquences spatiales. De fait, une solution simple pour éliminer ce bruit est l'application d'un filtre *passe-bas*, c'est-à-dire diminuant fortement l'intensité des hautes fréquences en ne modifiant pas les intensités dans la partie basse du spectre.

Le résultat présenté dans la figure 3.1 illustre ce procédé en deux dimensions et amène deux observations. La première, c'est que le filtre est efficace dans sa tâche première, à savoir ôter le bruit présent dans l'objet. La seconde, c'est que si nous souhaitons que le filtre soit suffisamment puissant pour supprimer complètement le bruit, nous allons aussi supprimer un certain nombre de structures dans l'objet.

Sur une image, il s'agit des zones à fort gradient, délimitant les contours des objets, et le résultat de l'application d'un filtre passe-bas est le floutage de l'image. Sur un maillage, il s'agit des *arêtes vives*, et le résultat du filtrage donne une surface très lisse. Dans chacun de ces cas, ce résultat peut être positif : si l'on souhaite créer un effet artistique sur une image, ou si le maillage est destiné à être utilisé dans une scène non réaliste aux formes lisses (un film d'animation, par exemple). Cependant, si le filtrage n'est qu'une étape dans le processus de traitement de l'objet, et que

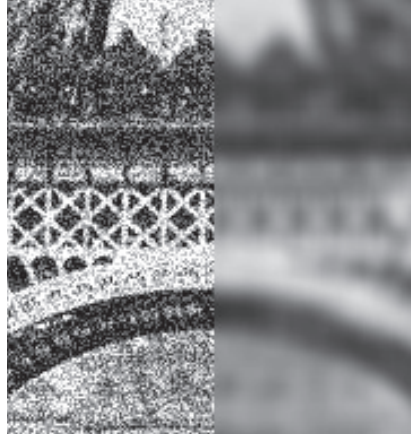


FIGURE 3.1: Gauche : image à laquelle a été ajouté un bruit blanc. Droite : le résultat d'un filtrage de l'image bruitée à l'aide d'un filtre passe-bas.

l'utilisateur a besoin de ces caractéristiques pour la suite du traitement, il est impératif de trouver un moyen d'altérer le filtrage dans ces zones.

3.1.2 Travaux précédents

Filtres passe-bas

Le travail déterminant de Taubin [Tau95] initia l'utilisation du traitement de signal pour les maillages, en introduisant le filtrage passe-bas pour le débruitage. En particulier, Taubin a montré le lien entre le filtrage passe-bas et le lissage de maillages. Il a introduit le Laplacien d'un signal surfacique discret comme une généralisation des descripteurs de Fourier, permettant des algorithmes de complexité linéaire pour le filtrage de maillage.

De nombreux filtrages passe-bas sont ainsi basés sur cet outil, leur principe étant de déplacer un sommet en fonction de la position d'un certain nombre de sommets dans son k -voisinage, cette interpolation pouvant être pondérée. L'approximation discrète la plus basique de l'opérateur de Laplace-Beltrami, aussi appelé Laplacien topologique, est donné par la formulation suivante

$$L(\mathbf{x}_i) = \frac{1}{n_i} \sum_{j \in N_i} \mathbf{x}_j - \mathbf{x}_i \quad (3.1)$$

où N_i est le 1-voisinage du sommet i et n_i désigne le cardinal de N_i . Une autre formulation intéressante est donnée dans [PP93], qui relie l'opérateur de Laplace-Beltrami à la courbure de la surface

$$L(\mathbf{x}_i) = \frac{1}{2A_i} \sum_{j \in N_i} (\cot \alpha_{ij} + \cot \beta_{ij})(\mathbf{x}_j - \mathbf{x}_i) \quad (3.2)$$

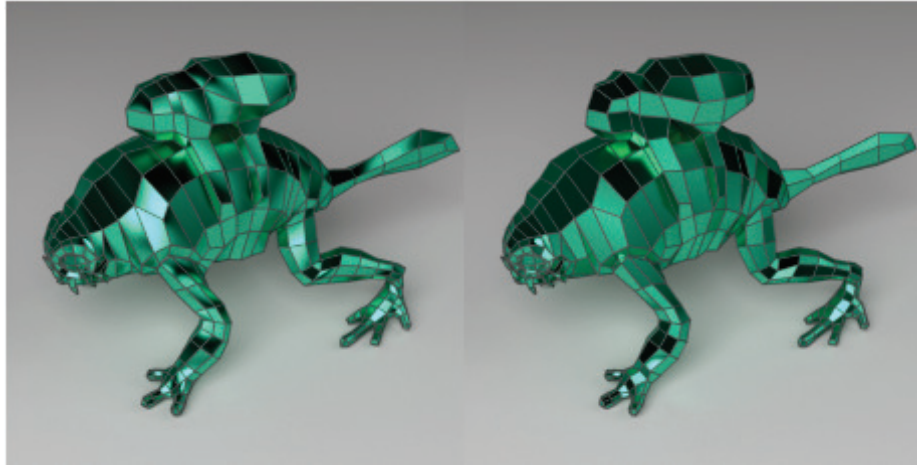


FIGURE 3.2: Gauche : maillage original. Droite : résultat de l'application du Laplacien présenté dans [AW11]. source : [AW11]

où A_i est l'aire de la cellule de Voronoi de \mathbf{x}_i , α_{ij} et β_{ij} étant les angles opposés de l'arête ij . On trouve une troisième écriture dans [Fuj95] :

$$L(x_i) = \frac{2}{E} \sum_{j \in N_i} \frac{\mathbf{v}_j - \mathbf{v}_i}{|e_{ij}|} \quad (3.3)$$

avec e_{ij} l'arête entre \mathbf{v}_j et \mathbf{v}_i , et $E = \sum_{j \in N_i} |e_{ij}|$. Cette troisième expression, l'*opérateur parapluie dépendant d'échelle*, est utilisé par Desbrun *et al.* [DMSB99a], couplé à une méthode d'Euler implicite pour intégrer le flux de courbure.

Quelques approximations et variantes ont aussi été proposées pour traiter des maillages quadrangulaires, dans [LXZ08], par exemple, où Liu *et al.* propose la discrétisation suivante : à partir de la dérivation de la normale à la courbure moyenne, et en définissant la surface à partir de quatre points (p_i, p_j, p_k, p_l) comme ceci

$$S(u, v) = (1 - u)(-v)p_i + v(1 - u)p_j + u(1 - v)p_k + uv p_l, (u, v) \in [0, 1]^2 \quad (3.4)$$

l'opérateur de Laplace-Beltrami peut être approximé par une combinaison linéaire de ces quatre points, dont les poids sont fonction de plusieurs intégrales sur l'espace (u, v) prenant en compte u, v, S et leurs normales et moyennes.

Une formulation de cet opérateur par Alexa et Wardetzky [AW11] (fig.3.2) permet de l'étendre aux maillages polygonaux. De même, dans [Pao02], Paoletti cherche à améliorer la qualité du maillage à l'aide du tenseur d'interpolation. Il utilise ce tenseur pour définir un critère de qualité pour n'importe quel type de polygone (la norme de Frobenius du tenseur), et déplace les arêtes localement pour minimiser ce critère, à l'aide de l'algorithme de Newton.



FIGURE 3.3: Gauche : maillage original. Droite : résultat de l'application de la stratégie présentée dans [PSZ01]. *source : [PSZ01]*

Desbrun *et al.* [DMSB99b] introduisirent un schéma d'intégration implicite pour calculer l'équation de diffusion sur un maillage. Ils proposèrent une nouvelle définition de l'opérateur Laplacien, comme un opérateur de filtrage pouvant être vu comme l'intégration temporelle de l'équation de la chaleur. Ceci leur a permis de définir le flux de courbure, basé uniquement sur les propriétés géométriques intrinsèques et offre une bonne alternative au flux de diffusion.

Donoho [Don95] présenta un traitement basé sur les ondelettes, calculant la transformée en ondelettes du maillage, et seuillant ses coefficients avant d'appliquer la transformation inverse. Peng *et al.* [PSZ01] ont adopté une stratégie similaire, mais utilisèrent un filtre de Wiener et les *Gaussian Scale Mixtures* (GSM) comme opérateurs de transformation.

Également inspiré du vaste répertoire de techniques de débruitage d'images, Yoshizawa *et al.* [YBS06] ont récemment adapté le filtre utilisant les moyennes non-locales (NLM) [BCM05] aux maillages, en cherchant parmi tous les sommets ceux possédant un voisinage similaire à un point donné à filtrer, leur donnant un poids important indépendamment de leur localisation.

Yagou et Ohtake [OBS02][YOB02][YOB03] ont proposé plusieurs techniques basées sur un schéma en deux temps, consistant à filtrer les normales des faces, puis modifier les positions des sommets en essayant de les rendre compatibles avec les nouvelles normales.

Pour une vue d'ensemble récente des méthodes de lissage de maillages polygonaux, le lecteur pourra se référer au livre de Botsch *et al.* [BKP⁺10]. Nous allons à

présent nous focaliser sur la formulation proposée par Jones et Durand [JDD03], car leur méthode est simple, et donne un attribut dans le domaine des caractéristiques à chaque centre de face du maillage, facilement manipulable.

Le filtre bilatéral



FIGURE 3.4: Gauche : image bruitée. Centre : résultat d'un filtrage gaussien. Droite : résultat du filtrage bilatéral.

Considérant un objet bruité (par exemple une image ou un maillage), le moyen le plus simple d'ôter ce bruit est, comme nous l'avons évoqué plus haut, d'appliquer un filtre passe-bas (par exemple Gaussien) qui remplace chaque élément (pixel ou sommet) par une moyenne pondérée de ses voisins. Pour chaque échantillon, un tel filtre ignore la valeur de ces éléments, et ne considère que leur éloignement spatial pour déterminer la contribution de chacun d'entre eux dans une combinaison locale. Par conséquent, le bruit est filtré, mais les caractéristiques de l'objet sont floutés proportionnellement.

L'idée du filtre bilatéral (BL) [TM98] est d'introduire un second terme de pondération, basé sur la similarité de valeur entre les échantillons (par exemple la couleur d'un pixel). Dans le cas où les noyaux utilisés sont Gaussiens, le filtre bilatéral peut alors s'interpréter comme un filtrage gaussien, mais dans un espace de dimension supérieure [PD09], constitué des informations spatiales mais aussi de valeur des éléments.

Il est nécessaire de noter qu'un espace de ressemblance basé sur la différence de valeur des pixels est l'espace de similarité le plus simple que l'on puisse envisager pour des images. Des comparaisons plus robustes ont été introduites depuis, comme celles basées sur le voisinage local autour des éléments dans le contexte des moyennes non-locales [BCM05]. Pour une introduction plus complète au filtrage bilatéral, on se référera au cours de Paris *et al.* [PKTD08].

Du point de vue du traitement du signal, la différence principale entre une image et un maillage est que, si la première possède un espace global de paramétrisation trivial (les coordonnées des pixels) totalement décorrélé de la valeur de ceux-ci (la couleur des pixels), ce n'est pas le cas du second, dans lequel les coordonnées spatiales du sommet contiennent en générale à la fois le signal et la paramétrisation, c'est-à-dire, les données spatiales et de valeur.

Plusieurs stratégies ont été proposées pour appliquer le filtrage bilatéral aux maillages [JDD03] [FDCO03] et aux surfaces discrètes [JDZ04]. Bien que notre travail ne soit pas restreint à cette formulation particulière, nous nous concentrons ici sur la méthode proposée par Jones et Durand [JDD03] qui définissent un espace de valeur pour les maillages au moyen de prédictions, calculées comme la projection du sommet cible sur les plans tangents de ses voisins, tel que décrit sur la figure 3.5.

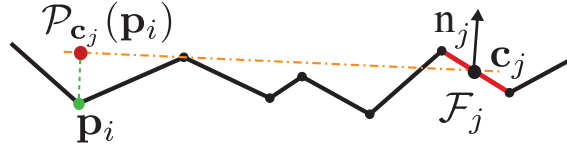


FIGURE 3.5: La projection $P_{c_j}(\mathbf{p}_i)$ d'un sommet \mathbf{p}_i en fonction d'une face voisine F_j . Le poids de F_j par rapport à \mathbf{p}_i est donné par la norme L^2 : $\|\mathbf{p}_i - P_{c_j}(\mathbf{p}_i)\|$.

Considérant un maillage M , et l'un de ses sommets \mathbf{p}_i , alors pour chacune des faces F_j de son voisinage N_i , il est possible de calculer la projection de \mathbf{p}_i sur le plan défini par le centre \mathbf{c}_j et la normale lissée de F_j , \mathbf{n}_j

$$P_{c_j}(\mathbf{p}_i) = \mathbf{p}_i + \mathbf{n}_j((\mathbf{c}_j - \mathbf{p}_i) \cdot \mathbf{n}_j) \quad (3.5)$$

En utilisant un noyau spatial G_{σ_s} et un noyau de valeur G_{σ_r} , un sommet \mathbf{p}_i est filtré à l'aide de l'ensemble de ses faces voisines N_i par

$$BL_{N_i}(\mathbf{p}_i) = \frac{1}{W(\mathbf{p})} \sum_{F_j \in N_i} a_j G_{\sigma_s}(d_{ij}) G_{\sigma_r}(h_{ij}) P_{c_j}(\mathbf{p}_i) \quad (3.6)$$

avec

$$W(\mathbf{p}) = \sum_{F_j \in N_i} a_j G_{\sigma_s}(d_{ij}) G_{\sigma_r}(h_{ij}) \quad (3.7)$$

a_j l'aire de la face j , $d_{ij} = \|\mathbf{p}_i - \mathbf{c}_j\|$ et $h_{ij} = \|\mathbf{p}_i - P_{c_j}(\mathbf{p}_i)\|$. Notons que la robustesse est améliorée par un lissage préalable des normales des faces en utilisant un noyau spatial $G_{\sigma_s/2}$ [JDD03]. L'ensemble N_i peut être collecté à l'aide d'un procédé de remplissage commençant avec \mathbf{p}_i , et sélectionnant toutes les faces situées dans un rayon σ_s . Dans notre implémentation, nous utilisons des noyaux Gaussiens

$$G_{\sigma}(x) = e^{-\frac{x^2}{2\sigma^2}} \quad (3.8)$$

Bien que local, le filtrage bilatéral sur les maillages demeure un processus lent, et différentes techniques ont été développées afin d'accélérer cet algorithme, présentées dans la section suivante.

3.2 Filtre bilatéral séparable

3.2.1 Accélération

Approches orthogonales

Chen *et al.* [CPD07] ont proposé une accélération en introduisant une structure nommée *grille bilatérale*. Partant du principe que, dans le cas où les noyaux utilisés pour le filtrage bilatéral sont des gaussiennes, le filtrage bilatéral peut s'exprimer lui-même comme un noyau gaussien, dans un espace de dimension supérieur, alors les pixels d'une image pourraient être stockés dans une grille dont les dimensions correspondent d'une part aux coordonnées spatiales des pixels, et d'autre part à leur intensité. Une telle structure permet ainsi, entre autres, d'accélérer le processus de filtrage bilatéral.

L'amélioration de cette technique proposée par Adams *et al.* [AGDL09] substitue un *kd-tree* à la grille et est extensible aux maillages. La construction de cet arbre permet une accélération des phases d'interpolation et d'échantillonnage de la grille bilatérale. Le principe de cet algorithme est décrit sur la figure 3.6.

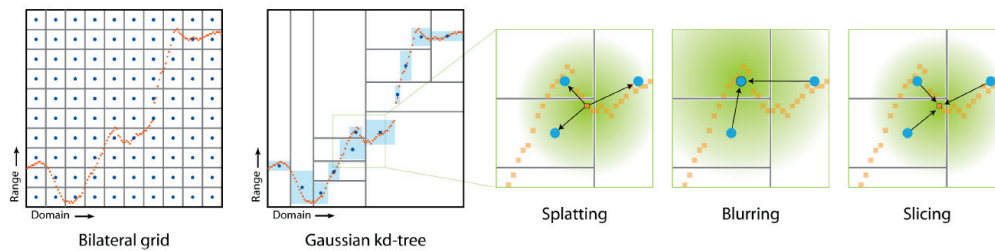


FIGURE 3.6: Illustration de la structure de *gaussian kd-tree*, issue de [AGDL09].

Séparabilité

L'observation de base de notre méthode est la séparabilité d'un noyau gaussien en dimension n . Ainsi, dans \mathbb{R}^2 , on a $G(x, y) = G(x) \cdot G(y)$, et ainsi, si l'on souhaite appliquer un filtrage gaussien à une image, l'idée première qui est de convoluer l'image avec un masque gaussien revient alors à appliquer ce filtrage séparément

dans chaque direction, d'où un gain de temps conséquent. C'est cette séparabilité que nous souhaitons reproduire.

L'idée principale de notre modèle d'approximation est d'accélérer le filtre bilatéral en réduisant la taille de N_i tout en se basant sur un support de taille équivalente. Pour ce faire, nous utilisons la composition de deux filtres, dont les supports sont orthogonaux. Notre approximation fonctionne ainsi en deux passes : dans la première, nous collectons un ensemble de faces voisines restreintes à une direction tangente à la surface, puis nous filtrons en utilisant seulement cet ensemble. Cette passe est appliquée à tous les sommets. Dans un second temps, nous filtrons le résultat de la passe précédente en utilisant une direction tangente orthogonale.

Cette approche est inspirée par la séparabilité du filtre bilatéral sur les images, dont l'efficacité visuelle est maximale lorsque les directions de séparation utilisées correspondent aux gradient locaux de l'image. Il est cependant nécessaire de noter que, contrairement à un filtrage gaussien, la décomposition ne mène pas à une solution exacte mais à une approximation [PvV05].

3.2.2 Notre approche

Dans le cas d'un maillage, la problématique principale est de déterminer deux directions pour chaque sommet permettant de collecter les ensembles restreints. Notre observation clé est que l'utilisation pour cela des directions de courbures minimales et maximales amène une décomposition consistante du filtre bilatéral en fonction de l'anisotropie locale de la surface, parvenant à capturer les arêtes vives, lesquelles sont efficaces pour "masquer" les artefacts d'approximation.

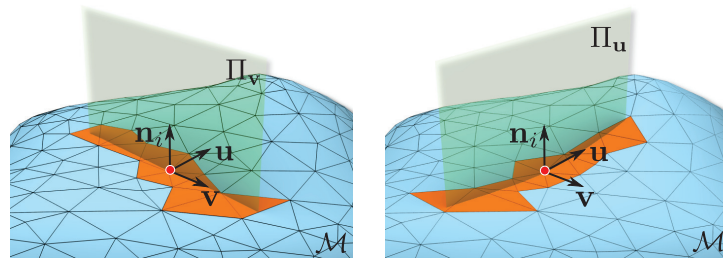


FIGURE 3.7: Deux plans locaux Π_u^i et Π_v^i (vert), permettant la requête de deux ensembles restreints de faces voisines (orange), une pour chaque passe, alignés avec les directions de courbures.

Nous devons à présent calculer les directions de courbures maximales et minimales $\{\mathbf{u}_i, \mathbf{v}_i\}$ à chaque sommet \mathbf{p}_i de M pour définir localement la direction de filtrage pour nos deux passes. Nous utilisons l'estimateur de Rusinkiewicz [Rus04] pour les directions de courbures. L'estimateur utilise le second tenseur fondamental sur le plan tangent pour obtenir les contraintes sur les dérivées de la normales, déterminées au sens des moindres carrés et moyennées sur les faces. Comme le bruit perturbe la plupart des estimateurs de courbure, nous lissons le champs des directions de courbures en utilisant le noyau $G_{\sigma_s/2}$. En pratique, cette opération est réalisée simultanément avec le pré-traitement des normales.

À l'aide des ces directions de courbures, nous pouvons définir deux plans orthogonaux $\Pi_{\mathbf{u}}^i = \{\mathbf{n}_i, \mathbf{u}_i\}$ et $\Pi_{\mathbf{v}}^i = \{\mathbf{n}_i, \mathbf{v}_i\}$, un pour chaque direction de courbure, et s'intersectant le long de la normale au sommet. Ces plans rencontrent les faces voisines et offrent un prédicat immédiat pour collecter l'ensemble restreint de faces voisines pour chaque passe. Ces plans et ensembles restreints sont illustrés sur la figure 3.7.

Les ensembles restreints de faces voisines liées à chaque passe sont définis comme l'intersection du voisinage complet N_i avec les plans

$$N_i^{\mathbf{u}} = \left\{ F_j \in N_i \mid F_j \cap \Pi_{\mathbf{u}}^i \neq \emptyset \right\} \quad (3.9)$$

Nous calculons $N_i^{\mathbf{v}}$ de façon similaire, et, dans chacun des cas, nous utilisons un procédé de remplissage par diffusion à l'aide d'une file dans laquelle seule les faces intersectant le plan courant sont placées, jusqu'à la distance maximale σ_s .

Finalement, notre approximation (SBL) consiste en la combinaison de deux filtrages BL restreints

$$SBL(\mathbf{p}_i) = BL_{N_i^{\mathbf{u}}} \left(BL_{N_i^{\mathbf{v}}}(\mathbf{p}_i) \right) \approx BL_{N_i}(\mathbf{p}_i) \quad (3.10)$$

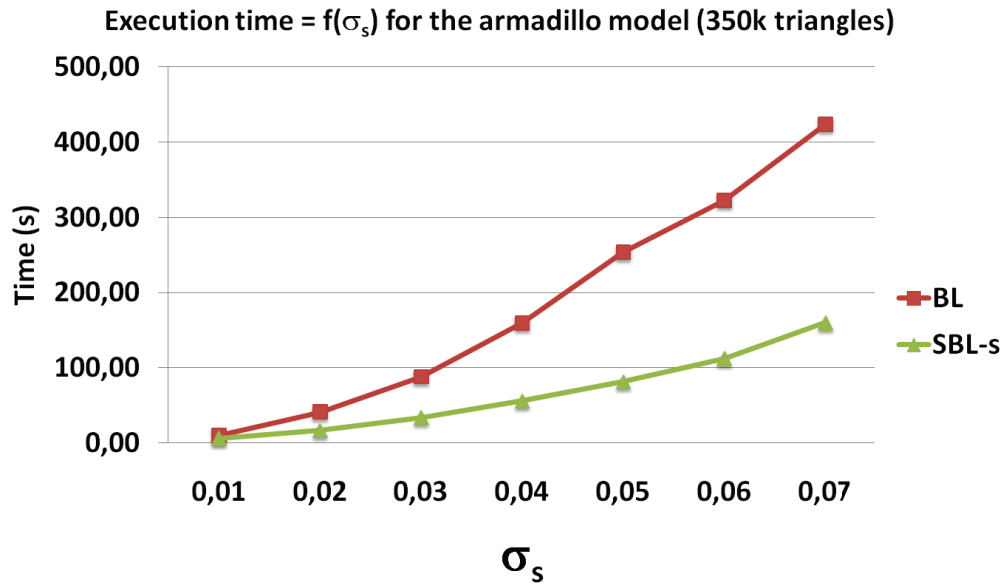
Variations Dans l'ensemble, le choix des directions de courbures lissées donne de meilleurs résultats d'approximation par rapport à l'utilisation directe des directions de courbure, ou le choix d'une paire de directions orthogonales aléatoires en chaque point. Dans la suite, nous utiliserons les notations décrites dans la table 3.1 pour les différentes variantes de l'algorithme.

3.2.3 Résultats

Nous analyserons dans cette partie l'amélioration de performance offerte par le filtre SBL et l'erreur induite par rapport au filtre BL original. Nous comparons aussi les trois différentes variations du filtre SBL par rapport à la précision de l'approximation. Les performances sont mesurées sur un Intel Core 2 Duo, 2.53 GHz, avec 4.0 GB de RAM, sur une seule thread. Toutes les tailles de filtrages sont données en proportion de la diagonale de la boîte englobante de l'objet.

algorithm	$\{\mathbf{u}, \mathbf{v}\}$
SBL-r	tangentes orthogonales aléatoires
SBL-u	directions de courbures
SBL-s	directions de courbures lissées

TABLE 3.1: Les différentes variations du filtre SBL.

FIGURE 3.8: Comparaison entre les filtres BL et SBL-s en fonction de σ_s sur le modèle Armadillo.

Dans la figure 3.8, nous mesurons le temps d'exécution de plusieurs processus de filtrage à l'aide des filtres BL et SBL-s, en augmentant la taille du support σ_s . On observe que le filtre SBL-s demeure entre 2 et 3 fois plus rapide que le filtre BL sur cet exemple, avec une accélération plus importante lorsque la taille du support augmente. Il faut noter que ces temps de calculs incluent aussi le processus de filtrage des normales, qui ne peut être évité pour des questions de robustesse. Le processus de filtrage des courbures utilisé pour l'algorithme SBL-s consomme une part conséquente du temps d'exécution, les variantes SBL-u et SBL-r étant environ deux fois plus rapides. Une réduction du support de filtrage permet un compromis vitesse-précision.

La figure 3.9 illustre l'erreur par sommet introduite par les filtres BL et SBL appliqués sur un modèle bruité artificiellement (c'est-à-dire pour lequel on peut comparer les

Modèle	Faces	BL (s)	SBL-s (s)	erreur RMS
sphere	20k	1.0	0.7	$8.3 \cdot 10^{-4}$
armadillo	350k	95	38	$3.4 \cdot 10^{-4}$
filigree	1M	647	152	$4.7 \cdot 10^{-4}$
Red_box	1.4M	600	193	$3.6 \cdot 10^{-4}$

TABLE 3.2: Erreurs et temps d'exécution pour différents modèles ($\sigma_s = 0.03$ et $\sigma_r = 0.03$). L'erreur RMS mesure la différence par sommet entre le BL exact et notre approximation. Les modèles sont normalisés suivant le cube unité.

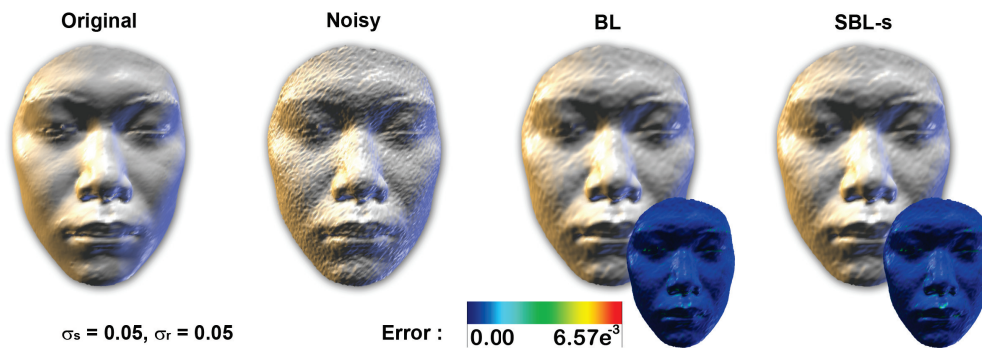


FIGURE 3.9: Comparaison entre l'efficacité des filtres BL et SBL-s. De gauche à droite : le modèle original, la même surface bruitée (40% de la longueur moyenne d'arête), le résultat du BL et le résultat du SBL-s.

résultats débruités à un modèle *parfait*). Qualitativement, on voit que notre filtre est globalement aussi efficace que le filtre BL que l'on tente d'approximer. La figure 3.10 confirme que les différences entre les deux résultats débruités (BL et SBL) sont très faibles.

Sur la figure 3.11 nous avons tracé la distribution des erreurs introduites par les trois variantes de notre schéma d'approximation en comparaison du filtre BL exact. Tout d'abord, on note que l'erreur globale est dans l'ensemble relativement faible dans les trois cas, et les distributions présentent une faible amplitude. Ensuite, nous observons comment notre stratégie basée sur les directions de courbures filtrées amène des résultats significativement meilleurs que les deux autres variantes. De façon assez surprenante, l'utilisation de directions aléatoires pour récolter les voisinages n'est pas particulièrement moins précise que l'utilisation des directions non lissées.

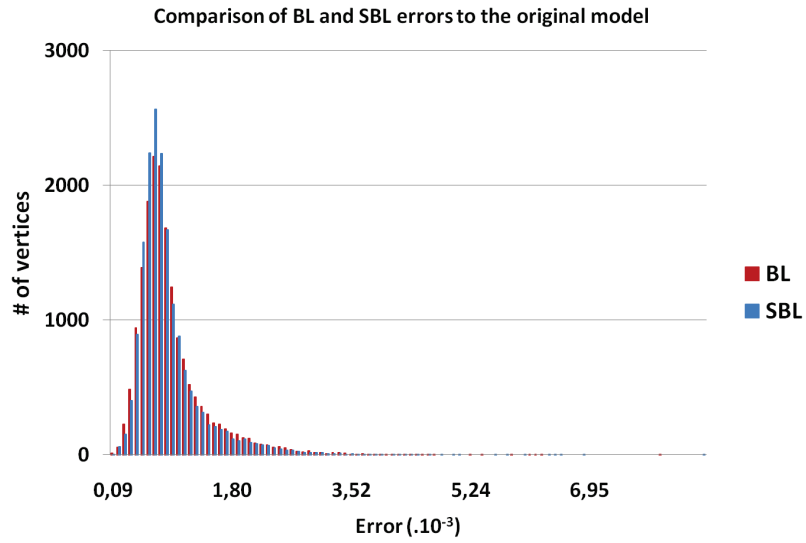


FIGURE 3.10: Histogrammes de l'erreur par sommet entre les modèles filtrés (BL et SBL) et l'original. Les différences entre les deux courbes sont difficiles à détecter pour une large plage de valeurs, démontrant la bonne approximation réalisée par notre méthode.

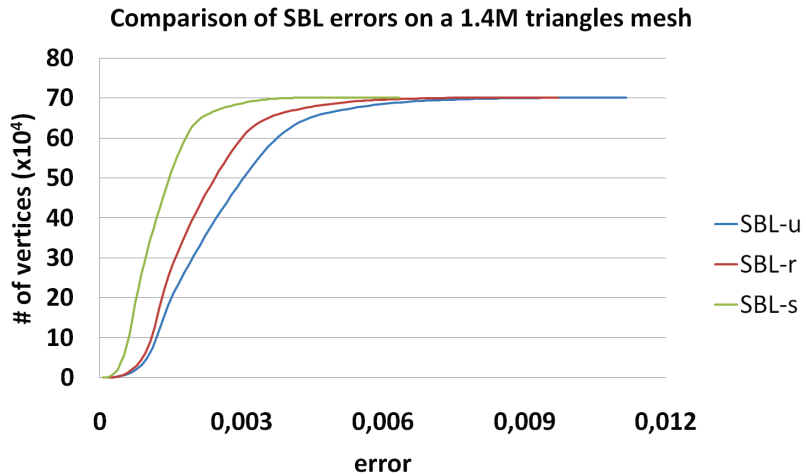


FIGURE 3.11: Histogrammes d'erreurs pour les différentes versions de notre filtre SBL (par rapport au filtre BL exact). Ces distributions ont été calculées pour le modèle `Red_circular_box`, avec $\sigma_s = 0.05$ et $\sigma_r = 0.02$.

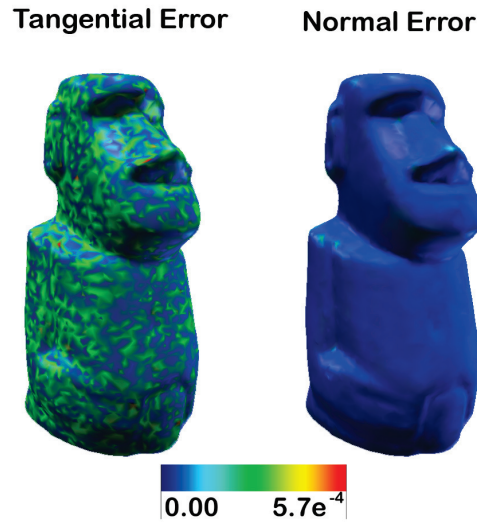


FIGURE 3.12: Erreur L^2 par sommet décomposée selon les composantes normales et tangentielles. L'erreur très faible sur la composante normale explique la fidélité visuelle de notre approximation.

La table 3.2 donne les résultats de temps d'exécution et d'erreur quadratique entre le BL exact et le filtre SBL-s sur quatre modèles différents, quelques-uns étant illustrés sur la figure 3.13. Nous pouvons observer que l'erreur reste particulièrement basse, même pour des tailles importantes de support (5% de la taille de l'objet) tandis que le processus est accéléré par un facteur allant de 2 à 3. D'après nos expériences, nous avons observé que l'erreur RMS est systématiquement inférieure à 10^{-3} , en considérant un maillage unitaire. Lorsque l'erreur est décomposée en composantes normales et tangentielles, il apparaît que la seconde domine drastiquement (fig. 3.12).

Finalement, la figure 3.13 montre les maillages finaux obtenus avec les filtres SBL. Tous les maillages sont redimensionnés pour être unitaires. Nous traçons l'erreur d'approximation entre le filtre BL exact et les trois variantes de l'approximation SBL pour chaque sommet. Il est intéressant de noter que l'erreur maximale relevée sur chaque maillage reste faible devant la taille de l'objet. En analysant ces différentes mesures, nous pouvons conclure que bien que non formellement séparable, notre approximation séparable du filtre bilatéral est très proche de la solution exacte.

Le choix particulier des directions de courbures lissées pour la décomposition du filtre s'est révélé plus intéressant que les autres variantes en termes de précisions.

La raison en est que les erreurs d'approximation sont distribuées le long des axes de séparation, et par conséquent mieux masquées lorsque ceux-ci sont alignés avec les anisotropie de la surface, c'est-à-dire les directions de courbures. Cependant, il est important de noter que les méthodes SBL-r et SBL-u sont environ deux fois plus rapides que le filtre SBL-s, puisque la phase de lissage des directions de courbures — bien que réalisée sur un support bien plus petit que le SBL lui-même — reste chère. Ces variantes apportent néanmoins un compromis qualité-vitesse intéressant.

3.2.4 Discussion

Nous avons introduit une approximation séparable du filtre bilatéral sur les maillages qui est simple, et facile à implémenter. En décomposant le filtre en deux filtres à supports plus petits, et alignés avec les courbures lissées, nous avons pu obtenir des maillages lisses en conservant leurs arêtes vives équivalents visuellement au filtre bilatéral, tout en étant plusieurs fois plus rapide. Notre modèle d'approximation peut être contrôlé intuitivement en choisissant judicieusement la taille du support du filtrage des directions de courbures, réalisant le compromis qualité-vitesse souhaité.

Cette technique d'accélération est orthogonale aux méthodes considérant des espaces de plus hautes dimensions, et peut par conséquent être combinée avec ces processus. Ces techniques accélèrent les requêtes d'informations sur les voisins, tandis que notre approche réduit le nombre de ces requêtes.

Il serait intéressant d'étudier l'application de ce principe de séparabilité à d'autres filtres, comme les moyennes non-locales (*NL-means*), ou même à d'autres dimensions : est-il envisageable de séparer approximativement un filtrage volumique ? Selon quelles directions ? Nous avons essayé de séparer les filtrages de normales ou de directions de courbures dans notre approche, mais de la petite taille des supports de ces filtrages résulte une accélération faible, et, du fait de la séparation aléatoire, une approximation de qualité moyenne.

Nous allons dans la suite de ce chapitre nous intéresser de façon plus générale à l'étude statistique du voisinage d'un élément du maillage, au travers des histogrammes locaux et de la médiane géométrique.

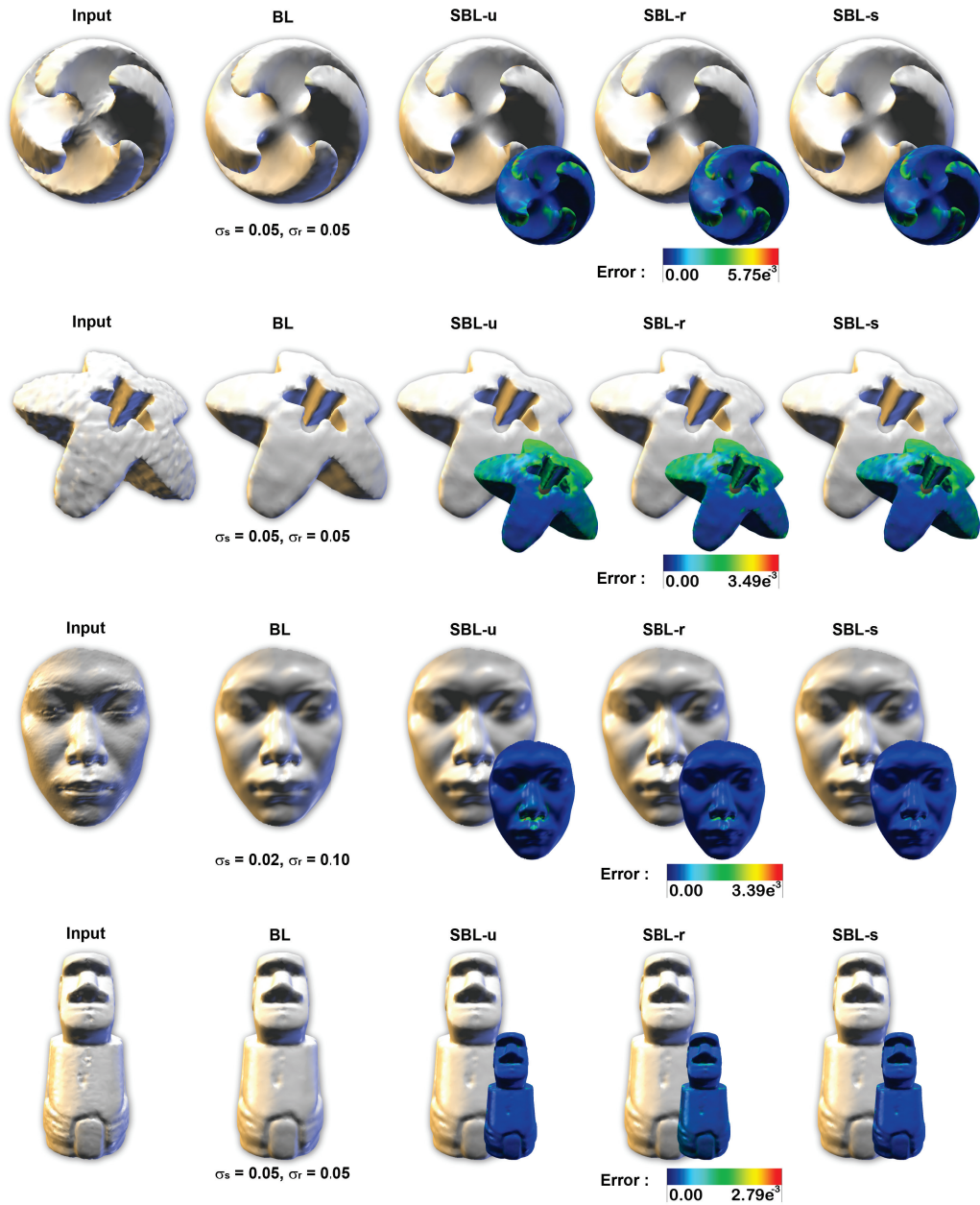


FIGURE 3.13: Les résultats de nos différents algorithmes sur des modèles variés. De gauche à droite : le modèle d'entrée, le résultat du filtre BL, la variante SBL-u (utilisant les directions de courbures sans la phase de lissage), la technique SBL-r (basée sur des directions tangentes aléatoires pour chaque point), la méthode SBL-s (les champs de directions de courbures ont été préalablement lissés). L'erreur indiquée est la norme L^2 de la distance par sommet entre le résultat du BL et notre algorithme, en proportion de la taille du modèle.

3.3 Histogrammes locaux lissés

3.3.1 Présentation

Nous nous basons dans toute cette partie sur les travaux de Solomon et Kass [KS10]. Le principe de leurs travaux est le suivant : de nombreuses approches de traitements d'images consistent à requérir les valeurs d'un certain nombre de voisins, en vue d'en déterminer la moyenne, la médiane, ou un percentile quelconque. Il s'agit, en quelque sorte, de construire l'histogramme des valeurs des pixels voisins, ou l'histogramme des valeurs cumulées, puis de déduire des propriétés statistiques de cet histogramme.

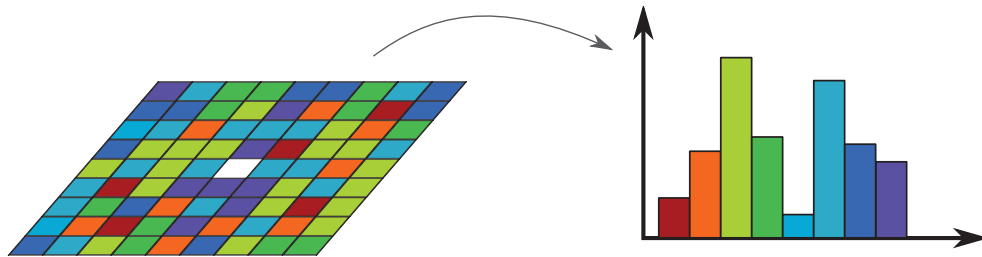


FIGURE 3.14: Principe de construction de l'histogramme local d'un pixel.

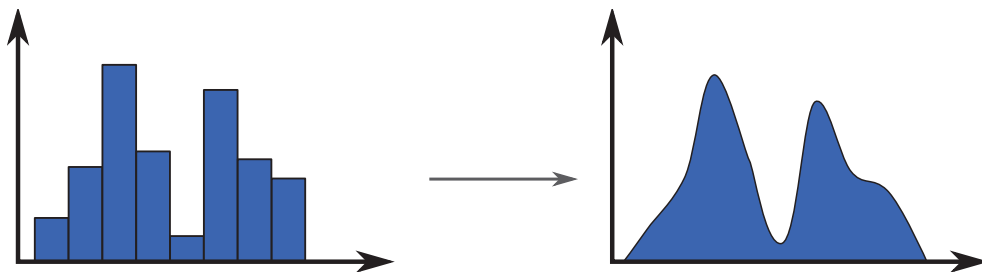


FIGURE 3.15: Évolution de l'histogramme : une fonction continue est utilisée en lieu et place d'une fonction porte, permettant la construction d'un histogramme continu, dont on peut extraire des modes et des percentiles.

Les auteurs ont alors eut l'idée de modifier la construction de cet histogramme : pourquoi, en lieu et place d'une fonction porte, ne pas utiliser une fonction continue, positive, centrée sur la valeur, et maximale en ce point, afin de créer un histogramme continu et lisse des valeurs avoisinantes ? Avec une image continue de la distribution des valeurs voisines, il est alors possible de déterminer des modes

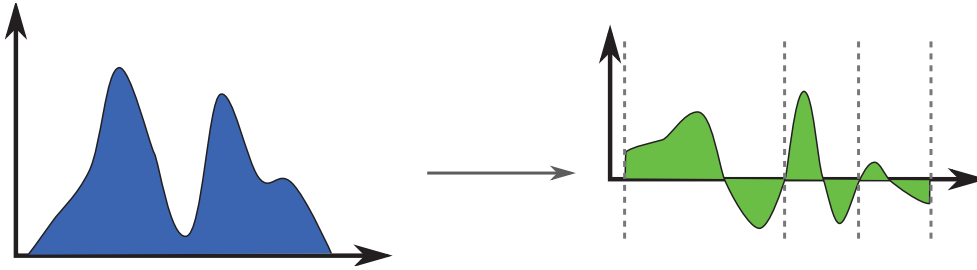


FIGURE 3.16: La détection des différents modes de l'histogrammes est facilitée par l'utilisation de la dérivée de la fonction continue. La détection de minima et maxima locaux est remplacée par la détection de zéros montants et descendants. Ici, 3 modes sont détectés sur la figure de gauche.

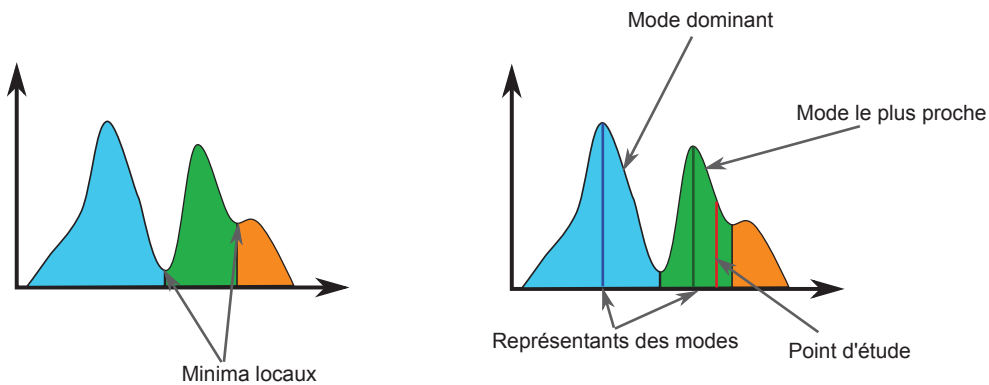


FIGURE 3.17: Lorsque les modes sont détectés, il est possible de choisir d'attribuer au sommet en cours de traitement le représentant du mode dominant de l'histogramme (bleu) ou celui du mode le plus proche de l'échantillon (vert).

parmi ces valeurs, de rechercher le mode dominant, ou le plus proche du pixel cible. Et en utilisant une fonction continue, l'histogramme cumulé continu permet la requête de n'importe quel percentile des valeurs.

Pour pouvoir détecter les différents modes de l'histogramme, il est nécessaire de détecter les minima et maxima de la courbe. Une solution chère et peu précise consisterait à dériver l'histogramme continu, et à analyser par la suite sa dérivée. Cependant, en gardant à l'esprit que l'histogramme est le résultat d'une somme de fonctions, la dérivée de cette somme s'écrit aussi comme la somme des dérivées de ces fonctions. Il suffit donc d'utiliser directement la dérivée d'une fonction gaussienne, par exemple, et de repérer les zéros montants et descendants afin d'en

déduire les minima et maxima de l'histogramme. Ces travaux nous ont inspiré l'utilisation d'une médiane géométrique pour le filtrage de surface.

3.3.2 Médiane géométrique

Définition

Considérant un ensemble X de n points x_i dans un espace E , et une distance $d(.,.)$ sur cet espace, la médiane géométrique \tilde{m} de X est définie comme

$$\tilde{m} = \operatorname{argmin}_m \sum_i d(x_i, m) \quad (3.11)$$

Pour mémoire, la moyenne \hat{m} de X peut s'écrire

$$\hat{m} = \operatorname{argmin}_m \sum_i d^2(x_i, m) \quad (3.12)$$

Géométriquement parlant, cela se traduit par les égalités suivantes :

$$\sum_i \mathbf{X}_i \hat{\mathbf{M}} = \vec{0} \quad (3.13)$$

$$\sum_i \frac{\mathbf{X}_i \tilde{\mathbf{M}}}{\|\mathbf{X}_i \tilde{\mathbf{M}}\|} = \vec{0} \quad (3.14)$$

L'expression visuelle de ces équations est présentée fig. 3.18.

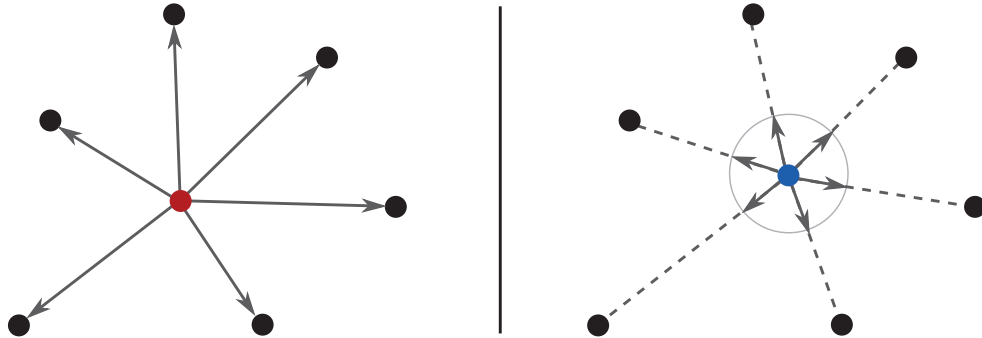


FIGURE 3.18: La moyenne (à gauche, en rouge) est positionnée de sorte que la somme des vecteurs $\mathbf{X}_i \hat{\mathbf{M}}$ soit nulle, tandis que la médiane (à droite, en bleu) est placée de sorte que la somme des vecteurs *unitaires* $\frac{\mathbf{X}_i \tilde{\mathbf{M}}}{\|\mathbf{X}_i \tilde{\mathbf{M}}\|}$ soit nulle.

Dans le cadre d'une métrique euclidienne, et en considérant la distance qui en découle, l'expression $d^2(x_i, g)$ est alors décomposable sur une base de l'espace, et l'expression explicite de g qui en résulte est :

$$\hat{m} = \frac{1}{n} \sum x_i \quad (3.15)$$

Cependant, il n'existe pas d'expression équivalente pour la médiane géométrique. La solution la plus simple pour parvenir à une approximation de la position est d'utiliser la formule de récurrence suivante, appelée algorithme de Weiszfeld.

$$\tilde{m}_k = \sum \frac{x_i}{\|x_i - m_{k-1}\|} / \sum \frac{1}{\|x_i - m_{k-1}\|} \quad (3.16)$$

Application

Afin de connaître mieux le comportement de ce descripteur, nous avons réalisé l'expérience suivante : dans \mathbb{R}^2 , une courbe C est approximée par un ensemble points x_i , doté de normales n_i , telles que

$$n_i = \frac{n_{i-1,i} + n_{i,i+1}}{2} \quad (3.17)$$

où $n_{i-1,i}$ désigne la normale du segment $[x_{i-1}, x_i]$.

On calcule, pour cet ensemble, la moyenne des positions, et la moyenne des normales (couleur verte), la médiane géométrique des positions, et la médiane géométrique des normales (couleur bleue). Enfin, une dernière grandeur, notée m_4 est calculée, la médiane géométrique dans l'espace \mathbb{R}^4 formé des positions et des moyennes des points. La figure 3.19 présente un résultat.

Afin d'obtenir des informations plus complètes sur le comportement de ces descripteurs, l'expérience est répétée plusieurs fois, et les positions \tilde{m} , \hat{m} et m_4 sont stockées. La figure 3.20 présente la moyenne (croix), l'écart-type (traits gras), et les extrema (traits fins), avec le même code de couleurs que précédemment.

On note tout d'abord que la médiane, si elle possède un écart-type légèrement plus important que la moyenne, se trouve plus proche de la courbe que cette dernière. De fait, la médiane est un descripteur intrinsèquement plus apte à préserver les caractéristiques sous-jacentes à la distribution que la moyenne.

3.3.3 Extension aux maillages

Histogrammes Locaux

Les algorithmes basés sur les histogrammes locaux ont été adaptés tels quels sur les maillages surfaciques, en traitant séparément chaque composante des quantités

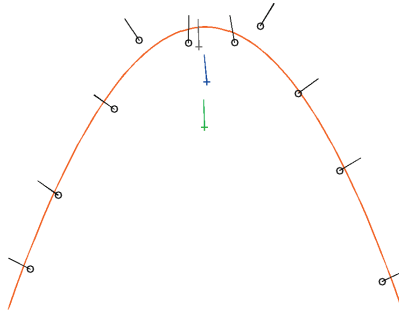


FIGURE 3.19: Exemple de calcul des moyenne (vert) et médiane (bleu) d'un ensemble de points (noir). Les cercles représentent les positions, les traits l'orientation de la normale. La courbe orange matérialise la courbe idéale (les points sont à l'origine sur la courbe, puis bruités).

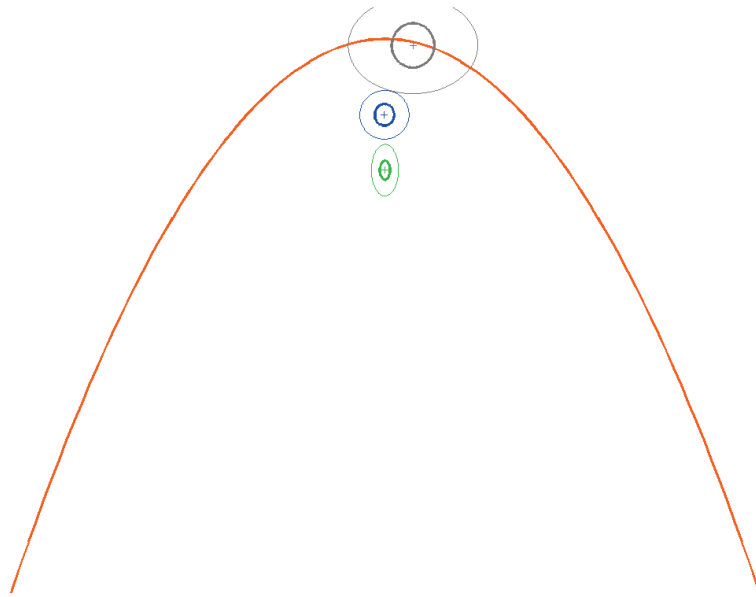


FIGURE 3.20: Exemples comparant les positions de la moyenne et de la médiane, ainsi que la régularité des positions. La moyenne est représentée en vert, la médiane en bleu. La médiane dans l'espace \mathbb{R}^4 est illustrée en gris. Différents estimateurs sont affichés ici : la moyenne (croix), l'écart-type (traits gras), et les extrema (traits fins).

vectérielles que l'on souhaite filtrer. Nous avons appliqué cet algorithme à quatre quantités : la position des sommets, p , leur normale n , leur Laplacien δ , leur couleur c . Les positions et normales correspondent aux définitions classiques, le Laplacien considéré ici est la discrétisation de l'opérateur de Laplace-Beltrami à l'aide des poids cotangents. Enfin, la couleur peut, par exemple, être le produit d'un calcul d'occlusion ambiante (*ambient occlusion*).

Une autre approche possible eut consisté en la construction d'un histogramme à n dimensions, mais plusieurs problématiques feraient alors leur apparition. Comment définir un mode ? Quelle serait la définition d'un percentile ?

Médiane géométrique

Nous avons utilisé la médiane géométrique, ou une médiane pondérée, de la même façon qu'une moyenne géométrique aurait pu être utilisée dans le cadre du filtrage de maillage. Conformément à ce qui a été réalisé dans l'étude des histogrammes locaux, nous nous sommes intéressé à la position, la normale, le Laplacien et la couleur des pixels, toutes ces quantités étant considérée dans \mathbb{R}^3 .

3.3.4 Résultats

L'application des histogrammes locaux sur les maillages n'a pas donné de résultats intéressants. En revanche, l'utilisation de la médiane a permis de dégager un certain nombre de résultats intéressants.

La figure 3.21 présente les résultats que nous avons obtenus en utilisant des filtres médian et moyen de support identiques. On constate que le filtre médian a une efficacité bien plus importante que le filtre moyen. Sur cette même figure, nous avons choisi de présenter le résultat d'un filtre médian sur chaque direction, et il nous est possible de constater que ce résultat est loin de celui produit par l'utilisation d'une médiane géométrique. On remarque qu'à support spatial équivalent, le pouvoir filtrant de la médiane géométrique est plus important que celui de la moyenne.

Discussion

L'application des histogrammes locaux s'est avérée infructueuse pour l'instant, mais il serait intéressant de prolonger l'étude de cet outil sur les maillages, afin de trouver un espace d'application permettant d'obtenir des résultats probants.

Cet étude des histogrammes a toutefois permis de développer un outil intéressant dans le cadre du traitement de maillages : la médiane géométrique. Nous avons pour l'instant restreint nos recherches à l'étude basique de ses propriétés, et

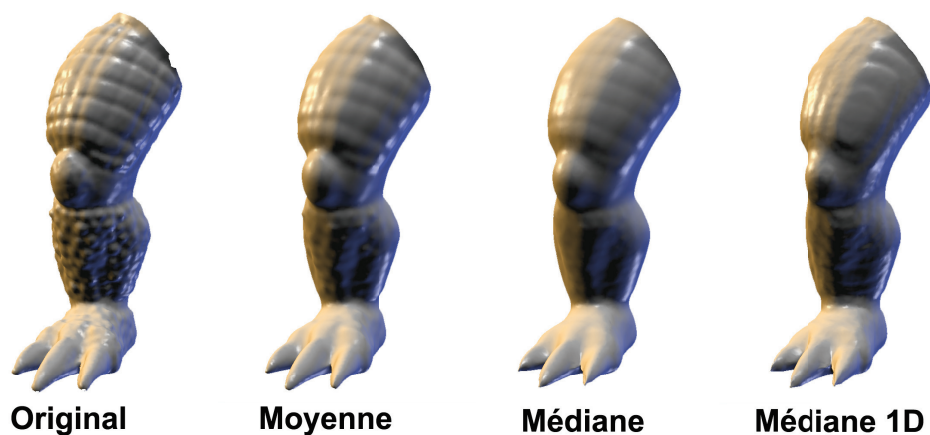


FIGURE 3.21: Résultats de l'application des filtres moyen et médian sur un maillage surfacique. On constate que le pouvoir filtrant de la médiane géométrique est bien plus important que celui de la moyenne. Cette illustration permet également de constater que la médiane géométrique est loin d'être composée des 3 médianes sur chaque composante (*médiane 1D* sur la figure).

son application directe aux maillages. Une piste possible d'études futures serait l'utilisation de cet outil, ou de ses variantes pondérées, dans d'autres filtres utilisant une moyenne locale, et d'observer l'effet de l'introduction de la médiane dans ces techniques.

Nous venons, au cours de ce chapitre, de parcourir différents filtres surfaciques, permettant d'améliorer la qualité d'un maillage, ou d'en éliminer le bruit. Dans le chapitre qui suit, nous allons aborder la dimension volumique de cette problématique.

Chapitre 4

Lissage volumique

Dans le cadre des simulations numériques, qu'il s'agisse de mécanique des fluides, ou du solide, il est impératif de disposer non pas seulement d'une modélisation de la surface, mais du volume complet. Ce volume doit également se trouver sous forme de maillage, et, de même qu'il existe de nombreux moyens d'obtenir un modèle surfacique, il existe beaucoup de méthodes permettant de générer des maillages volumiques. La multiplicité de ces techniques conduit à des objets répondant à des critères de qualité bien particuliers, propre à la méthode de génération utilisée. Par conséquent, il va régulièrement s'avérer nécessaire de lisser le maillage pour l'adapter aux critères de qualité correspondant à la tâche pour laquelle il sera employé.

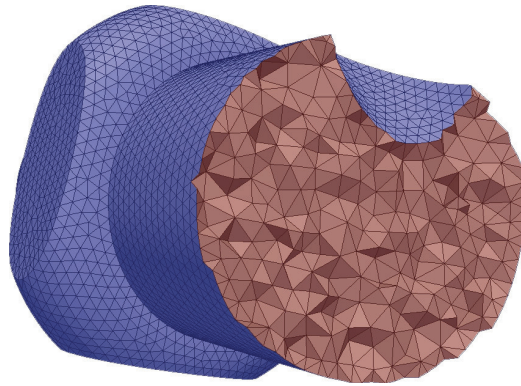


FIGURE 4.1: Exemple de maillage volumique. En bleu, les faces de bord F_b , en rouge, les faces internes F_i .

Primal		Dual	
Élément	Dimension	Élément	Dimension
Sommet	0	Cellule	3
Arête	1	Face	2
Face	2	Arête	1
Cellule	3	Sommet	0

TABLE 4.1: Correspondance entre les éléments constitutif des maillages primal et dual.

4.1 Contexte

4.1.1 Maillages volumiques

Ce que nous désignerons dans ce chapitre comme maillage volumique est la donnée des ensembles topologiques de sommets V , de faces F , éventuellement scindé en deux ensembles notés F_b et F_i , désignant respectivement les faces de bords (c'est-à-dire formant la surface du maillage) et les faces internes, de cellules C , ainsi que les données de positions P pour tous les sommets. L'ensemble N_b des normales des sommets de bord peut être également donné. Nous nous intéressons exclusivement à des maillages *conformes*, c'est-à-dire qu'un sommet ne peut pas se trouver au milieu d'un arête.

4.1.2 Notion de dual

Le dual M^* d'un maillage surfacique M est défini comme suit : à chaque sommet est associé une face, à chaque arête est associée une arête, à chaque face est associée un sommet. La figure 4.2 présente le dual d'un maillage plan. Cette structure est très utilisée dans les maillages surfaciques, notamment dans le cadre du calcul extérieur discret (DEC). Cette notion peut être étendue à des maillages volumiques, via le *Spatial Twist Continuum* (STC) défini par Murdoch *et al.* [MBBM97]. Le tableau 4.1 résume les équivalences alors obtenues entre les éléments du maillage de départ et ceux du dual.

4.1.3 Travaux existants

Les méthodes les plus élémentaires d'optimisation de maillages tétraédriques ou triangulaires impliquent des ajouts/suppression d'arêtes, afin de supprimer localement les éléments dégénérés. Cependant, ces méthodes ne sont pas applicables dans le cas de maillages quadrangulaires ou hexaédriques, et les méthodes

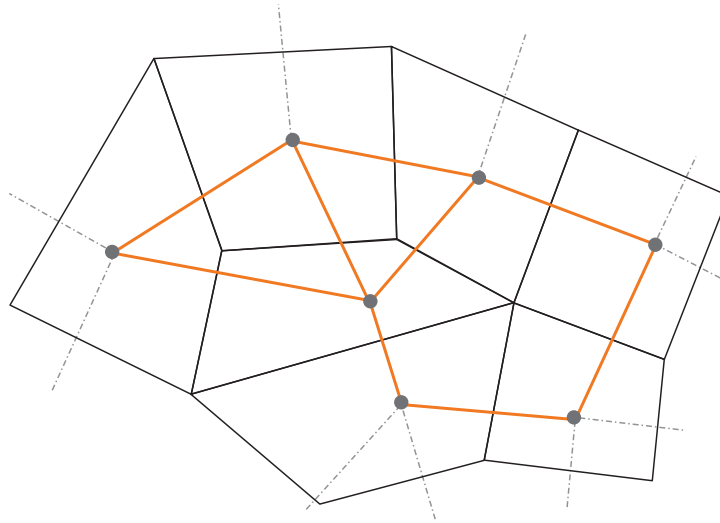


FIGURE 4.2: Dual d'un maillage plan : en traits noirs forts, le maillage initial. En trait orange forts, les arêtes du maillage dual. Les disques gris représentent les sommets de M^* associés aux faces de M .

employées consistent alors à déplacer les sommets en vue d'optimiser différents critères ou fonctionnelle.

Une première famille de méthodes s'appuie sur l'opérateur de Laplace-Beltrami et ses variantes, soit la combinaison moyennée des éléments pondérés du voisinage [Geo98, ZS00a, SV03]. Ces méthodes, simples d'implémentation, qui améliorent la qualité globale du maillage, mais peuvent générer localement des éléments de mauvaise qualité.

Une extension de l'opérateur de Laplace-Beltrami est le flux géométrique, utilisé par Zhang *et al.* [ZBX05] pour optimiser, dans un premier temps, les sommets de surface, avant de traiter les sommets internes afin d'obtenir de bons facteurs de forme (définis notamment à partir du déterminant ou du conditionnement de la matrice jacobienne des cellules).

Parmi les méthodes utilisant l'optimisation d'une fonctionnelle locale, on peut citer les travaux de Feitag *et al.* [FP]. Afin de réduire le coût de ces approches, certaines efficaces mais bien plus complexes que les précédentes, des techniques combinant les deux approches ont été proposées [CTS98, FOg97].

4.2 Dégauchissement

4.2.1 Problématique

Lorsqu'un polygone plongé dans \mathbb{R}^3 est composé de plus de 3 sommets, sa planéité, autrement dit le fait que tous les sommets le composant soient coplanaires, n'est plus assurée. Or, dans la mesure où ce polygone, en tant que face interne d'un maillage volumique, sert à une discrétisation de l'espace, on doit être en mesure de définir pour lui, entre autres propriétés, une normale. Seulement, quelle serait la signification de la normale d'un polygone non plan? La figure 4.3 présente l'objet connu sous le nom de lanterne de Schwarz, et qui présente ce problème sous la forme d'un assemblage de quadrangles non plans, dont il serait bien délicat de décider d'une triangulation, autrement dit une discrétisation en objets plans.

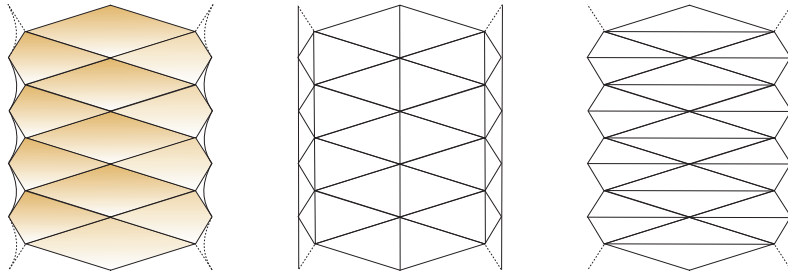


FIGURE 4.3: À gauche, la lanterne de Schwarz, qui illustre la difficulté de définir une approximation planaire pour un assemblage de polygones non plans. Ici, la lanterne est constituée de quadrangles, que les discrétisations proposées ici (au centre et à droite) ne permettent pas d'approximer de façon satisfaisante.

4.2.2 Notre approche

Normale d'un polygone

Soit une face f , composée des points P_i , $i = 1 \dots n$. On peut alors définir son centre de gravité G

$$G = \frac{1}{n} \sum_i P_i \quad (4.1)$$

On définit également une normale \mathbf{n}_f comme

$$\mathbf{n}_f = \frac{1}{2} \sum_i \frac{\mathbf{MP}_i \times \mathbf{MP}_{i+1}}{\|\mathbf{MP}_i \times \mathbf{MP}_{i+1}\|} \quad (4.2)$$

M étant un point quelconque de l'espace. En pratique, on choisit $M = G$, pour des questions de stabilité numérique. Cette normale a pour module une mesure de

l'aire du polygone, égale à la définition classique lorsque celui-ci est plan. La figure 4.4 illustre ce processus.

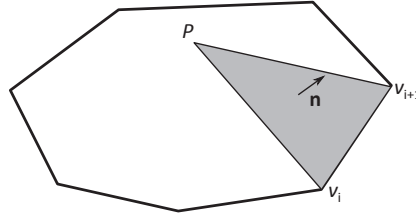


FIGURE 4.4: Calcul de la contribution à la normale d'une arête d'un polygone.

Gauchissement

Pour quantifier l'écart à la planéité d'un polygone, nous utiliserons dans la suite de cette partie le *gauchissement*, défini comme suit : soit une face f , composée des points P_i , $i = 1 \dots n$, et de normale \mathbf{n}_f , définie comme décrit ci-dessus. Alors le gauchissement g_f en degrés est donné par :

$$g_f = 90 - \arccos(\max_i (\cos(\mathbf{P}_i \mathbf{P}_{i+1}, \mathbf{n}_f))) \frac{180}{\pi} \quad (4.3)$$

Dégauchissement

Notre algorithme fonctionne de la manière suivante : pour chaque face f du maillage, on calcule la position de son centre de gravité G_f et sa normale \mathbf{n}_f . Ces deux éléments permettent de définir un plan cible pour les sommets de la face. Pour chaque point de la face, on calcule alors le vecteur de projection de ce point sur le plan cible (fig.4.5). En chaque point, les contributions de toutes les faces auxquelles il appartient sont sommées. Les points sont alors déplacés d'une fraction du déplacement final obtenu. Ce processus est itéré, jusqu'à convergence ou atteinte d'un nombre limit de boucles.

4.2.3 Résultats

Les figures 4.6 et 4.7 présentent le résultat de l'application de notre algorithme sur un maillage initialement dépourvu de tout gauchissement, sur lequel nous avons ajouté un bruit blanc afin de modifier la position de ses sommets. Nous pouvons constater, grâce au codage couleur du gauchissement en chaque sommet du maillage, que notre algorithme a rempli son rôle de dégauchissement. En l'absence de toute contrainte, il sera possible de dégauchir parfaitement toutes les faces de bord, mais au prix d'un nombre d'itérations parfois élevé.

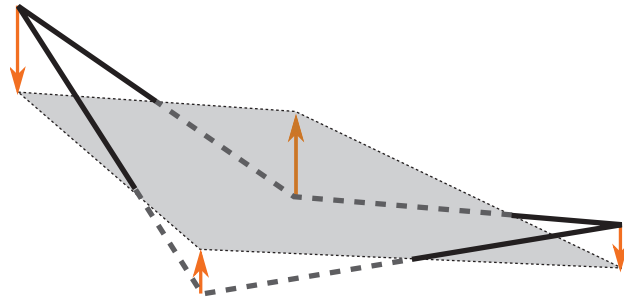


FIGURE 4.5: Illustration de la projections des sommets d'un quadrangle sur le plan moyen de la face.

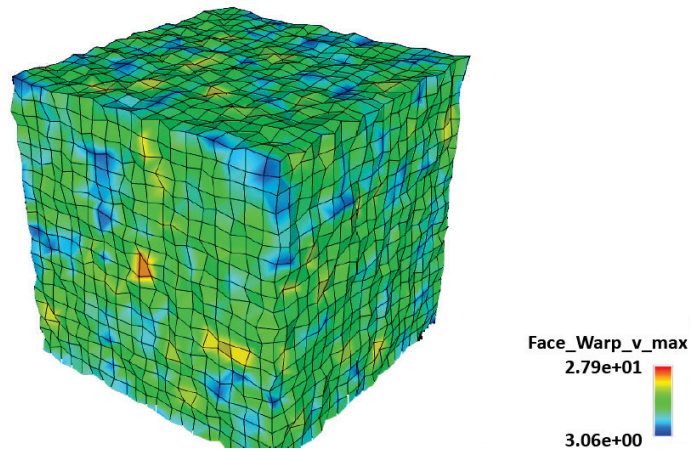


FIGURE 4.6: Un maillage volumique bruité auquel nous allons appliquer notre algorithme de dégauchissement. La couleur des sommets correspond au gauchissement maximal des faces auxquelles ils appartiennent.

Il est cependant important de constater que cette technique ne peut s'apparenter à un débruitage. En effet, le maillage initial était celui d'un cube parfait, géométrie dont le résultat final de notre technique se trouve relativement éloigné. On constate ainsi que, si la planéité des faces est un élément crucial dans le cadre des simulations physiques, il s'agit d'un critère très localisé, qui, de fait, ne permet pas une optimisation globale du maillage.

Ainsi, c'est dans l'optique d'un filtrage moins local d'un maillage volumique que nous avons mené les travaux présentés dans la suite de ce chapitre : y a-t-

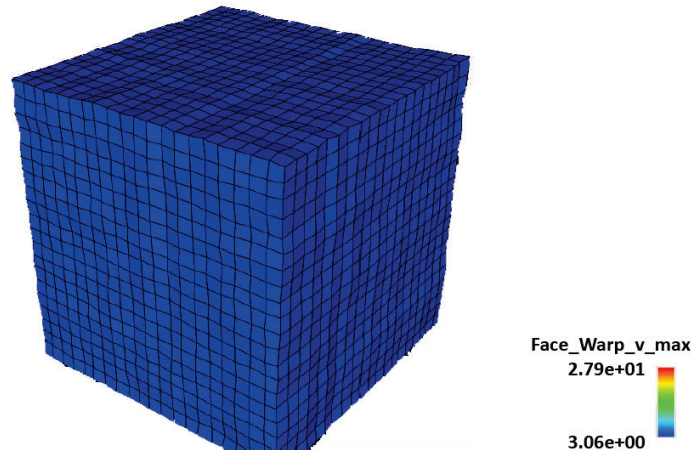


FIGURE 4.7: Résultat du dégauchissement du maillage de la figure 4.6. La couleur des sommets correspond au gauchissement maximal des faces auxquelles ils appartiennent.

il des critères locaux de forme ou d'orientation qui permettraient de retrouver la géométrie d'un modèle volumique bruité ?

4.3 Filtre bilatéral volumique

4.3.1 Problématique

Comme nous l'avons vu dans le chapitre 3, il arrive bien souvent que l'on soit amené à lisser ou filtrer un maillage, y compris en volume. Cependant, une différence majeure apparaît : les sommets, arêtes ou faces que l'on cherche à déplacer ne constituent plus la discrétisation ou l'approximation d'une surface. Il n'existe donc plus de notion de normale pour les sommets, et donc plus de courbure ou de plan tangents. Les techniques les plus classiques ou intuitives de lissage ne sont donc pas directement transposables au filtrage volumique.

En plus de la planéité des faces, d'autres critères de qualité peuvent être la cible d'algorithmes de lissages, dont certains seront présentés plus loin. Ceci permet de diminuer les risques d'instabilité numérique dans la simulation.

Le filtre bilatéral, introduit dans la section 3.1.2, nous a semblé un outil intéressant pour aborder cette problématique. En effet, c'est un type de filtre dont les propriétés ont été intensivement étudiées, sur les images comme sur les surfaces.

De plus, son caractère anisotrope convient parfaitement à l'application recherchée : pouvoir filtrer des maillages anisotropes sans pour autant détruire les irrégularités qui les caractérisent. La question est évidemment de savoir quelle sera l'espace de filtrage permettant de caractériser les similarités entre cellules.

4.3.2 Qualité de maillage

Les calculs de mécanique des fluides numérique (MFN) dans les couches limites requièrent des cellules de bonne qualité dans cette zone, en accord avec un certain nombre de critères. Nous présentons ici ceux pour lesquels nous avons essayé d'optimiser le maillage. Considérant deux cellules A et B d'un maillage M , A et B leurs centres respectifs, I l'intersection du segment $[AB]$ et de la face commune à A et B , de centre F , et de normale \mathbf{n} . M est le milieu de $[AB]$. Nous définissons à présent trois quantités afin de mesurer la qualité de l'interface entre A et B , illustrés sur la figure 4.8 et résumés dans la table 4.2.

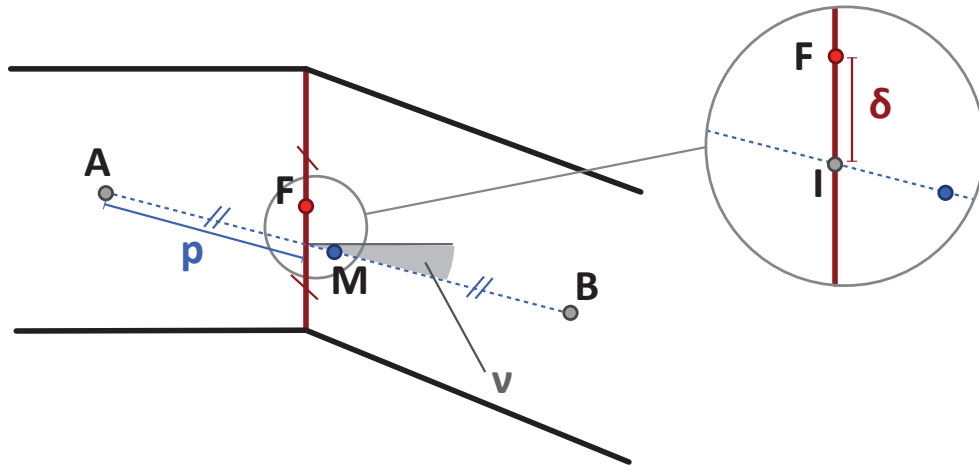


FIGURE 4.8: Illustration géométrique des critères de qualité de maillage.

Le décentrement est défini comme $\delta = \|\vec{FI}\|$, et, si non nul, introduit une erreur d'approximation dans la direction tangente à la face. Tandis que la quantité calculée aux centres des cellules est supposée être la valeur discrète en F , elle est en réalité la valeur en I . La pondération est définie comme $p = \frac{\min(\|\vec{AI}\|, \|\vec{BI}\|)}{\|\vec{AB}\|}$, et, si différente de $1/2$, donne une idée de l'erreur dans la direction de \vec{AB} , puisque I et M ne sont alors plus confondus. La non-orthogonalité, définie comme $v = (\vec{AB}, \mathbf{n})$, mesure la différence d'orthogonalité entre (AB) et la face. Chaque fois qu'une quantité vectorielle est calculée depuis A et B aux faces, elle est supposé se trouver dans la direction de \mathbf{n} , ce qui n'est plus le cas si ce paramètre diffère de 0.

TABLE 4.2: Définition des critères de qualité

Critère	Définition	Valeur optimale
δ	$\ \vec{IF}\ $	0
p	$\frac{\min(\ \vec{AI}\ , \ \vec{BI}\)}{\ \vec{AB}\ }$	$\frac{1}{2}$
v	(\vec{AB}, \mathbf{n})	0

4.3.3 Notre approche

Nous avons décidé de baser le noyau caractéristique de notre algorithme sur un outil d'analyse statistique : l'Analyse en Composantes Principales, ou ACP. Il s'agit d'un outil statistique permettant de dégager les axes principaux d'une série de données corrélées. Dans notre cas, on s'attachera à dégager les orientations principales des quadrilatères (ou hexaèdres) du maillage.

Soit $\mathbf{X}_1 \dots \mathbf{X}_n$ une famille de vecteurs à m -dimensions, \mathbf{u} le vecteur moyenne, on peut alors définir

$$M = \begin{pmatrix} \mathbf{X}_{1,1} - \mathbf{u}_1 & \cdots & \mathbf{X}_{n,1} - \mathbf{u}_1 \\ \vdots & \ddots & \vdots \\ \mathbf{X}_{1,m} - \mathbf{u}_m & \cdots & \mathbf{X}_{n,m} - \mathbf{u}_m \end{pmatrix} \quad (4.4)$$

puis $V = M^T M$, la matrice de covariance. Alors, le vecteur propre de V associé à la plus grande valeur propre correspond à la direction dans laquelle la corrélation des points est la plus forte.

Nous avons ainsi adapté cette définition aux quadrangles et hexaèdres, en considérant pour une cellule C la matrice $M_C = \{\mathbf{v}_i - \mathbf{g}_C\}$ des sommets de la cellule, centrée en \mathbf{g}_C , le centre de gravité de C .

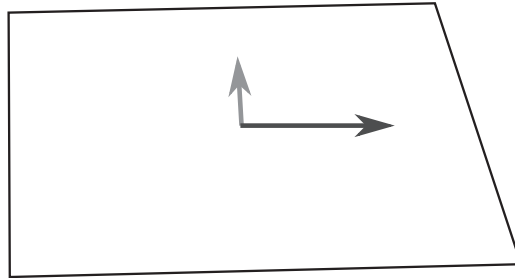


FIGURE 4.9: Exemple de l'application de l'ACP à la détermination des axes principaux d'un quadrangle. En gris foncé, l'axe principal de la cellule.

Cependant, les résultats de cet algorithme seul ne nous ont pas permis d'obtenir des cellules de qualité intéressante, et nous avons donc ajouté des termes dans le mouvement des sommets. En particulier, nous avons fait le constat qu'une cellule de *bonne* qualité requerrait le plus souvent des sommets équidistants du centre de gravité de la cellule. Nous avons par conséquent basé une partie du mouvement des sommets sur l'écart des sommets d'une cellule à la distance moyenne au sein de la cellule, comme illustré sur la figure 4.10.

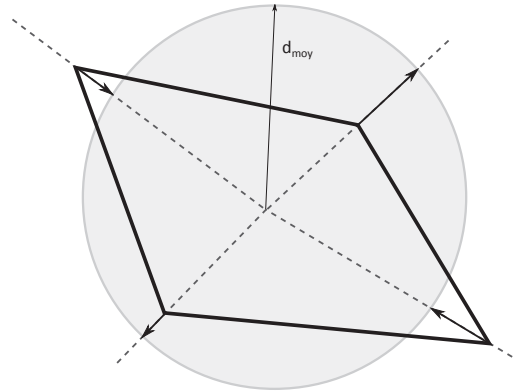


FIGURE 4.10: Mouvement des sommets induits par l'écart à la moyenne des distances au centre de la cellule.

4.3.4 Résultats

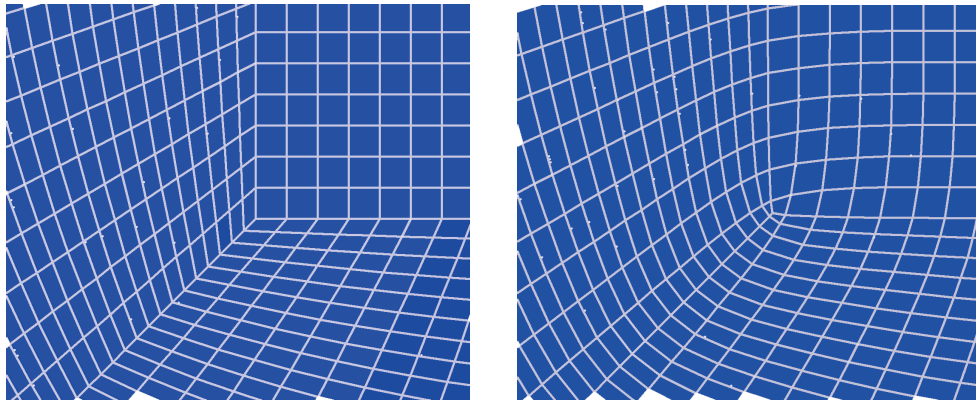


FIGURE 4.11: Gauche : maillage source. Droite : résultat de notre algorithme de filtrage bilatéral.

La figure 4.11 présente le résultat qualitatif de notre méthode sur un maillage volumique (vu de dessus sur l'illustration, son épaisseur est d'une cellule). On observe bien le but de l'algorithme : il ne s'agit pas d'améliorer la qualité moyenne d'une face ou cellule, mais bien de diminuer la déformation de la plus mauvaise des mailles. Nous cherchons en somme à *étaler* l'erreur sur plusieurs mailles, afin de s'assurer que la plus mauvaise d'entre elles ne soit pas trop mauvaise. Ce comportement, appliqué à la surface du maillage, présente un effet de lissage.

Nous avons appliqué notre algorithme à un maillage structuré afin d'en étudier les capacités d'un point de vue quantitatif. Nous présentons dans la table 4.3.4 les résultats en terme d'erreur maximale, les figures 4.13 à 4.15 présentent les histogrammes complets après une itération. L'amélioration du maillage au cours du processus est clairement visible. La valeur substantiellement haute du décentrement est due à la géométrie initiale du maillage, présenté sur la figure 4.12, qui présente volontairement un gradient de taille de mailles, et donc un décentrement non nul.

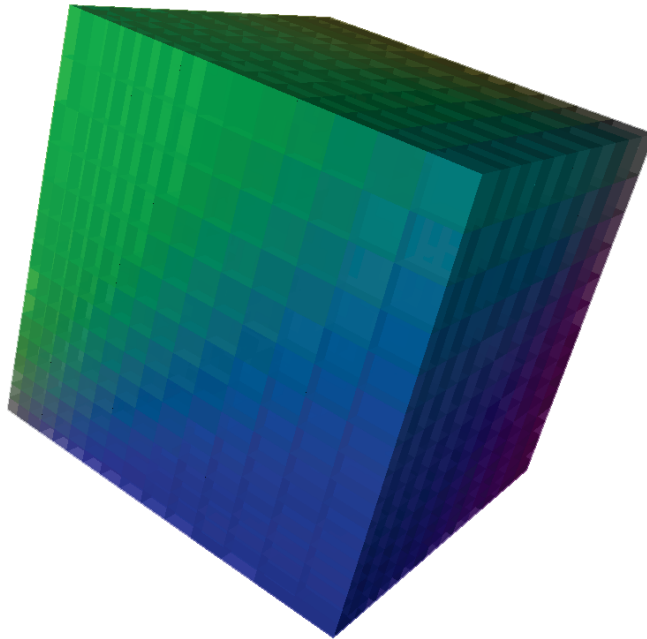


FIGURE 4.12: Maillage initial non bruité ayant servi aux calculs de performance de notre algorithme de filtrage bilatéral.

nombre d'itérations	Non-Orth.	gauchissement	décentrement
0	17.08	17.97	0.17
1	13.87	12.05	0.16
20	8.35	2.96	0.13

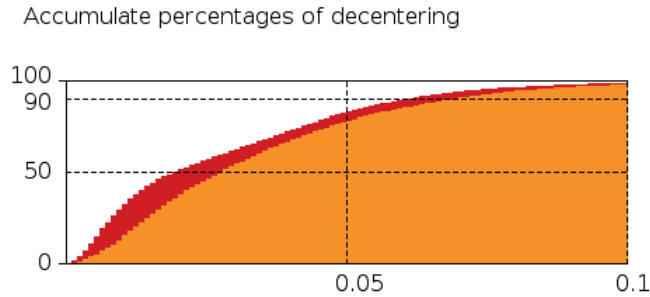


FIGURE 4.13: Histogramme cumulé des erreurs de décentrement. La partie orangée représente le maillage bruité, la courbe rouge indique l'évolution après une itération.

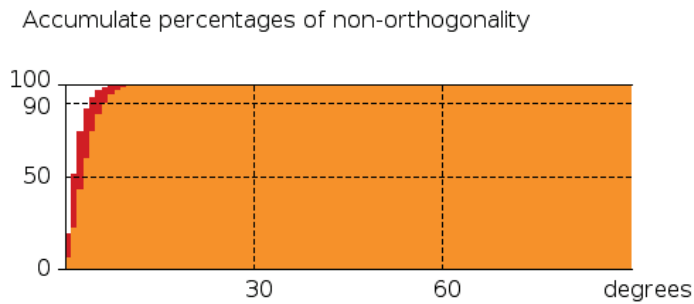


FIGURE 4.14: Histogramme cumulé des erreurs de non-orthogonalité. La partie orangée représente le maillage bruité, la courbe rouge indique l'évolution après une itération.

Discussion

Nous venons, au cours de ce chapitre, de présenter quelques méthodes pour corriger certains des défauts que l'on peut trouver dans un maillage volumique, en

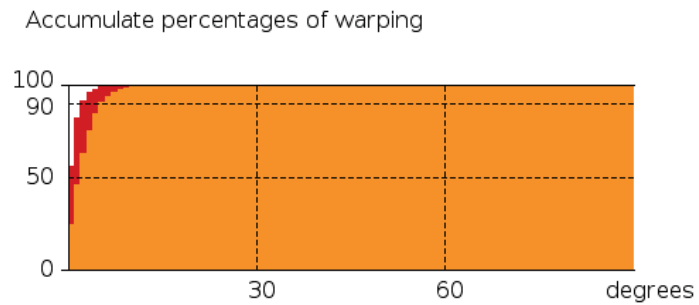


FIGURE 4.15: Histogramme cumulé des erreurs de gauchissement. La partie orangée représente le maillage bruité, la courbe rouge indique l'évolution après une itération.

ne travaillant que sur la position des sommets, c'est-à-dire en conservant toutes les propriétés topologiques de celui-ci. Cependant, il est un certain nombre de cas pour lesquels des changements de topologie sont nécessaires, et nous allons présenter dans le chapitre qui suit, une méthode permettant d'adapter un maillage à des besoins précis, en modifiant localement sa topologie (connectivité des mailles), tout en essayant d'en conserver la géométrie de frontière.

Chapitre 5

Remaillage : Prismes

Après avoir passé en revue différentes méthodes permettant l'amélioration de la qualité d'un maillage en modifiant ses composantes géométriques (essentiellement la position des sommets le composant), nous allons nous focaliser dans ce chapitre sur les modifications topologiques que l'on peut apporter, et, plus précisément sur la génération de mailles à proximité de la surface, en vue notamment de calculs de mécanique des fluides numérique.

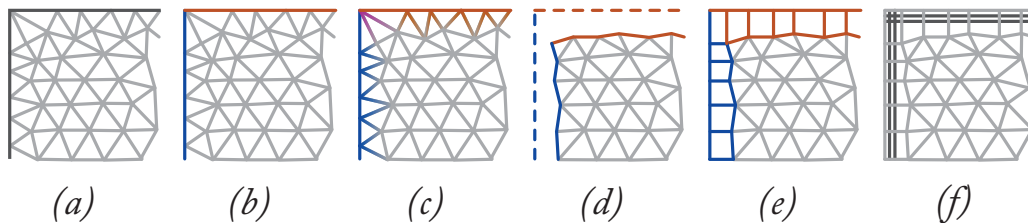


FIGURE 5.1: Résumé de l'algorithme. (a) Maillage initial, tétraédrique (b) Les faces de surface sont rassemblées (c) Les informations de regroupement sont transmises à la sous-couche en utilisant la connectivité sommet-sommet (d) La première couche de cellules est supprimée (e) Les prismes et hexaèdres sont construits depuis la sous-couche (f) Il est alors possible de diviser les cellules nouvellement créées pour obtenir des couches limites.

5.1 Introduction

La génération de maillages dont les cellules de bord sont adaptées aux calculs de couches limites, et dont les cellules internes sont suffisamment peu raffinées pour alléger le coût de calcul, à déjà donné lieu à de nombreuses recherches. Les maillages hexaédriques offrent une bonne base pour l'insertion de couches limites, mais sont bien plus complexes à générer que les maillages tétraédriques. Nous

proposons ici un compromis en créant une couche hybride prismes/hexaèdres à la surface d'un maillage tétraédrique, combinant les avantages des deux types de cellules. Nous allons d'abord présenter les critères de qualité de maillage qui ont motivé nos travaux, puis nous introduirons les travaux déjà effectués sur ce sujet. Nous discuterons ensuite des différentes étapes de notre algorithme, avant de conclure par une discussion autour des résultats.

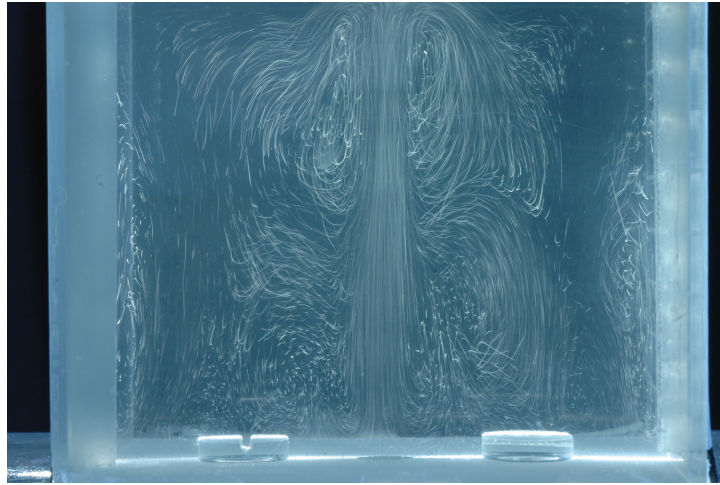


FIGURE 5.2: Exemple d'un mesure de *Particle Image Velocimetry* (PIV) : une photographie à temps d'ouverture élevé permet de suivre les trajectoires des particules. On observe de grandes disparités spatiales dans les vitesses, et mailler cet espace de façon adéquate pour réaliser une simulation de ce type d'écoulement demanderait un maillage anisotrope.

Les problèmes principaux de la génération de maillages adaptés à l'étude des couches limites proviennent du fait qu'ils sont hautement anisotropes. Dans les régions de viscosité, des cellules fines sont nécessaires, parfaitement alignées avec les frontières du domaine, tandis que le reste du maillage doit être le plus isotrope possible, avec le moins de cellules possibles afin d'alléger les temps de calcul. En se basant sur le travail de Garimella [Gar98], il est possible de regrouper les différentes techniques de génération de maillage anisotrope en trois catégories :

1. Utiliser une transformation de métrique dans un algorithme de génération isotrope afin de générer directement un maillage anisotrope.
2. Modifier localement un maillage isotrope existant, en repositionnant les nœuds et changeant la topologie locale.
3. Générer la zone anisotrope par une méthode spécifique, puis mailler *classiquement* le reste du volume par une méthode isotrope.

Génération directe Une première méthode pour obtenir le type de maillage souhaité est de générer directement un maillage isotrope, utilisant une transformation de domaine afin d'obtenir un maillage anisotrope dans les zones désirées. Mavripilis [Mav90] utilise des vecteurs d'étirement pour générer un champ de métrique. Un maillage isotrope est généré à l'aide d'une triangulation de Delaunay dans un espace de dimension supérieure, puis reprojété dans l'espace de départ. Borouchaki *et al.* ([Bor97a, Bor97b, Geo98]) généralisèrent l'idée d'une génération anisotrope via une triangulation de Delaunay basée sur une métrique spécifique, et Castro-Diaz *et al.* [Cas95] changèrent localement cette métrique, en ajoutant une condition d'orthogonalité à la paroi dans la couche, et leur métrique est définie de façon relative plutôt que globale, ce qui leur permet de travailler sur des problèmes multi-échelles. Ces techniques ne produisent que des maillages tétraédriques, ce qui entraîne un nombre de cellules plus important que dans le cas d'un maillage hexaédrique, et la génération d'éléments de mauvaise qualité (*slivers*) ne peut pas vraiment être évitée.

Remaillage local Les méthodes de modification de maillages isotropes sont plutôt bien représentées par la technique du front avançant (*advancing front*), développée par Hassan *et al.* [Has96]. Les méthodes de front avançant suivent toutes le même schéma : la surface de l'objet cible est triangulée et, depuis ces faces, des tétraèdres sont créés à l'intérieur du maillage en ajoutant de nouveaux points. Chaque face nouvellement générée fait à présent partie du *front*, et sera utilisée pour créer de nouveaux tétraèdres. Dans leur approche, Hassan *et al.* ajoutent une étape intermédiaire, dans laquelle le front de cellules nouvellement formées est contraint de se déplacer vers le précédent, afin de générer des couches fines. Ne créant que des tétraèdres, ces méthodes ont les mêmes inconvénients que les techniques exposées précédemment.

Couches avançantes Kallinderis *et al.* ([Kal95, Kal93, Kha95]) proposent une technique qui permet de générer des couches de prismes autour d'un corps, la hauteur des prismes augmentant avec la distance à la paroi, en fonction d'une spécification de l'utilisateur. Après que des couches spécifiques ont été générées, le reste du domaine est maillé à l'aide d'une combinaison d'*octree* et de méthodes de front avançant. Sharov et Nakahashi [Sha98] décrivent une méthode similaire, modifiée afin de générer des éléments de meilleure qualité, ainsi que tous les tétraèdres, en utilisant une méthode de Delaunay pour générer les tétraèdres internes. Cependant, les auteurs utilisent seulement des prismes à la surface du maillage, créant des cellules de faible qualité au vu du décentrement. Löhner [Loh95] et Pirzadeh [Pir94] ont proposé des méthodes similaires, et ces méthodes ont été généralisées par Garimella [Gar98]. On peut également citer dans cette famille d'algorithme les travaux proposés par Dyedov *et al.* [DEJ⁺09].

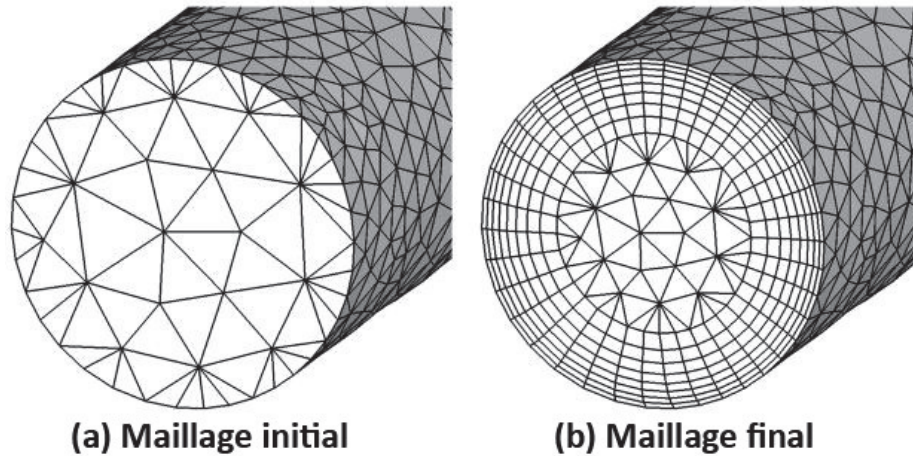


FIGURE 5.3: Illustration de l'algorithme décrit dans [DEJ⁺09]. source : [DEJ⁺09].

Maillages hexaédriques Dans les dernières années, de nombreux travaux de recherche ont tenté d'appliquer ces méthodes à des maillages purement hexaédriques, ou à dominante hexaédrique. Kovalev [Kov05] a proposé une méthode pour générer des maillages qui sont ensuite raffinés sur les parois ciblées pour créer des couches limites. L'inconvénient de cette méthode est la non-conformité du maillage résultant. Yamakawa et Shimada [Yam11] ont présenté une méthode pour créer un maillage purement hexaédrique à partir d'un maillage tétraédrique, en utilisant des maillages intermédiaires hybrides. La première phase de leur algorithme est une contraction globale du maillage (*pillowing*) du maillage, en vue d'insérer une couche d'hexaèdres.

Ito *et al.* [IN02] ont proposé une technique permettant d'insérer une couche de prismes dans un maillage tétraédrique, en utilisant des pyramides pour garantir la conformité du maillage. Le principal inconvénient des techniques exposées jusqu'ici est illustré sur la figure 5.4 : le décentrement, comme défini plus haut, est relativement élevé autour des zones de fortes courbures. Karman [Kar07] a présenté une méthode permettant de résoudre ce problème, en proposant l'insertion d'hexaèdres dans ces zones. Toutefois, l'utilisation de la méthode de *pillowing* a pour effet de modifier la géométrie complète du maillage, ce que nous souhaitons éviter.

5.2 Création de la couche limite

Notre méthode est composée de trois phases principales : il s'agit d'abord de détecter les différents ensembles de faces définissant la géométrie de la surface du

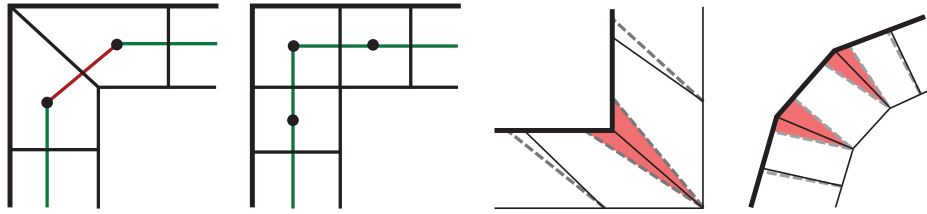


FIGURE 5.4: De gauche à droite : vue en coupe d'un maillage favorisant des valeurs élevées de décentrement, même vue de la configuration recherchée par notre algorithme, deux configurations (concavités et courbures légères) dans lesquelles on préférera les prismes (traits continus) aux hexaèdres (pointillés).

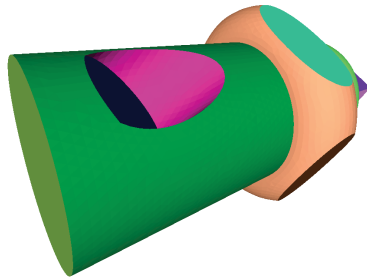


FIGURE 5.5: Le résultat de la partition des faces de surface sur un exemple simple.

maillage, puis la couche supérieure de cellules est supprimée, et finalement une couche de prismes et hexaèdres est créée dans l'espace laissé vide. La première étape vise à déterminer les endroits où insérer des hexaèdres en addition aux prismes afin d'obtenir un maillage de bonne qualité, en fonction des trois critères définis précédemment. Dans le processus de suppression, les faces et sommets n'appartenant qu'aux cellules que l'on va supprimer sont également rayés du maillage. Enfin, la projection de la sous-couche restante sur la surface initiale permet la création de prismes et hexaèdres.

5.2.1 Partition de la surface

Le but de cette étape est de partitionner les faces de surface en fonction de leur normale, afin de déterminer les positions futures des hexaèdres dans le maillage de sortie. Dans la phase d'initialisation, chaque face définit sa propre partition. Ensuite, on considère itérativement chaque paire de faces adjacentes qui ont des numéros de partition différents, et, si la différence de leurs normales est inférieure

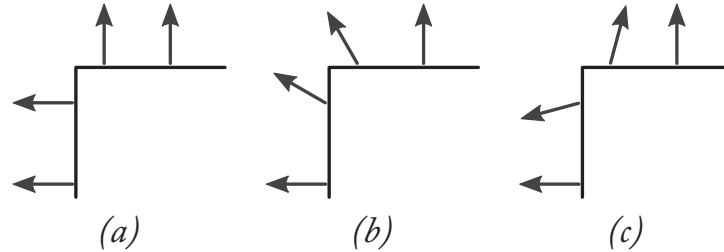


FIGURE 5.6: Processus de renforcement des normales. (a) La normale \vec{n} est déterminée en chaque face. (b) Puis, une version lissée \vec{n}_s est calculée. (c) Le résultat final \vec{n}_f est donné par $\vec{n}_f = \vec{n} - \lambda \times \vec{n}_s$, with $0 \leq \lambda \leq 1$.

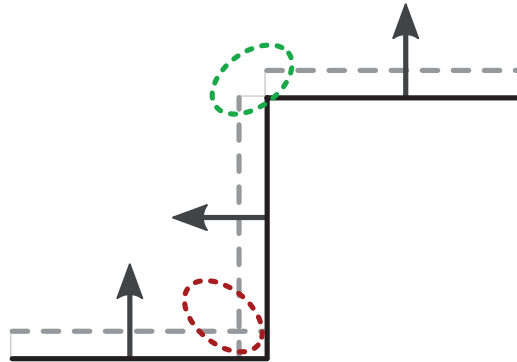


FIGURE 5.7: Le résultat de la détection de paires concaves : les faces du haut et de côté ne forment pas une paire concave, tandis que c'est le cas des face du bas et de côté.

à un seuil prédéfini t , les deux partitions sont fusionnées. L'algorithme s'arrête lorsqu'il n'y a plus de fusionnement à effectuer. La sortie de ce processus est illustré sur la figure 5.5.

Cet algorithme présente l'avantage d'être complètement contrôlé par un seul paramètre : le seuil de fusion t , qui est simple à appréhender pour l'utilisateur, étant directement lié à la précision du maillage final, via la détection ou non d'arêtes vives du maillage. Afin de rendre notre approche plus robuste, nous appliquons un renforcement des normales du maillages, comme décrit sur la figure 5.6. On détecte ensuite les zones concaves du maillage, qui amèneront des modifications des projections dans les étapes suivantes.

5.2.2 Suppression

L'étape suivante de notre algorithme est la détection et la suppression d'une couche de cellules, ce qui signifie qu'il nous faut supprimer chaque cellule dont au moins un des sommets se trouve sur la surface du maillage. Nous supposons que les faces de bord du maillage sont déjà connues (puisque déjà utilisées dans l'étape précédente), et elles sont supprimées. Puis, nous générons la liste V_s de tous les sommets se trouvant sur la surface. À partir de cette liste, il est possible de créer la liste F_s de toutes les faces internes possédant au moins un sommet dans V_s . L'étape suivante est la création de C_s , la liste de toutes les cellules ayant au moins une face appartenant à F_s . Il nous faut à présent revenir sur nos pas, et ajouter à F_s toutes les faces qui étaient seulement connectées à des cellules de C_s , et nous pouvons finalement ajouter à V_s tous les sommets uniquement connectés à des faces de F_s . Tous les éléments de V_s , F_s et C_s seront supprimés.

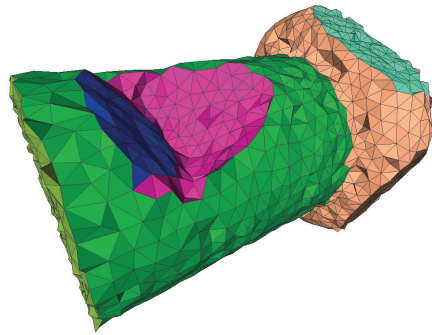


FIGURE 5.8: Résultat de la propagation des informations de surface à la sous-surface, puis de la suppression d'une couche de cellules.

5.2.3 Projections

Après la suppression des éléments de surface, la frontière du maillage est alors la *sous-surface*, la surface du maillage restant. Il nous faut projeter cette sous-surface sur la surface originale. Nous devons à présent déterminer sur combien de partitions nous allons devoir projeter chacun des sommets du maillage, et combien d'entre eux forment une paire concave, afin de déterminer combien de projections chaque sommet engendrera. Chaque sommet de la sous-surface hérite des informations de partition de ses voisins de surface, ce qui permet de répondre à la première question.

5.2.4 Cas d'un plan

Considérons dans un premier temps le cas d'un maillage dont la surface est un plan infini. Alors, chaque sommet de la sous-surface doit être projeté une seule fois sur la surface, et, depuis les triangles présents dans la sous-surface, des prismes sont extrudés jusqu'à la surface. Si le seuil de détection des partitions est suffisamment important, une sphère peut ainsi être considérée comme un plan infini, les faces de surfaces ne formant alors qu'une seule partition. De ce fait, tous les sommets de la sous-surface sont projetés sur la surface de la sphère, formant autant de prismes qu'il y a de triangles définissant la sous-surface. Afin de parvenir à une bonne reconstitution de la surface, nous utilisons ici l'algorithme des *Hermite Point Set Surface* [AA09], une reconstruction lisse interpolante. Nous cherchons alors les trois plus proches voisins sur la surface appartenant au(x) même(s) partitions, et nous sommes alors capables de projeter les points de sous-surface sur la surface.

5.2.5 Arêtes vives

Afin d'assurer la qualité du maillage de sortie d'après les critères que nous avons défini ci-dessus, et plus spécifiquement le décentrement, nous avons besoin de créer des hexaèdres le long des arêtes vives du maillage. Chaque fois qu'un sommet doit être projeté sur deux partitions différentes (à condition qu'elles ne forment pas une paire concave, auquel cas elles ne sont considérées que comme une seule partition), nous imposons la création d'un troisième projeté, situé sur l'intersection de ces deux partitions, autrement dit sur l'arête vive. De façon similaire, quatre projections sont ajoutées à un sommet qui doit être projeté sur trois partitions, comme décrit sur la figure 5.9.

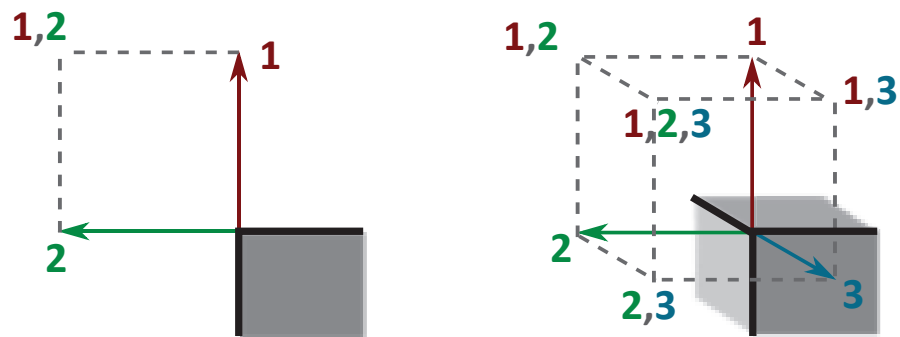


FIGURE 5.9: Gauche : modèle pour la projection d'un point sur deux partitions différentes. Droite : projection d'un point sur trois partitions.

5.2.6 Concavités

Si deux partitions, ou plus, forment un ensemble concave, il serait contre-productif d'ajouter des sommets dans cette zone en créant des hexaèdres en plus des prismes. C'est pourquoi les modèles utilisés pour les partitions multiples est ici altéré. Une paire concave est alors considérée comme une seule partition, et une seule projection d'un sommet de sous-surface est nécessaire à cet endroit. Lorsque la liste des différentes partitions auxquelles appartient un sommet est réalisée, les paires concaves sont notées en début de liste, par convention.

Considérons un point p de la sous-surface, appartenant aux partitions (a, b, c) , b et c formant une paire concave. Alors nous pouvons dire que p appartient à (b, c, a) , et doit être projeté sur deux partitions. Cela signifie que p aura trois projetés : p_1 , qui appartiendra à (b, c) , p_2 à (a) , et p_3 à (a, b, c) . p_1 sera donc la projection de p sur la courbe définie par les trois points de la surface les plus proches de p et appartenant à la fois à (b) et (c) , p_2 sera projeté sur la surface définie par les trois plus proches points de la surface appartenant à (a) , et p_3 sera projeté sur l'intersection de (a) , (b) et (c) , soit, dans de très nombreux cas, un seul point.

5.2.7 Relaxation

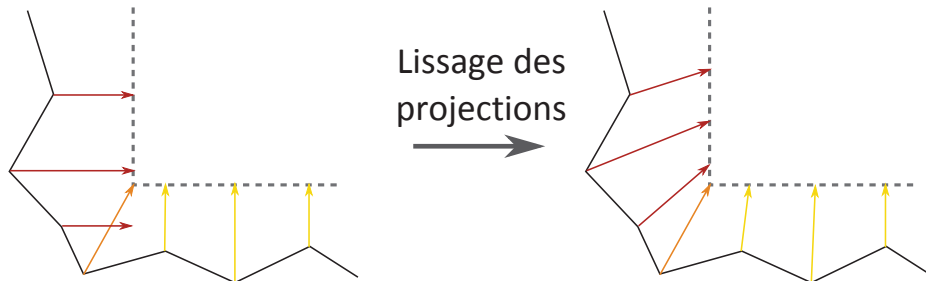


FIGURE 5.10: Lissage effectué sur les vecteurs de projection, afin d'éviter les auto-intersections de géométrie dans le résultat final.

Ces projections peuvent générer des auto-intersections de la géométrie, plus particulièrement autour des concavités. Nous appliquons ici une phase de relaxation sur les vecteurs de projection (fig. 5.10, en prenant soin de laisser intacts les projections sur les arêtes vives de la géométrie, suivie par une nouvelle projection sur la surface.

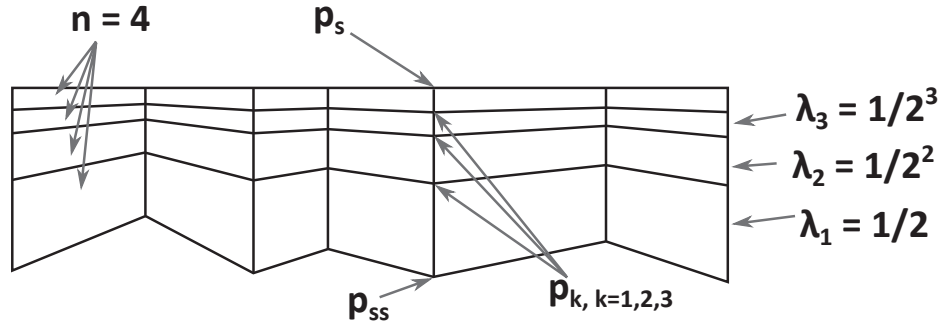


FIGURE 5.11: Création de la couche limite après la formation des prismes. Dans notre exemple, $n = 4$ et $\alpha = 2$.

5.2.8 Découpe en couches limites

À présent que la surface du maillage est faite de prismes et d'hexaèdres, il est relativement aisé de créer des couches limites aux endroits désirés du maillage. Nous pouvons associer à chaque partition un entier n indiquant combien de cellules doivent être créées entre la sous-surface et la surface, et un facteur α , qui encode la variation de la taille des cellules vers la surface. Pour chaque arête reliant un sommet de surface p_s à un sommet de sous-surface p_{ss} , nous insérons donc $n - 1$ points p_1, \dots, p_{n-1} de sorte que

$$p_k = \lambda_k p_{ss} + (1 - \lambda_k) p_s \quad (5.1)$$

, avec $\lambda_k = \frac{1}{\alpha^k}$ (fig. 5.11).

Cette méthode, par opposition à une création de cellules parfaitement parallèles à la surface, permet de répartir l'erreur de parallélisme sur toutes les couches, plutôt que d'avoir de très bonnes cellules près de la surface, et une couche de cellules très mauvaises faisant la liaison avec la sous-surface. En effet, il est préférable pour le code de calcul d'avoir une erreur maximale la plus faible possible, plutôt qu'une erreur moyenne faible.

5.3 Résultats

Tout le processus décrit ici a été intégré au logiciel libre de MFN *Code_Saturne*, développé par EDF R&D, dans une implémentation monoprocesseur. Les figures 5.12 à 5.14 présentent quelques-uns des maillages que nous sommes parvenu à générer. Le choix du facteur α lors de la création de couches limites permet le contrôle de la pondération, tandis que l'insertion d'hexaèdres au milieu des prismes

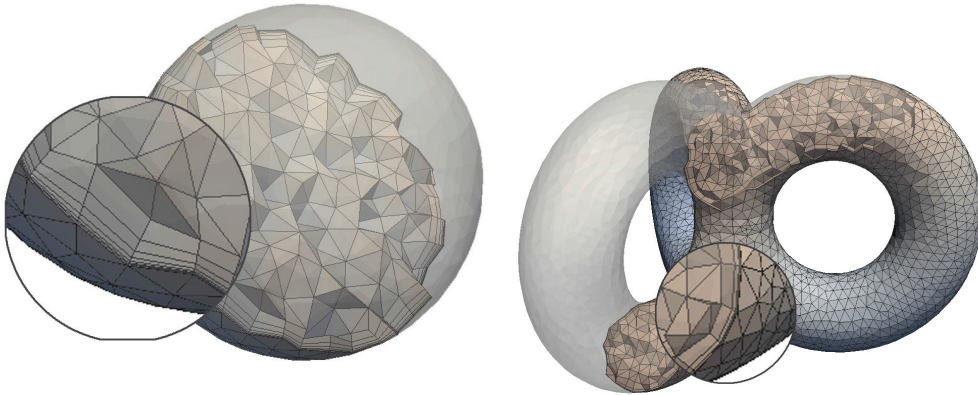


FIGURE 5.12: Deux résultats pour différents modèles et paramètres de couches limites : variation brutale sur les tores, plus lente sur la sphère.

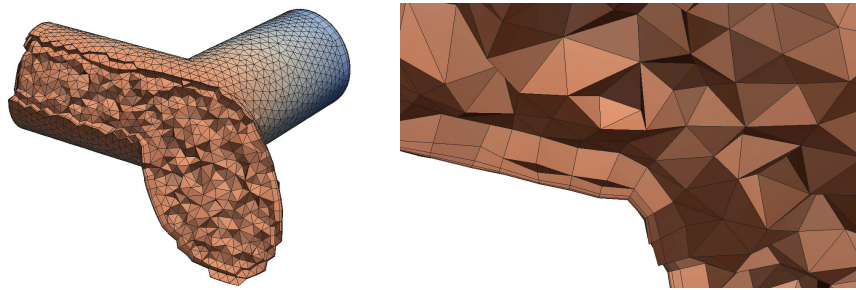


FIGURE 5.13: Le long des jonctions de cylindres, les faces sont sur la même partition, par conséquent la couche limite insérée ici est continue.

aux arêtes vives maintien un décentrement faible sur le maillage. La nature géométrique des prismes et hexaèdres assure que la non-orthogonalité reste à des seuils bas. La table 5.1 donne une vue d'ensemble des performances de l'algorithme. La projection et le remaillage (reconstruire toute la structure du maillage, en termes de sommets, faces, cellules) pèsent chacun pour environ 45% du temps de calcul, l'insertion de la couche limite pour environ 7%.

Discussion

Les résultats présentés ici sont encourageants, mais manquent de robustesse, de part le passage par la sous-surface. En effet, si la surface d'un maillage approxime la géométrie, et s'en trouve par conséquent plutôt régulière, il n'en est rien pour la sous-surface, qui peut présenter, selon le mailleur, de grandes irrégularités. Il est

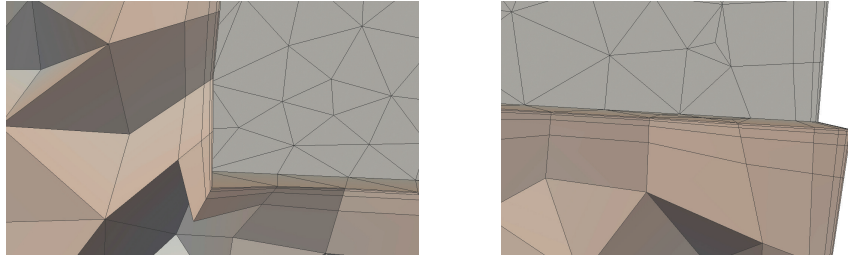


FIGURE 5.14: Un exemple de maillage de concavité (gauche) et de convexité (droite).

TABLE 5.1: Performances en temps de notre approche.

Modèle	Taille (# de cellules)	Temps de calcul
Sphere	13k	3.6s
Cylinder	54k	74s
TriCylinders	64k	106s
TriTorus	80k	190s

à noter que, selon le rapport aire/volume d'un modèle, la proportion des cellules modifiées par rapport au nombre de cellules total peut considérablement varier, et affecte d'autant les temps de calcul.

Nous avons amélioré les performances de notre algorithme en détectant certaines situations mettant à mal notre méthode, mais notre recherche n'est sans doute pas exhaustive, et pourrait faire l'objet d'études futures. Il est également à noter que cet algorithme ne fonctionne que dans les zones d'une certaine épaisseur, en l'occurrence au minimum trois épaisseurs de cellules. Cette limitation pourrait être prise en charge, mais demanderait d'ajouter à nouveau des cas particuliers à la technique.

Nous venons néanmoins de montrer comment il est possible, à partir d'un maillage non spécifique, de créer un modèle adapté à des calculs de mécanique des fluides numérique, à l'aide d'un remaillage local du modèle. Dans le prochain chapitre, nous allons étendre un peu plus notre champ d'action, en nous penchant sur la génération complète d'un maillage.

Chapitre 6

Maillages cristallins

Nous avons vu au cours des chapitres précédents comment, à partir d'un maillage existant, il était possible d'en modifier la topologie ou la géométrie pour l'adapter aux besoins des utilisateurs. Il est temps à présent de se pencher sur la question de la génération de ces maillages. Après un passage en revue des différentes méthodes existantes de génération, nous allons présenter une approche, basée sur le mouvement de particules et la triangulation de Delaunay. Notre idée est de pouvoir proposer un remaillage local, après suppressions de quelques mailles défectueuses dans un maillage complet par exemple.

6.1 Travaux précédents

Triangulation de Delaunay et diagramme de Voronoï

L'un des outils les plus puissants et les plus utilisés dans la génération de maillage est la triangulation de Delaunay d'un ensemble de points. Le principe de cet outil est le suivant : aucun des points de l'ensemble triangulé ne doit se trouver à l'intérieur du cercle circonscrit d'un triangle. Cette construction est le dual du diagramme de Voronoï de l'ensemble des points (illustré sur la figure 6.1). Ce dual est également très utilisé dans les techniques de maillages, via notamment les techniques de pavage. Les techniques visant à obtenir un maillage à l'aide d'une fonction objectif s'appuyant sur ces deux outils sont appelées triangulations de Delaunay optimisées (ODT) et les tessellations de Voronoï centroïdales (CVT), dont nous allons présenter quelques exemples.

Lévy et Liu [LL10] ont présenté récemment leur algorithme L_p -CVT. Là où les algorithmes de CVT se sont restreints à l'utilisation de la norme 2, les auteurs ont choisi de considérer la norme p , quel que soit l'entier p . Ils ajoutent également un terme modifiant localement la métrique, permettant d'introduire de l'anisotropie

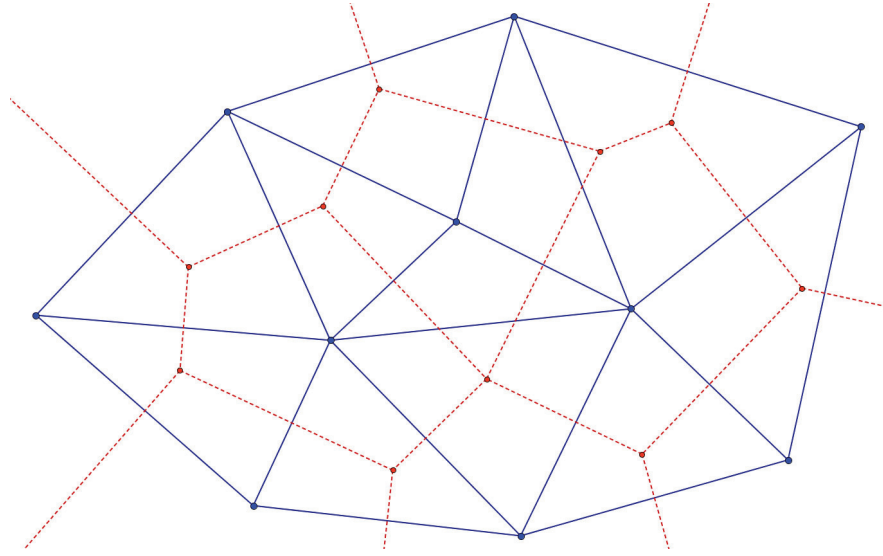


FIGURE 6.1: Triangulation de Delaunay (traits bleus continus) et diagramme de Voronoï (traits rouges pointillés) d'un ensemble de points.

dans le processus. Ainsi, la fonctionnelle classique régissant la CVT d'un ensemble de points \mathbf{X} s'écrit

$$F(\mathbf{X}) = \sum_i \int_{\Omega_i \cap \Omega} \|\mathbf{y} - \mathbf{x}_i\|^2 d\mathbf{y} \quad (6.1)$$

où Ω_i désigne la cellule de Voronoï du point \mathbf{x}_i , et Ω l'ensemble du domaine. La nouvelle formulation proposée par Lévy et Liu s'écrit alors

$$F_{L_p}(\mathbf{X}) = \sum_i \int_{\Omega_i \cap \Omega} \|M_y[\mathbf{y} - \mathbf{x}_i]\|_p^p d\mathbf{y} \quad (6.2)$$

Maillages hexaédriques

Les résultats que l'on peut attendre de la création d'un maillage hexaédrique sont loins de la qualité qui peut être obtenue par triangulation. Néanmoins, des schémas intéressants ont été proposés, comme celui présenté par Walton *et al.* [WBS02]. Les auteurs développent un algorithme de génération d'un maillage hexaédrique qui rappelle le processus de sculpture. À partir d'une boîte englobante sur-dimensionnée, les auteurs plongent le maillage dans une grille, ce qui crée un maillage hexaédrique isotropique, parfaitement régulier, *structuré*, mais très imprécis sur les arêtes vives. Ensuite, les hexaèdres de bord sont modifiés de façon à approcher au mieux la géométrie originale. Le principal désavantage de ce schéma réside dans l'uniformité de l'échantillonnage, d'où résulte un maillage isotrope, qui ne convient donc pas à des géométries présentant des variations d'échelle très fortes.

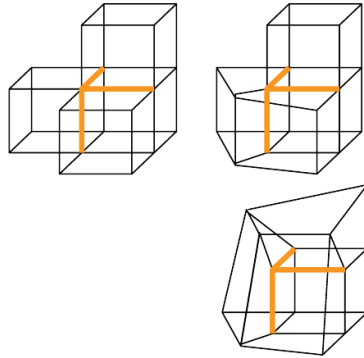


FIGURE 6.2: Exemple de contraction d’hexaèdres dans le cadre des travaux de Walton *et al.* [WBS02].

Quelques algorithmes permettent de générer aussi bien des hexaèdres que des tétraèdres, selon le choix de l’utilisateur. Zhang *et al.* [ZBS05] ont proposé un tel algorithme, afin de créer des maillages 3D à partir de scans 2D (CT, IRM, ...). Ils étendent la méthode de *dual contouring* [JSW02] proposée par Ju *et al.*, afin d’ajouter une étape d’extraction d’isosurface. Cette méthode consiste à analyser les arêtes ayant un sommet de chaque côté de l’isosurface extraite. Alors que Ju *et al.* utilisent une fonction d’erreur quadratique, Zhang *et al.* ont modifié cette métrique d’erreur, introduisant une sensibilité aux caractéristiques du maillage, obtenant de meilleurs résultats.

Maillages quadrangulaires

Dans les travaux s’approchant de notre objectif, à savoir réaliser le maillage isotropique quadrangulaire d’une forme dans le plan, les travaux de Pellenard *et al.* [PAM11] sont remarquables. Les auteurs proposent une méthode entièrement automatique, basée sur plusieurs optimisations successives, s’attachant d’une part à améliorer un point précis du maillage (conformité au champ de taille ou à l’orientation, nombre de sommets extraordinaires, ...) et d’autre part à ne pas déteriorer les acquis précédents.

Enfin, il est intéressant de noter l’approche de Bronson *et al.* [BLW10], qui utilisent une approche particulière pour la génération de maillages triangulaires et tétraédriques, ajustant notamment l’interaction entre particules à l’épaisseur du volume, permettant un résultat anisotrope, capturant la géométrie initiale.

6.2 Notre approche

6.2.1 Principe

Des triangles aux quadrangles

La question structurante de notre approche est la suivante : est-il possible de générer une distribution de points telle que, une fois la triangulation (de Delaunay, par exemple) de ces points effectuée, une règle simple permette de supprimer un certain nombre d'arêtes, éventuellement d'en ajouter certaines, nous permettant d'obtenir un maillage ne contenant que des quadrangles ? Une première idée est d'utiliser une distribution de points contenant des objets de deux natures différentes, de sorte que, si on ne conserve de la triangulation que les arêtes reliant des points de nature différente, alors on obtiendrait des polygones à $2n$ côtés, découpables en quadrangles à l'aide de subdivision à partir du point médian, par exemple (fig.6.3). Il se trouve que de tels arrangements existent dans la nature : ce sont les *cristaux ioniques diatomiques*. Notre approche consiste ainsi à exploiter les lois physiques régissant ces phénomènes dans le but de développer de nouveaux types de mailleurs. Dans la suite de ce chapitre une particule va correspondre à un sommet du maillage.

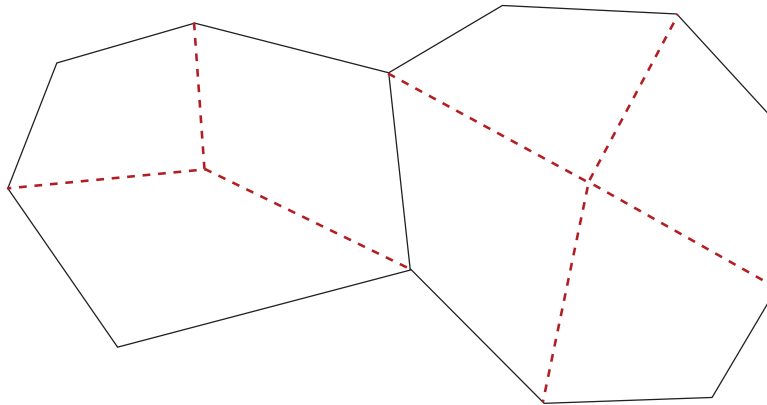


FIGURE 6.3: En noir continu, les $2n$ -gones originaux, en rouge pointillé les nouvelles arêtes permettant de créer des quadrangles.

Cristaux ioniques diatomiques

Les cristaux ioniques, dont l'exemple le plus familier est le chlorure de sodium (NaCl) ou sel de cuisine, sont, dans leur état solide, des arrangements d'atomes sous forme ionique reliés entre eux par des liaisons ioniques. Nous nous intéressons ici

au cas des cristaux diatomiques, c'est-à-dire formés de deux types d'ions monoatomiques. Pour chacune des particules de notre simulation, nous allons considérer que trois types d'interactions régissent leurs mouvements : l'interaction électrique, l'interaction atomique, l'agitation thermique.

L'interaction électrique

Elle est donnée par l'énergie électrostatique d'interaction. En effet, même si les particules sont amenées à bouger lors de notre processus, c'est bien une position d'équilibre statique que nous recherchons. L'énergie d'interaction entre une charge q_1 placée en un point \mathbf{r}_1 dans le champ électrique créé par une charge q_2 placée en un point \mathbf{r}_2 est donnée par

$$E_{int} = \frac{q_1 q_2}{4\pi\epsilon_0} \frac{1}{\|\mathbf{r}_2 - \mathbf{r}_1\|} \quad (6.3)$$

avec $\epsilon_0 \cong 8.85e^{-12} F.m^{-1}$ la permittivité du vide. Cette loi indique notamment que deux particules de charges identiques se repoussent tandis que deux particules de charges opposées s'attirent. C'est cette interaction qui régit la position des ions à l'équilibre dans le cristal, chacun des ions Na^+ étant attiré par ses 6 voisins Cl^- , et inversement, la figure 6.4 illustrant les positions résultantes des ions dans le cristal.

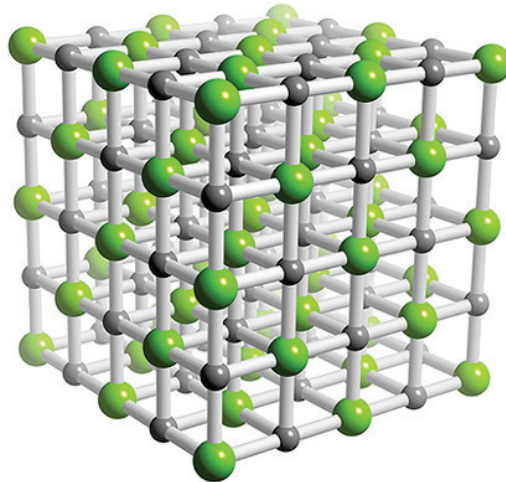


FIGURE 6.4: Cristal de NaCl (www.3dchem.com). Les sphères vertes représentent les ions Cl^- , les grises les ions Na^+ .

Le principe d'exclusion de Pauli

Cette règle interdit à deux électrons de se trouver dans le même état quantique. L'effet de ce principe est que, à très courte distance, les atomes se comportent

comme des sphères solides dont le rayon est à peu près égal à la distance de l'orbite de leur dernière couche électronique. Pour rendre compte de ce principe, nous ajoutons au potentiel électrostatique en $1/r$ une force de répulsion à courte distance en $1/r^5$. L'exposant est purement empirique, et ne sert ici qu'à rendre compte du phénomène tout en étant numériquement stable.

$$E_{ex} = \frac{\mathbf{r}_2 - \mathbf{r}_1}{\|\mathbf{r}_2 - \mathbf{r}_1\|^6} \quad (6.4)$$

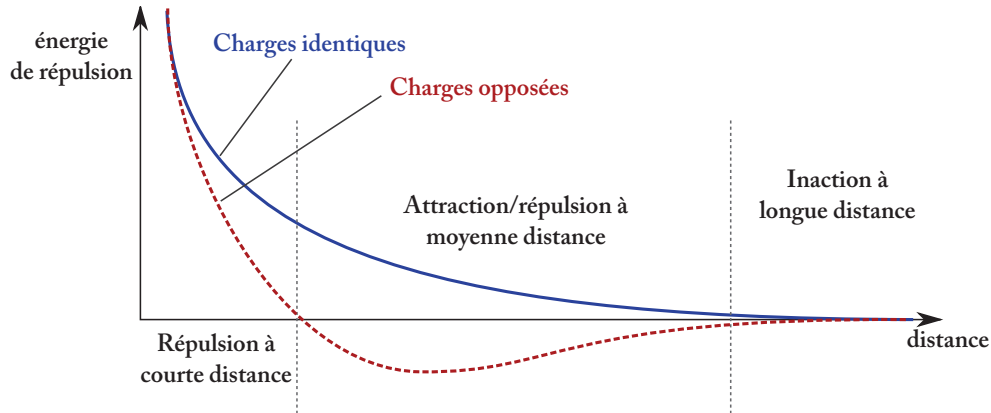


FIGURE 6.5: Graphe représentant l'allure de l'évolution du potentiel d'attraction/répulsion entre deux particules en fonction de la distance qui les séparent et de leur nature.

L'agitation thermique

Elle est simulée à l'aide d'un mouvement brownien, dont l'amplitude est proportionnelle à un paramètre assimilé à une température. Au cours de la simulation, la température va progressivement décroître, laissant moins d'opportunité aux particules de franchir les barrières énergétiques créées par l'agencement de leur voisins.

$$E_{th} = rand() \cdot \frac{T}{T_0} \quad (6.5)$$

6.2.2 Application en 2D

Nous modélisons les lois précédemment énoncées sous la forme d'un système de particules 2D dans notre programme, c'est-à-dire que l'on cherche à mailler une zone du plan représentée par l'ensemble des pixels blancs d'une image binaire, définissant ainsi les frontières du domaine de calcul. L'ensemble des *ions* manipulés vont donc se déplacer sur cette surface afin d'en réaliser un maillage quadrangulaire.

Le déplacement des particules s'effectue en deux étapes. On cherche dans un premier temps à minimiser la somme des énergies décrites plus haut, puis on applique à cette nouvelle position le déplacement lié à l'agitation thermique.

La minimisation des énergies se fait en allant chercher dans un voisinage proche une position permettant de diminuer l'énergie courante d'un point \mathbf{p} . Pour cela, une distance de déplacement d est définie, faible devant la distance aux plus proches voisins (en pratique de l'ordre de 1% de la distance au deuxième ou troisième voisin, plus stable que la distance au premier voisin), puis le cercle de centre \mathbf{p} et de rayon d est échantillonné en n positions (en pratique, $n = 16$ ou 32).

Plus d diminue, plus le point va pouvoir aller avec précision vers un minima local, mais plus la simulation sera lente. L'effet inverse est observé avec n . Pour chacune des n nouvelles positions, l'énergie du point est calculée, et le point sera alors déplacé à la nouvelle position optimale, d'énergie minimale parmi cet échantillonnage du plan.

Après cette minimisation, un mouvement aléatoire est ajouté au déplacement éventuel de la particule. Il s'agit encore une fois d'une fraction de la distance au deuxième ou troisième plus proche voisin, multipliée par l'énergie thermique définie ci-dessus, permettant aux particules de franchir les barrières de potentiel formées par leurs voisins, afin de se placer dans un autre minimum local.

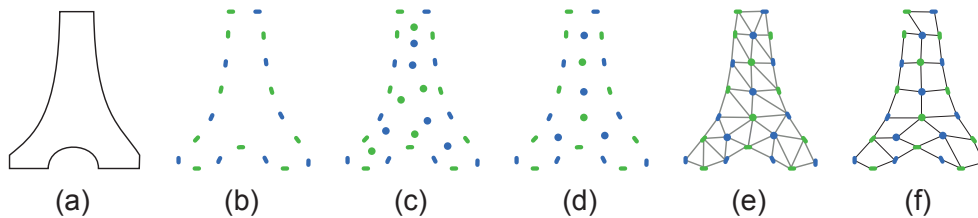


FIGURE 6.6: Les différentes étapes de notre méthode, de gauche à droite : (a) image initiale, (b) maillage du contour, (c) initialisation en bruit bleu multi-classe, (d) évolution des particules, (e) triangulation de Delaunay et (f) quadrangulation

6.3 Notre méthode

L'algorithme dans sa globalité fonctionne sur le plan suivant :

1. Mailler le contour (6.6.b)
2. Initialiser la position des points intérieurs (6.6.c)

3. Calculer le mouvement des points (6.6.d)
4. Réaliser la triangulation de Delaunay de l'ensemble des points, puis la quadrangulation (6.6.e et 6.6.f)

Nous allons à présent détailler chacune de ces étapes.

6.3.1 Maillage du contour

Principe

Cette étape n'était pas présente dans les premiers essais que nous avons effectués, mais il nous est apparu que les résultats sont de bien meilleure qualité en incluant cette phase. Elle consiste à placer sur le périmètre de la zone à mailler ces ions, en alternance. Le résultat de cette étape est que, lorsque dans l'étape 3, un ion s'approchera du bord de la zone à mailler, celui-ci sera *guidé* par le champ créé par ces ions de bord.

Cette étape est réalisée à l'aide d'une détection de contour de type *snake* [KWT88] : les ions sont initialisés comme une chaîne d'éléments disposés en cercle. Puis, chacune de ces particules est soumise à trois forces : deux forces internes, de cohésion, et une force externe, d'attache aux données. Le but des forces internes est de conserver l'aspect de chaîne des éléments, de faire en sorte que le contour dessiné soit régulier et les sommets répartis de manière équilibrés, tandis que la force d'attache aux données cherche à placer le contour le plus proche possible du bord.

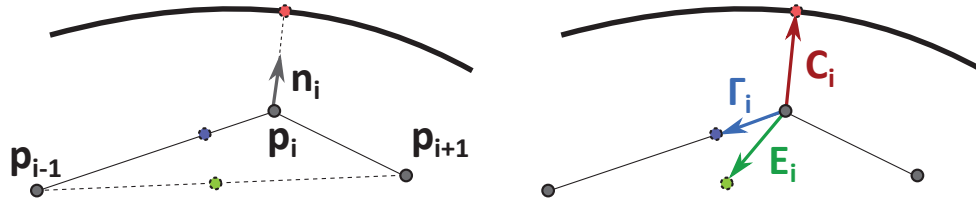


FIGURE 6.7: Méthode d'évolution du contour. Gauche, les points et vecteurs permettant de déterminer les trois forces. Droite : les forces résultantes.

Forces internes

La première force interne, notée Γ , représente la tension de surface de la chaîne d'éléments. En notant \mathbf{p}_i la position du i -ème point de la chaîne, cette force s'exprime comme

$$\Gamma_i = \lambda_\Gamma \left(\frac{\mathbf{p}_{i-1} + \mathbf{p}_{i+1}}{2} - \mathbf{p}_i \right) \quad (6.6)$$

autrement dit, chaque point est amené vers la position moyenne de ses voisins, ce qui a pour effet de lisser la courbe formée des éléments. Le facteur multiplicatif λ_Γ évolue au cours de la détection du contour. Initialement $\lambda_\Gamma = 1$, puis la valeur décroît progressivement au cours de l'évolution du contour, pour permettre d'épouser au mieux la forme à mailler.

La seconde force interne, notée \mathbf{E} , modélise l'effort d'équirépartition des points le long du contour. Avec les notations exposées précédemment, ainsi qu'une distance $d(.,.)$, elle s'écrit comme

$$\mathbf{E}_i = \lambda_E (\mathbf{p}_m - \mathbf{p}_i) \frac{\min(d(\mathbf{p}_i, \mathbf{p}_{i-1}), d(\mathbf{p}_i, \mathbf{p}_{i+1}))}{d(\mathbf{p}_i, \mathbf{p}_{i-1}) + d(\mathbf{p}_i, \mathbf{p}_{i+1})} \quad (6.7)$$

où $m = i - 1$ si $d(\mathbf{p}_i, \mathbf{p}_{i-1}) > d(\mathbf{p}_i, \mathbf{p}_{i+1})$, $m = i + 1$ sinon. Le facteur λ_E décroît également au cours de la simulation, pour les raisons exposées précédemment.

Forces externes

En chaque point, on définit une normale \mathbf{n}_i telle que

$$\mathbf{n}_i = \frac{(\mathbf{p}_i - \mathbf{p}_{i-1}) + (\mathbf{p}_{i+1} - \mathbf{p}_i)}{2} \quad (6.8)$$

Puis, pour chaque point, on cherche l'intersection \mathbf{c}_i avec le contour, dans la direction de la normale. La force externe \mathbf{C} est alors définie par

$$\mathbf{C} = \lambda_C (\mathbf{c}_i - \mathbf{p}_i) \quad (6.9)$$

où le coefficient λ_C augmente au cours de la simulation. En effet, il est préférable d'approcher grossièrement dans un premier temps le contour avec une chaîne régulière et structurée, puis ensuite seulement de relâcher les contraintes internes et d'augmenter la force d'attache aux données.

6.3.2 Initialisation

Dans un premier temps, nous utilisons une initialisation simple, parfaitement aléatoire. Cependant, nous avons réalisé une autre méthode, plus longue à exécuter lors de l'exécution, mais réduisant drastiquement le temps de convergence de l'étape 3. Basée sur les travaux de Wei [Wei10], nous réalisons une distribution multi-classe possédant un spectre de bruit bleu, autrement dit imposant une distance minimale entre chaque échantillon. Chacune des classes possède séparément cette propriété, et l'ensemble des deux classes la possède également, à une échelle différente.

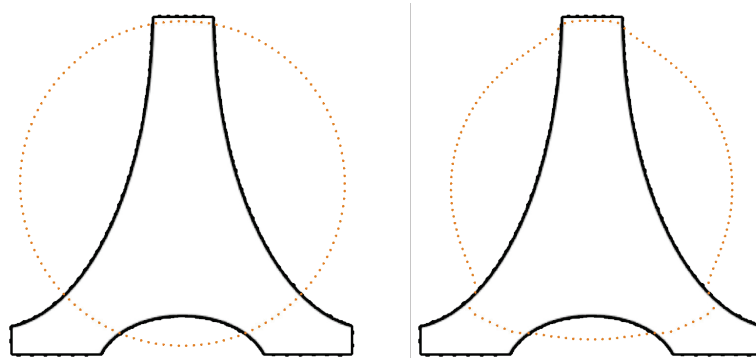


FIGURE 6.8: Gauche : première étape du maillage du contour. Droite : évolution après quelques itérations.

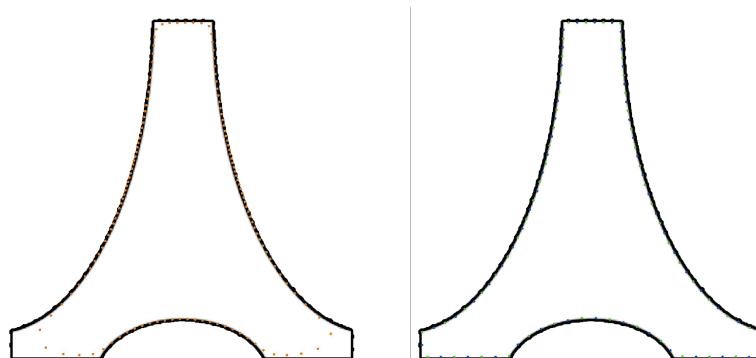


FIGURE 6.9: Gauche : le contour épouse presque l'objet à mailler. Droite : les points du contour sont initialisés comme des charges positives et négatives, en alternance.

6.3.3 Mouvement des points

Comme décrit dans la section 6.2.2, le mouvement des points se fait en deux étapes : une minimisation, puis un mouvement aléatoire dont l'intensité décroît au cours du temps. Ces deux étapes sont réalisées jusqu'à une convergence relative : lorsque le mouvement global des points passe sous un seuil pré-défini, la simulation s'arrête. Pour des raisons évidentes de temps de calcul, il est hors de question de calculer l'influence de tous les ions de la simulation sur un élément donné. Par conséquent, une structure de kd -tree est construite avant de calculer les contributions, puis les n plus proches voisins du point sont déterminés, dont les seules contributions seront prises en compte. En pratique, $n = 10$ fonctionne bien.

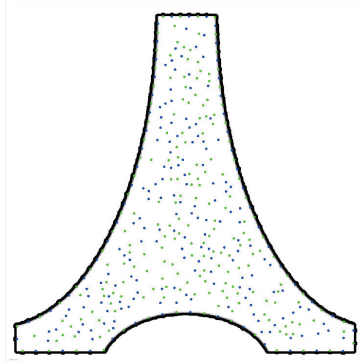


FIGURE 6.10: Position de départ des points, à la sortie du processus d'initialisation en bruit bleu.

6.3.4 Triangulation et quadrangulation

Nous réalisons ensuite une triangulation de Delaunay de l'ensemble des points, indifféremment de leur nature d'ion positif ou négatif. Cette triangulation est ensuite épurée de deux types d'arêtes successivement : les arêtes reliant deux points de même nature, et les arêtes de longueur trop importante. La première de ces phase de suppression est au cœur de notre algorithme, puisqu'elle assure la formation de quadrangles en majorité, de $2n$ -gones dans certains cas. La seconde suppression nous permet de éviter certains quadrangles trop disproportionnés.

6.4 Résultats

6.4.1 Résultats dans le plan

Les figures 6.8 et 6.9 présentent le processus d'initialisation du contour de la forme que nous cherchons à mailler. Les figures 6.10 à 6.13 présentent l'évolution de la position des points au cours du processus d'optimisation, ainsi que les quadrangulations résultantes.

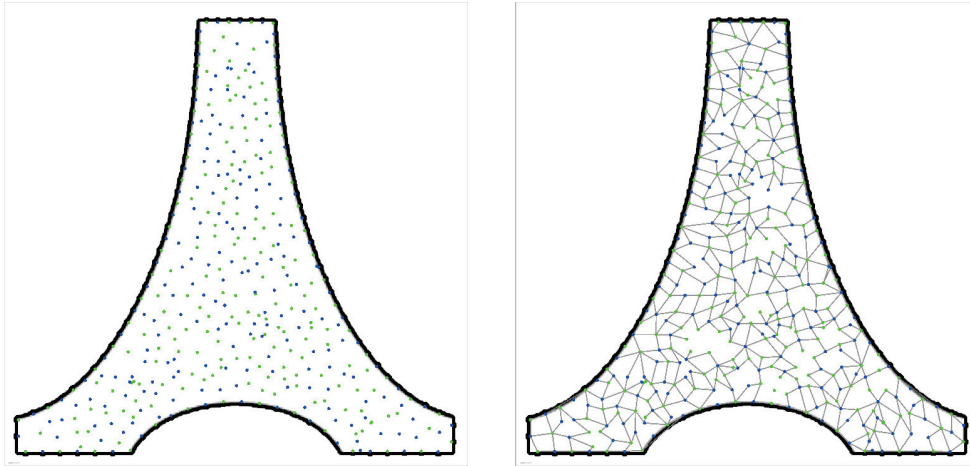


FIGURE 6.11: Position des points après quelques itérations. On constate que les positions sont encore très chaotiques.

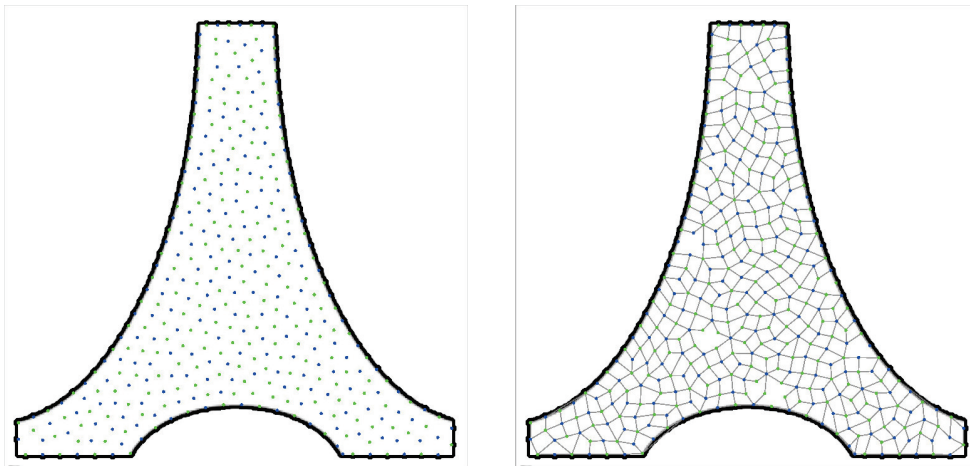


FIGURE 6.12: Etat intermédiaire du système. On constate que les distances inter-sommets sont dès à présent très isotropes.

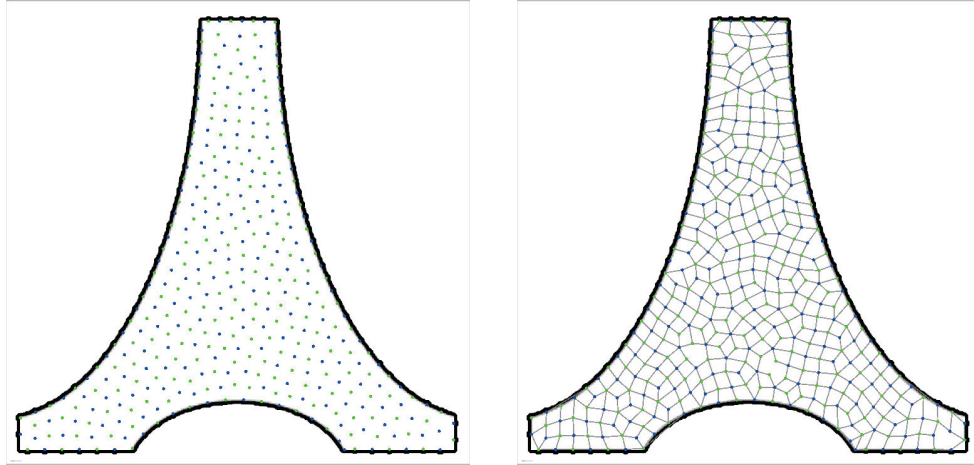


FIGURE 6.13: Etat final du système.

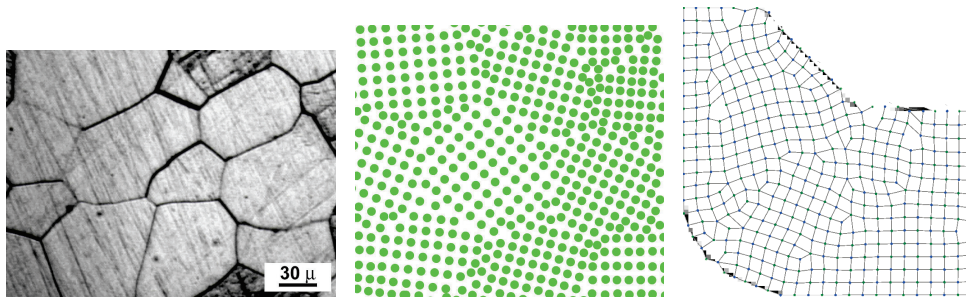


FIGURE 6.14: Gauche : un cristal métallique ($Ti_5Al_5Mo_5V_1,5Cr$). Centre : modèle de polycristal à l'échelle atomique. Droite : notre résultat. *source gauche et centre : wikipedia.*

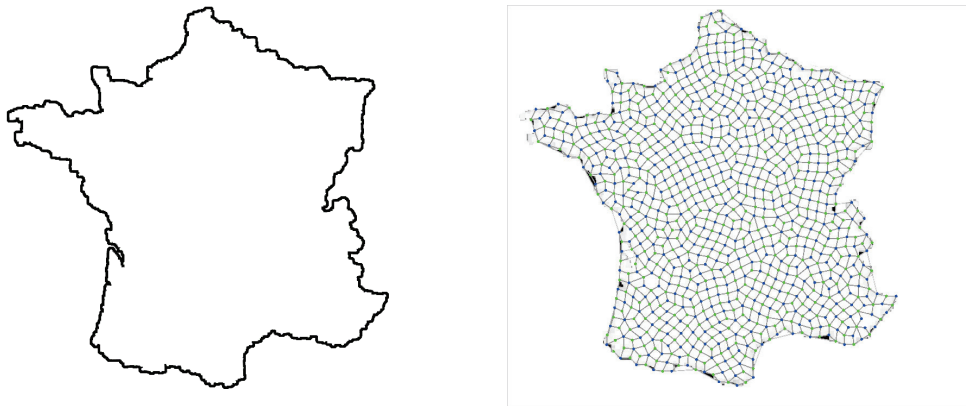


FIGURE 6.15: Gauche : image de départ. Droite : quadrangulation finale.

6.4.2 Analyse

Analyse physique

Une analyse visuelle simple nous permet de constater que les maillages sont constitués de zones maillées quasi-parfaitement (*grains*), avec des points bien alignés, et une valence de 4, et, entre ces zones, des aires de bien moindre qualité, faisant la jonction entre les différentes zones quadrillées, appelées *joint de grain*. Ce résultat, loin d'être surprenant, est parfaitement conforme à ce que l'on peut attendre d'un tel processus. La figure 6.14 illustre bien cette similarité.

Analyse fréquentielle

Afin de vérifier qu'un ordre (au sens physique, entropique du terme) est instauré, nous avons décidé d'utiliser la transformée de Fourier afin de caractériser la distribution en fréquence des points composant le maillage. Les figures 6.16 et 6.17 présentent les distributions de points lors de la phase d'initialisation en bruit bleu et dans l'état final, et les spectres de puissance associés. On note l'apparition d'un ordre à moyenne distance à la fin de notre algorithme, ce qui nous permet de mettre en évidence une structure intrinsèque des sommets du maillage, avant même l'ajout d'une topologie.

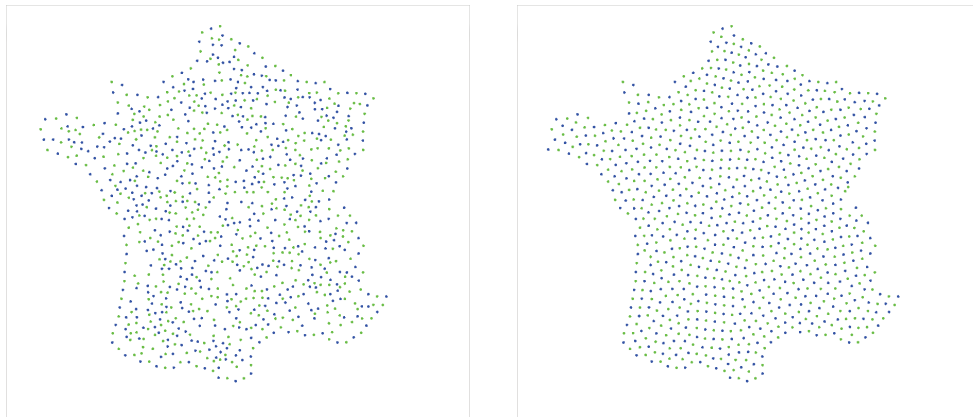


FIGURE 6.16: Distribution des points après l'initialisation (gauche) et dans l'état final du système (droite).

Analyse topologique

Nous avons également souhaité effectuer l'analyse de la valence des sommets de notre maillage. Sur l'exemple présenté dans les figures 6.8 à 6.13, nous avons étudié l'évolution du nombre de voisins au cours du temps, pour les sommets internes

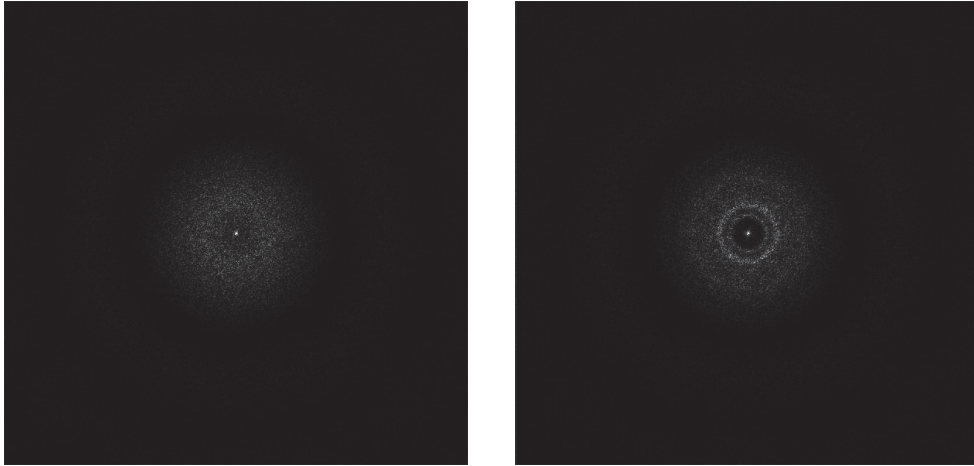


FIGURE 6.17: Spectre de puissance des distributions de la figure 6.16. On note l'apparition d'un ordre à moyenne distance sur le spectre du résultat final (droite) absent sur le spectre après l'initialisation (gauche)..

comme pour les sommets de bord. Les figures 6.18 et 6.19 présentent les résultats. Il est intéressant de noter la très forte proportion (plus de 75%) de sommets de valence 3, ce qui est très probablement dû à notre initialisation imposant une alternance parfaite de charges positives / négatives sur le contour.

Nous remarquons en revanche une proportion plus modeste de sommets internes de valence 4 (environ 55%), le reste étant principalement composé de sommets de valence 3 (25%) et 5 (17%). Ceci signifie d'une part qu'il existe quelques sommets de valences 2 ou 6 (ou plus), et d'autre part qu'en dehors de zones régulières, les joints de grains sont composés presque exclusivement de sommets extraordinaires. On note également le fort taux de sommets de valence 2 au départ du processus, celui-ci diminuant très rapidement pour atteindre un seuil après quelques 500 itérations. On peut imaginer un arrêt automatique de la méthode en fonction des taux des différentes valences.

6.4.3 Résultats en 3D

La figure 6.20 présente un exemple de génération dans l'espace (sans l'étape de détection de contour).

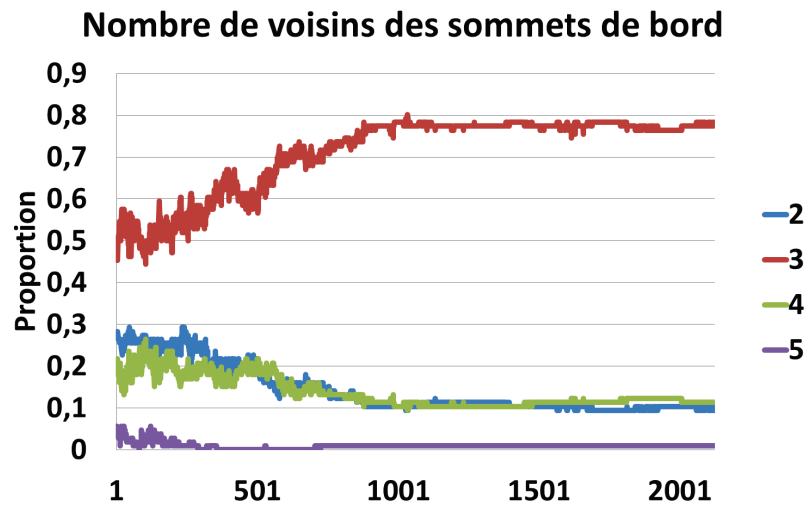


FIGURE 6.18: Evolution de la valence des sommets de bord au cours de la formation du maillage.

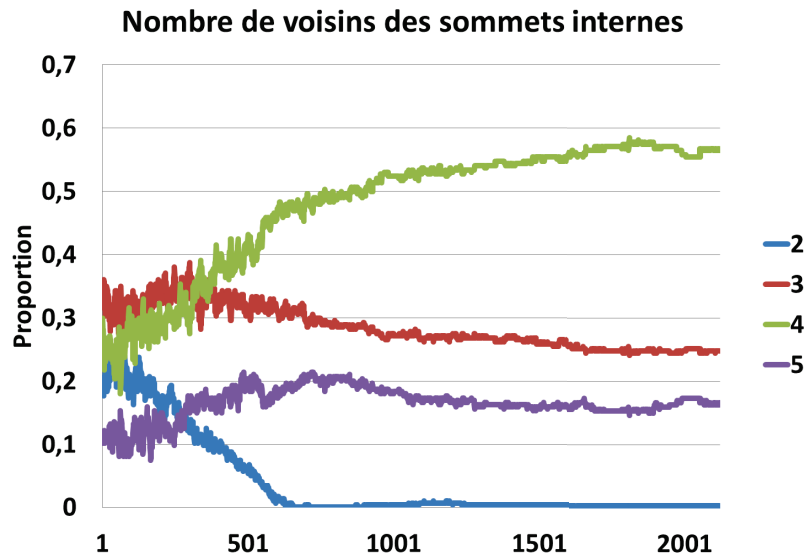


FIGURE 6.19: Evolution de la valence des sommets internes au cours de la formation du maillage.

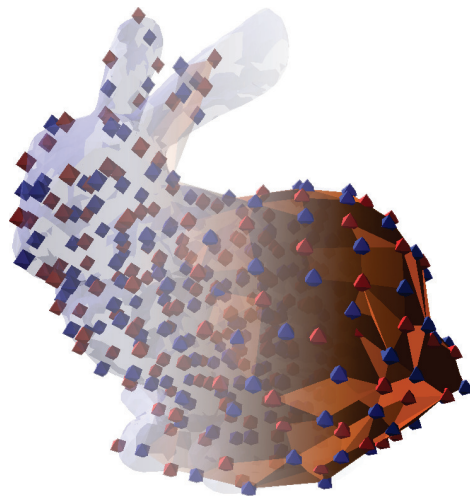


FIGURE 6.20: Exemple de résultat obtenu dans l'espace.

6.4.4 Perspectives

Il nous semble que le résultat pourrait être amélioré significativement à bas coût en ajoutant une étape de nettoyage du maillage, composé d'opérations topologiques locales et ciblées de type contraction d'arêtes ou duplication de sommets.

En vue d'utiliser l'algorithme dans \mathbb{R}^3 , il pourrait être intéressant de partir d'un maillage quadrangulaire existant, de donner une charge à tous les points de surface, pour ensuite utiliser exactement le processus employé dans le plan pour mailler l'intérieur du volume avec des hexaèdres.

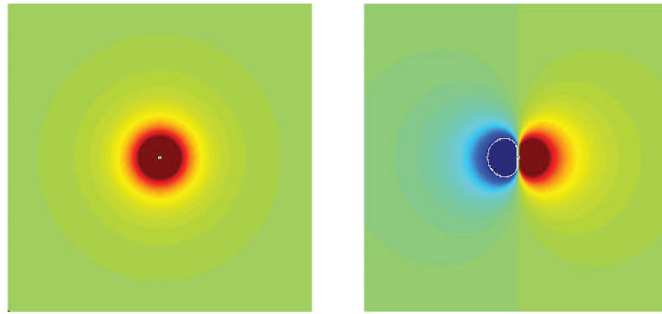


FIGURE 6.21: Gauche : champ généré par un monopôle électrique. Droite : champ créé par un dipôle.

En-dehors de l'aspect purement dimensionnel du problème, d'autres questions sont soulevées. Nous pouvons nous demander si des systèmes plus complexes peuvent être imaginés, en considérant, par exemple, les points comme des dipôles électriques au lieu de monopôles (fig. 6.21), ou bien encore si des interactions avec des points de trois, quatre familles différentes sont envisageables. D'autres règles topologiques nous permettraient peut-être également une quadrangulation aisée à partir d'une triangulation, et cette technique pourrait servir à l'obtention d'autres sortes de polygones.

Discussion

Au cours de ce chapitre, nous avons vu comment il était possible, en combinant la triangulation de Delaunay avec l'évolution d'un système particulière, de générer, dans le plan ou dans l'espace, des maillages quadrangulaires. Autrement dit, à partir de la seule donnée de la géométrie de surface, nous avons créé toutes les structures du maillage, géométriques comme topologiques. Il est à noter cependant qu'il n'existe pas de garantie de la validité du maillage final.

Chapitre 7

Conclusion

Au cours de cette thèse, nous avons pu aborder bien des aspects du traitement de maillages : depuis le filtrage et le lissage des surfaces, jusqu'à la création de maillages volumiques, en passant par le remaillage local pour créer des zones fortement anisotropes et l'optimisation de modèles hexaédriques. Nous rappelons dans cette partie les différents apports techniques proposés par cette thèse, et concluons, après la présentation des notions marquantes de ces travaux, en proposant de nouvelles perspectives de travail dans le cadre du traitement de maillages pour la simulation et le calcul parallèle.

7.1 Rappel des contributions

7.1.1 Parallélisation d'algorithmes

Nous avons commencé, au travers de deux exemples précis, par démontrer que la parallélisation d'un algorithme, si elle peut se présenter sous bien des formes, implique une grande rigueur si l'on souhaite préserver la qualité et le déterminisme d'un algorithme. Ceci peut aller du choix de quelques règles simples et arbitraires permettant de résoudre les conflits aux interfaces à l'approximation locale d'une fonction globale, permettant un résultat correct sinon exact.

Ces modestes contributions vont se retrouver dans tous les travaux exposés par la suite. Si les algorithmes n'ont pas été implémentés dans une architecture parallèle, il nous a fallu garder à l'esprit que c'est une évolution que nous souhaitons voir aboutir, et qu'il nous fallait dès leur conception envisager des techniques qui soient applicables ou adaptables en parallèle de façon aisée.

7.1.2 Filtrage surfacique

Nous avons exposé dans le chapitre 3 comment un filtre local sur la surface d'un maillage pouvait être accéléré de façon indépendante de l'architecture logicielle,

en composant deux filtres de supports de petite aire pour obtenir, avec un gain significatif de vitesse, des résultats semblables à ceux d'un filtrage classique. Nous avons également vu comment les histogrammes locaux, et l'étude de la médiane géométrique tout particulièrement, permettaient une approche différente du lissage de maillage.

La séparabilité du filtrage bilatéral montre comment il est possible d'accélérer intrinsèquement un algorithme, cette accélération pouvant être combinée avec une parallélisation, ou une adaptation des structures de données. De même, les propriétés de la médiane géométrique laisse entrevoir un algorithme efficace sur un support relativement restreint, permettant à nouveau une réalisation rapide.

7.1.3 Optimisation hexaédrique

Le chapitre 4 nous a permis d'exposer deux techniques d'optimisation de maillages volumique, dans le cas particulier où les cellules sont des hexaèdres, ou plus généralement composées de quadrangles. Nous avons montré comment il était possible de réduire le gauchissement des faces de façon significative, et par quel moyen il était possible de filtrer un maillage volumique en conservant les caractéristiques, à l'aide de l'analyse en composantes principales.

Les approches proposées sont très locales, et pour cause : la structure même d'un maillage hexaédrique conforme impose beaucoup sur la topologie de celui-ci, et les algorithmes récents permettant une bonne optimisation se basent sur le dual de ce maillage, travaillant sur les cordes et les feuilles du *Spatial Twist Continuum*, structures définies à l'échelle globale du maillage, et dont la manipulation dans le cadre d'un maillage distribué en mémoire n'est pas immédiate, mais pourrait constituer une famille d'approches intéressantes.

7.1.4 Création de couches limites

Le chapitre 5 nous a permis de mettre en avant un algorithme de remaillage local, dans le but d'adapter un maillage isotrope à un calcul de mécanique des fluides numérique. Il est à noter que, pour que le maillage final soit d'une qualité suffisante, notamment en termes de raffinement, il est nécessaire que le maillage initial soit également d'une qualité certaine. Ceci met en avant le fait que la qualité d'un maillage est intimement liée à l'utilisation qui en sera faite, et qu'un maillage ne peut être décrit comme intrinsèquement *bon* ou *mauvais*.

Si cet algorithme donne des résultats plutôt convaincants, il est à noter que le maillage intermédiaire, délimité par la *sous-surface* peut se révéler de piètre qualité, et il pourrait être intéressant de se pencher, lors du choix de mailleur destiné à la création du maillage initial, sur la qualité de la sous-surface générée.

7.1.5 Maillages cristallins

Dans le chapitre 6, nous avons montré une technique particulière s'appuyant sur les structures cristallines naturelles permettant de créer des maillages quads-dominants 2D ou 3D. L'idée maîtresse de cette méthode réside dans la combinaison de trois éléments : une initialisation en bruit bleu multi-classe, une modélisation efficace des mouvements ioniques, et des règles de suppression des arêtes. La quadrangulation finale n'est d'abord qu'un sous-ensemble de la triangulation de Delaunay de l'ensemble des points, et c'est sur la position et la nature des points que le choix des arêtes à conserver ou supprimer va s'appuyer.

7.2 Notions émergentes

Il ressort de nos travaux qu'il est souvent utile de plonger le maillage dans un espace de dimensions supérieure pour y effectuer des opérations, avant de revenir dans l'espace initial. Dans le cadre du filtrage bilatéral, qu'il s'agisse de l'appliquer à une surface ou bien à un volume, l'espace spatial est augmenté d'un espace des caractéristiques, ou structures de l'objet. De la même façon, on peut voir dans les maillages cristallins l'introduction d'un espace des charges, lequel est utilisé pour calculer le mouvement des particules.

De nos travaux sur les maillages cristallins, il ressort que la physique, et les phénomènes naturels qui nous entourent, sont de bonnes sources d'inspiration pour des techniques de traitement de maillages. Parce qu'il sont été créés à partir des lois naturelles, les objets résultants nous apparaissent moins artificiels, ou avec moins d'artefacts que ceux fabriqués à partir de techniques plus abstraites.

7.3 Perspectives

7.3.1 Structures de données

Parmi les travaux futurs portant sur les problématiques soulevées au cours de cette thèse, nous pouvons envisager des structures de données spécifiques aux maillages distribuées, et permettant de faire le lien entre les diverses architectures. En effet, les structures de maillage utilisées aujourd'hui sont adaptées à l'utilisation qui en est faite : aux machines distribuées, aux CPUs, aux GPUs. Il existe des structures optimisées pour une géométrie particulière, ainsi nous pouvons envisager des structures de données adaptées aux maillages et aux architectures.

Il serait intéressant de disposer de structures de données permettant de faire le lien entre les différentes architectures. En effet, on peut souhaiter utiliser différentes architectures au cours d'un calcul (CPU et GPU, GPU et super-ordinateurs,...), et

cette opération, qui demande aujourd'hui beaucoup de précautions, serait grandement facilitée par l'utilisation de structures permettant une "<traduction"> entre différentes architectures (SIMD et MIMD,...), dans le cadre du traitement de maillages.

7.3.2 Communication

Nous souhaitons également disposer de meilleurs outils de communication entre machines dans le cadre du traitement de maillages. En effet, pour le moment, les communications, aussi bien entre clusters via MPI qu'entre cœurs GPUs à l'aide de CUDA, sont complètement aveugles à la nature des données échangées, et n'adaptent pas leur précision au cas en cours.

Si les structures de données sont adaptées à l'utilisation matérielle, elles ne sont pas adaptées spécifiquement aux maillages. Par exemple, lorsqu'un filtrage est effectué sur un maillage partagé, il n'est pas nécessaire de transmettre toute l'information des nouvelles positions des sommets. En effet, les déplacements sont souvent petits devant les dimensions du problème, et il est envisageable de ne transmettre que les informations de variation des positions, réduisant ainsi la taille des données à échanger.

Bibliographie

- [AA09] Marc Alexa and Anders Adamson. Interpolatory point set surfaces-convexity and hermite data. *ACM Trans. Graph.*, 28(2) :20 :1–20 :10, May 2009.
- [AGDL09] Andrew Adams, Natasha Gelfand, Jennifer Dolson, and Marc Levoy. Gaussian kd-trees for fast high-dimensional filtering. In *ACM SIGGRAPH 2009 papers*, SIGGRAPH '09, pages 21 :1–21 :12, New York, NY, USA, 2009. ACM.
- [AW11] Marc Alexa and Max Wardetzky. Discrete laplacians on general polygonal meshes. In *ACM SIGGRAPH 2011 papers*, SIGGRAPH '11, pages 102 :1–102 :10, New York, NY, USA, 2011. ACM.
- [Bar07] Barbara Chapman, Gabriele Jost and Ruud van der Pas. *Using OpenMP*. The MIT Press, 2007.
- [BCM05] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60 – 65, june 2005.
- [BKP⁺10] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Levy. *Polygon Mesh Processing*. AK Peters, 2010.
- [BLW10] JonathanR. Bronson, JoshuaA. Levine, and RossT. Whitaker. Particle systems for adaptive, isotropic meshing of cad models. In *Proceedings of the 19th International Meshing Roundtable*, pages 279–296. Springer Berlin Heidelberg, 2010.
- [Bor97a] Borouchaki, Houman and George, Paul Louis and Hecht, Frédéric and Laug, Patrick and Saltel, Eric. Delaunay mesh generation governed by metric specifications. Part I Algorithms. *Finite Elem. Anal. Des.*, 25(1-2) :61–83, March 1997.
- [Bor97b] Borouchaki, Houman and George, Paul Louis and Mohammadi, Bijan. Delaunay mesh generation governed by metric specifications. Part II. Applications. *Finite Elem. Anal. Des.*, 25(1-2) :85–109, March 1997.

- [Cas95] Castro-Diaz, M. and Hecht, F. and Mohammadi, B. New Progress in Anisotropic Mesh Adaptation for Inviscid and Viscous Flow Simulations. *4th International Meshing Roundtable*, 1995.
- [CPD07] Jiawen Chen, Sylvain Paris, and Frédo Durand. Real-time edge-aware image processing with the bilateral grid. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.
- [CTS98] Scott A. Canann, Joseph R. Tristano, and Matthew L. Staten. An approach to combined laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes. In *INTERNATIONAL MESHING ROUNDTABLE*, pages 479–494, 1998.
- [DEJ⁺09] Volodymyr Dyedov, Daniel Einstein, Xiangmin Jiao, Andrew Kuprat, James Carson, and Facundo del Pin. Variational generation of prismatic boundary-layer meshes for biomedical computing. *International Journal for Numerical Methods in Engineering*, 79(8) :907–945, 2009.
- [DMSB99a] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of arbitrary meshes using diffusion and curvature flow. *ACM SIGGRAPH Proceedings*, 1999.
- [DMSB99b] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 317–324, 1999.
- [Don95] D.L. Donoho. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3) :613–627, May 1995.
- [FDCO03] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 950–953, 2003.
- [Fly72] Michael J. Flynn. Some computer organizations and their effectiveness. *IEEE Trans. Comput.*, 21(9) :948–960, September 1972.
- [FOg97] Lori A. Freitag and Carl Ollivier-gooch. Tetrahedral mesh improvement using swapping and smoothing. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, 40(21) :3979–4002, 1997.
- [FP] Lori A. Freitag and Paul Plassmann. Local optimization-based simplicial mesh untangling and improvement. *International Journal of Numerical Methods in Engineering*, 49 :109–125.
- [Fuj95] Koji Fujiwara. Eigenvalues of laplacians on a closed riemannian manifold and its nets. *Proceedings of AMS*, pages 2585–2594, 1995.
- [Gar98] Garimella, Rao. *Anisotropic Tetrahedral Mesh Generation*. PhD thesis, Rensselaer Polytechnic Institute, 1998.

- [Geo98] George, Paul Louis and Borouchaki, Houman. *Delaunay Triangulation and Meshing : Application to Finite Elements*. Hermes, 1998.
- [GS04] R. V. Garimella and M. J. Shashkov. Polygonal surface mesh improvement. *Engineering with Computers*, 20 :265–272, 2004.
- [Has96] Hassan, O. and Probert, E. and Morgan, K. and Peraire, J. Unstructured Tetrahedral Mesh Generation for Three-Dimensional Viscous Flows. *International Journal for Numerical Methods in Engineering*, 39 :549–567, 1996.
- [IN02] Yasushi Ito and Kazuhiro Nakahashi. Unstructured mesh generation for viscous flow computations. In *Proceedings of the 11th International Meshing Roundtable*, pages 367–377, 2002.
- [JDD03] Thouis R. Jones, Frédo Durand, and Mathieu Desbrun. Non-iterative, feature-preserving mesh smoothing. *ACM SIGGRAPH Proceedings*, 2003.
- [JDZ04] Thouis R. Jones, Fredo Durand, and Matthias Zwicker. Normal improvement for point rendering. *IEEE Comput. Graph. Appl.*, 24 :53–56, July 2004.
- [JSW02] T. Ju, S. Schaefer, and J. Warren. Dual contouring of hermite data. *ACM SIGGRAPH Proceedings*, pages 339–346, 2002.
- [Kal93] Kallinderis, Y. and Ward, S. Prismatic Grid Generation for Three-Dimensional Complex Geometries. *AIAA Journal*, 31(10) :1850–1856, 1993.
- [Kal95] Kallinderis, Y. and Khawaja, A. and McMorris, H. Hybrid Prismatic/Tetrahedral Grid Generation for Complex 3-D Geometries. *AIAA-95-0211*, 1995.
- [Kar07] S. L. Karman. Unstructured viscous layer insertion using linear-elastic smoothing. *AIAA Journal*, 45(1) :168–190, 2007.
- [Kha95] Khawaja, A. and McMorris, H. and Kallinderis, Y. Hybrid Grids for Viscous Flows Around Complex 3-D Geometries Including Multiple Bodies. *AIAA-95-1685*, 1995.
- [Kov05] Kovalev, Konstantin. *Unstructured Hexahedral Non-conformal Mesh Generation*. PhD thesis, Vrije Universiteit Brussel, 2005.
- [KS10] Michael Kass and Justin Solomon. Smoothed local histogram filters. *ACM Trans. Graph.*, 29 :100 :1–100 :10, July 2010.
- [KWT88] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes : Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4) :321–331, 1988.

- [LL10] Bruno Lévy and Yang Liu. Lp Centroidal Voronoi Tesselation and its applications. *ACM Transactions on Graphics*, 29(4), 2010.
- [Loh95] Lohner, R. Matching Semi-structured and Unstructured Grids for Navier-Stokes Calculations. *AIAA-93-3348-CP*, 1995.
- [LXZ08] Dan Liu, Guoliang Xu, and Qin Zhang. A discrete scheme of laplace-beltrami operator and its convergence over quadrilateral meshes. *Computers and Mathematics with Applications*, (55) :1081–1093, 2008.
- [Mav90] Mavriplis, Dimitri J. Adaptive Mesh Generation for Viscous Flows Using Delaunay Triangulation. *Journal of Computational Physics*, 90(2) :271–291, September 1990.
- [MBBM97] Peter Murdoch, Steven E. Benzley, Ted Blacker, and Scott A. Mitchell. The spatial twist continuum : A connectivity-based method for representing all-hexahedral finite element meshes. *Finite Elements in Analysis and Design*, (28) :137–149, 1997.
- [Mes12] Message Passing Interface Forum. *MPI : A Message-Passing Interface Standard, Version 3.0*. High Performance Computing Center Stuttgart, 2012.
- [OBS02] Y. Ohtake, A.G. Belyaev, and Hans-Peter Seidel. Mesh smoothing by adaptive and anisotropic gaussian filter applied to mesh normals. In *VISION MODELING AND VISUALIZATION*, pages 203–210, 2002.
- [PAM11] Bertrand Pellenard, Pierre Alliez, and Jean-Marie Morvan. Isotropic 2d quadrangle meshing with size and orientation control. In *Proceedings of the 20th International Meshing Roundtable*, pages 81–98. Springer Berlin Heidelberg, 2011.
- [Pao02] Stefano Paoletti. Polyhedral mesh optimization using the interpolation tensor. *International Meshing Roundtable Proceedings*, 2002.
- [PD09] Sylvain Paris and Frédo Durand. A fast approximation of the bilateral filter using a signal processing approach. *Int. J. Comput. Vision*, 81 :24–52, January 2009.
- [Pir94] Pirzadeh, S. Viscous Unstructured Three-Dimensional Grids by the Advancing Layer Method. *AIAA-94-0417*, 1994.
- [PKTD08] Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Frédo Durand. A gentle introduction to bilateral filtering and its applications. In *Class at ACM SIGGRAPH'08*, 2008.
- [PP93] Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2 :15–36, 1993.

- [PSZ01] Jianbo Peng, Vasily Strela, and Denis Zorin. A simple algorithm for surface denoising. In *Proceedings of the conference on Visualization '01*, VIS '01, pages 107–112, 2001.
- [PvV05] T.Q. Pham and L.J. van Vliet. Separable bilateral filtering for fast video preprocessing. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, july 2005.
- [Rus04] Szymon Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Symposium on 3D Data Processing, Visualization, and Transmission*, pages 486 – 493, September 2004.
- [Sha98] Sharov, D. and Nakahashi, K. Hybrid Prismatic/Tetrahedral Grid generation for Viscous Flow Applications. *AIAA Journal*, 36(2) :157–162, 1998.
- [Sor05] Olga Sorkine. Laplacian mesh processing. *Eurographics Proceedings*, 2005.
- [SV03] Suzanne M. Shontz and Stephen A. Vavasis. A mesh warping algorithm based on weighted laplacian smoothing. In *IMR*, pages 147–158, 2003.
- [Tau95] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 351–358, 1995.
- [TM98] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision, ICCV '98*, pages 839–, 1998.
- [WBS02] Kirk S. Walton, Steven E. Benzley, and Jason F. Shepherd. Sculpting : an improved inside-out scheme for all hexahedral meshing. *Sandia*, 2002.
- [Wei88] K. Weiler. The radial edge structure : a topological representation for non-manifold geometric boundary modeling. *Geometric Modeling for CAD Applications*, pages 3–36, 1988.
- [Wei10] Li-Yi Wei. Multi-class blue noise sampling. In *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, pages 79 :1–79 :8. ACM, 2010.
- [Yam11] Yamakawa, Soji and Shimada, Kenji. Automatic All-Hex Mesh Generation of Thin-Walled Solids via a Conformal Pyramid-Less Hex, Prisms and Tet Mixed Mesh. *20th International Meshing Roundtable*, 2011.
- [YBS06] Shin Yoshizawa, Alexander Belyaev, and Hans-Peter Seidel. Smoothing by example : Mesh denoising by averaging with similarity-based weights. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006*, pages 9–, 2006.

- [YOB02] Hirokazu Yagou, Yutaka Ohtake, and Alexander Belyaev. Mesh smoothing via mean and median filtering applied to face normals. In *Proceedings of the Geometric Modeling and Processing — Theory and Applications (GMP'02)*, GMP '02, pages 124–, 2002.
- [YOB03] H. Yagou, Y. Ohtake, and A.G. Belyaev. Mesh denoising via iterative alpha-trimming and nonlinear diffusion of normals with automatic thresholding. In *Computer Graphics International, 2003. Proceedings*, pages 28 – 33, july 2003.
- [ZBS05] Yongjie Zhang, Chandrajit Bajaj, and Bong-Soo Sohn. 3d finite element meshing from imaging data. *Computer Methods in Applied Mechanics and Engineering*, 194(48-49) :5083–5106, 2005.
- [ZBX05] Yongjie Zhang, Rajit Bajaj, and Guoliang Xu. Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. In *In Proceedings, 14th International Meshing Roundtable*, pages 449–468, 2005.
- [ZS00a] Tian Zhou and Kenji Shimada. An angle-based approach to two-dimensional mesh smoothing. In *In Proceedings, 9th International Meshing Roundtable*, pages 373–384, 2000.
- [ZS00b] Denis Zorin and Peter Schröder. Subdivision for modeling and animation. *ACM SIGGRAPH Proceedings*, 2000.