



HAL
open science

Génération de maillages hexaédriques pour des simulations de grandes déformations

David Desobry

► **To cite this version:**

David Desobry. Génération de maillages hexaédriques pour des simulations de grandes déformations. Analyse numérique [cs.NA]. Université de Lorraine, 2023. Français. NNT : 2023LORR0163 . tel-04286668

HAL Id: tel-04286668

<https://theses.hal.science/tel-04286668v1>

Submitted on 15 Nov 2023

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.



**UNIVERSITÉ
DE LORRAINE**

**BIBLIOTHÈQUES
UNIVERSITAIRES**

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact bibliothèque : ddoc-theses-contact@univ-lorraine.fr
(Cette adresse ne permet pas de contacter les auteurs)

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>



Génération de maillages hexaédriques pour des simulations de grandes déformations

THÈSE

présentée et soutenue publiquement le 23 août 2023

pour l'obtention du

Doctorat de l'Université de Lorraine
(mention informatique)

par

David DESOBRY

Président du jury : Emmanuel Jeandel
Rapporteurs : Christian GENTIL
Evgeny NIKULCHEV
Examineur : Simon CALDERAN
Encadrants : Dmitry SOKOLOV
Nicolas RAY
Jeanne PELLERIN

Mis en page avec la classe thesul.

Résumé

Cette thèse se concentre sur le développement de méthodes de maillage hexaédrique adaptées aux simulations de grandes déformations en mécanique non-linéaire. Les méthodes de paramétrisation du domaine basées sur les champs de repère sont utilisées pour générer des maillages hexaédriques de haute qualité alignés sur les bords du domaine. Cependant, lors de grandes déformations, la qualité du maillage peut se dégrader et potentiellement bloquer la simulation. Cette thèse explore l'idée de déterminer une connectivité optimale pour les éléments du maillage en tenant compte des déformations prévues.

En 2D, un pipeline complet est développé pour réaliser ce défi en combinant des travaux antérieurs et des contributions scientifiques. En 3D, des contributions sont apportées pour se rapprocher de cet objectif, notamment en contrôlant les valences de bord des maillages hexaédriques produits à partir de champs de repère. Les différentes parties de la thèse abordent les étapes de la simulation numérique de grandes déformations, les avantages des méthodes de paramétrisation globale, les résultats de simulations sur des maillages industriels 2D, ainsi que des contributions pour améliorer la flexibilité et la robustesse du processus de maillage hexaédrique. L'objectif final est de réduire le temps passé par les ingénieurs à générer un maillage adéquat pour une simulation en prenant en compte des informations a priori sur la déformation à laquelle le maillage et l'objet seront soumis.

Mots-clés : Maillage, quadrilatère, hexaèdre, déformation, simulation.

Abstract

This thesis focuses on the development of hexahedral meshing methods suitable for large deformation simulations in non-linear mechanics. Domain parameterization methods based on frame fields are used to generate high-quality hexahedral meshes aligned with the domain boundaries. However, during large deformations, the mesh quality may degrade and potentially block the simulation. This thesis explores the idea of determining an optimal connectivity for the mesh elements while taking into account the anticipated deformations.

In 2D, a complete pipeline is developed to tackle this challenge by combining previous work and scientific contributions. In 3D, contributions are made to approach this objective, particularly by controlling the boundary valences of hexahedral meshes produced from frame fields. The different parts of the thesis address the steps of large deformation numerical simulations, the advantages of global parameterization methods, the results of simulations on industrial 2D meshes, and contributions to improving the flexibility and robustness of the hexahedral meshing process. The ultimate goal is to reduce the time spent by engineers in generating an adequate mesh for a simulation by considering a priori information on the deformation to which the mesh and the object will be subjected.

Keywords : Meshing, quadrilateral, hexahedra, deformation, simulation.

Remerciements

Tout d'abord, merci à mes directeurs de thèse pour m'avoir permis de réaliser cette thèse. Merci ensuite à Étienne qui m'a encadré pendant toute ma thèse, et avec qui nous avons obtenu le prix du deuxième meilleur article de la conférence SPM 2021 pour notre travail sur les champs de repère non-orthogonaux. Sur ce travail ainsi que beaucoup d'autres, j'ai eu le plaisir de collaborer avec mon co-bureau Yoann avec qui les journées sont passées très vite. Un autre moment marquant de ma thèse était ma participation avec Yoann ainsi que mes voisins de bureau Guillaume et Justine au Google Hashcode 2022. Lors du round qualificatif qui finissait après l'heure de fermeture du LORIA, nous avons passé la dernière heure du concours dehors en 4G pour gagner les quelques places qui nous ont permis d'accéder à la finale. Puis après notre qualification et beaucoup d'entraînement, nous avons eu l'honneur de représenter la France lors de la finale, et sur les plus de 10000 équipes inscrites nous avons terminé 18ème mondial. Ma thèse a cependant également connu des moments difficiles, et je suis très heureux et reconnaissant d'avoir pu compter sur mes amis, ma famille et ma copine pour me soutenir pendant ces périodes. Finalement, je suis très heureux du bilan de ma thèse, que ce soit par les cinq publications ou celles qui n'ont pas abouti mais qui m'ont beaucoup appris de part les retours que j'en ai reçu. Je remercie beaucoup l'INRIA et le LORIA pour m'avoir permis tout cet enseignement et pour m'avoir accompagné jusqu'à la date de soutenance. Enfin, merci au jury de ma soutenance de thèse pour leur travail qui a permis une discussion scientifique très riche.

*On se reposera
quand on aura des hexs*

Table des matières

Introduction	11
--------------	----

Partie I État de l'art de génération de maillage pour réaliser des simulations de grandes déformations	14
--	----

Chapitre 1

Simulation et maillage

1.1 Introduction aux simulations numériques et aux maillages pour la recherche et développement	15
1.1.1 Simulation Numérique : Choix de la méthode	16
1.1.2 Définition et propriétés fondamentales du maillage	17
1.2 Simulation de grandes déformations et impact sur le maillage	18
1.2.1 Maillage et conditionnement de la matrice de rigidité	18
1.2.2 Matrice de rigidité élémentaire pour une déformation hyperélastique	19
1.2.3 Méthode de Newton complète pour calculer les déplacements nodaux	21
1.2.4 Un maillage conforme unique pour toute la simulation	22
1.2.5 Maillage hexaédrique de qualité pour ce type de simulation	22

Chapitre 2

Méthodes de maillage hexaédrique

2.1 Graphe de singularité d'un maillage hexaédrique	25
2.2 Maillage structuré	25
2.3 Maillage non-structuré	27
2.3.1 Méthode octree	27
2.3.2 Méthode d'avancée de front	28
2.4 Maillage semi-structuré	29
2.4.1 Maillage par balayage	29
2.4.2 Maillage via un descripteur médian	30

2.5	Maillage à valence semi-structurée	31
2.5.1	Polycube	31
2.5.2	Champs de repère	32

Chapitre 3

Méthode Cubecover basée sur les champs de repères

3.1	La méthode Quadcover pour la génération de maillage quadrilatère	35
3.1.1	Une méthode provenant de l’infographie	35
3.1.2	Champ de repère orthonormal 2D	37
3.1.3	Intégration avec Quadcover	40
3.1.4	Quantification de la paramétrisation	41
3.2	La méthode Cubecover pour la génération de maillage hexaédrique	43
3.2.1	Champ de repère orthonormal 3D	43
3.2.2	Intégration avec Cubecover	46
3.2.3	Quantification de la paramétrisation	47

Partie II Contribution : Contrôle sur le maillage produit par un champ de repère **49**

Chapitre 4

Champ de repère non orthogonal

4.1	Introduction	51
4.2	Champ de direction de dimension n en tant que polynômes sphériques	52
4.2.1	Représentation d’un seul repère	52
4.2.2	Espace tangent d’un ensemble de directions	53
4.2.3	Champ de polyvecteur polynomial constant par morceaux	54
4.2.4	Degré du polynôme de représentation	54
4.3	Champs de repère en 2D	55
4.3.1	Une direction en tant que polynôme de degré 4	55
4.3.2	Base de Fourier	56
4.3.3	Conception de champs de repère en 2D pour maillages quads	56
4.3.4	Résultats	59
4.3.5	Comparaison avec PolyVector field [DIAMANTI et al. 2015]	60
4.4	Champs de repère 3D	62
4.4.1	Décomposition en harmoniques sphériques	63
4.4.2	Construction de champs de repère 3D avec contraintes	64

4.4.3	Maillage hexaédrique de modèle CAO	65
-------	--	----

Chapitre 5

Champ de repère lisse selon une définition adaptée à la CAO

5.1	Introduction	71
5.2	Calcul de γ	72
5.3	Relaxation de l'orthogonalité	73
5.4	Résultats et applications	74

Chapitre 6

Champ de repère avec contraintes de bords adaptées à la CAO

6.1	Introduction	77
6.2	Calcul des contraintes de bord à partir de valences de bord	78
6.2.1	Énergies	80
6.2.2	Une méthode d'optimisation gloutonne	81
6.2.3	Minimisation d'un vecteur unique	83
6.2.4	Champ de repère lisse avec appariement fixe	83
6.3	Différentes stratégies pour déterminer les valences des arêtes de bord	84
6.3.1	Arête à faible angle dièdre	85
6.3.2	Angles diédraux ambigus	86
6.3.3	Modèles CAO avec valences d'arêtes prescrites par l'utilisateur	86
6.3.4	Modification des graphes de singularité internes	90
6.4	Résultats	90
6.4.1	Comparaisons avec le chapitre 4	90
6.4.2	Cas d'échec	91

Partie III Contribution : Quadcover pour des simulations 2D de grandes déformations **94**

Chapitre 7

Algorithme de quantification 2D rapide, robuste et open source

7.1	Introduction	95
7.2	Notre quantification	98
7.2.1	Simplification du problème	98
7.2.2	Optimisation des degrés de liberté entiers	100
7.2.3	Déplacement de la solution vers le problème initial	102
7.3	Implémentation	103

7.3.1	Optimisation numérique	103
7.3.2	Arêtes caractéristiques et bords	104
7.4	Resultats	104
7.4.1	Performances	105
7.4.2	Qualité	106

Chapitre 8

Génération de maillage adapté à des déformations avec Quadcover
--

8.1	Alternance entre maillage et simulation de déformation	111
8.1.1	Construction du maillage initial	111
8.1.2	Construction d'un maillage adapté à une simulation de déformation	111
8.2	Champ de repère adapté à plusieurs géométries	112
8.2.1	Valences sur le bord adapté à toute les géométries	112
8.2.2	Correspondances de champ de repère à valence sur le bord fixé	114
8.2.3	Plusieurs géométries de champs de repère de mêmes correspondances	114
8.3	Quadcover adapté à plusieurs géométries	114
8.3.1	Transfert des géométries de champ de repère	114
8.3.2	Processus itératif pour rendre la carte valide	115
8.4	Résultats	117
8.4.1	Réussite de la méthode sur les modèles CAO	117
8.4.2	Réussite de la méthode pour les simulations de déformation	117

Conclusion **125**

Bibliographie

Introduction

La génération de maillage est une étape clé de la modélisation et de l'analyse des objets naturels ou manufacturés qui nous entourent. Les maillages représentent ces objets de manière discrète et sont utilisés pour simuler numériquement leur comportement physique. Selon l'application, les méthodes de maillage doivent typiquement respecter des contraintes limitant le type des cellules du maillage, leur connectivité, leur nombre, leurs forme et taille. Ce projet s'intéresse plus particulièrement à la simulation mécanique de matériaux hyperélastiques. Pour ce cas d'application, les maillages quadrilatères en 2D et hexaédriques en 3D sont très largement préférés aux maillages triangulaires et tétraédrique car ces maillages sont plus facilement déformables.

Cependant, la génération automatique de maillage totalement hexaédrique dont les éléments sont alignés sur les bords du domaine est un problème extrêmement difficile, et il n'existe à l'heure actuelle aucun algorithme pour mailler en hexaèdres des domaines quelconques. Une autre difficulté découle du fait que les très grandes déformations auxquelles sont soumis les objets en mécanique non linéaire se répercutent sur le maillage, dont la qualité se dégrade au cours de la simulation. Lorsqu'un élément du maillage s'inverse, c'est-à-dire que son jacobien de déformation par rapport à un carré devient négatif, la simulation échoue, et il faut alors reconstruire complètement le maillage.

Une des pistes les plus prometteuses pour générer des maillages hexaédriques qui répondraient à ces fortes contraintes est d'utiliser une paramétrisation du domaine à mailler, calculée à partir d'un champ de croix (champ de repère) qui s'aligne sur les bords du domaine et définit localement l'orientation et la structure des hexaèdres. L'équipe Pixel travaille sur ce type de méthode depuis une dizaine d'années et a obtenu d'excellents résultats.

Le but de cette thèse est de développer des méthodes de maillage hexaédrique qui soient adaptées aux simulations de grandes déformations en mécanique non-linéaire. Pour relever ces défis, l'équipe Pixel de l'Inria Nancy Grand-Est, Total et sa filiale Hutchinson ont des expertises complémentaires et cet objectif commun. L'équipe Pixel a une grande expertise en maillage hex-dominant, et développe des logiciels de maillage de qualité mais qui sont historiquement principalement définis pour des applications en informatique graphique. Total et Hutchinson utilisent les maillages hexaédriques dans un contexte industriel, et ce principalement pour la simulation de grandes déformations (Figure 1). Ce projet de collaboration vise à tirer parti de la complémentarité des objectifs de Total et Hutchinson avec ceux de l'équipe Pixel. Une des principales pistes pour diminuer le temps passé par les ingénieurs à générer un maillage adéquat pour une simulation est de prendre en compte des informations a priori sur la déformation à laquelle le maillage et l'objet vont être soumis.

La génération de maillage quadrilatère est un sujet beaucoup plus mature que la génération de maillage hexaédrique. Des méthodes permettant de générer un maillage de bonne qualité aligné sur les bords d'un domaine ont été proposées. Les méthodes utilisant une paramétrisation du domaine permettent d'obtenir de très bons résultats. Mais lorsque le maillage est soumis à des très grandes déformations au cours de la simulation, il risque de s'inverser et d'empêcher

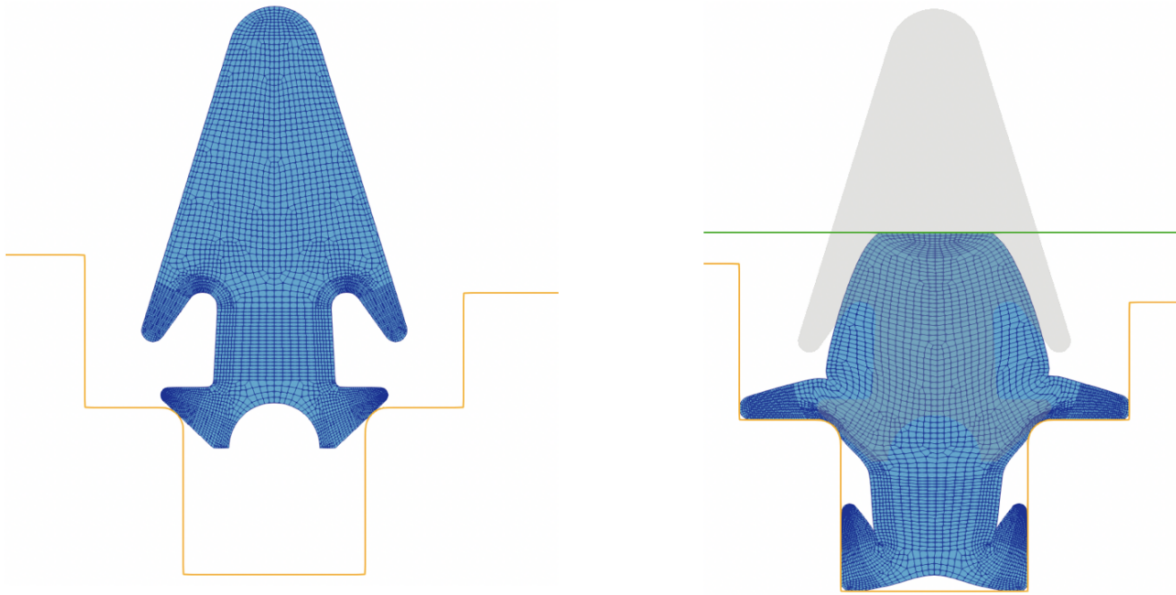


FIGURE 1 – À gauche : un maillage quadrilatère, l'initialisation de la simulation. À droite : la déformation du maillage au cours de la simulation risque de retourner des mailles ce qui bloque la simulation [Images Hutchinson].

la simulation d'aboutir au résultat. Actuellement, un premier maillage du domaine de départ est généré, puis la déformation est calculée en déplaçant les nœuds du maillage. Un maillage de qualité optimale au début de la simulation verra sa qualité se détériorer et potentiellement bloquer la simulation (Figure 2, (a)). De la même manière, un maillage optimisé pour l'objet déformé posera le même type de problème, la Figure 2, (b) nous en donne un exemple : le quadrilatère marqué par le point rouge a une très mauvaise qualité.

L'objectif du manuscrit est de trouver des connectivités de maillage permettant de garder des éléments quadrilatères / hexaédriques de bonne qualité au cours de toute la simulation de déformation. Ceci requiert d'avoir une bonne estimation des déformations auxquelles l'objet va être soumis. Nous allons réaliser une première simulation sur un maillage généré par paramétrisation globale sur l'état initial de l'objet. Ensuite nous allons utiliser l'évolution des positions des points du maillage lors de cette première simulation pour calculer un nouveau

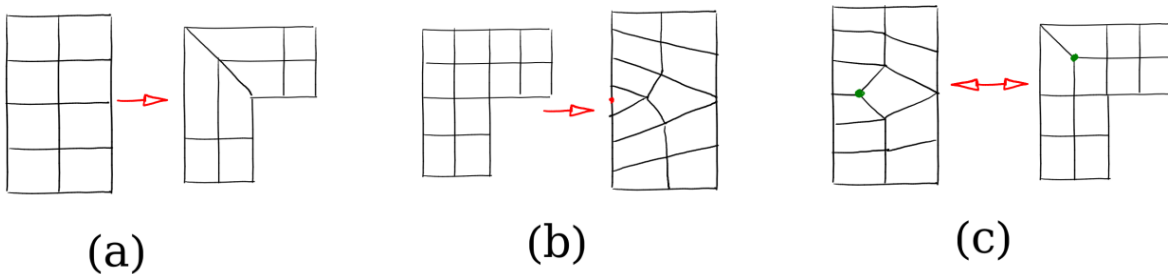


FIGURE 2 – Déformation d'une bande en L. (a) et (b) : on optimise la qualité du maillage au départ ou à l'arrivée, nous n'avons aucune garantie que le maillage soit d'une qualité acceptable pour d'autres pas de simulation. (c) L'idée est de trouver un maillage qui serait un compromis acceptable tout au long de la simulation.

maillage plus adapté. Itérativement, en continuant d’alterner entre simulation sur le maillage et recalcul d’un nouveau maillage initial à partir des résultats de la simulation précédente, nous espérons converger vers un maillage adapté à la simulation.

Les travaux de cette thèse ont eu pour objectif de trouver un moyen de réussir ce challenge de générer un maillage adapté à une simulation donnée. En 2D, nous avons réalisé ce défi via un pipeline complet mélangeant travaux antérieurs et contributions de la thèse. En 3D, nous avons simplement apporté des contributions permettant de nous approcher de cet objectif, notamment en apportant du contrôle sur les valences de bord d’un maillage hexaédrique produit à partir d’un champ de repère. En effet, les méthodes basées sur les champs de repère ne sont pas encore suffisamment robuste en 3D pour pouvoir être utilisées de manière automatique sur des modèles industriels.

Dans la partie I, nous allons d’abord décrire les différentes étapes pour réaliser une simulation numérique de grandes déformations. Nous verrons l’impact de l’étape de discrétisation (maillage) sur la convergence et la validité de la simulation, pour exprimer des critères de qualité sur les maillages hexaédriques utilisés pour des simulations de grandes déformations. Ensuite nous verrons les avantages des méthodes de paramétrisation globale comme l’algorithme Cubecover pour obtenir des maillages hexaédriques de bonne qualité.

Dans la partie II, nous allons nous focaliser sur nos contributions donnant du contrôle sur les résultats obtenus par des méthodes de champ de repère. Nous donnons notamment une méthode pour contrôler la valence des arêtes de bord dans les maillages hexaédriques produits par Cubecover, ce qui est très utile dans le cas de simulation de grandes déformations car la valence sur le bord doit absolument être suffisamment grande par rapport à la déformation de l’objet pour ne pas trop détériorer la qualité des éléments du bord.

Dans la partie III, nous nous concentrons sur le cas 2D, avec des contributions qui rendent le processus de remaillage quadrilatère plus rapide et robuste. Nous allons évaluer la validité de nos maillages quads pour la réalisation de simulation de grandes déformations, via le logiciel de simulation Numea [*NUMEA* s. d.].

Première partie

État de l'art de génération de maillage pour réaliser des simulations de grandes déformations

1

Simulation et maillage

Ce travail de thèse vise à élaborer des techniques de création automatisée de maillage hexaédrique pour répondre aux besoins spécifiques des simulations numériques de grandes déformations. Le premier chapitre de cette thèse se concentre sur l'introduction des simulations numériques et la justification du besoin de maillages adaptés. Nous abordons le cas qui motive cette thèse et notre partenaire Hutchinson : les simulations de grandes déformations sur un maillage hexaédrique unique. Nous introduisons également les caractéristiques essentielles d'un maillage hexaédrique utilisable dans ce contexte.

1.1 Introduction aux simulations numériques et aux maillages pour la recherche et développement

L'approche scientifique repose sur un processus itératif d'observation, de compréhension et d'expérimentation visant à appréhender et maîtriser notre environnement et à développer des outils adaptés. Prenons pour exemple l'évolution de la roue depuis son invention en Mésopotamie en -3500 avant J.C (Fig 1.1, gauche) jusqu'aux roues à pneus en caoutchouc utilisées aujourd'hui (Fig 1.1, droite).

Le développement et l'amélioration des outils, ici les roues, reposent sur l'analyse de leurs propriétés et performances, ainsi que sur l'expérimentation de nouvelles conceptions et matériaux.



FIGURE 1.1 – *L'évolution de la roue depuis son invention -3500 avant J.C (gauche), son allègement avec les inventions des roues à rayons en -2000 avant J.C (milieu), jusqu'aux roues à pneus en caoutchouc des véhicules que l'on utilise aujourd'hui (droite).*

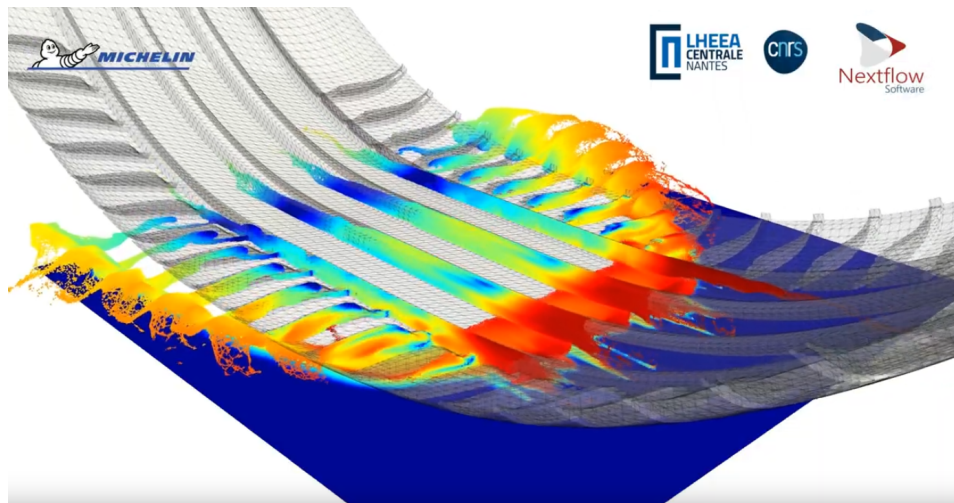


FIGURE 1.2 – Simulation d’aquaplanage par Michelin. L’objectif est de s’assurer que les pneus développés par l’entreprise respectent les normes de sécurité en termes d’aquaplanage, même lorsqu’ils sont usés. Image issue de https://www.youtube.com/watch?v=S6csZMRy_xk

Une innovation marquante a été réalisée en -2000 avant J.C : on a observé, compris et expérimenté que l’on peut alléger les roues sans les rendre fragiles avec le modèle des roues à rayon que nous utilisons encore aujourd’hui (Fig 1.1, milieu). Par la suite, il y a eu des améliorations régulières quand de nouveaux matériaux ont été découverts : le remplacement du bois par le métal pour solidifier la roue, puis l’invention des pneus en caoutchouc. Les pneumatiques, éléments clés des roues modernes, doivent avoir une adhérence respectant des normes de sécurité tout en minimisant les pertes énergétiques dues aux frottements. Étant donné la complexité des paramètres et des modèles physiques en jeu, l’expérimentation demeure essentielle.

La simulation numérique a révolutionné notre capacité à explorer et optimiser les conceptions. Elle permet d’expérimenter le comportement d’un objet soumis à divers phénomènes physiques, sans avoir à construire de prototype réel. Aujourd’hui, la simulation numérique accélère le cercle vertueux d’observation, de compréhension et d’expérimentation, et est devenu un incontournable pour le développement de nouveaux pneumatiques (Fig 1.2).

Dans cette thèse, nous nous intéressons aux simulations numériques de grandes déformations, en particulier pour les matériaux hyperélastiques tels que l’élastomère, composant principal des pneumatiques.

1.1.1 Simulation Numérique : Choix de la méthode

Comme il est impossible de représenter les nombres réels avec une précision exacte sur un ordinateur, il est tout aussi impossible de réaliser une simulation numérique sans erreur numérique [PRESS et al. 2007]. À partir des contraintes de temps, de précision numérique, et des ressources disponibles, il faut choisir une méthode de simulation numérique qui va nous permettre de répondre à la question qui nous intéresse (par exemple, la simulation de la Fig 1.2 teste la résistance de pneus à l’aquaplanage).

Le modèle physique La première étape consiste à choisir un modèle physique adapté au problème à résoudre. Les modèles physiques décrivent les phénomènes et les interactions entre différentes variables du problème [CHAPRA et al. 2011]. Par exemple, les modèles de fluide tels que

Navier-Stokes, Euler et Stokes [BATCHELOR et al. 1967] sont utilisés pour étudier les écoulements de fluide, tandis que les modèles de mécanique tels que l'hyperélasticité, la plasticité et la rupture [SIMO et al. 2006] conviennent aux problèmes de contrainte et de déformation des matériaux.

La méthode numérique Une fois le modèle physique choisi, l'étape suivante consiste à déterminer la méthode numérique la plus adaptée pour le résoudre. Par exemple, la méthode des différences finies est couramment utilisée pour résoudre des équations aux dérivées partielles sur des grilles régulières en approximant les dérivées par des différences discrètes [PRESS et al. 2007]. La méthode des volumes finis est souvent employée pour les problèmes impliquant la conservation de quantités physiques telles que la masse ou l'énergie dans les écoulements de fluide [LEVEQUE 2002] : les intégrales de volume sont transformées en intégrales de surface grâce aux termes de divergence des équations. La méthode des éléments finis est privilégiée pour les problèmes de mécanique des solides et des structures [ZIENKIEWICZ et al. 2005] : elle permet d'approximer une solution d'équation aux dérivées partielles sur un volume.

La discrétisation en maillage Les méthodes numériques des différences finies, volumes finis, ou éléments finis sont des bonnes approximations lorsque le volume où est réalisée l'approximation est de forme simple et est de petite taille. La discrétisation en maillage d'un volume permet de décomposer l'objet étudié en un ensemble d'éléments simples interconnectés de petite taille, sur lesquels les approximations des méthodes numériques deviennent suffisamment précises pour respecter les critères de précision numérique que l'on s'est fixé [THOMPSON et al. 1985]. Un maillage structuré, idéalement une grille régulière, est souvent composé de quadrilatères en 2D ou d'hexaèdres en 3D. Une description complète de la structure d'un maillage hexaédrique est donnée dans la section 2.1. Un maillage non structuré est généralement composé de triangles en 2D ou de tétraèdres en 3D.

Les critères de convergence Le critère de convergence permet de définir la précision numérique à partir de laquelle on estime que la solution a convergé. Un critère trop large peut créer des erreurs numériques qui éloignent les résultats de ceux qui seraient observés dans la réalité, tandis qu'un critère trop petit va étendre le temps et le coût de la simulation. Un critère de convergence trop petit peut également poser des problèmes de stabilité, menant à un éloignement de la solution réelle plutôt qu'à une approximation plus précise [BERGAN et al. 1972].

Validation et Vérification Pour assurer la fiabilité de la simulation, les étapes de validation et de vérification sont nécessaires. La validation implique de comparer les résultats obtenus avec des données de référence issues d'expériences réelles ou de simulations éprouvées, d'évaluer les erreurs numériques et de déterminer un intervalle de confiance pour les résultats de la simulation [ROACHE 1998]. En cas de non-satisfaction, cette étape permet d'orienter vers les points de la méthode à améliorer.

1.1.2 Définition et propriétés fondamentales du maillage

Les maillages jouent un rôle crucial dans le domaine de la simulation numérique, permettant de résoudre des problèmes complexes sur des domaines géométriques variés. Dans cette sous-section, nous nous intéressons à la définition des maillages volumiques et à leurs propriétés fondamentales, indépendamment des simulations numériques.

Soit un domaine géométrique fermé Ω dans l'espace tridimensionnel \mathbb{R}^3 . Le bord de Ω , noté $\partial\Omega$, constitue une surface fermée. Un maillage \mathcal{M} vise à décomposer Ω en un ensemble fini

d'éléments polyédriques $(\sigma_i)_N \in \mathcal{M}$. Cette décomposition peut être formalisée de la manière suivante :

$$\Omega = \bigcup_{0 \leq i \leq N} \sigma_i$$

Dans le cas général, il convient de souligner que le bord $\partial\Omega$ ne peut pas être représenté de manière exacte si les sous-domaines $(\sigma_i)_N$ sont uniquement composés d'éléments polyédriques. Pour obtenir une précision adéquate dans la représentation du bord, il est nécessaire d'ajuster le nombre et la taille des éléments constituant le maillage.

Dans le contexte de cette thèse, notre attention se portera principalement sur la génération de maillages hexaédriques *conformes* pour des domaines géométriques définis par des descriptions issues de la *Conception Assistée par Ordinateur (CAO)*.

Définition de maillage conforme : Un maillage est conforme si l'intersection entre deux éléments adjacents est vide, un sommet commun, une arête commune ou une face commune.

Définition de description CAO : Il s'agit de la représentation numérique d'un objet dans un espace 2D ou 3D. Les modèles géométriques peuvent être constitués de points, de lignes, de courbes, et de surfaces, et sont généralement créés à l'aide d'entités géométriques primitives telles que les cercles, les sphères, les polygones et les splines.

1.2 Simulation de grandes déformations et impact sur le maillage

Nous allons dans cette section nous concentrer sur les simulations de grandes déformations. Plus particulièrement, nous allons nous focaliser sur les simulations d'hyperélasticité non-linéaire que nous avons réalisées avec le logiciel [NUMEA s. d.] pour tester la performance de nos maillages dans le chapitre 8. L'analyse de la méthode de résolution de ces simulations va nous permettre de déduire des caractéristiques importantes sur les maillages utilisés pour représenter les objets à déformer.

1.2.1 Maillage et conditionnement de la matrice de rigidité

Construction de la matrice de rigidité Le maillage d'un objet permet sa décomposition en éléments simples, par exemple des quadrilatères (Sec 1.2.2). La matrice de rigidité globale de l'objet, souvent notée K , est la pièce maitresse des simulations de déformation. Elle permet de définir comment l'objet se déforme lorsque des forces de déformation lui sont appliquées. Dans la section 1.2.2 nous présentons une méthode pour calculer la matrice de rigidité d'un élément quadrilatère e , que nous appelons matrice de rigidité élémentaire K_e . La connectivité du maillage quadrilatère (que nous supposons conforme) nous permet de construire la matrice de rigidité globale de l'objet K , à partir d'un assemblage des matrices de rigidité élémentaires K_e . Nous utilisons dans la section 1.2.3 la matrice K pour résoudre des systèmes linéaires de la forme $KX = b$, pour calculer les déplacements des nœuds du maillage à partir des forces de déformation appliquées sur l'objet.

Impact du conditionnement de K sur la précision numérique Le conditionnement d'une matrice est le rapport entre sa plus grande et sa plus petite valeur propre. Dans un système linéaire $KX = b$, une matrice K mal conditionnée (très supérieure à 1) fait qu'une petite erreur numérique sur K ou b entraîne une grande erreur numérique sur X . Or, comme

nous travaillons sur des simulations de déformation d'hyperélasticité non-linéaire, linéariser le problème itérativement par une méthode de Newton (Sec 1.2.3) entraîne a fortiori des approximations numériques. Lorsque K est bien conditionné, cette approximation numérique a peu d'impact sur la position X des nœuds du maillage. Cependant dans le cas où K est mal conditionné, les itérations de Newton (Sec 1.2.3) vont grandement amplifier les erreurs numériques sur les positions des nœuds du maillage, menant à des solutions éloignées de la réalité, ou parfois même à une impossibilité de convergence.

1.2.2 Matrice de rigidité élémentaire pour une déformation hyperélastique

Dans cette sous-section, nous expliquons comment dériver la matrice de rigidité d'un élément (\mathbf{K}_e) pour un problème de grandes déformations hyperélastiques en 2D avec un élément quadrilatère à 4 nœuds (Q4) et des fonctions de forme linéaires, en utilisant le modèle Neo-Hookean [RIVLIN 1948]. Un matériau de type Néo-Hookean est souvent utilisé pour modéliser des matériaux similaires au caoutchouc. Nous utilisons des notations similaires au chapitre 9 du livre [BONET et al. 1997].

Fonctions de forme Les fonctions de forme linéaires pour un élément Q4 sont définies comme suit :

$$\begin{aligned} N_1 &= \frac{1}{4}(1-x)(1-y), & N_2 &= \frac{1}{4}(1+x)(1-y), \\ N_3 &= \frac{1}{4}(1+x)(1+y), & N_4 &= \frac{1}{4}(1-x)(1+y) \end{aligned}$$

où x et y sont les coordonnées normalisées à l'intérieur de l'élément.

Matrice jacobienne de déformation En utilisant les fonctions de forme linéaires N_1, N_2, N_3, N_4 , nous pouvons exprimer les coordonnées globales déformées (X et Y) en fonction des coordonnées locales (x et y) et des coordonnées globales des nœuds de l'élément (P_1, P_2, P_3, P_4) :

$$\begin{aligned} X(x, y) &= N_1 P_{1x} + N_2 P_{2x} + N_3 P_{3x} + N_4 P_{4x} \\ Y(x, y) &= N_1 P_{1y} + N_2 P_{2y} + N_3 P_{3y} + N_4 P_{4y} \end{aligned}$$

Les coordonnées globales déformées X, Y nous donne la matrice jacobienne de la déformation \mathbf{F} et son déterminant \mathbf{J} :

$$\mathbf{F} = \begin{bmatrix} \frac{\partial X}{\partial x} & \frac{\partial X}{\partial y} \\ \frac{\partial Y}{\partial x} & \frac{\partial Y}{\partial y} \end{bmatrix}, \quad \mathbf{J} = \det \mathbf{F} = \frac{\partial X}{\partial x} \frac{\partial Y}{\partial y} - \frac{\partial X}{\partial y} \frac{\partial Y}{\partial x}$$

Matrice constitutive spatiale La matrice constitutive spatiale \mathbf{D} établit la relation entre la contrainte spatiale (également appelée contrainte de Cauchy) et la déformation spatiale (ou déformation de Green-Lagrange) d'un matériau dans sa configuration déformée. Pour un élément 2D et un matériau de type Neo-Hookean, nous pouvons exprimer la matrice constitutive spatiale \mathbf{D} en fonction des constantes de Lamé $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$, $\mu = \frac{E}{2(1+\nu)}$ qui sont définies à partir des paramètres du matériau étudié : le module de Young E et le coefficient de Poisson ν . Dans

un modèle d'hyperélasticité non-linéaire, il faut de plus prendre en compte la déformation de l'élément [BONET et al. 1997] : $\lambda' = \frac{\lambda}{J}$ et $\mu' = \frac{(\mu - \lambda \ln J)}{J}$.

$$D = \begin{bmatrix} \lambda' + 2\mu' & \lambda' & 0 \\ \lambda' & \lambda' + 2\mu' & 0 \\ 0 & 0 & \mu' \end{bmatrix}$$

Matrice déformation-déplacement La matrice déformation-déplacement non linéaire \mathbf{B} relie les déplacements nodaux aux déformations non-linéaires du matériau. Pour la calculer, nous devons utiliser les dérivées des 4 fonctions de forme N_1, N_2, N_3, N_4 par rapport aux coordonnées globales déformées (X et Y) en utilisant la matrice jacobienne de la déformation (\mathbf{F}) et son inverse (\mathbf{F}^{-1}) :

$$\begin{bmatrix} \frac{\partial N_i}{\partial X} \\ \frac{\partial N_i}{\partial Y} \end{bmatrix} = F^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1}{\partial X} & 0 & \frac{\partial N_2}{\partial X} & 0 & \frac{\partial N_3}{\partial X} & 0 & \frac{\partial N_4}{\partial X} & 0 \\ 0 & \frac{\partial N_1}{\partial Y} & 0 & \frac{\partial N_2}{\partial Y} & 0 & \frac{\partial N_3}{\partial Y} & 0 & \frac{\partial N_4}{\partial Y} \\ \frac{\partial N_1}{\partial Y} & \frac{\partial N_1}{\partial X} & \frac{\partial N_2}{\partial Y} & \frac{\partial N_2}{\partial X} & \frac{\partial N_3}{\partial Y} & \frac{\partial N_3}{\partial X} & \frac{\partial N_4}{\partial Y} & \frac{\partial N_4}{\partial X} \end{bmatrix}$$

Calcul de la matrice de rigidité élémentaire À partir de la matrice déformation-déplacement non linéaire (\mathbf{B}) et la matrice constitutive spatiale (\mathbf{D}), le calcul de la matrice de rigidité élémentaire (\mathbf{K}_e) pour le problème de grandes déformations hyperélastiques se fait par l'intégrale suivante :

$$\mathbf{K}_e = \int_{-1}^1 \int_{-1}^1 (\mathbf{B}^T(x, y) \mathbf{D}(x, y) \mathbf{B}(x, y)) \mathbf{J} dx dy$$

Cette intégrale doit être évaluée sur la surface de l'élément quadrilatère, ce qui nécessite généralement des techniques d'intégration numérique telles que la méthode de quadrature de Gauss.

Quadrature de Gauss La méthode de quadrature de Gauss consiste à approximer l'intégrale en sommant les valeurs de la fonction aux points de Gauss choisis. Pour un élément quadrilatère à 4 nœuds, un schéma d'intégration à 2x2 points de Gauss est couramment utilisé. Les coordonnées des points de Gauss sont :

$$(x_1, y_1) = \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}} \right), \quad (x_2, y_2) = \left(+\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}} \right)$$

$$(x_3, y_3) = \left(+\frac{1}{\sqrt{3}}, +\frac{1}{\sqrt{3}} \right), \quad (x_4, y_4) = \left(-\frac{1}{\sqrt{3}}, +\frac{1}{\sqrt{3}} \right)$$

En utilisant ces points de Gauss, nous pouvons approximer l'intégrale de la matrice de rigidité élémentaire comme suit :

$$\mathbf{K}_e \approx \sum_{i=1}^4 (\mathbf{B}^T(x_i, y_i) \mathbf{D}(x_i, y_i) \mathbf{B}(x_i, y_i)) \mathbf{J}(x_i, y_i)$$

où x_i, y_i représente les coordonnées du point de Gauss i .

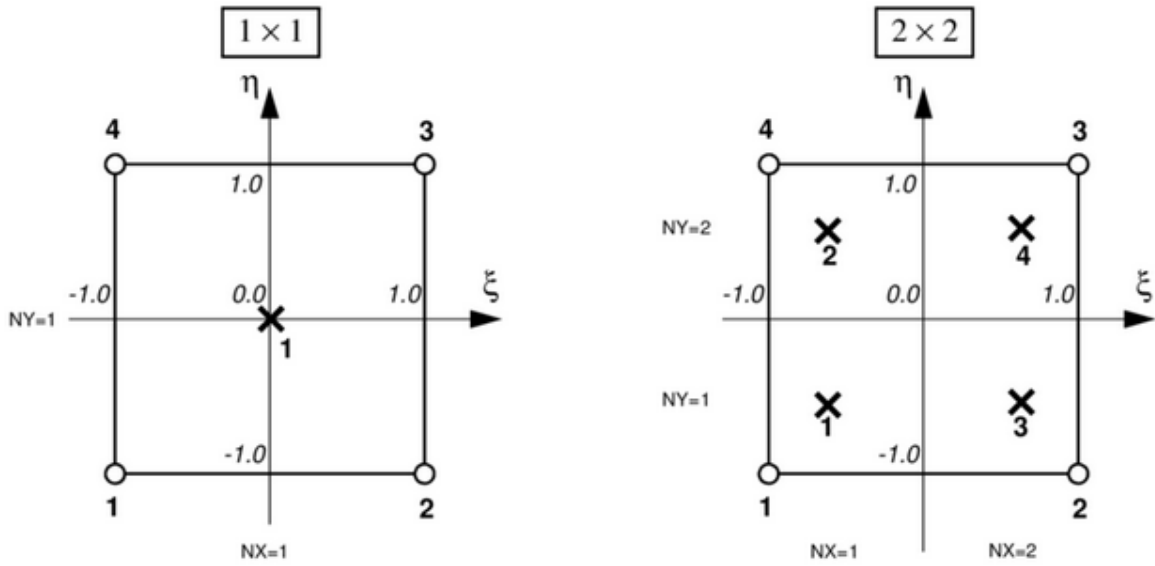


FIGURE 1.3 – Position des points de Gauss dans un élément Q_4

Impact de la qualité des éléments Chaque matrice de rigidité élémentaire \mathbf{K}_e a un conditionnement dépendant de la matrice jacobienne F_e sur les points de Gauss de l'élément e . Plus un élément e est éloigné d'un carré, plus le conditionnement de K_e est mauvais, ce qui détériore le conditionnement de la matrice de rigidité globale K après assemblage. L'article [KNUPP 2007] qualifie la qualité d'un élément dans un maillage pour une simulation comme sa déformation par rapport à l'élément idéal de la méthode des éléments finis utilisée. Dans cette méthode présentée et utilisée dans le chapitre 8, l'élément idéal est un carré, et il serait un cube dans le cas 3D. Par cette définition, le conditionnement de la matrice de rigidité globale, lié à la précision numérique de la simulation, dépend directement de la qualité des éléments quadrilatères du maillage utilisé.

1.2.3 Méthode de Newton complète pour calculer les déplacements nodaux

Après avoir construit la matrice de rigidité, nous pouvons désormais calculer les déplacements des nœuds du maillage. Dans cette sous-section, nous présentons la méthode de Newton complète utilisée dans le logiciel [NUMEA s. d.] pour calculer les déplacements nodaux dans un problème de déformation non-linéaire. L'algorithme commence par une estimation initiale du déplacement nodal, U_0 , et itère jusqu'à atteindre le déplacement nodal U_n qui respecte des critères de convergence. Le logiciel [NUMEA s. d.] utilise les critères de convergence DN_L2 et EN décrits ci-dessous pour déterminer si la méthode a convergé et que le déplacement nodal U_n obtenu est suffisamment proche de celui désiré.

Méthode de Newton complète

Étant donné le déplacement initial : U_0

Itérer jusqu'à respecter DN_L2 et EN :

1. Calcul de K (Sec 1.2.2) :

$$K(U_k)$$

2. Calcul du déplacement :

$$K(U_k)\Delta U = R(U_k)$$

où :

$$R(U_k) = F^{ext}(t) - F^{int}(U_k)$$

3. Mis à jour du déplacement :

$$U_{k+1} = U_k + \Delta U$$

Fin de l'itération

Résultat : U_n est la solution convergée

Critère de déplacements (DN_L2)

À la k-ième itération de Newton :

$$\|\Delta U_k\|_{L2-mean} \leq tol_U \cdot \|U_{cumul}\|_{L2-mean}$$

où :

$$U_{cumul} = \Delta U_1 + \dots + \Delta U_k$$

$$\Delta U_k \text{ est sol de } K(U_{total})\Delta U_k = R(U_{total})$$

tol_U est un paramètre

$\|\cdot\|_{L2-mean}$ est la norme L2 moyennée

Critère d'énergie (EN)

À la k-ième itération de Newton :

$$|E_k| \leq tol_E \cdot |E_0|$$

où :

$$E_k = R(U_k) \cdot \Delta U_k$$

E_0 est l'énergie initiale

tol_E est un paramètre

1.2.4 Un maillage conforme unique pour toute la simulation

Les étapes de remaillage lors d'une simulation nécessitent le transfert des données de la simulation (intensité des déformations), ce qui peut entraîner des erreurs numériques et un coût de simulation élevé. L'objectif de la thèse est de créer un maillage unique adapté à l'ensemble de la simulation de déformation. Il doit donc conserver une qualité suffisante malgré les importantes déformations subies.

L'avantage des maillage hexaédriques par rapport aux maillage tétraédriques pour des simulations de grandes déformations avec maillage unique est qu'il est possible de déformer un hexaèdre sans affecter ses angles dièdres, ce qui n'est pas possible pour un tétraèdre (Fig 1.4).

Réussir une simulation de grandes déformations sans remaillage en gardant des éléments de bonne qualité est souvent idéal en terme de précision numérique. Cependant ces maillages, souvent composés exclusivement d'hexaèdres, sont si difficiles à obtenir qu'ils résultent souvent de plusieurs mois de travail ingénieur.

1.2.5 Maillage hexaédrique de qualité pour ce type de simulation

Comme relevé avec un cas 2D dans la section 1.2.2, notre critère de qualité est guidé par le jacobien de déformation de chaque élément hexaédrique par rapport à un cube unitaire. Ces objectifs de qualité sont déterminés en fonction de la particularité du problème de simulation que nous voulons résoudre avec le logiciel [NUMEA s. d.] : des simulations hyperélastiques avec un maillage hexaédrique unique du début à la fin de la simulation.

Angles dièdres Les angles dièdres, formés entre les faces adjacentes des éléments hexaédriques, doivent être proches de 90 degrés pour prévenir les problèmes de convergence (Section 1.2.2). Un maillage hexaédrique de qualité possède uniquement des angles dièdres entre 45 et 135 degrés.

Rapport d'aspect Le rapport d'aspect, défini comme le rapport entre la plus grande et la plus petite arête de l'hexaèdre, influence également la qualité du maillage. Cependant, l'impact d'un

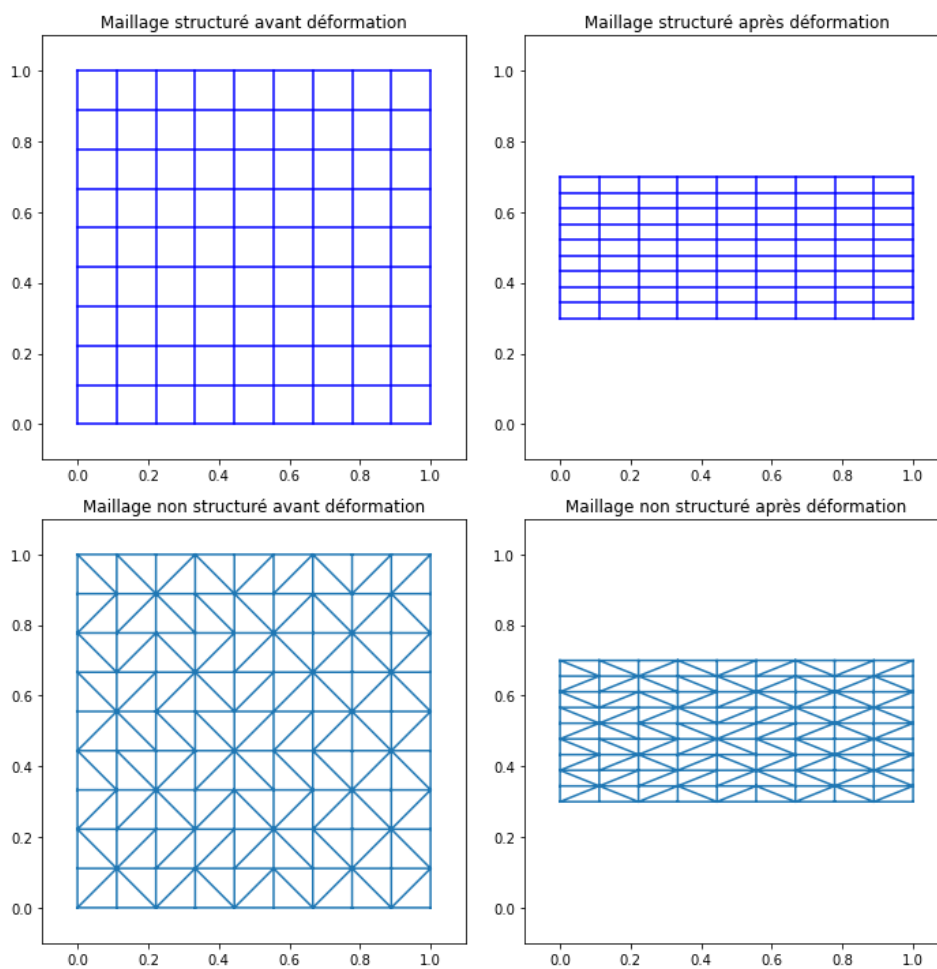


FIGURE 1.4 – Une déformation d'un maillage quadrilatère/hexaédrique peut se faire en préservant les angles de ses éléments, ce qui n'est pas possible pour un maillage triangulaire/tétraédrique.

haut rapport d'aspect est souvent moins problématique que l'impact d'un angle dièdre faible sur le jacobien de déformation et donc sur la convergence de la simulation. Il est également important de noter que si l'on désire de très grands rapports d'aspect pour réduire le nombre d'éléments d'un maillage, il est possible de construire un schéma numérique pour lequel l'élément idéal est un parallélépipède rectangle très allongé plutôt qu'un cube. Nos éléments idéaux étant des cubes uniformes, nous nous fixons pour objectif un rapport d'aspect maximum de 100 entre la plus grande et la plus petite arête d'un hexaèdre.

Alignement avec les bords Lors des déformations de contact, les forces de déformation s'appliquent généralement orthogonalement aux bords des objets. Aligner les arêtes du maillage avec les forces de déformation permet de préserver la qualité des angles (entre 45 et 135 degrés) pendant la simulation, au prix d'une détérioration du rapport d'aspect des éléments qui est moins problématique car restant loin de la valeur limite 100 (Fig 1.4).

Haute qualité localisée Il est crucial d'avoir des éléments de haute qualité aux endroits où les gradients de force de déformation sont élevés pour minimiser les erreurs numériques, quitte à détériorer le maillage là où les gradients sont faibles et où un mauvais conditionnement de la matrice de rigidité est moins problématique. Il est parfois nécessaire de raffiner le maillage sur ces zones à haut gradient de force pour obtenir la précision numérique désirée.

Dans ce chapitre, nous avons présenté la spécificité des simulations que nous avons étudié pendant la thèse : les simulations de grandes déformations hyperélastiques avec maillage unique. Puis, nous en avons déduit des critères de qualité de maillage quadrilatère/hexaédrique pour ce type de simulation. L'objectif de la thèse est de trouver une méthode de maillage quadrilatère/hexaédrique automatique respectant ces critères de qualité, et de démontrer son efficacité en lançant des simulations hyperélastiques avec le logiciel Numea [*NUMEA* s. d.].

2

Méthodes de maillage hexaédrique

Nous allons dans ce chapitre faire un état de l'art des différentes techniques de maillage hexaédrique, et discuter de l'utilisabilité de ces techniques pour la simulation de grandes déformations avec un maillage unique. Comme le montre la figure 2.1, la qualité du maillage hexaédrique utilisé pour réaliser une simulation de grandes déformations par la méthode décrite dans la section 1.2 a un impact sur la convergence de la solution.

2.1 Graphe de singularité d'un maillage hexaédrique

Le maillage hexaédrique le plus régulier est une grille cartésienne infinie, où chaque sommet, arête et face est incident à 8, 4 et 2 hexaèdres, respectivement. Les maillages hexaédriques généraux contiennent des éléments de connectivité locale différente, qui sont donc appelés irréguliers ou singuliers. Pour un maillage hexaédrique, nous parlons de valence d'une arête pour désigner le nombre d'hexaèdres partageant l'arête.

Arêtes singulières : le graphe de singularité Comme dans [PIETRONI et al. 2022], nous définissons le graphe de singularité d'un maillage hexaédrique comme étant constitué des arêtes intérieures qui n'ont pas une valence de 4, ainsi que des arêtes de bord qui n'ont pas une valence de 2. La structure d'un maillage hexaédrique est déterminée par le graphe de singularité : plus le graphe de singularité est simple, plus le maillage hexaédrique est structuré.

Sommets singuliers d'un maillage hexaédrique Bien qu'un seul entier k soit suffisant pour caractériser le type structurel d'une arête (sa valence), la spécification des types de sommet est plus complexe. En se limitant à des valences d'arête intérieur $k \in \{3, 4, 5\}$, il s'avère que seulement 11 configurations différentes de types de sommets intérieurs existent. Plus précisément, pour un sommet intérieur, il est impossible d'être incident à une seule arête singulière, et dans le cas de deux arêtes singulières incidentes, elles ne peuvent être que du même type. Par conséquent, le graphe de singularité est formé par des arcs singuliers, qui sont des chaînes d'arêtes singulières de type identique. Ces arcs singuliers se connectent à d'autres arcs singuliers aux sommets singuliers, ou se terminent sur le bord [PIETRONI et al. 2022].

2.2 Maillage structuré

Les maillages structurés, souvent construits à partir d'un système de coordonnées, ont une structure régulière et répétitive. Ces maillages n'ont pas de singularité intérieure, et le graphe de

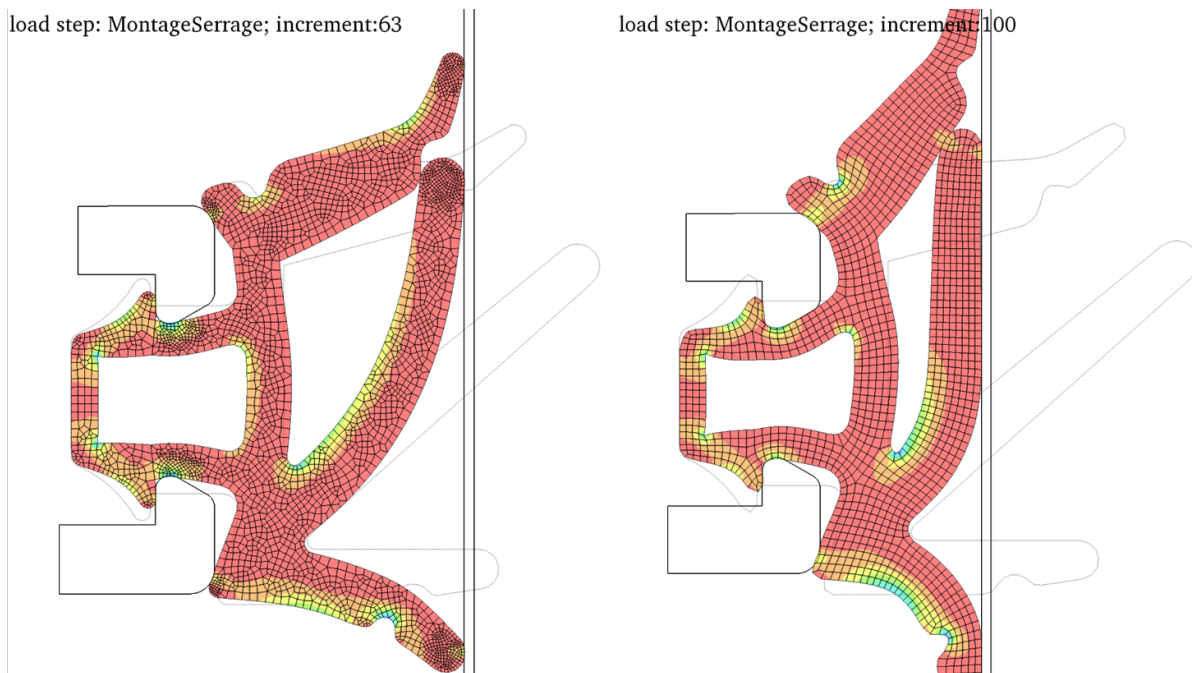


FIGURE 2.1 – Comparaison entre deux méthodes de génération de maillage automatique pour la simulation de déformation sur le logiciel [NUMEA s. d.]. La simulation avec un maillage non-structurée (gauche) diverge après l’itération 63, tandis que la simulation avec un maillage à valence semi-structurée (droite) parvient à atteindre l’itération 100 de la simulation.

singularité de bord dépend du système de coordonnées utilisé. Ce manque de flexibilité du graphe de singularité empêche l’utilisation de maillage structuré pour des modèles CAO industriels.

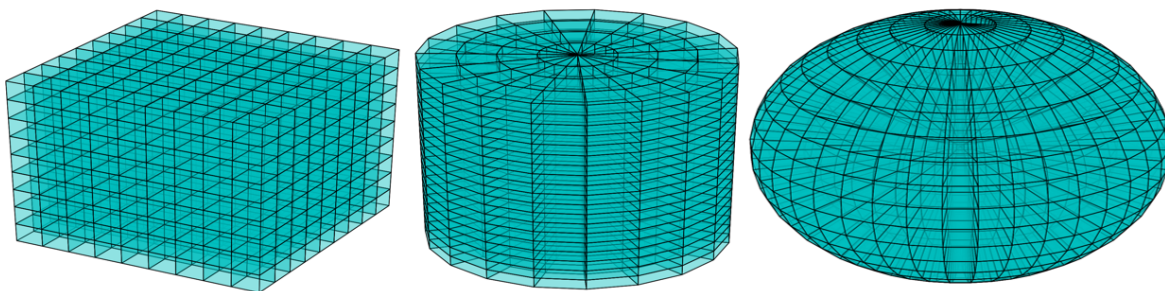


FIGURE 2.2 – Maillage cartésien, cylindrique et sphérique, structuré successivement selon les coordonnées cartésiennes (x, y, z) , cylindriques $(r, \theta$ et $z)$ et sphériques $(r, \theta$ et $\phi)$.

Maillage cartésien, cylindrique et sphérique Les maillages structurés hexaédriques utilisent un système de coordonnées pour placer de manière régulière leurs éléments (Fig 2.2). Selon la forme de maillage que l’on veut obtenir, on peut utiliser un système de coordonnées cartésien, cylindrique ou sphérique. L’avantage de ces maillages utilisant un système de coordonnées et que l’on peut passer d’un élément à son voisin directement par leur indice de coordonnée. Ainsi, des schémas numériques optimisés pour ces types de maillages peuvent souvent être bien plus performants en temps et en précision que les schémas numériques adaptés à des maillages hexaédriques

quelconques.

Déformation du système de coordonnées Pour obtenir un maillage d'une autre forme qu'un parallélépipède rectangle, cylindre ou sphère, il est possible d'appliquer des déformations sur le système de coordonnées utilisé. Par exemple, une déformation hyperbolique permet de modifier la forme d'un modèle et/ou de concentrer les hexaèdres proches des bords situés sur des zones à fort gradient d'une simulation (Fig 2.3).

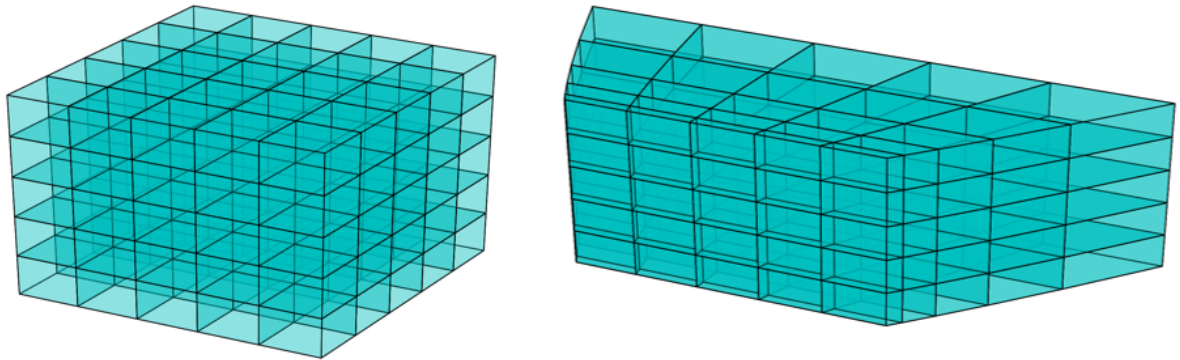


FIGURE 2.3 – Exemple de déformation hyperbolique appliquée à un maillage cartésien pour changer la forme et la répartition des éléments du maillage.

Utilisabilité avec Numea Lorsque le modèle sur lequel on veut appliquer une simulation de déformation a une forme naturellement représentable par un système de coordonnées, il est idéal d'utiliser des maillages structurés avec un schéma numérique adapté. Malheureusement, les modèles industriels définis par des descriptions CAO (Fig 2.1) sont souvent trop compliqués pour pouvoir être maillé entièrement avec un maillage structuré de bonne qualité. Dans ce cas, il faut plutôt utiliser des maillages hexaédriques qui ne sont pas parfaitement structurés.

2.3 Maillage non-structuré

Les maillages non-structurés possèdent un graphe de singularité complexe, voir chaotique, mais sont les plus simples à adapter de manière automatique à des modèles CAO industriels.

2.3.1 Méthode octree

Construction d'un octree non-conforme La méthode octree, introduite dans [SCHNEIDERS 2000], commence par englober l'objet ou le domaine d'intérêt dans un cube, qui constitue le nœud racine de l'octree. Le cube est ensuite subdivisé en huit sous-cubes de taille égale, créant les huit enfants du nœud racine. Cette subdivision est répétée récursivement pour chaque nœud, jusqu'à ce que la taille des éléments soit celle désirée (Fig 2.4).

Construction d'un maillage hexaédrique conforme Il s'avère que si tous les sommets de la grille ont une valence de 6 et que toutes les arêtes de la grille ont une valence de 4, le dual de la grille est un maillage hexaédrique conforme [MARÉCHAL 2009]. Ainsi régulariser la valence des éléments de la grille par un ensemble de schémas de découpage permet d'obtenir un maillage hexaédrique conforme par dualisation.

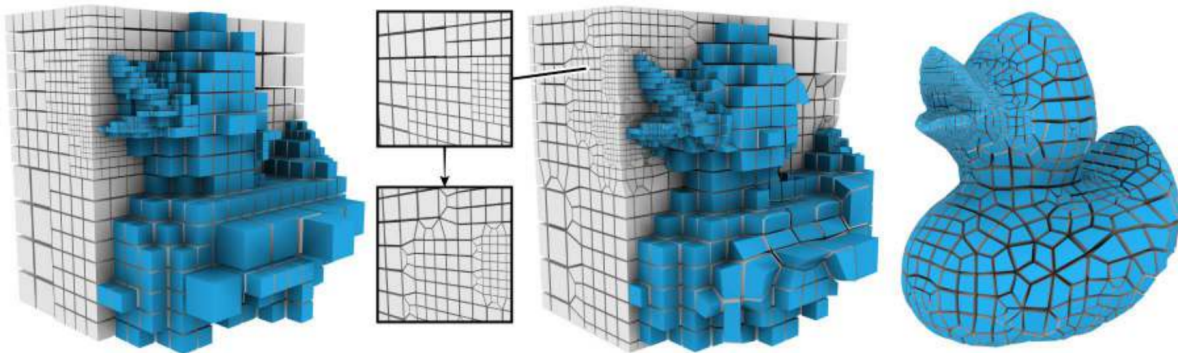


FIGURE 2.4 – Maillage hexaédrique (droite) obtenu à partir d'une grille d'octree (gauche). Image issue de [Livesu et al., 2022].

Utilisabilité avec Numea Les méthodes basées sur des octrees pour construire un maillage hexaédrique conforme de manière automatique sont les seules qui sont aujourd'hui fiables industriellement. Cependant, le grand taux de réussite est notamment expliqué par le fait que ces méthodes produisent les résultats les moins structurés avec des éléments de mauvaise qualité sur le bord. Comme détaillé dans la section 1.2.5 les éléments de mauvaise qualité sur le bord engendrent une matrice de rigidité avec un mauvais conditionnement là où les gradients de déformation sont les plus importants, ce qui empêche la convergence. De plus, sachant que notre objectif est de garder le même maillage du début à la fin de la simulation, le non-alignement des éléments avec les forces de déformation amène à une détérioration rapide des angles dièdres des éléments du bord.

2.3.2 Méthode d'avancée de front

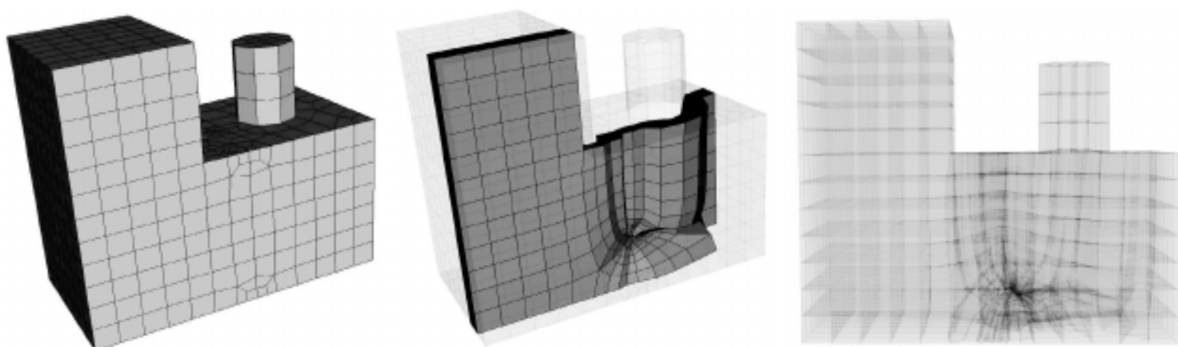


FIGURE 2.5 – Maillage obtenu par une méthode d'avancée de front. Le maillage extérieur est non-structuré autour du cylindre (gauche), et la structure intérieure peut posséder une grande concentration d'éléments de mauvaise qualité (zone sombre, droite). Image issue de [Ledoux et Weill 2008].

Construction d'un maillage hexaédrique par avancée de front La méthode d'avancée de front, introduite dans [BLACKER et al. 1993], commence par initialiser le front de maillage

en se basant sur les bords de la géométrie du domaine (Fig 2.5). À chaque étape, un élément non-maillé du front est sélectionné pour être étendu. De nouveaux éléments sont alors générés en connectant cet élément non-maillé aux nœuds du maillage existant ou en ajoutant de nouveaux nœuds, selon des critères de qualité et de résolution. Une fois qu'un nouvel élément est inséré, le front de maillage est mis à jour pour inclure les nouveaux éléments non-maillés qui en résultent.

Remplissage des cavités Lorsqu'il n'est plus possible d'insérer de nouveaux éléments sur un front de maillage, pour cause de contradiction avec d'autres fronts par exemple, il faut remplir les cavités finales par une autre méthode algorithmique. Ce remplissage est souvent basé sur la recherche d'un maillage hexaédrique à partir des bords quadrilatères de la cavité. Si le bord de la cavité possède un nombre pair de quadrilatère, un maillage hexaédrique est garanti d'exister [MITCHELL 1996]. Cependant, lorsque les quadrilatères entourant la cavité ne sont pas disposés d'une façon cohérente avec la géométrie de la cavité, remplir la cavité par des hexaèdres induit souvent des éléments de mauvaise qualité avec un graphe de singularité inadapté.

Utilisabilité avec Numea Les méthodes d'avancée de front ont généralement l'avantage par rapport aux méthodes d'octrees de produire des maillages hexaédriques bien aligné avec les forces de déformation au bord, ce qui fait que les déformations ont un impact amoindri sur la qualité du maillage (section 1.2.5). De plus, il est en général possible d'avoir des éléments de bonne qualité proche des bords où sont appliqués les forces de déformations. Cependant, pour des modèles CAO industriels délimités par des bords très irréguliers, le remplissage de cavités de grande taille par des maillages non-structurés avec des éléments de faible qualité pose souvent des problèmes de convergence.

2.4 Maillage semi-structuré

Les maillages semi-structurés, ou structurés par bloc, possèdent un graphe de singularité simple divisant le volume en un petit nombre de blocs structurés. Ces maillages sont souvent idéaux, mais nécessitent une grande interaction humaine pour être construits sur des modèles CAO industriels.

2.4.1 Maillage par balayage

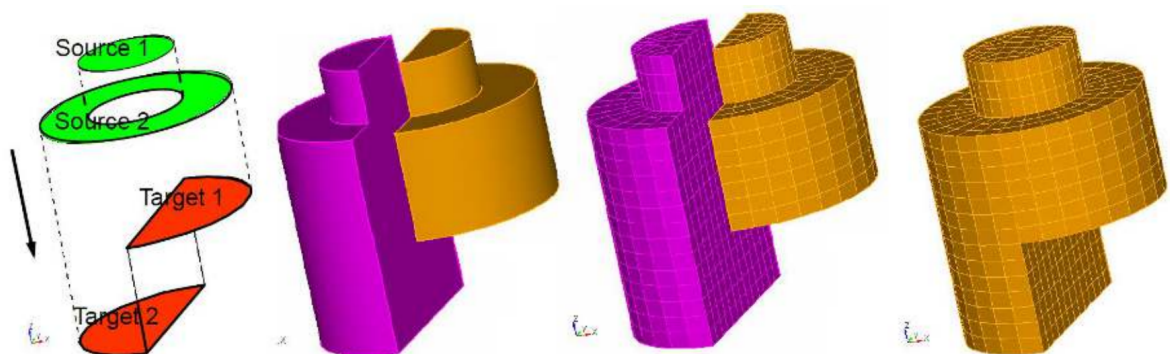


FIGURE 2.6 – Illustration d'un algorithme de balayage. Image issue de [WEILL and Ledoux, 2016].

Maillage par balayage d'un modèle 2D extrudé La méthode par balayage pour la construction de maillage hexaédrique consiste à générer un maillage en extrudant une surface d'une zone source maillée en quadrilatères vers une zone cible à travers un volume donné [WHITE et al. 1995]. Dans le cas d'un volume formé par extrusion d'un domaine 2D, on peut construire avec une méthode automatique un maillage quadrilatère du domaine 2D source et en obtenir une extrusion hexaédrique.

Maillage par balayage multi-sources et multi-cibles Pour les modèles plus complexes, plusieurs zones sources et cibles sont nécessaires pour décomposer la géométrie d'extrusion en sous-volumes plus simples à traiter [LAI et al. 2000]. Ces méthodes multi-sources et multi-cibles sont bien adaptées aux modèles CAO, car de nombreuses formes sont créées par extrusion (Fig 2.6). Cependant, pour les modèles CAO général où il n'existe aucune direction d'extrusion générale, la décomposition automatique d'un volume en sous-volumes qui peuvent être maillés avec des schémas prédéterminés ne peut pas être réalisé de manière automatique, et nécessite une intervention ingénieur.

Utilisabilité avec Numea Ces méthodes produisent des maillages structurés par bloc qui sont adaptés à notre besoin de maillage avec des éléments réguliers ressemblant à des cubes, et qui sont très bien alignés avec les bords où les déformations sont appliquées 1.2.5. Cependant, dans les cas de modèles CAO industriels, la décomposition d'un volume en sous-volume par un ingénieur pour construire un maillage par balayage prend plusieurs mois de travail.

2.4.2 Maillage via un descripteur médian

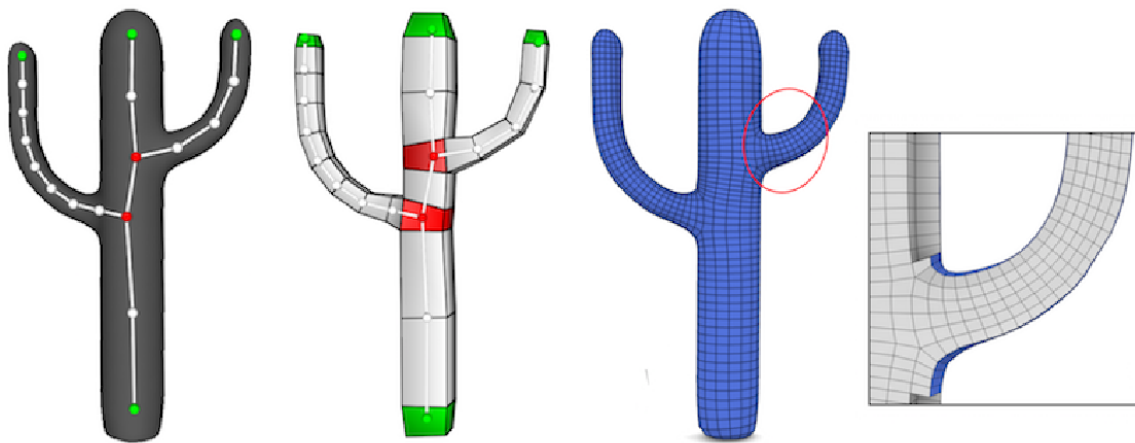


FIGURE 2.7 – Illustration de l'utilisation d'un squelette pour générer un maillage hexaédrique semi-structuré. Image issue de [LIVESU et al. 2016].

Représentation d'une volume par descripteur médian Une représentation d'un volume par un descripteur médian approprié permet une génération de maillage hexaédrique semi-structuré. Le descripteur médian, souvent représenté par un squelette tridimensionnel [PRICE et al. 1995]; [PRICE et al. 1997], sert de guide pour diviser le domaine en sous-domaines plus simples, facilitant ainsi la génération du maillage (Fig 2.7).

Primitives de maillage Pendant la construction du squelette d'un volume, des primitives de maillage sont assignées à chaque partie du squelette pour guider le processus de maillage hexaédrique. Ces primitives, déterminées en fonction de la géométrie du volume couvrant le squelette, assurent des transitions adaptées entre les différentes parties du maillage, correspondant à un graphe de singularité de maillage hexaédrique idéal.

Utilisabilité avec Numea Les méthodes de maillage hexaédrique basées sur des descripteurs médians sont particulièrement adaptées lorsque le volume peut être décomposé en un ensemble de "cônes généralisés", qui sont des formes obtenues à partir d'une transformation continue d'un domaine 2D le long d'une troisième dimension. Pour de tels modèles CAO, il est possible de générer rapidement des maillages hexaédriques semi-structurés qui satisfont les critères de qualité énoncés dans la section 1.2.5. Cependant, pour des modèles plus généraux, l'élaboration d'un descripteur médian approprié pour l'extraction d'un maillage hexaédrique semi-structuré nécessite un temps ingénieur similaire aux méthodes par balayage multi-sources et multi-cibles.

2.5 Maillage à valence semi-structurée

Les maillages à valence semi-structurée ont un graphe de singularité simple, comme les maillages semi-structurés. La différence est que ces maillages, souvent générés de manière automatique, comportent un nombre de blocs structurés plus important que le maillage semi-structuré qu'un ingénieur construirait intuitivement avec une méthode semi-automatique.

2.5.1 Polycube

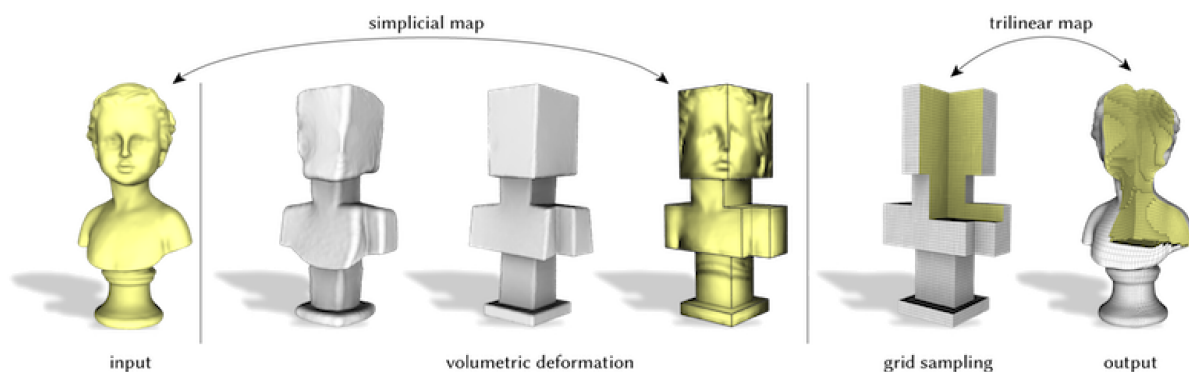


FIGURE 2.8 – Illustration d'un algorithme de Polycube. Image issue de [Pietroni et al., 2022].

Méthode des polycubes La méthode des polycubes, introduite dans [TARINI et al. 2004], fonctionne en cartographiant un objet volumique dans un polycube. Le polycube est échantillonné à une grille entière dense, créant une connectivité hexaédrique régulière, dont les nœuds peuvent être positionnés à l'intérieur de l'objet initial par cartographie inverse. La méthode des polycubes repose sur deux éléments fondamentaux : la définition de la structure du polycube et la génération de la carte volumique.

Structure du polycube La structure du polycube est généralement définie en assignant à chaque élément de surface de l'objet une étiquette représentant l'un des six axes globaux ($\pm X, \pm Y, \pm Z$). Les clusters d'éléments adjacents de même étiquette identifient les faces du polycube (Fig 2.8). Cependant, l'attribution d'une étiquette en se basant sur l'orientation locale de surface ne produit souvent pas une structure globalement cohérente, et un post-traitement heuristique est nécessaire pour corriger l'étiquetage.

Cartographie volumique La génération de la carte volumique est obtenue à l'aide d'énergies de déformation dédiées qui font tourner itérativement les normales de surface jusqu'à ce qu'elles se fixent sur les axes globaux (selon l'étiquetage obtenu).

Utilisabilité avec Numea Les méthodes de polycube permettent de générer automatiquement des maillages à valence semi-structurée, cependant il est fréquent que l'étape d'étiquetage mène à des erreurs que l'on ne sait pas résoudre autrement que par intervention humaine. De plus, la cartographie de l'objet volumique dans un polycube fait qu'il ne peut pas y avoir d'arête singulière à l'intérieur du volume, seulement sur le bord. C'est problématique pour notre application de simulation de grandes déformations, car c'est sur le bord que nous désirons la structure la plus régulière possible et un alignement des arêtes avec les forces de déformation.

2.5.2 Champs de repère

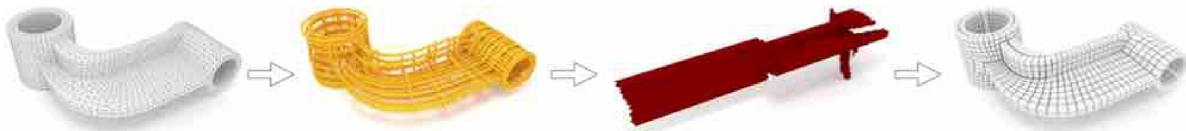


FIGURE 2.9 – Illustration d'un algorithme basé sur un champ de repère. Le modèle CAO est déformé vers une grille régulière (rouge), de façon à utiliser les lignes de la grille pour former un maillage hexaédrique sur le volume d'entrée (droite). Image issue de [Liu et al., 2018].

Représentation d'un champ de repère Un repère, composé localement de trois vecteurs linéairement indépendants, décrit la forme d'un cube déformé linéairement. Un champ de repère est souvent défini sur un maillage composé de tétraèdres : à chaque tétraèdre est affecté un repère. Pour calculer un maillage hexaédrique d'un objet 3D à partir d'un champ de repère, l'étape initiale est donc de calculer un maillage tétraédrique de l'objet par des méthodes rapides et automatiques [SI 2015]; [MAROT et al. 2018]. Le champ de repère défini sur le maillage tétraédrique utilise les propriétés de voisinage du maillage : deux repères sont voisins s'ils sont définis sur deux tétraèdres qui sont voisins.

Optimisation d'un champ de repère Le problème d'optimisation des champs de repère consiste généralement à trouver un champ de repère lisse (dont tous les repères voisins sont les plus similaires possibles) qui s'aligne sur le bord du domaine. Cette optimisation est souvent mesurée à l'aide de l'énergie de Dirichlet, avec pour contrainte que chaque repère correspondant à un tétraèdre de bord soit aligné avec la normale de la face de bord du tétraèdre.

Graphe de singularité d'un champ de repère Une fois un champ de repère calculé sur un maillage tétraédrique, l'analyse du comportement du champ de repère autour de chaque arête du maillage tétraédrique donne un graphe d'arêtes singulières. Pour un champ de repère lisse, ce graphe de singularité est très simpliste, et est un point de départ souvent idéal pour calculer un maillage hexaédrique à valence semi-structurée. Cependant, il arrive qu'un graphe de singularité obtenu à partir d'un champ de repère lisse sur un maillage tétraédrique ne corresponde pas directement à un graphe de singularité d'un maillage hexaédrique valide, ce qui peut faire échouer la méthode.

Extraction d'un maillage hexaédrique par cartographie En utilisant le champ de repère (et en supposant que son graphe de singularité est valide), l'étape suivante est de calculer une carte (fonction de \mathbb{R}^3 dans \mathbb{R}^3) telle que le volume défini par le maillage tétraédrique d'entrée se déforme en une grille 3D régulière (discontinue) par la carte (Fig 2.9). Si la carte est bijective et suit un certain nombre de règles de continuité déduites du champ de repère (détaillées dans la section 3.2.2), on peut utiliser les nœuds et arêtes de la grille 3D pour projeter (par cartographie inverse) un maillage hexaédrique valide sur le maillage tétraédrique d'entrée.

Utilisabilité avec Numea Les méthodes basées sur des champs de repère permettent la construction automatique d'un graphe de singularité simple à l'intérieur comme sur le bord du volume d'entrée. L'alignement des champs de repère avec les normales des bords du domaine permet la construction de maillage hexaédrique avec des éléments de bord qui restent de bonne qualité malgré des déformations appliquées pendant une simulation. Bien que ces méthodes ne garantissent pas la validité des graphes de singularité produits, et sont encore loin de réussir à le faire sur des modèles CAO industriels, ce sont les méthodes qui sont les plus proches de générer de manière automatique des maillages hexaédriques satisfaisant les critères de qualité décrits dans la section 1.2.5.

Nous déduisons de ce chapitre 2 que les méthodes utilisant des champs de repères semblent être les plus prometteuses pour construire de manière automatique des maillages hexaédriques respectant les critères de qualité donnés dans le chapitre 1. Dans le chapitre 3, nous allons étudier l'optimisation d'un champ de repère lisse, puis l'utilisation de son graphe de singularité pour construire un maillage hexaédrique par la méthode Cubecover.

Méthode Cubecover basée sur les champs de repères

Dans cette thèse, nous nous concentrons sur les méthodes de maillage basées sur des champs de repère. Ces méthodes sont réputées pour produire des maillages hexaédriques alignés avec les bords, avec des structures et connectivités simples. Nous allons d'abord présenter la méthode Quadcover [KALBERER et al. 2007] (la version 2D de Cubecover [NIESER et al. 2011]), que nous avons utilisée dans le chapitre 8 pour générer des maillages quadrilatères pour des simulations de grandes déformations avec le logiciel [NUMEA s. d.]. Ensuite, nous allons présenter la méthode Cubecover qui est une méthode de maillage hexaédrique qui n'a pas encore atteint un niveau suffisant de maturité pour être utilisée sans intervention humaine sur des modèles industriels, mais qui est prometteuse pour les simulations de grandes déformations étudiées dans le chapitre 1.

3.1 La méthode Quadcover pour la génération de maillage quadrilatère

Dans cette section, nous allons présenter la méthode Quadcover [KALBERER et al. 2007] pour construire automatiquement un maillage quadrilatère à valence semi-structurée à partir d'un maillage triangulaire en entrée. La figure 3.1 illustre les trois étapes principales de la méthode Quadcover. La première étape consiste à calculer un champ de repère, qui est un champ de croix aligné avec les bords du domaine (Sec 3.1.2). Dans la seconde étape, nous extrayons deux champs vectoriels du champ de repère, qui sont ensuite intégrés pour produire deux champs scalaires formant une carte sans couture u (Sec 3.1.3). Enfin dans la dernière étape, dite étape de quantification, nous faisons en sorte que les lignes de valeur entière de u forment un maillage quadrilatère (Sec 3.1.4).

3.1.1 Une méthode provenant de l'infographie

Cartographie de texture Le remaillage à partir de cartes est une méthode introduite dans la communauté de l'infographie par [ALLIEZ et al. 2002] pour améliorer la qualité d'un maillage triangulaire initial. Cette méthode consiste à générer une triangulation dans une carte 2D, puis à la projeter sur la surface d'intérêt. Cette technique est basée sur les algorithmes de cartographie de texture, pour lesquels les auteurs ont utilisé des coordonnées 2D attachées aux sommets des triangles de la triangulation d'entrée. En interpolant les coordonnées barycentriques de trois

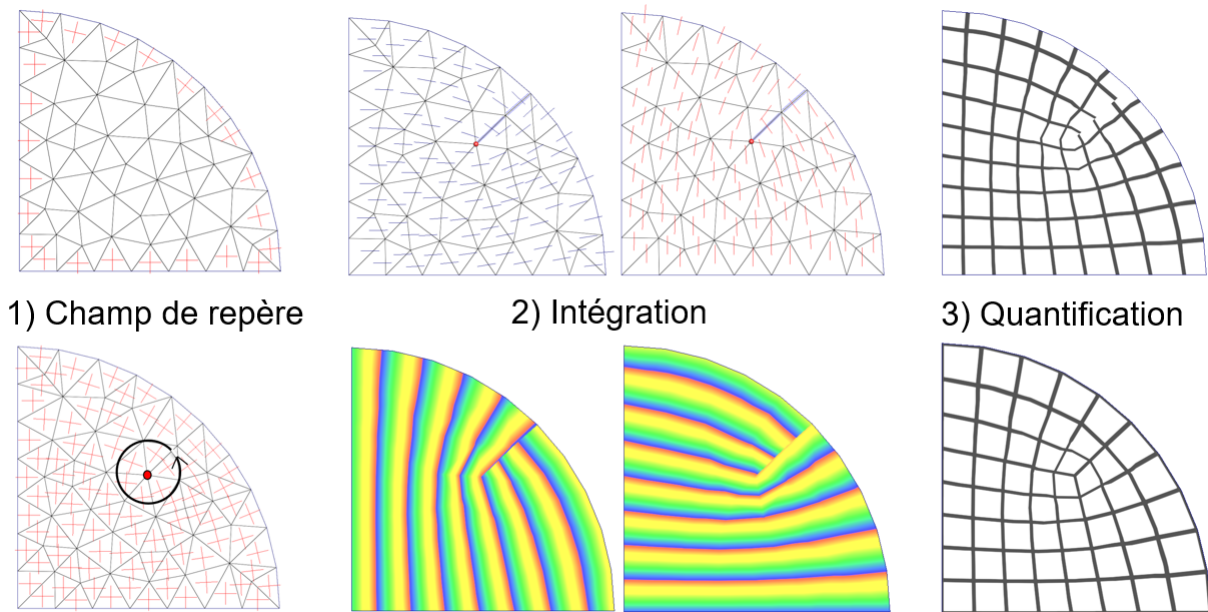


FIGURE 3.1 – Les différentes étapes de la méthode Quadcover pour construire un maillage quadrilatère.

coins de triangles, il est possible de projeter n’importe quel point de la carte sur la surface. Dans la suite, la carte d’une surface triangulée est définie comme une fonction u qui est linéaire par triangle. Ces cartes sont calculées par des méthodes numériques [HORMANN et al. 2007] minimisant une mesure de la distorsion de la carte sous certaines contraintes (injectivité, alignement avec arêtes caractéristiques, etc.). Quelques années plus tard et en utilisant des champs de repère [HERTZMANN et al. 2000], cette approche de remaillage par cartographie a été généralisée à la génération de maillages quadrilatères [RAY et al. 2006]; [KALBERER et al. 2007]; [BOMMES et al. 2009]. L’idée est de produire une carte d’une surface triangulée en entrée telle qu’une grille régulière définie dans la carte forme un maillage quadrilatère sur la surface.

Champ de repère Les champs de repère, ou champs de croix, ont été largement étudiés au cours de la dernière décennie en infographie. Ils ont été initialement introduits pour placer des hachures dans le rendu non photoréaliste [HERTZMANN et al. 2000]. Ils ont été généralisés en champs vectoriels à n -symétries [PALACIOS et al. 2007]; [RAY et al. 2008] en représentant une croix comme une racine $n^{\text{ième}}$ d’un nombre complexe. Le lissage d’un champ de croix est obtenu en mesurant la similarité de paires de repères adjacents, ce qui conduit à un problème d’optimisation quadratique simple permettant des contraintes d’alignement [RAY et al. 2008]; [KNÖPPEL et al. 2013] ou des contraintes de singularité [CRANE et al. 2010]. Des vues d’ensemble sur la conception de champs de repère peuvent être trouvées dans [VAXMAN et al. 2016] et [GOES et al. 2016].

Les champs de repère qui nous intéressent sont les champs vectoriels unitaires et orthogonaux à 4-symétries, car les singularités de ces champs de repère forment un graphe de singularité d’un maillage quadrilatère à valence semi-structurée [BOMMES et al. 2013b]. Notre objectif va être d’optimiser le lissage de ces champs de repère pour réduire le nombre de singularités, et ainsi construire des maillages quadrilatère de bonne qualité avec une méthode automatique.

3.1.2 Champ de repère orthonormal 2D

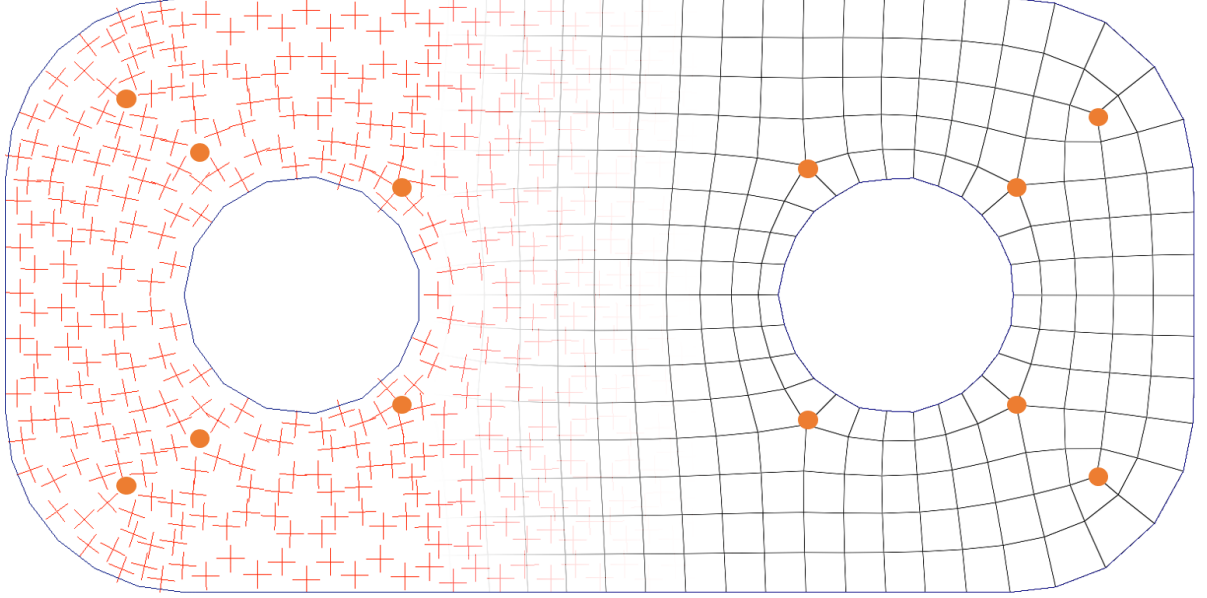


FIGURE 3.2 – Les singularités d’un champ de repère, qui sont des sommets autour desquels les repères font une rotation totale de $\pi/2$, sont utilisées par la méthode Quadcover pour déterminer la position des sommets de valence différente de 4 dans le maillage quadrilatère de sortie.

Notations Nous avons, en entrée, un maillage triangulaire (T, V) où nous notons T l’ensemble des triangles et V l’ensemble des sommets du maillage. L’objectif est de construire un champ de repère aligné avec les bords du maillage triangulaire (Fig 3.1, gauche). Nous supposons dans tout le manuscrit que chaque triangle n’a au plus qu’une arête appartenant au bord du domaine. Si ce n’est pas le cas pour un triangle de l’entrée, il suffit de le diviser en trois en ajoutant un sommet au centre pour que cela devienne le cas. L’ensemble des triangles du bord est noté T_b . Pour un maillage en 2D plat ($z = 0$), l’angle (par rapport au repère global) formé par l’arête de bord du triangle $t \in T_b$ est noté η_t , et va nous permettre de contraindre notre champ de repère.

Champ de repère 2D plat Un repère orthogonal 2D peut être représenté par une croix, qui est invariante sous les rotations de $\pi/2$. Pour avoir une représentation de même périodicité, [RAY et al. 2008] ont proposé de représenter la croix avec le vecteur $(X, Y) = (\cos 4\theta, \sin 4\theta)$, où θ représente la rotation de la croix par rapport au repère global. Pour obtenir un champ de repère lisse, nous fixons d’abord (X, Y) sur les bords, puis interpolons X et Y dans le domaine 2D (voir Fig. 3.1, gauche). Le problème d’optimisation consiste à rendre les valeurs de X et Y pour les triangles voisins t et t' aussi similaires que possible, tout en satisfaisant les valeurs contraintes sur les bords (Fig 3.2). En utilisant la notation $\mathcal{N}(t)$ pour représenter les triangles voisins d’un triangle t , le problème d’optimisation peut être formulé comme suit :

$$\begin{aligned} \arg \min_{X,Y} \quad & \sum_{t \in T} \sum_{t' \in \mathcal{N}(t)} \left\| \begin{pmatrix} X_{t'} \\ Y_{t'} \end{pmatrix} - \begin{pmatrix} X_t \\ Y_t \end{pmatrix} \right\|^2, \\ \text{sous contrainte :} \quad & \forall t \in T_b, \begin{pmatrix} X_t \\ Y_t \end{pmatrix} = \begin{pmatrix} \cos 4\eta_t \\ \sin 4\eta_t \end{pmatrix}. \end{aligned} \quad (3.1)$$

Nous utilisons une méthode des moindres carrés pour résoudre ce problème d'optimisation. À partir des valeurs X_t et Y_t calculées, nous pouvons calculer l'angle θ_t du champ de croix dans tous les triangles de notre maillage triangulaire d'entrée en utilisant la formule suivante : $\theta_t = \frac{1}{4} \tan^{-1}\left(\frac{Y}{X}\right)$.

Champ de repère 2D surfacique Lors du calcul d'un champ de repère sur une surface 2D qui n'est pas plate ($z \neq 0$), nous devons comparer les angles entre des faces adjacentes qui ne sont pas dans le même plan. Pour ce faire, nous pouvons utiliser l'arête commune entre les faces comme référence.

Pour chaque triangle, nous définissons θ_t comme l'angle autour de la normale de la face triangulaire t entre une branche de la croix et la première arête du triangle $e_{t,0}$. Pour comparer θ_t à l'angle correspondant dans un triangle adjacent t' , nous devons calculer la différence d'angle $\alpha_{t,i}$ entre $e_{t,0}$ et l'arête partagée $e_{t,i}$. Nous faisons de même pour t' , en calculant $\alpha_{t',i'}$ comme l'angle entre $e_{t',0}$ et $e_{t',i'}$. Ensuite, la différence d'angle entre θ_t et $\theta_{t'}$ peut être exprimée comme $(\theta_{t'} - \alpha_{t',i'} + \pi) - (\theta_t - \alpha_{t,i}) = \theta_{t'} - \theta_t - \omega_{tt'}$, en définissant $\omega_{tt'} = \alpha_{t',i'} - \alpha_{t,i} - \pi$ entre toutes les paires de faces adjacentes dans le maillage triangulaire d'entrée. Il faut également réexprimer les contraintes d'alignement au bord η , soit $t \in T_b$ un triangle du bord, i l'indice de son arête de bord, alors $\eta_t = \alpha_{t,i}$.

Avec cette définition, le problème d'optimisation pour calculer un champ de repère sur la surface d'un modèle 3D devient :

$$\begin{aligned} \arg \min_{X,Y} \quad & \sum_{t \in T} \sum_{t' \in \mathcal{N}(t)} \left\| \begin{pmatrix} X_{t'} \\ Y_{t'} \end{pmatrix} - \begin{pmatrix} \cos 4\omega_{tt'} & -\sin 4\omega_{tt'} \\ \sin 4\omega_{tt'} & \cos 4\omega_{tt'} \end{pmatrix} \begin{pmatrix} X_t \\ Y_t \end{pmatrix} \right\|^2, \\ \text{sous contrainte :} \quad & \forall t \in T_b, \begin{pmatrix} X_t \\ Y_t \end{pmatrix} = \begin{pmatrix} \cos 4\eta_t \\ \sin 4\eta_t \end{pmatrix}. \end{aligned} \quad (3.2)$$

Cette fois ci, l'angle θ_t obtenu par $\theta_t = \frac{1}{4} \tan^{-1}\left(\frac{Y}{X}\right)$, correspond à l'angle du repère par rapport au vecteur de l'arête $e_{t,0}$ en tournant autour de la normale de la face triangulaire t . Pour deux faces triangulaires voisines t, t' n'appartenant pas à un même plan, nous utilisons la valeur $\omega_{tt'}$ qui prend en compte le changement de vecteur de référence des angles $\theta_t, \theta_{t'}$ pour exprimer la rotation entre les repères (intrinsèquement $\pi/2$ périodique) : $r_{tt'} = \theta_{t'} - \theta_t - \omega_{tt'} \pmod{\frac{\pi}{2}}$.

Correspondances et singularités Nous allons maintenant utiliser notre champ de repère pour déterminer le graphe de singularité du maillage quadrilatère que nous voulons construire (la position des points singuliers de valence différente de 4). C'est le calcul des correspondances entre les repères voisins qui permet de transmettre un graphe de singularité de champ de repère vers un graphe de singularité de maillage quadrilatère. Comme dans [RAY et al. 2008], nous faisons référence aux correspondances du champ de repère comme étant les valeurs entières $p_{tt'}$ définies sur la demi-arête d'une face t , opposée à la face t' . À partir des angles θ_t obtenus par l'optimisation du champ de repère, nous définissons pour toutes les paires de faces triangulaires

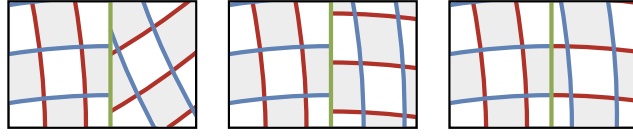


FIGURE 3.3 – Classification des cartes en fonction de leur comportement à travers une discontinuité (en vert) : carte arbitraire (à gauche), carte sans couture (au milieu) et carte préservant la grille (à droite).

voisines t, t' :

$$p_{tt'} = \text{round} \left(\frac{\theta_{t'} - \theta_t - \omega_{tt'}}{\pi/2} \right)$$

Cela nous permet d'exprimer la rotation entre deux repères adjacents directement en utilisant leurs angles sans rencontrer de problème de périodicité : $r_{tt'} = \theta_{t'} - \theta_t - \omega_{tt'} + p_{tt'} \cdot \pi/2$.

Les valeurs entières $p_{tt'}$ sont directement liées à l'emplacement des singularités du champs de repère [RAY et al. 2008]. Modifier ces valeurs de manière symétrique (incrémenter $p_{tt'}$ et décrémenter $p_{t't}$) permet de déplacer les singularités du champ de repère.

Optimisation à correspondance fixé Optimiser un champ de repère lisse à correspondance $p_{tt'}$ fixé consiste à minimiser les rotations $r_{tt'}$ des paires de triangles adjacents t, t' . Maintenant que nous n'avons plus à gérer de problème de périodicité, nous pouvons directement utiliser les angles θ comme variables du problème :

$$\begin{aligned} \arg \min_{\theta} \quad & \sum_{t \in T} \sum_{t' \in \mathcal{N}(t)} \|\theta_{t'} - \theta_t - \omega_{tt'} + p_{tt'}\pi/2\|^2. \\ \text{sous contrainte :} \quad & \forall t \in T_b, \theta_t = \eta_t. \end{aligned} \quad (3.3)$$

Lorsque l'on réoptimise un champ de repère à correspondance $p_{tt'}$ fixée, il faut fixer η_t à la valeur d'angle θ_t précédente (celle qui nous a permis d'obtenir les correspondances $p_{tt'}$).

Calcul d'une carte locale dont le repère est le gradient de direction En sélectionnant le vecteur du repère orienté par θ_t , et le vecteur orienté par $\theta_t + \pi/2$, on construit une matrice E_t contenant deux vecteur de \mathbb{R}^3 . Les matrices E_t forment des champs de vecteurs que nous allons localement intégrer dans chaque triangle t (Fig 3.1, milieu). À partir de E_t et des coordonnées géométriques des trois sommets du triangle $t : (a, b, c)$, on construit une carte locale linéaire $\mu : \mathbb{R}^2 \mapsto \mathbb{R}^2$ à partir de coordonnées de cartographie sur les trois coins $i, j, k \in \mathcal{C}(t)$ du triangle t :

$$\begin{cases} \mu_i = (0, 0) \\ \mu_j = E_t(b - a) \\ \mu_k = E_t(c - a) \end{cases} \quad (3.4)$$

Dans l'étape d'intégration, nous allons utiliser la carte paramétrique locale μ pour transférer les informations géométriques du champ de repère (orientation des quadrilatères), puis nous allons utiliser les valeurs entières $p_{tt'}$ pour transférer les informations de correspondance du champ de repère (graphe de singularité du maillage quadrilatère).

3.1.3 Intégration avec Quadcover

Dans cette sous-section, nous allons calculer une carte u sur un maillage triangulaire en entrée. L'objectif va être d'extraire un maillage quadrilatère à partir des lignes d'isovaleurs entières de la carte u sur le maillage triangulaire en entrée. La carte u est linéaire par triangle, et nous allons utiliser le champ de repère pour contraindre des règles de continuité sur chaque arête de triangle, de façon à que les lignes entières de la carte forment un maillage quadrilatère sans discontinuité.

Transfert de la géométrie du champ de repère Nous voulons que la carte u soit dans chaque triangle du maillage la plus proche possible de la carte local μ (à translation près). Nous pouvons par exemple faire en sorte que pour chaque paire de coin $i, j \in \mathcal{C}(t)$ d'un triangle t , la différence $u_i - u_j$ soit la plus proche possible de la différence $\mu_i - \mu_j$.

$$\arg \min_u \sum_{t \in T} \sum_{i, j \in \mathcal{C}(t)} \|(u_i - u_j) - (\mu_i - \mu_j)\|^2 \quad (3.5)$$

Minimiser cette énergie sans fixer aucune règle de continuité de la carte paramétrique u à travers les arêtes du maillage donne une carte arbitraire (Fig 3.3, gauche) identique à la carte μ . Pour contraindre les singularités de la carte et obtenir une carte dite sans couture (Fig 3.3, milieu), nous allons ajouter les contraintes correspondantes.

Contrainte de carte sans couture Dans une carte sans couture définie sur un maillage triangulaire, nous passons d'une carte d'un triangle à celle de son voisin par une translation et une rotation d'un multiple de 90 degrés. Pour toute paire de sommets adjacents v, w partagée par deux triangles t et t' , et en notant $i, j \in \mathcal{C}(t)$ et $i', j' \in \mathcal{C}(t')$ les coins de triangles correspondant aux sommets v, w , on ajoute la contrainte :

$$(u_{i'} - u_{j'}) = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}^{p_{tt'}} (u_i - u_j). \quad (3.6)$$

Contrainte d'alignement au bord Satisfaire les contraintes de carte sans couture 3.6 peut avoir l'effet indésirable de désaligner la carte paramétrique des arêtes de bord. Nous avons donc besoin d'une nouvelle contrainte pour nous assurer que nos maillages quadrilatères en sortie suivent le bord initial.

Soit i un coin de triangle, nous notons $u_{i,0}$ pour dénoter la première coordonnée du coin de triangle i dans le domaine paramétrique u . Pour tout coin de bord i, j d'un même triangle, et en supposant que le champ de repère est par construction aligné au bord sur sa première coordonnée $\mu_{i,0} = \mu_{j,0}$, on contraint :

$$u_{i,0} = u_{j,0} \quad (3.7)$$

Nous avons donc une carte sans couture u où sur chaque arête est définie une translation est une rotation d'un multiple de 90 degrés. Pour que la carte soit dite préservant la grille, et que ses lignes entières puissent être utiliser pour extraire un maillage quadrilatère, il faut de plus que les translations de la carte soient entières. Nous allons donc maintenant utiliser la carte sans couture obtenue (Fig 3.3, milieu) pour calculer une carte préservant la grille (Fig 3.3, droite) qui va nous permettre d'extraire un maillage quadrilatère.

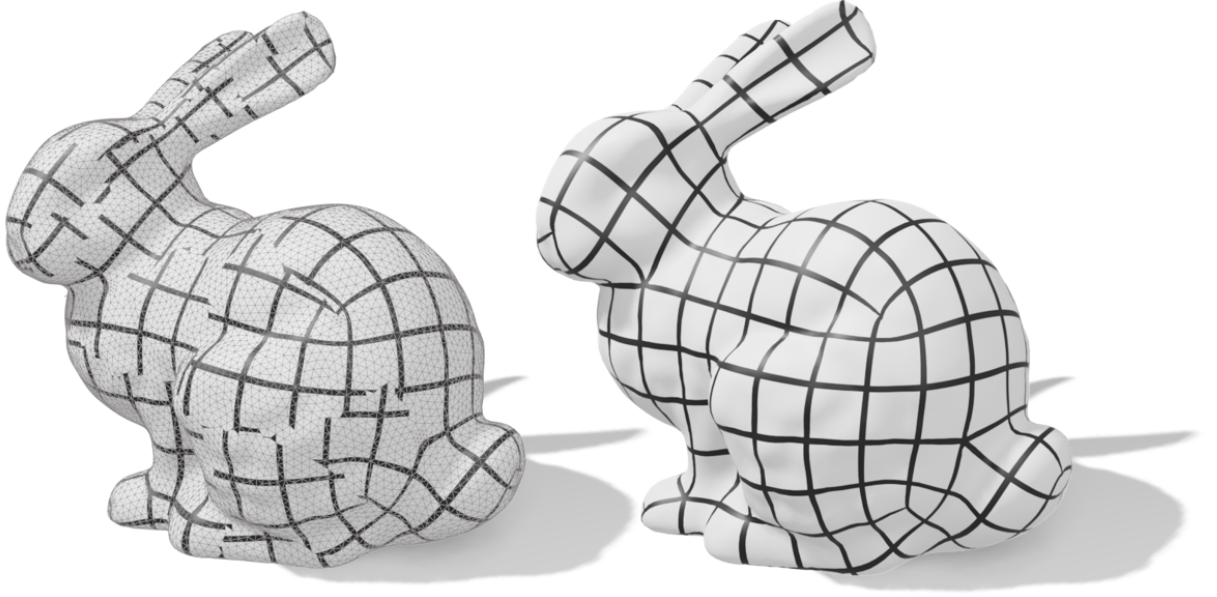


FIGURE 3.4 – L'étape de quantification permet de passer d'une cartographie sans couture (gauche) à une cartographie préservant la grille (droite).

3.1.4 Quantification de la paramétrisation

Contrainte de carte préservant la grille Pour obtenir une carte u préservant la grille (Fig 3.4, droite) il faut s'assurer que les lignes de valeurs entières de la carte soient alignées de part et d'autre de chaque arête du maillage triangulaire. Pour ce faire, il faut ajouter à la contrainte 3.6, la contrainte que chaque coin de triangle partageant un même sommet ait des coordonnées paramétriques (dans u) égales à translation entière et rotation d'un multiple de 90 degrés près. Pour tout sommet v partagé par deux triangles adjacents t et t' , et en notant $i \in \mathcal{C}(t), i' \in \mathcal{C}(t')$ les coins de triangles de t et t' correspondant au sommet v , on ajoute la contrainte :

$$u_{i'} - \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}^{p_{tt'}} u_i = \begin{pmatrix} n_{tt'} \in \mathbb{Z} \\ m_{tt'} \in \mathbb{Z} \end{pmatrix} \quad (3.8)$$

Contrainte d'alignement au bord sur une valeur entière Pour s'assurer que l'extraction d'un maillage quadrilatère par l'extraction des lignes de valeur entière de la grille produise un maillage aligné avec le bord du maillage triangulaire en entrée, il faut s'assurer que l'alignement de u sur le bord dans l'équation 3.7 le soit sur une valeur entière. Pour tout coin de bord i, j d'un même triangle, on contraint :

$$u_{i,0} = u_{j,0} \in \mathbb{Z} \quad (3.9)$$

Quantification par arrondi Dans les premiers travaux [KÄLBERER et al. 2007] ; [BOMMES et al. 2009], cette étape de quantification est réalisée par arrondi : on calcule les entiers $n_{tt'}, m_{tt'}$ et les entiers de bord en arrondissant les valeurs décimales correspondantes dans la carte sans couture. Puis, on ajoute les contraintes [3.8, 3.9] à partir de ces valeurs arrondies. Cette méthode permet effectivement d'obtenir une carte préservant la grille (Fig 3.3, droite), cependant elle peut être inutilisable pour l'application de maillage quadrilatère si cette carte possède des retournements.

Critère de validité d'une carte préservant la grille Une carte préservant la grille u définie sur un maillage triangulaire est dite valide, ou sans retournement, si elle respecte les contraintes [3.6, 3.8, 3.9], et si chaque triangle possède une aire strictement positive dans l'espace paramétrique définie par la carte.

Cette dernière condition est souvent exprimée par rapport au jacobien de déformation des triangles du maillage vers l'espace paramétrique. Ce jacobien de déformation doit être strictement positif.

Nécessité de carte sans retournement pour construire un maillage quadrilatère Supposons que nous avons une carte sans couture sans retournement, où nous appliquons une quantification par arrondi à l'entier le plus proche. Un cas d'échec courant de cette méthode pour construire une carte préservant la grille valide (sans retournement) est le cas où un triangle t possède deux sommets singuliers dont les coins correspondants i, j sont contraints à être à la même coordonnée entière dans la carte $u : u_{i,0} = u_{j,0} \in \mathbb{Z}$ et $u_{i,1} = u_{j,1} \in \mathbb{Z}$. Dans ce cas d'échec, nous obtenons une carte préservant la grille avec retournement. De manière général, tout point du maillage qui possède des valeurs entières pour les deux coordonnées de u devient un point du maillage quadrilatère de sortie. Dans notre cas d'échec et après interpolation linéaire, tout point de l'arête séparant i et j possède les mêmes valeurs entières pour les deux coordonnées de u après arrondi ; ce qui suggérerait une infinité de point pour le maillage de sortie sur cette arête et qui fait échouer la méthode. Certains algorithmes de construction de maillage quadrilatère à partir de carte préservant la grille possédant de faibles retournements ont été étudiés [EBKE et al. 2013], mais aucune méthode robuste n'existe.

La recherche s'est donc plutôt concentrée à l'élaboration d'une méthode de quantification garantissant de trouver des valeurs entières $n_{tt'}, m_{tt'}$ permettant la construction d'une carte préservant la grille valide à partir d'une carte sans couture valide.

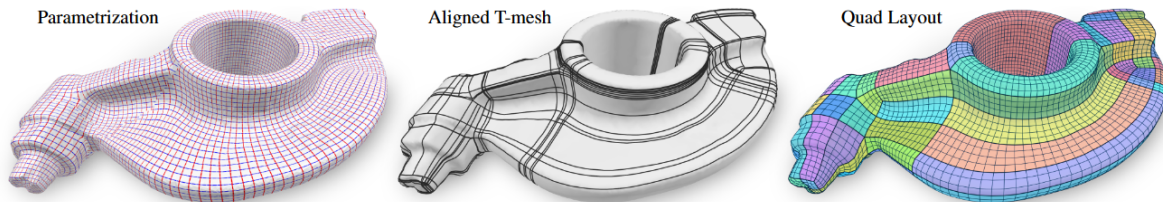


FIGURE 3.5 – Le calcul d'un maillage en T (milieu) à partir d'une carte sans couture (gauche), permet la construction d'un maillage quadrilatère (droite). Image issue de [LYON et al. 2021a].

Quantification avec garantie de non-retournement La garantie de validité peut être obtenue en utilisant un algorithme de séparation et évaluation qui minimise des fonctions de variables entières. Cependant, dans ce cas, il est nécessaire de recalculer une carte pour chaque ensemble de variables entières à évaluer, ce qui est très coûteux. Pour utiliser cette méthode, [BOMMES et al. 2013a] réalise une décimation du maillage triangulaire en entrée (pour réduire le nombre de sommets et triangles du maillage), puis applique l'algorithme de séparation et évaluation sur le maillage décimé. Ensuite, les variables entières obtenues sont transmises au maillage triangulaire de départ pour calculer une carte préservant la grille sur le maillage d'entrée. La méthode QGP [CAMPEN et al. 2015] adopte un autre point de vue : elle décompose la surface triangulée en subdivisions de forme quadrilatère (un maillage en T : Fig 3.5) et optimise la taille

d'une carte alignée sur un axe pour chaque subdivision, de manière à ce que les cartes puissent être assemblées en une unique carte préservant la grille. Cette idée vient de [MYLES et al. 2010] ; [KOVACS et al. 2011], qui génèrent des maillages en T à partir d'un champ de repère et proposent d'optimiser chaque subdivision pour générer un maillage quad sans jonction en T. Basé sur QGP, des travaux plus récents ajoutent certaines améliorations, comme une déviation angulaire bornée par rapport au champ de repère [LYON et al. 2021a], prennent en charge les bords libres [LYON et al. 2019] ou permettent même de fusionner des sommets singuliers [LYON et al. 2021b].

Quantification rapide sur un maillage décimé Dans l'état de l'art, les méthodes utilisant des maillages en T ont présenté beaucoup d'avantages en terme de rapidité sur la méthode [BOMMES et al. 2013a] appliquée sur un maillage décimé du maillage en entrée. Dans ce manuscrit, nous présentons dans le chapitre 7 une nouvelle méthode de quantification rapide sur un maillage décimé plutôt qu'un maillage en T, que nous utilisons dans notre pipeline de génération de maillage quadrilatère pour des simulations de grandes déformations 2D (chapitre 8).

3.2 La méthode Cubecover pour la génération de maillage hexaédrique

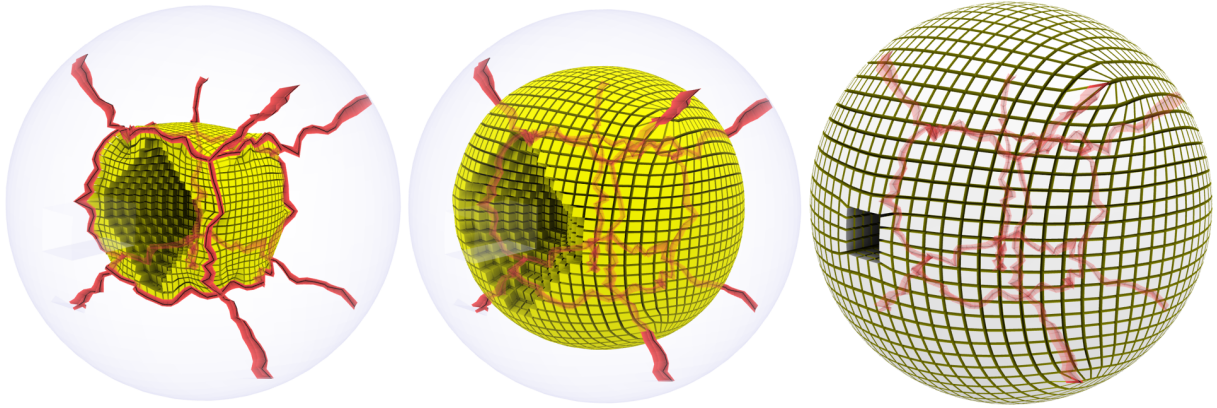


FIGURE 3.6 – En utilisant le graphe de singularité d'un champ de repère 3D (lignes rouges correspondant à des arêtes du maillage tétraédrique en entrée), la méthode cubecover permet de construire un maillage hexaédrique partageant le même graphe de singularité.

La méthode cubecover [NIESER et al. 2011] est une extension 3D de la méthode quadcover [KALBERER et al. 2007] pour la génération de maillage hexaédrique. Ainsi, dans ce chapitre, nous ferons souvent mention de ce que nous avons présenté dans la section précédente 3.1 pour simplifier l'explication de chacune des trois étapes de la méthode. La figure 3.6 présente un maillage hexaédrique construit par la méthode Cubecover à partir des arêtes singulière du champ de repère 3D (donc des arêtes du maillage tétraédrique d'entrée). Dans ce manuscrit nous utilisons hexalab [BRACCI et al. 2019] pour générer des images de nos maillages hexaédriques avec le graphe de singularité intérieur (lignes rouges et vertes).

3.2.1 Champ de repère orthonormal 3D

Historique Les champs de repère 3D sont couramment utilisés pour le maillage hexaédrique [NIESER et al. 2011]. Un seul repère orthonormal est constitué de trois vecteurs orthogonaux et de leurs opposés, de sorte qu'un repère orthonormal 3D a les mêmes symétries qu'un cube. Comme dans

le cas 2D, des généralisations aux champs de vecteurs 3D à n -symétries [SHEN et al. 2016] et aux champs de repère orthogonaux anisotropes [CHEMIN et al. 2018] ; [PALMER et al. 2020] ont été proposées en étudiant les propriétés de la base des harmoniques sphériques. L'objectif de l'optimisation du champ de repère est de générer un graphe de singularité interne valide en interpolant les contraintes d'alignement au bord. La représentation la plus courante d'un champ de repère orthogonal utilise 9 coefficients dans la décomposition en harmoniques sphériques [HUANG et al. 2011] ; [RAY et al. 2016]. De nombreux autres algorithmes sont possibles. Palmer *et al.* [PALMER et al. 2020] suggèrent d'alterner entre une étape de diffusion et une étape de projection pour obtenir un champ plus lisse. Vaxman *et al.* [VAXMAN et al. 2016] utilisent une méthode d'élément de bord pour interpoler les repères de bord à l'intérieur. Nous allons dans les prochains paragraphes présenter une méthode similaire à [RAY et al. 2016] pour générer des champs de repère 3D orthogonaux, qui est la méthode que nous allons utiliser comme initialisation des champs de repère 3D non-orthogonaux du chapitre 4, et celle que nous allons améliorer dans le chapitre 6.

$$\begin{aligned}
 Y_4^{-4}(\theta, \varphi) &= \frac{3}{16} \sqrt{\frac{35}{2\pi}} \cdot e^{-4i\varphi} \cdot \sin^4 \theta = \frac{3}{16} \sqrt{\frac{35}{2\pi}} \cdot \frac{(x-iy)^4}{r^4} \\
 Y_4^{-3}(\theta, \varphi) &= \frac{3}{8} \sqrt{\frac{35}{\pi}} \cdot e^{-3i\varphi} \cdot \sin^3 \theta \cdot \cos \theta = \frac{3}{8} \sqrt{\frac{35}{\pi}} \cdot \frac{(x-iy)^3 z}{r^4} \\
 Y_4^{-2}(\theta, \varphi) &= \frac{3}{8} \sqrt{\frac{5}{2\pi}} \cdot e^{-2i\varphi} \cdot \sin^2 \theta \cdot (7 \cos^2 \theta - 1) = \frac{3}{8} \sqrt{\frac{5}{2\pi}} \cdot \frac{(x-iy)^2 \cdot (7z^2 - r^2)}{r^4} \\
 Y_4^{-1}(\theta, \varphi) &= \frac{3}{8} \sqrt{\frac{5}{\pi}} \cdot e^{-i\varphi} \cdot \sin \theta \cdot (7 \cos^3 \theta - 3 \cos \theta) = \frac{3}{8} \sqrt{\frac{5}{\pi}} \cdot \frac{(x-iy) \cdot (7z^3 - 3zr^2)}{r^4} \\
 Y_4^0(\theta, \varphi) &= \frac{3}{16} \sqrt{\frac{1}{\pi}} \cdot (35 \cos^4 \theta - 30 \cos^2 \theta + 3) = \frac{3}{16} \sqrt{\frac{1}{\pi}} \cdot \frac{(35z^4 - 30z^2 r^2 + 3r^4)}{r^4} \\
 Y_4^1(\theta, \varphi) &= \frac{-3}{8} \sqrt{\frac{5}{\pi}} \cdot e^{i\varphi} \cdot \sin \theta \cdot (7 \cos^3 \theta - 3 \cos \theta) = \frac{-3}{8} \sqrt{\frac{5}{\pi}} \cdot \frac{(x+iy) \cdot (7z^3 - 3zr^2)}{r^4} \\
 Y_4^2(\theta, \varphi) &= \frac{3}{8} \sqrt{\frac{5}{2\pi}} \cdot e^{2i\varphi} \cdot \sin^2 \theta \cdot (7 \cos^2 \theta - 1) = \frac{3}{8} \sqrt{\frac{5}{2\pi}} \cdot \frac{(x+iy)^2 \cdot (7z^2 - r^2)}{r^4} \\
 Y_4^3(\theta, \varphi) &= \frac{-3}{8} \sqrt{\frac{35}{\pi}} \cdot e^{3i\varphi} \cdot \sin^3 \theta \cdot \cos \theta = \frac{-3}{8} \sqrt{\frac{35}{\pi}} \cdot \frac{(x+iy)^3 z}{r^4} \\
 Y_4^4(\theta, \varphi) &= \frac{3}{16} \sqrt{\frac{35}{2\pi}} \cdot e^{4i\varphi} \cdot \sin^4 \theta = \frac{3}{16} \sqrt{\frac{35}{2\pi}} \cdot \frac{(x+iy)^4}{r^4} \\
 Y_{4,-4} &= i \sqrt{\frac{1}{2}} (Y_4^{-4} - Y_4^4) = \frac{3}{4} \sqrt{\frac{35}{\pi}} \cdot \frac{xy(x^2 - y^2)}{r^4} \\
 Y_{4,-3} &= i \sqrt{\frac{1}{2}} (Y_4^{-3} + Y_4^3) = \frac{3}{4} \sqrt{\frac{35}{2\pi}} \cdot \frac{y(3x^2 - y^2) \cdot z}{r^4} \\
 Y_{4,-2} &= i \sqrt{\frac{1}{2}} (Y_4^{-2} - Y_4^2) = \frac{3}{4} \sqrt{\frac{5}{\pi}} \cdot \frac{xy \cdot (7z^2 - r^2)}{r^4} \\
 Y_{4,-1} &= i \sqrt{\frac{1}{2}} (Y_4^{-1} + Y_4^1) = \frac{3}{4} \sqrt{\frac{5}{2\pi}} \cdot \frac{y \cdot (7z^3 - 3zr^2)}{r^4} \\
 Y_{4,0} &= Y_4^0 = \frac{3}{16} \sqrt{\frac{1}{\pi}} \cdot \frac{35z^4 - 30z^2 r^2 + 3r^4}{r^4} \\
 Y_{4,1} &= \sqrt{\frac{1}{2}} (Y_4^{-1} - Y_4^1) = \frac{3}{4} \sqrt{\frac{5}{2\pi}} \cdot \frac{x \cdot (7z^3 - 3zr^2)}{r^4} \\
 Y_{4,2} &= \sqrt{\frac{1}{2}} (Y_4^{-2} + Y_4^2) = \frac{3}{8} \sqrt{\frac{5}{\pi}} \cdot \frac{(x^2 - y^2) \cdot (7z^2 - r^2)}{r^4} \\
 Y_{4,3} &= \sqrt{\frac{1}{2}} (Y_4^{-3} - Y_4^3) = \frac{3}{4} \sqrt{\frac{35}{2\pi}} \cdot \frac{x(x^2 - 3y^2) \cdot z}{r^4} \\
 Y_{4,4} &= \sqrt{\frac{1}{2}} (Y_4^{-4} + Y_4^4) = \frac{3}{16} \sqrt{\frac{35}{\pi}} \cdot \frac{x^2(x^2 - 3y^2) - y^2(3x^2 - y^2)}{r^4}
 \end{aligned}$$

FIGURE 3.7 – Liste des harmoniques sphériques d'ordre 4 complexes (gauche) et réelles (droite). Source : https://en.wikipedia.org/wiki/Table_of_spherical_harmonics.

Notations Nous avons, en entrée, un maillage tétraédrique (T, V) où nous notons T l'ensemble des tétraèdres et V l'ensemble des sommets du maillage. L'objectif est de construire un champ de repère 3D aligné avec les bords du maillage tétraédrique. Nous supposons que chaque tétraèdre n'a au plus qu'une face appartenant au bord du domaine. Si ce n'est pas le cas pour un tétraèdre de l'entrée, il suffit de le diviser en quatre en ajoutant un sommet au centre pour que cela devienne le cas. L'ensemble des tétraèdres du bord est noté T_b . Soit $t \in T_b$, le vecteur normal à la face de bord du tétraèdre t est noté n_t , et va nous permettre de contraindre notre champ de repère 3D.

Représentation en harmoniques sphériques Pour rappel, l'optimisation d'un champ de repère 2D se fait en minimisant dans le sens des moindres carrés la distance entre les représentations $(X, Y) = (\cos 4\theta, \sin 4\theta)$ pour chaque triangle, avec des contraintes pour les triangles du bord. En 3D, on ne peut pas utiliser ces fonctions de Fourier d'ordre 4, et on utilise donc l'extension 3D qui sont les 9 fonctions harmoniques sphériques réelles d'ordre 4 pour représenter un repère dans un tétraèdre $t : Y_{t,l}$ pour l allant de -4 à 4. La liste des harmoniques sphériques réelles d'ordre 4 est donnée sur la figure 3.7. Pour obtenir un champ lisse, on contraint les harmoniques sphériques sur le bord, puis on fait en sorte que les valeurs $Y_{t,l}$ soit les plus proches de leurs

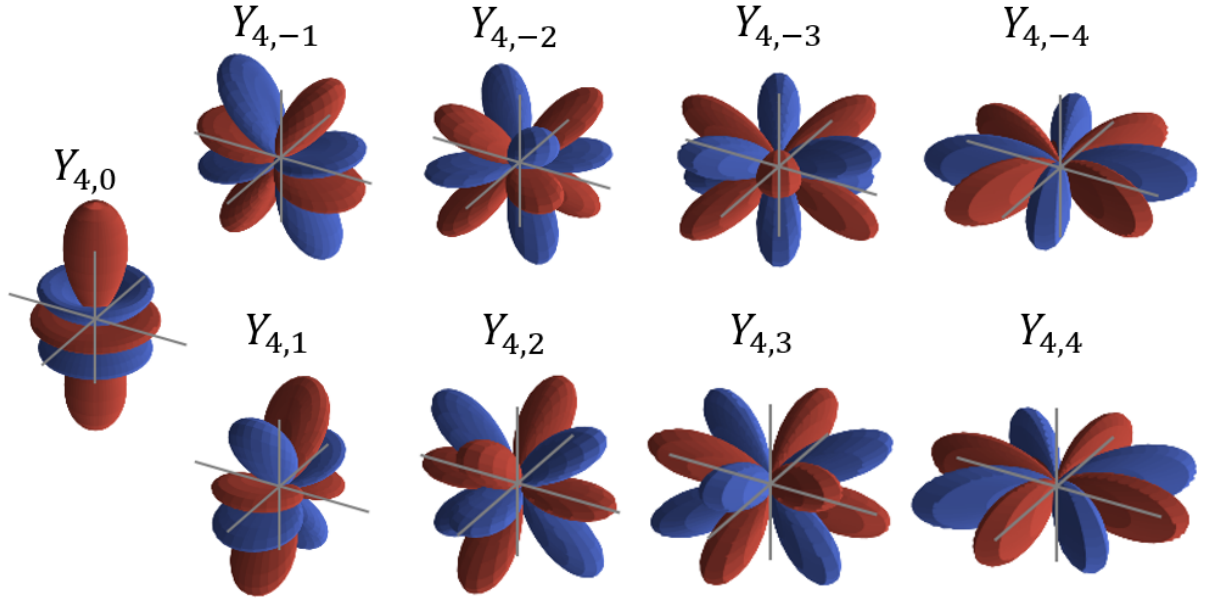


FIGURE 3.8 – Représentation des harmoniques sphériques réelles d'ordre 4.

voisins possible au sens des moindres carré :

$$\begin{aligned} \arg \min_Y \quad & \sum_{t \in T} \sum_{t' \in \mathcal{N}(t)} \sum_{l=-4}^4 (Y_{t',l} - Y_{t,l})^2, \\ \text{sous contrainte faible :} \quad & \forall t \in T_b, \forall l \in [-4, 4], \alpha_{n_t,l}(Y_{t,l} + \beta_{n_t,l} + \gamma_{n_t,l}) = 0 \end{aligned} \quad (3.10)$$

Les valeurs α_{n_t} , β_{n_t} et γ_{n_t} s'obtiennent à partir de ψ_t , la rotation 9D (dans l'espace des harmoniques sphériques) amenant le vecteur $(0, 0, 1)$ vers le vecteur n_t :

$$\begin{aligned} \alpha_{n_t} &= \psi_t(0, 0, 0, 0, \sqrt{\frac{7}{12}}, 0, 0, 0, 0)^T \\ \beta_{n_t} &= \psi_t(\sqrt{\frac{5}{12}}, 0, 0, 0, 0, 0, 0, 0, 0)^T \\ \gamma_{n_t} &= \psi_t(0, 0, 0, 0, 0, 0, 0, 0, \sqrt{\frac{5}{12}})^T \end{aligned} \quad (3.11)$$

Pour chaque tétraèdre t , on obtient (Y_t) un vecteur de dimension 9 dans la base des harmoniques sphériques. La projection des coefficients dans la base des harmoniques sphériques vers un repère 3D orthonormal se fait plus difficilement que dans le cas 2D qui ne nécessitait qu'un "atan". Pour projeter Y_t dans la base des matrices de rotation $SO(3)$, nous partons d'une matrice de rotation E_t , dont l'harmonique sphérique est proche de Y_t , puis nous améliorons itérativement E_t en approchant son harmonique sphérique de Y_t par la méthode du gradient conjugué. Après cette dernière étape de projection, chaque tétraèdre $t \in T$ est doté de trois vecteurs orthonormaux stockés en tant que colonnes de la matrice de rotation $E_t \in SO(3)$.

Matrice de correspondance entre deux repères À partir de notre champ de repère 3D orthogonal $(E_t)_{t \in T}$, nous allons définir les matrices de correspondances $g_{tt'}$ qui portent les informations de la correspondance entre les branches des repères adjacents. La transformation pour passer d'un repère $t \in T$ à un repère adjacent $t' \in T$ peut être décomposée en la rotation $R_{tt'} \in SO(3)$ et la matrice de correspondance $g_{tt'}$:

$$E_{t'} = R_{tt'} g_{tt'} E_t. \quad (3.12)$$

Les matrices de correspondance $g_{tt'}$ prennent en compte le changement de symétrie et appartiennent au groupe de symétrie octaédrique à orientation préservante Γ , dont le cardinal est de 24. La rotation du repère $R_{tt'}$ rend compte de la partie de la rotation qui est indépendante du saut de symétrie. Les deux matrices sont fortement corrélées car, pour chaque matrice de correspondance, la rotation $R_{tt'}$ est définie de manière unique.

Lien entre matrice de correspondance et graphe de singularité Comme noté dans [LIU et al. 2018], le graphe de singularité est entièrement caractérisé par les matrices de correspondance de repère g . Ainsi, pour obtenir les singularités d'un champ de repère, nous devons identifier correctement les symétries entre les repères adjacents. En pratique, étant donné un champ de repère symétrique, les matrices de correspondance sont inconnues et sont calculées dans le sens des moindres carrés :

$$g_{tt'} = \arg \min_{g \in \Gamma} \|E_{t'} - gE_t\|_F^2, \quad (3.13)$$

ce qui implique que $R_{tt'}$ doit être aussi proche que possible de l'identité.

Cependant, sur le bord et surtout le long d'une arête caractéristique, les repères sont fixes, et les correspondances sont fortement contraintes par l'alignement à la fois avec la direction normale et la direction de l'arête caractéristique. Plus précisément, pour une arête de bord $e_{tt'}$ appartenant à deux tétraèdres de bord $t, t' \in T_b$, une colonne de E_t doit être colinéaire à n_t et une colonne de $E_{t'}$ doit être colinéaire à $n_{t'}$. De plus, comme $E_t, E_{t'}$ satisfont l'équation (3.12), la rotation $R_{tt'}$ aligne la colonne de E_t colinéaire à n_t avec une colonne de $E_{t'}$ (ou son opposé). Si ces directions de colonne sont les normales de la face n_t et $n_{t'}$ (ou $-n_t$ et $-n_{t'}$), alors $e_{tt'}$ est une arête régulière de valence 2.

Plus généralement, la valence de bord $k_{tt'}$ du maillage hexaédrique en sortie à l'arête de bord $e_{tt'}$ est directement liée à la rotation $R_{tt'}$ (et à la correspondance $g_{tt'}$) par la formule :

$$k_{tt'} = 2 - \text{round} \left[\frac{2}{\pi} \text{atan2} \left(\langle n_t \times (R_{tt'} n_{t'}), \frac{e_{tt'}}{|e_{tt'}|} \rangle, \langle n_t, R_{tt'} n_{t'} \rangle \right) \right]. \quad (3.14)$$

Calcul d'une carte locale dont le repère est le gradient de direction Les matrices E_t forment des champs de vecteurs que nous allons localement intégrer dans chaque tétraèdre t . À partir de E_t et des coordonnées géométriques des quatre sommets du tétraèdre $t : (a, b, c, d)$, on construit une carte locale linéaire $\mu : \mathbb{R}^3 \mapsto \mathbb{R}^3$ à partir de coordonnées de cartographie sur les quatre coins $i, j, k, l \in \mathcal{C}(t)$ du tétraèdre t :

$$\begin{cases} \mu_i = (0, 0) \\ \mu_j = E_t(b - a) \\ \mu_k = E_t(c - a) \\ \mu_l = E_t(d - a) \end{cases} \quad (3.15)$$

Dans l'étape d'intégration, nous allons utiliser la carte paramétrique locale μ pour transférer les informations géométriques du champ de repère (orientation des hexaèdres), puis nous allons utiliser les matrices de correspondance g pour transférer les informations combinatoires du champ de repère (graphe de singularité du maillage hexaédrique).

3.2.2 Intégration avec Cubecover

Transfert de la géométrie du champ de repère On veut que la carte u soit dans chaque tétraèdre du maillage la plus proche possible de la carte locale μ (à translation près). Similairement à la 2D (Sec 3.1.3), nous pouvons faire en sorte que pour chaque pair de coin i, j d'un

tétraèdre t , la différence $u_i - u_j$ soit la plus proche possible de la différence $\mu_i - \mu_j$.

$$\arg \min_u \sum_{t \in T} \sum_{i, j \in \mathcal{C}(t)} \|(u_i - u_j) - (\mu_i - \mu_j)\|^2 \quad (3.16)$$

Minimiser cette énergie sans fixer aucune règle de continuité de la carte paramétrique u à travers les faces du maillage volumique donne une carte arbitraire identique à la carte μ . Nous allons dans la suite ajouter des contraintes similaires à la 2D (Sec 3.1.3) pour contraindre les singularités de la carte et obtenir une carte sans couture alignée avec les bords.

Contrainte de carte sans couture Pour toute paire de sommets adjacents v, w d'une face commune à deux tétraèdres adjacents t et t' , et en notant $i, j \in \mathcal{C}(t)$ et $i', j' \in \mathcal{C}(t')$ les coins de tétraèdres correspondant aux sommets v, w , on ajoute la contrainte :

$$(u_{i'} - u_{j'}) = g_{tt'} (u_i - u_j). \quad (3.17)$$

Alignement au bord Comme dans le cas 2D, nous avons besoin d'une contrainte d'alignement au bord pour nous assurer que nos maillages hexaédrique en sortie suivent le bord initial. Pour tout triplet de coins de tétraèdre i, j, k sur le bord (i, j, k étant des coins d'un même tétraèdre de bord $t \in T_b$), on ajoute la contrainte :

$$u_{i,0} = u_{j,0} = u_{k,0} \quad (3.18)$$

Il reste maintenant à étendre le principe de quantification présentée en 2D (Sec 3.1.4) à la 3D pour calculer une carte préservant la grille à partir de la carte sans couture obtenue. Une carte préservant la grille valide permet d'extraire un maillage hexaédrique valide.

3.2.3 Quantification de la paramétrisation

Contrainte de carte préservant la grille Pour obtenir une carte préservant la grille (Fig 3.3, droite) il faut s'assurer que les lignes de valeur entière de la carte soient alignées de part et d'autre de chaque face du maillage tétraédrique. Pour ce faire, il faut ajouter à la contrainte 3.6, la contrainte que chaque coin de tétraèdre partageant un même sommet ait des coordonnées paramétriques égales à rotation et translation entière près. Pour tout sommet v partagé par deux tétraèdres t et t' , et en notant $i \in \mathcal{C}(t), i' \in \mathcal{C}(t')$ les coins des tétraèdres t et t' correspondant au sommet v , on ajoute la contrainte :

$$u_{i'} - g_{tt'} u_i = \begin{pmatrix} n_{tt'} \in \mathbb{Z} \\ m_{tt'} \in \mathbb{Z} \\ z_{tt'} \in \mathbb{Z} \end{pmatrix} \quad (3.19)$$

Alignement sur une valeur entière des coins de bord Pour s'assurer que l'extraction d'un maillage hexaédrique par l'extraction des lignes de valeurs entière de la grille 3D produise un maillage aligné avec le bord du maillage tétraédrique de l'entrée, il faut s'assurer que l'alignement de la carte sur le bord dans l'équation 3.7 se fasse sur une valeur entière :

$$u_{i,0} = u_{j,0} = u_{k,0} \in \mathbb{Z} \quad (3.20)$$

Quantification par arrondi Pour les résultats de maillage hexaédrique présentés dans les chapitres 4 et 6 de ce manuscrit, nous avons réalisé cette étape de quantification par arrondi : On calcule les entiers $n_{tt'}$, $m_{tt'}$, $z_{tt'}$ et les entiers de bord en arrondissant les valeurs décimales correspondantes dans la carte sans couture. Puis, on ajoute les contraintes [3.19, 3.20] à partir de ces valeurs arrondies. Cependant, comme dans le cas 2D, arrondir les valeurs à l'entier le plus proche peut produire des retournements dans la carte préservant la grille produite. Lorsque les retournements ne sont pas trop importants, l'extraction de maillage hexaédrique [LYON et al. 2016] parvient à produire des maillages hexaédriques valides. Cependant, pour avoir la garantie de pouvoir extraire un maillage hexaédrique valide, il faut une carte préservant la grille sans retournement.

Quantification avec garantie de non-retournement Des algorithmes de quantification 3D qui trouvent des valeurs entières permettant le calcul d'une carte préservant la grille valide à partir d'une carte sans couture valide ont été publiés pendant le déroulé de ma thèse.

En 2021, l'article [BRÜCKLER et al. 2021] décrit une généralisation du concept de graphe de moto (permettant de construire des maillages en T en 2D) à la configuration volumique tridimensionnelle. Les auteurs proposent des algorithmes pour la construction d'un complexe de moto 3D sur la base de maillages hexaédriques ou de paramétrisations volumiques sans couture. L'utilité de ce concept est illustrée sur des exemples de génération de maillage hexaédrique et de construction de T-spline volumique.

Puis en 2022, l'article [BRÜCKLER et al. 2022] donne une façon de quantifier de manière robuste les paramétrisations volumiques pour le maillage hexaédrique en utilisant une décomposition de cellules non-conformes du volume, qui est un analogue 3D d'un maillage en T. Ils exploitent le complexe de moto [BRÜCKLER et al. 2021], la généralisation du graphe de moto, pour obtenir cette décomposition volumique. Les valeurs entières sont exprimées de manière différentielle sur les arêtes de ce complexe, permettant la formulation efficace des conditions requises pour empêcher strictement la dégénérescence de la cartographie. Les auteurs démontrent que leur méthode est garantie de produire des valeurs entières sans conflit, quelle que soit la résolution.

À partir de ces 2 travaux, voici les étapes pour quantifier une carte sans couture valide et obtenir une carte préservant la grille valide :

1. Construire un complexe de moto à partir de la carte sans couture [BRÜCKLER et al. 2021].
2. Attribuer des longueurs entières aux arcs du complexe de moto via une série de programmes quadratiques entiers [BRÜCKLER et al. 2022].
3. Intégrer les valeurs entières différentielles pour obtenir une quantification valide [BRÜCKLER et al. 2022].
4. Ajouter les contraintes entières de la quantification valide pour calculer une carte préservant la grille valide [BRÜCKLER et al. 2022].

Dans ce chapitre 3, nous avons fait un état de l'art de la méthode Quadcover/Cubecover pour générer des maillages quadrilatères/hexaédriques. Nous allons maintenant présenter dans les chapitres suivants les contributions que nous avons apportées à cette méthode générale pour construire des maillages de modèles CAO utilisables dans le cas de simulation de déformation hyperélastique à maillage unique (Chap 1, 8).

Deuxième partie

Contribution : Contrôle sur le maillage
produit par un champ de repère

Champ de repère non orthogonal

Dans ce chapitre, nous proposons une nouvelle méthode de génération de champ de repère en 2D et en 3D. Les champs de repère orthogonaux de l'état de l'art (Sec 3.1.2, 3.2.1) produisent des correspondances (p_{ij} en 2D, g_{ij} en 3D) qui sont invalides lorsque le modèle en entrée possède un coin à petit angle (Fig 4.4). Plutôt que d'optimiser des champs de repère orthogonaux, nous choisissons d'optimiser des champs de repère non-orthogonaux en construisant une définition adéquate de "lissage d'un champ non-orthogonal". Cette définition de lissage possède un paramètre d'orthogonalité qui donne à l'utilisateur du contrôle sur les maillages produits. Lorsque nos champs non-orthogonaux rencontrent des coins à petit angle, les repères utilisent leur non-orthogonalité pour s'adapter à la forme du modèle, comme le ferait un rectangle qui se déforme en parallélogramme. Ainsi, ces champs de repère produisent des correspondances p_{ij}/g_{ij} valides pour la génération de maillages quads/hex avec la méthode présentée dans le chapitre 3.

Nous montrons l'applicabilité de notre méthode en générant des maillages quadrilatères et hexaédriques anisotropes, qui sont particulièrement utiles pour le remaillage de modèles CAO.

4.1 Introduction

Dans les modèles CAO, il est fréquent de rencontrer des coins à petit angle, et les champs de croix sont incapables de répondre à ces contraintes de conception, comme le montrent les figures 4.4 et 4.11. Plusieurs articles proposent des solutions spécifiques pour échapper à la représentation trop restrictive de champs de repère orthogonaux isotropiques, en créant une nouvelle représentation [DIAMANTI et al. 2014], en déformant le maillage [PANOZZO et al. 2014] ou en modifiant le tenseur métrique [JIANG et al. 2015].

Un objectif commun de la conception de champ de repère est de trouver le champ le plus lisse interpolant entre les contraintes. Une fois la représentation déterminée, le deuxième défi consiste à établir une mesure de lissage géométriquement significative du champ. Cela est crucial, car une mesure qui ne prend pas en compte la géométrie pourrait aboutir à une conception de champ de repère non intuitive.

Le troisième défi consiste à permettre à l'utilisateur de contrôler la représentation choisie. Cela peut inclure la contrainte de certains repères, la contrainte d'une seule direction de certains repères (tout en laissant les autres directions libres de tourner) ou la possibilité de contrôler l'orthogonalité du champ de repère.

Contributions Dans ce chapitre, nous représentons des ensembles de directions (ou champs de direction) sous forme de fonction polynomiale sur un cercle ou une sphère (selon la dimen-

sion). Cette représentation s’inspire directement de la représentation d’harmoniques sphériques de champs de croix 3D proposée dans [HUANG et al. 2011]. En utilisant des polynômes, nous pouvons choisir le nombre de symétries et la forme du repère, comme illustré sur les figures 4.2 et 4.8. Cette représentation permet la rotation indépendante de chaque direction, ce qui crée un espace paramétrique de repères anisotropes non-orthogonaux de n’importe quelle dimension. Contrairement aux méthodes précédentes, la nôtre aborde efficacement les trois défis :

1. La représentation du champ est immédiatement applicable aux maillages surfaciques et volumiques ;
2. Elle fournit une notion de lissage géométriquement significative ;
3. Tout type de champ de direction peut être conçu et un simple paramètre permet de contrôler l’orthogonalité.

Le reste du chapitre est organisé de la manière suivante. La section 4.2 présente une "recette" générale pour concevoir des champs de direction de toute dimension en utilisant une représentation basée sur une somme de polynômes atomiques. Ce cadre théorique est ensuite appliqué à la conception de champ de repère 2D dans la section 4.3 et à la génération de champ de repère 3D dans la section 4.4. Dans ces sections, nous montrons l’utilité de notre représentation pour la génération de maillage quadrilatère et hexaédrique.

4.2 Champ de direction de dimension n en tant que polynômes sphériques

Soit un n -complexe simplicial \mathcal{F} , nous considérons chaque face $f \in \mathcal{F}$ comme un espace tangent discret. Par conséquent, dans ce chapitre, les champs de repère seront constants par morceaux. Sur un maillage triangulaire, nos champs de repère sont constants sur les triangles. Sur un maillage tétraédrique, ils sont constants sur les tétraèdres. Cette configuration est couramment utilisée pour la paramétrisation globale et le maillage structuré [KALBERER et al. 2007] ; [NIESER et al. 2011]. Nous notons \mathcal{E} l’ensemble des paires de faces adjacentes, c’est-à-dire que $i, j \in \mathcal{F}$ sont adjacents si et seulement si $(i, j) \in \mathcal{E}$. Pour imposer des conditions de bord, nous aurons également besoin de $\partial\mathcal{F}$, l’ensemble des faces du bord.

Pour faciliter l’explication, nous nous limitons aux domaines *plats* de \mathbb{R}^n . Cependant, tous les concepts introduits dans ce chapitre peuvent être étendus aux surfaces *courbes* en définissant une notion de transport parallèle comme dans la section 3.1.2.

4.2.1 Représentation d’un seul repère

Nous appelons *ensemble de N -directions* un ensemble non ordonné de N vecteurs tangents *unitaires* et de leurs opposés, c’est-à-dire $\{u_1, -u_1, u_2, -u_2, \dots, u_N, -u_N\}$ (un ensemble de N -direction forme un repère). Un *champ de N -directions* est une affectation d’un ensemble de directions à chaque face. Notre objectif est de trouver le champ de N -directions le plus lisse sur un maillage de triangles ou de tétraèdres qui respecte les contraintes de direction définies par l’utilisateur. Comme un ensemble de directions est non ordonné, il peut être difficile de mesurer la similarité entre deux ensembles car cela nécessite de bien faire correspondre chaque ensemble. Par conséquent, nous avons besoin d’une représentation d’un seul ensemble de directions qui soit invariante par changement de signe et réordonnement. Inspiré par les représentations de champs de repère 2D et 3D introduites dans [HUANG et al. 2011] ; [RAY et al. 2016], nous allons identifier une direction à un polynôme sphérique dont la forme et la symétrie sont identiques.

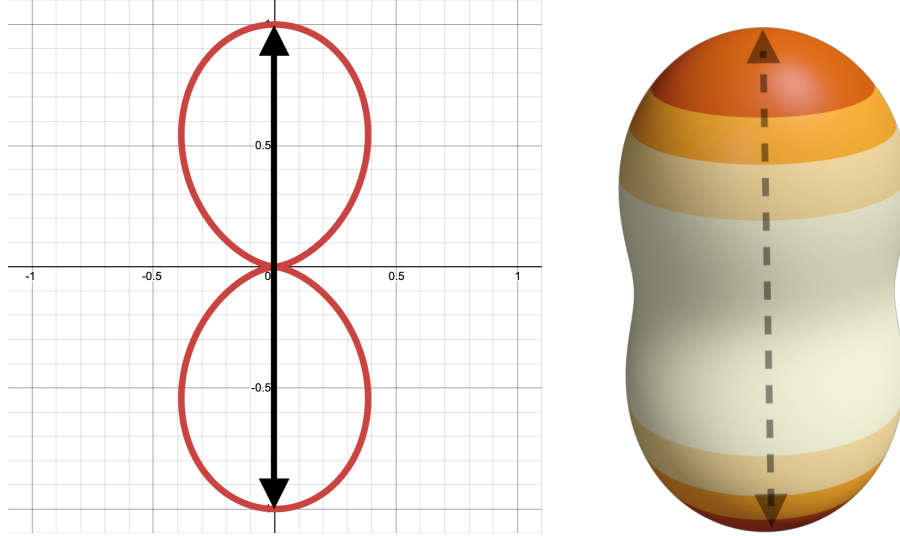


FIGURE 4.1 – Une direction unique (flèches noires) représentée par un polynôme polaire (à gauche) et sphérique (à droite) de degré 2. La valeur du polynôme en chaque point du cercle est représentée par une variation du rayon en coordonnées polaires.

Comme une direction est un ensemble de deux vecteurs opposés, elle doit être représentée par un polynôme de *degré pair* afin de correspondre à la symétrie. En effet, pour un vecteur unitaire u , le polynôme sphérique $p(s) = \langle u, s \rangle^{2k}$ pour $k \geq 1$ entier est indépendant du signe de u . Pour représenter l'ensemble de N -directions $\{u_i, -u_i | i = 1 \dots N\}$, nous additionnons simplement ces polynômes atomiques :

$$p(s) = \sum_{i=1}^N \langle u_i, s \rangle^{2k}. \quad (4.1)$$

La figure 4.1 illustre le polynôme représentant une direction unique en coordonnées polaires et sphériques. Les repères en 2D et en 3D illustrés dans les figures 4.2 et 4.8 pour $k = 2$ sont particulièrement utiles pour les applications de paramétrisation.

Il peut sembler difficile de travailler avec des polynômes polaires ou sphériques, mais cela peut être simplifié en les décomposant en bases orthonormales canoniques.

4.2.2 Espace tangent d'un ensemble de directions

En considérant un ensemble de N directions pour tous les vecteurs unitaires possibles, on obtient un sous-espace polynomial noté $\mathbb{P}_{2k,N}$:

$$\mathbb{P}_{2k,N} := \left\{ \sum_{i=1}^N \langle u_i, s \rangle^{2k} \mid \forall u_i \in \mathbb{S}^n \right\}.$$

L'espace de fonctions $\mathbb{P}_{2k,N}$ ne contient que des polynômes *homogènes* de degré $2k$. En tant que tel, c'est un espace à dimension finie et est engendré par un nombre fini de fonctions orthogonales Y_ℓ . Plus tard, nous utiliserons la base de Fourier pour les champs de directions en 2D et la base des Harmoniques Sphériques pour les champs de directions en 3D. Par conséquent, tout ensemble de N directions $U = \{u_i, -u_i; | i = 1 \dots N\}$ est uniquement représenté par le vecteur

de coefficients $c_\ell(U)$, $\ell = 1, \dots, L$ dans la décomposition dans la base orthogonale :

$$p(s) = \sum_{\ell=1}^L c_\ell(U) Y_\ell(s).$$

Il est crucial de noter que les coefficients $c_\ell(U)$, tout comme le polynôme $p(s)$, ne dépendent pas de l'ordonnement ou du changement de signe des vecteurs dans l'ensemble U . Ils ne dépendent que de la direction des vecteurs dans cet ensemble. Cette représentation simplifie considérablement le calcul de la distance entre deux ensembles de N directions, car il suffit de calculer la distance entre leur représentation polynomiale. Les polynômes de représentation ont les mêmes formes que les ensembles (voir les figures 4.2 et 4.8 pour un exemple de champ de repère), ce qui donne à cette distance un sens géométrique clair.

En utilisant le fait que la base Y_ℓ est orthogonale, la distance entre un polynôme p_1 représentant l'ensemble de directions U_1 et p_2 représentant l'ensemble de directions U_2 est simplement la norme de la différence des coefficients c :

$$\begin{aligned} \|p_1 - p_2\|_2^2 &= \int_{\mathbb{S}^n} (p_1(s) - p_2(s))^2 ds \\ &= \sum_{\ell=1}^L (c_\ell(U_1) - c_\ell(U_2))^2. \end{aligned}$$

4.2.3 Champ de polyvecteur polynomial constant par morceaux

Un champ de N directions consiste en un ensemble U_i assigné à chaque face $i \in \mathcal{F}$. Notre représentation remplace ces ensembles U_i par les coefficients $c_\ell(U_i)$, où $\ell = 1, \dots, L$. Notre objectif est de concevoir des champs de N directions lisses. Un champ de N directions est considéré comme parfaitement lisse sur un bord s'il est égal à son polynôme, et donc à ses coefficients c . Pour refléter cette idée, nous introduisons une énergie de lissage, analogue à l'énergie de Dirichlet discrète, en accumulant la distance entre les repères adjacents :

$$E_s(U) = \sum_{(ij) \in \mathcal{E}} \sum_{\ell=1}^L (c_\ell(U_i) - c_\ell(U_j))^2. \quad (4.2)$$

Le fait de concevoir des champs de directions lisses qui interpolent des contraintes revient à minimiser l'énergie de lissage E_s tout en imposant des contraintes de direction directement sur l'ensemble de vecteurs U . Cela rend les problèmes de conception beaucoup plus simples à résoudre qu'auparavant.

4.2.4 Degré du polynôme de représentation

Le degré du polynôme est un paramètre qui a une influence sur la forme du repère et sur la complexité du problème d'optimisation. Il existe une limite inférieure pour le degré du polynôme. Dans [SHEN et al. 2016], les auteurs ont étudié l'utilisation de polynômes sphériques pour représenter des ensembles de N directions répartis de manière uniforme en 3D. Ils ont constaté qu'un polynôme de degré au moins 4 était nécessaire pour représenter 3 directions orthogonales, et qu'un polynôme de degré 6 était requis pour représenter 6 directions formant les sommets d'un dodécaèdre régulier. De plus, pour que les N directions d'un ensemble de N directions puissent être modifiées indépendamment, le polynôme doit être de degré suffisamment

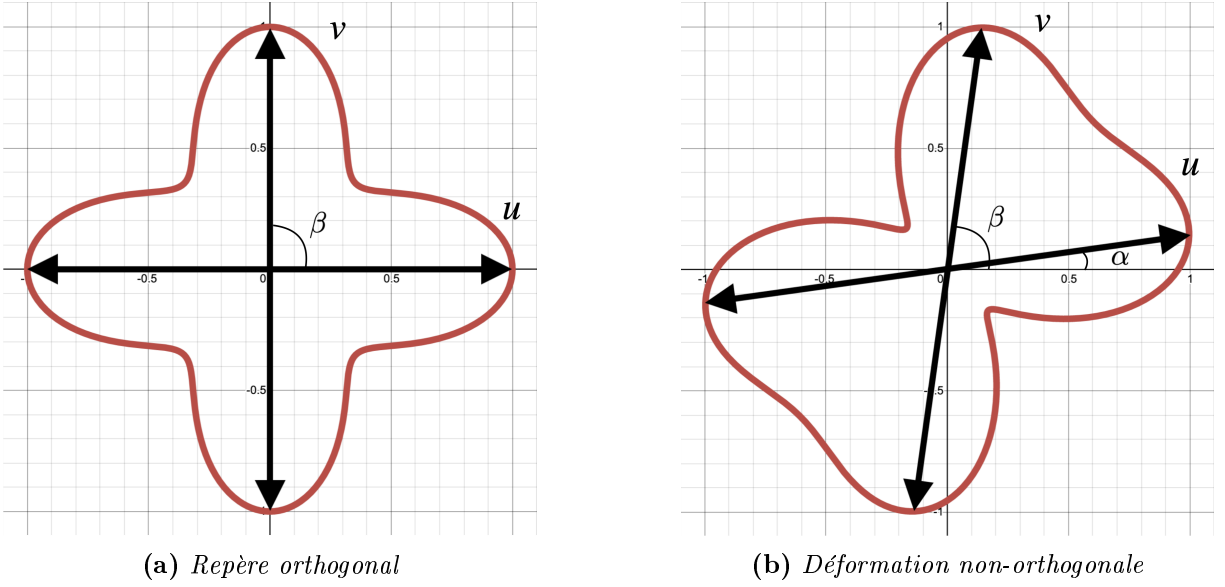


FIGURE 4.2 – Un ensemble de 2-Directions $\{u, -u, v, -v\}$ (flèches noires) représenté par un polynôme polaire de degré quatre.

élevé pour représenter ces variations. Cependant, la complexité des coefficients c augmente avec le degré du polynôme, ce qui rend la minimisation de l'énergie de lissage dans l'équation 4.2 de plus en plus difficile.

Dans la suite de ce chapitre, nous nous concentrons sur le cas particulier des champs de repère en 2D et en 3D. Nous choisissons le degré polynomial le plus bas acceptable, qui est de degré 4 ou $k = 2$ dans l'équation (4.1). Les figures 4.2b et 4.8b montrent que des polynômes de degré 4 sont suffisants pour capturer les déformations d'un repère 2D ou 3D.

4.3 Champs de repère en 2D

Dans cette section, nous nous intéressons aux champs de repère (ou champs de directions 2D) sur des domaines en 2D. Ces champs ont été introduits pour la première fois dans [PANOZZO et al. 2014] et sont très utiles pour le maillage quadrilatère. Pour un domaine en 2D, le polynôme de représentation p est défini sur un cercle et l'ensemble des repères tangents admissibles $\mathbb{P}_{2k,2}$ est décomposé sur la base de Fourier. Nous allons voir que, avec ce choix de base et lorsque les deux directions sont orthogonales, notre représentation est exactement égale à la représentation complexe des champs de repère utilisée dans de nombreux articles [PALACIOS et al. 2007] ; [RAY et al. 2008] ; [KNOPPEL et al. 2013] ; [DIAMANTI et al. 2014] et présentée dans la section 3.1.2.

4.3.1 Une direction en tant que polynôme de degré 4

Comme mentionné dans [RAY et al. 2016], les champs de repère 2D peuvent être représentés par un polynôme polaire de degré quatre : $p(s) = s_1^4 + s_2^4$. C'est en fait le polynôme de plus bas degré présentant la même symétrie qu'une croix. Pour cette raison, nous utiliserons des polynômes de degré 4 pour représenter notre champ de repère.

Un repère en 2D est l'ensemble de deux directions $\{u, -u, v, -v\}$ paramétrées respectivement par les angles α et β , tels que $u = (\cos \alpha, \sin \alpha)$ et $v = (\cos \beta, \sin \beta)$. Ces notations sont résumées

par la figure 4.2. Suivant l'équation (4.1), notre polynôme de représentation de degré quatre s'écrit comme la somme :

$$p(s) = \langle u, s \rangle^4 + \langle v, s \rangle^4. \quad (4.3)$$

La figure 4.2 donne une illustration de la forme d'un repère orthogonal (fig.4.2a) et d'un repère non-orthogonal telle que représentée par le polynôme polaire de l'équation (4.3).

4.3.2 Base de Fourier

Le polynôme de représentation de l'équation (4.3) est défini sur un cercle unitaire. Il semble naturel d'utiliser la base de Fourier comme base orthogonale pour décomposer ce signal périodique. Nous noterons $c_{\ell,1}(\alpha, \beta)$ les coefficients correspondant aux fonctions de base $\cos(\ell t)$ et $c_{\ell,2}(\alpha, \beta)$ les coefficients correspondant aux fonctions de base $\sin(\ell t)$ où $\ell \in \mathbb{N}$ est l'indice de fréquence.

La base sinus/cosinus apparaît naturellement lorsque l'on exprime $s = (\cos t, \sin t)$ dans le système de coordonnées polaires. Le polynôme de direction en u s'écrit simplement :

$$\begin{aligned} \langle u, s \rangle^4 &= |\cos t \cos \alpha + \sin t \sin \alpha|^4 \\ &= \frac{3}{4} + \frac{1}{2} \cos(2t) \cos(2\alpha) + \frac{1}{2} \sin(2t) \sin(2\alpha) \\ &\quad + \frac{1}{8} \cos(4t) \cos(4\alpha) + \frac{1}{8} \sin(4t) \sin(4\alpha) \end{aligned}$$

Ainsi, les coefficients de la décomposition du polynôme représentant un repère se réduisent à :

$$\begin{cases} c_{2,1}(\alpha, \beta) = 4 \cos(2\alpha) + 4 \cos(2\beta) \\ c_{2,2}(\alpha, \beta) = 4 \sin(2\alpha) + 4 \sin(2\beta) \\ c_{4,1}(\alpha, \beta) = \cos(4\alpha) + \cos(4\beta) \\ c_{4,2}(\alpha, \beta) = \sin(4\alpha) + \sin(4\beta) \end{cases} \quad (4.4)$$

Notre repère est représenté par seulement quatre coefficients en fréquences $\ell = 2$ et $\ell = 4$. De manière cruciale, le repère est orthogonal (*c'est-à-dire* $\beta = \alpha + \pi/2$) si et seulement si les coefficients d'ordre deux sont nuls. Dans ce cas, un repère est uniquement représenté par le vecteur $(\cos(4\alpha), \sin(4\alpha))$, ainsi nous retrouvons la représentation complexe classique des champs orthogonaux [PALACIOS et al. 2007] ; [RAY et al. 2008].

4.3.3 Conception de champs de repère en 2D pour maillages quads

Dans cette section, nous établirons un problème d'optimisation afin de trouver le champ de repère le plus lisse interpolant entre des contraintes de direction imposées par l'utilisateur. Un paramètre permettra de contrôler l'orthogonalité du champ.

Variables Pour faciliter la résolution du problème d'optimisation, nous n'utilisons pas directement l'angle α, β ou le vecteur u, v comme variables. Au lieu de cela, nous utilisons les vecteurs à double angle $\bar{u} = (\cos(2\alpha), \sin(2\alpha))$ et $\bar{v} = (\cos(2\beta), \sin(2\beta))$. Le vecteur \bar{u} est mappé sur u ou $-u$, donc le changement de variables est valide. Avec ces nouvelles variables, les coefficients de

l'équation (4.4) sont maintenant :

$$\begin{cases} c_{2,1}(\bar{u}, \bar{v}) = 4\bar{u}_1 + 4\bar{v}_1 \\ c_{2,2}(\bar{u}, \bar{v}) = 4\bar{u}_2 + 4\bar{v}_2 \\ c_{4,1}(\bar{u}, \bar{v}) = 2\bar{u}_1^2 + 2\bar{v}_1^2 - 2 \\ c_{4,2}(\bar{u}, \bar{v}) = 2\bar{u}_1\bar{u}_2 + 2\bar{v}_1\bar{v}_2 \end{cases} \quad (4.5)$$

Contraintes de direction La contrainte la plus courante pour la conception de champs de repère consiste à imposer des repères constants sur certaines faces. Dans notre cas, définir ces contraintes est simple : il suffit de calculer l'angle de chaque direction cible et de contraindre \bar{u} et \bar{v} aux vecteurs à double angle.

Une contrainte plus intéressante, souvent utilisée pour le maillage quadrilatère non-orthogonal (voir fig. 4.5), consiste à fixer uniquement une direction sur deux. Pour ce faire, on peut contraindre soit \bar{u} soit \bar{v} à la direction souhaitée. Notez que le choix de \bar{u} ou \bar{v} n'a pas d'impact sur les résultats car la représentation polynomiale est invariante par permutation des directions. Contraindre \bar{u} ou \bar{v} n'affecte pas le second vecteur du repère qui peut tourner librement.

Cependant, lorsqu'une seule direction est fixée, la seconde direction peut devenir dégénérée si aucune contrainte de norme unitaire ne l'oblige à être non nulle. Pour éviter ce phénomène, nous ajoutons, uniquement pour les triangles de bord, une énergie de norme unitaire :

$$E_b(\bar{u}, \bar{v}) = \sum_{i \in \partial\mathcal{F}} (|\bar{u}_i|^2 - 1)^2,$$

où \bar{u} est la direction non contrôlée. Cette énergie est pondérée par le paramètre b .

Pour calculer des maillages quadrilatères à partir de nos champs de repère, nous imposons aux repères sur les bords d'avoir une direction tangente à l'arête du bord. Si un triangle a deux arêtes de bord, alors le repère est entièrement contrôlée. Dans l'introduction, nous mettons en avant le défi de générer des maillages quads sur des modèles avec des coins aigus comme dans les figures 4.4 et 4.11. Pour capturer ces caractéristiques géométriques, il est important que les directions soient entièrement contrôlées lorsque plusieurs arêtes caractéristiques se rencontrent selon des angles aigus ou obtus.

L'énergie de lissage En suivant la Section 4.2.3, l'énergie de lissage E_s sur un champ de repère 2D dépend uniquement des coefficients de la décomposition Eq. (4.5). Nous modifions légèrement l'énergie pour ajouter un paramètre λ qui équilibre la fréquence d'ordre 2 :

$$\begin{aligned} E_s(\bar{u}, \bar{v}) = & \lambda \sum_{(ij) \in \mathcal{E}} \sum_{m=1,2} (c_{2,m}(\bar{u}_i, \bar{v}_i) - c_{2,m}(\bar{u}_j, \bar{v}_j))^2 \\ & + \sum_{(ij) \in \mathcal{E}} \sum_{m=1,2} (c_{4,m}(\bar{u}_i, \bar{v}_i) - c_{4,m}(\bar{u}_j, \bar{v}_j))^2. \end{aligned}$$

Le paramètre d'orthogonalité λ Comme mentionné dans la Section 4.3.2, l'orthogonalité est directement liée aux fréquences d'ordre 2. Si le champ possède des singularités, en raison de la topologie du domaine, les repères sont forcés à être orthogonaux. En effet, un repère non-orthogonal possède uniquement une symétrie de rotation de π . Pour qu'une singularité apparaisse, le champ doit soit tourner et changer d'angle en même temps, soit avoir des singularités d'indice $1/2$. Ces deux solutions sont nettement plus coûteuses en termes d' E_s que d'avoir des repères orthogonaux. Par conséquent, l'introduction d'un simple poids λ sur les coefficients d'ordre 2

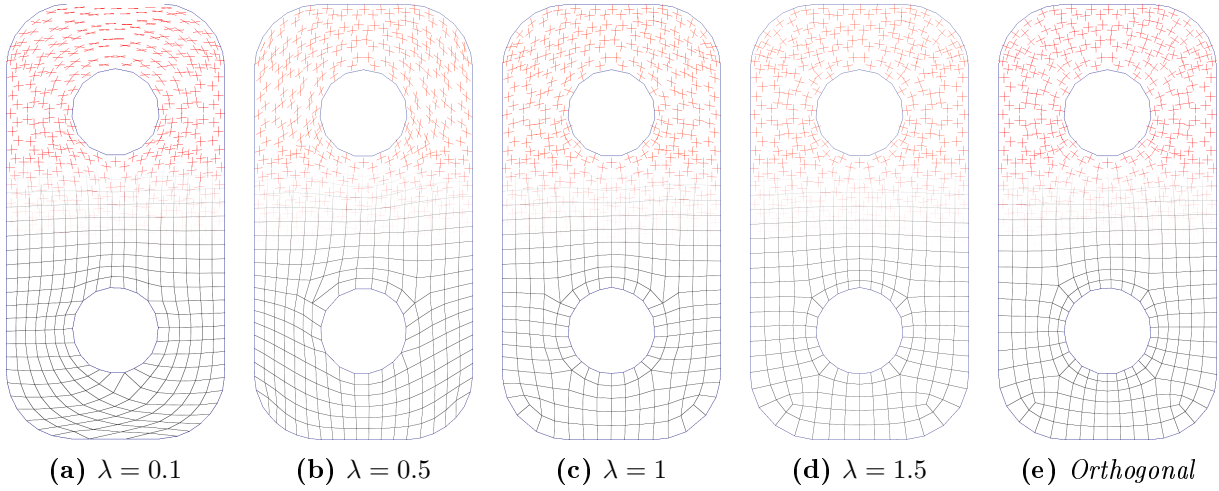


FIGURE 4.3 – Résultats de notre méthode de conception de champ de repère pour des valeurs croissantes du paramètre d’orthogonalité λ . Des valeurs faibles (à gauche) poussent les singularités en dehors du domaine tandis que des valeurs élevées se rapprochent du champ de repère obtenu par Viertel et al. [VIERTEL et al. 2019] (à droite).

favorise naturellement les repères orthogonaux autour des singularités et, par extension, sur toute la forme.

L’effet du paramètre d’orthogonalité λ sur le champ de repère et les maillages quadrilatères est illustré dans la Figure 4.3. La valeur $\lambda = 1$ correspond à la notion de lissage *géométrique* expliquée dans la Section 4.2.3. Une valeur supérieure à 1 favorise l’orthogonalité et le champ de repère devient de plus en plus similaire à un champ de repère orthogonal. Un λ inférieur à 1 donne plus de liberté au champ de repère qui peut être dégénéré. En conséquence, les singularités sont repoussées sur les bords du domaine.

Le choix de ce paramètre est dicté par l’application et résulte souvent d’un compromis. D’une part, les repères orthogonaux conduisent à des maillages quadrilatères avec des éléments proches de carrés parfaits mais ne peuvent pas traiter les coins aigus, omniprésents dans les modèles CAO. D’autre part, les champs de repère non orthogonaux compensent les caractéristiques aiguës en introduisant une torsion et, dans les cas extrêmes, en enlevant les singularités de l’intérieur du domaine.

Optimisation Notre schéma d’optimisation est basé sur l’algorithme MBO introduit dans [VIERTEL et al. 2019] pour optimiser la position des singularités des champs de repère. L’idée est d’atténuer la contrainte de norme unitaire sur \bar{u} et \bar{v} en alternant entre une étape de diffusion et une étape de renormalisation. Plus précisément, la paire de variables $(\bar{u}, \bar{v})_{n+1}$ à l’itération $n+1$ est obtenue en minimisant l’énergie de lissage tout en restant proche de l’itération n :

$$(\bar{u}, \bar{v})_{n+1} = \arg \min_{x,y} E_s(x, y) + bE_b(x, y) + \tau \|(x, y) - (\bar{u}, \bar{v})_n\|_2^2, \quad (4.6)$$

puis en renormalisant $(\bar{u}, \bar{v})_{n+1}$ sur une norme unitaire. Ce schéma est celui qui nous donne les meilleurs résultats en terme de lissage du champ et de placement des singularités. L’étape de minimisation est effectuée avec LBFGS [ZHU et al. 1997].

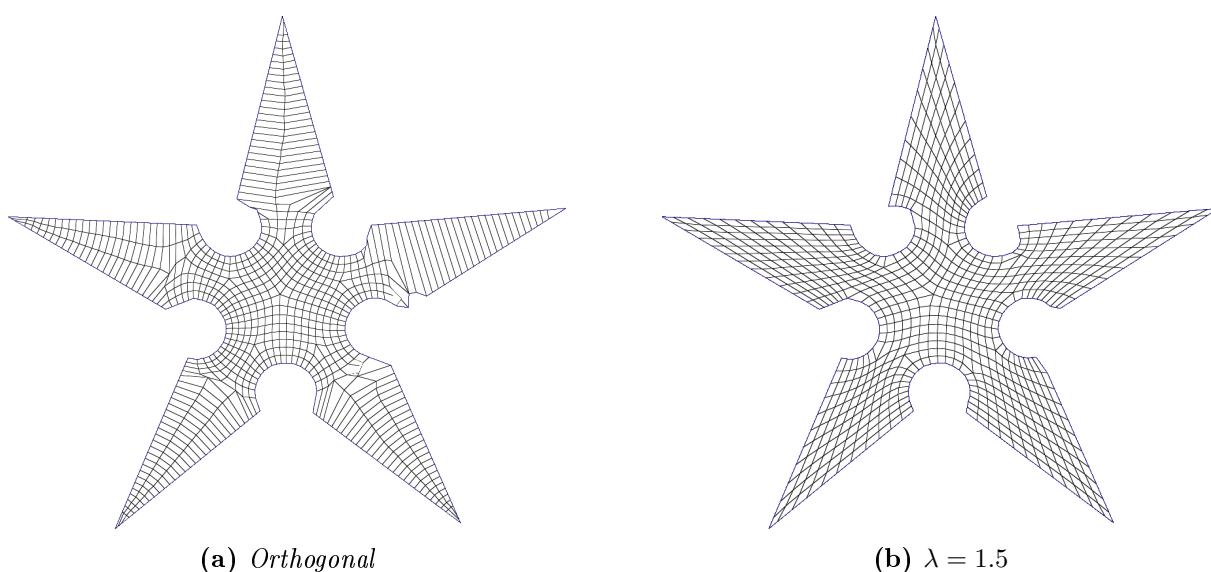


FIGURE 4.4 – Un champ de repère orthogonal (4.4a) ne peut pas être utilisé pour générer un maillage quadrangulaire pour un modèle avec des coins aigus. Notre champ de repère non orthogonal (4.4b) permet une solution de bonne qualité.

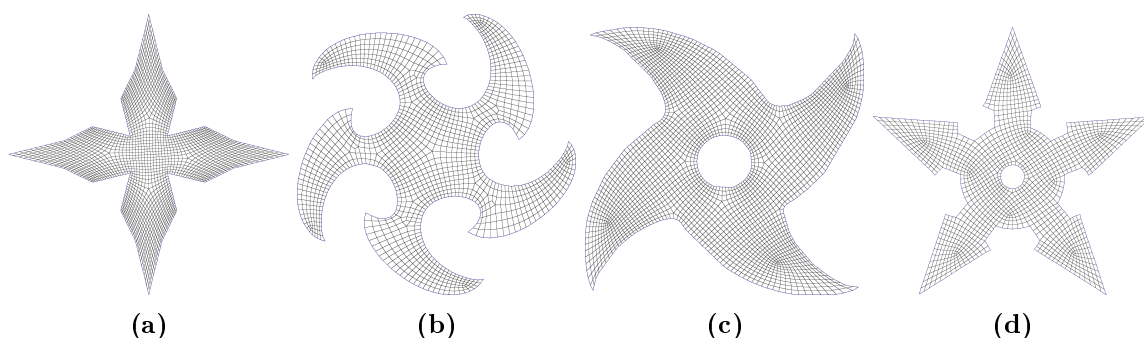


FIGURE 4.5 – Notre méthode permet la construction de maillages quadrilatères de bonne qualité en ajustant le paramètre $\lambda = 1.5$.

4.3.4 Résultats

Nous évaluons l'utilisabilité de notre méthode de conception de champs de repère dans le contexte de génération de maillages quads. Tous les champs de repère sont contraints à avoir une direction tangente au bord du domaine. Pour éviter les solutions de norme nulle, l'énergie promouvant la norme unitaire au bord est pondérée par $b = 10$. Dans toutes nos expériences, le paramètre de diffusion est fixé à $\tau = 1/2$. Le paramètre d'orthogonalité λ est accordé en fonction de nos besoins, mais la valeur 1.5 s'avère être un bon compromis dans la plupart des cas.

Le problème d'optimisation étant non convexe, une bonne initialisation est cruciale. Nous initialisons avec un champ de repère orthogonal [RAY et al. 2008] pour éviter les solutions complètement dégénérées. Le choix du champ initial et son impact sur la solution de l'optimisation sont discutés dans le cas tridimensionnel dans la section 4.4.3, mais la conclusion s'applique également en 2D. Notre optimisation a généralement besoin de 10 itérations pour converger.

Les maillages quadrilatères sont extraits d'un champ de repère à l'aide des méthodes de l'état de l'art [KALBERER et al. 2007] ; [BOMMES et al. 2013a].

Fig.	Faces	Init	Opti	Meshing
4.3	658	0.017	<1	2
4.4b	1282	0.026	<1	3
4.5c	9968	0.177	26	13
4.5a	18430	0.355	66	30
4.5d	29000	0.611	120	45
4.5b	36047	0.775	161	46

TABLE 4.1 – Temps de calcul en secondes pour chaque étape de notre méthode : initialisation avec le champ de repère orthogonal de [RAY et al. 2008], résolution de notre problème d’optimisation non convexe et extraction du maillage quadrilatère avec le chapitre 3.

Maillage quads pour les modèles CAO Le maillage quadrilatère de modèles CAO est rendu difficile par la présence de coins à petit angle : à ces endroits, un carré parfait ne peut pas s’adapter. La figure 4.4 illustre ce problème. Le champ de repère orthogonal de la figure 4.4a est incapable de satisfaire les contraintes de chaque côté du coin à petit angle, ce qui entraîne l’échec de la production d’un maillage quadrilatère valide. Notre méthode génère un maillage quadrilatère anisotrope (figure 4.4b) qui s’adapte naturellement aux contraintes de coin à petit angle. La figure 4.5 montre une collection de maillages quadrilatères générés à partir de nos champs de repère non-orthogonaux, qui permettent de satisfaire de manière adaptée les contraintes de bord avec un petit nombre de singularités tout en évitant les quadrilatères très dégénérés.

Timings Le tableau 4.1 montre les temps d’exécution de chaque étape de notre méthode pour chaque figure du document. L’étape la plus coûteuse en temps est le calcul du champ de repère non-orthogonal, car elle nécessite de résoudre un problème d’optimisation non convexe. Notre algorithme est implémenté en C++ mais n’a pas été optimisé pour atteindre de hautes performances. Par exemple, nous avons remarqué qu’il est souvent inutile d’attendre la convergence de l’étape LBFGS et que l’optimisation peut être arrêtée plus tôt. En outre, une méthode de Newton pourrait être utilisée. Toutes les expériences sont effectuées sur un ordinateur portable avec un Intel Core i5-10400H quadricœur à 2,6 GHz et 32 Go de RAM.

4.3.5 Comparaison avec PolyVector field [DIAMANTI et al. 2015]

Dans cette sous-section, nous comparons notre méthode à celle de Diamanti et al. [DIAMANTI et al. 2015], qui utilise une représentation des champs de repère sous forme de coefficients d’un polynôme complexe. Comme notre représentation, la leur est indépendante de l’ordre du jeu de directions. De plus, ils introduisent une énergie favorisant l’intégrabilité du champ, c’est-à-dire de telle sorte que chaque direction soit le gradient d’une fonction. Comme nous, ils utilisent directement les vecteurs de direction comme variables dans leur schéma numérique. Ainsi, leur énergie d’intégrabilité pourrait être ajoutée immédiatement à notre optimisation.

Distance entre deux repères En mettant de côté les aspects d’optimisation numérique, la principale différence réside dans l’énergie de lissage. La représentation utilisée par Diamanti et al. [DIAMANTI et al. 2015] obscurcit la signification géométrique de la représentation du repère, ce qui conduit les auteurs à chercher un lissage des coefficients du polynôme complexe plutôt qu’un lissage du champ. Dans notre notation, la représentation polyvectorielle d’un repère unitaire se

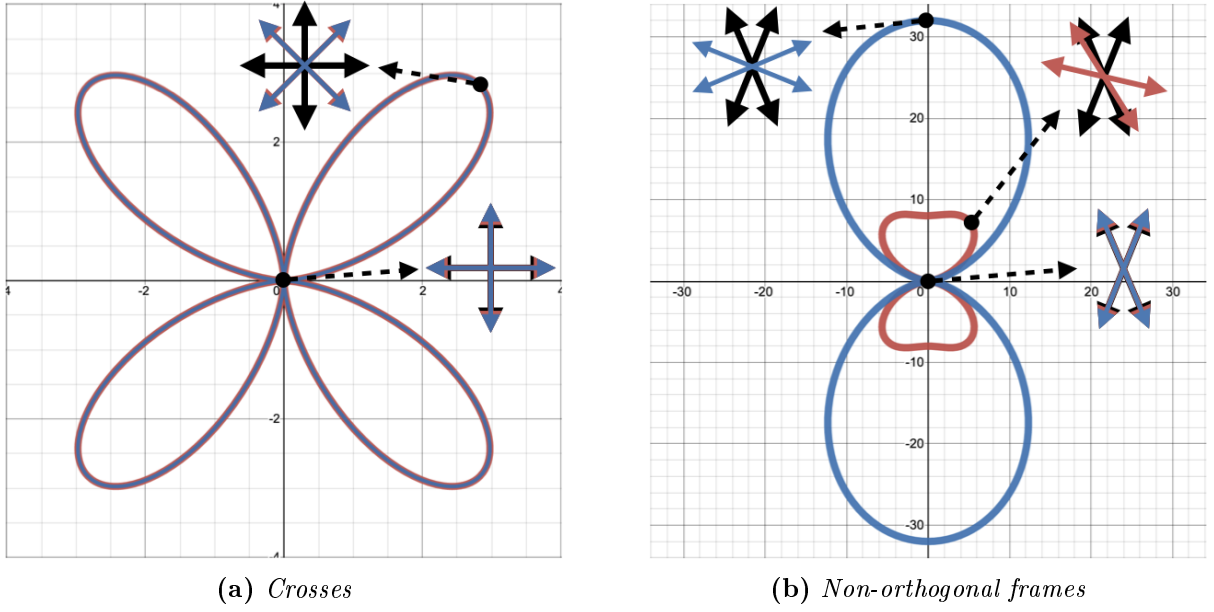


FIGURE 4.6 – Diagramme polaire de la distance entre deux repères orthogonaux (figure 4.6a) et deux repères non-orthogonaux (figure 4.6b) par rapport à l’angle de rotation. La distance définie par notre représentation est tracée en bleu et la distance de PolyVector [DIAMANTI et al. 2014] en rouge. Les encadrés mettent en évidence la position des repères aux extrémités locales. La distance de PolyVector n’est pas géométriquement correcte car la distance maximale est atteinte pour des repères non-orthogonaux qui se chevauchent.

résume à remplacer nos coefficients c par \tilde{c} :

$$\begin{cases} \tilde{c}_{2,1}(\alpha, \beta) = 2 \cos(2\alpha) + 2 \cos(2\beta) \\ \tilde{c}_{2,2}(\alpha, \beta) = 2 \sin(2\alpha) + 2 \sin(2\beta) \\ \tilde{c}_{4,1}(\alpha, \beta) = 2 \cos(2\alpha + 2\beta) \\ \tilde{c}_{4,2}(\alpha, \beta) = 2 \sin(2\alpha + 2\beta) \end{cases} . \quad (4.7)$$

Si les coefficients du second ordre correspondent à ceux de notre équation (4.4), les coefficients du quatrième ordre sont significativement différents. Pour comparer la distance définie par nos coefficients c et les coefficients \tilde{c} de l’équation (4.7), nous avons tracé la distance entre un repère et toutes ses rotations dans la figure 4.6. Pour les repères orthogonaux, les deux distances sont identiques (figure 4.6a). Pour les repères non-orthogonaux, les distances diffèrent significativement sur deux aspects. Tout d’abord, notre distance a des valeurs globalement plus élevées, de sorte que, avec $\lambda = 1$, nous pénalisons davantage le changement d’angle entre les repères que l’article PolyVector [DIAMANTI et al. 2014]. En second lieu, la distance de PolyVector [DIAMANTI et al. 2014] n’atteint pas son maximum à l’angle $\pi/2$, mais à un angle où les repères se chevauchent (figure 4.6b).

Comparaison de résultats En utilisant le changement de variable $(\alpha, \beta) \rightarrow (\bar{u}, \bar{v})$ dans l’équation (4.7) :

$$\begin{cases} \tilde{c}_{2,1}(\bar{u}, \bar{v}) = 2\bar{u}_1 + 2\bar{v}_1 \\ \tilde{c}_{2,2}(\bar{u}, \bar{v}) = 2\bar{u}_2 + 2\bar{v}_2 \\ \tilde{c}_{4,1}(\bar{u}, \bar{v}) = 2\bar{u}_1\bar{v}_1 - 2\bar{u}_2\bar{v}_2 \\ \tilde{c}_{4,2}(\bar{u}, \bar{v}) = 2\bar{u}_1\bar{v}_2 + 2\bar{u}_2\bar{v}_1 \end{cases} ,$$

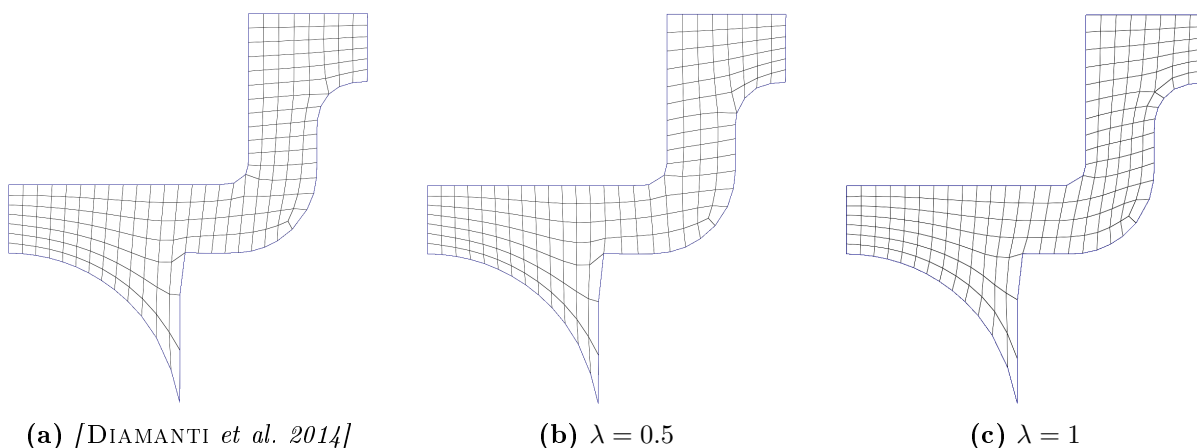


FIGURE 4.7 – L'énergie de lissage du champ PolyVector [DIAMANTI et al. 2015] obtient des résultats similaires aux nôtres en terme de placement des singularités pour $\lambda = 0, 5$.

nous pouvons générer les mêmes champs de repère que PolyVector [DIAMANTI et al. 2014] en remplaçant c par \tilde{c} dans notre énergie de lissage E_s .

La figure 4.7 compare des maillages quadrilatères générés à partir de nos champs de directions non-orthogonaux avec celui obtenu à partir des coefficients \tilde{c} . Nous évaluons notre méthode avec les paramètres $\lambda = 0.5$ et $\lambda = 1$, tous les autres paramètres étant égaux. Comme prévu avec $\lambda = 1$, notre méthode conduit à un maillage quadrilatère avec une singularité supplémentaire dans la partie supérieur droite qui n'est pas présent dans la méthode PolyVector [DIAMANTI et al. 2015]. Nous avons constaté que les deux méthodes donnent des résultats similaires lorsque nous réglons λ sur 0.5. En effet, bien que la distance de repères PolyVector ne suivent pas une intuition géométrique, la différence avec notre énergie de lissage est la plus marquée lorsque les repères sont éloignés l'un de l'autre. Par conséquent, cela n'a pas d'impact considérable sur la conception de champs lisses, car les repères adjacents sont proches les uns des autres.

4.4 Champs de repère 3D

Dans le cas particulier des champs de repère 3D, chaque repère a trois directions notées u, v, w . Au lieu de la base de Fourier, l'espace polynomial $\mathbb{P}_{2k,3}$ est décomposé avec la base des harmoniques sphériques. Lors de l'utilisation de polynômes de degré 4, notre représentation de champ de repère 3D nécessite uniquement des fonctions d'harmoniques sphériques de quatrième ordre, tout comme la représentation de repère 3D la plus utilisée [HUANG et al. 2011].

Comme indiqué dans [HUANG et al. 2011] ; [RAY et al. 2016], les champs de repère 3D peuvent être représentés par le polynôme sphérique de degré quatre : $p(s) = s_1^4 + s_2^4 + s_3^4$. Il s'agit en fait du polynôme de plus faible degré présentant la même symétrie qu'un repère 3D. Cependant, des polynômes d'ordre supérieur de degré pair pourraient être considérés pour la génération de champs de repère. Afin de maintenir la complexité de calcul aussi faible que possible, nous utiliserons des polynômes de degré 4.

Dans ce cas, le polynôme de représentation adapté de l'équation (4.1) s'écrit :

$$p(s) = \langle u, s \rangle^4 + \langle v, s \rangle^4 + \langle w, s \rangle^4, \quad (4.8)$$

et est tracé dans deux configurations dans la figure 4.8.

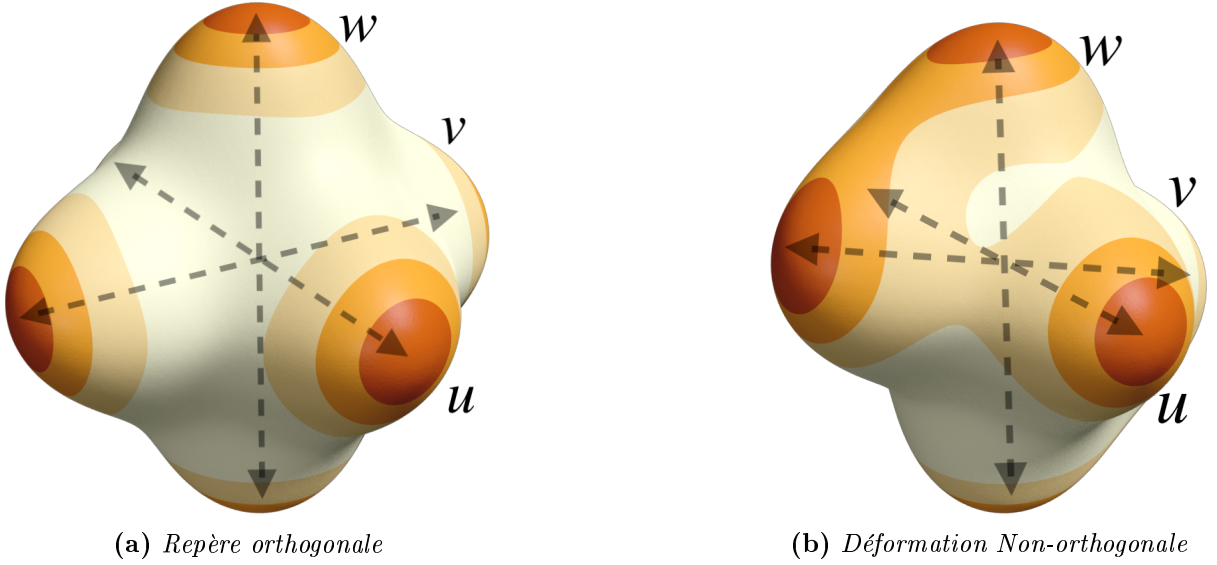


FIGURE 4.8 – Ensemble de 3-directions $\{u, -u, v, -v, w, -w\}$ représentés par des polynômes sphériques de degré 4.

4.4.1 Décomposition en harmoniques sphériques

Les fonctions d'harmoniques sphériques constituent une base orthogonale pour les fonctions sphériques.

Les fonctions d'harmoniques sphériques $Y_\ell^m : \mathbb{S}^2 \rightarrow \mathbb{R}$ sont identifiées par deux indices : ℓ l'indice de bande, correspondant à la fréquence du signal, et m couvrant toutes les fonctions à la fréquence ℓ . Toute fonction sphérique $f : \mathbb{S}^2 \rightarrow \mathbb{R}$ est décomposée de façon unique comme suit :

$$f(s) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} f_{\ell,m} Y_\ell^m(s).$$

Harmoniques zonales La décomposition du polynôme de représentation de l'équation (4.8) dans la base des Harmoniques Sphériques (SH) est facilitée en raison de leur invariance par rotation. En particulier, le polynôme sphérique $f : s \mapsto s_3^4$, qui représente la direction z , est invariant par rotation autour de l'axe z . Ainsi, sa décomposition en SH n'utilise que les fonctions Y_ℓ^0 , $\ell \geq 0$:

$$f(s) = 63Y_0^0 + 36Y_2^0(s) + 8Y_4^0(s).$$

Ces fonctions sont appelées *fonctions harmoniques zonales* et ont une propriété fondamentale : leur rotation vers un axe unitaire quelconque u peut être facilement calculée dans la base SH [SLOAN 2008] :

$$\langle u, s \rangle^4 = 63Y_0^0 Y_0^0 + 36 \sum_{m=-2}^2 Y_2^m(u) Y_2^m(s) + 8 \sum_{m=-4}^4 Y_4^m(u) Y_4^m(s).$$

Ainsi, la décomposition du polynôme de champ de repère de l'équation (4.8) peut se faire à partir des coefficients :

$$\begin{cases} c_{2,m}(u, v, w) = Y_2^m(u) + Y_2^m(v) + Y_2^m(w), \\ c_{4,m}(u, v, w) = Y_4^m(u) + Y_4^m(v) + Y_4^m(w). \end{cases} \quad (4.9)$$

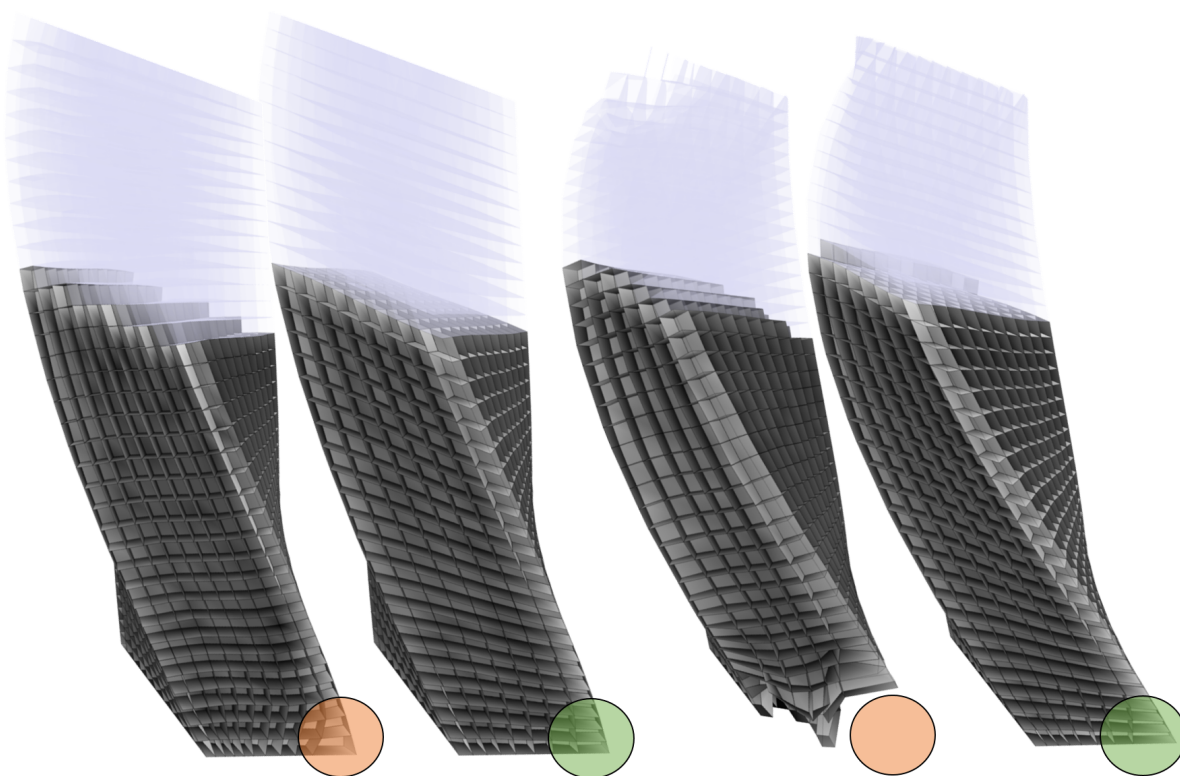


FIGURE 4.9 – Carré extrudé avec rotation. Le premier résultat utilise un champ orthogonal, le second utilise nos champs. Les deux images suivantes ajoutent simplement plus de rotation à l’extrusion. Les coins sont mieux capturés avec nos champs. La couche d’héxaèdres à l’intérieur du volume reste également parallèle aux extrémités.

Tout comme les repères 2D, les repères 3D n’ont que des coefficients dans les bandes $\ell = 2$ et $\ell = 4$. On peut vérifier à partir de l’expression des SH que le repère avec une longueur uniforme (*i.e.* $|u| = |v| = |w|$) est orthogonal si et seulement si tous les coefficients $c_{2,m}$ sont nuls. Ainsi, pour les repères orthogonaux, nous retrouvons la propriété de base utilisée lors de la conception de champs de croix 3D [HUANG et al. 2011] ; [RAY et al. 2016] : toutes les rotations du polynôme $\sum_{i=1,2,3} s_i^4$ sont couvertes par les SH de la bande 4. Cependant, l’équation (4.9) est plus précise et donne la relation exacte entre les coefficients SH et la déformation du repère.

4.4.2 Construction de champs de repère 3D avec contraintes

Contrairement au cas 2D, il n’y a pas de changement de variables évident simplifiant l’expression des coefficients de l’équation (4.9). Par conséquent, nous conservons les trois vecteurs unitaires $u, v, w \in \mathbb{S}^2$ comme variables. Comme dans les travaux précédents [HUANG et al. 2011] ; [RAY et al. 2016] ; [PALMER et al. 2020], la réalisation de calculs dans l’espace de champs de direction nécessite de résoudre un problème d’optimisation non convexe.

Contraintes de direction Les contraintes de direction sont facilement définies en contraignant directement u, v ou w . Nous pouvons fixer une, deux ou trois directions à la fois. Le choix de s’il faut fixer u, v ou w n’a pas d’impact sur les résultats car notre représentation est indépendante de l’ordre du jeu de directions. Cela est très pratique pour les contraintes de bords

dans les problèmes de paramétrisation globale.

Dans toutes nos expériences, un repère associé à une face du bord a ses directions contraintes par des vecteurs normaux au bord. Si un tétraèdre a un, deux ou trois triangles de bord, le repère doit satisfaire une, deux ou trois contraintes de direction.

Énergie de lissage L'énergie de lissage est une simple distance entre les coefficients de la décomposition de deux repères adjacents, comme expliqué dans la section 4.2.3 :

$$E_s(u, v, w) = \lambda \sum_{(ij) \in \mathcal{E}} \sum_{m=-2}^2 (c_{2,m}(u_i, v_i, w_i) - c_{2,m}(u_j, v_j, w_j))^2 + \sum_{(ij) \in \mathcal{E}} \sum_{m=-4}^4 (c_{4,m}(u_i, v_i, w_i) - c_{4,m}(u_j, v_j, w_j))^2.$$

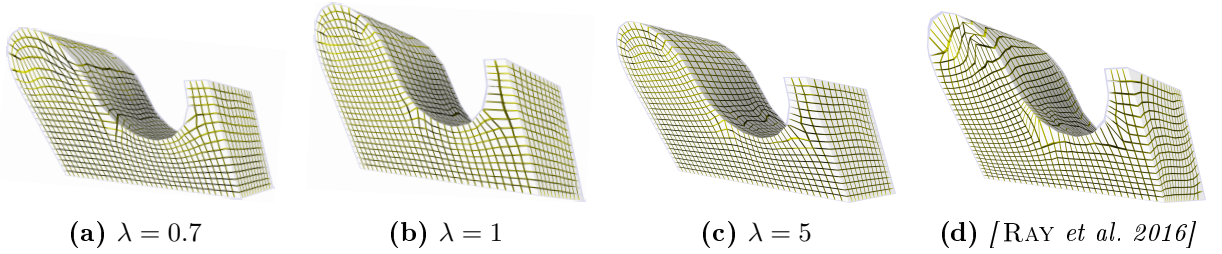


FIGURE 4.10 – Résultats de notre méthode de conception de champ de repère 3D pour des valeurs croissantes du paramètre d'orthogonalité λ . Des valeurs faibles (à gauche) poussent les singularités en dehors du domaine tandis que des valeurs élevées approchent la paramétrisation de champ de croix 3D obtenue par Ray et al. [RAY et al. 2016] (à droite).

Paramètre d'orthogonalité Comme mentionné dans la section 4.4.1, l'orthogonalité du repère est liée aux coefficients de la seconde bande. Nous pénalisons donc les coefficients de bande $\ell = 2$ dans l'énergie avec un poids λ . Augmenter λ signifie qu'il est plus difficile pour le champ de changer son angle et favorise les champs avec un angle constant entre ses directions. Comme les singularités ont une énergie plus faible lorsque le champ est orthogonal autour des singularités, ce simple paramètre sur les coefficients d'ordre 2 force l'orthogonalité globale du champ.

La figure 4.10 illustre l'effet du paramètre λ sur la génération de maillages hexaédriques. Des valeurs inférieures à un génèrent une paramétrisation de type polycube dans laquelle les singularités se trouvent aux bords du domaine, tandis que des valeurs supérieures à un s'approchent de la solution de champ de croix générée par [RAY et al. 2016].

4.4.3 Maillage hexaédrique de modèle CAO

Nous évaluons la qualité de nos champs de repère 3D par leur capacité à générer des maillages hexaédriques. Dans toutes nos expériences, les maillages hexaédriques sont extraits du champ de repère à l'aide des méthodes de l'état de l'art présentées dans le chapitre 3. Les repères orthogonaux sont générés à l'aide de [RAY et al. 2016].

L'optimisation se fait de la même manière que pour les champs de repère 2D. Nous répétons la minimisation de l'équation (4.6) avec des paramètres de diffusion $\tau = 1, b = 100$ et une renormalisation des vecteurs u, v et w .

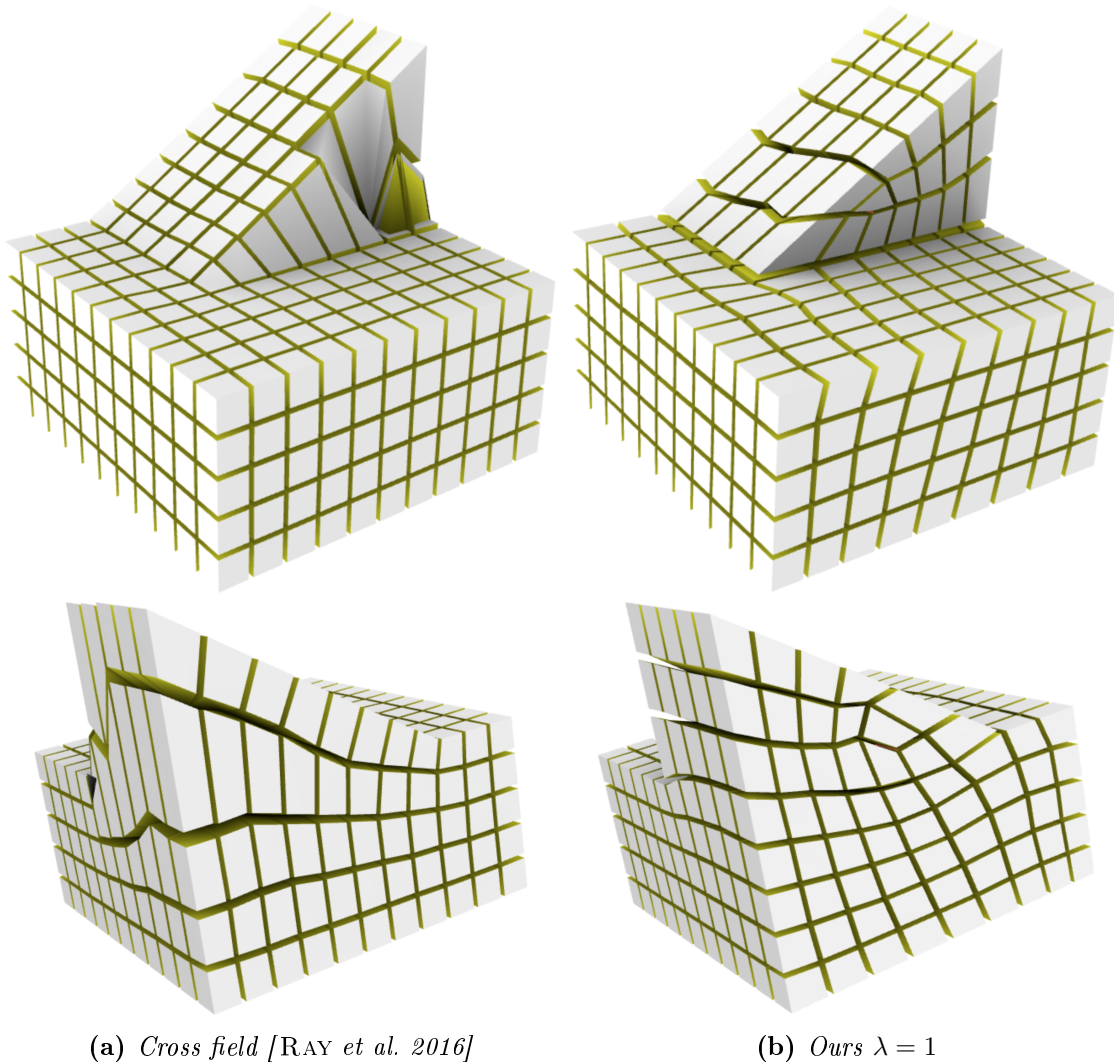


FIGURE 4.11 – Sur un modèle CAO comportant un petit angle, un champ de repère orthogonal 3D provoque un écrasement d'une partie du modèle (Fig. (4.11a)). Notre méthode ajoute avec succès une singularité pour compenser la pente (Fig. (4.11b)).

La présence de coins à petit angle est un défi classique pour les méthodes de maillage hexaédrique. Cette situation se produit fréquemment pour les modèles CAO et est souvent mal gérée par les champs de croix qui échouent à placer des hexaèdres dans les coins comme le montre la figure 4.11a. Au contraire, le champ de direction conçu par notre méthode introduit une singularité, et un maillage hexaédrique valide peut être extrait (figure 4.11b). Pour obtenir de tels résultats, il est primordial que les repères soient entièrement contraints sur les coins à petit angle.

La figure 4.9 met également en lumière la nécessité de champs de repère non orthogonaux pour capturer les coins à petit angle. Les champs de croix génèrent des maillages hexaédriques avec des éléments fortement distordus ou manquants, tandis que notre méthode est capable de produire des hexaèdres de bonne qualité dans le coin.

Dans la figure 4.12, la présence d'un cisaillement oblige les repères orthogonaux à osciller pour satisfaire les contraintes de bords, tandis que notre champ de repère peut rester constant.

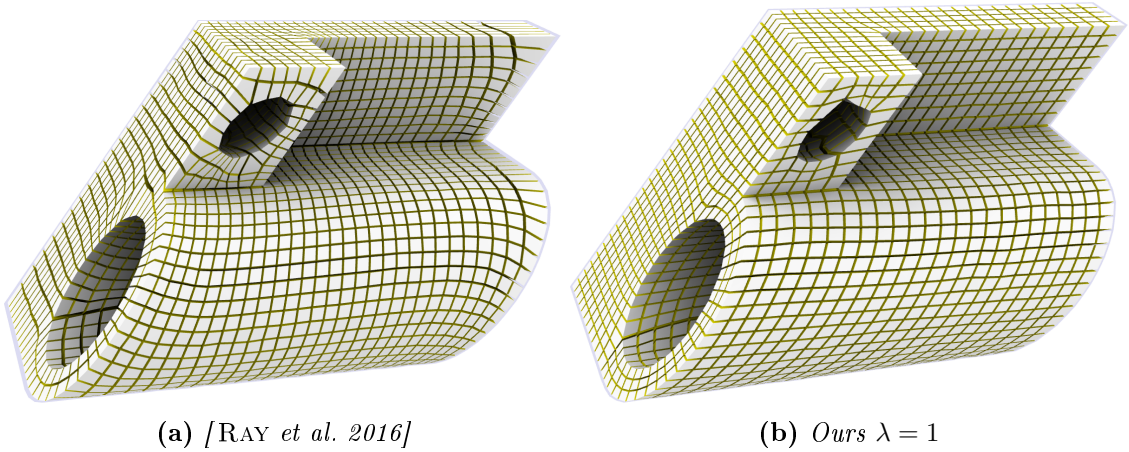


FIGURE 4.12 – Sur un modèle CAO présentant un fort cisaillement, un champ de repère orthogonal 3D crée des oscillations inutiles (Fig. (4.12a)). Notre méthode produit des hexaèdres bien formés et réguliers (Fig. (4.12b)).

Initialisation L'énergie E_s est non-convexe, de sorte que le résultat de sa minimisation dépend de l'initialisation. La figure 4.13 compare les solutions de notre problème d'optimisation pour trois points de départ différents. L'initialisation avec un champ de repère orthonormal aléatoire converge toujours vers la même solution (Fig. 4.13b). Une initialisation avec un champ de repère lisse [RAY et al. 2016] a besoin de moins d'itérations pour atteindre la convergence mais la solution est toujours la même (Fig. 4.13a). Comme indiqué dans le Tableau 4.2, cela réduit (parfois considérablement) le temps de calcul. Cependant, notre schéma d'optimisation tombe dans un minimum local lorsqu'il est initié avec un champ aléatoire avec des repères presque plats (Fig. 4.13c). À la convergence, l'énergie totale est supérieure à la solution optimale. De plus, le champ est corrompu par des repères dégénérés (entourés en rouge) ce qui le rend inutile pour les applications de maillage.

Dans toutes nos expériences, nous initialisons l'optimisation avec le champ de repère orthogonal lisse obtenu par [RAY et al. 2016].

Timings Le Tableau 4.2 liste le temps de calcul pour générer un champ de repère non-orthogonal avec notre méthode. Encore une fois, l'étape la plus coûteuse en temps est le calcul du champ de repère non-orthogonal car il nécessite de résoudre un problème d'optimisation non-convexe. Toutes les expériences sont menées sur un ordinateur portable avec un processeur Intel Core i5-10400H à 4 cœurs et 2,6 GHz et 32 Go de RAM. Tous les algorithmes sont implémentés en C++.

Ce chapitre a présenté une méthode pour représenter un polyvecteur de dimension quelconque par un polynôme pour construire une énergie d'optimisation. Cette méthode nous a donné une représentation de repère 2D et 3D non-orthogonal par un polynôme, ce qui nous a permis d'optimiser des champs de repère non-orthogonaux, de manière à ce que chaque repère soit le plus proche possible de son voisin malgré des contraintes difficiles d'alignement au bord. Nous avons utilisé ces champs non-orthogonaux pour produire des maillages quadrilatères et hexaédriques sur des modèles CAO présentant des coins à petit angle, ce qu'un champ orthogonal traditionnel (Sec 3.1.2, 3.2.1) ne permet pas en 2D comme en 3D. Malheureusement, pour des modèles CAO

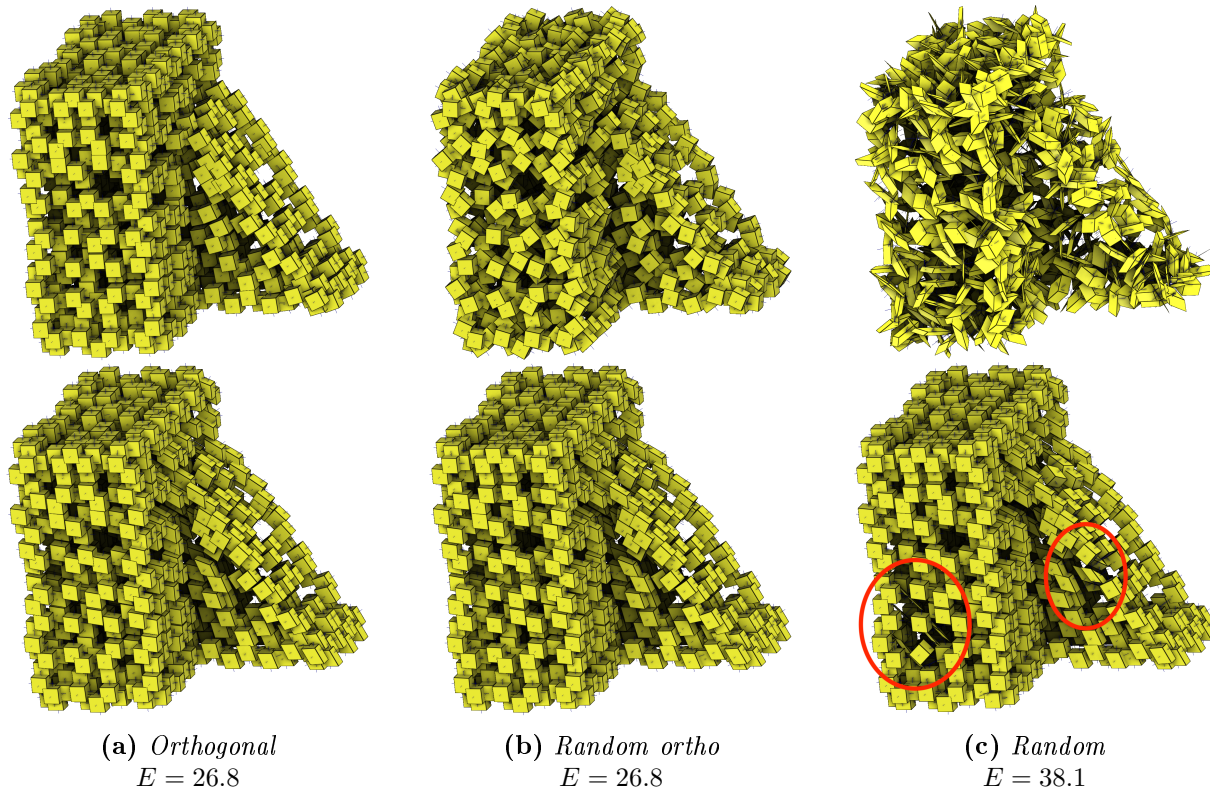


FIGURE 4.13 – Résultats de notre méthode de conception de champ de repère 3D à partir de trois initialisations différentes. Partir d'un champ de repère orthogonal lisse [RAY et al. 2016] (Fig.4.13a en haut) ou d'un champ orthogonal aléatoire (Fig.4.13b en haut) conduit à la même solution (Fig.4.13a–Fig.4.13b en bas). L'initialisation avec un champ complètement aléatoire, y compris des repères dégénérés, tombe dans un minimum local dégénéré, entraînant une énergie totale E plus élevée (Fig. 4.13c).

industriels composés de plusieurs millions de triangles ou tétraèdres, l'optimisation d'un champ non-orthogonal demande un temps de calcul trop long (jusqu'à 1000 fois plus important que les champs orthogonaux en 3D). Pour cette raison, nous allons dans les deux prochains chapitres résoudre les problèmes de coin à petit angle dans les modèles CAO avec des méthodes plus rapides : le chapitre 5 pour le cas 2D, et le chapitre 6 pour le cas 3D.

Fig.	Tets	Init	Opti	Meshing
4.13a	862	0.016	22	0.7
4.13b	862	–	25	0.7
4.13c	862	–	26	–
4.9	4420	0.096	31	3.0
4.10	13225	0.349	454	8.4
4.12b	10788	0.332	305	7.7

TABLE 4.2 – Temps de calcul en seconde pour chaque étape de notre méthode de conception de champ de repère 3D : initialisation avec un champ orthogonal [RAY et al. 2016], résolution de notre problème d’optimisation non-convexe et extraction de maillage hexaédrique en utilisant le chapitre 3.

Champ de repère lisse selon une définition adaptée à la CAO

Nous avons vu dans le chapitre précédent que les champs de repère orthogonaux présentés dans le chapitre 3 ne peuvent pas être utilisés pour générer des maillages quadrilatères ou hexaédriques si le modèle en entrée possède un coin à petit angle. Le chapitre 4 présente une définition de lissage d'un champ non-orthogonal pour régler ce problème. Dans ce chapitre, nous étudions une autre façon de définir le lissage d'un champ de repère orthogonal 2D, de façon adaptée au modèle en entrée pour éliminer le problème des coins à petit angle pour les champs orthogonaux (Fig 5.1). Cette méthode est restreinte à la 2D, mais permet une optimisation beaucoup plus rapide que les optimisations non-linéaires des champs non-orthogonaux du chapitre 4. Le principe est de modifier la définition du transport parallèle que nous avons vu dans le cas 2D surfacique dans la section 3.1.2. Dans la définition de rotation entre deux triangles voisins $r_{tt'} = \theta'_t - \theta_t - \omega_{tt'} \pmod{\frac{\pi}{2}}$, nous ajoutons à $\omega_{tt'}$ une valeur $\gamma_{tt'}$ proche des coins à petit angle, pour faire comme si l'angle du coin était de 90° dans le modèle d'entrée. Cette méthode de champ de repère a un paramètre de diffusion du transport parallèle à partir de chaque coin à petit angle qui donne du contrôle sur les maillages quadrilatères produits avec quadcover (Chap 3).

5.1 Introduction

Comme présenté dans le chapitre 3, calculer un champ de repère orthogonal lisse sur un maillage triangulaire permet de construire un graphe de singularité de maillage quadrilatère. Cependant le graphe de singularité obtenu n'a aucune garantie d'être valide, et en pratique la méthode d'optimisation de champ de repère orthogonal présentée dans la section 3.1.2 échoue à produire des singularités valides lorsque le modèle triangulaire en entrée possède un coin à petit angle. Plutôt que de complètement changer la façon d'optimiser les champs de repères, en faisant par exemple une optimisation de champ non-orthogonal comme dans le chapitre 4, nous allons ajouter une étape redéfinissant le transport parallèle du maillage triangulaire en entrée avant de lancer l'optimisation du champ de repère. Dans le cas surfacique, le transport parallèle est défini à partir de valeurs ω dépendant entièrement de la géométrie de la surface du maillage triangulaire de l'entrée (Sec 3.1.2). Lorsque deux arêtes caractéristiques se rencontrent sur un modèle surfacique, on peut calculer l'angle formé par ces arêtes grâce au transport parallèle défini par ω . Lorsque cet angle est inférieur à 90 degrés, l'idée est de modifier le transport parallèle pour que la mesure de l'angle soit exactement 90 degrés. Tout le challenge de ce chapitre est de déterminer pour chaque paire de triangles voisins t, t' des valeurs $\gamma_{tt'}$ à ajouter aux valeurs

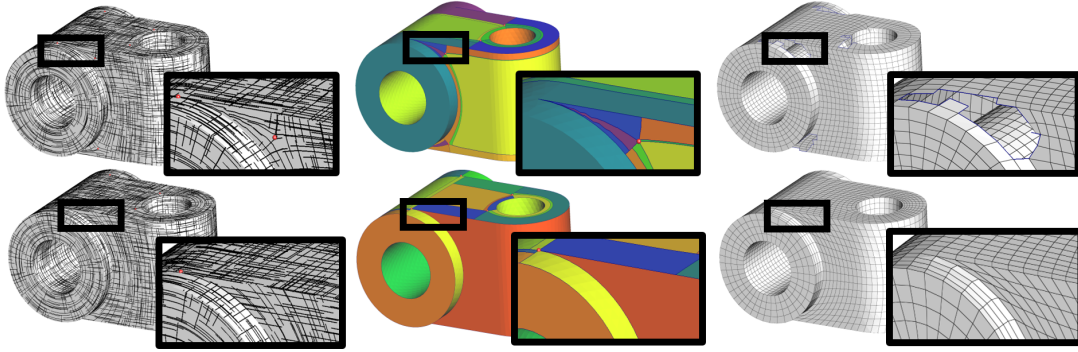


FIGURE 5.1 – Produire un champ de repère qui a une rotation minimale sur un modèle CAO ne suffit pas toujours pour obtenir un maillage quad de qualité (haut). Notre algorithme permet de gérer les cas où il y a des configurations d'arêtes caractéristiques formant des petits angles (bas).

$\omega_{tt'}$ pour obtenir un modèle qui a fictivement tous ses angles supérieurs à 90 degrés, sur lequel optimiser un champ de repère orthogonal. Une fois ces valeurs γ obtenus, l'optimisation de champ de repère se fait de manière très similaire à ce que nous avons vu dans la section 3.1.2 :

$$\begin{aligned} \arg \min_{X,Y} \quad & \sum_{t \in T} \sum_{t' \in \mathcal{N}(t)} \left\| \begin{pmatrix} X_{t'} \\ Y_{t'} \end{pmatrix} - \begin{pmatrix} \cos(4\omega_{tt'} + 4\gamma_{tt'}) & -\sin(4\omega_{tt'} + 4\gamma_{tt'}) \\ \sin(4\omega_{tt'} + 4\gamma_{tt'}) & \cos(4\omega_{tt'} + 4\gamma_{tt'}) \end{pmatrix} \begin{pmatrix} X_t \\ Y_t \end{pmatrix} \right\|^2, \\ \text{sous contrainte : } \quad & \forall t \in T_b, \begin{pmatrix} X_t \\ Y_t \end{pmatrix} = \begin{pmatrix} \cos 4\eta_t \\ \sin 4\eta_t \end{pmatrix}. \end{aligned} \quad (5.1)$$

Ensuite, le calcul des correspondances du champ de repère (Sec 3.1.2) utilise également cette nouvelle définition du transport parallèle pour produire un graphe de singularité valide proche des coins à petit angle :

$$p_{tt'} = \text{round} \left(\frac{\theta_{t'} - \theta_t - \omega_{tt'} - \gamma_{tt'}}{\pi/2} \right)$$

5.2 Calcul de γ

Dans cette section, nous présentons d'abord la manière très locale de calculer des valeurs γ garantissant un graphe de singularité valide proche des coins à petit angle, puis nous présentons une manière de diffuser γ de manière uniforme avec un paramètre de distance pour obtenir du contrôle sur le positionnement des singularités induites par des coins à petit angle.

Une solution très locale Le but de cette correction est de placer une singularité juste à côté du coin à petit angle. Plus précisément, pour chaque sommet de bord v avec un angle de bord $\theta_v < \pi/2$, nous fixons la somme des $\gamma_{tt'}$ à la valeur $\pi/2 - \theta_v$ pour tout triangle $t, t' \in T$ voisins et partageant v : $d\gamma(v) = \pi/2 - \theta_v$, où $d\gamma(v) = \sum_{t, t' \in \mathcal{N}(v)} \gamma_{tt'}$ désigne la dérivée extérieure de γ , étendue au bord de la surface. Construire γ de cette façon permet que l'énergie de lissage basée sur le transport parallèle $\omega + \gamma$ soit identique à celui de l'énergie de lissage d'un maillage surfacique avec un coin à angle 90° , qu'un champ de repère orthogonal gère très bien.

Cependant, en modifiant seulement le transport parallèle dans le 1-voisinage de v , le sommet v' adjacent à v va devenir une singularité recevant beaucoup de distorsion (Fig. 5.2). Cette

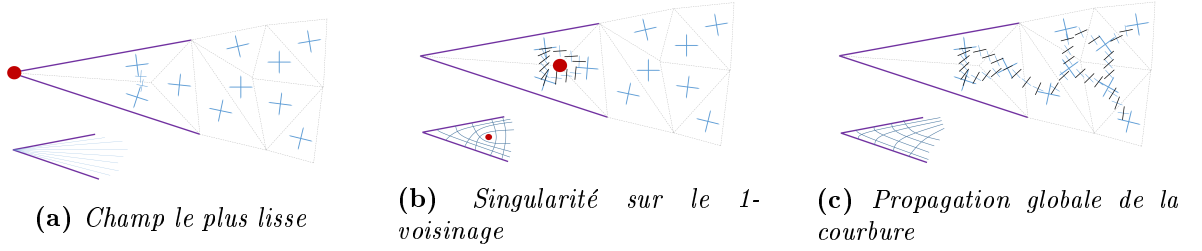


FIGURE 5.2 – Sur un sommet de coin pointu, le champ le plus lisse produit des lignes de quads qui se concentrent au niveau du sommet (a). Une correction locale peut résoudre ce problème (b), mais la propagation de la courbure du champ est ce qui permet d’obtenir les lignes de quads les plus utilisables (c).

solution locale peut donc produire des résultats de mauvaise qualité si un sommet très proche d’un coin (du maillage triangulaire initial) devient une singularité (appartenant au maillage quadrilatère de sortie). Pour diffuser le transport parallèle de manière uniforme et selon un paramètre de distance contrôlé par l’utilisateur, nous faisons une optimisation globale de γ .

Une solution plus globale Au lieu de forcer le sommet le plus proche à devenir une singularité, il est souvent préférable que le sommet singulier soit plus loin (Fig. 5.2 droite). Pour ce faire, il faudrait que la modification du transport parallèle soit propagée plus loin que seulement sur les arêtes adjacentes au sommet problématique v . Nous utilisons une méthode d’optimisation en deux étapes pour déterminer γ :

- On calcule un scalaire $K(v)$ pour chaque sommet v de sorte que $\sum_v K(v)^2$ soit minimal, avec $K(v) = \pi/2 - \theta_v$ pour les sommets de coin pointus v , et $\sum_v K(v) = 0$. La solution consiste à répartir la somme des $K(v)$ des sommets de coin pointus sur les autres sommets. Cette optimisation est effectuée indépendamment pour chaque composante connectée de la surface (qui peut être générée lors de la découpe de la surface le long des arêtes caractéristiques).
- Ensuite, nous minimisons $\arg \min \sum_{tt'} |\gamma_{tt'}|^2$ sous la contrainte $d\gamma(v) = K(v)$. La contrainte $\sum_v K(v) = 0$ garantit l’existence d’une solution.

En pratique, nous initialisons $K(v)$ en le définissant comme égal à $\pi/2 - \theta_v$ pour les sommets de coin pointus et à 0 pour les autres sommets. Ensuite, pour chaque sommet de coin pointu v_s , nous calculons l’ensemble des sommets qui peuvent être atteints selon une distance géodésique prédéfinie par l’utilisateur. Soit n la taille de cet ensemble de sommets, nous mettons à jour K pour tous les sommets de cet ensemble en utilisant la formule suivante : $K(v) \leftarrow K(v) - K(v_s)/n$. Nous optimisons ensuite les valeurs de γ en résolvant le problème $\arg \min \sum_{ij} |\gamma_{ij}|^2$ sous la contrainte $d\gamma(v) = K(v)$.

Ce contrôle sur la propagation de γ permet d’obtenir des maillages quadrilatères de bonne qualité (Fig 5.2).

5.3 Relaxation de l’orthogonalité

Plutôt que de réoptimiser un champ orthogonal à partir des $p_{tt'}$ comme dans l’équation 3.3, on peut relaxer la contrainte d’orthogonalité en représentant un repère non-orthogonal par deux angles θ_t, ϕ_t . Cela nous permet d’obtenir une géométrie de champ de repère non-orthogonale

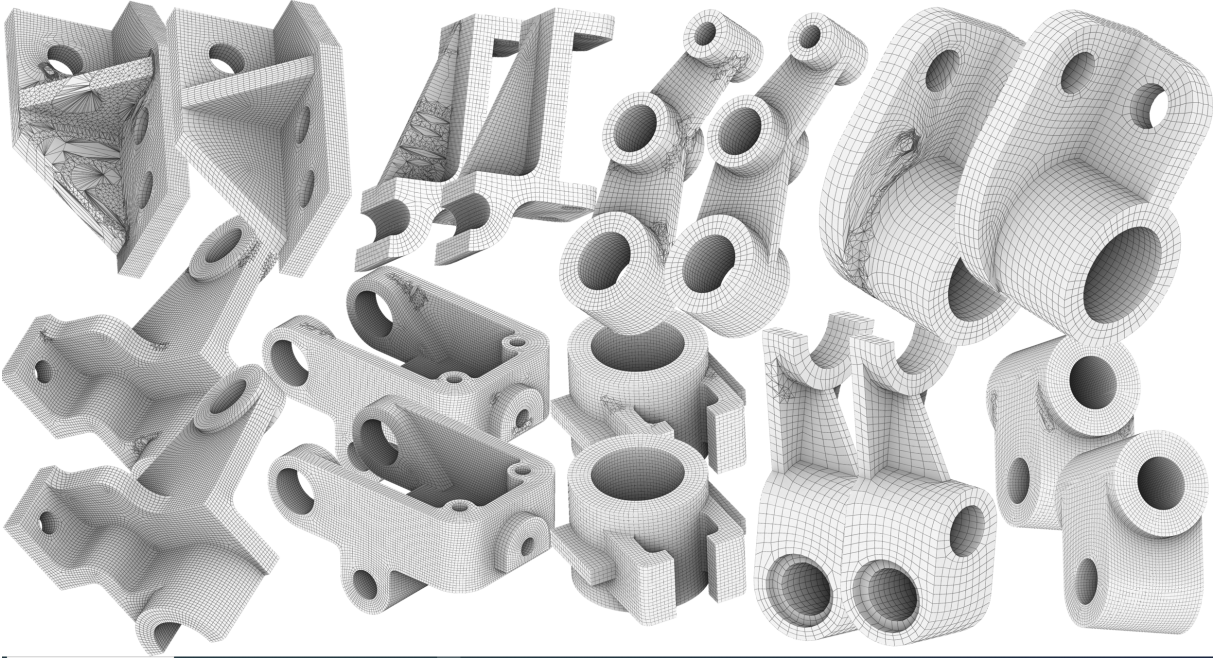


FIGURE 5.3 – Chaque modèle est remaillé avec (premier plan) et sans (arrière-plan) correction des angles vifs par notre méthode.

correspondant d'avantage au modèle que nous avons en entrée. En réoptimisant la géométrie de manière non-orthogonale, nous pouvons obtenir des résultats (Fig 5.3) de qualité similaire à ceux que nous avons avec les champs non-orthogonaux 2D du chapitre 4. Maintenant que nous sommes passé sur une représentation à deux angles, il faut définir une fonction d'association f_p qui va nous permettre de faire correspondre les angles de deux repères voisins à partir de la correspondance $p_{tt'} \in \mathbb{Z}$:

$$f_p(\theta, \phi) = \left(2\pi \lfloor \frac{p}{4} \rfloor, 2\pi \lfloor \frac{p}{4} \rfloor\right) + \begin{cases} (\theta, \phi) & \text{si } p = 0 \pmod{4} \\ (\phi, \theta + \pi) & \text{si } p = 1 \pmod{4} \\ (\theta + \pi, \phi + \pi) & \text{si } p = 2 \pmod{4} \\ (\phi + \pi, \theta + 2\pi) & \text{si } p = 3 \pmod{4} \end{cases}$$

Nous pouvons maintenant optimiser un champ de repère non-orthogonal à partir de la correspondance fixée $p_{tt'}$:

$$\begin{aligned} \arg \min_{\theta, \phi} \quad & \sum_{t \in T'} \sum_{t' \in \mathcal{N}(t)} \left\| \begin{pmatrix} \theta_{t'} \\ \phi_{t'} \end{pmatrix} - f_{p_{tt'}}(\theta_t, \phi_t)^T - \begin{pmatrix} \omega_{tt'} + \gamma_{tt'} \\ \omega_{tt'} + \gamma_{tt'} \end{pmatrix} \right\|^2. \\ \text{sous contrainte :} \quad & \forall t \in T_b, \theta_t = \eta_t. \end{aligned} \quad (5.2)$$

5.4 Résultats et applications

Le temps d'exécution n'était pas une préoccupation majeure dans ce travail car, tant que nous restons dans le cadre de la résolution de quelques systèmes linéaires, la génération du champ de repère n'est pas la partie la plus lente du processus de maillage quad. Notre algorithme est 3 à 5 fois plus lent que la méthode la plus rapide pour produire des champs de repère

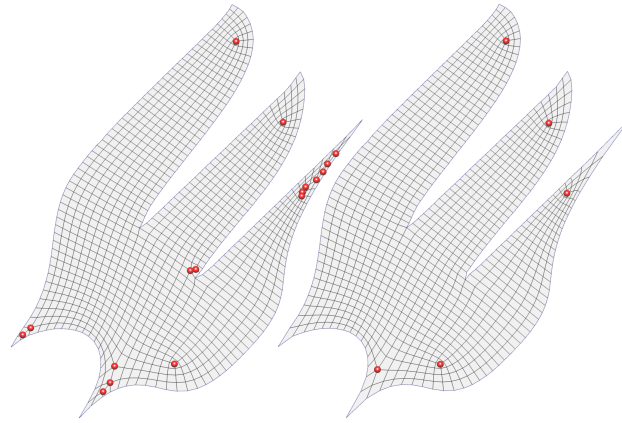


FIGURE 5.4 – *Le maillage quadrilatère généré par [FANG et al. 2018] (à gauche) nécessite beaucoup plus de singularités pour traiter les arêtes caractéristiques que le nôtre (à droite).*

orthogonaux (Sec 3.1.2). Le temps supplémentaire est consommé par le calcul de γ et la relaxation de l'orthogonalité.

Il existe des autres méthodes de génération de maillage quadrilatère qui fonctionnent avec les champs de repère orthogonaux présentés dans la section 3.1.2, cependant ces méthodes génèrent bien plus de singularités que la méthode Quadcover [KALBERER et al. 2007] que nous utilisons avec le champ de repère présenté dans ce chapitre. En effet, notre méthode de maillage quadrilatère présentée dans la section 3.1 transfère les singularités du champ de repère en entrée, sans en ajouter de nouvelles (voir Fig.5.4).

Dans ce chapitre 5, nous avons établi une méthode rapide pour rendre les champs orthogonaux compatibles avec les coins à petit angle des modèles CAO. Dans le chapitre 8, nous utilisons cette même méthode pour contrôler plus généralement la valence de tous les points du bord d'un maillage quadrilatère produit à partir d'un champ de repère orthogonal. En effet, plutôt que de simplement modifier le transport parallèle lorsqu'un point de bord est sur un coin à petit angle, nous le modifions partout où la valence de bord n'est pas adaptée à une simulation de déformation donnée. Malheureusement, cette méthode est difficile à étendre en 3D. Pour atteindre l'objectif de contrôler les valences des arêtes de bord d'un maillage hexaédrique produit à partir d'un champ de repère 3D orthogonal, nous allons employer une autre méthode que nous présentons dans le prochain chapitre 6.

6

Champ de repère avec contraintes de bords adaptées à la CAO

Ce chapitre présente une méthode pour contrôler les valences sur le bord des maillages hexaédriques produits par les champs de repère 3D présentés dans la section 3.2.1. Ce contrôle des valences de bords permet notamment de rendre les champs orthogonaux compatibles aux modèles CAO possédant des coins à petit angle. Dans les chapitres 4 et 5, nous avons rendu les champs de repère compatibles aux coins à petits angles en définissant une nouvelle définition de lissage de champ. Dans ce chapitre, nous apportons une solution qui agit seulement sur les contraintes d’alignement au bord, en gardant la définition de lissage classique présentée dans le chapitre 3.2.1. Cette solution est présentée en 3D, et est de l’ordre de 1000 fois plus rapide que la solution présentée dans le chapitre 4. Nous verrons également dans le chapitre 8 que contrôler les valences sur le bord permet de construire des maillages résistants à des déformations extrêmes dans les simulations qui opèrent de grandes déformations sans étape de remaillage.

6.1 Introduction

Nous présentons une méthode pour imposer le graphe de singularité de bord d’un champ de repère 3D avant de l’optimiser. En contrôlant les indices de singularité aux bords, cette technique peut générer des maillages hexaédriques avec des valences d’arête de bord spécifiées. Dans les situations où les arêtes caractéristiques d’un modèle CAO se croisent à des angles vifs, l’approche traditionnelle consistant à aligner le champ de repère avec les normales de surface conduit à des singularités de bord non maillables. Au lieu de cela, nous alignons le champ de repère avec un champ de direction précalculé spécifiquement conçu pour produire les singularités de bord valides données en entrée. Nous présentons un algorithme pour générer ce champ de direction et fournissons des heuristiques pour obtenir les singularités de bord d’entrée nécessaires. Nous démontrons l’efficacité de cette approche en produisant des maillages valides malgré des petits angles à travers un pipeline de maillage hexaédrique standard, et nous montrons que la modification des singularités de bord d’un champ de repère peut également améliorer les performances des techniques d’optimisation de champ de repère actuelles dans la construction d’un graphe de singularité intérieur valide.

Le principal problème des méthodes de maillage hexaédrique basées sur des champs de repère 3D vient du fait que les matrices de correspondance $g_{tt'}$ obtenues à partir des champs de repère (sec 3.2.1) peuvent ne pas correspondre à un graphe de singularité de maillage hexaédrique. Ce scénario, rare en 2D mais fréquent en 3D, fait très souvent échouer la méthode cubecover que

nous avons décrit dans la section 3.2. En effet, seulement un sous-ensemble de tous les graphes de singularité de champ de repère possibles peut être utilisé pour le maillage hexaédrique [LIU et al. 2018] ; [REBEROL et al. 2019]. Rien que sur le bord, les indices de singularité du champ de repère peuvent être incompatibles avec la création de maillage hexaédrique.

Typiquement, si l'angle diédral $\theta_{tt'}$ d'une arête de bord commune à $t, t' \in T_b$ tombe dans la plage d'angle $[\pi - \pi/4, \pi + \pi/4]$, un champ de repère lisse qui s'aligne avec les normales du bord l'ajuste à un angle de π et traite l'arête de bord comme régulière. Au lieu de cela, si l'angle diédral tombe en dehors de cette plage, l'arête est considérée comme singulière, et son indice est déterminé par le multiple le plus proche de $\pi/2$. Cependant, lorsqu'il s'agit de modèle CAO comportant des arêtes caractéristiques se croisant à des angles fortement aigus ou obtus, les indices de singularité déterminés en arrondissant l'angle diédral géométrique aboutissent souvent à un graphe de singularité invalide.

Prenons l'exemple d'un modèle ayant des angles diédraux inférieurs à $\pi/4$ (Figure 6.1). D'une part (Figure 6.1, gauche), un champ de repère aligné avec les normales du bord l'ajuste à un angle de 0, ce qui donne une arête de valence 0 qui ne peut pas être maillée. D'autre part (Figure 6.1, droite), un champ de repère aligné avec un champ de vecteur soigneusement choisi l'ajuste à un angle de $\pi/2$, ce qui correspond à une arête de valence 1 dans le maillage hexaédrique de sortie. Cela démontre que nous pouvons influencer les singularités des arêtes de bord d'un champ de repère en ajustant ses contraintes d'alignement au bord. De plus, la modification du graphe de singularité au bord affecte également le graphe de singularité à l'intérieur du modèle, ce qui peut être bénéfique.

Dans ce chapitre, nous proposons un algorithme simple pour générer des champs de repère avec des valences prescrites sur les arêtes de bord. L'idée clé est de contraindre les repères du bord à s'aligner avec une direction v qui peut différer de la normale du bord n de sorte que l'arrondi conduise à la valence d'arête désirée. Ensuite, nous utilisons l'optimisation de champ de repère symétrique de la section 3.2.1 pour interpoler de manière lisse le champ et obtenir le graphe de singularité interne. Si ce champ de repère a les singularités désirées, il ne peut pas être utilisé directement pour le maillage car il n'est pas aligné avec le bord. Par conséquent, nous calculons un second champ de repère avec correspondances $g_{tt'}$ fixé pour chaque couple de tétraèdres t, t' adjacent, mais cette fois aligné avec les normales du bord. Cette nouvelle géométrie de champ de repère et les matrices de correspondance $g_{tt'}$ sont ensuite utilisées pour générer des maillages hexaédriques avec la méthode présentée dans le chapitre 3.

Nous démontrons que notre méthode peut être utilisée pour résoudre le problème de maillage de modèle CAO avec des arêtes caractéristiques formant des coins à petit angle (Fig 6.2). De plus, nous proposons des heuristiques pour déterminer des valences d'arête en entrée permettant d'éviter les problèmes de cycles limites décrits dans [REBEROL et al. 2019].

Pour résumer, notre algorithme suit les étapes suivantes :

1. Calcul de contraintes de bords v (Sec 6.2) ;
2. Génération d'un champ de repère lisse avec les contrainte de bords v (Sec 3.2.1).
3. Recalcul d'un champ de repère à combinatoire fixée aligné avec les normales du bord (Sec 6.2.4).
4. Extraction d'une carte sans couture (Sec 3.2.2) et un maillage hexaédrique (Sec 3.2.3).

6.2 Calcul des contraintes de bord à partir de valences de bord

En pratique, choisir la matrice de correspondance la plus proche, comme dans l'Équation (3.13), revient à choisir les valences d'arête en arrondissant les angles dièdres de bord. Pour un angle

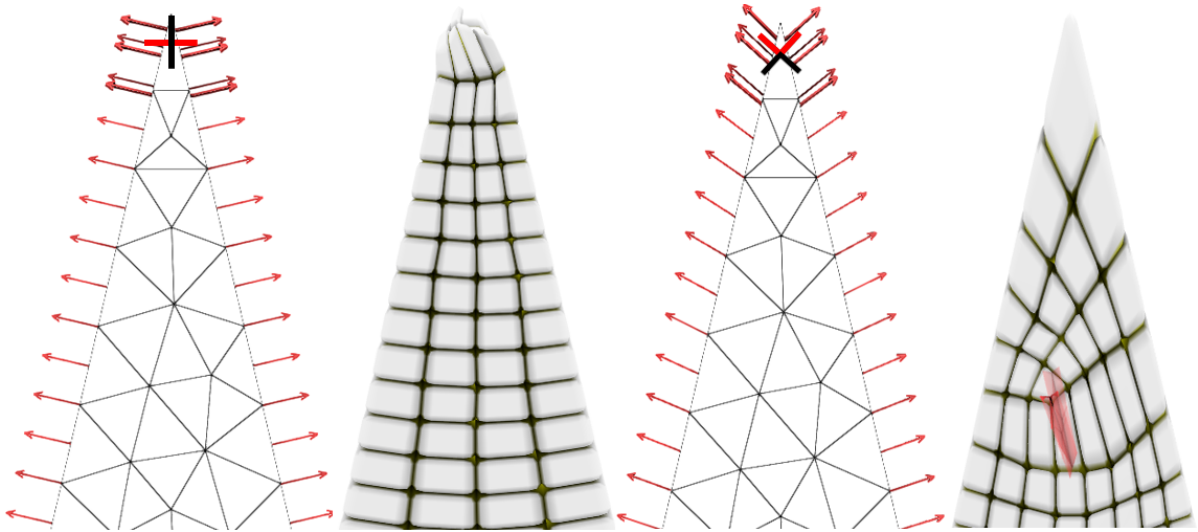


FIGURE 6.1 – Dans le cas du maillage d’une arête à petit angle, un champ de repère orthogonal aligné avec les normales de surface satisfait les deux contraintes avec des vecteurs de direction opposés d’un repère (en haut, à gauche), conduisant à une arête de valence 0 qui ne peut pas être maillée. Nous calculons de nouvelles contraintes de direction de sorte qu’un champ de repère utilise deux de ses directions pour s’aligner (en haut, à droite), ce qui donne une arête de valence 1 qui peut être maillée. La régularité globale de nos contraintes de direction permet la production d’une ligne de singularité interne valide (ligne rouge, en bas à droite) en utilisant la méthode d’optimisation de champ de repère [RAY et al. 2016].

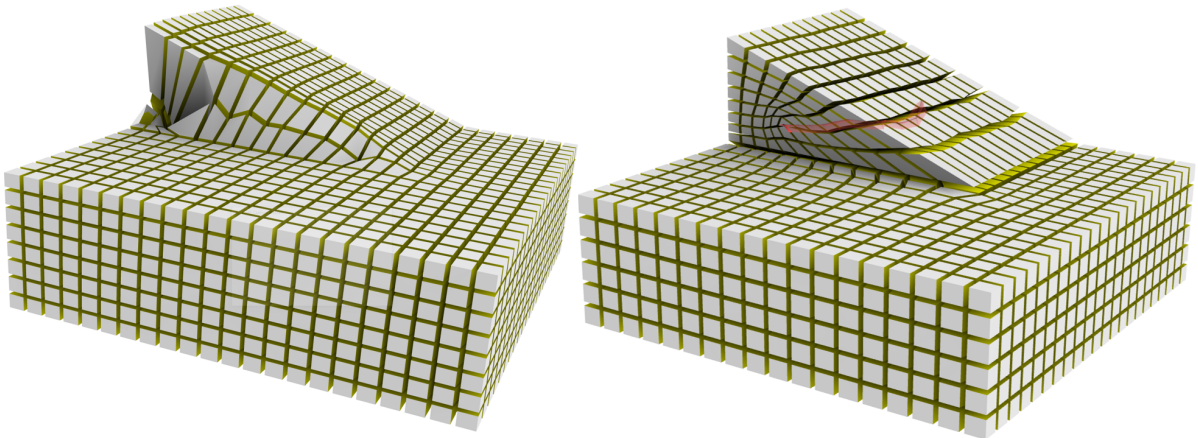


FIGURE 6.2 – Parce que l’angle de la pente est trop faible, un champ de repère orthogonal suivant les directions normales au bord échoue à séparer l’extrémité de la pente du support de la pente, ce qui donne un résultat dégénéré (à gauche). Au lieu de cela, nos contraintes de direction obtenues avec notre méthode séparent la pente et son support avec un angle de 90° , ce qui donne une arête de valence 3 à l’extrémité de la pente. La ligne rouge correspond à une ligne de singularité interne de valence 3 produite par l’optimisation du champ de repère [RAY et al. 2016] avec nos contraintes de direction.

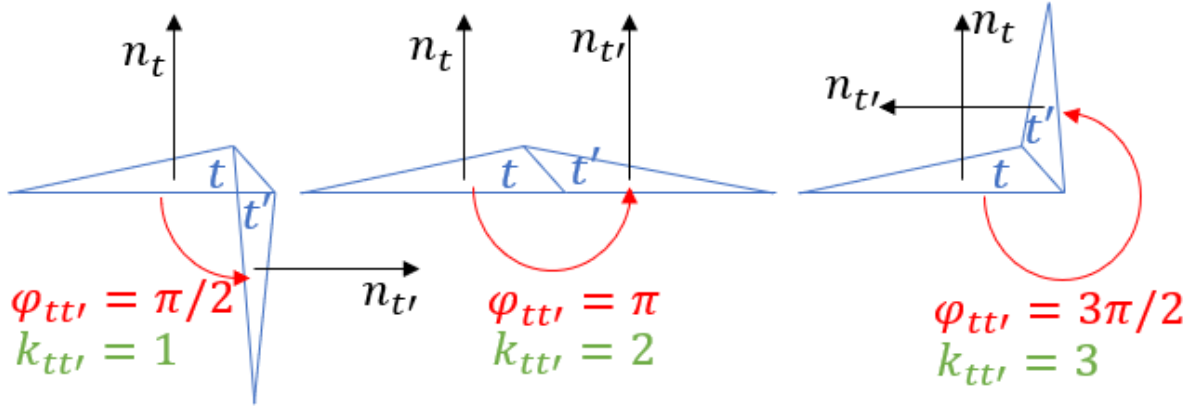


FIGURE 6.3 – L’angle dièdre $\varphi_{tt'}$ et la valence géométrique d’arête $k_{tt'}$ sont calculés en utilisant les deux normales de faces n_t et $n_{t'}$ adjacentes à l’arête de bord adjacente à t et t' .

dièdre $\varphi_{tt'}$, déduit des normales au bord des tétraèdres $t, t' \in T_b$ d’un champ de repère, la valence d’arête associée avec le pipeline décrit dans le chapitre 3 est $\text{round}(2\varphi_{tt'}/\pi)$. Dans cette section, notre objectif est de trouver un nouveau champ de direction de bord v de telle sorte que l’arrondi de $\varphi_{tt'}$ mène aux valences de bord $k_{tt'}$ prescrites par l’utilisateur. Pour ce faire, nous nous appuyons sur une optimisation gloutonne pour minimiser deux objectifs, l’un contrôlant les angles dièdres et l’autre imposant de préserver la courbure du bord.

En raison de l’arrondissement, nous voulons être certains que la déviation d’angle maximale par rapport à l’angle dièdre cible est inférieure à $\pi/4$. Pour cette raison, nous n’utilisons pas un objectif de moindres carrés, mais plutôt une minimisation de la déviation d’angle maximale.

La discussion sur la façon de choisir les valences $k_{tt'}$ des arêtes est reportée à la Section 6.3.

6.2.1 Énergies

Nous désignons par $\varphi(n_t, n_{t'})$ l’angle dièdre entre deux plans dont les normales sont n_t et $n_{t'}$ et qui se coupent le long de l’arête $e_{tt'}$.

$$\varphi(n_t, n_{t'}) = \pi - \text{atan2} \left(\langle n_t \times n_{t'}, \frac{e_{tt'}}{|e_{tt'}|} \rangle, \langle n_t, n_{t'} \rangle \right).$$

La Figure 6.3 illustre toutes ces variables.

À un tétraèdre t , nous voulons que notre nouvelle contrainte de direction v définisse un angle dièdre correspondant aux valences du bord $k_{tt'}$:

$$\delta_t^d(v) := \max_{t' \in \mathcal{N}(t)} |\varphi(v_t, v_{t'}) - k_{tt'}\pi/2|. \quad (6.1)$$

Cet objectif seul est insuffisant car il est entièrement indépendant de la courbure de la surface. Pour une sphère, minimiser l’Équation (6.1) conduirait à un champ de vecteur constants v à partir duquel il serait impossible de générer un graphe de singularité intérieur valide. Ainsi, nous désirons également que le nouveau champ du bord v reste aussi proche que possible de la courbure initiale. Dans cet objectif, nous définissons, à chaque arête commune à deux tétraèdres du bord $t, t' \in T_b$, une paire de normales idéales $\bar{n}_{tt'}$ et $\bar{n}_{t't}$. Les deux normales doivent être proches, au sens des moindres carrés, des normales originales mais doivent également satisfaire la contrainte

d'angle dièdre. Plus formellement, notre paire de normales idéales est la solution du problème d'optimisation :

$$\begin{aligned} \min_{\bar{n}_{tt'}, \bar{n}_{t't} \in \mathbb{R}^3} & \quad |\bar{n}_{tt'} - n_t|^2 + |\bar{n}_{t't} - n_{t'}|^2 \\ \text{subject to :} & \quad \varphi(\bar{n}_{tt'}, \bar{n}_{t't}) = k_{tt'}\pi/2 \\ & \quad |\bar{n}_{tt'}| = |\bar{n}_{t't}| = 1 \end{aligned} \quad (6.2)$$

Soit $R_{tt'}$ la rotation d'angle $\pi - k_{tt'}\pi/2$ et d'axe le vecteur d'arête $e_{tt'}$, nous l'utilisons pour trouver les solutions de l'Équation (6.2) pour obtenir nos normales idéales :

$$\bar{n}_{tt'} = \frac{n_t + R_{tt'}n_{t'}}{|n_t + R_{tt'}n_{t'}|}.$$

L'énergie de lissage mesure la distance angulaire maximale par rapport à nos normales idéales :

$$\delta_t^n(v) := \max_{t' \in \mathcal{N}(t)} \arccos \left(v_t^\top \bar{n}_{tt'} \right). \quad (6.3)$$

La déviation d'angle au tétraèdre $t \in T$ est définie comme le maximum de la distance angulaire par rapport aux angles dièdres cibles (Équation (6.1)) et de la distance angulaire par rapport aux normales idéales (Équation (6.3)) :

$$\mathcal{E}_t(v) := \max(\delta_t^d(v), \delta_t^n(v)).$$

La déviation d'angle maximale parmi tous les tétraèdres t pour un ensemble donné de vecteurs de direction v est notée comme suit :

$$\mathcal{E}(v) := \max_{t \in V_b} \mathcal{E}_t(v).$$

Notre objectif dans la sous-section suivante (Sec 6.2.2) est de trouver un ensemble de vecteurs de direction v qui minimise $\mathcal{E}(v)$. L'arrondissement de l'angle dièdre garantit qu'un champ de repère orthogonal lisse contraint par les directions v a un graphe de singularité de bord correspondant aux valences d'arête de bord d'entrée $k_{tt'}$ chaque fois que $\mathcal{E}(v) < \pi/4$ (voir Sec.6.4.2).

6.2.2 Une méthode d'optimisation gloutonne

Comme nous souhaitons minimiser une fonction objective non convexe, nous nous appuyons sur un algorithme glouton simple.

Nous initialisons les directions v_t pour être égales aux vecteurs normaux n_t , et nous remplaçons itérativement v_t par le vecteur unitaire u_t qui minimise la déviation d'angle $\mathcal{E}_i(v)$ lorsque $v_{t'}$ sont considérées comme constantes pour tous les $j \neq i$. Le vecteur u_t est déterminé par les directions de vecteurs voisins au moment de la modification. Son calcul exact est détaillé dans la Sec. 6.2.3.

Après que le vecteur v_t a été modifié, nous devons mettre à jour les vecteurs des tétraèdres voisins. Ainsi, notre algorithme glouton utilise une file d'attente pour déterminer les indices des faces de bords qui doivent être mis à jour. Tout d'abord, la file d'attente est remplie avec tous les indices des faces. Ensuite, tant que la file d'attente n'est pas vide, nous choisissons et supprimons l'index de devant. Si un vecteur u_t réduit l'énergie de la face courante, nous remplaçons le vecteur de direction v_t par u_t et ajoutons toutes les faces adjacentes à la file d'attente (si elles ne s'y trouvent pas déjà). Cette procédure est résumée dans le pseudo-code de l'Algo. 1.

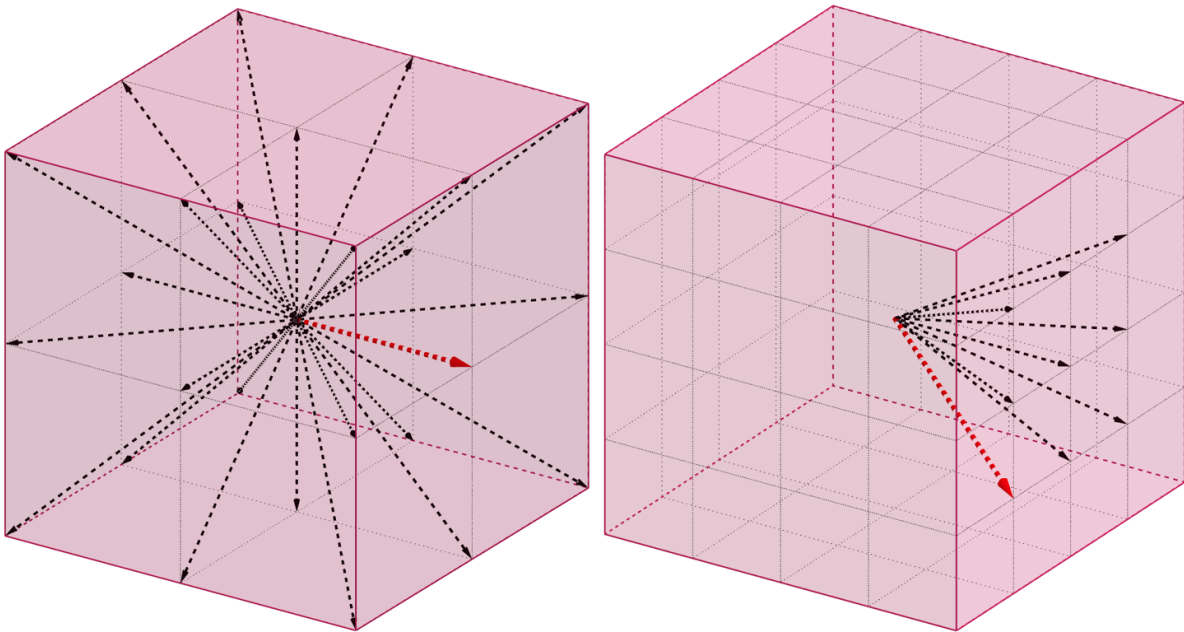


FIGURE 6.4 – Illustration de la recherche d'un vecteur minimum parmi une discrétisation donnée par un cube. Dans la première étape, le vecteur de direction qui minimise \mathcal{E}_t est choisi parmi 26 vecteurs de direction, chacun représenté par des points de coordonnées entières à la surface d'un cube centré sur l'origine avec une taille de 2. Ensuite, la taille du cube est doublée pour identifier le vecteur de direction optimal (affiché en rouge, à droite) parmi les points entiers qui entourent la solution précédente (rouge, à gauche). Cette procédure est répétée 9 fois pour atteindre un vecteur de direction qui est proche d'être optimal.

Algorithm 1: Greedy boundary vector optimization

Input: Vecteurs normaux initiaux
Output: Vecteurs optimisés

```

1 Function IterativeOptimization( $n$ ) :
2    $v \leftarrow nQ \leftarrow$  boundary facet indices ; while  $Q$  is not empty do
3      $i \leftarrow Q.\text{pop\_front}()$  ;  $u \leftarrow \text{MinFacetDirection}(i)$  ; if  $\mathcal{E}(u) < \mathcal{E}(v)$  then
4        $v_t \leftarrow u_t$  ; forall  $t' \in \mathcal{N}(t)$  do
5          $Q.\text{push}(j)$  ;
6       end
7     end
8   end
9   return  $v$  ;

```

6.2.3 Minimisation d'un vecteur unique

Afin d'utiliser l'Algo. 1, nous devons résoudre le sous-problème de minimisation de $\mathcal{E}(v)$ pour la variable v_t en considérant toutes les autres composantes fixes. Notre énergie dépend uniquement de l'angle entre les vecteurs, donc la longueur de v_t n'affecte pas la valeur de \mathcal{E} , seule la direction est importante. Ainsi, nous utilisons un algorithme de recherche itératif en échantillonnant des directions de coordonnées entières à la surface d'un cube centré de longueur d'arête 2^n pour des valeurs croissantes de $n \in \mathbb{N}$.

Nous commençons avec un cube de longueur $n = 2$, ce qui donne 26 directions échantillonnées. À la première étape, nous testons exhaustivement toutes les directions et gardons celle avec l'énergie la plus basse. À l'itération suivante, nous doublons la taille du cube et testons uniquement les 8 (ou 6 à un coin) directions adjacentes avec des coordonnées entières, comme illustré dans Fig. 6.4. Nous gardons à nouveau le meilleur candidat. Nous répétons cette procédure 9 fois.

Un cube de longueur 2^p a un total de $24 \times 4^p + 2$ directions échantillonnées. Au lieu de tester toutes les directions, notre méthode de dichotomie ne teste que $9k + 26$ vecteurs de direction après p itérations pour trouver une approximation d'un minimiseur local v_t . En pratique, nous arrêtons à $p = 9$. Cette procédure est résumée dans le pseudo-code de l'Algo. 2.

6.2.4 Champ de repère lisse avec appariement fixe

Après avoir obtenu un champ de repère 3D $(E_t)_{t \in T}$ aligné avec les directions de bord $(v_t)_{t \in T_b}$ par la méthode de l'état de l'art décrite dans le chapitre 3.2.1, ainsi que les matrices de correspondance $g_{tt'}$ correspondant à ce champ de repère, nous voudrions réaligner les repères de bord avec leur normale de surface correspondante. Pour ce faire, nous devons identifier, pour chaque repère de bord E_t , le vecteur de colonne qui s'aligne avec la direction v_t et transférer cette contrainte au vecteur normal n_t . Ainsi, un repère aligné sur le bord \bar{E} doit satisfaire la contrainte :

$$n_t^\top \bar{E}_t = v_t^\top E_t, \quad \forall t \in T_b.$$

Avec ces contraintes de bord, nous aimerions trouver le champ de repère le plus lisse dont les matrices de correspondance sont les $g_{tt'}$ obtenues avec l'Équation (3.13) :

$$\begin{aligned} \min_{\bar{E}} \quad & \sum_{t \in T} \sum_{j \in \mathcal{N}(t)} \|\bar{E}_t - g_{tt'} \bar{E}_t\|_F^2, \\ \text{sous contrainte :} \quad & n_t^\top \bar{E}_t = v_t^\top E_t, \forall t \in T_b. \end{aligned} \tag{6.4}$$

Algorithm 2: Iterative searching algorithm

Input: initial normal vectors of boundary facets, index i

Output: a normalized vector

```

1 Function MinFacetDirection( $v, i$ ) :
2    $N \leftarrow 2$ ;
3    $st \leftarrow (-1, -1, -1)$ ;
4    $end \leftarrow (1, 1, 1)$ ;
5   for  $iter \leftarrow 0$  to 10 do
6      $pt \leftarrow \text{MinPointOfCubeSurface}(v, i, st, end, N)$ ;
7      $N \leftarrow 2 \cdot N$ ;
8      $st \leftarrow 2 \cdot pt - (1, 1, 1)$ ;
9      $end \leftarrow 2 \cdot pt + (1, 1, 1)$ ;
10  end
11  return normalizedVec3( $pt$ );
12 Function MinPointOfCubeSurface( $v, i, st, end, N$ ) :
13   $u \leftarrow v$ ;
14   $pt \leftarrow (1, 0, 0)$ ;
15  for  $x \leftarrow st_x$  to  $end_x$  do
16    for  $y \leftarrow st_y$  to  $end_y$  do
17      for  $z \leftarrow st_z$  to  $end_z$  do
18        if  $\max(|x|, |y|, |z|) == N/2$  then
19           $u_t \leftarrow \text{normalizedVec3}(x, y, z)$ ;
20          if  $\mathcal{E}_t(u) < \mathcal{E}_t(v)$  then
21             $v_t \leftarrow u_t$ ;
22             $pt \leftarrow (x, y, z)$ ;
23          end
24        end
25      end
26    end
27  end
28  return  $pt$ ;

```

Le problème d'optimisation dans l'Équation (6.4) n'impose pas que \bar{E}_t soit une matrice de rotation. Ainsi, nous projetons à posteriori la matrice dans l'espace des matrices de rotation en utilisant une décomposition en valeurs singulières.

6.3 Différentes stratégies pour déterminer les valences des arêtes de bord

Dans cette section, nous présentons trois heuristiques pour choisir les valences des arêtes de bord. La Sec 6.3.1 se concentre sur la reproduction des valences d'arêtes de bord suggérées par les normales de surface (voir Fig 6.3), sauf sur les arêtes à faible angle pour éviter le cas d'échec $k_{tt'} = 0$. Cependant, les arêtes à faible angle ne sont pas les seuls cas où le bord crée naturellement une configuration non maillable. Dans la Sec 6.3.2, nous présentons une heuristique qui utilise la géométrie de surface pour résoudre les problèmes de non-orthogonalité qui ne sont pas liés aux

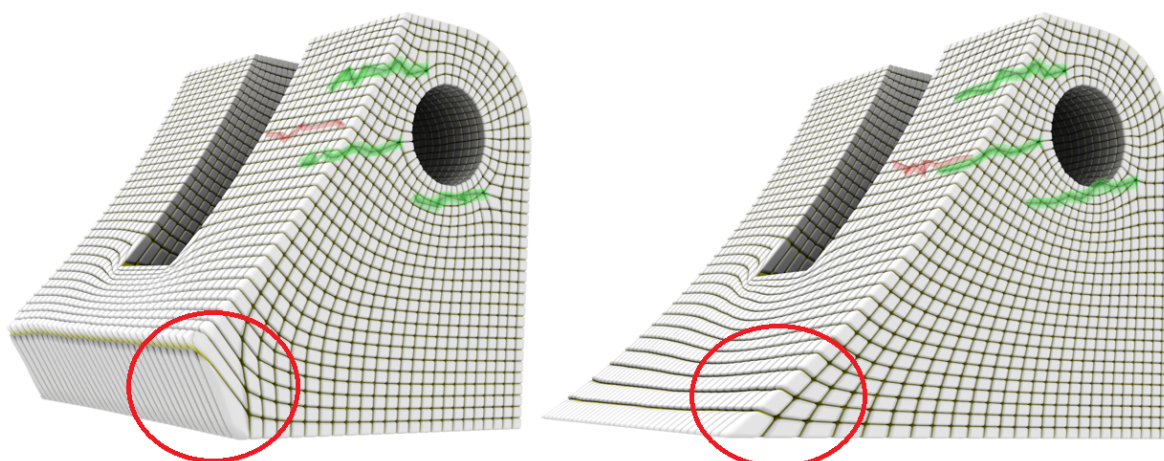


FIGURE 6.5 – Un champ de repère orthogonal suivant les normales de surface du bord près d’une arête à faible angle dièdre considère l’arête comme une arête de sortie de valence 0, ce qui conduit à un résultat dégénéré (à gauche). Nous le résolvons en calculant de nouvelles contraintes de direction qui agissent comme si l’arête à faible angle dièdre était en fait une arête à angle dièdre $\pi/2$, qu’un champ de repère orthogonal gère sans problème (à droite).

arêtes à faible angle. La Sec 6.3.3 se concentre sur les modèles CAO dont la géométrie ne peut pas être utilisée pour déterminer un graphe de singularité du bord valide. Ceux-ci sont fréquemment associés à des cas de non-orthogonalité extrême induit par des arêtes caractéristiques, et nous montrons comment résoudre de tels cas en utilisant des valences d’arêtes données en entrée. Enfin, dans la Sec 6.3.4, nous montrons des exemples de modèles CAO dans lesquels la modification du graphe de singularité du bord est facultative mais aide la méthode d’optimisation de champ de repère présentée dans la section 3.2.1 à trouver un graphe de singularité intérieur valide.

6.3.1 Arête à faible angle dièdre

Si le maillage tétraédrique en entrée a des angles dièdres $\varphi_{tt'}$ inférieurs à $\pi/4$, un champ de repère orthogonal aligné sur les directions normales de surface a un graphe de singularité du bord correspondant à une arête de valence $k_{tt'} = 0$ dans le maillage hexaédrique en sortie, qui n’est pas maillable (voir Fig. 6.5, à gauche).

Nous proposons une heuristique simple consistant à assigner une valence d’arête de $k_{tt'} = 1$ chaque fois que l’angle dièdre $\varphi_{tt'}$ est inférieur à $\pi/4$. Pour une arête de bord appartenant à $t, t' \in T_b$, nous calculons les valences d’arête comme suit :

$$k_{tt'} = \max \left(\text{round} \left(\frac{2\varphi_{tt'}}{\pi} \right), 1 \right).$$

En conséquence, les contraintes de direction v calculées à partir de ces valences (Sec 6.2) éliminent les problèmes d’arête à faible angle dièdre sans en créer de nouveaux, comme illustré dans la Figure 6.5 (à droite) et la Figure 6.6. Cela signifie que l’utilisation de ces contraintes de direction v est toujours préférable à l’utilisation des contraintes de direction n de normales de surface.

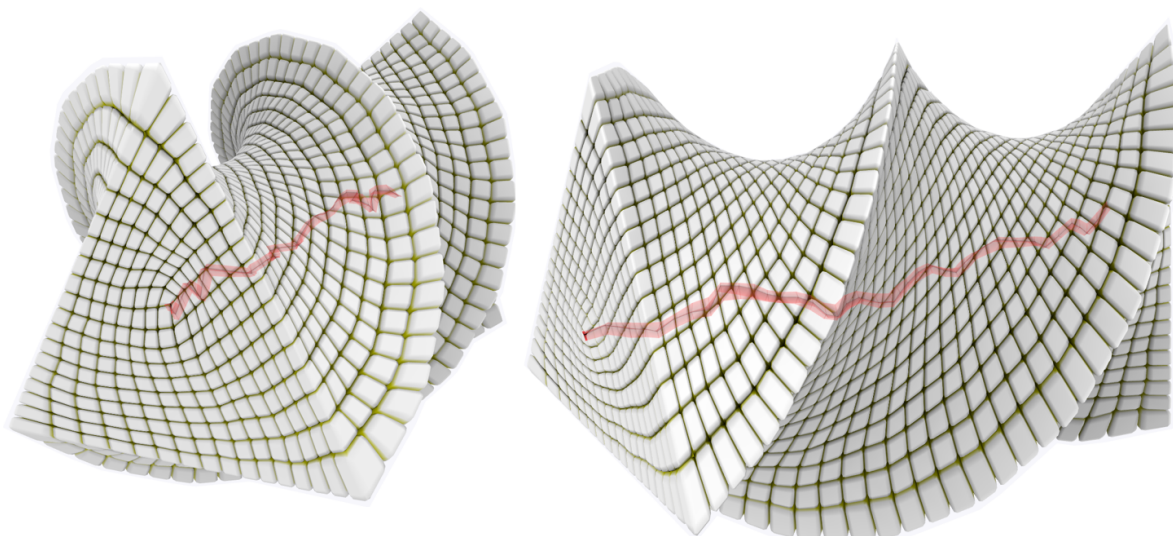


FIGURE 6.6 – *Tourner un modèle pendant son extrusion permet de générer beaucoup de non-orthogonalité dans de nombreuses directions différentes. Notre méthode construit un maillage hexaédrique comme si le modèle avait été extrudé normalement.*

6.3.2 Angles diédraux ambigus

Beaucoup des problèmes que nous rencontrons avec le jeu de données MAMBO [LEDOUX 2019] sont causés par des problèmes de non-orthogonalité sur des arêtes avec des angles dièdres $\varphi \approx p\pi/2 + \pi/4, p \in \mathbb{N}$. Dans ce cas, choisir la valeur $k = p$ peut conduire à des valences d’arête du bord qui ne sont pas compatibles avec le maillage hexaédrique. Ce problème de maillage peut être résolu en augmentant systématiquement la valence d’arête de un chaque fois que l’angle dièdre est trop proche de $\pi/4$ modulo $\pi/2$:

$$k_{tt'} = \max \left(\text{round} \left(\frac{2\varphi_{tt'}}{\pi} + 0.05 \right), 1 \right).$$

La Figure 6.7 montre 5 maillages hexaédriques générés avec succès grâce à cette heuristique simple.

6.3.3 Modèles CAO avec valences d’arêtes prescrites par l’utilisateur

Pour le maillage hexaédrique de modèles CAO, l’utilisateur peut également ajouter à la description des arêtes caractéristiques la valence d’arête désirée. En procédant ainsi, notre méthode peut construire des contraintes de direction du bord en fonction de ces valences d’entrée. Si nous atteignons un angle de déviation maximal $\mathcal{E}(v)$ inférieur à $\pi/4$ (Sec. 6.4.2), nous nous assurons qu’un champ de repère suivant ces contraintes de direction aura un graphe de singularité du bord valide (correspondant aux valences d’arête valides données sur les arêtes caractéristiques). La Figure 6.8 montre des exemples de modèles CAO pour lesquels nous avons besoin des valences d’arête du bord en entrée car nos heuristiques ne parviennent pas à produire des maillages valides.

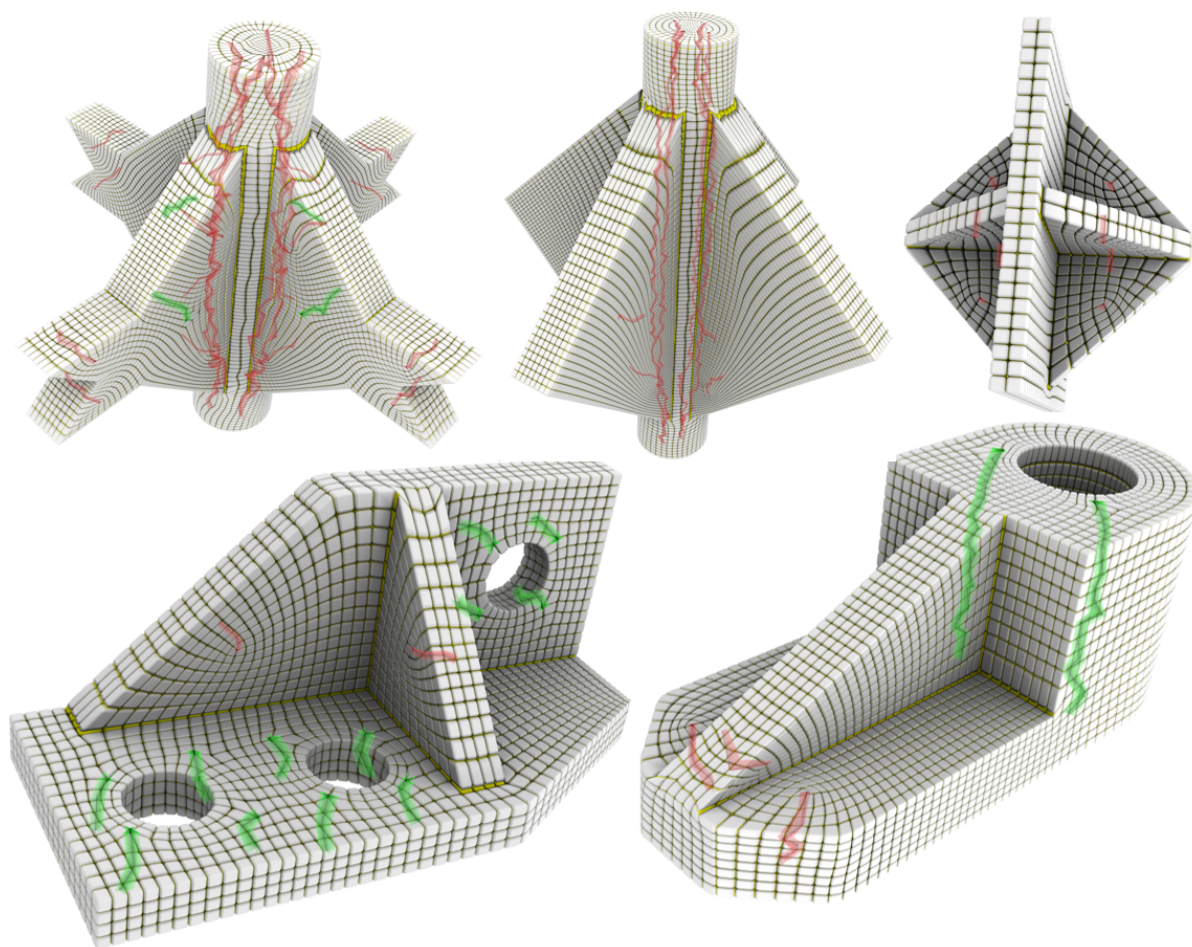


FIGURE 6.7 – Lorsqu'un angle dièdre est proche de $p\pi/2 + \pi/4$ avec $p \in \mathbb{N}$, nous remarquons que choisir soit $k = p$ soit $k = p + 1$ conduit à un graphe de singularité de bord valide, tandis que l'autre ne le fait pas. Pour obtenir les résultats ci-dessus, notre heuristique consiste à choisir la valence d'arête $k = p + 1$ si l'angle dièdre φ est supérieur à $p\pi/2 + 0.9\pi/4$.

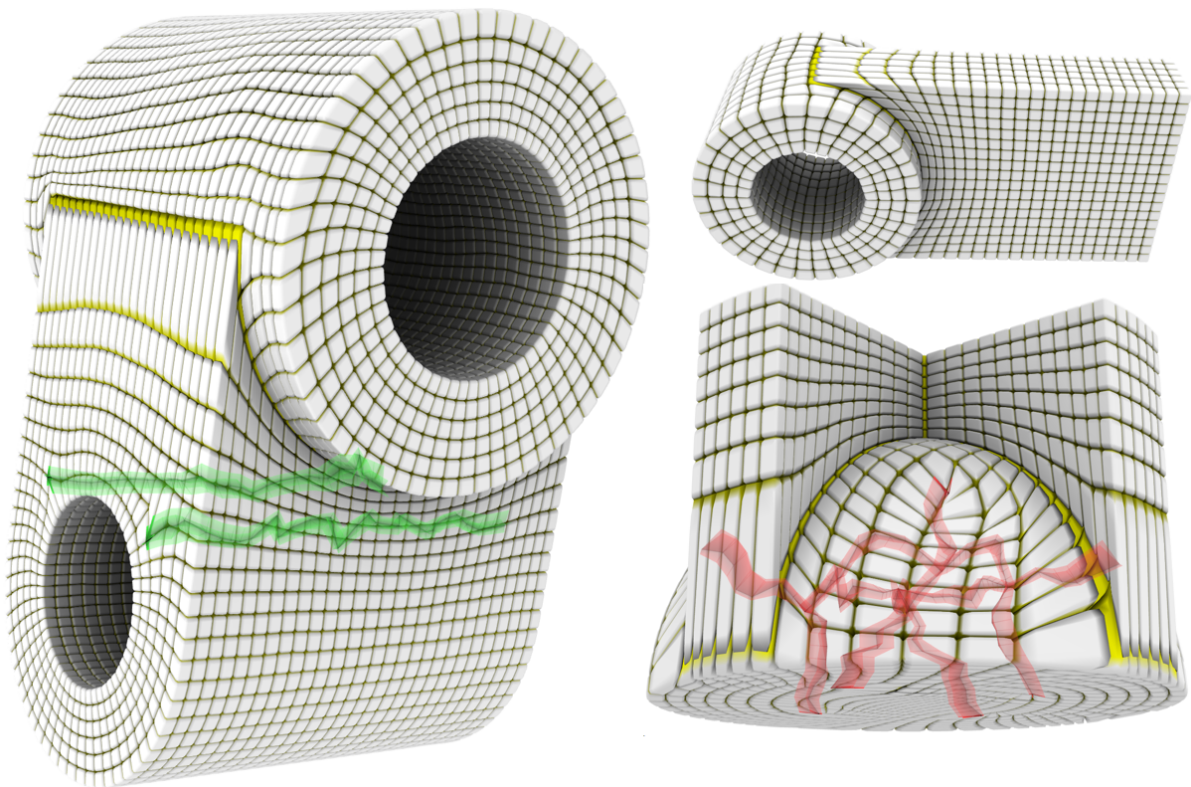


FIGURE 6.8 – Dans le cas de non-orthogonalité extrême dans des modèles CAO, il est difficile de déterminer géométriquement les valences des arêtes caractéristiques. Pour obtenir ces résultats, nous avons prescrit les valences des arêtes caractéristiques dans les descriptions CAO des modèles en entrée.

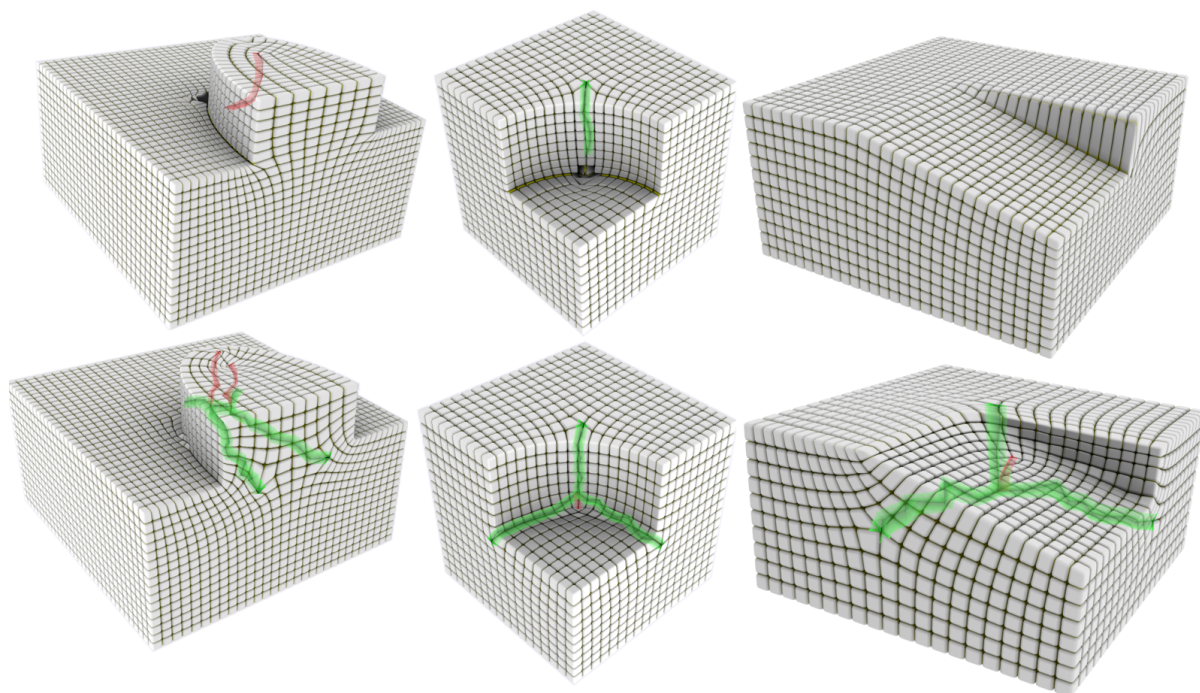


FIGURE 6.9 – Remplacer toutes les arêtes de valence 3 (en haut) par des arêtes de valence 2 (en bas) peut transformer des graphes de singularité non maillables en graphes valides.

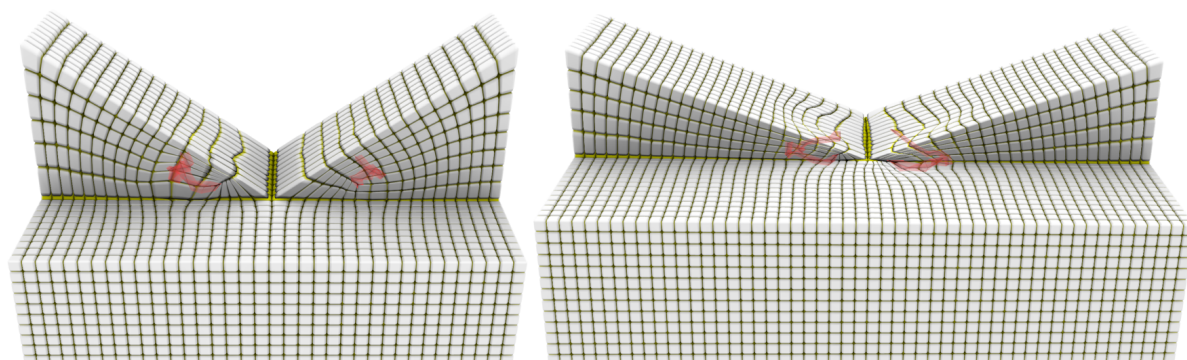


FIGURE 6.10 – Pour calculer un maillage hexaédrique valide, une valence d'arête prescrite de 4 sur les arêtes d'intersection des pentes est requise. À partir des angles intérieurs d'arête de $4,37$ (à gauche) et $3,87$ (à droite), l'optimisation de champ de repère orthogonal [RAY et al. 2016] produit un graphe de singularité intérieur valide comme si les angles intérieurs des arêtes d'intersection des pentes étaient proches de 2π . Notre algorithme réduit les angles de déviation maximale δ_{\max} sur les contraintes de direction du bord de $1,91$ (à gauche) et $2,41$ (à droite) à $0,38$ et $0,48$.

6.3.4 Modification des graphes de singularité internes

Reberol *et al.* [REBEROL et al. 2019] illustrent plusieurs configurations où le calcul d'un champ de repère lisse conduit à des graphes de singularité non maillables. Ils proposent une solution à ce problème en modifiant la géométrie du maillage tétraédrique en entrée. Au lieu de changer la géométrie d'entrée, nous remplaçons simplement les arêtes de valence 3 par des arêtes de valence 2. Pour les modèles de la Figure 6.9, nous avons utilisé :

$$k_{tt'} = \min \left(\text{round} \left(\frac{2\varphi_{tt'}}{\pi} \right), 2 \right),$$

sur toutes les arêtes du bord.

La Figure 6.9 montre trois exemples dans lesquels nous avons évité le problème des singularités non valides en modifiant simplement les valeurs $k_{tt'}$ sur certaines arêtes du bord. Cependant, ces modifications apportées aux singularités du bord ont un impact négatif sur la distance de Hausdorff entre le maillage tétraédrique d'entrée et le maillage hexaédrique de sortie. De plus, le maillage hexaédrique perd partiellement l'alignement avec les arêtes caractéristiques.

6.4 Résultats

Nous avons testé notre algorithme sur les 74 modèles "de base" du jeu de données MAMBO [LEDOUX 2019]. La Figure 6.11 montre les temps de calcul en fonction du nombre de tétraèdres pour chaque modèle de notre jeu de données. Parmi les trois étapes principales de notre algorithme (calcul des nouvelles contraintes du bord, génération d'un champ de repère symétrique et réalignement avec les normales du bord), le calcul d'un champ de repère symétrique avec Ray *et al.* [RAY et al. 2016] est le plus coûteux.

Le calcul des contraintes du bord à partir des valences des arêtes repose sur l'optimisation gloutonne décrite dans la Section 6.2. Bien que nous n'ayons aucune preuve de convergence ou de borne à la solution optimale, ce schéma fonctionne bien en pratique. La Figure 6.12 compare, pour tous nos modèles, l'angle maximal de déviation \mathcal{E} évalué avec les normales du bord initiales et avec le champ de vecteur v , sortie de l'Algorithme 1. Nous parvenons toujours à abaisser la déviation d'angle bien en dessous de la barre de $\pi/4$, évitant ainsi l'erreur d'arrondi.

6.4.1 Comparaisons avec le chapitre 4

Dans le chapitre 4, nous proposons une représentation de champ de repère non-orthogonaux qui peut résoudre certains problèmes de maillage induits par des coins fortement non-orthogonaux. Nous nous appuyons sur un processus d'optimisation non convexe lisse, initialisé avec un champ de repère orthogonal calculé avec la section 3.2.1.

Si la méthode du chapitre 4 offre une solution alternative à la construction de maillages hexaédriques dans des cas non-orthogonaux, elle n'est pas aussi flexible que cette méthode et ne peut résoudre qu'une gamme plus limitée de problèmes. Par exemple, lorsque l'angle diédral $\varphi_{tt'}$ d'une arête du bord est dans la plage $[p\pi/2, (p+1)\pi/2]$, un maillage hexaédrique calculé à partir d'un champ de repère non-orthogonal aura une valence d'arête du bord de p ou $p+1$, selon la façon dont le champ est contraint par les directions normales de la surface.

Notre méthode de ce chapitre 6 peut générer des valences de bord avec des valeurs arbitraires. La figure 6.10 montre des modèles avec des pentes doubles ayant des arêtes d'intersection avec un angle intérieur dans la plage $[\pi, 3\pi/2]$. Avec des valences d'arête prédéfinies de 4 sur les arêtes d'intersection, notre méthode construit des contraintes de direction qui permettent le calcul

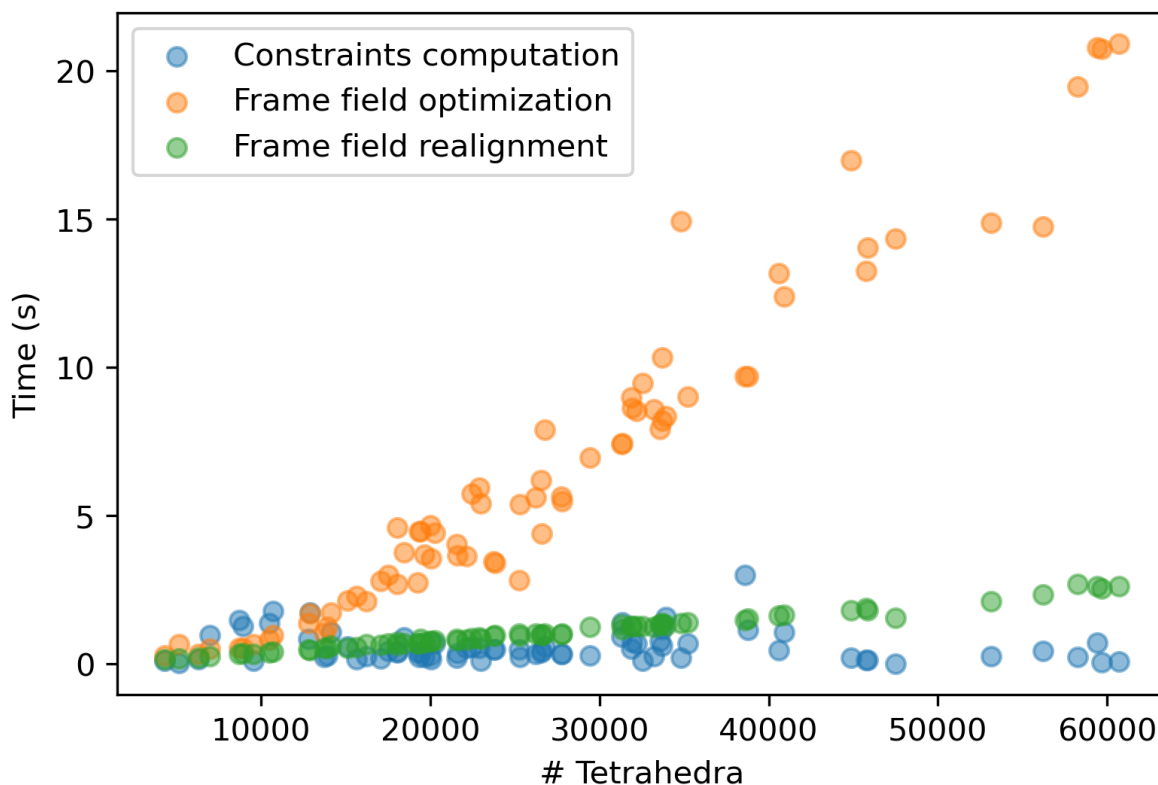


FIGURE 6.11 – Temps de calcul des trois étapes de notre algorithme de génération de champ de repère : calcul des contraintes de bord, optimisation du champ de repère orthogonal et réalignement du champ de repère. Notre algorithme ajoute une surcharge de temps très faible (courbe verte et bleue) par rapport aux méthodes standard de génération de champ de repère (seulement la courbe orange).

d'un maillage hexaédrique valide. Un champ de repère non-orthogonal aligné avec les directions normales de la surface ne peut pas produire ce graphe de singularité de bord valide car les arêtes d'intersection ne peuvent qu'au mieux être de valence 3.

En termes de performances, dans le chapitre 4, nous rapportons un temps de calcul environ 1000 fois lent que pour la génération d'un champ de repère orthogonal. Comme le montre la Figure 6.11, la méthode présentée dans ce chapitre est beaucoup plus rapide car le temps de calcul est largement dominé par la génération du champ de repère orthogonal.

6.4.2 Cas d'échec

La valence d'arête de bord caractéristique $k_{tt'}$ d'un maillage hexaédrique produit en utilisant la méthode décrite dans le chapitre 3 dépend de l'angle dièdre $\varphi(v_t, v_{t'})$ des deux contraintes de direction des tétraèdres t et t' , comme décrit dans la section 6.2.1. L'objectif de l'équation (6.1) est d'atteindre une valeur de $\varphi(v_t, v_{t'})$ qui est au pire à une distance de $\pi/4$ de $k_{tt'}\pi/2$. Cependant, comme $\varphi(v_t, v_{t'})$ est un angle compris entre 0 et 2π , notre valence d'arête cible ne peut pas dépasser 4. Pour surmonter cette limitation, pour les valences $k_{tt'} \geq 5$, l'équation (6.1) doit être

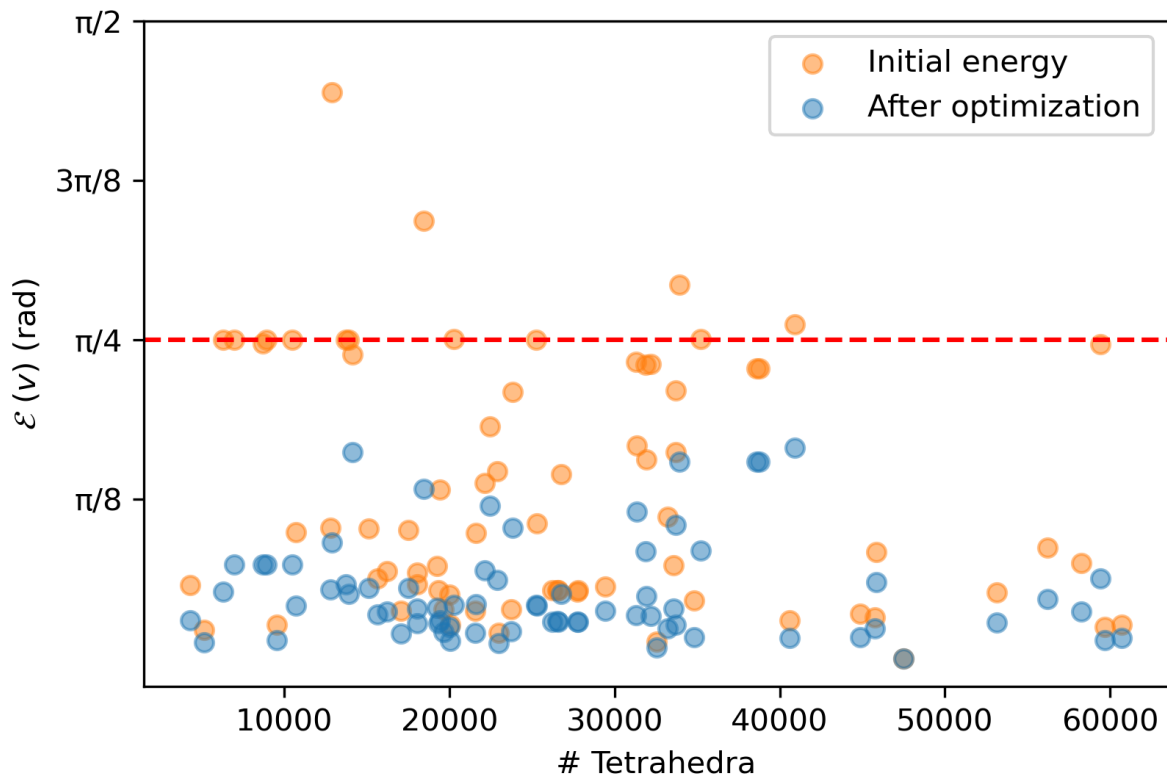


FIGURE 6.12 – Comparaison entre l'énergie initiale et l'énergie après optimisation. Sur les modèles "B" de l'ensemble de données mambo [LEDOUX 2019] et avec des valences d'arêtes obtenues avec l'heuristique de la section 6.3.1, nous obtenons des directions v telles que l'angle de déviation maximum $\mathcal{E}(v)$ est inférieur à $\pi/4$ après optimisation. Cela signifie que le graphe de singularité de bord de notre champ de repère correspond aux valences d'arêtes d'entrée.

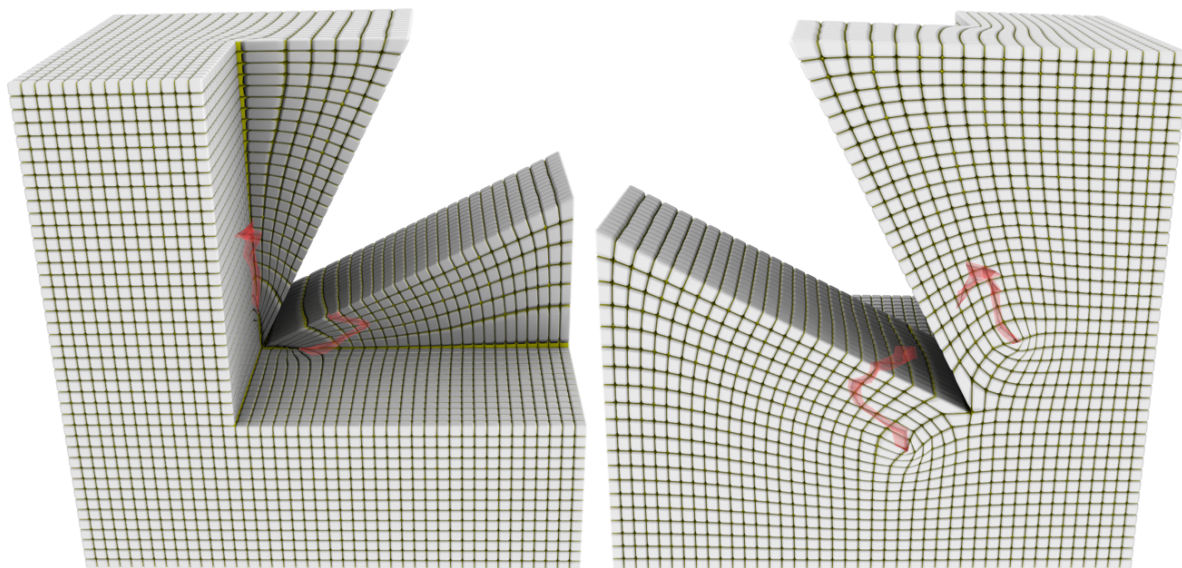


FIGURE 6.13 – Notre méthode produit des contraintes de bord qui entraînent une valence d’arête de bord de 5, simulant une surface de volume qui s’auto-intersecte (angle intérieur de $5\pi/2 > 2\pi$) autour de l’arête des deux pentes. L’angle de déviation initial de $\mathcal{E} = 2.2$ est réduit à $\mathcal{E} = 0.46$ après optimisation.

modifiée pour prendre en compte la plus petite valeur modulo 2π :

$$\delta_t^d(v) = \max_{t' \in \mathcal{N}(t)} \min_{p \in \mathbb{Z}} |\varphi(v_t, v_{t'}) - k_{tt'}\pi/2 + 2p\pi|.$$

Cette modification est illustrée dans la figure 6.13 où une valence d’arête de bord de 5 est désirée et atteinte. Dans ce cas, l’angle de déviation maximum initial $\mathcal{E} = 2.2$, qui correspond à un angle dièdre initial de $5\pi/2 - 2.2$, est réduit à $\mathcal{E} = 0.46 < \pi/4$. Cependant, plus l’angle de déviation maximum initial est élevé, plus il devient difficile de le réduire à une valeur inférieure à $\pi/4$. Pour cette raison, les valences strictement supérieures à 6 peuvent être difficiles à réaliser.

Ce chapitre a présenté une méthode pour contrôler les valences (ou graphe de singularité de bord) des champs de repère 3D orthogonaux. Nous verrons dans le chapitre 8 que le contrôle des valences sur le bord est souvent nécessaire dans le cas de simulation de déformation hyperélastique, pour construire un maillage adapté à des déformations extrêmes sur le bord. De plus, nous avons vu dans ce chapitre qu’à partir de valences écrites sur la description CAO de modèles, nous réussissons, malgré des configurations difficiles, à obtenir automatiquement un graphe de singularité de champ de repère valide dont on peut calculer un maillage hexaédrique. Malheureusement, pour les modèles 3D industriels utilisés dans Numea, les champs de repères que nous produisons dans ce chapitre ont des graphes de singularités internes présentant toujours trop de problèmes pour produire des maillages hexaédriques valides. Ainsi, pour déterminer dans le chapitre 8 la pertinence de la méthode Cubecover (Chap 3) et des contributions de cette thèse pour les simulations de déformation à maillage unique (Chap 1), nous allons nous restreindre à des simulations industrielles en 2D (maillage quadrilatère).

Troisième partie

Contribution : Quadcover pour des simulations 2D de grandes déformations

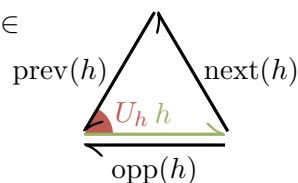
Algorithme de quantification 2D rapide, robuste et open source

Dans ce chapitre, nous présentons un algorithme de quantification que nous avons créé pour compléter la pièce manquante de notre pipeline de génération de maillage quadrilatère. Cet algorithme de quantification se veut plus simple d'implémentation que les précédents nécessitant une décomposition en T-mesh, tout en offrant des performances similaires en terme de fiabilité et de rapidité d'exécution. Une implémentation de ce qui est présenté dans ce chapitre va être rendu open-source. Dans ce chapitre qui est restreint à la 2D, nous utilisons une structure de données en demi-arêtes [MULLER et al. 1978] pour accéder à la connectivité du maillage. Contrairement au chapitre 3, nous représentons la carte U par les coordonnées U_h de l'origine de chaque demi-arête h dans la carte, où H est le nombre total de demi-arêtes dans la triangulation en entrée. Pour caractériser les discontinuités admissibles de U à travers les arêtes, nous introduisons des variables supplémentaires : pour chaque demi-arête h , la fonction de transition entre sa face et sa face opposée est définie par une translation par un nombre complexe de Gauss t_h (de partie réelle et imaginaire entière) et une rotation $r_h = \iota^k$, où $k \in \mathbb{Z}$ (voir Fig. 7.1). Dans cette formulation, notre problème d'optimisation a les variables suivantes : les coordonnées de la carte $(U_h)_{h=1}^H$, les rotations $(r_h)_{h=1}^H$ et les translations $(t_h)_{h=1}^H$. Par rapport aux notations de la section 3.1, $r_h = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}^{p_{tt'}}$, et $t_h = n_{tt'} + im_{tt'}$ si on suppose que la demi-arête h est la demi-arête de t opposée à t' .

7.1 Introduction

Notre objectif est de produire une carte préservant la grille avec une distorsion minimale, ce qui se formule sous forme de problème d'optimisation, consistant à minimiser une fonction objectif $f(U)$ tout en respectant des contraintes de préservation de la grille. Il existe plusieurs façons de définir cette fonction, mais la plupart incluent une estimation de la distorsion de la carte sur tous les triangles. Un exemple de fonction objectif est donnée dans la section 3.1.3. Pour les problèmes plus complexes, des fonctions plus élaborées [GARANZHA et al. 2021] sont utilisées pour produire des cartes valides avec un Jacobien positif.

Formalisation Reformulons les conditions d'une carte préservant la grille 2D présentée dans le chapitre 3.1.3 qui lie toute demi-arête $h \in [1 \dots H]$ à son opposée $\text{opp}(h)$.



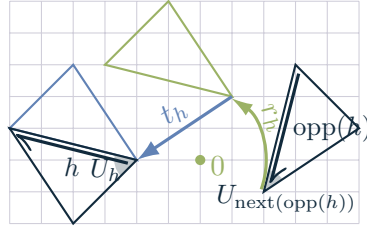


FIGURE 7.1 – Fonction de transition préservant la grille à travers une demi-arête h donnée par une rotation r_h et une translation t_h .

- La première condition est que la fonction de transition est une combinaison d’une rotation r_h et d’une translation t_h :

$$\begin{cases} U_h & = r_h U_{\text{next}(\text{opp}(h))} + t_h \\ U_{\text{next}(h)} & = r_h U_{\text{opp}(h)} + t_h \\ r_h & \in \{1, i, i^2, i^3\} \end{cases} \quad (7.1)$$

- La seconde condition est que certaines variables doivent être entières. Évidemment, les translations t_h doivent être des nombres complexes de Gauss, mais en plus de cela, les sommets singuliers doivent également être des nombres complexes de Gauss. En effet, lorsque les angles des coins du triangle (dans la carte) partageant un sommet v ne somment pas à 360° , le sommet v est appelé singulier et doit être un nombre complexe de Gauss pour produire un sommet irrégulier dans le maillage quad final. Ces contraintes peuvent s’exprimer comme suit :

$$\begin{cases} t_h \in \mathbb{Z}[i] \quad \forall h \in \{1 \dots H\} \\ U_h \in \mathbb{Z}[i] \text{ si le sommet correspondant à } h \text{ est une singularité} \end{cases} \quad (7.2)$$

où $\mathbb{Z}[i] = a + bi \mid a, b \in \mathbb{Z}$ sont les nombres complexes de Gauss.

Enfin, le problème de génération de carte préservant la grille peut être formulé comme un problème d’optimisation :

$$\arg \min_{\{U_h, r_h, t_h\}_{h=1}^H} f(U) \quad \text{s.t. (7.1) and (7.2)} \quad (7.3)$$

Énoncé du problème Bien que la formulation générale du problème soit simple, le mélange de variables entières et réelles, et surtout le grand nombre de variables entières, rendent le problème extrêmement difficile à résoudre. En pratique, la plupart des tentatives pour résoudre le problème réduisent le nombre de degrés de liberté en décomposant le problème en deux parties. La première partie définit l’orientation souhaitée pour les quadrilatères (le champ de repère), et la seconde partie définit deux champs scalaires (les coordonnées de texture) de telle sorte que leurs iso-valeurs soient alignées avec le champ de repère. Comme observé dans [BOMMES et al. 2009], les deux problèmes impliquent des valeurs entières et peuvent être résolus par le même type de solveur. Les valeurs entières dans le premier problème correspondent à la topologie du champ de repère, comme décrit dans le chapitre 3.1.2, qui est liée à la valence des sommets singuliers dans le maillage quadrilatère final. Dans le deuxième problème, les valeurs entières

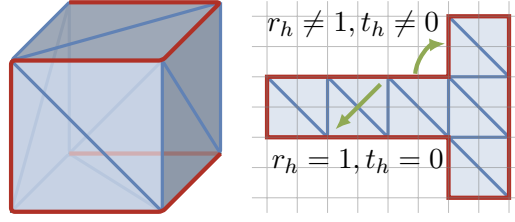


FIGURE 7.2 – *Graphe de découpe pour un maillage triangulaire d'un cube (en rouge). La rotation et la translation à travers les arêtes qui n'appartiennent pas au graphe de coupe sont respectivement 1 et 0.*

codent essentiellement le nombre de bandes de quadrilatères entre les sommets singuliers, les arêtes caractéristiques et les bords.

L'idée est donc de réduire considérablement le nombre de variables entières dans le problème (7.3) en calculant un champ de repère \mathcal{F} sur la triangulation en entrée (Chap 3.1.2). Cela nous permet de fixer toutes les rotations r_h , de sorte qu'elles ne soient plus des variables de notre problème. Nous pouvons aller encore plus loin et éliminer une grande partie des translations t_h . En ayant fixé le champ de repère, nous avons fixé les sommets singuliers, de sorte que nous pouvons découper la surface en un disque topologique le long d'un ensemble de demi-arêtes appelé *graphe de découpe* $\mathcal{C} \subset 1 \dots H$. L'idée est d'obtenir un disque topologique avec toutes les singularités de \mathcal{F} situées sur le bord [KALBERER et al. 2007]. Cela nous permet de mettre toutes les translations t_h et les rotations r_h respectivement à zéro et un à travers chaque arête qui n'appartient pas à \mathcal{C} (voir Fig. 7.2). De cette manière, les conditions $t_h \in \mathbb{Z}[i]$ sont satisfaites pour les demi-arêtes concernées. Nous pouvons formuler les contraintes de manière formelle de la manière suivante :

$$\begin{cases} t_h = 0 & \forall h \notin \mathcal{C} \\ r_h = 1 & \forall h \notin \mathcal{C} \\ r_h \text{ constraint par } \mathcal{F} & \forall h \in \mathcal{C} \end{cases} \quad (7.4)$$

Pour résumer, dans le reste de ce chapitre, nous sommes intéressés par le problème suivant :

$$\arg \min_U f(U) \quad \text{t.q. (7.1), (7.2) et (7.4)} \quad (7.5)$$

Notez que les translations $t_h \in \mathcal{C}$ dans cette formulation ne représentent pas de degrés de liberté car elles peuvent être calculées à partir de $U_h h = 1^H$ et $r_h h = 1^H$.

Contributions Au lieu d'utiliser un maillage en T (Sec 3.1.3), notre algorithme de quantification repose sur une décimation du maillage triangulaire d'entrée en tant que structure intermédiaire. Cette nouvelle méthode de quantification présente les avantages suivants par rapport à QGP [CAMPEN et al. 2015] :

- **Facilité de mise en œuvre** : la génération d'un maillage en T et l'exploitation des résultats de quantification sont toutes deux difficiles à mettre en œuvre. En revanche, notre algorithme repose sur une décimation classique [BOMMES et al. 2013a] et les résultats de quantification sont aisément transférables dans la carte finale (voir section 7.2.3).
- **Absence de bords** : avec les maillages en T, il est difficile de gérer les bords [LYON et al. 2019], tandis que notre méthode le permet aisément.
- **Moins restrictif** : nous pouvons obtenir des résultats alignant les singularités comme dans QGP, mais sans aucune face dégénérée dans la carte grossière.

De plus, l'utilisation d'une structure plus simple et moins contrainte ouvre de nouvelles perspectives pour des travaux futurs, tels que la gestion des arêtes caractéristiques à l'étape de quantification ou l'extension de cette méthode au cas volumique.

7.2 Notre quantification

Comparé aux travaux précédents présentés dans le chapitre 3.1.4, nous effectuons notre quantification avec un maillage décimé plutôt qu'un maillage en T. Un aperçu de notre méthode est présenté dans l'algorithme 3, et les étapes sont détaillées dans la suite de cette section.

Algorithm 3: Quantize — Decimated mesh version

Input: Mesh \mathcal{M} , seamless map U^{seamless}
Output: Matrix Q such that $Q \cdot \begin{pmatrix} U \\ 1 \end{pmatrix} = \vec{0}$ implies (7.2).
/ Compute a simpler proxy problem (see §7.2.1) */*
1 $\mathcal{M}^{\text{proxy}}, D \leftarrow \text{decimate}(\mathcal{M});$
2 $U^{\text{proxy}} \leftarrow D \cdot U^{\text{seamless}};$
/ Optimize for integer degrees of freedom (see §7.2.2) */*
3 $\{\omega_h\}_{h=1}^{H^{\text{proxy}}} \leftarrow \text{solve edge vector}(U^{\text{proxy}});$
/ Move the solution back to the original problem (see §7.2.3) */*
4 $Q \leftarrow$ augmented matrix for the linear system
 $\{(D_{\text{next}(h)} - D_h) \cdot U^{\text{seamless}} = \omega_h \quad \forall h \in \{1 \dots H^{\text{proxy}}\}\};$
5 **Function decimate :**
 Input: Mesh \mathcal{M} , seamless map U^{seamless}
 Output: Decimated mesh \mathcal{M}^- and a H^- -by- H mapping relation matrix D
6 $\mathcal{M}^- \leftarrow \mathcal{M};$
7 $H^- \leftarrow H;$
8 $D \leftarrow Id_{H^- \times H};$
9 **while** *can improve* **do**
10 choose the best edge to collapse or to flip;
11 update \mathcal{M}^- and H^- accordingly;
12 find matrix O that updates maps ;
13 $D \leftarrow O \cdot D ;$
14 **end**
15 **return** $\mathcal{M}^-, D;$

7.2.1 Simplification du problème

Pour calculer les degrés de liberté entiers, nous calculons une carte préservant la grille avec un algorithme d'optimisation combinatoire (voir section 7.2.2), qui contraint toutes les variables à être des entiers. Cependant, cette contrainte de "variables entières uniquement" nous éloignerait considérablement d'une solution optimale si elle était directement appliquée à notre maillage d'entrée.

Pour réduire autant que possible le nombre de variables à valeurs réelles, nous utilisons l'algorithme de décimation proposé par [BOMMES et al. 2013a] sur le maillage original. La

décimation alterne entre le renversement et la suppression d'arêtes. Contrairement aux algorithmes de décimation classiques, le choix de l'opération locale à appliquer n'est pas déterminé par la géométrie du maillage, mais par la géométrie de la carte sans couture. Cela force la carte à préserver un Jacobien positif pendant le processus et permet d'utiliser un critère de triangulation de Delaunay pour renverser les arêtes.

Une différence notable avec [BOMMES et al. 2013a] est notre manière de maintenir le lien avec le maillage d'origine. Lorsqu'une opération de renversement ou de suppression d'arête est appliquée, les coordonnées de la carte doivent être propagées sur le nouveau maillage. De nouvelles coordonnées de carte sont données par une expression linéaire des coordonnées de carte précédentes. Dans notre cas, nous ne les évaluons pas pendant la décimation, mais stockons plutôt cette expression linéaire dans un vecteur creux D_h tel que $U^-h = D_h U^{\text{seamless}}$. La matrice $D = [D_1, \dots, D_{H^-}]^T$ est initialisée à l'identité, puis mise à jour à chaque opération d'édition. Pour ce faire, nous construisons une matrice O qui propage les coordonnées de la carte à travers l'opération (voir ci-dessous) et mettons à jour $D \leftarrow OD$.

La matrice O est construite en deux étapes. La première étape "brosse" la carte, c'est-à-dire qu'elle déplace tous les triangles affectés par l'opération de manière à rendre la carte continue. Elle applique essentiellement une rotation de $k\pi/2$ et une translation à chaque triangle, qui peut être représentée par une seule matrice $H^- \times H^-$ à appliquer aux coordonnées de la carte de la surface (voir figure 7.3). La seconde étape applique l'opération combinatoire (soit un renversement, soit une suppression). Elle génère et supprime certains triangles. Les coordonnées de carte des coins des nouveaux triangles sont égales aux coordonnées de carte des sommets correspondants ; cela peut également être représenté dans une matrice. La matrice O est obtenue après redimensionnement et réordonnement des lignes, pour refléter le nouvel index des coins de triangles après l'opération.

Avec la matrice D , nous pouvons transférer n'importe quelle carte du maillage d'entrée vers le maillage décimé. Nous l'utilisons directement pour transférer la carte sans couture vers le maillage décimé $U^{\text{proxy}} = D \cdot U^{\text{seamless}}$ afin d'obtenir un objectif géométrique pour notre carte préservant la grille grossière. L'autre avantage du calcul de D est que les contraintes agissant sur les cartes du maillage décimé peuvent être propagées aux cartes du maillage d'entrée. C'est une façon simple de transférer les résultats de quantification au problème d'origine (voir section 7.2.3).

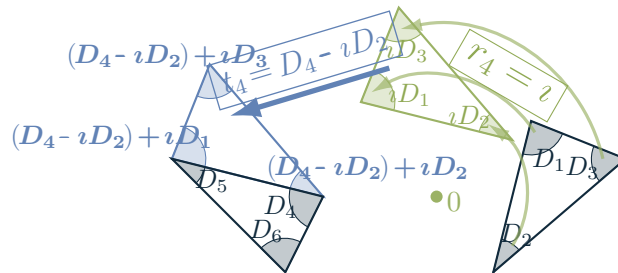


FIGURE 7.3 – Pour enlever la discontinuité entre les triangles noirs, les coordonnées de carte symboliques D_i du triangle noir de droite doivent être déplacées vers le triangle bleu par l'expression linéaire $D_i \leftarrow t_4 + r_4 D_i = D_4 - iD_2 + iD_i$. De plus, les rotations sont mises à jour avec $r_i \leftarrow r_4 r_i$ et $r_{\text{opp}(i)} \leftarrow r_i^{-1}$.

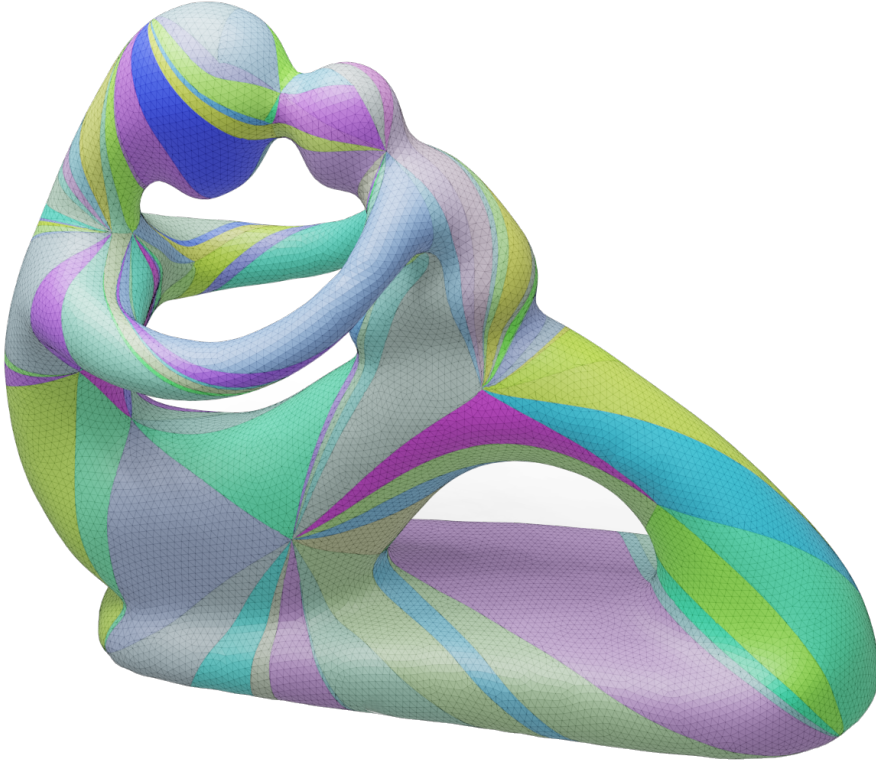


FIGURE 7.4 – La première étape de notre algorithme de quantification consiste à décimer le maillage triangulaire initial en un maillage grossier ne comportant que des nœuds singuliers.

7.2.2 Optimisation des degrés de liberté entiers

Cette étape recherche la carte de préservation de grille U^- du maillage décimé qui a uniquement des coordonnées de carte à valeur entière et qui est aussi proche que possible de U^{proxy} , la carte sans couture transférée du maillage d'entrée. Pour ce faire, nous définissons la différence entre U^{proxy} et U^- comme suit :

$$E(U^-) = \sum_h \frac{\|(U_{\text{next}(h)}^- - U_h^-) - (U_{\text{next}(h)}^{\text{proxy}} - U_h^{\text{proxy}})\|^2}{\|U_{\text{next}(h)}^{\text{proxy}} - U_h^{\text{proxy}}\|^2} \quad (7.6)$$

On peut remarquer que cette énergie est équivalente à E dans QGP pour les arêtes alignées avec un axe dans la carte.

Pour faciliter la manipulation de la carte, introduisons un autre ensemble de variables :

$$\omega_h = U_{\text{next}(h)}^- - U_h^- \quad (7.7)$$

Cette représentation est préférée car elle caractérise exactement la structure combinatoire du maillage quad à extraire à la fin, c'est-à-dire le nombre de bandes de quads à traverser le long des chemins entre les singularités. De plus, c'est l'extension naturelle du nombre de subdivisions à appliquer à chaque arête qui est optimisé sur les maillages en T.

Les conditions pour que ω corresponde à une carte préservant la grille U^- sont les suivantes :

$$\begin{cases} \omega_h \in \mathbb{Z}[z] & (7.8a) \\ \omega_h = -r_h \omega_{\text{opp}(h)} & (7.8b) \\ \sum_{h \in T} \omega_h = 0, \forall T & (7.8c) \\ \det(J_T) > 0, \forall T & (7.8d) \end{cases}$$

où T désigne les triangles de $\mathcal{M}^{\text{proxy}}$, $\det(J_T)$ est le déterminant de la matrice de Jacobienne de la carte sur le triangle T , et r_h est une fois de plus la rotation de la carte à travers l'arête h , mais maintenant définie sur le maillage grossier.

Après avoir défini $U_0^- \leftarrow 0$, la condition (7.8a) force toutes les coordonnées à être des complexes de Gauss (en supposant que le graphe des arêtes est connecté), (7.8b) impose la rotation à être r_h , (7.8c) est nécessaire pour reconstruire U^- sur un triangle, et (7.8d) garantit que la fonction est une carte valide. Maintenant, notre problème est de minimiser (7.6) sous ces contraintes.

Étape d'initialisation L'attribution de $\omega \leftarrow 0$ respecte toutes les conditions sauf (7.8d). Dans QGP, l'algorithme qui optimise la qualité de la carte peut également être modifié pour imposer (7.8d) à partir de $\omega \leftarrow 0$, mais cette approche n'a pas été prouvée pour converger vers une carte valide dans notre cas. Pour cette raison, nous préférons initialiser ω directement avec une carte valide, qui est disponible en mettant à l'échelle la carte sans couture par un facteur de 2 jusqu'à ce que toutes les variables deviennent entières, ce qui est garanti par la précision limitée des points flottants.

En pratique, nous évitons l'échelle extrême en alignant les valeurs $\omega_h^{\text{proxy}} = U_{\text{next}(h)}^{\text{proxy}} - U_h^{\text{proxy}}$ sur le nombre de Gauss le plus proche qui n'est pas nul. L'inconvénient est que ω peut ne pas respecter (7.8c). Dans ce cas, nous le corrigeons en changeant l'un des ω_h du triangle. Ensuite, l'application de (7.8b) sur la demi-arête choisie h déplace probablement le problème vers le triangle adjacent, et le processus se poursuit jusqu'à ce qu'il atteigne un triangle où la modification de ω rend (7.8c) valide. La séquence de ω_h à modifier est trouvée par l'algorithme de plus court chemin décrit dans le paragraphe suivant.

Étape d'optimisation Notre algorithme d'optimisation parcourt $(h_{\text{seed}}, a_{\text{seed}})$ où h_{seed} est une demi-arête du maillage décimé et $a_{\text{seed}} \in \{1, \iota, -1, -\iota\}$. À chaque itération, l'algorithme cherche un vecteur α qui respecte (7.8a), (7.8b) et (7.8c) à ajouter à ω de sorte que $\alpha_{h_{\text{seed}}} = a_{\text{seed}}$; puis, si $\omega + \alpha$ respecte (7.8d) et $E(\omega + \alpha) < E(\omega)$, α est ajouté à ω . L'optimisation se répète jusqu'à ce qu'il n'y ait plus de couple $(h_{\text{seed}}, a_{\text{seed}})$ qui puisse améliorer l'énergie.

Le calcul de α à partir de $(h_{\text{seed}}, a_{\text{seed}})$ est effectué en appliquant des modifications atomiques : $(\alpha_h, \alpha_{h'}) \leftarrow (\alpha_h + a, \alpha_{h'} - a)$, où $h' \in \{\text{next}(h), \text{prev}(h)\}$. Comme illustré dans la Fig. 7.5, cela déforme essentiellement la carte d'un triangle comme si l'un de ses coins était translaté de a . Chaque modification de ce type préserve toutes les conditions de carte préservant la grille sauf (7.8b). Cette dernière condition étant satisfaite en choisissant soigneusement l'ensemble de modifications atomiques à appliquer.

Nous cherchons une liste de modifications atomiques, qui commence par une modification telle que $\alpha_{h_{\text{seed}}} = a_{\text{seed}}$, et telle que chaque modification ultérieure corrige une condition (7.8b) qui a été rompue lors de la modification précédente. La dernière modification de la liste complète un cycle et résout (7.8b) sur h_{seed} au lieu de la rompre sur une autre demi-arête.

Cette liste de modifications atomiques est calculée sous la forme d'un chemin dans un graphe. Nous définissons ce graphe comme suit ; chaque nœud est une paire (h, a) qui signifie qu'il faut

ajouter a à α_h . Chaque arête orientée représente une des deux modifications atomiques qui ajoutent a à α_h . Dans la plupart des cas, ces modifications rompent (7.8b) sur h' , et sont représentées par l'arête $(h, a) \rightarrow (\text{opp}(h'), r_{h'}a)$. Dans le cas contraire, l'arête est $(h, a) \rightarrow \text{succes}$. Comme illustré dans la Fig. 7.6, un chemin dans le graphe est un ensemble d'opérations atomiques cohérentes.

Concrètement, nous trouvons l'ensemble de modifications atomiques en utilisant l'algorithme de Dijkstra qui démarre du nœud $(h_{\text{seed}}, a_{\text{seed}})$ et se termine lorsqu'il atteint un nœud de réussite (par exemple, en complétant un cycle). Les poids des arêtes sont déterminés comme suit :

- si l'ajout de a inverse le triangle, $w_{h,a} = 1/\epsilon^2$
- sinon

$$w_{h,a} = \begin{cases} (3 - \delta)/\epsilon & \text{if } \delta < 1 \\ 1/(1 + \delta) & \text{if } \delta \geq 1 \end{cases} \quad (7.9)$$

avec $\delta = 2a \cdot (U_{\text{next}(h)}^{\text{proxy}} - U_h^{\text{proxy}} - \omega_h)$. Nous observons que si $\delta > 1$, l'ajout de a à ω_h diminue l'énergie. Cependant, si $\delta < 1$, l'ajout de a éloigne ω_h de l'optimal et finalement augmente l'énergie. C'est pourquoi nous séparons les cas à $\delta = 1$.

Ces poids sont choisis pour favoriser les cycles qui diminuent l'énergie $E(U^-)$ et qui conservent $\det(J_T) > 0$. Passer par des arêtes qui diminuent la différence entre ω et l'objectif est peu coûteux ($\delta > 1$), augmenter cette différence est beaucoup plus coûteux ($\delta < 1$), et nous permettent également (à un coût élevé) d'inverser un triangle ($\det(J_T) > 0$). Dans ce dernier cas, le cycle sera accepté seulement s'il passe une nouvelle fois par le même triangle pour rétablir la positivité du Jacobien.

La séparation de ces cas est réalisée par $\epsilon \rightarrow 0^+$, qui est implémentée en stockant les poids sous la forme d'un couple n, x pour $\epsilon^{-n}x$.

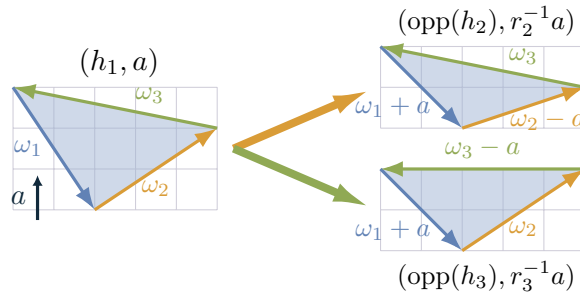


FIGURE 7.5 – Les arêtes qui démarrent au nœud (h_1, a) doivent ajouter a à ω_1 , ce qui implique de soustraire a à ω_2 ou ω_3 ... ce qui nous amène aux nœuds $(\text{opp}(h_2), r_2^{-1}a)$ ou $(\text{opp}(h_3), r_3^{-1}a)$.

7.2.3 Déplacement de la solution vers le problème initial

Nous devons maintenant calculer la carte finale préservant la grille U^{grid} : cela signifie résoudre le problème d'optimisation de carte sans couture, avec des contraintes supplémentaires qui fixent les variables entières pour qu'elles soient compatibles avec notre quantification, et $t_h \in \mathbb{Z}[i]$.

Pour obtenir ces contraintes, nous rappelons que ω est défini par $\omega_h = U_{\text{next}(h)}^- - U_h^-$ et que cette carte est une version décimée de U^{grid} si et seulement si $U^- = DU^{\text{grid}}$. Par conséquent, les contraintes linéaires $\omega_h = D_{\text{next}(h)}U^{\text{grid}} - D_hU^{\text{grid}}$ imposent que l'intégration d'un chemin entre des singularités (dans la carte) donne un vecteur entier. Si nous ajoutons qu'une singularité est située à 0, toutes les singularités seront sur une position de complexe Gaussien.

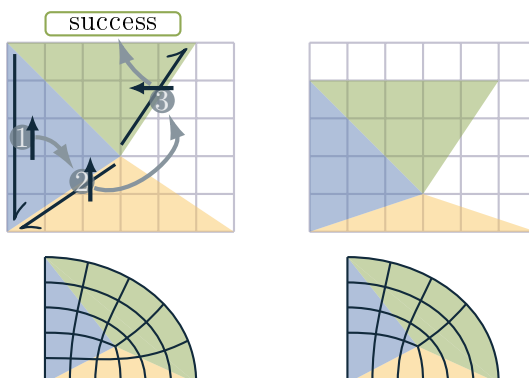


FIGURE 7.6 – L’ajout du vecteur α (flèches noires) à ω modifie la carte de gauche à droite en haut. La structure du maillage quad est modifiée en conséquence (en bas). La liste des arêtes grises du graphe dirigé est le moyen de calculer α . Notez que les arêtes grises ne forment pas de cycle ici, car la surface a un bord (voir la section 7.3.2).

Pour compléter les conditions (7.2), nous devons également nous assurer que $t_h \in \mathbb{Z}[2]$, ce qui est garanti automatiquement grâce au graphe de découpe. En effet, pour chaque demi-arête h du graphe de découpe, nous pouvons tracer un cycle (non contractible) qui traverse la découpe exactement une fois, et sur h . L’intégral des coordonnées de la carte le long de ce cycle est nécessairement entier en raison de la quantification... et égal à t_h .

Relaxer les contraintes entières inutiles La décimation d’un maillage peut ne pas avoir supprimé tous les degrés de liberté à valeur réelle. Dans ce cas, la quantification les force à être des entiers, ce qui est inutile. Pour récupérer ces degrés de liberté, nous définissons un vecteur creux Y tel que Y soit nul sur les variables entières, U respecte (7.1) et $Y + U$ respecte (7.1,7.2). La carte U^{grid} est le U qui minimise $f(U)$ sous ces contraintes.

7.3 Implémentation

Cette section présente comment notre implémentation optimise $f(U)$ sous des contraintes linéaires (§7.3.1) et comment aligner la carte avec des réseaux complexes d’arêtes caractéristiques (§7.3.2) qui sont courants sur les modèles CAO.

7.3.1 Optimisation numérique

Comme décrit dans la section 3.1.3, le calcul d’une carte préservant la grille se fait en deux étapes. D’abord, on calcule une carte sans couture valide, puis on utilise cette carte pour trouver des contraintes de quantification nous permettant de calculer une carte préservant la grille valide.

Ces deux optimisations sont effectuées comme suggéré dans [BOMMES et al. 2012] avec une matrice de réduction de variable M telle que $U = MX$ satisfasse toutes les contraintes linéaires pour tout vecteur X . Cela permet de minimiser $f(MX)$ sans contraintes et d’obtenir le résultat par $U = MX$.

7.3.2 Arêtes caractéristiques et bords

L'alignement de la carte de grille préservant les arêtes caractéristiques et/ou les bords de surface permet de produire des maillages quads alignés avec ces arêtes. Cette contrainte peut être formulée comme suit : pour chaque demi-arête h qui est sur une arête caractéristique, la partie réelle ou la partie imaginaire (selon le champ de repères) de U doit être constante. Cette contrainte doit être propagée à l'étape de quantification, où elle écarte certains nœuds (h, a) du graphe auxiliaire. De plus, pour les surfaces avec des bords, notre recherche de Dijkstra ne doit pas seulement chercher des cycles, mais également des chemins qui commencent et finissent sur le bord (comme illustré sur la Fig. 7.6).

Contrairement à [LYON et al. 2019], il est suffisant de ne pas placer une arête caractéristique sur le bord de la surface pour générer des cartes de grille préservant des bords libres. La Fig. 7.7 montre un exemple de carte de grille entière à bord libre.

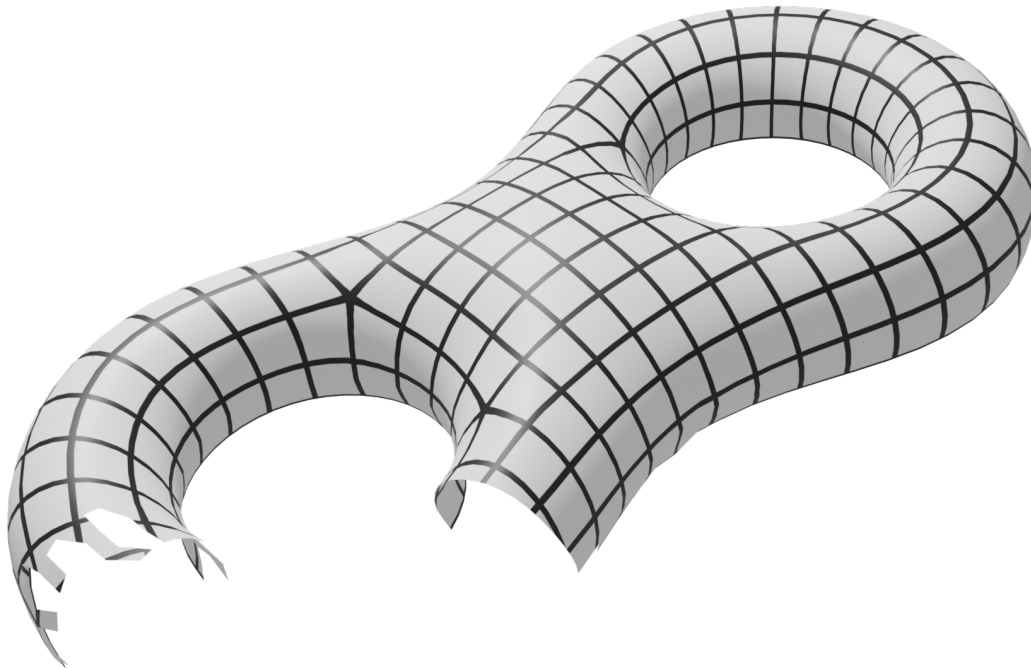


FIGURE 7.7 – *La génération de carte préservant la grille à bord libre avec notre algorithme de quantification est simplement réalisée en omettant les contraintes d'alignement des bords. Pour ce modèle, l'alignement des bords n'est pas imposé là où le maillage a été coupé.*

7.4 Resultats

Les performances de notre algorithme ont été évaluées sur la même base de données CAO que dans [REBEROL et al. 2021]. Elle contient 1000 modèles de la base de données ABC [KOCH et al. 2019] et les 114 modèles de la base de données MAMBO [LEDOUX 2019]. Ces modèles CAO ont de nombreuses arêtes caractéristiques qui rendent difficile la production de cartes préservant la grille. Nous avons réussi à générer une entrée valide pour notre algorithme (cartes sans couture) avec un alignement de bords/ d'arêtes caractéristiques pour 1111 modèles. Sur chaque modèle, nous avons effectué deux tests de quantification : l'un pour produire un résultat grossier (Fig.7.9), l'autre pour générer un maillage quad à la résolution de l'entrée (Fig.7.8). Les résultats grossiers

sont obtenus en réduisant l'échelle de U^{proxy} avant d'appliquer l'optimisation des degrés de liberté entiers (Sec. 7.2.2).

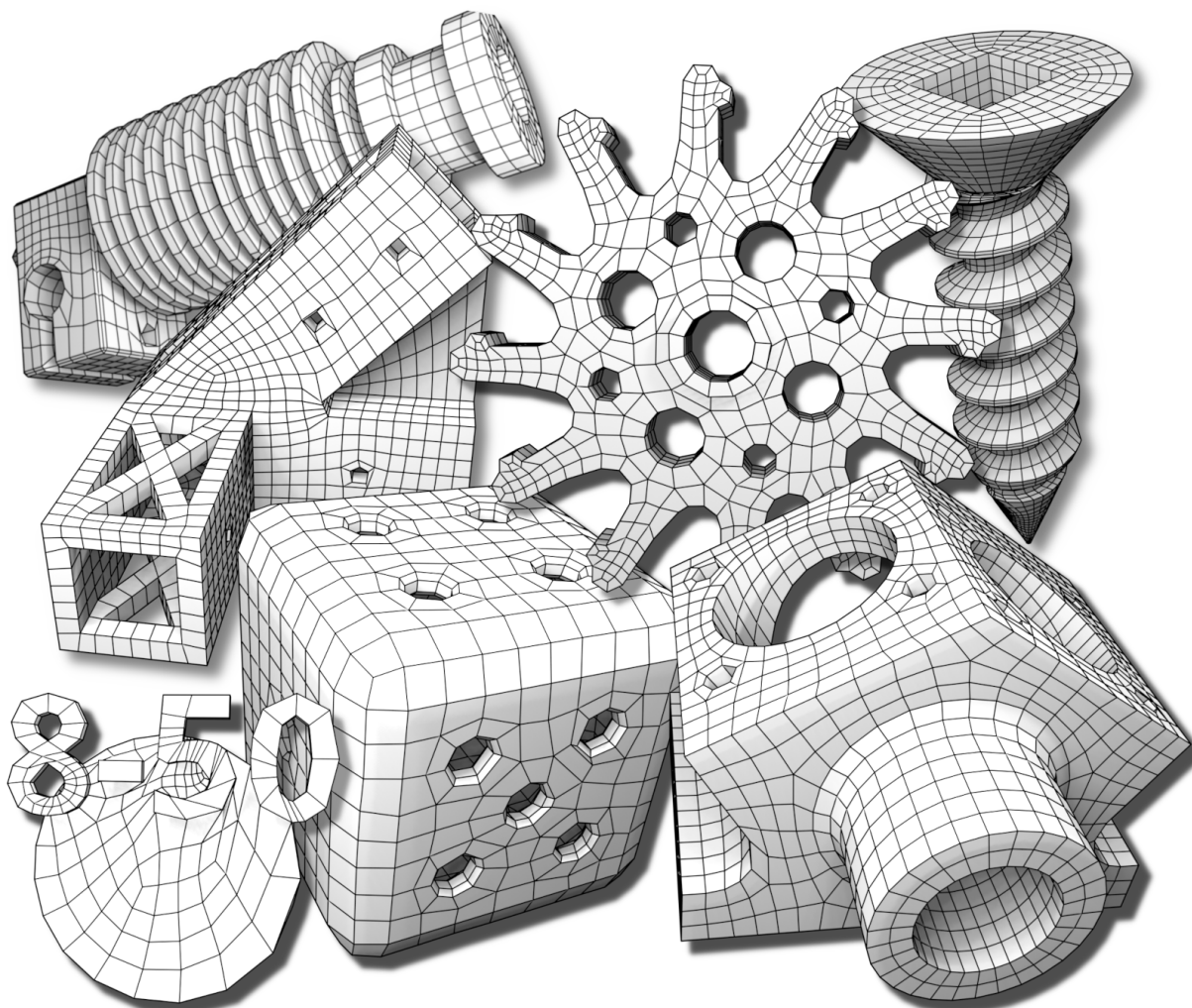


FIGURE 7.8 – Exemples de maillages quads générés par notre algorithme.

7.4.1 Performances

Notre algorithme est généralement aussi rapide que QGP, ce qui est presque négligeable lorsqu'il est comparé à la chaîne complète qui calcule les cartes préservant les grilles. Plus précisément, notre étape de décimation prend du temps proportionnel à la taille du maillage d'entrée (quelques secondes). L'étape de quantification, évaluée sur la génération d'un maillage grossier, prend généralement aussi quelques secondes, et est proportionnelle au carré de la taille du maillage décimé. Nous pouvons observer que, lorsque l'initialisation de ω nécessite de mettre à l'échelle la carte sans couture d'entrée, le temps de quantification est augmenté par le même facteur.

En pratique, la mise à l'échelle de la carte sans couture d'entrée n'est nécessaire que lorsque le maillage a des arêtes caractéristiques très proches les unes des autres. Le pire cas que nous avons observé a nécessité une mise à l'échelle de 64 et est dû au triangle rouge de la Fig. 7.10 ; il

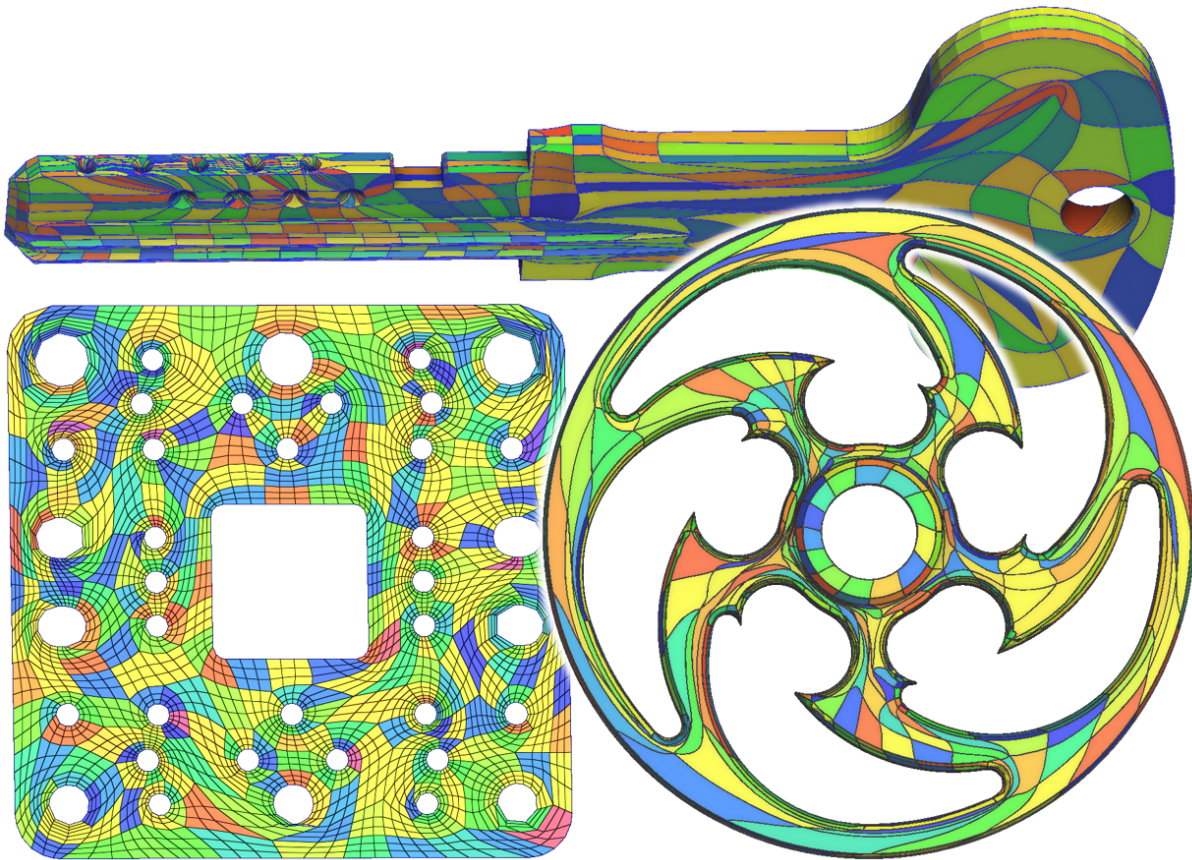


FIGURE 7.9 – La quantification grossière segmente le modèle en subdivision en forme de quadrilatère. Il est souvent nécessaire de subdiviser les cartes (en bas à gauche) pour obtenir un maillage quadrilatère avec une géométrie juste.

a un point singulier (étoile) comme sommet, une arête de caractéristique (blanche), un mauvais rapport hauteur/largeur et une très petite taille.

7.4.2 Qualité

Il existe de nombreux critères pour évaluer la qualité d'un maillage quadrilatère, mais il est assez difficile d'estimer l'impact de l'algorithme de quantification indépendamment des autres étapes de la chaîne (champ de repère, calcul d'une carte sans couture valide, extraction de maillage quadrilatère, lissage, etc.).

Dans la suite, nous présentons une comparaison qualitative avec QGP, puis discutons de la façon dont l'utilisation d'un maillage intermédiaire différent influence l'ensemble des structures de maillage quad qui peuvent être générées à partir de la quantification.

Comparaison Nous avons comparé notre algorithme avec QGP sur quelques modèles de la base de données utilisée et le lapin de Stanford pour avoir un exemple sans arête caractéristique. Nous avons obtenu des résultats très similaires présentés dans la Fig. 7.11. Pour ces modèles, nous observons que QGP est de 2 à 5 fois plus rapide que notre méthode.

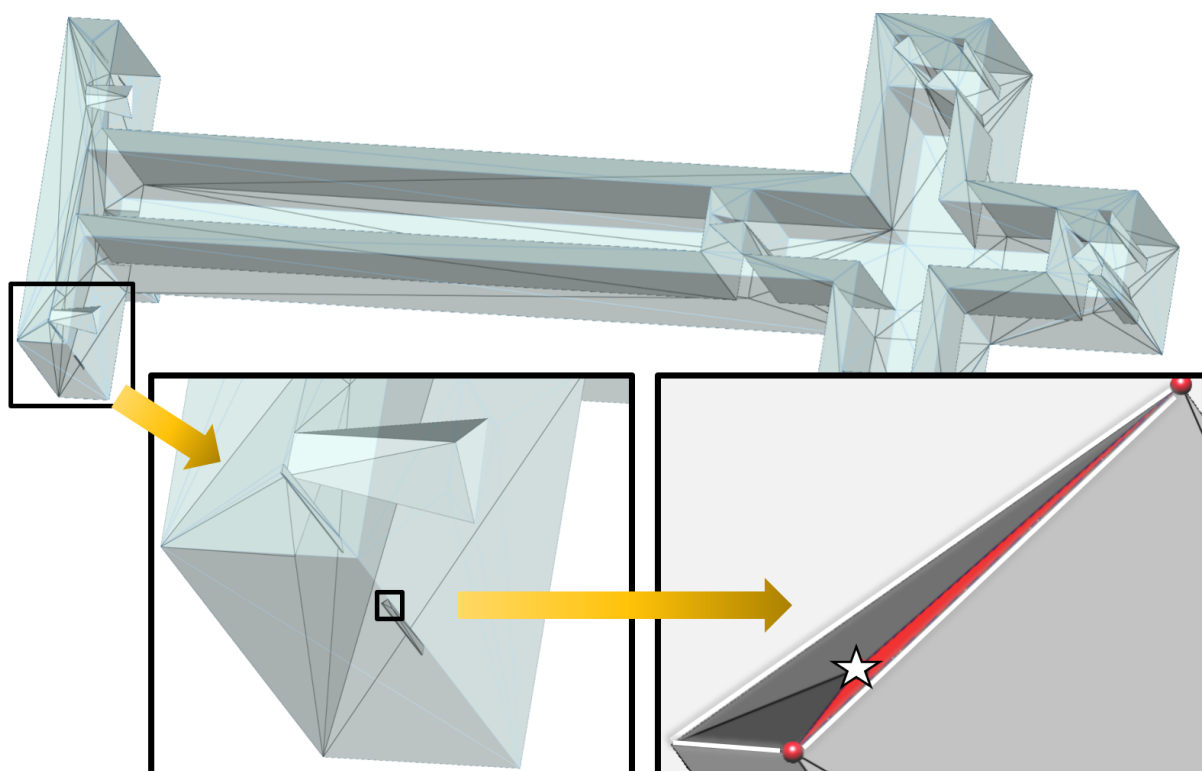


FIGURE 7.10 – *Initialisation de la quantification avec une mise à l'échelle élevée. Le triangle rouge de ce maillage décimé (visible dans le second plan rapproché) nécessite une mise à l'échelle de la carte sans couture par un facteur 64 pour éviter une carte dégénérée due au sommet en étoile qui est fixé à son arête blanche.*

Discussion sur la dégénérescence des faces Les cartes préservant la grille sont représentées par une fonction linéaire sur chaque face du maillage. Comme les cartes doivent avoir un jacobien positif sur chaque face, le maillage a une influence sur les valeurs que les degrés de liberté entiers peuvent prendre. En conséquence, le maillage d'origine, le maillage décimé et le maillage en T ne sont pas tous capables de représenter le même ensemble de maillage quadrilatère.

Avec la condition supplémentaire que les arêtes du maillage en T doivent être alignées sur un axe de la carte, les maillages en T ne peuvent définir qu'un ensemble très limité de cartes préservant la grille. Ce problème est toutefois compensé en acceptant certaines faces dégénérées ; longueur d'arête nulle, c'est-à-dire jacobien nul dans la carte. Le prix de cette relaxation des contraintes est que le maillage en T doit être modifié et ré-inséré avant de produire le maillage quadrilatère final.

En utilisant un maillage décimé, nous pouvons représenter approximativement le même nombre de cartes préservant la grille sans avoir besoin de dégénérer une quelconque face (voir la Fig.7.12). Pour des résultats similaires, il est plus simple de transférer la carte vers le maillage d'origine. Plus précisément, comme illustré dans la Fig.7.13, il est très souhaitable de pouvoir aligner certaines singularités ou coins de bord concaves, ce qui est possible avec une carte valide sur notre maillage décimé, mais nécessite des faces dégénérées sur les maillages en T.

Dans ce chapitre 7, nous avons présenté un algorithme de quantification rapide et open-source,

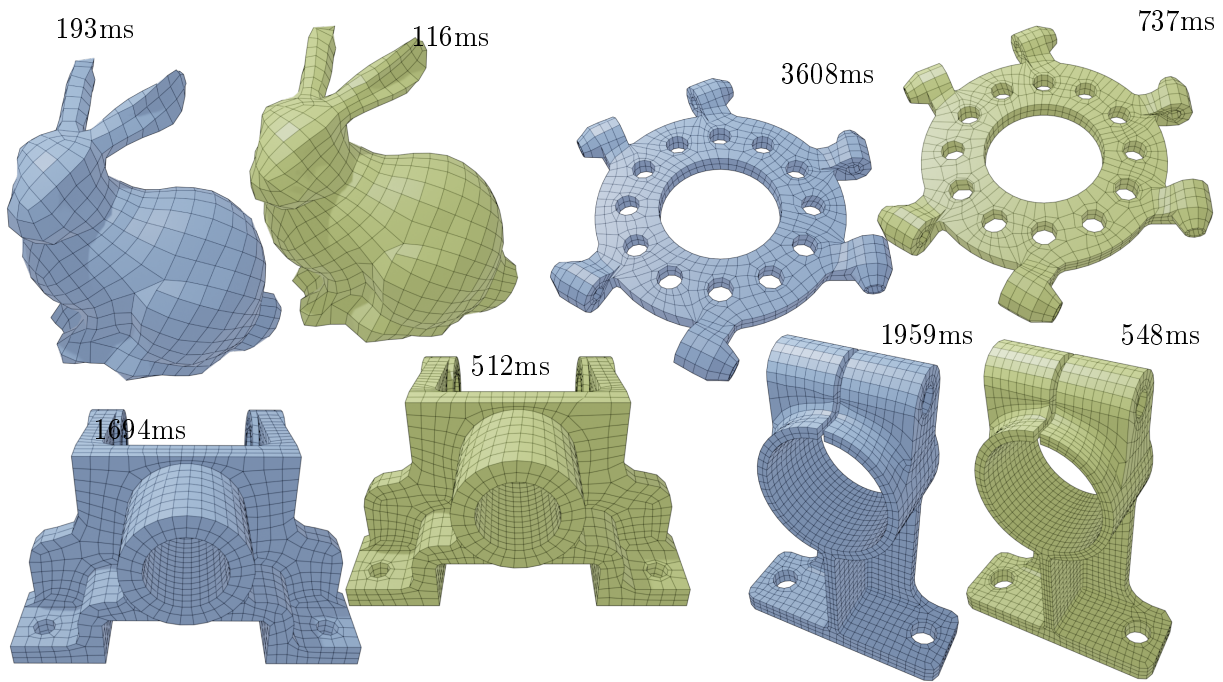


FIGURE 7.11 – Nos résultats (bleu) sont très similaires à ceux de QGP (vert) [CAMPEN et al. 2015]. Les temps de la chaîne de quantification sont donnés à côté de chaque modèle. Les temps incluent le calcul d'un maillage intermédiaire (maillage décimé ou maillage en T) et l'algorithme de quantification lui-même.

qui nous permet de nous assurer que le passage d'une carte sans couture à une carte préservant la grille pour extraire un maillage quadrilatère n'induit pas d'erreur. Un point important que nous n'avons pas abordé dans ce chapitre est une façon d'obtenir une carte sans couture sans retournement qui est l'entrée de cet algorithme de quantification. Dans le prochain chapitre 8, nous allons donner une méthode itérative pour enlever les retournements d'une carte sans couture afin de la donner en entrée de l'algorithme de quantification de ce chapitre, dans le but de produire un maillage quadrilatère adapté à une simulation de déformation donnée.

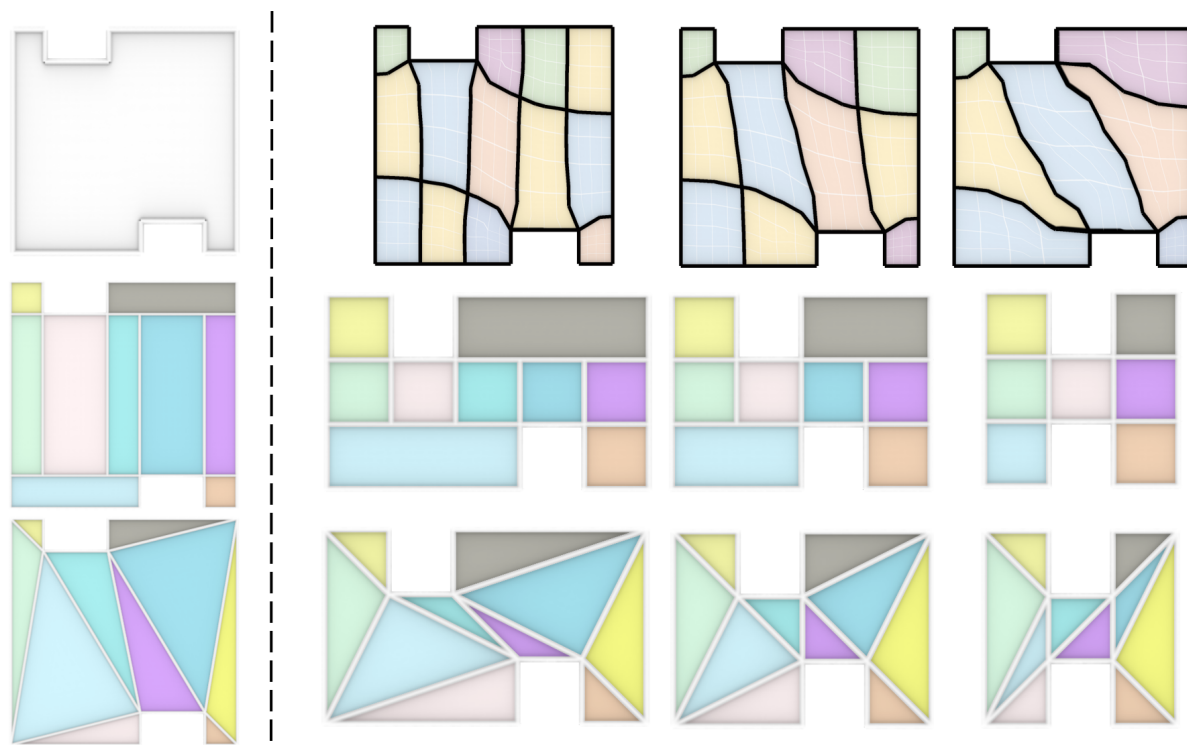


FIGURE 7.12 – Côté gauche : un modèle (haut) avec un maillage en T intermédiaire (milieu) et un maillage décimé intermédiaire (bas). Côté droit : 3 décompositions en blocs possibles (haut), la carte correspondante sur le maillage en T (milieu) et sur le maillage décimé (bas). Un quad du maillage en T doit être dégénéré dans la colonne du milieu, et des quads inversés sont nécessaires dans la colonne de droite, alors que les cartes sont toujours valides avec notre maillage décimé.

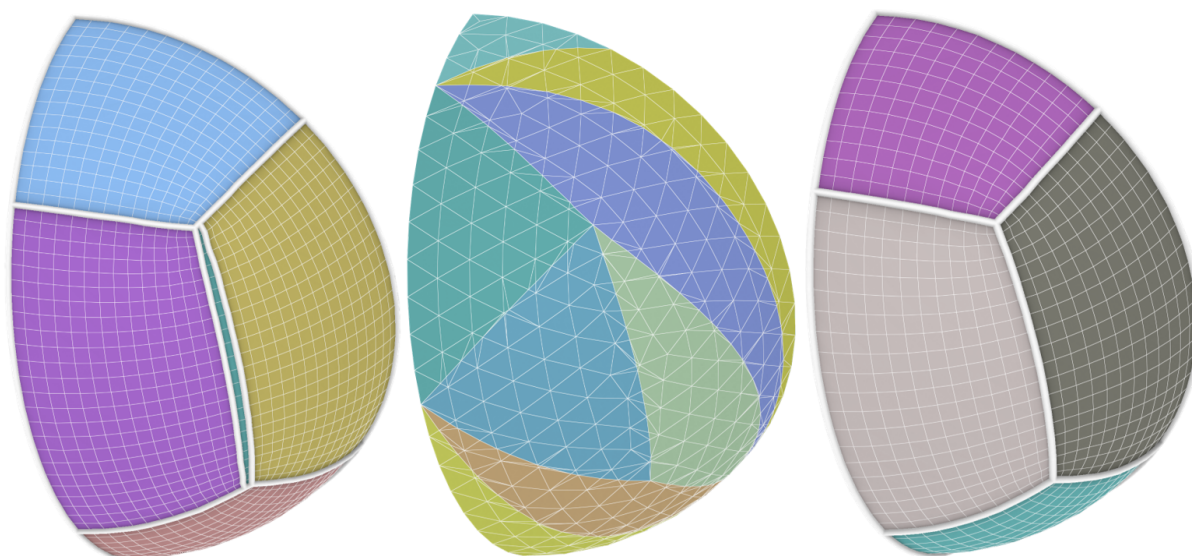


FIGURE 7.13 – Sans subdivision dégénérées, la quantification de maillage en T ne peut pas enlever la subdivision mince pour aligner les singularités (gauche). Notre maillage décimé (milieu) n'a pas besoin de dégénérer des faces pour le faire (droite).

Génération de maillage adapté à des déformations avec Quadcover

Dans ce chapitre, nous allons détailler la méthode que nous avons utilisée pour réaliser des simulations de grandes déformations sur le logiciel industriel [*NUMEA* s. d.]. Ce chapitre se restreint à la 2D, car nous ne sommes pas parvenus à construire avec la méthode Cubecover (Sec 3.2) et nos contributions de champ de repère 3D (Chap 4, 6) des maillages hexaédriques sur les modèles industriels 3D utilisés dans [*NUMEA* s. d.]. Nous donnerons dans la conclusion de ce manuscrit notre avis sur les avancées qu'il reste à faire pour que ce processus que nous réalisons en 2D puisse être utilisé en 3D.

8.1 Alternance entre maillage et simulation de déformation

Dans cette section, nous présentons le processus qui nous permet d'alterner entre des étapes de maillage quadrilatère de la forme initiale de l'objet simulé, et des étapes de simulation de déformation sur le maillage généré. Itérativement, le maillage utilise les résultats de la simulation précédente afin de s'adapter aux déformations.

8.1.1 Construction du maillage initial

Notre objectif est de générer un maillage quadrilatère qui reste de bonne qualité pendant toute une simulation de déformation pour qu'elle aboutisse. Pour ce faire, nous avons besoin d'information sur comment la simulation de déformation va agir sur l'objet 2D simulé, afin de déterminer un maillage adapté. La première étape est donc de concevoir un maillage à partir de l'état initial de l'objet et de lancer une simulation de déformation sur ce maillage dont nous utiliserons les résultats pour concevoir un nouveau maillage adapté à la simulation (Sec 8.1.2). À partir de la description CAO de l'état initial de l'objet, nous utilisons une méthode de triangulation [SHEWCHUK 2005] pour en obtenir un maillage triangulaire T_0 . Nous appliquons la méthode Quadcover (Chap 3) sur le maillage triangulaire pour obtenir un maillage quadrilatère de l'état initial de l'objet Q_0 .

8.1.2 Construction d'un maillage adapté à une simulation de déformation

Nous utilisons Q_0 comme maillage initial de la simulation de déformation. Pour chaque pas de temps n de la simulation, nous obtenons le maillage déformé Q_n . Si la simulation converge

jusqu'au pas de temps final, nous considérons que la simulation a réussi. Sinon, si la simulation diverge à l'étape n_f , pour tout $n \leq n_f$, on construit des maillages triangulaires T_n en divisant en deux les quadrilatères des maillages Q_n . Il est important que la méthode de division soit la même pour tous les pas de temps n , afin que tous les maillages triangulaires aient la même connectivité, avec simplement des positions de sommets différentes. Puis, à partir des maillages triangulaires T_n , on utilise les sections 8.2 et 8.3 pour générer un nouveau maillage de l'état initial Q_0 , d'avantage adapté à la simulation de déformation réalisée. Nous répétons ce processus (voir Algo 4) jusqu'à ce que la simulation converge entièrement.

Algorithm 4: Alternance maillage et simulation

Input: Maillage de l'état initial de l'objet Q_0

Output: Un maillage par pas de temps de la simulation $(Q_n)_{n \leq N}$

```

1 Function QuadcoverIteratif( $Q_0$ ) :
2    $n_f \leftarrow 0$ 
3   while  $n_f < N$  do
4     Lancement de la simulation de déformation sur le maillage  $Q_0$ .
5      $(Q_n)_{n \leq n_f} \leftarrow$  un maillage quadrilatère par pas de temps réussi de la simulation
6     forall  $n \leq n_f$  do
7        $T_n \leftarrow Q_n$  divisé en un maillage triangulaire
8     end
9      $Q_0 \leftarrow$  calcul d'un maillage initial adapté aux déformations  $(T_n)_{n \leq N}$ 
10  end
11  return  $(Q_n)_{n \leq N}$ 

```

8.2 Champ de repère adapté à plusieurs géométries

Maintenant que nous avons exprimé le processus d'alternance entre maillage et simulation, nous allons présenter dans cette section les améliorations que l'on a apporté à la construction de champ de repère présentée dans la section 3.1.2 pour prendre en compte les déformations qui vont être appliquées au maillage quadrilatère produit. Plutôt que de n'avoir qu'un seul maillage triangulaire en entrée de l'état initial de l'objet, nous avons un maillage triangulaire par pas de temps d'une simulation de déformation. Les maillage $(T_n)_{n \leq N}$ sont tous de même connectivité, mais ont des géométries qui évoluent.

8.2.1 Valences sur le bord adapté à toute les géométries

Nous avons vu dans la section 6.2 consacré à la 3D que pour un angle dièdre de bord φ d'un maillage tétraédrique, un champ de repère 3D associe la valence d'arête $\text{round}(2\varphi/\pi)$ par la méthode décrite dans la section 3.2. C'est la même chose en 2D, si deux arêtes de bord intersectant en un sommet de bord v forment un angle α_v , le sommet v reçoit par la méthode 3.1 la valence $\text{round}(2\alpha_v/\pi)$. Nous avons déjà vu dans les chapitres 4 et 5 que cela pose un problème dans le cas où les deux arêtes de bord forment un petit angle inférieur à 45 degrés. Dans ce cas, la valence du sommet d'intersection est 0 et il est impossible d'obtenir un maillage quadrilatère à partir d'un champ de repère obtenu par la méthode de la section 3.1.2. Lorsque l'on veut construire un maillage adapté à plusieurs géométries en entrée, la valence $\text{round}(2\alpha_v/\pi)$ obtenue sur la géométrie initiale peut être trop faible pour permettre la validité du maillage sur les géométries

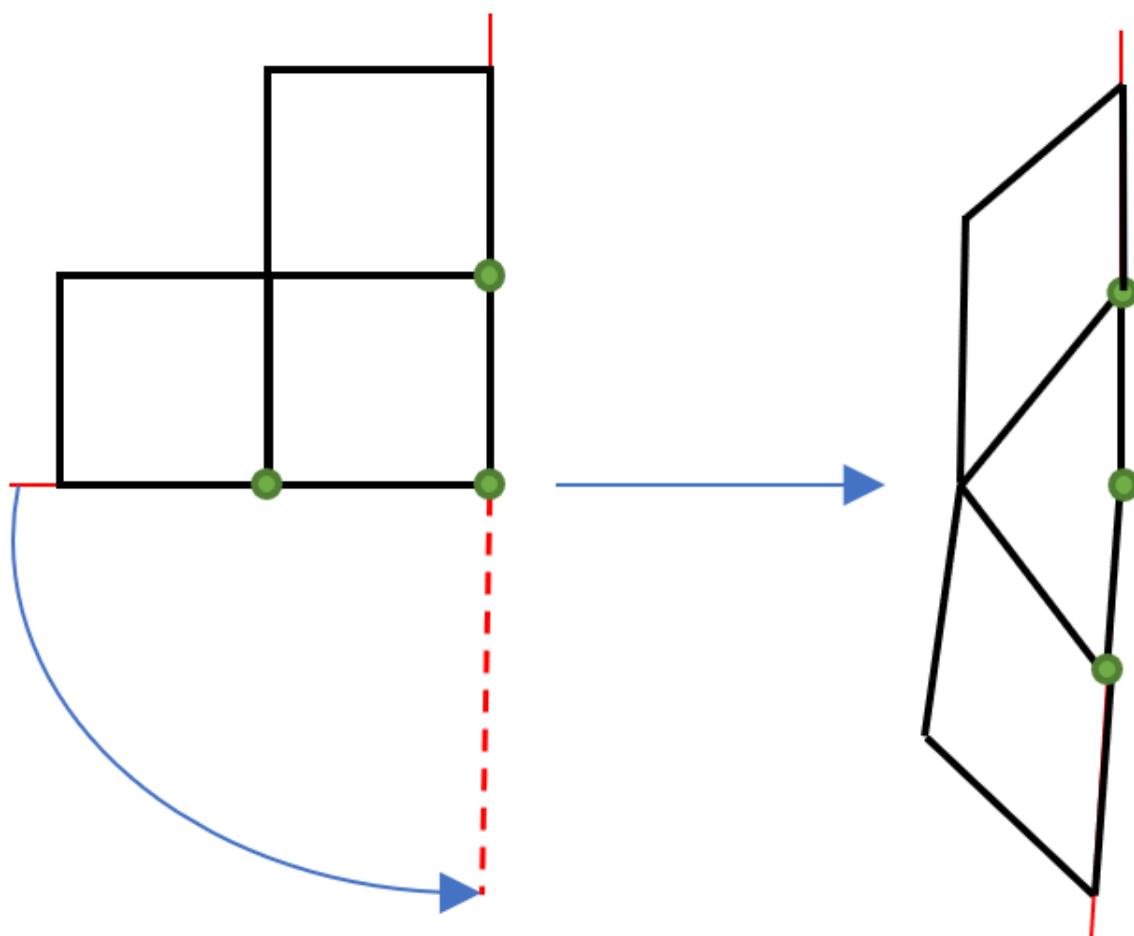


FIGURE 8.1 – Après déformation, le maillage devient invalide car le sommet du bord n'est adjacent qu'à un quadrilatère, ce qui n'est pas suffisant.

suivantes (voir Fig 8.1). Une valence trop élevée par rapport à un angle de bord limite la qualité des éléments mais n'empêche pas leur validité, contrairement à une valence trop faible.

La section 8.1 nous a donné N maillages triangulaires $(T_n)_{n \leq N}$ de même connectivité mais avec des positions de sommets (géométries) différentes. Pour assurer que les valences de bord soient compatibles à la totalité des géométries des maillages $(T_n)_{n \leq N}$ en entrée, nous utilisons l'angle maximal β_v de chaque sommet de bord v parmi toutes les géométries :

$$\beta_v = \max_{n \leq N} \alpha_{v,n}$$

Ainsi, à partir des angles maximums β , nous pouvons déterminer des valences de bord $k \geq 1$ adapté à toutes les géométries pour chaque sommet de bord v :

$$k_v = \max\{2\beta_v/\pi, 1\}$$

8.2.2 Correspondances de champ de repère à valence sur le bord fixé

Nous avons donné dans le chapitre 6 une méthode pour calculer en 3D un champ de repère à partir de valences sur le bord fixé. Pour réaliser la même chose en 2D, nous avons adapté la méthode du chapitre 5 pour qu'elle agisse sur tous les sommets de bord plutôt que sur seulement les sommets de petit angle qui recevaient une valence invalide 0 par la méthode de champ de repère de la section 3.1.2. Nous utilisons le maillage triangulaire de la première géométrie T_0 , et nous adaptons la section 5.2 pour obtenir la valence que nous désirons sur chaque sommet de bord v . Plus précisément, pour chaque sommet de bord v avec un angle de bord $\alpha_v \neq k_v\pi/2$ sur le maillage T_0 , nous fixons la dérivée extérieure du champ de rotation cible γ :

$$d\gamma(v) = k_v\pi/2 - \alpha_v$$

Il n'y a rien d'autre à adapter dans la méthode décrite dans le chapitre 4 pour obtenir un champ de repère correspondant aux valences de bord en entrée k_v sur le maillage triangulaire de la géométrie initiale T_0 . Nous utilisons ce champ de repère pour calculer pour chaque pair de triangles adjacents t, t' du maillage T_0 les correspondances $p_{tt'}$ (Sec 3.1.2).

8.2.3 Plusieurs géométries de champs de repère de mêmes correspondances

A partir des correspondances fixés $p_{tt'}$ correspondant aux valences de bord que nous désirons, nous pouvons désormais calculer une géométrie de champ de repère pour chaque maillage $(T_n)_{n \leq N}$. Pour ce faire nous utilisons pour chaque maillage $(T_n)_{n \leq N}$ la réoptimisation de champ de repère de la section 3.1.2, et nous obtenons pour tout $n \leq N$ la carte locale $\mu_{.,n} : \mathbb{R}^2 \mapsto \mathbb{R}^2$.

8.3 Quadcover adapté à plusieurs géométries

Nous avons dorénavant un champ de repère qui nous garantit que le maillage produit aura des valences suffisamment grandes sur le bord par rapport à la simulation de déformation réalisée. Il nous reste maintenant à prendre en compte les cartes locales $(\mu_{.,n})_{n \leq N}$ pour construire une carte préservant la grille u de l'état initial de l'objet, de sorte que les formes de ses éléments soient adaptées aux déformations à venir. Nous allons également présenter la méthode qui nous permet de faire en sorte que cette carte préservant la grille u soit valide afin d'en extraire un maillage quadrilatère.

8.3.1 Transfert des géométries de champ de repère

Pour pouvoir utiliser Quadcover à partir d'un champ de repère comme dans la section 3.1.3, nous devons construire une unique carte μ définis sur les coins de triangles du maillage triangulaire initial T_0 . Plutôt que de définir μ par simplement la carte initiale $\mu_{.,0}$, nous allons définir la carte μ comme la moyenne des cartes $(\mu_{.,n})_{n \leq N}$, d'avantage adaptée aux déformations de la simulation qui a donné les triangulations $(T_n)_{n \leq N}$. Pour tout coin de triangle i de T_0 :

$$\mu_i = \frac{1}{N} \sum_{n \leq N} \mu_{i,n}$$

À partir de ces coordonnées sur les coins de triangle, et par interpolation linéaire dans chaque triangle de T_0 , on obtient une carte μ définie sur T_0 et construite à partir des champs de repères de chaque triangulation $(T_n)_{n \leq N}$.

8.3.2 Processus itératif pour rendre la carte valide

À partir des maillages triangulaires en entrée $(T_n)_{n \leq N}$ obtenus en alternant maillage et simulation (section 8.1.2), la section 8.2.2 nous permet de générer des correspondances de champ de repère $p_{tt'}$ (pour toute paire de triangles t, t'), et la section 8.3.1 donne une géométrie unique de champ de repère définie par la carte μ . Nous pouvons donc par la suite utiliser la section 3.1.3 pour calculer une carte préservant la grille.

Une carte préservant la grille garantit la construction d'un maillage quadrilatère valide seulement si la carte est sans retournement. Dans cette sous-section, nous présentons une méthode itérative permettant de résoudre des problèmes de retournement qui sont soit dû à des correspondances $p_{tt'}$ valides mais difficiles à respecter vis à vis de la géométrie du modèle, soit des correspondances $p_{tt'}$ invalides dont il est impossible de construire un maillage quadrilatère du modèle. Ces améliorations sont également applicables dans le cas d'une seule géométrie en entrée, et nous ont permis de générer les cartes sans couture valide du chapitre 7, pour que notre algorithme de quantification produise des cartes préservant la grille valides.

Calcul itératif d'une carte sans couture pour en enlever les retournements Après avoir calculé une carte sans couture u avec la méthode 3.1.3, on peut déterminer si la carte u est valide en vérifiant que le jacobien de la déformation de chaque triangle dans le domaine paramétrique est strictement positif. Pour chaque triangle t et ses coins $i, j, k \in \mathcal{C}(t)$, on vérifie la stricte positivité du jacobien :

$$\det(u_j - u_i, u_k - u_i) > 0 \quad (8.1)$$

Pour chaque triangle ayant un jacobien négatif, on incrémente de 1 le poids du triangle dans l'énergie d'optimisation de Quadcover, décrit dans l'équation 3.5, puis on réoptimise u avec cette nouvelle énergie d'optimisation. Lorsque le poids d'un triangle est augmenté, le jacobien négatif de déformation du triangle dans la carte u se rapproche du jacobien de déformation du triangle dans la carte μ qui est positif. L'objectif est de réaliser plusieurs itérations de réoptimisation jusqu'à obtenir une carte sans couture u sans aucun retournement, que l'on donne en entrée de l'algorithme de quantification du chapitre 7 pour obtenir une carte préservant la grille valide et le maillage quadrilatère associé. La section 8.4 présente le taux de réussite de cette méthode sur une base de données de modèles CAO. En pratique, cette méthode échoue lorsque les correspondances du champ de repère sont invalides. Dans les cas d'échecs, nous essayons donc de faire des modifications locales des correspondances déduites du champ de repère, pour essayer de récupérer des correspondances valides.

Rendre les correspondances de champ de repère p valides Pour deux triangles voisins t, t' de sommets communs v et w , incrémenter $p_{tt'}$ et décrémenter $p_{t't}$ a pour effet d'incrémenter la valence de v et décrémenter la valence de w dans le maillage quadrilatère de sortie (par la méthode Quadcover de la section 3.1). De cette manière, on peut ajouter un sommet singulier de valence 3 et un sommet singulier de valence 5 sur des sommets qui n'étaient pas des singularités du champ de repère initial. Cet ajout de dipôle de singularité s'avère nécessaire lorsque le graphe de singularité du champ de repère en entrée ne permet pas de construire un maillage quadrilatère (Fig 8.2). Bien qu'il soit préférable de sélectionner les sommets v, w de façon à maximiser la qualité du maillage quadrilatère de sortie, ce problème est NP-complet, et il n'y a pas vraiment d'heuristique qui permet de trouver de bonne solution. C'est pourquoi nous sélectionnons des triangles voisins t et t' aléatoirement dans l'intérieur du modèle, pour incrémenter $p_{tt'}$ et décrémenter $p_{t't}$. Ensuite nous recalculons une géométrie de champ de repère correspondant à

ces nouvelles correspondances fixées (section 3.1.2), et une nouvelle carte sans couture u . La section 8.4 montre qu'il n'est en général pas nécessaire d'ajouter plus de trois dipôles pour résoudre les problèmes de non-intégrabilité du champ de repère.

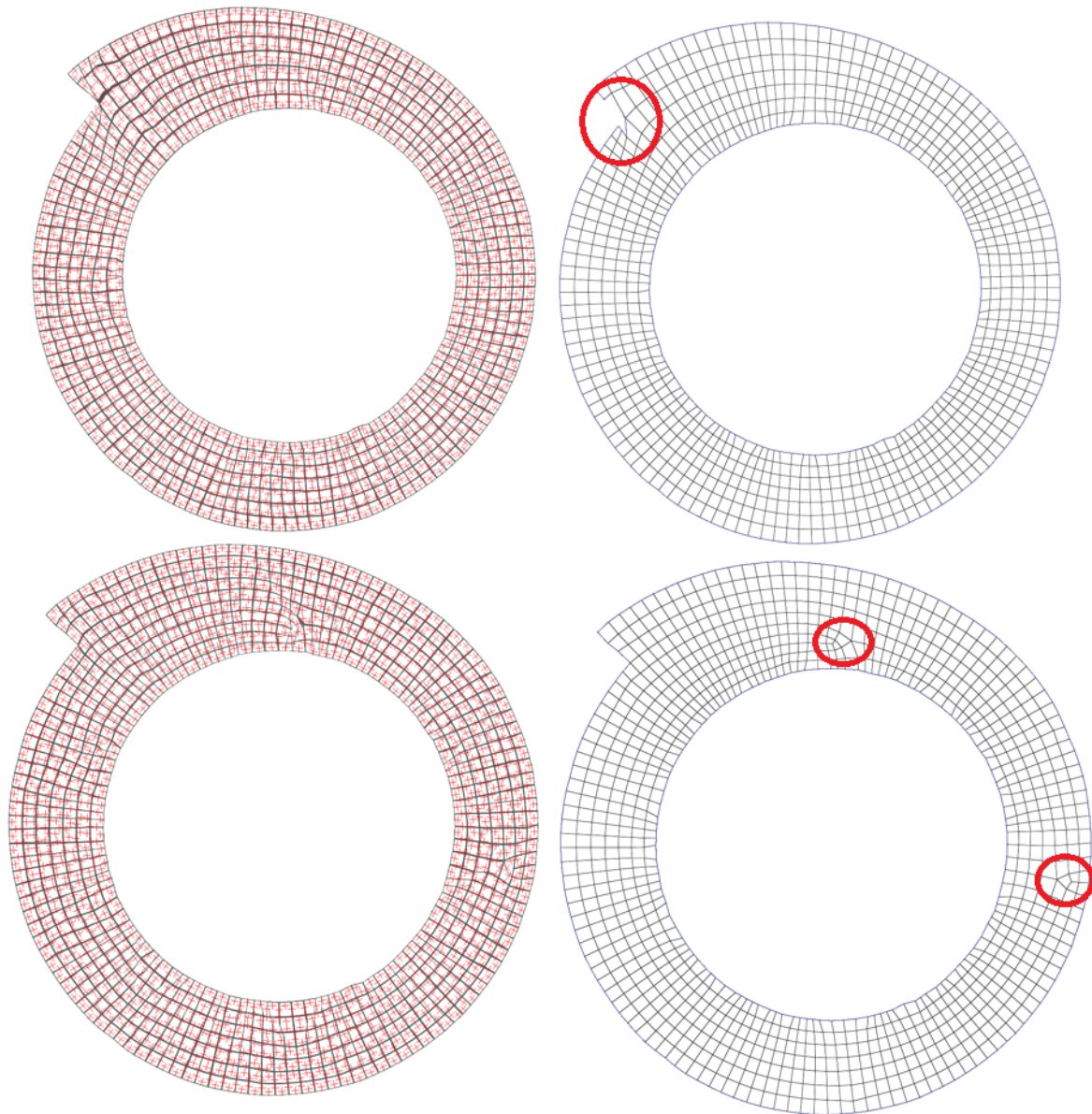


FIGURE 8.2 – Il arrive qu'un champ de repère parfaitement lissé ne puisse pas être intégré, en raison de conditions d'alignement des limites qui ne peuvent pas être satisfaites pendant le processus d'intégration. Le problème peut être résolu en effectuant des modifications aléatoires de correspondance dans le champ de repère. Ici, deux arêtes aléatoires ont reçu une modification de correspondance, ce qui a entraîné la création de deux dipôles de sommets de valence 3 et de valence 5.

8.4 Résultats

Dans cette section, nous présentons une évaluation de la méthode présentée dans ce chapitre. Notre implémentation consiste en un code de moins de 2000 lignes de code C++, n'utilisant rien d'autre que la bibliothèque standard. Nous prenons en entrée un (ou plusieurs) maillage triangulaire avec les arêtes caractéristiques qu'il faut préserver, et nous retournons un maillage quadrilatère correspondant. Une version open-source compilée en webassembly est accessible à l'adresse <https://ddesobry.github.io/quadmasher.html>. Une autre version, réalisée pour plusieurs géométries en entrée dans le cas de simulation numérique est du code propriétaire appartenant à TotalEnergies.

8.4.1 Réussite de la méthode sur les modèles CAO

Nous avons testé notre implémentation open-source sur la base de donnée [LEDOUX 2019] composé de 114 modèles.

1. 38/114 modèles ont des coins à angle faible, nécessitant l'utilisation de la contribution du chapitre 5.
2. 102/114 modèles ont des retournements dans la carte sans couture générés par la méthode classique (section 3.1.3), nécessitant plusieurs itérations de calcul de carte par la contribution de la section 8.3.2.
3. 4/114 modèles ont des correspondances de champ de repère $p_{tt'}$ non valides, nécessitant des modifications locales aléatoires par la méthode de la section 8.3.2
4. Notre implémentation open-source génère des maillages quadrilatères valides pour tous les modèles.

Pour comparaison avec une autre méthode qui permet de générer des cartes sans couture sans retournement [GARANZHA et al. 2021] par une optimisation non-convexe, la méthode [GARANZHA et al. 2021] n'apporte pas de solution pour les 4 modèles (B17, B18, B49, S22) qui ont des correspondances de champ de repère non-valides. De plus, bien que la méthode [GARANZHA et al. 2021] résolve la plupart des problèmes de retournement, elle rencontre des problèmes de 2-coverings qui rend difficile l'extraction d'un maillage quadrilatère valide pour 8 modèles qui ont des correspondances de champ de repère valides (S1, S4, S9, M3, M4, M5, M7, M8).

8.4.2 Réussite de la méthode pour les simulations de déformation

Nous avons testé notre processus de génération de maillage quadrilatère adapté à une simulation de grandes déformations en alternant de manière automatique maillage quadrilatère avec lancement de simulation sur le logiciel Numea (Sec 8.1). Parmi les modèles de joints 2D que Hutchinson nous a donné, nous avons pu réussir à compléter la simulation de la figure 8.3 avec les méthodes de l'état de l'art appliquées sur le modèle triangulaire de l'état initial du modèle. Comme illustré dans la figure 8.4, nous sommes capables de préserver les frontières entre matériaux en y introduisant des arêtes caractéristiques, qui agissent comme des bords avec laquelle la carte préservant la grille u est contrainte à être parfaitement alignée (Sec 3.1.3).

Nous avons ensuite essayé de lancer la même simulation sur des formes non conventionnelles pour tester l'utilisabilité de nos maillages quadrilatères sur d'autres modèles (Fig 8.5). Nous avons pu remarquer que notre façon de gérer les coins à petit angle dans le chapitre 5 nous permet de lancer des simulations de déformations sur des coins à petit angle (Fig 8.5, bas).

Pour les simulations de déformations induisant une grande distorsion des éléments du bords, les valences de bord des maillages quadrilatères produits à partir de l'état initial de l'objet

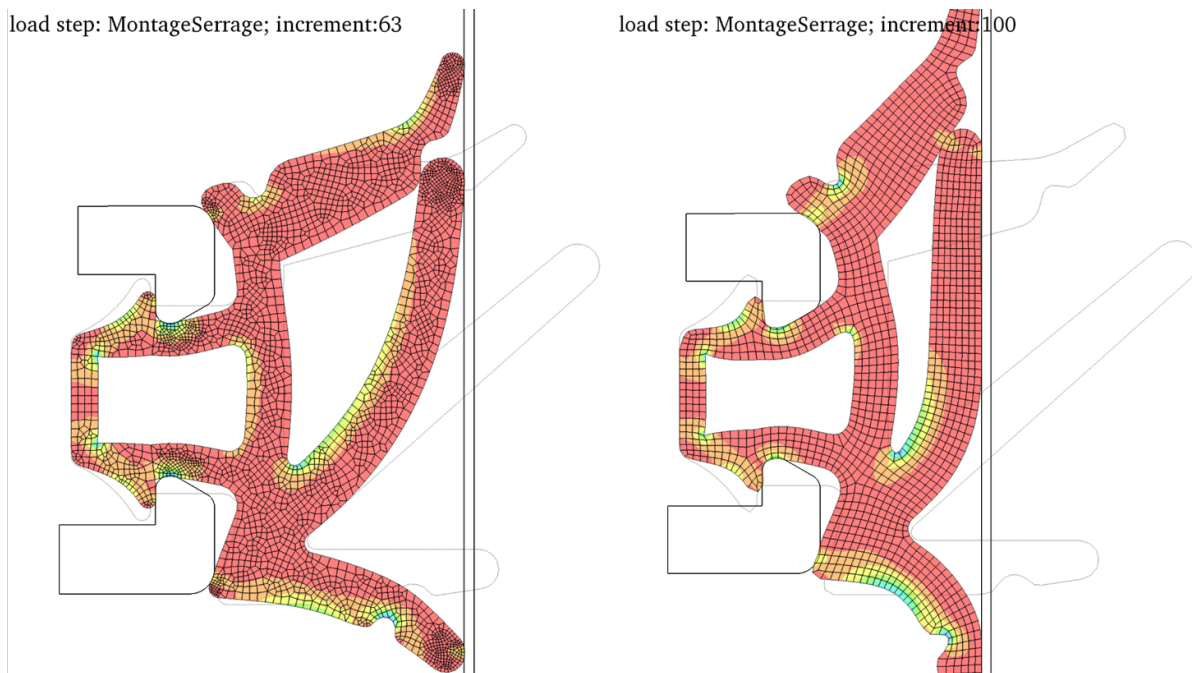


FIGURE 8.3 – Comparaison entre deux méthodes de génération de maillage automatique pour la simulation de déformation sur le logiciel [NUMEA s. d.]. La simulation avec un maillage non-structurée (gauche) diverge après l'itération 63, tandis que la simulation avec un maillage à valence semi-structurée (droite) parvient à atteindre l'itération 100 de la simulation.

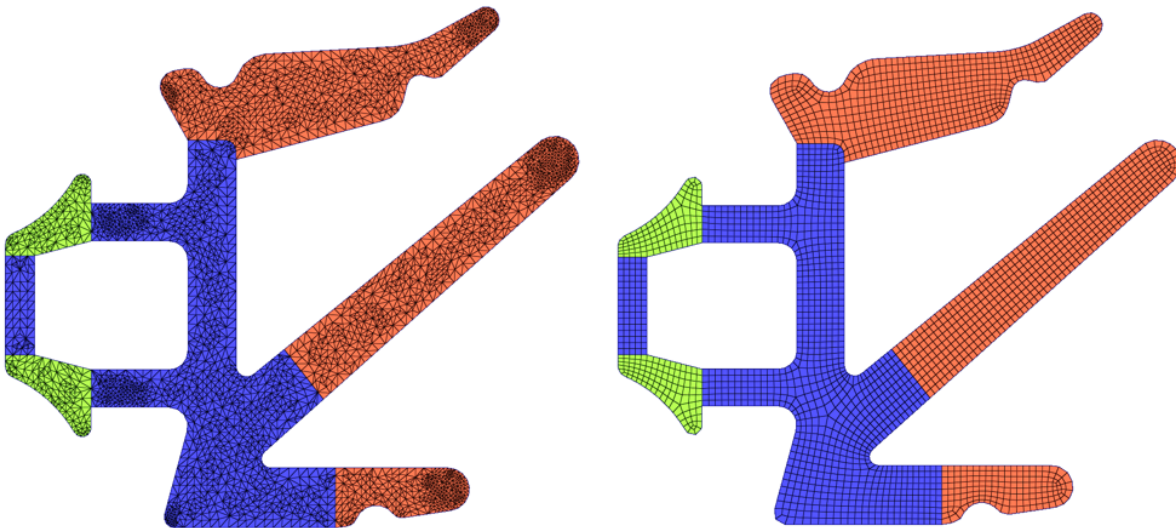


FIGURE 8.4 – La méthode présentée dans cet article génère des maillages quadrilatères qui s'alignent naturellement avec les frontières des matériaux en les traitant comme des arêtes caractéristiques.

conduisent à des retournements inévitables (Fig 8.6). Cependant, prendre en compte les déformations d'une simulation qui a précédemment échoué (Sec 8.1) permet de converger vers une solution (Fig 8.7) où les valences deviennent adaptées à la simulation réalisée en utilisant la contribution

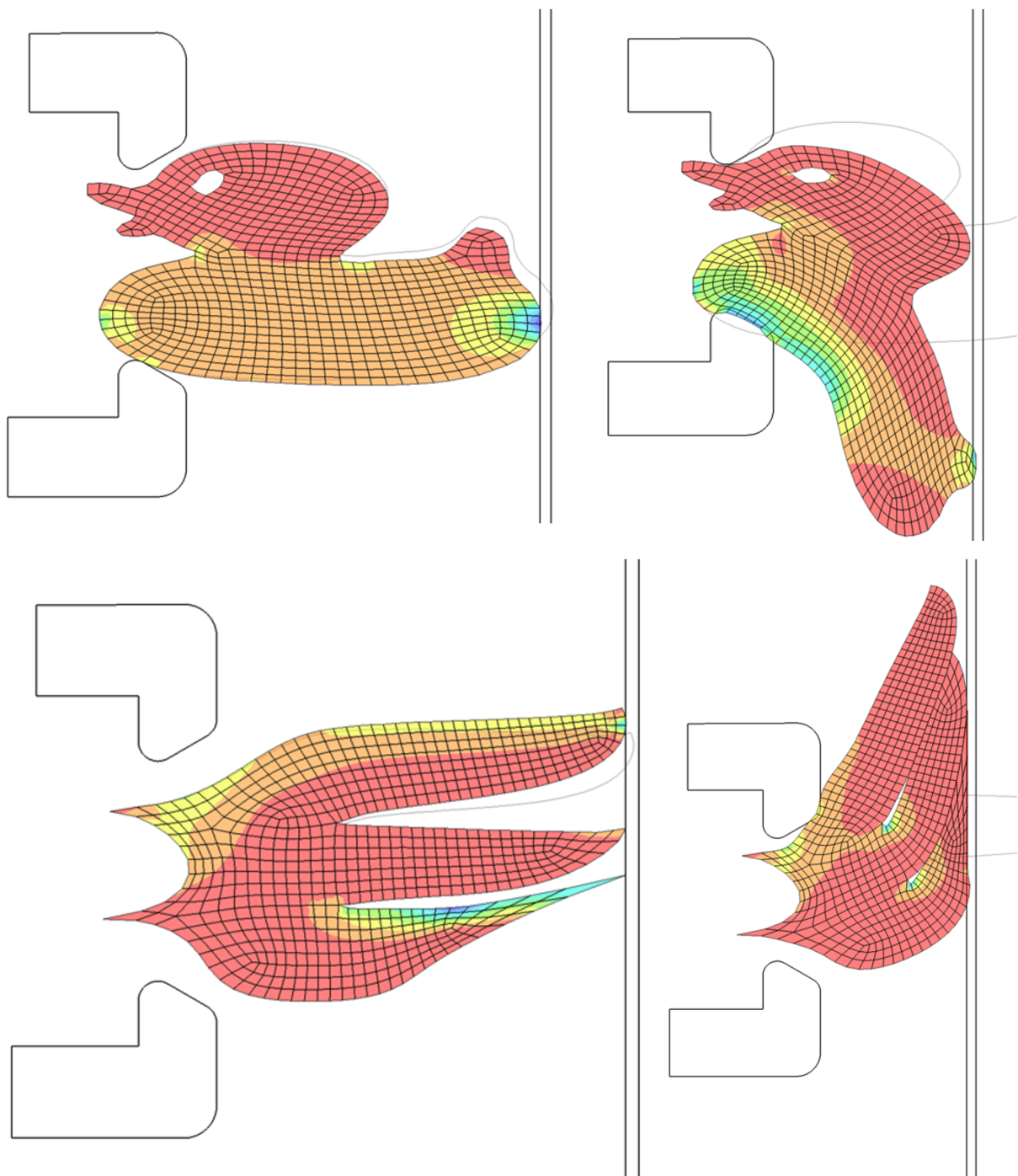


FIGURE 8.5 – Notre méthode de génération de maillage quadrilatère permet la déformation de formes non conventionnelles (canard 2D en haut, modèles 2D avec coins très aigus en bas) avec le solveur industriel Numea. Dans ces deux cas, nous avons fixé le modèle sur la gauche, et pour le modèle du bas, nous avons utilisé une barre glissante avec une très faible friction.

de la section 8.2.

Cette correction de valence sur le bord par la méthode décrite dans la section 8.2 nous a permis de construire de manière automatique des maillages quadrilatères adaptés à des simulations de

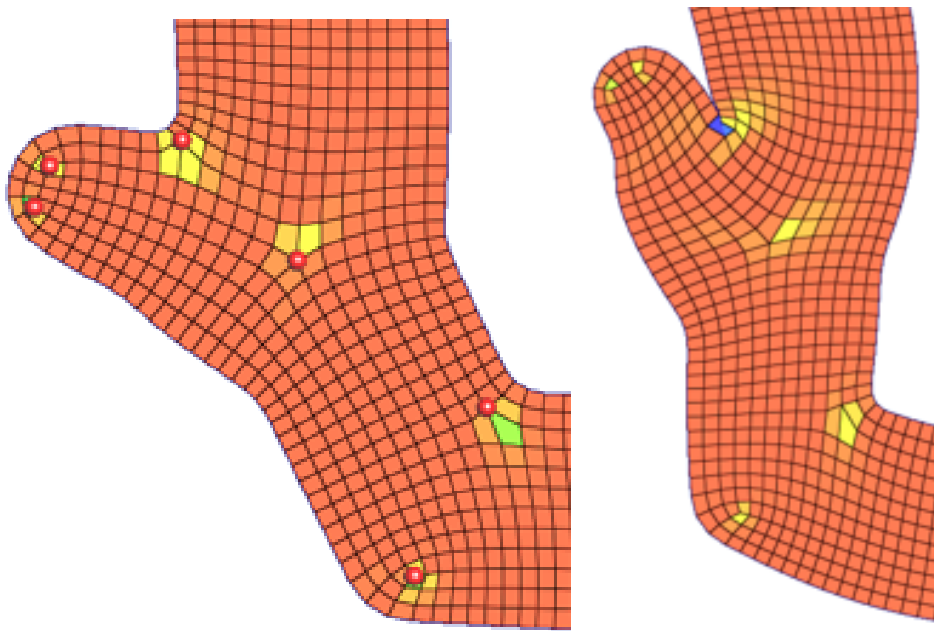


FIGURE 8.6 – *Les positions de singularités optimales pour une géométrie initiale peuvent conduire à des quadrilatères de qualité médiocre après une déformation.*

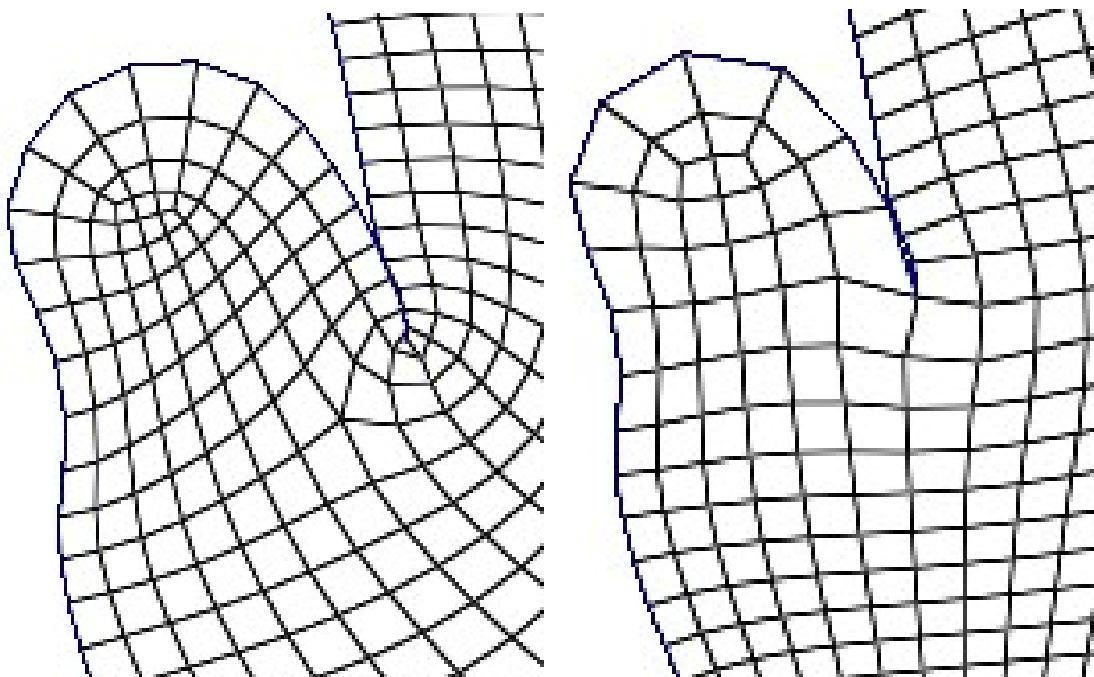


FIGURE 8.7 – *En utilisant toutes les géométries de l'itération précédente de la simulation échouée, 4 quadrilatères adjacents sont automatiquement placés au point de bord où la déformation est la plus prononcée.*

jointes sur lesquels sont appliqués de grosses déformations de bords (Fig 8.8, Fig 8.9).

Enfin, pour une simulations de grandes déformations impliquant un écrasement de joint (Fig 8.11), utiliser l'état initial du modèle pour générer un maillage quadrilatère n'a pas suffi

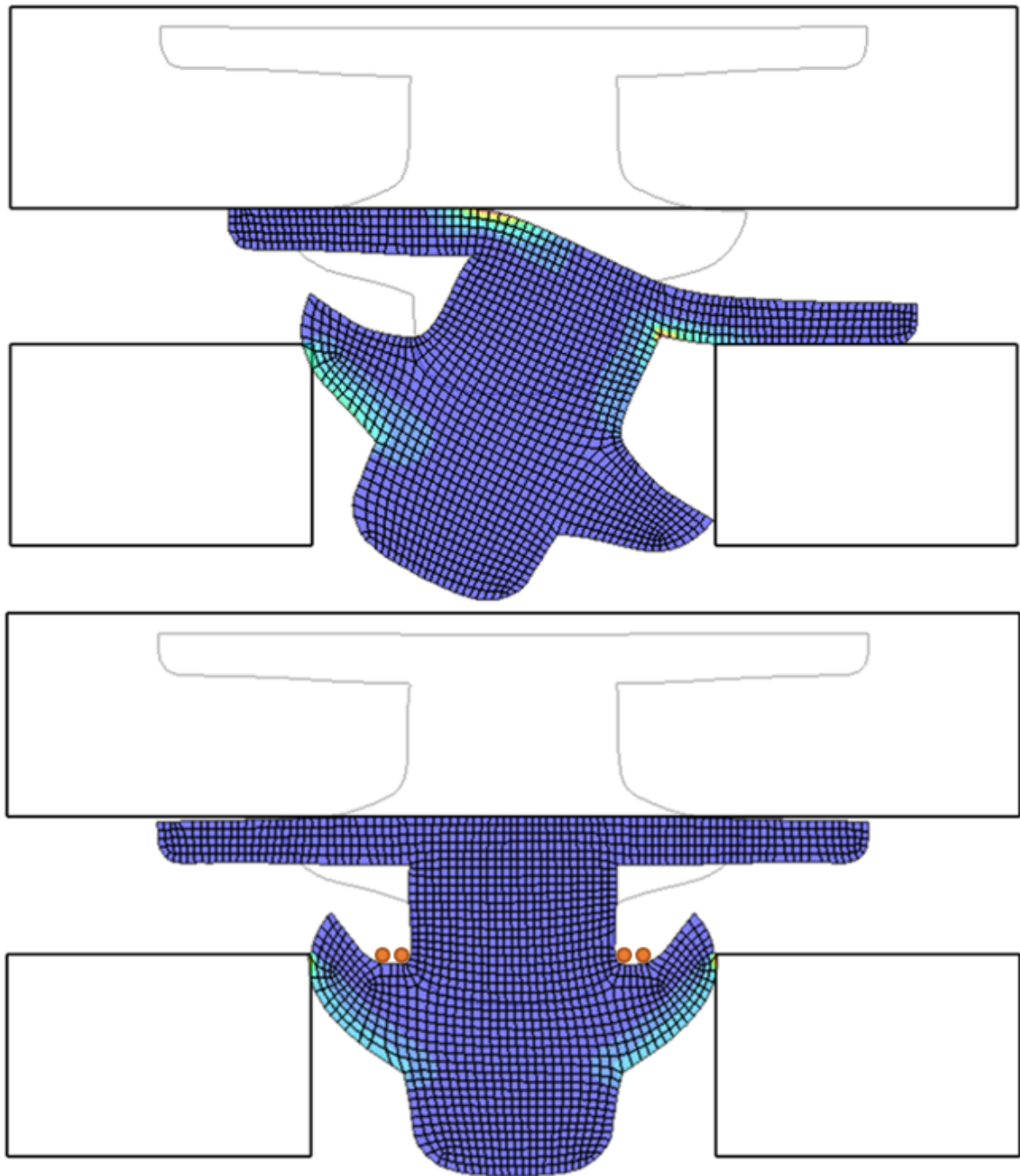


FIGURE 8.8 – *Lorsqu’une même simulation de déformation est exécutée sur deux maillages quadrilatères différents, les résultats peuvent être très différents. En raison de problèmes de convergence, le joint dans le premier cas a percé son support. Dans le deuxième cas, le joint a fléchi pour s’adapter normalement à son support grâce à l’ajout automatique de sommets de valence de bord 3 sur les points rouges.*

pour atteindre l’itération finale de la simulation. Cependant, en calculant des champs de repères sur toutes les géométries de la déformation qui a échoué et en moyennant les cartes locales obtenues sur chaque géométrie (Sec 8.3.1), nous avons pu automatiquement converger vers un modèle

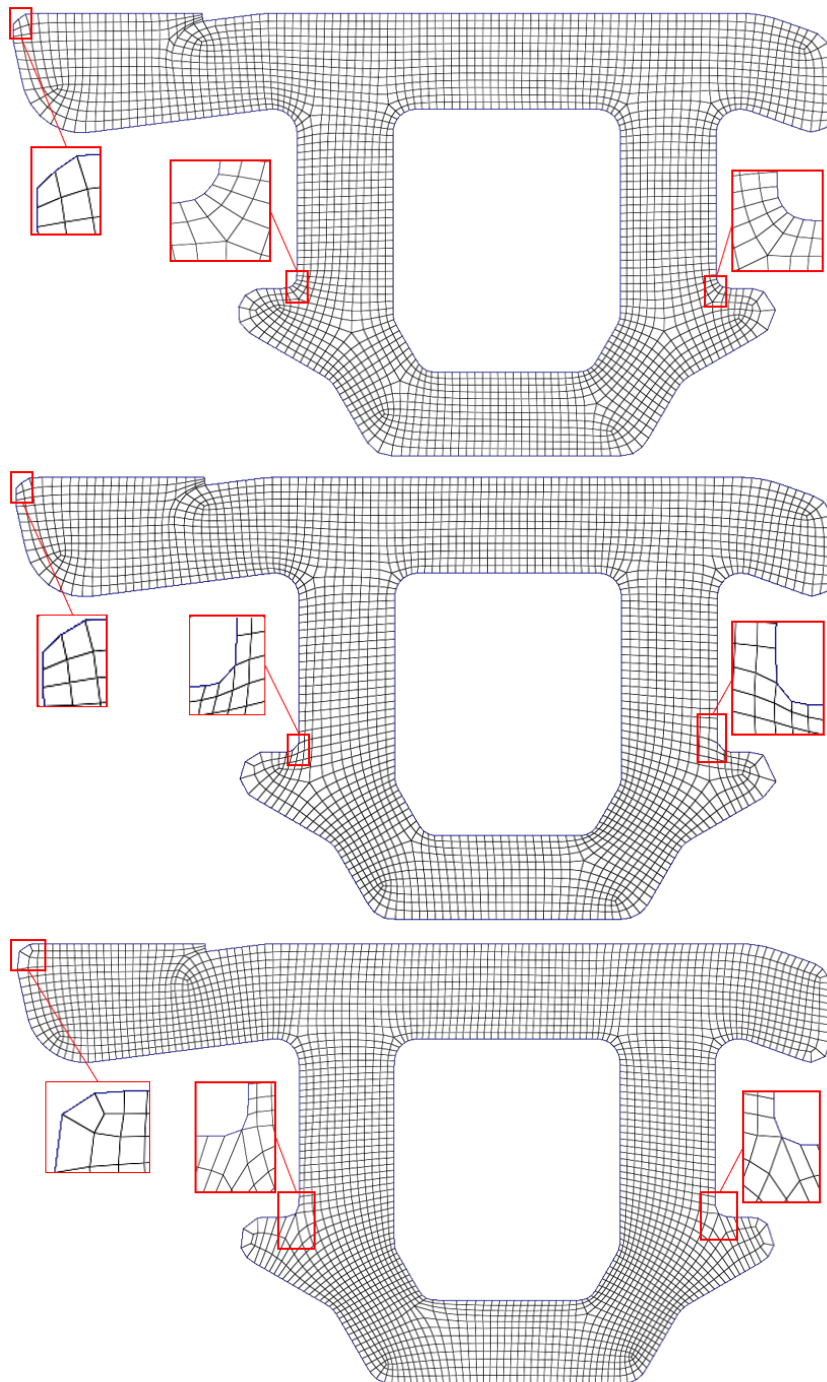


FIGURE 8.9 – *En haut : un maillage quadrilatère généré à partir du domaine 2D initial. Au milieu : un maillage quadrilatère dérivé de toutes les géométries de la première simulation échouée, En bas : un maillage quadrilatère dérivé de toutes les géométries de la deuxième simulation échouée. Le maillage quadrilatère du bas a réussi la simulation grâce à une géométrie globale améliorée et des valences de bord plus adaptées que les maillages précédents (carrés rouges).*

quadrilatère qui a réussi à atteindre l'itération finale de la simulation. Après quatre itérations de calcul de maillage quadrilatère (< 1 min) et de calcul de simulation de déformation non-

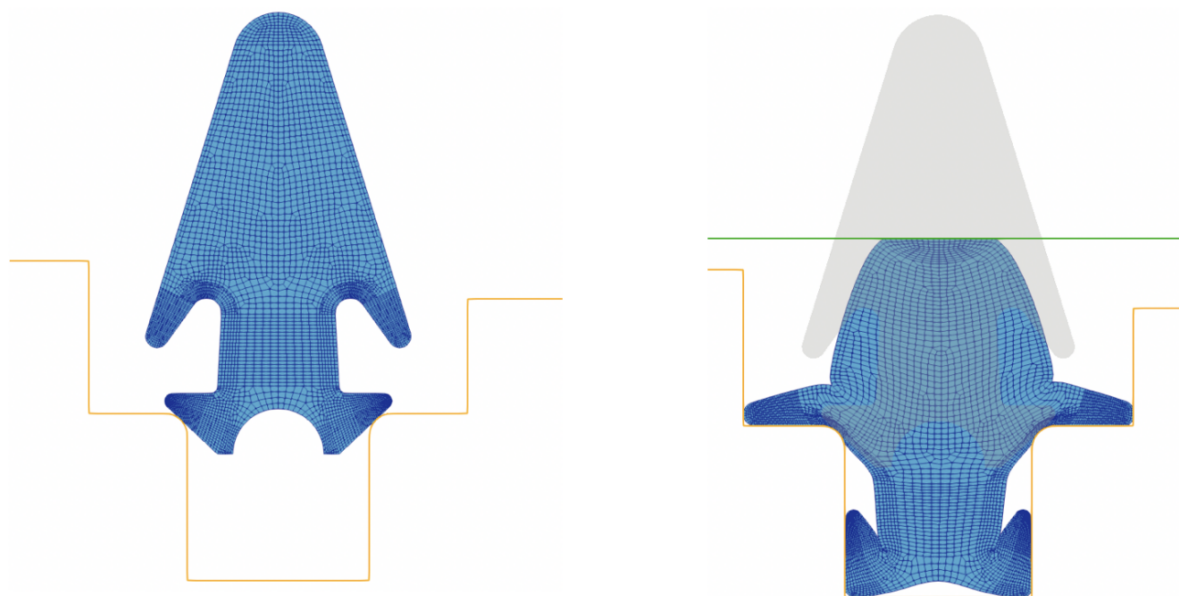


FIGURE 8.10 – À gauche : un maillage quadrilatère, l’initialisation de la simulation. À droite : la déformation du maillage au cours de la simulation risque de retourner des mailles ce qui bloque la simulation [Images Hutchinson].

linéaire (≈ 30 minutes), notre méthode automatique a construit un maillage qui a permis la réalisation complète de la simulation (durée totale : 2h). En comparaison, la simulation Fig 8.11 avait nécessité la construction d’un maillage quadrilatère par un ingénieur pendant une durée de trois semaines pour pouvoir être mené à bout (Figure 8.10).

Dans ce chapitre, nous nous sommes concentré sur l’élaboration d’un processus qui alterne simulation de déformation et construction d’un maillage quadrilatère adapté à la simulation. Lorsqu’une simulation de déformation échoue à cause d’un élément quadrilatère devenu invalide, les géométries de déformation de la simulation sont utilisées pour calculer un nouveau maillage quadrilatère de l’état initial ne présentant pas le même problème d’invalidité après déformation. Lorsqu’une simulation de déformation montre que les valences sur le bord sont insuffisantes, nous la modifions en adaptant la méthode du chapitre 5. En 3D, un processus similaire alternant simulation et maillage semblerait envisageable (en utilisant par exemple le chapitre 6 pour contrôler les valences de bord lorsqu’elles s’avèrent inadaptés à la simulation de déformation), mais la méthode Cubecover (Chap 3) est encore loin d’avoir atteint un niveau de maturité suffisant pour produire des maillages hexaédriques valides sur des modèles CAO industriels. Nous discuterons en conclusion des progressions restantes à réaliser dans des travaux futurs pour atteindre l’objectif de générer de manière automatique des maillages hexaédriques adaptés à des simulations de grandes déformations.

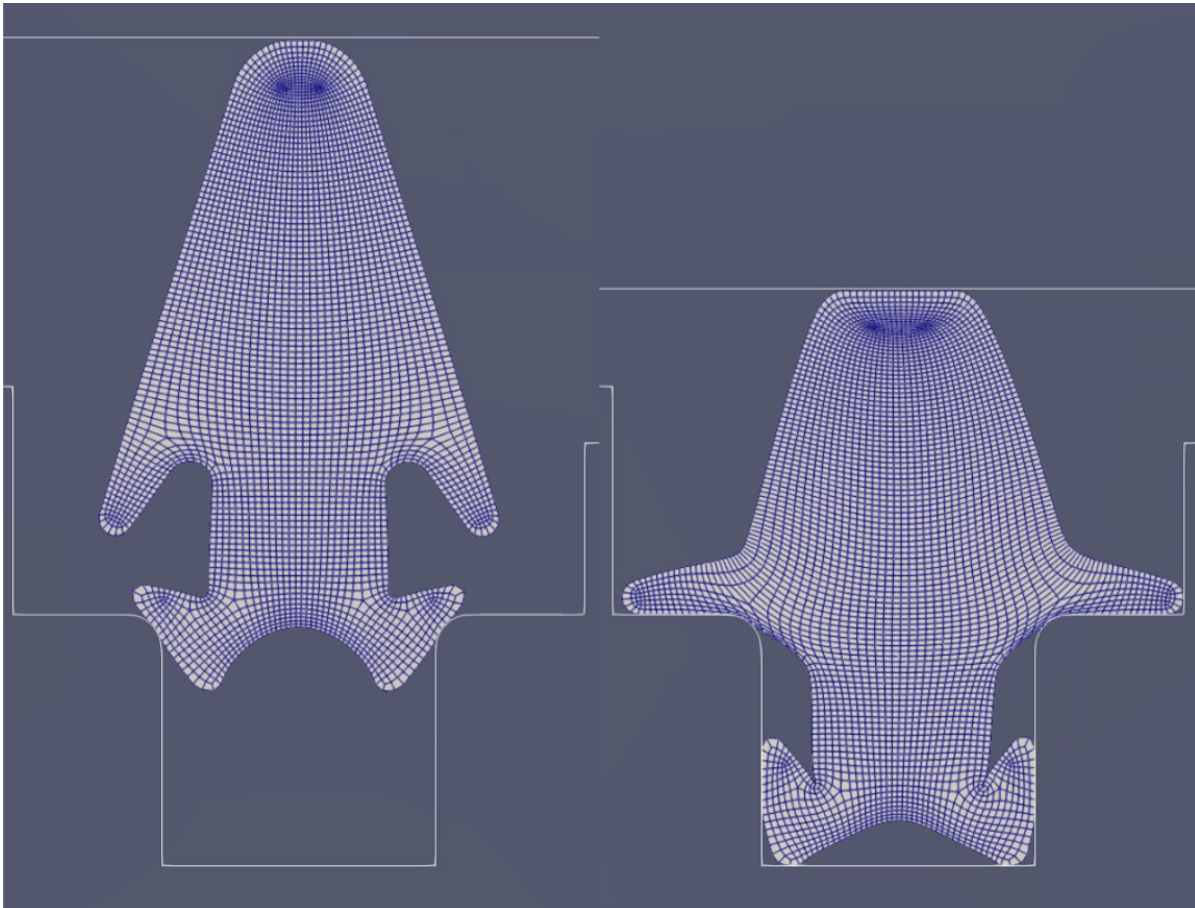


FIGURE 8.11 – *Notre méthode d’alternance automatique entre calcul de maillage et lancement de simulation nous a permis d’obtenir en 2h un maillage quadrilatère résistant à une simulation de compression qui avait nécessité une construction manuel de maillage par un ingénieur pendant une durée de 3 semaines.*

Conclusion

Au cours de cette thèse, nous avons développé des méthodes automatiques de génération de maillage quadrilatère et hexaédrique adaptées pour l'application de simulation de grandes déformations.

Dans la première partie de cette thèse, nous avons présenté l'état de l'art de la génération de maillage, en mettant l'accent sur les méthodes basées sur des champs de repères, telles que Quadcover et Cubecover. Ces méthodes sont prometteuses pour les simulations de grandes déformations car elles génèrent des maillages réguliers dont les arêtes sont alignées avec les forces de déformations.

Dans la deuxième partie, nous avons développé trois approches différentes pour augmenter le contrôle que donne les méthodes de champ de repère sur les maillages hexaédriques produits. Nous avons étudié l'utilisation de champs de repères non-orthogonaux, une redéfinition de lissage d'un champ de repère adapté à des modèles CAO, et le contrôle des valences de bord pour les maillages hexaédriques.

Dans la troisième partie, nous nous sommes focalisés sur la construction de maillage 2D quadrilatères. Nous avons proposé un algorithme de quantification ainsi qu'une méthode itérative d'intégration de champ de repère, permettant d'avoir une méthode complète basée sur Quadcover qui produit des modèles quadrilatères de qualité sur des bases de données CAO difficiles. Également, nous avons utilisé les champs de repère de la deuxième partie pour générer des maillages quadrilatères adaptés à plusieurs géométries d'entrée, dont les valences sur le bord permettent la validité des éléments pendant une simulation de déformation.

Cette thèse a contribué à l'amélioration des méthodes basées sur les champs de repère. Les champs de repère sont utilisés pour construire automatiquement des graphes de singularités, et nous avons développé de nouvelles méthodes de calcul de champ de repère pour faire en sorte d'éviter certaines configurations de graphe de singularité non-valide. Cependant, une caractérisation complète de validité de graphe de singularité manque encore pour espérer que les méthodes de maillage hexaédrique basée sur Cubecover soit utilisable pour des modèles volumiques difficiles. Heureusement, dans le cas 2D, des cas de graphe de singularité invalides sont beaucoup plus rares, et nous en avons donné une méthode de résolution simple pour les rares cas où cela arrive dans la section 8.3.2. Nous avons grâce à cela pu produire une méthode fiable qui génère des maillages quadrilatères adaptés à une simulation de déformation donnée. Nous espérons que dans le futur, la méthode Cubecover aura le même taux de réussite que la méthode Quadcover d'aujourd'hui, pour construire automatiquement des maillages hexaédriques adaptés à une simulation de déformation par extension de ce que nous avons présenté en 2D.

Perspectives

Champ de repère intégrable Dans le chapitre 4, nous avons introduit une représentation de champs de direction commune aux surfaces et aux volumes, permettant des contraintes sur une, deux ou trois directions indépendamment avec un contrôle intuitif de l'orthogonalité des repères. Bien que notre optimisation soit non-convexe et plus difficile que la représentation standard de champ de repère, elle permet des contraintes plus flexibles qui conviennent mieux aux problèmes de paramétrisation globale. Notamment, il est possible d'adapter la méthode pour étudier les champs de direction *intégrables*, c'est-à-dire les champs dont les directions sont des gradients de fonctions scalaires. Cela est fondamental pour garantir l'existence d'une paramétrisation globale *bijective*. Des premiers travaux ont été menés en 2D pour générer un champ non-orthogonal 2D intégrable [DIAMANTI et al. 2015] à partir d'une représentation de champ non-orthogonal 2D [DIAMANTI et al. 2014]. Nous pouvons espérer que des avancées similaires d'intégrabilité peuvent être réalisés en 3D sur nos champs non-orthogonaux 3D du chapitre 4.

Caractérisation d'un réseau de singularité valide sur le bord Dans le chapitre 6, nous présentons une méthode pour calculer un champ de repère avec des singularités de bord prescrites en modifiant les contraintes d'alignement de bord. Ces champs peuvent ensuite être utilisés dans le pipeline standard de maillage hexaédrique. Actuellement, nous utilisons des valences prescrites sur le bord dans la description CAO de l'objet pour calculer nos contraintes d'alignement. Un travail futur pourrait être de déterminer automatiquement des valences sur le bord correspondant à un maillage hexaédrique, grâce à une caractérisation approfondie des graphes de singularité de bord. Un autre travail futur pourrait être de calculer un graphe de singularité intérieur valide à l'aide de champ de repère, en supposant que le graphe de singularité de bord, fixé par notre méthode, est valide.

Arêtes caractéristiques à l'étape de quantification 2D La plupart des modèles CAO sont accompagnés d'un réseau complexe d'arêtes caractéristiques. La contrainte d'alignement sur les arêtes caractéristiques devient difficile ou impossible à satisfaire, en particulier sans sacrifier la distorsion de la carte. Avec la méthode de quantification du chapitre 7, il est probablement possible d'utiliser une carte sans couture qui n'est pas alignée avec les arêtes caractéristiques, puis d'imposer l'alignement lors de l'étape de quantification. Pour ce faire, il faudrait exploiter le fait que les arêtes du maillage décimé que nous utilisons pour la quantification ne sont pas nécessairement alignées sur un axe de la carte (contrairement aux arêtes d'un maillage en T). De plus, pour les cas où l'alignement d'une carte préservant la grille avec des arêtes caractéristiques n'est pas possible pour un graphe de singularité donné, l'alignement avec les arêtes caractéristiques semble pouvoir être imposé *a posteriori* par une édition de maillage ajoutant des singularités (Fig 12). Le grand avantage est qu'ajouter les contraintes d'alignements aux arêtes caractéristiques pendant la quantification rend plus facile le calcul d'une carte sans couture valide avant quantification (voir Fig 13), et ne fait pas échouer la méthode si le graphe de singularité obtenu par un champ de repère ne correspond pas à un maillage quadrilatère aligné avec les arêtes caractéristiques.

Quantification 3D à partir d'un maillage décimé Le chapitre 7 démontre que les algorithmes de quantification n'ont pas nécessairement besoin de s'appuyer sur un maillage en T dans le cas 2D. La décimation d'un maillage tétraédrique 3D semble plus simple à réaliser que la construction de complexe de moto 3D décrit dans la section 3.2.3. Cependant, garder un jacobien

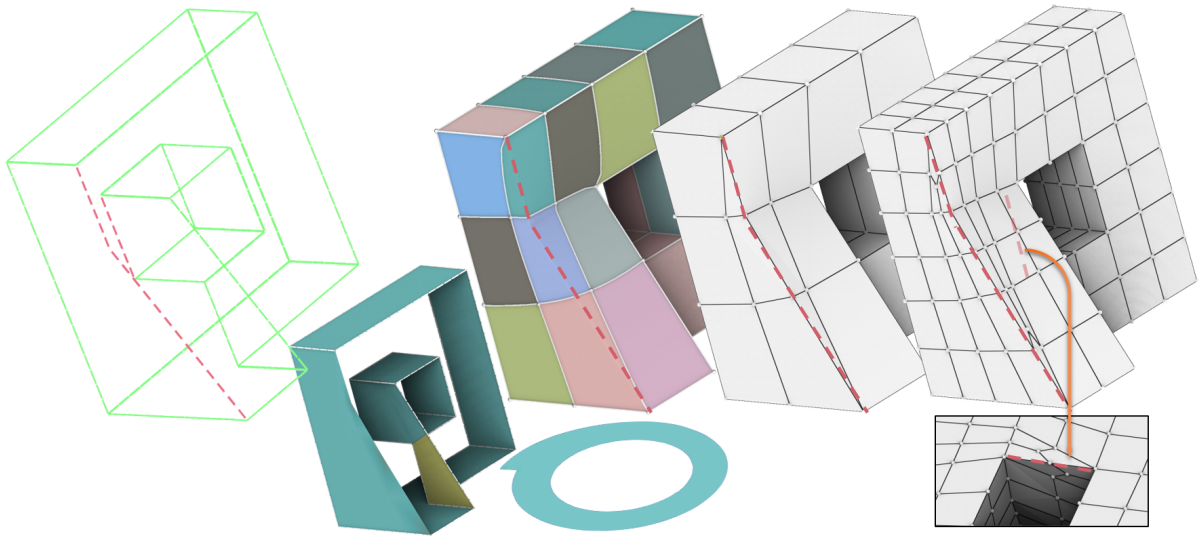


FIGURE 12 – *Ce modèle a des arêtes caractéristiques incompatibles ; deux sous-ensembles de faces bleues (en bas à gauche) sont topologiquement équivalents au modèle célèbre : annulus (en bas, Fig. 5 de [MYLES et al. 2014]). Nous pouvons découvrir des arêtes caractéristiques conflictuelles à l'étape de quantification (lignes de points rouges), et les supprimer (milieu – gauche). En imprimant les arêtes caractéristiques (milieu – droite) et en appliquant une subdivision médiane, nous obtenons un alignement avec les arêtes caractéristiques (droite).*

positif dans une carte est une contrainte qui ralentirait grandement l'optimisation en nombre entier comparé à optimiser des distances entières positives dans la version 3D d'un maillage en T. Il reste donc à trouver une méthode gloutonne performante pour optimiser les positions des singularités dans la carte préservant la grille sur le maillage décimé 3D, comme ce que nous avons fait en 2D et ce qu'a fait [CAMPEN et al. 2015] pour obtenir une quantification rapide. Une telle méthode pourrait accélérer de manière conséquente l'étape de quantification 3D.

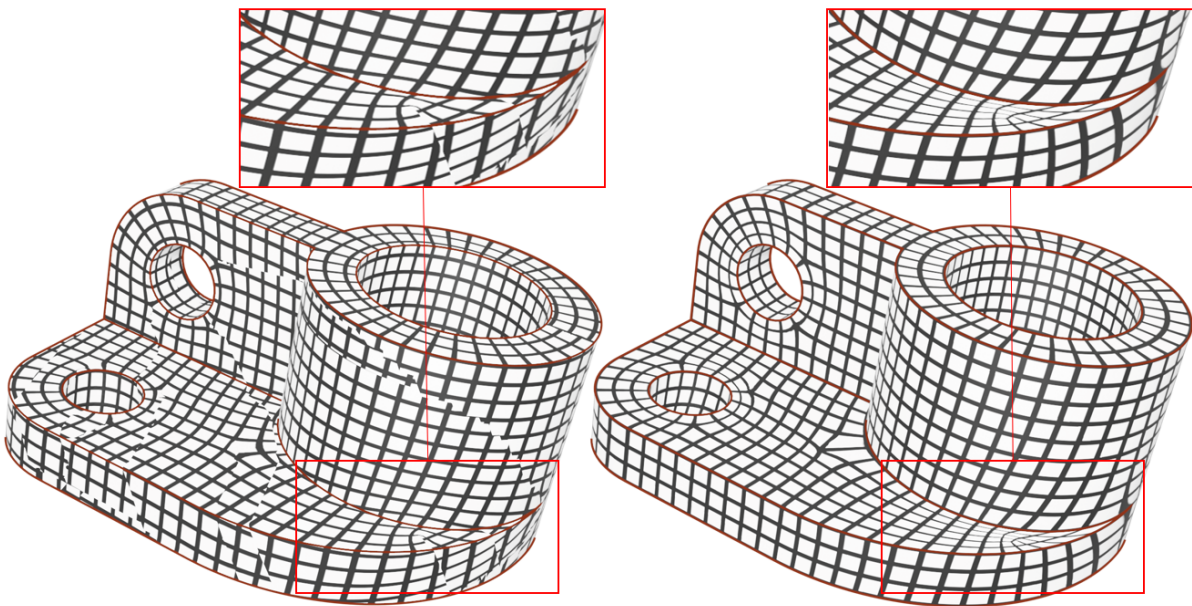


FIGURE 13 – Nous avons pu corriger toutes les contraintes d'alignements avec les arêtes caractéristiques pendant le processus de quantification d'une carte sans couture non alignée initialement (à droite). Le gros plan montre une partie où la satisfaction de la contrainte d'alignement aurait introduit beaucoup de distorsion dans la carte sans couture d'entrée.

Bibliographie

- Pierre ALLIEZ, Mark MEYER et Mathieu DESBRUN (juill. 2002). “Interactive Geometry Remeshing”. In : *ACM Trans. Graph.* 21.3. Place : New York, NY, USA Publisher : Association for Computing Machinery, p. 347-354. ISSN : 0730-0301. DOI : 10.1145/566654.566588. URL : <https://doi.org/10.1145/566654.566588>.
- Cx K BATCHELOR et George Keith BATCHELOR (1967). *An introduction to fluid dynamics*. Cambridge university press. ISBN : 0-521-66396-2.
- Pal G BERGAN et Ray W CLOUGH (1972). “Convergence criteria for iterative processes”. In : *AIAA journal* 10.8, p. 1107-1108. ISSN : 0001-1452.
- Ted D BLACKER et Ray J MEYERS (1993). “Seams and wedges in plastering : a 3-D hexahedral mesh generation algorithm”. In : *Engineering with computers* 9. Publisher : Springer, p. 83-93. ISSN : 0177-0667.
- David BOMMES, Marcel CAMPEN, Hans-Christian EBKE, Pierre ALLIEZ et Leif KOBBELT (2013a). “Integer-grid maps for reliable quad meshing”. In : *ACM Transactions on Graphics (TOG)* 32.4. Publisher : ACM New York, NY, USA, p. 1-12.
- David BOMMES, Bruno LÉVY, Nico PIETRONI, Enrico PUPPO, Claudio SILVA, Marco TARINI et Denis ZORIN (2013b). “Quad-mesh generation and processing : A survey”. In : *Computer Graphics Forum*. T. 32. Issue : 6. Wiley Online Library, p. 51-76.
- David BOMMES, Henrik ZIMMER et Leif KOBBELT (2009). “Mixed-integer quadrangulation”. In : *ACM Transactions On Graphics (TOG)* 28.3. Publisher : ACM New York, NY, USA, p. 1-10.
- (2012). “Practical mixed-integer optimization for geometry processing”. In : *Proceedings of the 7th international conference on Curves and Surfaces*. event-place : Avignon, France. Berlin, Heidelberg : Springer-Verlag, p. 193-206.
- Javier BONET et Richard D WOOD (1997). *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press. ISBN : 0-521-57272-X.
- Matteo BRACCI, Marco TARINI, Nico PIETRONI, Marco LIVESU et Paolo CIGNONI (2019). “Hexa-Lab. net : An online viewer for hexahedral meshes”. In : *Computer-Aided Design* 110. Publisher : Elsevier, p. 24-36.
- Hendrik BRÜCKLER, David BOMMES et Marcel CAMPEN (2022). “Volume parametrization quantization for hexahedral meshing”. In : *ACM Transactions on Graphics (TOG)* 41.4. Publisher : ACM New York, NY, USA, p. 1-19. ISSN : 0730-0301.
- Hendrik BRÜCKLER, Ojaswi GUPTA, Manish MANDAD et Marcel CAMPEN (2021). “The 3D Motorcycle Complex for Structured Volume Decomposition”. In : *CoRR* abs/2112.05793. arXiv : 2112.05793. URL : <https://arxiv.org/abs/2112.05793>.
- Marcel CAMPEN, David BOMMES et Leif KOBBELT (2015). “Quantized global parametrization”. In : *Acm Transactions On Graphics (TOG)* 34.6. Publisher : ACM New York, NY, USA, p. 1-12.
- Steven C CHAPRA et Raymond P CANALE (2011). *Numerical methods for engineers*. T. 1221. Mcgraw-hill New York.

- A CHEMIN, F HENROTTE, JF REMACLE et J VAN SCHAFTINGEN (2018). “Representing three-dimensional cross fields using 4th order tensors, IMR2018 : 27th International Meshing Roundtable (X. Roca and A. Loseille, eds.)” In : *Lecture Notes in Computational Science and Engineering* 127, p. 89-108.
- Keenan CRANE, Mathieu DESBRUN et Peter SCHRODER (2010). “Trivial connections on discrete surfaces”. In : *Computer Graphics Forum*. T. 29. Issue : 5. Wiley Online Library, p. 1525-1533.
- Olga DIAMANTI, Amir VAXMAN, Daniele PANOZZO et Olga SORKINE-HORNUNG (2014). “Designing N-PolyVector fields with complex polynomials”. In : *Computer Graphics Forum*. T. 33. Issue : 5. Wiley Online Library, p. 1-11.
- (2015). “Integrable polyvector fields”. In : *ACM Transactions on Graphics (TOG)* 34.4. Publisher : ACM New York, NY, USA, p. 1-12.
- Hans-Christian EBKE, David BOMMES, Marcel CAMPEN et Leif KOBELT (2013). “QEx : Robust quad mesh extraction”. In : *ACM Transactions on Graphics (TOG)* 32.6. Publisher : ACM New York, NY, USA, p. 1-10.
- Xianzhong FANG, Hujun BAO, Yiyong TONG, Mathieu DESBRUN et Jin HUANG (juill. 2018). “Quadrangulation through Morse-Parameterization Hybridization”. In : *ACM Trans. Graph.* 37.4. Place : New York, NY, USA Publisher : Association for Computing Machinery. ISSN : 0730-0301. DOI : 10.1145/3197517.3201354. URL : <https://doi.org/10.1145/3197517.3201354>.
- Vladimir GARANZHA, Igor KAPORIN, Liudmila KUDRYAVTSEVA, François PROTAIS, Nicolas RAY et Dmitry SOKOLOV (juill. 2021). “Foldover-Free Maps in 50 Lines of Code”. In : *ACM Trans. Graph.* 40.4. Place : New York, NY, USA Publisher : Association for Computing Machinery. ISSN : 0730-0301. DOI : 10.1145/3450626.3459847. URL : <https://doi.org/10.1145/3450626.3459847>.
- Fernando de GOES, Mathieu DESBRUN et Yiyong TONG (2016). “Vector field processing on triangle meshes”. In : *ACM SIGGRAPH 2016 Courses*, p. 1-49.
- Aaron HERTZMANN et Denis ZORIN (2000). “Illustrating smooth surfaces”. In : *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, p. 517-526.
- Kai HORMANN, Bruno LÉVY et Alla SHEFFER (2007). “Mesh Parameterization : Theory and Practice Video Files Associated with This Course Are Available from the Citation Page”. In : *ACM SIGGRAPH 2007 Courses*. SIGGRAPH '07. event-place : San Diego, California. New York, NY, USA : Association for Computing Machinery, 1-es. ISBN : 978-1-4503-1823-5. DOI : 10.1145/1281500.1281510. URL : <https://doi.org/10.1145/1281500.1281510>.
- Jin HUANG, Yiyong TONG, Hongyu WEI et Hujun BAO (2011). “Boundary aligned smooth 3D cross-frame field”. In : *ACM transactions on graphics (TOG)* 30.6. Publisher : ACM New York, NY, USA, p. 1-8.
- Tengfei JIANG, Xianzhong FANG, Jin HUANG, Hujun BAO, Yiyong TONG et Mathieu DESBRUN (2015). “Frame field generation through metric customization”. In : *ACM Transactions on Graphics (TOG)* 34.4. Publisher : ACM New York, NY, USA, p. 1-11.
- Patrick KNUPP (2007). *Remarks on Mesh Quality*. Rapp. tech. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).
- Felix KNÖPPEL, Keenan CRANE, Ulrich PINKALL et Peter SCHRODER (2013). “Globally optimal direction fields”. In : *ACM Transactions on Graphics (ToG)* 32.4. Publisher : ACM New York, NY, USA, p. 1-10.
- Sebastian KOCH, Albert MATVEEV, Zhongshi JIANG, Francis WILLIAMS, Alexey ARTEMOV, Evgeny BURNAEV, Marc ALEXA, Denis ZORIN et Daniele PANOZZO (juin 2019). “ABC : A Big CAD Model Dataset For Geometric Deep Learning”. In : *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Denis KOVACS, Ashish MYLES et Denis ZORIN (2011). “Anisotropic quadrangulation”. In : *Comput. Aided Geom. Des.* 28.8, p. 449-462. DOI : 10.1016/j.cagd.2011.06.003. URL : <https://doi.org/10.1016/j.cagd.2011.06.003>.
- Felix KÄLBERER, Matthias NIESER et Konrad POLTHIER (2007). “QuadCover - Surface Parameterization using Branched Coverings”. In : *Comput. Graph. Forum* 26.3, p. 375-384. DOI : 10.1111/j.1467-8659.2007.01060.x. URL : <https://doi.org/10.1111/j.1467-8659.2007.01060.x>.
- Mingwu LAI, Steven BENZLEY et David WHITE (2000). “Automated hexahedral mesh generation by generalized multiple source to multiple target sweeping”. In : *International Journal for Numerical Methods in Engineering* 49.1-2. Publisher : Wiley Online Library, p. 261-275. ISSN : 0029-5981.
- Franck LEDOUX (2019). *Mambo dataset*. URL : <https://gitlab.com/franck.ledoux/mambo>.
- Randall J LEVEQUE (2002). *Finite volume methods for hyperbolic problems*. T. 31. Cambridge university press. ISBN : 0-521-00924-3.
- Heng LIU, Paul ZHANG, Edward CHIEN, Justin SOLOMON et David BOMMES (juill. 2018). “Singularity-Constrained Octahedral Fields for Hexahedral Meshing”. In : 37.4. Place : New York, NY, USA Publisher : Association for Computing Machinery. ISSN : 0730-0301. DOI : 10.1145/3197517.3201344. URL : <https://doi.org/10.1145/3197517.3201344>.
- Marco LIVESU, Alessandro MUNTONI, Enrico PUPPO et Riccardo SCATENI (2016). “Skeleton-driven adaptive hexahedral meshing of tubular shapes”. In : t. 35. Issue : 7. Wiley Online Library, p. 237-246. ISBN : 0167-7055.
- Max LYON, David BOMMES et Leif KOBELT (2016). “HexEx : Robust hexahedral mesh extraction”. In : *ACM Transactions on Graphics (TOG)* 35.4. Publisher : ACM New York, NY, USA, p. 1-11.
- Max LYON, Marcel CAMPEN, David BOMMES et Leif KOBELT (juill. 2019). “Parametrization Quantization with Free Boundaries for Trimmed Quad Meshing”. In : *ACM Trans. Graph.* 38.4. Place : New York, NY, USA Publisher : Association for Computing Machinery. ISSN : 0730-0301. DOI : 10.1145/3306346.3323019. URL : <https://doi.org/10.1145/3306346.3323019>.
- Max LYON, Marcel CAMPEN et Leif KOBELT (2021a). “Quad Layouts via Constrained T-Mesh Quantization”. In : *Comput. Graph. Forum* 40.2, p. 305-314. DOI : 10.1111/cgf.142634. URL : <https://doi.org/10.1111/cgf.142634>.
- (2021b). “Simpler Quad Layouts using Relaxed Singularities”. In : *Computer Graphics Forum* 40.5.
- Célestin MAROT, Jeanne PELLERIN et Jean-François REMACLE (déc. 2018). “One machine, one minute, three billion tetrahedra”. en. In : *International Journal for Numerical Methods in Engineering*, nme.5987. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/nme.5987. URL : <https://onlinelibrary.wiley.com/doi/10.1002/nme.5987> (visité le 11/04/2023).
- Loïc MARECHAL (2009). “Advances in octree-based all-hexahedral mesh generation : handling sharp features”. In : Springer, p. 65-84. ISBN : 3-642-04318-6.
- Scott A MITCHELL (1996). “A characterization of the quadrilateral meshes of a surface which admit a compatible hexahedral mesh of the enclosed volume”. In : Springer, p. 465-476. ISBN : 3-540-60922-9.
- D. E. MULLER et F. P. PREPARATA (1978). “Finding the intersection of two convex polyhedra”. In : *Theoretical Computer Science* 7.2, p. 217-236. ISSN : 0304-3975. DOI : [https://doi.org/10.1016/0304-3975\(78\)90051-8](https://doi.org/10.1016/0304-3975(78)90051-8). URL : <https://www.sciencedirect.com/science/article/pii/0304397578900518>.

- Ashish MYLES, Nico PIETRONI, Denis KOVACS et Denis ZORIN (2010). "Feature-aligned T-meshes". In : *ACM Trans. Graph.* 29.4, 117 :1-117 :11. DOI : 10.1145/1778765.1778854. URL : <https://doi.org/10.1145/1778765.1778854>.
- Ashish MYLES, Nico PIETRONI et Denis ZORIN (juill. 2014). "Robust Field-Aligned Global Parametrization". In : *ACM Trans. Graph.* 33.4. Place : New York, NY, USA Publisher : Association for Computing Machinery. ISSN : 0730-0301. DOI : 10.1145/2601097.2601154. URL : <https://doi.org/10.1145/2601097.2601154>.
- Matthias NIESER, Ulrich REITEBUCH et Konrad POLTHIER (2011). "CubeCover - Parameterization of 3D Volumes". In : *Computer Graphics Forum*. Publisher : The Eurographics Association and Blackwell Publishing Ltd. ISSN : 1467-8659. DOI : 10.1111/j.1467-8659.2011.02014.x.
- NUMEA (s. d.).
- Jonathan PALACIOS et Eugene ZHANG (2007). "Rotational symmetry field design on surfaces". In : *ACM Transactions on Graphics (TOG)* 26.3. Publisher : ACM New York, NY, USA, 55-es.
- David PALMER, David BOMMES et Justin SOLOMON (2020). "Algebraic representations for volumetric frame fields". In : *ACM Transactions on Graphics (TOG)* 39.2. Publisher : ACM New York, NY, USA, p. 1-17.
- Daniele PANOZZO, Enrico PUPPO, Marco TARINI et Olga SORKINE-HORNUNG (2014). "Frame fields : Anisotropic and non-orthogonal cross fields". In : *ACM Transactions on Graphics (TOG)* 33.4. Publisher : ACM New York, NY, USA, p. 1-11.
- Nico PIETRONI, Marcel CAMPEN, Alla SHEFFER, Gianmarco CHERCHI, David BOMMES, Xifeng GAO, Riccardo SCATENI, Franck LEDOUX, Jean REMACLE et Marco LIVESU (2022). "Hex-mesh generation and processing : a survey". In : *ACM transactions on graphics* 42.2. Publisher : ACM New York, NY, p. 1-44.
- William H PRESS, Saul A TEUKOLSKY, William T VETTERLING et Brian P FLANNERY (2007). *Numerical recipes 3rd edition : The art of scientific computing*. Cambridge university press. ISBN : 0-521-88068-8.
- Mark A PRICE et Cecil G ARMSTRONG (1997). "Hexahedral mesh generation by medial surface subdivision : Part II. Solids with flat and concave edges". In : *International Journal for Numerical Methods in Engineering* 40.1. Publisher : Wiley Online Library, p. 111-136. ISSN : 0029-5981.
- Mark A PRICE, Cecil G ARMSTRONG et MA SABIN (1995). "Hexahedral mesh generation by medial surface subdivision : Part I. Solids with convex edges". In : *International Journal for Numerical Methods in Engineering* 38.19. Publisher : Wiley Online Library, p. 3335-3359. ISSN : 0029-5981.
- Nicolas RAY, Wan Chiu LI, Bruno LÉVY, Alla SHEFFER et Pierre ALLIEZ (2006). "Periodic global parameterization". In : *ACM Transactions on Graphics (TOG)* 25.4. Publisher : ACM New York, NY, USA, p. 1460-1485.
- Nicolas RAY, Dmitry SOKOLOV et Bruno LEVY (2016). "Practical 3D frame field generation". In : *ACM Transactions on Graphics (TOG)* 35.6. Publisher : ACM New York, NY, USA, p. 1-9.
- Nicolas RAY, Bruno VALLET, Wan Chiu LI et Bruno LÉVY (2008). "N-symmetry direction field design". In : *ACM Transactions on Graphics (TOG)* 27.2. Publisher : ACM New York, NY, USA, p. 1-13.
- Maxence REBEROL, Alexandre CHEMIN et Jean-François REMACLE (2019). "Multiple approaches to frame field correction for CAD models". In : *arXiv preprint arXiv :1912.01248*.

- Maxence REBEROL, Christos GEORGIADIS et Jean-François REMACLE (2021). “Quasi-structured quadrilateral meshing in Gmsh - a robust pipeline for complex CAD models”. In : *CoRR* abs/2103.04652. arXiv : 2103.04652. URL : <https://arxiv.org/abs/2103.04652>.
- Ronald S RIVLIN (1948). “Large elastic deformations of isotropic materials IV. Further developments of the general theory”. In : *Philosophical transactions of the royal society of London. Series A, Mathematical and physical sciences* 241.835. Publisher : The Royal Society London, p. 379-397. ISSN : 0080-4614.
- Patrick J ROACHE (1998). *Verification and validation in computational science and engineering*. T. 895. Hermosa Albuquerque, NM.
- Robert SCHNEIDERS (2000). “Octree-based hexahedral mesh generation”. In : *International Journal of Computational Geometry & Applications* 10.04. Publisher : World Scientific, p. 383-398. ISSN : 0218-1959.
- Zhongwei SHEN, Xianzhong FANG, Xinguo LIU, Hujun BAO et Jin HUANG (2016). “Harmonic functions for rotational symmetry vector fields”. In : *Computer Graphics Forum*. T. 35. Issue : 7. Wiley Online Library, p. 507-516.
- Jonathan Richard SHEWCHUK (2005). “Triangle : Engineering a 2D quality mesh generator and Delaunay triangulator”. In : *Applied Computational Geometry Towards Geometric Engineering : FCRC'96 Workshop, WACG'96 Philadelphia, PA, May 27-28, 1996 Selected Papers*. Springer, p. 203-222.
- Hang SI (2015). “TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator”. In : *ACM Transactions on Mathematical Software* 41, p. 1-36.
- Juan C SIMO et Thomas JR HUGHES (2006). *Computational inelasticity*. T. 7. Springer Science & Business Media. ISBN : 0-387-22763-6.
- Peter-Pike SLOAN (2008). “Stupid spherical harmonics (sh) tricks”. In : *Game developers conference*. T. 9, p. 42.
- Marco TARINI, Kai HORMANN, Paolo CIGNONI et Claudio MONTANI (2004). “Polycube-maps”. In : *ACM transactions on graphics (TOG)* 23.3. Publisher : ACM New York, NY, USA, p. 853-860. ISSN : 0730-0301.
- Joe F THOMPSON, Zahir UA WARSI et C Wayne MASTIN (1985). *Numerical grid generation : foundations and applications*. Elsevier North-Holland, Inc. ISBN : 0-444-00985-X.
- Amir VAXMAN, Marcel CAMPEN, Olga DIAMANTI, Daniele PANOZZO, David BOMMES, Klaus HILDEBRANDT et Mirela BEN-CHEN (2016). “Directional field synthesis, design, and processing”. In : *Computer Graphics Forum*. T. 35. Issue : 2. Wiley Online Library, p. 545-572.
- Ryan VIERTTEL et Braxton OSTING (2019). “An approach to quad meshing based on harmonic cross-valued maps and the Ginzburg–Landau theory”. In : *SIAM Journal on Scientific Computing* 41.1. Publisher : SIAM, A452-A479.
- David R WHITE, Lai MINGWU, Steven E BENZLEY et Gregory D SJAARDEMA (1995). “Automated hexahedral mesh generation by virtual decomposition”. In : Citeseer, p. 165-176.
- Ciyou ZHU, Richard H BYRD, Peihuang LU et Jorge NOCEDAL (1997). “Algorithm 778 : L-BFGS-B : Fortran subroutines for large-scale bound-constrained optimization”. In : *ACM Transactions on Mathematical Software (TOMS)* 23.4. Publisher : ACM New York, NY, USA, p. 550-560.
- Olek C ZIENKIEWICZ et Robert L TAYLOR (2005). *The finite element method for solid and structural mechanics*. Elsevier. ISBN : 0-08-045558-1.