



HAL
open science

Champs de repères 2,5D pour la génération de maillages hexaédriques

Yoann Coudert–Osmont

► **To cite this version:**

Yoann Coudert–Osmont. Champs de repères 2,5D pour la génération de maillages hexaédriques. Informatique [cs]. Université de Lorraine, 2024. Français. NNT : . tel-04907393

HAL Id: tel-04907393

<https://hal.science/tel-04907393v1>

Submitted on 22 Jan 2025

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.



Distributed under a Creative Commons Attribution 4.0 International License

Champs de repères 2,5D pour la génération de maillages hexaédriques

THÈSE

présentée et soutenue publiquement le 3 décembre 2024

pour l'obtention du

Doctorat de l'Université de Lorraine
(mention informatique)

par

Yoann COUDERT--OSMONT

Composition du jury

Président : David COEURJOLLY

Rapporteurs : David COEURJOLLY CNRS, Université de Lyon 1, INSA Lyon - LIRIS
Frédéric CORDIER Université de Haute Alsace - IRIMAS

Examineur : Margaux RAGUENEL TotalEnergies

Encadrants : Dmitry SOKOLOV Université de Lorraine - LORIA
Nicolas RAY Centre Inria de l'Université de Lorraine

Résumé

Un maillage hexaédrique est une représentation discrète d'un volume sous forme d'un ensemble de cellules hexaédriques (polyèdres à six faces quadrangulaires). Les maillages hexaédriques sont la représentation privilégiée de certaines simulations numériques, mais sont malheureusement difficiles à générer de façon entièrement automatique. Des interventions humaines, parfois longues, sont souvent nécessaires pour leur création. L'automatisation de la génération de ce type de maillages est un sujet de recherches qui a vu naître différentes approches. Cette thèse s'intéresse à l'approche par paramétrisation globale qui s'appuie sur la construction d'une carte possédant certaines propriétés dites de préservation de la grille. Pour générer une telle carte, la méthode semblant la plus prometteuse repose sur la construction d'un champ de repères. Cette méthode ayant fait ses preuves pour la génération de maillages quadrangulaires en 2D se heurte à un nouveau problème en 3D ; le graphe de singularité induit par le champ de repères peut n'être compatible à aucun maillage hexaédrique valide. Une solution à ce problème consiste en l'interdiction de toute singularité interne au volume que l'on cherche à remailler. C'est ce que l'on appelle la méthode des polycubes. La conséquence de ce choix est une forte diminution de la qualité du maillage produit.

Cette thèse propose une alternative en autorisant cette fois-ci des singularités internes, toutes orientées le long d'une même direction, formant en quelque sorte une extrusion d'un maillage quadrangulaire. L'intérêt de cette méthode est illustré dans le cas du remaillage de domaines géologiques.

Mots-clés: Maillage, Paramétrisation, Champ de repères.

Abstract

A hexahedral mesh is a discrete representation of a volume as a set of hexahedral cells (polyhedra with six quadrangular faces). Hexahedral meshes are the preferred representation for certain numerical simulations, but are unfortunately difficult to generate fully automatically. A (long) human intervention is often required to create them. The automation of the generation of hexahedral meshes is a research area that has given rise to various approaches. This thesis focuses on the global parametrization approach, which is based on the construction of a map with the so-called grid preserving property. The most promising method for generating such a map is based on the construction of a frame field. This tried and tested method for generating quadrangular meshes in 2D, comes up against a new problem in 3D : the singularity graph induced by the frame field may not be compatible with any valid hexahedral mesh. One solution to this problem is to prohibit any internal singularity inside the volume we are seeking to remesh. This is known as the polycube method. The consequence of this choice is a significant deterioration of the produced mesh quality.

This thesis proposes an alternative by allowing this time internal singularities, all oriented along the same direction, forming a kind of quadrangular mesh extrusion. The benefits of this method are illustrated in the case of geological domain remeshing.

Keywords: Mesh, Parametrization, Frame field.

Remerciements

Tout d'abord, merci à mes encadrants, Dmitry et Nicolas, pour m'avoir permis de réaliser cette thèse. Merci également aux membres du jury, David Coeurjolly, Frédéric Cordier et Margaux Raguanel, pour avoir relu ce manuscrit et pour m'avoir permis de soutenir.

Mes remerciements s'étendent ensuite naturellement aux autres membres de l'équipe Pixel. Dans un premier temps aux permanents : Dobrina, Laurent (avec qui j'ai pu avoir des discussions intéressantes sur des problèmes du projet Euler), et Étienne, qui s'est toujours montré disponible et patient.

Je voudrais évidemment remercier les doctorants, désormais docteurs, qui étaient présents dès mon arrivée dans l'équipe : François, le premier à soutenir, qui, par sa passion a su donner une bonne dynamique à l'équipe; David, plus qu'un co-bureau, un ami avec qui nous avons pu partager de très belles journées; Justine et Guillaume, avec qui j'ai eu le plaisir de faire de l'escalade.

J'en profite pour remercier le petit groupe de grimpeurs qui a pu se former durant ces années de thèse. Merci à Julia, Louis, Camille, Noémie, Bertrand, Mehdi et Floriane.

Floriane me permet de faire une transition vers la cantine du Loria, véritable point fort du laboratoire. Y manger durant ces trois années a été un véritable plaisir, que ce soit par sa qualité mais aussi par la bonne humeur communicative d'Isabelle.

Le temps passé au Loria a aussi été embelli par de nombreux autres doctorants, que ce soit durant les événements tels que le PhD Pizza Time, le week-end d'intégration ou autour d'un baby-foot ou d'une table de ping-pong. Merci à toutes les rencontres que j'ai pu y faire (désolé, je ne pourrai nommer tout le monde). Petite pensée pour Hee-Soo, la meilleure partenaire de la team pancakes.

Merci également aux personnes avec qui j'ai pu courir de temps en temps le midi : Alexandre, Pierrick, Véronique, Emmanuel et Paul.

Merci à Vincent de m'avoir entraîné dans ce challenge qu'est de courir un trail de 100 km d'ici un an. Je suis heureux de commencer cette nouvelle carrière de traileur professionnel. Merci également à Rémi, qui nous accompagne dans ce périple, et aux autres membres de l'équipe Sémagramme s'étant mis à courir.

Ces trois années ont aussi été l'opportunité de faire plein de choses différentes, n'ayant souvent rien à voir avec le sujet de cette thèse. D'un point de vue académique, je tiens à remercier tous ceux avec qui j'ai pu collaborer sur d'autres publications. Je pense à l'équipe MFX, et en particulier à Adrien, Jonàs, Sylvain et Pierre-Alexandre, avec qui j'ai vécu une de mes plus belles expériences de la recherche.

Je pense aussi à Gabriel, Gaétan, Amadeus et Mathis, rencontrés pour certains à l'ENS de Lyon, et qui ont pensé à moi pour leur participation au PACE Challenge. J'ai adoré écrire ce solveur du problème du coupe-cycle de sommets, et quelle magnifique seconde place nous avons obtenu !

C'est durant cette thèse qu'est née ma réputation d'éternel deuxième dans les compétitions d'algorithmie. Alors, comment ne pas remercier tous ceux qui m'ont privé du haut du podium ? Merci à Rafaël, Arthur, Théophile, le multi-récidiviste, et bien sûr Noé, pour la plus belle des secondes places. Merci également à tous ceux qui font vivre cette discipline. La programmation compétitive m'a vraiment permis de vivre plein de super moments. Merci aux frères Hammel pour le serveur H25, qui m'a semblé essentiel dans l'engouement français que l'on a pu voir sur les derniers ICPC challenges. Merci à Bernhard et Johan, avec qui j'ai pu vivre ma plus belle expérience de compétition en équipe lorsque nous étions en Égypte. Merci à tous les coéquipiers que j'ai pu avoir : je pense à nouveau à David, Justine et Guillaume pour cette

magnifique finale de la Google Hash Code; Marion et Nicolas pour cette victoire à TRACS. Merci à Vivien et Zac de m'avoir accompagné pour ma dernière participation au SWERC, et à mes étudiants des Mines de Nancy et de Télécom Nancy pour m'avoir permis de revivre un SWERC en tant que coach.

Enfin, j'aimerais remercier mes parents et tous les p'tits lapins pour leur amour inconditionnel tout au long de cette aventure.

J'aimerais finalement finir par remercier Marguerite, avec qui j'ai pu partager beaucoup de moments forts agréables durant la dernière année de cette thèse.

Sommaire

Introduction

Chapitre 1

Génération de maillages hexaédriques structurés

1.1	Qualité d'un maillage hexaédrique	3
1.1.1	Graphe de singularité	3
1.1.2	Qualité des éléments	4
1.2	Premières approches de génération de maillages quadrangulaires et hexaédriques non structurés	6
1.3	Paramétrisation globale	9
1.3.1	Méthode des champs de repères	10
1.3.2	Passage à la 3D	13

Chapitre 2

Maillage hexaédriques 2,5D pour la simulation de réservoirs [Lopez et al., 2024]

2.1	Introduction	17
2.2	Méthode principale	21
2.2.1	Calcul du maillage quadrangulaire dans le plan XY	21
2.2.2	Découpage du modèle dans la direction Z	23
2.2.3	Génération du maillage hexaédrique	24
2.3	Généralisation au failles non verticales	25
2.3.1	Formulation avec des failles faiblement verticales	26
2.3.2	Déterminer $org(i), dest(i)$ et λ_i	27
2.3.3	Optimisation de la déformation \mathcal{U}	29
2.4	Résultats	29
2.4.1	Robustesse et efficacité	31
2.4.2	Raffinement local	33
2.4.3	Rétrocompatibilité avec les simulateurs historiques d'écoulement structurés	35

2.4.4	Exemple d'application : simulation d'injection de fluide	35
2.5	Limitations	38
2.5.1	Formulation de l'objectif	38
2.5.2	Limitations algorithmiques	38
2.6	Conclusion	39

Chapitre 3

Champs de directions non orthogonales [Desobry et al., 2021a]

3.1	Introduction	41
3.1.1	Travaux précédents	42
3.1.2	Élaborer une méthode d'optimisation de champs d'orientation	43
3.1.3	Contributions	44
3.2	Champs de directions de dimension n en tant que polynômes sphériques	45
3.2.1	Représentation d'un ensemble de directions	45
3.2.2	Base orthonormée des ensembles de N -directions	45
3.2.3	Champ de N -directions constant par morceaux	47
3.2.4	Degré du polynôme de représentation	47
3.3	Champs de directions 2D	49
3.3.1	Base de Fourier	49
3.3.2	Conception de champs de repères pour le remaillage quadrangulaire	50
3.3.3	Résultats	53
3.3.4	Comparaison avec les champs de polyvecteurs [Diamanti et al., 2014]	55
3.4	Champs de directions 3D	58
3.4.1	Base d'harmoniques sphériques	58
3.4.2	Conception de champs de repères pour le remaillage hexaédrique	59
3.4.3	Remaillage hexaédriques de modèles CAO	60
3.5	Conclusion	63

Chapitre 4

Quantification de cartes sans coutures [Coudert-Osmont et al., 2023]

4.1	Introduction	65
4.2	Cartes préservant la grille	66
4.2.1	Formalisation (irréalisable)	67
4.2.2	Rendre le problème accessible, étape N°1 : champ de repères	67
4.2.3	Rendre le problème accessible, étape N°2 : quantification	68
4.3	État de l'art de la quantification	69
4.4	Notre quantification	71

4.4.1	Un maillage décimé comme proxy	73
4.4.2	Optimisation des degrés de liberté entiers	74
4.4.3	Déduction des contraintes pour le problème initial	79
4.4.4	Détails d'implémentation	79
4.5	Résultats	80
4.5.1	Temps d'exécution	81
4.5.2	Qualité	83
4.5.3	Différences d'expressivité avec QGP	85
4.6	Possibles extensions	86
4.6.1	Bords libres	86
4.6.2	Renforcement des contraintes de courbes caractéristiques durant la quantification	87
4.6.3	Faces dégénérées	87
4.7	Conclusion	89

Conclusion

Bibliographie

Introduction

Les maillages volumiques permettent de représenter mathématiquement des objets dans un ordinateur afin d'estimer leurs comportements physiques lorsqu'ils sont soumis à des forces internes ou externes. Ces maillages sont composés d'un ensemble de cellules élémentaires qui lorsqu'elles sont suffisamment nombreuses permettent de représenter des objets complexes avec précision. Dans l'industrie, on retrouve deux types de cellules massivement utilisés : les tétraèdres (polyèdres composés de quatre faces triangulaires) et les hexaèdres « topologiquement réguliers » (polyèdres composés de six faces quadrangulaires). Ce dernier type de cellule est celui qui nous intéresse dans cette thèse. Il permet de construire des maillages localement très structurés et est utilisé pour résoudre des équations aux dérivées partielles pertinentes pour l'industrie automobile, navale, aérospatiale, médicale et géologique. De nombreux logiciels utilisent actuellement des maillages hexaédriques : [Altair, 2024, Ansys, 2024, Coreform, 2024, Spatial Corp, 2024, Tessel, 2024].

Au sein de la recherche académique, la génération de maillages hexaédriques est étudiée depuis plus de 30 ans. Il existe à ce jour différentes approches à ce problème. Au sein de l'équipe Pixel, nous nous intéressons à l'approche par paramétrisation globale, qui elle-même peut être divisée en plusieurs méthodes. Certaines de ces méthodes sont robustes (polycubes), quand d'autres produisent des maillages de très bonne qualité au détriment de la robustesse (champs de repères). Malheureusement, aucune approche ne combine robustesse et qualité.

Le manque de robustesse de la méthode des champs de repères provient de la difficulté à obtenir un champ de repères induisant un graphe de singularité valide. L'approche par polycube évite ce problème en interdisant toute singularité interne au maillage. Cette solution permet certes de gagner en robustesse, mais réduit également considérablement la qualité des maillages générés. Cette thèse propose, à l'instar des polycubes, de réduire l'ensemble des graphes de singularité possible, mais en autorisant désormais des singularités internes orientées le long d'une direction stable. Cette expressivité supplémentaire sur le graphe de singularité peut permettre de générer des maillages de meilleure qualité en s'appuyant notamment sur la paramétrisation globale 2D qui est désormais relativement bien maîtrisée. Les champs d'orientations 3D dont les singularités sont orientées selon une direction stable peuvent être vus comme des champs d'orientation 2D extrudés. On parlera alors de champs 2,5D pour faire référence à cette réduction de la 3D à la 2D. Pour justifier l'intérêt de cette approche, on pourra s'appuyer sur le remaillage de domaines géologiques qui sont des modèles qui se prêtent très bien à la 2,5D.

En outre, cette nouvelle approche peut directement bénéficier des dernières avancées réalisées dans le cadre du remaillage quadrangulaire. Deux des contributions de ce manuscrit présentent ainsi de nouvelles méthodes pour réaliser certaines étapes du remaillage quadrangulaire par paramétrisation globale. L'une de ces contributions porte sur la génération de champs de repères non orthogonaux, tandis que l'autre propose une nouvelle méthode de quantification de carte sans coutures.

Autre travaux

En parallèle des travaux présentés dans ce manuscrit, ce doctorat aura pour moi été l’occasion d’explorer d’autres domaines avec quatre autres contributions scientifiques.

La première porte sur la génération de courbes remplissantes avec un contrôle sur leur orientation pour l’impression 3D (Fig. 1). Ces travaux en collaboration avec l’équipe Inria MFX ont amené à une publication dans le journal *Computer Graphics Forum* et une présentation dans la conférence internationale *Eurographics* :

Bedel, A., Coudert-Osmont, Y., Martínez, J., Nishat, R. I., Whitesides, S., and Lefebvre, S. (2022). Closed space-filling curves with controlled orientation for 3d printing. In Computer Graphics Forum, volume 41, pages 473–492. Wiley Online Library

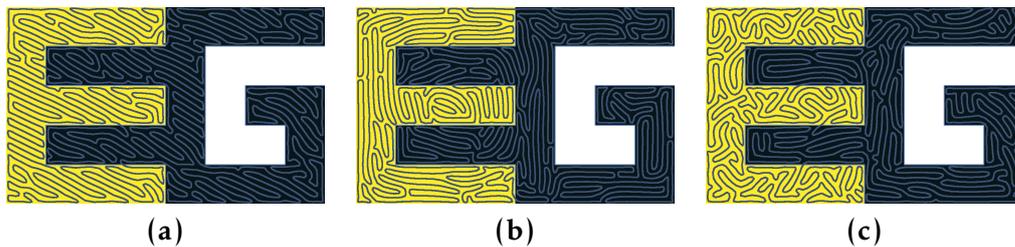


FIGURE 1 – Courbes remplissantes optimisées pour être (a) anisotrope, (b) alignée au bord (c) isotrope à l’intérieur du E et aligné au bord du G. L’isotropie et l’anisotropie sont optimisées en termes de distribution des angles, alors que l’alignement est spécifié par un champ de vecteur (ici parallèle au bord). Pour ces trois courbes, le nombre de transitions entre les deux lettres est minimisé.

Une seconde collaboration avec deux autres chercheurs de TU Delft a aboutie à une présentation à *Eurographics Workshop on Graphics and Cultural Heritage* pour des travaux sur l’extraction automatique de lignes d’horizon et de points de fuite associés à des ensembles de lignes parallèles dans des peintures :

Coudert, Y., Eisemann, E., Marroquim, R., et al. (2022). Semi-automatic perspective lines from paintings. In Eurographics Workshop on Graphics and Cultural Heritage

Finalement, en compagnie d’autres anciens élèves de l’École Normale Supérieure de Lyon, nous avons participé deux années de suite au challenge *Parameterized Algorithms and Computational Experiments (PACE)* consistant au développement de solveurs pour des problèmes NP-difficile de théorie des graphes. Le sujet de notre première participation portait sur le problème du coupe-cycle de sommets. Étant donné un graphe orienté, il s’agit de trouver un plus petit ensemble de sommets à supprimés laissant le graphe acyclique. Notre solveur heuristique s’est classé deuxième de la compétition et a donné lieu à la publication suivante :

Bathie, G., Berthe, G., Coudert-Osmont, Y., Desobry, D., Reinald, A., and Rocton, M. (2022). Pace solver description: Dreyfus. Leibniz International Proceedings in Informatics

Le sujet de notre seconde participation portait sur le calcul de la *Twin-width* d’un graphe [Bonnet et al., 2021]. Notre solveur exacte s’est classé troisième aboutissant à une présentation à la conférence *International Symposium on Parameterized and Exact Computation (IPEC)* :

Berthe, G., Coudert-Osmont, Y., Dobler, A., Morelle, L., Reinald, A., and Rocton, M. (2023). Pace solver description: Touiwidth. In IPEC 2023-18th International Symposium on Parameterized and Exact Computation, volume 285, page 4

Chapitre 1

Génération de maillages hexaédriques structurés

Dans ce chapitre, nous introduisons l'état de l'art de la génération de maillage hexaédrique et présentons la méthode phare de la génération de maillage dit structuré, à savoir les approches par paramétrisation globale.

1.1 Qualité d'un maillage hexaédrique

Désirant construire des maillages hexaédriques de qualité, il est important de commencer par définir ce que nous désirons. Dans un premier temps, nous expliquons la structure et la topologie des maillages que nous souhaitons construire. Puis dans un second temps, nous donnons une intuition des éléments pour mesurer la qualité de la géométrie d'un maillage hexaédrique.

1.1.1 Graphe de singularité

L'intérêt des maillages hexaédriques est leur structure qui localement peut s'apparenter à une grille déformée. On peut remarquer qu'au sein d'une grille chaque sommet est incident à 8 hexaèdres, chaque arête est incidente à 4 hexaèdres et chaque face incidente à 2 hexaèdres. Cependant, les maillages hexaédriques généraux contiennent des éléments qui peuvent être de connectivité différente. Un élément n'ayant pas la même connectivité que dans une grille est dit irrégulier ou encore singulier. Par exemple, les faces irrégulières sont incidentes à un seul hexaèdre, et correspondent aux faces de bord du maillage.

Un sommet singulier, en revanche, peut avoir une connectivité très variée. Toutefois, cette connectivité doit vérifier certaines contraintes comme démontré dans [Nieser et al., 2011]. En effet, tracer une petite sphère englobant un sommet interne au maillage, induit une triangulation de la sphère, où les sommets de cette triangulation correspondent aux intersections entre la sphère et les arêtes du maillage, et les arcs de la triangulation correspondent aux intersections entre la sphère et les faces du maillage hexaédriques (Fig. 1.1). En particulier, en utilisant la relation d'Euler $V - E + F = 2$, où V est le nombre de sommets de la triangulation, E le nombre d'arêtes et F le nombre de faces, ainsi que la relation $2E = 3F$ pour une triangulation et enfin la relation $\sum_i \text{valence}(e_i) = 2E$, où les e_i sont les arêtes incidentes au sommet du maillage

hexaédrique (et donc les sommets de la triangulation), on obtient l'égalité suivante :

$$\sum_i (6 - \text{valence}(e_i)) = 12 \quad (1.1)$$

Cette dernière identité, bien que nécessaire, n'est pas suffisante pour correspondre à une triangulation valide de la sphère. Il n'est en effet pas possible qu'un sommet du maillage hexaédrique soit incident à 6 arêtes régulières (de valence 4) et une arête singulière de valence 6, même si cette dernière configuration satisfait (1.1). De manière générale, il est même impossible pour un sommet interne au maillage d'être incident à une seule arête singulière. Si un sommet interne est incident à une arête singulière, il est nécessairement incident à au moins une autre arête singulière. De plus si un sommet est incident à précisément deux arêtes singulières, alors celles-ci sont de même valence.

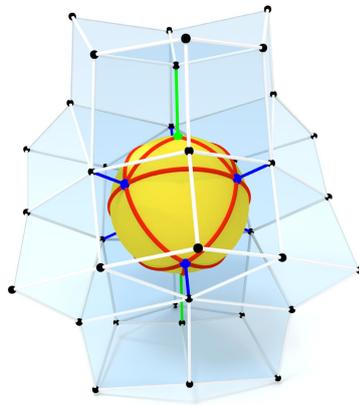


FIGURE 1.1 – Une sphère jaune est tracée autour d'un sommet interne à un maillage hexaédrique. L'intersection des faces du maillage avec la sphère forme une triangulation de cette dernière en rouge. Image issue de [Liu et al., 2018]

Cette dernière propriété implique que l'ensemble des arêtes singulières forme des courbes soit fermées, soit ayant leurs extrémités sur le bord du maillage ou sur un sommet où plusieurs courbes s'intersectent. Par la suite, on nommera cet ensemble, le graphe de singularités. Les courbes seront appelées arcs du graphe de singularités et les sommets singuliers désigneront alors uniquement les extrémités de ces arcs. Il est également important de noter qu'une arête de bord est régulière si et seulement si sa valence est 2 (et non 4). La figure 1.2 représente un graphe de singularité.

Désirant générer des maillages structurés/réguliers, il sera souhaitable par la suite de construire des maillages hexaédriques avec un graphe de singularité le plus petit et le plus simple possible.

1.1.2 Qualité des éléments

En plus de la structure/topologie du maillage généré, il est également important de s'intéresser à la qualité de la géométrie des éléments produits. Pour un maillage hexaédrique, il est préférable que les cellules ressemblent le plus possible à de petits cubes. Pour cela, il existe de nombreuses métriques mesurant la qualité d'un hexaèdre. [Knupp, 2003] en proposent 3 principales en plus de présenter plusieurs combinaisons entre ces dernières :

- La métrique de taille relative

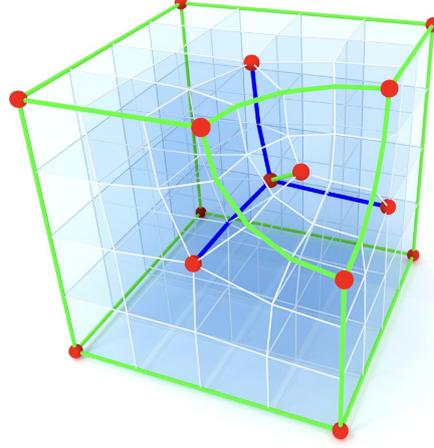


FIGURE 1.2 – Exemple d'un maillage hexaédrique. Les arêtes vertes et bleues sont celles de valence 3 et 5 respectivement (valence 1 et 3 sur le bord). Les sommets rouges sont les sommets singuliers. *Image issue de [Liu et al., 2018]*

- La métrique de forme
- La métrique d'inclinaison

Ces métriques reposent généralement sur la matrice jacobienne d'une fonction envoyant le cube unité $[0, 1]^3$ sur l'hexaèdre en question. Cette jacobienne est alors évaluée sur les 8 sommets du cube, la jacobienne étant interpolée de façon convexe à l'intérieur du cube. Nous proposons de l'illustrer sur l'une des métriques.

Métrique de taille relative La métrique de taille relative mesure à quel point le « volume » de l'hexaèdre diffère d'un volume de référence \mathcal{V} . On note f la fonction qui envoie le cube unité sur l'hexaèdre. La jacobienne du k -ième sommet de coordonnées $(u, v, w) \in \{0, 1\}^3$ est donnée par :

$$J_k = \begin{pmatrix} f(1, v, w)^T - f(0, v, w)^T \\ f(u, 1, w)^T - f(u, 0, w)^T \\ f(u, v, 1)^T - f(u, v, 0)^T \end{pmatrix}^T \quad (1.2)$$

Le volume de l'hexaèdre est estimé par la moyenne des déterminants des jacobienes. On note ensuite τ le rapport avec le volume de référence \mathcal{V}

$$\tau = \frac{1}{8\mathcal{V}} \sum_{k=1}^8 \det J_k \quad (1.3)$$

La métrique de taille relative f_{taille} est alors définie par :

$$f_{\text{taille}} = \min(\tau, 1/\tau) \quad (1.4)$$

f_{taille} est compris dans $[0, 1]$, plus sa valeur est grande, plus le volume de l'hexaèdre est proche de la référence \mathcal{V} , avec égalité lorsque $f_{\text{taille}} = 1$. Cette valeur ne dépend que du volume et pas de la forme de l'élément. C'est pourquoi elle est souvent couplée avec d'autres métriques. Les métriques de forme et d'inclinaison font par exemple intervenir les valeurs de la diagonale de J_k , afin de mesurer la distorsion de l'élément considéré. La métrique de forme est maximale lorsque l'élément est un cube et la métrique d'inclinaison est maximale lorsque l'élément est un pavé droit (dont les côtés ne sont pas nécessairement de la même longueur).

1.2 Premières approches de génération de maillages quadrangulaires et hexaédriques non structurés

Dans cette section, nous présentons quelques approches très simples de génération de maillages quadrangulaires et hexaédriques. Ces méthodes construisent des maillages non structurés qui pourront nous permettre d'illustrer ce que l'on ne veut pas. On sera alors amené à considérer d'autres approches pour obtenir des résultats plus structurés, dont les méthodes de paramétrisation globales étudiées durant cette thèse.

Génération de maillages tétraédriques

Les méthodes de génération de maillage hexaédriques peuvent être séparées en deux catégories : les méthodes directes prenant en entrée une description de la surface du volume à mailler (souvent à l'aide de primitives telles que des pavés droits ou des sphères), et les méthodes dites indirectes, qui prennent en entrée un maillage tétraédrique. On s'intéressera essentiellement à ces dernières. Les méthodes indirectes font sens, car la génération de maillages tétraédriques est désormais correctement maîtrisée. Des algorithmes quasi robustes de tétraédrisation par des méthodes d'avancée de front existaient déjà il y a 30 ans [Schöberl, 1997]. Aujourd'hui, une référence pour la génération de maillages tétraédriques de qualité est un programme écrit en C++ du nom de TetGen [Si, 2015], dont le code source est libre. Ce programme très robuste repose sur des algorithmes de Delaunay incrémentaux et contraints (pour imposer la présence de certaines arêtes ou certains triangles dans le résultat). En plus d'être robuste, les algorithmes de triangulation de Delaunay sont également très efficaces. De récents travaux ont permis de calculer un maillage contenant trois milliards de tétraèdres en seulement une minute sur un seul ordinateur [Marot et al., 2018], mais aussi de développer des algorithmes de Delaunay incrémentaux de manière concurrente.

Il est ainsi raisonnable de considérer que nous disposons de maillages tétraédriques des modèles volumiques dont nous voulons obtenir des maillages hexaédriques. La suite de cette section présente quelques méthodes indirectes et non structurées.

Méthode de subdivisions

La méthode probablement la plus immédiate pour construire un maillage hexaédrique est probablement de subdiviser un maillage tétraédrique en insérant un nouveau sommet au milieu de chaque arête, de chaque face, et de chaque tétraèdre (Fig. 1.3). Le maillage obtenu n'est pas du tout structuré. En 2D, un sommet singulier de valence 3 est inséré à l'intérieur de chaque face du maillage triangulaire. En 3D, 4 arêtes singulières de valence 3 sont insérées à l'intérieur de chaque cellule du maillage tétraédrique. En plus d'obtenir une mauvaise topologie, les éléments quadrangulaires et hexaédriques produits sont de mauvaise qualité. Cette méthode très simpliste est donc à proscrire. Toutefois, l'idée de subdivision a pu être utilisée par le passé, en subdivisant une décomposition d'un modèle en plusieurs régions polyédriques [Li et al., 1995]. Si le nombre de régions est faible, le maillage obtenu peut alors avoir une topologie correcte. Cette décomposition était soit calculée à la main, soit avec des algorithmes s'appuyant souvent sur la surface médiane du modèle (ensemble des points internes au modèle dont le plus proche point du bord n'est pas unique) [Price et al., 1995, Price and Armstrong, 1997]. La subdivision d'un tétraèdre décrite au début de ce paragraphe s'étend naturellement à n'importe quel polyèdre permettant alors de remailler chaque région polyédrique avec des cellules hexaédriques.

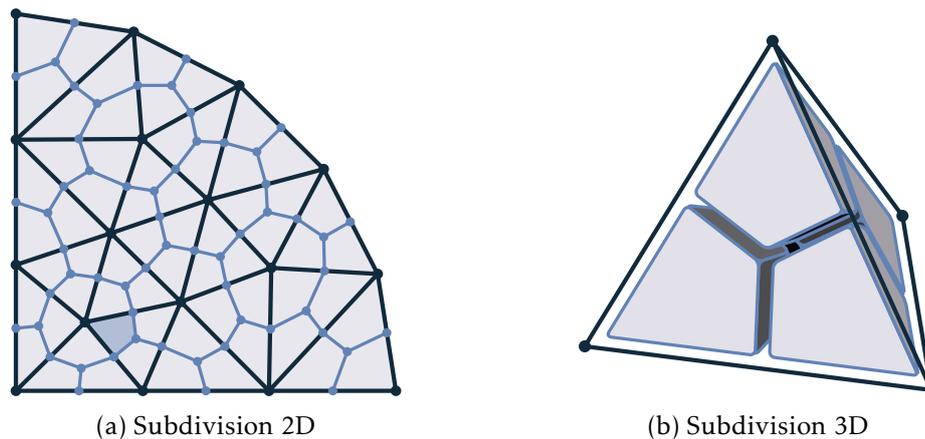


FIGURE 1.3 – Subdivision d’un maillage simplicial (noir) pour obtenir un maillage quadrangulaire en 2D ou hexaédrique en 3D (bleu). Un quadrilatère est mis en évidence en bleu sur la figure 2D.

Fusion de simplexes

Une approche qui s’est avérée plus intéressante que la subdivision en 2D est la fusion des simplexes. En effet, il est possible de fusionner deux triangles voisins (partageant une arête en commun) pour former un quadrilatère. En revanche, pour un maillage triangulaire quelconque, il n’existe aucune garantie sur la possibilité de fusionner tous les triangles afin d’obtenir un nouveau maillage ne contenant que des quadrilatères. Comme l’illustre la figure 1.4, par exemple, un nombre de triangles impair rend impossible l’obtention d’un maillage entièrement quadrangulaire. Heureusement, pour une surface fermée (sans bord), chaque triangle est voisin de trois autres triangles. Le graphe de voisinage des triangles est alors un graphe cubique, c’est à dire, que tous les sommets de ce graphe sont de valence 3. Le fait qu’une surface soit sans bord implique également que ce graphe est sans ponts, permettant alors l’utilisation du théorème de Petersen, pour en déduire l’existence d’un couplage parfait [Petersen, 1891]. Il est alors possible d’obtenir un maillage entièrement quadrangulaire en fusionnant les couples de triangles de ce couplage parfait. Pour obtenir le couplage, il est possible de recourir à l’algorithme d’Edmonds, aussi connu sous le nom de l’algorithme des fleurs et des pétales [Edmonds, 1965]. La complexité est cubique en la taille de la triangulation, mais en pratique l’algorithme est extrêmement efficace. Il existe toutefois des algorithmes ayant une meilleure complexité, comme [Micali and Vazirani, 1980] avec une complexité en $\mathcal{O}(EV^{1/2})$ où E et V sont respectivement les nombres d’arêtes et de sommets dans le graphe considéré. Le nombre d’arêtes étant linéaire en le nombre de sommets dans notre cas, on aboutit à une complexité en $\mathcal{O}(V^{3/2})$. Quant à la qualité du maillage produit, lorsque le maillage triangulaire initial et les fusions effectuées sont faits naïvement, la structure contient toujours de nombreuses singularités qui pourraient être évitées, et de nombreux éléments sont de faible qualité comme l’illustre la figure 1.4 (en particulier l’élément en bas à gauche). Des travaux ont exploré plus sérieusement cette piste de fusion de triangles pour générer des maillages quadrangulaires. En associant des poids aux paires de triangles voisins pour ensuite utiliser un algorithme de couplage de poids minimal, il est possible d’obtenir de bien meilleurs résultats [Remacle et al., 2012].

Malheureusement, étendre cette approche à la 3D semble compliqué. En effet, la fusion de tétraèdres partageant une face donne un hexaèdre à faces triangulaires, appelé diamant triangulaire. Les méthodes de fusions de simplexes 3D fusionnent alors plus de deux tétra-

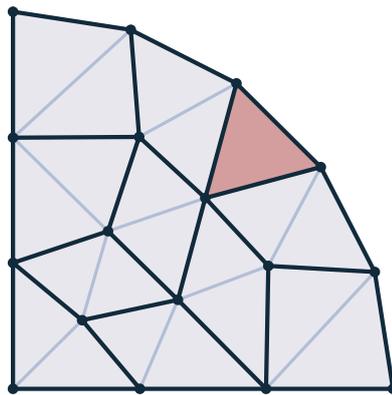


FIGURE 1.4 – Fusion de paires de triangles pour former un maillage à dominance quadrangulaire. Les paires de triangles incidents aux arêtes bleu clair sont fusionnées pour former des quadrilatères. Le nombre de triangles initiaux étant impair, un triangle (en rouge) ne peut être fusionné à la fin du processus.

édres à la fois, rendant le processus plus complexe. Une méthode de référence est *H-morph* [Owen and Saigal, 2000], qui après avoir quadrangulé le bord du domaine, utilise une méthode d'avancée de front combinant les tétraèdres pour former des hexaèdres. Ce processus aboutit généralement à un maillage mixte, contenant à la fois des hexaèdres, des tétraèdres, et des pyramides à base quadrangulaire pour connecter les tétraèdres et les hexaèdres entre eux.

Octree

Il y a 30 ans, [Schneiders, 1996] proposait de construire une grille régulière alignée aux axes euclidiens à l'intérieur du modèle, qui était ensuite reliée à une quadrangulation de la surface pour former un maillage hexaédrique conforme au bord. Contrairement aux deux méthodes présentées précédemment, le maillage obtenu est très régulier à l'intérieur du modèle. En revanche, les éléments proches du bord sont de faible qualité et contiennent de nombreuses singularités, rendant le résultat finalement peu exploitable pour des simulations de déformation. Dans ce même article, l'auteur proposait une méthode de raffinement local pour les grilles 2D avec une structure similaire à un quadtree à l'aide des éléments de transitions présentés dans [Schneiders and Debye, 1995]. Plus tard, cette idée est reprise pour construire directement un octree qui est ensuite transformé en un maillage hexaédrique dont les cellules sont strictement convexes, plutôt que de construire une grille pour la raffiner par la suite [Schneiders et al., 1999]. En réalité il ne s'agit pas d'un octree mais de ce qu'on pourrait appeler un 27-tree puisque les noeuds de l'arbre sont divisés en $3^3 = 27$ plutôt que $2^3 = 8$ pour octree. Toutefois, l'octree ne permet de mailler que l'intérieur du modèle et doit toujours être étendu au bord du domaine.

Quelques années plus tard, une nouvelle approche a permis de mailler entièrement le modèle via un octree et pas seulement l'intérieur [Maréchal, 2009]. L'idée est alors d'englober l'entièreté du modèle dans un cube, la racine de l'octree. Celui-ci est alors subdivisé récursivement en 8 pour obtenir la densité voulue localement. L'octree est ensuite modifié à l'aide de patrons pour obtenir un maillage polyédrique dont toutes les cellules sont strictement convexes et dont chaque sommet est de valence 6 et chaque arête est de valence 4. Ce dernier maillage n'est pas hexaédrique et peut par exemple contenir des tétraèdres. En revanche, en construisant son dual, on obtient un maillage hexaédrique – les sommets devenant des cellules, les arêtes des

faces, et inversement, les cellules du nouveau maillage possèdent 6 faces (sommet de valence 6 dans le primal) quadrangulaires (arête de valence 4 dans le primal). Cette dernière approche (illustré par la figure 1.5) est très robuste et de nouvelles contributions continuent d’y être apportées [Livesu et al., 2021].

Aujourd’hui, les méthodes à base d’octree sont les seules entièrement automatiques capables de remailler hexaédriquement n’importe quel modèle d’entrée, quelle que soit la complexité géométrique ou topologique. Pour cette raison, c’est souvent cette solution qui est implémentée dans de nombreux logiciels tels que [Coreform, 2024, Spatial Corp, 2024].

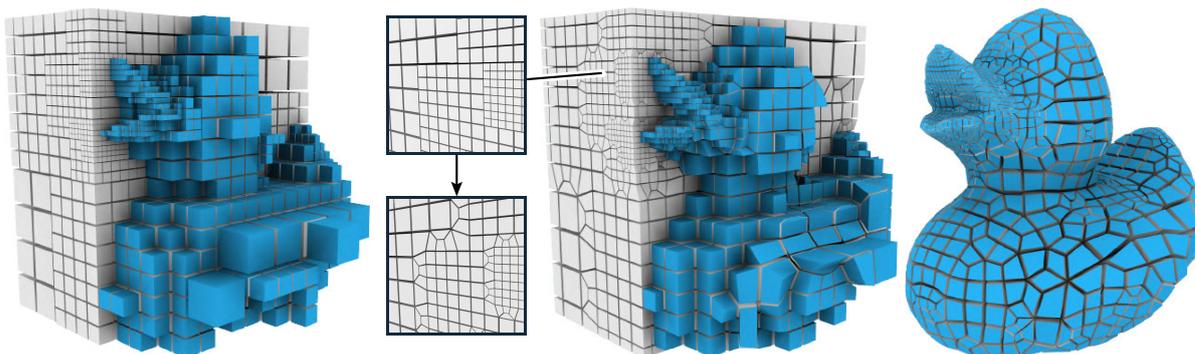


FIGURE 1.5 – Un maillage hexaédrique (droite) peut être obtenu par dualisation (milieu) d’un premier maillage polyédrique obtenu à l’aide d’un octree (gauche). *Images issues de [Livesu et al., 2021]*

Nous avons abordé quelques-unes des premières méthodes de génération de maillages hexaédriques, qui bien que robustes pour certaines, ne permettent de pas de générer de manière automatique des maillages de bonne qualité. Pour d’autres approches, il est possible de se référer à une récente étude de l’état de l’art [Pietroni et al., 2022]. Il en ressort que les méthodes les plus prometteuses pour générer des maillages hexaédriques de bonne qualité de façon automatique sont celles qui s’appuient sur la paramétrisation globale, que nous développons dans la prochaine section.

1.3 Paramétrisation globale

L’idée de la paramétrisation globale est de calculer une paramétrisation/carte u envoyant le domaine \mathcal{M} que l’on cherche à mailler vers un sous-ensemble de l’espace euclidien $\Omega \subset \mathbb{R}^n$, de sorte que l’application de la carte inverse u^{-1} sur la grille unité produise un maillage quadrangulaire/hexaédrique du domaine \mathcal{M} . Par cette méthode, on espère pouvoir mieux contrôler la structure du maillage produit ainsi que la qualité de ses éléments. La carte u est construite comme une application linéaire par cellule d’un maillage simplicial du domaine \mathcal{M} . Un maillage est dit simplicial lorsque ses cellules sont des simplexes (triangle, tétraèdre, ...).

Pour que l’image de la grille unité par u^{-1} produise un maillage quadrangulaire/hexaédrique, il est nécessaire que u satisfasse certaines contraintes.

- Premièrement la carte ne doit pas avoir de retournements, c’est-à-dire que sur chaque simplexe, le déterminant de la jacobienne de u doit être strictement positif. Pour calculer la jacobienne, une base directe de l’espace tangent associé au simplexe doit être considérée (pas de base indirecte).

- Dans un deuxième temps la transition de u entre deux cellules adjacentes doit préserver la grille. Considérons la restriction linéaire u_1 à la première cellule, et la restriction u_2 à la seconde. Alors, sur la facette correspondant à l'intersection entre les deux cellules, l'application linéaire $u_2 \circ u_1^{-1}$ doit préserver la grille. Ici, linéaire est un abus de langage et doit être compris comme affine. Les applications affines qui envoient la grille unité sur elle-même sont engendrées (via l'opération de composition) par les translations entières ($x \mapsto x + a$, avec $a \in \mathbb{Z}^n$) et les rotations d'un angle de $\pi/2$ autour des axes euclidiens.
- Enfin, l'image par u , d'un point du bord du domaine, doit avoir une coordonnée entière, afin que le maillage obtenu soit conforme au bord.

Une carte satisfaisant ces contraintes est dite « *préservant la grille* ». Pour obtenir des éléments quadrangulaires ou hexaédriques de qualité, on peut remarquer que la jacobienne décrite dans la section 1.1.2 et intervenant dans les métriques de qualité est en réalité la jacobienne de u^{-1} . Il est alors directement possible de contrôler la qualité du maillage produit en optimisant u pour minimiser certaines énergies s'appuyant sur les métriques de qualité. Pour ce qui est de la structure du maillage produit, cela dépend des rotations des fonctions de transitions entre cellules adjacentes. Ces dernières définissent la topologie et les singularités du maillage. On reviendra plus tard sur ce point.

Finalement, l'approche par paramétrisation globale consiste à optimiser une carte u sous certaines contraintes. Les contraintes faisant entre autres apparaître des valeurs entières font du calcul de u une tâche difficile. C'est pour cela que des méthodes pour calculer une carte préservant la grille ont été développées, divisant l'approche en plusieurs étapes plus simples.

1.3.1 Méthode des champs de repères

La méthode des champs de repères introduite en 2007 [Kälberer et al., 2007] propose une façon de séparer la quadrangulation par paramétrisation globale en plusieurs étapes plus accessibles. La première est la création d'un champ de repères permettant de fixer la topologie du maillage généré, c'est-à-dire son graphe de singularités. La seconde étape transforme le champ de repères défini sur la surface en un champ de vecteur défini sur un 4-revêtement de la surface. Ce champ de vecteur est ensuite intégré pour obtenir un champ scalaire sur le 4-revêtement. Les coordonnées d'un point de la surface au sein de la carte u sont alors obtenues par les valeurs du champ scalaire sur les 4 images de ce point par le 4-revêtement. Enfin, les dernières fonctions de transition ne préservant pas la grille à cause d'une translation non entière sont corrigées par une étape dite de quantification. La figure 1.6 illustre ces étapes.

Champ de repères

Un champ de repères est la donnée de deux vecteurs tangents à chaque face d'une triangulation de la surface à remailler. Ces deux vecteurs peuvent être vus comme les deux lignes d'une matrice 2×2 , et plus particulièrement ici, comme les lignes de la matrice jacobienne J de la carte u (c'est-à-dire les gradients des coordonnées de la carte $u : \mathcal{M} \rightarrow \mathbb{R}^2$). L'idée derrière la construction d'un champ de repères est de calculer une approximation de la jacobienne d'une carte u en chaque triangle. Les propriétés que doit satisfaire localement un tel champ sont plus simples que celles devant être satisfaites par u . En effet, les fonctions de transitions entre deux triangles voisins ne sont plus des compositions de translations entières et rotations d'angle $\pi/2$, mais simplement des rotations d'angles multiples de $\pi/2$. Les seules variables entières qui interviennent alors dans le calcul d'un champ de repères correspondent à ces rotations. Les contraintes d'entièrement des translations intervenantes dans les fonctions de transitions de u

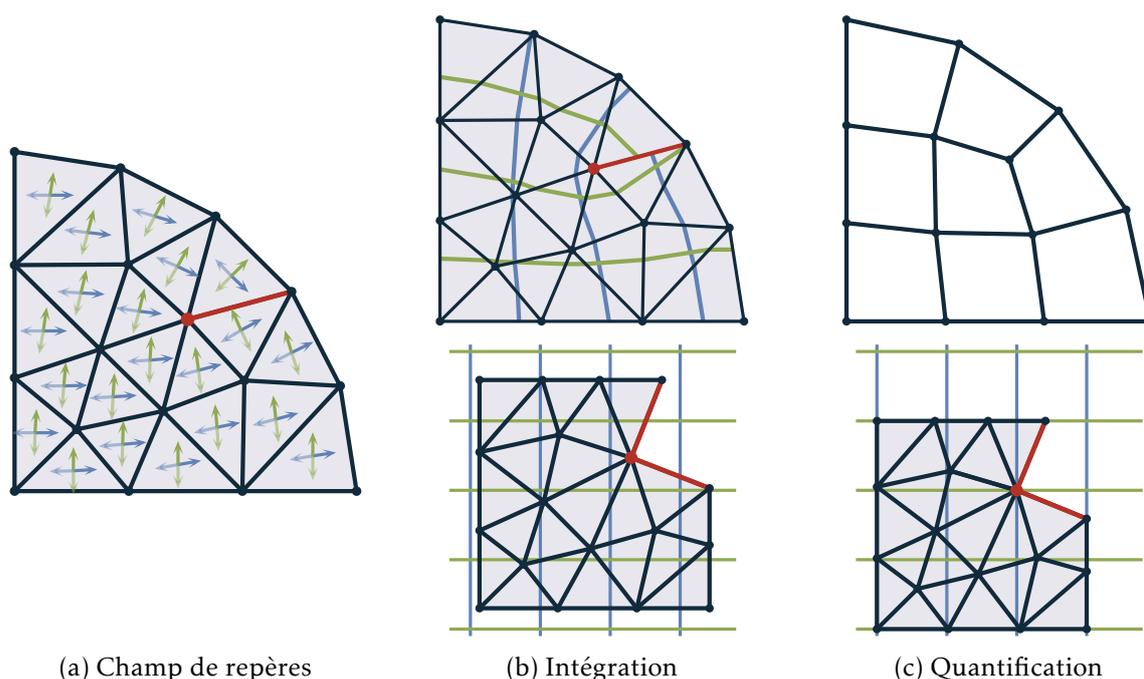


FIGURE 1.6 – Les trois étapes de la méthode de remaillage quadrangulaire par champ de repères. (a) Un champ de repères est d’abord construit. Une direction du repère est orthogonale à chaque arête de bord. Le champ k obtenu dans l’exemple est partout égale à l’identité sauf pour l’arête rouge pour laquelle il est nécessaire d’appliquer une rotation d’un angle de $\pi/2$ pour passer du repère d’un côté de l’arête à celui de l’autre côté. Cette arête rend le sommet en rouge singulier puisque ce dernier est contraint à être de valence congrue à 3 modulo 4 dans le maillage quadrangulaire final. (b) Ce champ de repères est ensuite intégré pour obtenir une carte qui ne préserve pas la grille à travers l’arête rouge et dont les bords ne sont pas alignés à la grille unité. (c) Enfin, l’étape de quantification fixe les valeurs entières des dernières translations et aligne les bords de la carte avec la grille unité permettant ainsi d’extraire un maillage quadrangulaire depuis la carte préservant la grille obtenue.

deviennent des contraintes globales et non locales pour J . Ces contraintes globales peuvent être dans un premier temps ignorées (elles seront satisfaites lors de l’étape de quantification).

Les champs de repères considérés vérifieront toujours $\det J > 0$ afin de satisfaire le premier point des contraintes sur les cartes préservant la grille.

Le troisième point des contraintes devant être satisfaites par u indique qu’une coordonnée de u doit être constante le long des arêtes de bord du domaine. Cela se traduit par une ligne de J devant être orthogonale à la direction de l’arête de bord (une des directions du repère doit être orthogonale au bord).

Considérons les jacobiniennes de deux faces adjacentes notées J_1 et J_2 . La fonction de transition entre les deux faces est une rotation d’un angle $k_{12} \frac{\pi}{2}$ notée $R_{\pi/2}^{k_{12}}$. Si on note e le vecteur représentant l’arête commune aux deux faces alors $(J_2 - R_{\pi/2}^{k_{12}} J_1) \cdot e = 0$. La méthode des champs de repères propose de relaxer cette égalité et d’utiliser un champ de repères dont la distance entre J_2 et $R_{\pi/2}^{k_{12}} J_1$ est la plus faible possible pour une métrique donnée. Il existe de nombreuses métriques accompagnées de leurs différentes méthodes pour être minimisées en étant sujettes aux contraintes de bord du paragraphe précédent. Il est possible de se référer à une étude des

différentes méthodes de génération de champs de directions pour cela [Vaxman et al., 2016]. À noter que le problème décrit possède pour variables, à la fois les jacobiniennes et les entiers k associés aux transitions. Bien que certaines méthodes comme [Ray et al., 2008] procèdent en deux étapes, en déterminant dans un premier temps les valeurs de k pour ensuite optimiser les jacobiniennes à k fixé, la majorité des méthodes les plus récentes proposent une métrique invariante par rotation de $\pi/2$ à gauche d'une jacobienne (i.e. la distance entre J_2 et $R_{\pi/2}^{k_{12}} J_1$ est la même que celle entre J_2 et J_1) permettant d'optimiser J indépendamment de k , puis de calculer k a posteriori. Le chapitre 3 propose une telle méthode avec la particularité d'autoriser les colonnes de J à être non orthogonales.

Finalement les variables k permettent de définir la topologie de la carte qui sera obtenue par intégration du champ de repères J . En particulier un sommet de la triangulation produira une singularité dans le maillage quadrangulaire si la somme des valeurs de k associées aux arêtes qui lui sont incidentes est congrue à une valeur non nulle modulo 4. Cette somme est en fait congrue à la valence du sommet modulo 4.

Intégration

Comme le champ de repère obtenu à l'étape précédente ne satisfait pas strictement l'égalité $(J_2 - R_{\pi/2}^{k_{12}} J_1) \cdot e = 0$ pour chaque paire de triangles adjacents, le champ J ne correspond pas exactement à la jacobienne d'une carte u . Il n'existe donc pas de façon triviale d'obtenir une carte à partir de J . Toutefois, il est possible de calculer une carte u dont la partie rotationnelle des fonctions de transition est donnée par le champ k et dont la jacobienne est la plus proche possible de J . C'est ce qu'on appelle l'intégration d'un champ de repère. Il existe différentes manières de mesurer la distance entre la jacobienne de u (notée Du) et J . Cette distance peut être mesurée au sens des moindres carrés avec la norme de Frobenius $\|Du - J\|^2$ comme le propose [Kälberer et al., 2007, Bommès et al., 2009]. Malheureusement minimiser cette distance a tendance à produire des retournements au sein de la carte ($\det(Du) < 0$). Pour pallier ce problème, il existe d'autres énergies pouvant être minimisées [Garanzha et al., 2021].

Quantification

La carte u obtenue après intégration vérifie désormais toutes les contraintes nécessaires pour préserver la grille sauf une. Les translations dans les fonctions de transitions ne sont pas nécessairement entières. Pour corriger cela, [Kälberer et al., 2007] introduit la notion de graphe de coupure. Il est en effet possible de découper la surface à remailler le long d'un ensemble d'arêtes passant par toutes les singularités de sorte que la translation à travers toutes les arêtes n'appartenant pas à ce graphe puisse être fixée à zéro. La translation de la fonction de transition le long d'arêtes du graphe de coupure reliant deux singularités est constante, réduisant ainsi drastiquement le nombre de translations auxquelles il faut imposer d'être entier. La quantification consiste à calculer des valeurs entières pour ces dernières translations et à recommencer un processus d'intégration du champ de repère J en contraignant cette fois ci les translations aux valeurs entières nouvellement calculées pour obtenir une nouvelle carte préservant la grille (Fig. 1.7). Ces valeurs entières ont historiquement été obtenues par des processus d'arrondissement [Kälberer et al., 2007, Bommès et al., 2009]. Puis, de nouveaux algorithmes aboutissant à de meilleurs résultats ont été conçus [Campen et al., 2015]. Le chapitre 4 présente un nouvel algorithme se voulant plus simple, mais aussi apportant de nouveaux avantages par rapport à [Campen et al., 2015].

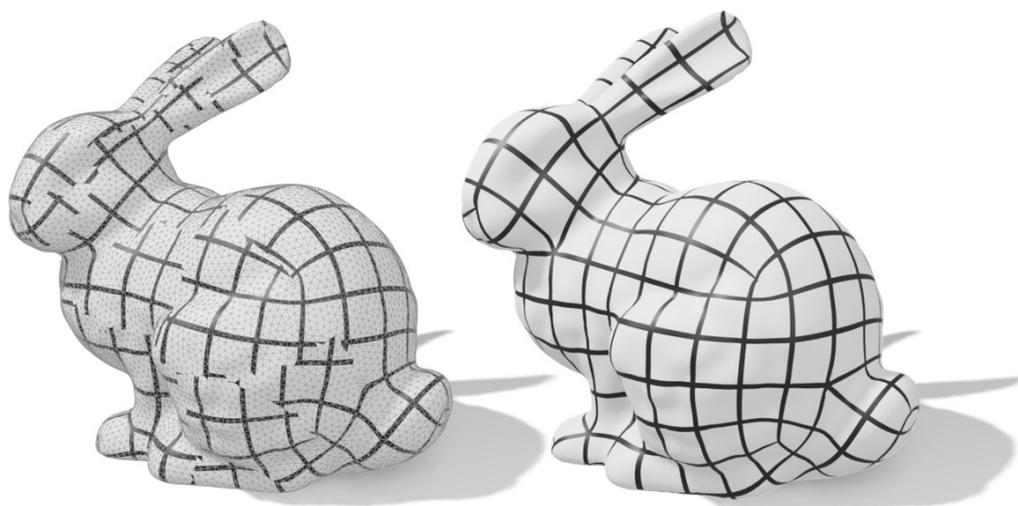


FIGURE 1.7 – Quantification d’une carte possédant des translations non entières le long d’un graphe de coupure (gauche) pour obtenir une carte préservant la grille (droite). L’algorithme de quantification utilisé, est celui présenté dans le chapitre 4.

1.3.2 Passage à la 3D

Champs de repères 3D

Adapter la méthode des champs de repères à la génération de maillages hexaédriques en 3D ne se fait pas aussi naturellement qu’on pourrait le vouloir. [Nieser et al., 2011] propose une telle adaptation nommée *CubeCover*, mais montre également que de nouveaux défis apparaissent en 3D.

En particulier, l’équation (1.1) nécessaire à la validité d’un sommet singulier y est démontrée. Aucune telle condition ne devait être vérifiée en 2D. Tout champ k était localement valide (au sens qu’il correspond à un maillage quadrangulaire) en 2D. Malheureusement, un champ k associant une symétrie directe du cube à chaque face triangulaire d’un maillage tétraédrique (au lieu d’une symétrie directe du carré à chaque arête en 2D), n’est pas toujours valide. Similairement à la 2D, la somme des valeurs de k à travers les faces incidentes à une arête du maillage tétraédrique est congrue à la valence de cette arête modulo 4. L’équation (1.1) conditionnant la somme des valences des arêtes incidentes à un sommet, cette équation restreint localement les champs k correspondant à un maillage hexaédrique. Les méthodes de génération de champs de repères en 3D doivent donc tenir compte de cette équation rendant la création d’un champ de repère valide bien plus difficile puisque le champ J ne peut désormais plus être optimisé indépendamment de k comme le font la plupart des méthodes 2D.

L’étape d’intégration ne présente aucun nouveau défi par rapport à la 2D, et peut être accomplie sans nouvelles difficultés. En revanche, les méthodes de quantifications plus avancées que les processus d’arrondissement s’étendent difficilement à la 3D. La méthode QGP [Campen et al., 2015] a pu être étendue au coût de devoir recourir à un solveur générique de problème à valeurs entières plutôt que des heuristiques efficaces conçues spécialement pour la quantification de cartes préservant la grille [Brückler et al., 2022a]. Afin de guider le solveur générique, des contraintes réduisant l’espace des cartes préservant la grille doivent être ajoutées durant l’optimisation. La méthode que nous présentons dans le chapitre 4, elle, n’a pas encore été étendue à la 3D.

Méthode polycube

Une possibilité pour faire face à la difficulté de la génération d'un champ de repères 3D valide est d'interdire toute singularité interne. Le remaillage hexaédrique par l'approche des polycubes propose ainsi de remplacer la partie génération d'un champ de repères, par une coloration des faces de bord du maillage tétraédrique d'entrée. Par coloration, nous entendons un choix de la coordonnée de la carte u devant être constante le long de chaque face de bord du domaine. Plus précisément, une normale dans l'espace paramétrique est assignée à chaque face de bord. Les normales assignées sont contraintes à être orientées le long des axes euclidiens résultant en 6 choix possibles $(\pm X, \pm Y, \pm Z)$. Malheureusement, même pour la coloration des bords d'un modèle, nous ne possédons pas encore de méthode entièrement robuste. Cette tâche est toutefois aujourd'hui mieux maîtrisée que la génération de champs de repères valides, avec de multiples méthodes pour la résoudre. C'est le grand avantage de la méthode polycube. On peut citer des approches s'appuyant sur une tessellation de Voronoï centroïdale [Hu and Zhang, 2016], ou d'autres traduisant le problème en un problème de coupe minimum d'un graphe [Livesu et al., 2013]. Pour plus d'approches, il est possible de se référer à l'étude des méthodes de remaillage hexaédrique [Pietroni et al., 2022].

Construisant une carte sans singularités internes, les fonctions de transitions de la carte u peuvent être toutes fixées à l'identité (une translation nulle et une partie rotationnelle valant l'identité). L'étape d'intégration de la méthode des champs de repères peut être réalisée exactement de la même manière, hormis que cette fois-ci, aucun champ de repères J guide la déformation des tétraèdres. Cette absence de champ J ne pose pas réellement de problèmes pour une optimisation comme celle proposée par [Garanzha et al., 2021], le champ peut être supposé constant et valoir l'identité en tout point.

Enfin, l'approche de la paramétrisation globale par polycube possède comme second avantage qu'elle simplifie l'étape de quantification. La quantification des cartes polycuboidales, est aujourd'hui bien plus mûre avec des algorithmes robustes [Protais et al., 2022], contrairement à la quantification des cartes possédant des singularités internes.

Un résumé de l'approche par polycubes est illustré par la figure 1.8. Le principal problème de cette approche est la qualité des éléments produits, en particulier près des singularités placées sur le bord. Sur la figure illustrant la méthode, on peut constater que les éléments incidents aux 4 singularités situées à la hauteur du tunnel cylindrique sont de mauvaise qualité. La qualité peut toutefois être améliorée par des étapes de rembourrages (*padding* en anglais) consistant en l'ajout de nouvelles couches d'hexaèdres enveloppant le domaine et repoussant ainsi les singularités à l'intérieur du maillage. Bien que le rembourrage puisse améliorer la qualité des éléments, la topologie du graphe de singularité, elle, est inchangée encore que la topologie produite par la méthode polycube ne soit pas souvent celle qui semble la plus naturelle. Toujours sur le même exemple, les deux paires de singularités de valence 1 et 3 situées sur la partie supérieure du cylindre pourraient être combinées pour faire disparaître toutes singularités dans cette région du domaine.

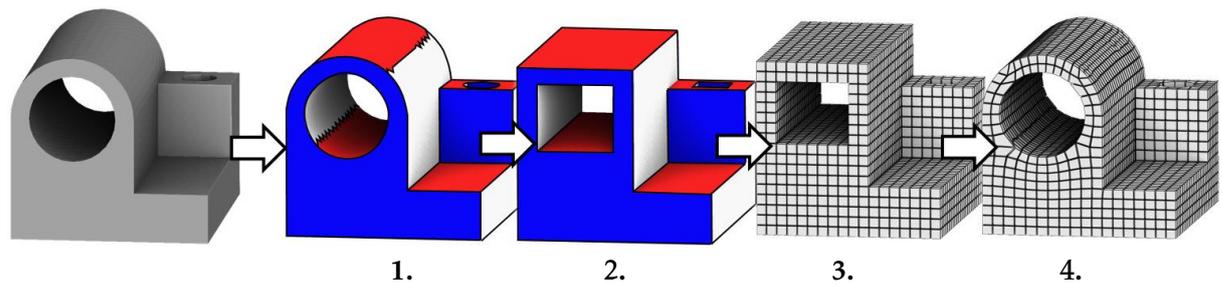


FIGURE 1.8 – Les étapes de la méthode des polycubes pour le remaillage hexaédrique. Le modèle de gauche est dans un premier temps coloré (1.) puis déformé (équivalent de l'intégration d'un champ de repère) (2.), avant d'être quantifié (3.) pour enfin subir la déformation inverse et obtenir le maillage hexaédrique conforme au bord souhaité (4.). *Image issue de [Protais, 2022].*

Approche 2,5D

Cette thèse propose une approche intermédiaire entre la méthode des champs de repères et la méthode polycube, dans le sens où nous restreignons l'espace des graphes de singularités possibles, mais en autorisant cette fois, des singularités internes. L'idée est d'orienter toutes les singularités le long d'une direction stable. De cette manière, le graphe de singularité est nécessairement valide. Cela peut être fait en quadrangulant une portion du bord du domaine puis en extrudant cette quadrangulation à tout l'intérieur du domaine pour former un maillage hexaédrique. Les singularités internes peuvent ainsi permettre d'obtenir des éléments de meilleure qualité. De plus, le défi de cette approche est reporté à celui du remaillage quadrangulaire d'une surface, ce qui est nettement mieux maîtrisé que la génération de champs de repères 3D. C'est ce pont entre la 2D et la 3D qui donne son nom de 2,5D à l'approche que nous proposons.

Le prochain chapitre (2) illustre cette approche sur des modèles géologiques qui peuvent très bien être modélisés comme des domaines 2D extrudés dans le sens de la verticalité. Ce chapitre permet ainsi de profiter directement des nouvelles avancées en remaillage quadrangulaire pour pouvoir remailler hexaédriquement des modèles géologiques. Les deux chapitres qui suivent présentent chacun de nouvelles méthodes pour réaliser certaines étapes du remaillage quadrangulaire par champs de repères. Le chapitre 3 présente une méthode de génération de champs de repères dont les directions ne sont pas nécessairement orthogonales. La non-orthogonalité s'avère très pratique voir essentielle pour le remaillage de modèles CAO et même le remaillage de modèles géologiques lorsque que des failles s'intersectent en formant un angle très aigu entre elles. Enfin le chapitre 4 présente une nouvelle méthode de quantification 2D se voulant plus simple que l'état de l'art [Campen et al., 2015].

Chapitre 2

Maillage hexaédriques 2,5D pour la simulation de réservoirs [Lopez et al., 2024]

Dans ce chapitre, nous proposons une nouvelle méthode de génération de maillages hexaédriques pour la simulation de réservoirs. Pour effectuer ces simulations, il est nécessaire de mailler des domaines de terrain délimités par un pavé droit. Le terrain, lui, est composé de plusieurs strates de roches sédimentaires. Chaque strate possède des caractéristiques propres et est séparée des autres par des surfaces appelées horizons. Enfin, ces domaines peuvent également contenir des failles, qui sont des structures tectoniques consistant en une surface divisant le volume rocheux en deux compartiments qui ont glissé l'un par rapport à l'autre (entraînant une discontinuité des horizons). Pour mailler de tels domaines, nous utilisons l'idée de la 2,5D en extrudant un maillage quadrangulaire obtenu par paramétrisation globale. Les hexaèdres construits sont alignés à certaines surfaces fournies en entrées (les bords du domaine, les failles et les horizons), rendant possibles de multiples simulations physiques sur le même maillage. Ce chapitre a donné lieu à une publication dans le journal *Mathematical Geosciences* :

2.1 Introduction

Les simulations de réservoir actuelles reposent généralement sur la définition d'une grille volumique sur laquelle il est possible de discrétiser les équations mathématiques décrivant la physique. Les grilles composées d'éléments hexaédriques sont souvent préférées, car les trois axes s'alignent naturellement avec la colonne stratigraphique et avec les deux directions principales du réseau de failles. De plus, elles permettent un calcul précis du flux entre les cellules à l'aide de la méthode des volumes finis [Eymard et al., 2000], sous réserve d'une bonne orthogonalité et d'une faible distorsion des cellules.

L'un des types de grilles hexaédriques les plus couramment utilisés pour les études de réservoir est une grille 2D sans singularités, extrudée le long de splines partant de chaque sommet de la grille 2D, et appelées piliers [Mallet, 2002, Fremming, 2002]. Une façon de produire de telles grilles (voir Fig. 2.1) consiste à définir un seul élément hexaédrique délimité par les frontières du domaine, puis à le découper en fines tranches dans les trois directions. Afin d'être alignées avec les failles et horizons, les coupes sont définies par les isovaleurs d'un champ scalaire dans chaque direction (I , J et K). Ces champs scalaires sont définis en deux étapes. Dans

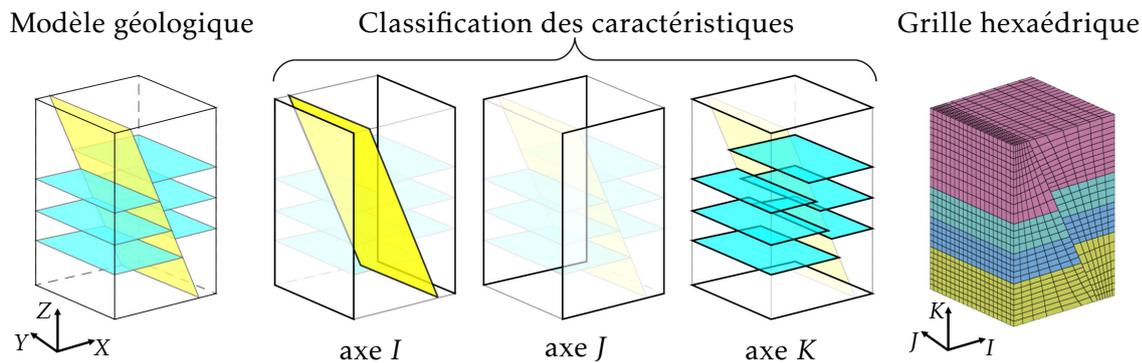


FIGURE 2.1 – Une grille hexaédrique structurée classique dans la littérature.

la première étape, les failles sont assignées aux directions I ou J en fonction de leur orientation, tandis que les horizons sont associés à K . De même, les bords sont associés à la direction correspondante à leur normale. Il peut arriver qu'une faille ne puisse être assignée à une direction. La grille produite n'est alors pas alignée avec la faille, rendant ensuite la simulation physique plus laborieuse. Dans la deuxième étape, les champs scalaires I, J et K sont calculés en résolvant un problème d'optimisation numérique : trouver un champ scalaire lisse qui soit constant sur chaque faille/horizon/bord lui étant associé, et continu partout, sauf le long des failles pour la direction K . Ces problèmes d'optimisation impliquent généralement la résolution d'un système linéaire pour chaque direction [Wellmann and Caumon, 2018]. Les coupes finales sont définies comme des isovaleurs des champs scalaires I, J, K pour les valeurs correspondantes à celles associées aux failles/horizons/bords, ainsi que certaines valeurs intermédiaires si une résolution plus élevée est souhaitée (Fig. 2.1–droite).

Les fonctions scalaires I, J, K calculées peuvent être utilisées pour établir une paramétrisation bijective du volume, permettant la création d'une grille dans l'espace paramétrique. Cette carte fait naturellement correspondre les cellules hexaédriques à un système d'indexation global. Chaque cellule hexaédrique peut ainsi être indexée par un triplet d'entiers (i, j, k) . Cette structure est prise en charge par la plupart des simulateurs d'écoulement disponible dans le commerce. Son empreinte mémoire est faible et sa simplicité rend les calculs efficaces. Le fait que la géométrie des failles soit honorée rend également cette grille intéressante pour les simulations de couplage fluide/géomécanique.

Définition du problème

Lorsqu'un réseau de failles devient plus complexe, avec des intersections non triviales entre les failles et les bords du domaine (Fig. 2.2), il peut parfois être impossible d'attribuer une direction I ou J à chacune de ces surfaces. Dans de tels cas, la technique simple décrite ci-dessus ne peut pas être appliquée pour construire une grille hexaédrique alignée avec les surfaces. Les méthodes existantes basées sur les piliers nécessitent généralement une simplification de la structure du réseau de failles ou une approximation de sa géométrie à l'aide d'une représentation en escalier (Fig. 2.2–droite).

Notre objectif est de développer une méthode de génération de maillages hexaédriques par extrusion, qui honore la géométrie de toutes les failles, horizons et bords. Cette approche offre la possibilité d'obtenir une discrétisation purement hexaédrique sur un plus grand ensemble de réseaux de failles 2,5D.

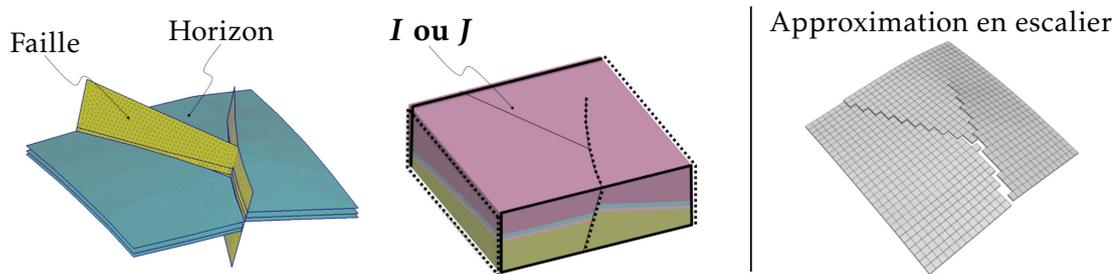


FIGURE 2.2 – *Gauche* : Un réseau de failles non trivial qui ne peut être maillé à l’aide d’une grille régulière classique. Les lignes pointillées représentent les surfaces sur lesquelles le champ I doit être constant, tandis que les lignes épaisses correspondent au champ J . On remarque que la dernière faille ne peut être assignée ni à I ni à J sans partager sa valeur avec une autre surface. Par conséquent, une partie du modèle doit nécessairement s’affaisser et devenir dégénérée. *Droite* : Une approximation en escalier.

Travaux précédents

Diverses méthodes ont déjà été proposées pour résoudre le problème de la génération d’une grille hexaédrique sur des réseaux de failles non triviaux. Bien que ces méthodes présentent des idées intéressantes, aucune d’entre elles ne permet d’obtenir une grille entièrement hexaédrique conforme à la géométrie de toutes les failles.

- *Entièrement hexaédrique* : La méthode proposée par [Mallet, 2002, Fremming, 2002] implique l’utilisation de l’approximation des failles en escaliers horizontaux pour la modélisation de réseaux de failles non triviales. Cette idée a été étendue dans des méthodes telles que [Hoffman et al., 2003, Gringarten et al., 2009] pour approximer les failles non verticales en utilisant des marches d’escalier verticales.

Inconvénients : Lors du couplage de simulations d’écoulement avec des simulations mécaniques, l’approximation en escalier n’est pas adaptée en raison de la nécessité d’une représentation lisse de la géométrie pour satisfaire les conditions mécaniques aux bords. Par conséquent, il est souvent nécessaire d’utiliser deux grilles distinctes : une grille en escalier pour la simulation de l’écoulement et un maillage tétraédrique ou d’autres grilles non structurées avec une approximation lisse de la géométrie des failles pour la simulation mécanique. Ce processus de couplage implique le transfert des propriétés mécaniques et d’écoulement entre les deux représentations différentes, ce qui peut introduire des erreurs et consommer beaucoup de temps et de ressources mémoires.

- *Partiellement hexaédrique* : Une autre approche consiste à mélanger une majorité d’éléments hexaédriques loin des failles avec des éléments polyédriques arbitraires proches des failles [Mallison et al., 2013, Santoshini et al., 2018]. Dans certains cas, les éléments non hexaédriques sont restreints à certains types spécifiques, comme les tétraèdres ou les pyramides [Gringarten et al., 2017]. Cette restriction peut permettre d’optimiser les simulations. D’autres travaux proposent de combiner les éléments d’un maillage tétraédrique pour former des hexaèdres et obtenir un nouveau maillage à dominance hexaédrique, mais contenant également des tétraèdres résiduels non contraints à être proches des failles [Sokolov et al., 2016, Botella et al., 2016].

Inconvénients : Les maillages construits de cette manière ne sont pas compatibles avec les simulateurs d’écoulement standards qui n’acceptent que les grilles hexaédriques structurées, qui sont encore couramment utilisées. En outre, la présence d’éléments tels

que des tétraèdres ou des pyramides peut dégrader l'orthogonalité des cellules, ce qui conduit à des approximations plus importantes lors de l'utilisation de schémas tels que l'approximation de flux à deux points, pour la simulation de l'écoulement.

- *Polyédrique* : Une autre option consiste à utiliser des grilles non structurées, telles que des diagrammes de Voronoï, également appelés grilles PEBI (*PErpendicular BIsector*) en géologie pétrolière [Palagi and Aziz, 1994, Merland et al., 2011, Klemetsdal et al., 2017], pour les simulateurs qui peuvent les gérer.

Inconvénients : Outre les défauts des grilles partiellement hexaédriques, il convient de noter que les maillages polyédriques sont encore plus gourmands en mémoire. En effet, la topologie et la géométrie des cellules peuvent varier considérablement et ne sont pas prédéfinies.

Contributions

Dans les approches fondées sur les piliers, le domaine est divisé en une série de couches constituées d'hexaèdres. Chaque couche représente une déformation d'une grille régulière bi-dimensionnelle épaisse (Fig. 2.3–gauche). Dans la solution que nous proposons, nous utilisons également une pile de couches, mais avec une approche différente. Chaque couche est une expansion d'un maillage quadrangulaire (Fig. 2.3–droite). La principale différence réside dans l'introduction de sommets singuliers, c'est-à-dire de sommets dont la valence n'est pas égale à 4, ce qui permet au maillage d'être conforme à toutes les surfaces présentes dans le domaine (Fig. 2.4).

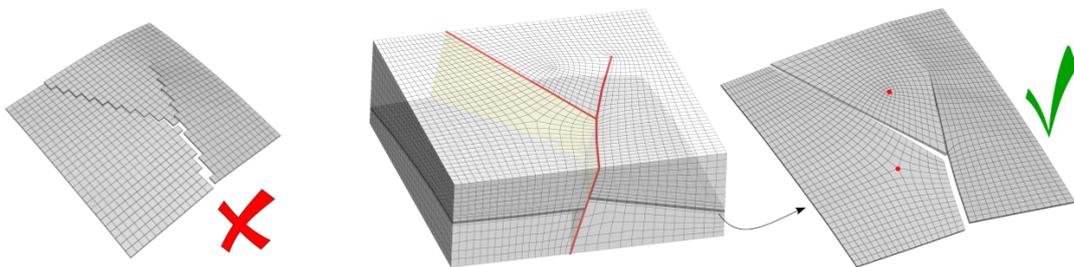


FIGURE 2.3 – Comparaison entre l'approche classique (à gauche) et notre solution (à droite). La nôtre introduit des singularités (en rouge) et parvient à s'aligner aux failles et limites du domaine.

Notre contribution peut être résumée par deux aspects principaux :

- Nous proposons une méthode de génération de maillages hexaédriques par extrusion (§2.2). Cette méthode comprend un algorithme de construction de couches stratigraphiques composées de cellules hexaédriques à l'aide d'algorithmes de l'état de l'art en remaillage quadrangulaire par paramétrisation globale. Elle suppose que les failles présentes dans le modèle géologique sont verticales.
- Nous introduisons un processus appelé verticalisation qui permet la déformation du modèle géologique d'entrée, entraînant l'alignement vertical des failles. Cette avancée permet de généraliser la méthode de construction de maillages hexaédriques pour prendre en compte les failles non verticales (§2.3).

Un premier maillage tétraédrique est généré à partir d'un modèle B-Rep (Représentation par les Bords) à l'aide d'une triangulation de Delaunay contrainte [Si, 2015]. Les failles, les hori-

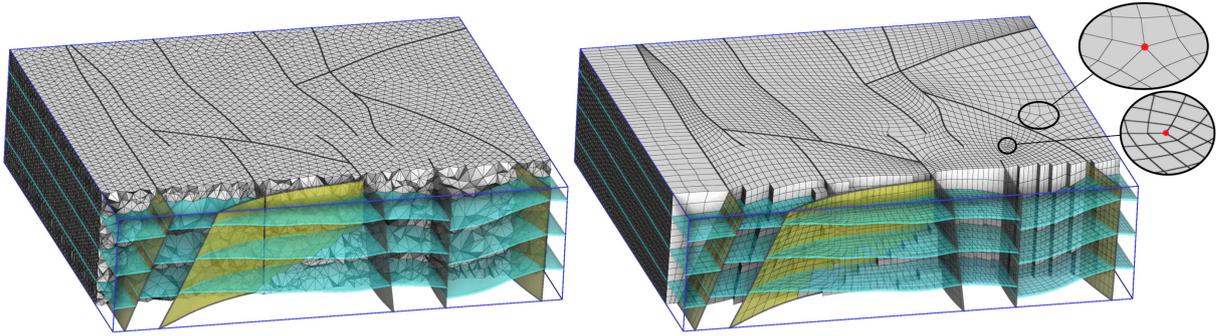


FIGURE 2.4 – À partir d’un modèle géologique (à gauche), notre méthode génère automatiquement un maillage entièrement hexaédrique aligné aux failles et horizons (à droite) au prix de l’ajout de singularités (gros plans).

zons et les limites du domaine servent de contraintes pendant le processus de génération du maillage. Ces contraintes sont associées à des faces triangulaires du maillage résultant, qui sont étiquetées en fonction de leur type de surface (voir la figure 2.5–en haut à gauche). C’est ce maillage tétraédrique qui sert d’entrée à notre méthode. La section 2.4 présente nos résultats ainsi que deux extensions de la méthode proposée : le raffinement local du maillage autour des puits et le long des failles. Enfin, dans la section 2.4.4, nous discutons de l’applicabilité de ces résultats aux simulations numériques.

Limitations : Notre méthode nous permet d’aborder une grande variété de problèmes tant qu’ils restent extrudables, c’est-à-dire exprimables selon une décomposition 2,5D. Cependant, les modèles comportant des failles en Y ou des dômes de sel sont exclus de notre étude. De plus, notre solution repose sur deux déformations successives qui ne permettent pas de gérer les configurations de plis renversés ou de failles quasi horizontales (Fig. 2.6). En d’autres termes, nous supposons qu’il est possible d’obtenir des failles verticales par déformation du modèle. Si ce n’est pas le cas, nous recommandons à l’utilisateur de supprimer un sous-ensemble de contraintes d’alignement des failles.

2.2 Méthode principale

Avant de commencer à travailler avec des ensembles de données réalistes, cette section se concentre sur les cas où toutes les failles sont verticales. L’objectif est de générer un maillage hexaédrique dont la structure combinatoire est plus complexe qu’une simple grille régulière bidimensionnelle, tout en respectant les horizons. Le processus commence par le calcul d’un maillage 2D dans le plan XY (§2.2.1) et le découpage subséquent du modèle en couches (§2.2.2). Chaque couche est ensuite remplie d’hexaèdres (§2.2.3) par la propagation du maillage 2D.

2.2.1 Calcul du maillage quadrangulaire dans le plan XY

Notre objectif est de générer un maillage quadrangulaire 2D qui s’aligne avec la projection des failles sur le plan XY . Pour ce faire, nous utilisons les outils de maillage quadrangulaire existants qui reposent sur des techniques de paramétrisation global. Tout d’abord, nous convertissons et simplifions notre problème pour fournir une entrée appropriée à l’algorithme de remaillage quadrangulaire (Fig. 2.5–en haut à droite). Ensuite, nous générons un maillage quadrangulaire (Fig. 2.5–en bas à droite) qui satisfait les contraintes spécifiées en entrée.

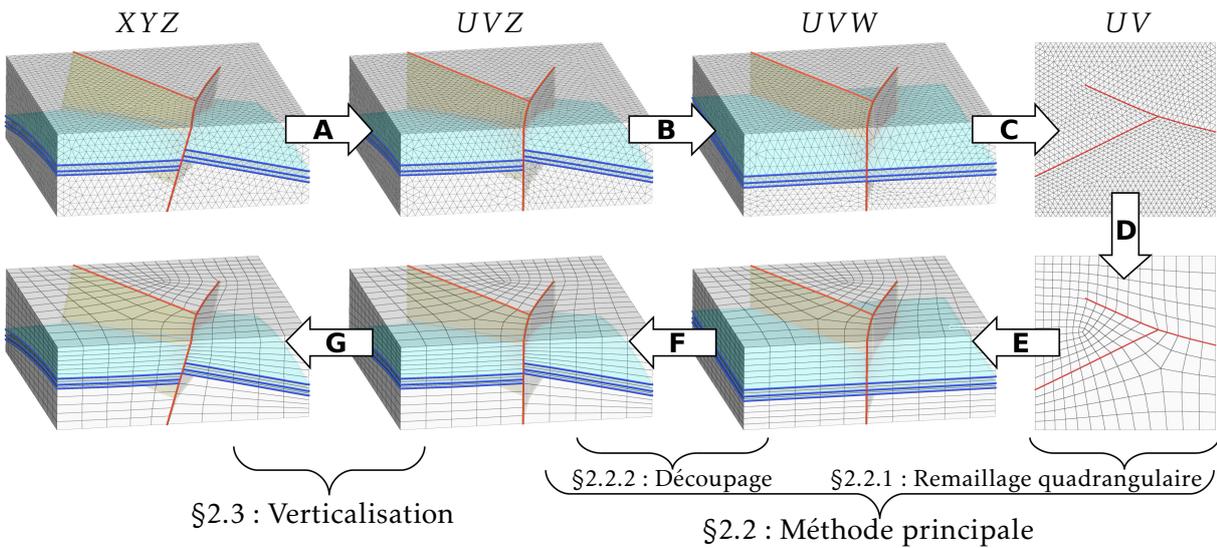


FIGURE 2.5 – Aperçu de la méthode : (A) déformation du modèle pour obtenir des failles verticales, (B) déformation du modèle pour obtenir des horizons horizontaux, (C) création d’un problème de remaillage quadrangulaire à partir d’une vue de dessus, (D) génération d’un maillage quadrangulaire, (E) extrusion dans la direction W pour générer un maillage hexaédrique, (F) déformation inverse $UVW \rightarrow UVZ$, et (G) déformation inverse $UVZ \rightarrow XYZ$.

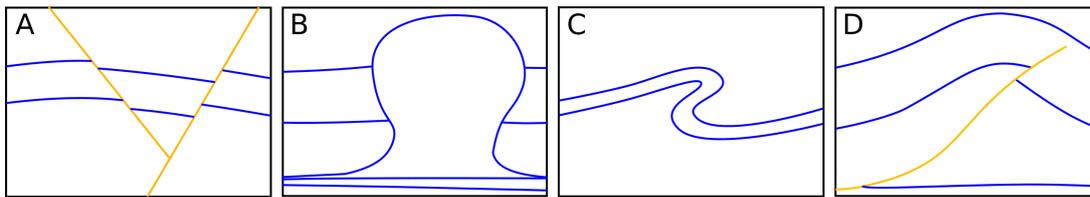


FIGURE 2.6 – Vues en coupe de structures géologiques qui n’entrent pas dans le cadre de l’étude : faille en Y (A), dôme de sel (B), pli renversé (C) et faille quasi horizontale (D).

Génération du problème 2D

Les failles sont représentées par une collection de triangles correspondant aux facettes de tétraèdres. Dans cette section, nous supposons que les failles sont verticales et que leurs projections sur le plan XY sont des segments. Par conséquent, la projection d’une faille sur le plan XY peut être représentée comme un ensemble de segments connectés. Sur la base de cette observation, nous calculons une entrée appropriée au remaillage quadrangulaire en suivant les étapes suivantes :

- Tous les sommets du maillage tétraédrique sont projetés sur le plan XY .
- Les paires de sommets qui correspondent à une arête du dessous d’une faille sont reliées pour former des contraintes de la future triangulation (Fig. 2.7–gauche).
- Le graphe obtenu est ensuite nettoyé en contractant les arêtes qui sont plus courtes que la longueur d’arête désirée pour le maillage hexaédrique final. La préservation des extrémités des courbes est imposée durant le processus.
- Une triangulation de Delaunay contrainte par les arêtes est générée et affinée pour atteindre une densité de maillage souhaitée (Fig. 2.7–milieu).

Ce processus produit l’entrée type requise pour les algorithmes de remaillage quadrangulaire

par paramétrisation globale, à savoir un maillage triangulaire dont certaines arêtes sont marquées comme des caractéristiques.

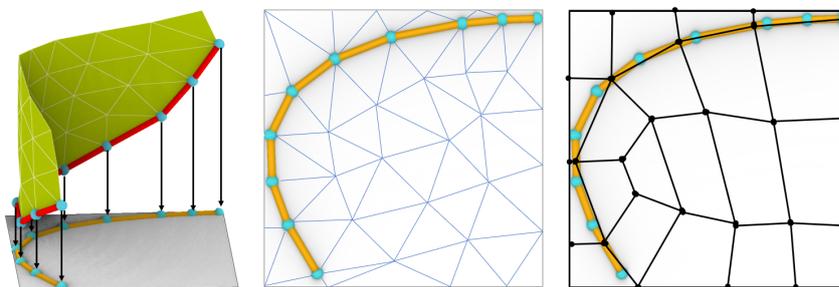


FIGURE 2.7 – À Gauche : les arêtes inférieures (rouge) de la faille (jaune) sont projetées sur le plan XY (gris) pour produire les contraintes 2D (orange). Au milieu : une triangulation du domaine XY avec les contraintes (orange) situées sur les arêtes. À droite : un maillage quadrangulaire conforme aux contraintes.

Génération du maillage quadrangulaire

Pour le remaillage quadrangulaire, nous avons opté pour une approche par paramétrisation globale [Kälberer et al., 2007, Bommès et al., 2009]. Dans un premier temps, nous calculons un champ de repères indiquant l'orientation que devront avoir les quadrilatères. Beaucoup d'algorithmes existent pour construire des champs de repères [Vaxman et al., 2016], mais puisque nos arêtes caractéristiques peuvent se toucher avec des angles très aigus, nous sommes amenés à choisir un algorithme qui gère correctement ces cas difficiles. Le chapitre 3 de cette thèse propose une telle méthode. Toutefois, en pratique, nous préférons utiliser une autre méthode également développée au sein de l'équipe [Desobry et al., 2021b]. Une carte sans coutures est ensuite calculée en minimisant l'énergie de [Garanzha et al., 2021]. Puis cette carte est finalement quantifiée grâce à l'algorithme décrit dans le chapitre 4, pour obtenir une carte préservant la grille. Enfin, le maillage quadrangulaire est extrait depuis cette dernière carte en utilisant l'algorithme proposé par [Ebke et al., 2013] (Fig. 2.7–droite).

2.2.2 Découpage du modèle dans la direction Z

Le maillage hexaédrique que nous cherchons à générer est constitué de couches empilées d'hexaèdres. Chaque couche suit la structure du maillage quadrangulaire obtenu dans la section précédente, extrudée d'une épaisseur d'un hexaèdre. Nous devons maintenant déterminer comment découper le modèle en couches d'hexaèdres. Le défi consiste à s'assurer que les horizons sont positionnés précisément à l'interface entre deux couches adjacentes. Pour représenter les couches, nous utilisons une fonction scalaire linéaire par morceaux \mathcal{W} définie sur le modèle. Cette fonction est caractérisée par ses valeurs w_i attribuées à chaque sommet i . Chaque couche est alors définie comme la partie du modèle où \mathcal{W} se situent dans un intervalle entre deux valeurs entières consécutives.

Pour calculer \mathcal{W} , les horizons sont disposés dans un ordre stratigraphique et des nombres entiers leur sont attribués en fonction de leur profondeur stratigraphique de façon strictement monotone. Tous les sommets d'un même horizon h sont contraints à avoir la même valeur W_h . La fonction \mathcal{W} est alors définie comme un champ harmonique interpolant ces contraintes à

l'ensemble du domaine. Le modèle est ensuite décomposé en le coupant le long des ensembles de niveaux à valeurs entières. En pratique, \mathcal{W} est obtenu comme solution du problème suivant :

$$\begin{cases} \min \sum_{(i,j) \in \mathcal{E}} c_{ij} \|w_i - w_j\|^2 \\ w_i = W_h \quad \text{si } i \text{ appartient à l'horizon } h \end{cases} \quad (2.1)$$

Où $(i, j) \in \mathcal{E}$ si et seulement si les sommets i et j sont reliés par une arête. Durant nos expérimentations, nous avons utilisé $c_{ij} = 1$, car nos maillages tétraédriques d'entrée étaient plutôt isotropes. Cependant, pour des maillages anisotropes, il est recommandé d'utiliser les poids cotangents résultant de la discrétisation de l'opérateur de Laplace-Beltrami [Crane, 2019].

Prétraitement

Discontinuités de \mathcal{W} : Pour tenir compte des discontinuités de l'horizon à travers les failles, une étape de prétraitement est effectuée pour déconnecter le maillage tétraédrique le long des failles. Cette étape consiste à diviser chaque sommet de faille en deux nouveaux sommets, un de chaque côté de la faille. En conséquence, le champ \mathcal{W} est représenté par sa valeur w_i sur chaque sommet i du maillage tétraédrique modifié, ce qui permet de représenter les discontinuités de l'horizon à travers les failles.

Discordances stratigraphiques : Un prétraitement est également nécessaire dans le cas de discordances stratigraphiques causées par des successions d'érosions et sédimentations. Il est en effet nécessaire de modéliser les parties manquantes des couches sédimentaires à la hauteur des discordances. Pour ce faire, le maillage d'entrée est complété de tétraèdres de volume nul jusqu'à ce que chaque couche sépare effectivement le domaine d'intérêt en deux compartiments distincts (Fig. 2.8).

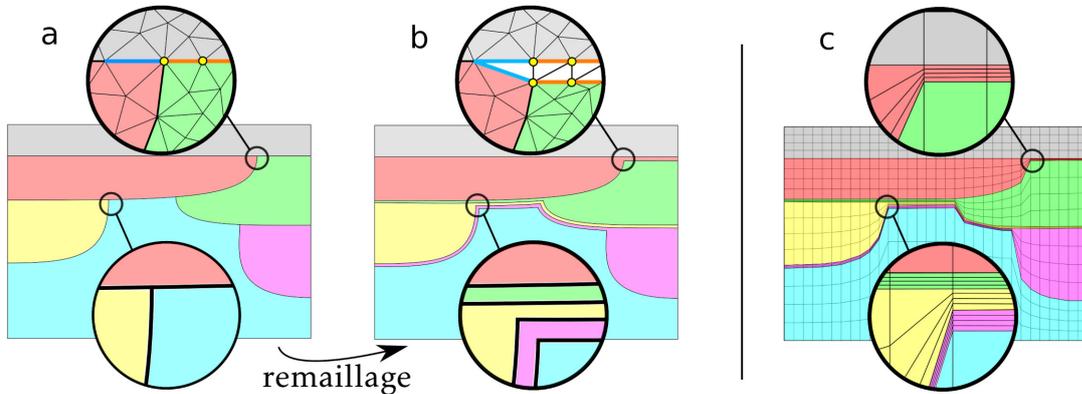


FIGURE 2.8 – Les phénomènes d'érosion et de sédimentation conduisent à des intersections d'horizons (a). Des tétraèdres supplémentaires de volume nul sont insérés pour faire disparaître la discontinuité (b). Le maillage hexaédrique résultant est conforme, il contient donc également des cellules de volume nul (c).

2.2.3 Génération du maillage hexaédrique

Le maillage final est obtenu en élevant le maillage quadrangulaire calculé dans le plan XY à chaque ensemble de niveau à valeur entière de \mathcal{W} et en générant des hexaèdres (dans l'espace

XYZ) entre chaque paire d'ensembles de niveau successifs. L'élévation est réalisée en projetant chaque sommet du maillage quadrangulaire sur l'ensemble de niveau $\mathcal{W} = w$. La projection d'un sommet de coordonnées (x, y) est obtenue via un lancer de rayon traversant le modèle verticalement (selon la direction Z/W). Les tétraèdres traversés par le rayon sont parcourus de bas en haut jusqu'à trouver la cellule contenant le point (x, y, w) . Une fois ce point trouvé, la carte locale du tétraèdre nous permet de retrouver ses coordonnées (x, y, z) . Un sommet du maillage hexaédrique est alors placé à cette dernière position dans l'espace XYZ .

Toutefois, un problème se pose pour les sommets situés sur une faille. En effet, comme ils ont été dédoublés, plusieurs sommets peuvent se trouver à une même position (x, y) mais doivent être projetés sur des valeurs de Z différentes. Un lancer de rayon attribuerait la même valeur de Z à deux sommets se trouvant à une même position (x, y) . La valeur obtenue pour l'un des deux sommets serait alors nécessairement fautive. Pour s'assurer de calculer la bonne valeur de Z pour un sommet i de coordonnées (x, y) situé sur une faille, nous utilisons l'algorithme suivant :

1. Calculer tous les tétraèdres \mathcal{T}_i qui intersectent une petite sphère centrée en (x, y, w) dans l'espace XYW .
2. Trouver le tétraèdre t_q qui contient la projection du barycentre d'un quadrilatère q incident à i sur l'ensemble de niveau $\mathcal{W} = w$. Cette projection est calculée par lancer de rayon.
3. Sélectionner le tétraèdre $t \in \mathcal{T}_i$ qui est le plus proche de t_q . La distance entre les tétraèdres t et t_q est définie ici comme le nombre minimal de tétraèdres sur un chemin entre t et t_q qui ne traverse aucune faille.
4. Calculer les coordonnées (x, y, z) associées au point (x, y, w) grâce à la carte locale de t_q .

La figure 2.9 illustre l'algorithme dans un espace 2D, XZ . Le concept clé de cet algorithme est de supposer que la projection du barycentre d'un quadrilatère q incident à i est situé du bon côté de la faille dans le maillage tétraédrique. Par conséquent, la valeur Z doit être évaluée à partir d'un tétraèdre situé de ce côté de la faille. Nous sélectionnons ce tétraèdre en choisissant celui qui peut atteindre le barycentre de q sans traverser la faille et avec le chemin le plus court. Par défaut, nous fixons le rayon de la petite sphère à 10% de la taille désirée des arêtes des hexaèdres, ce qui est significativement plus grand que l'erreur géométrique maximale causée par la verticalité faible (décrite dans la prochaine section).

2.3 Généralisation au failles non verticales

L'algorithme décrit jusque-là suppose que les failles sont verticales, ce qui n'est généralement pas le cas en pratique. Pour traiter les failles avec des géométries plus générales, nous proposons de représenter le maillage tétraédrique du modèle géologique dans un espace UVZ , où les failles sont verticales. Nous appliquons ensuite notre algorithme dans l'espace UVZ , puis appliquons la transformation inverse pour amener à nouveau le maillage hexaédrique dans l'espace XYZ du monde réel. Le défi consiste à définir la fonction de correspondance $\mathcal{U} : (x, y, z) \mapsto (u, v, z)$. Comme le modèle géologique d'entrée est un maillage tétraédrique et que notre algorithme opère sur des maillages tétraédriques, la fonction \mathcal{U} est naturellement linéaire à l'intérieur de chaque tétraèdre. Elle peut être représentée par l'image U_i de chaque sommet i . À noter que la déformation n'implique pas de déplacements dans la direction Z ; il suffit de déplacer des points dans les dimensions X et Y pour obtenir la verticalisation des failles.

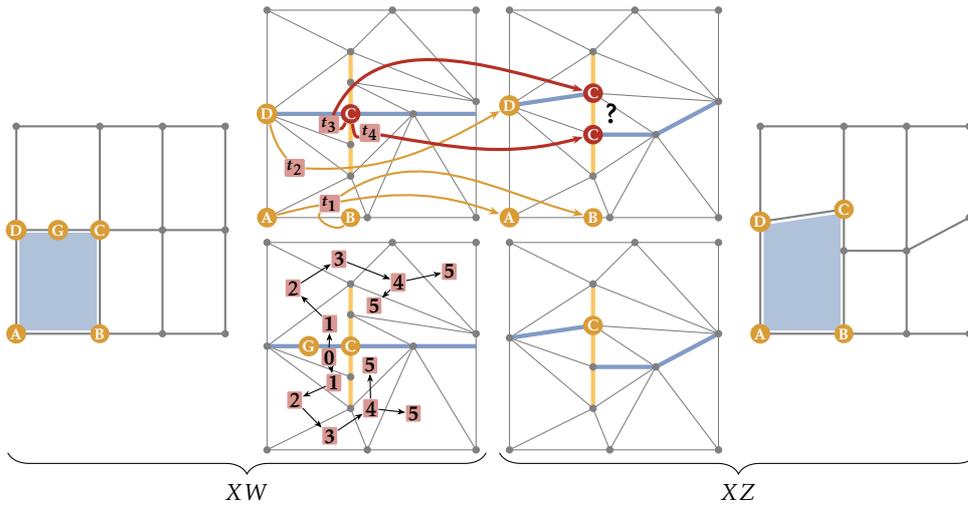


FIGURE 2.9 – Génération du maillage illustrée dans un espace 2D XZ . Le modèle d'entrée (troisième colonne) consiste en un maillage avec une faille verticale (jaune) et un horizon (bleu). En remplaçant la coordonnée Z par le champ \mathcal{W} (deuxième colonne), l'ensemble de niveau entier correspondant à l'horizon devient horizontal, ce qui permet d'obtenir une grille alignée aux axes dans l'espace XW (à gauche). En appliquant la transformation $XW \rightarrow XZ$, on obtient le remaillage quadrangulaire du modèle (à droite). Considérons les 4 coins (A, B, C, D) d'un quadrilatère. Les sommets A, B et D sont situés sur les triangles t_1, t_1 et t_2 respectivement (milieu gauche), et leur projection est déterminée par les cartes linéaires de leurs triangles respectifs (colonnes du milieu en haut). En revanche, le sommet C peut être projeté sur différentes valeurs de Z en fonction de s'il est projeté depuis le triangle t_3 ou t_4 . Pour déterminer la projection à utiliser, nous trouvons le triangle (ici t_3) qui contient le centre G d'une arête incidente (CD) et calculons un plus court chemin pour atteindre chacun des autres triangles (colonnes du milieu en bas). Le triangle le plus proche (ici, t_3 , avec une distance nulle) est choisi pour définir la projection du coin C .

2.3.1 Formulation avec des failles faiblement verticales

Dans notre contexte spécifique, une faille est définie comme un sous-ensemble de triangles à l'intérieur d'un maillage tétraédrique. Il serait naturel de considérer une faille comme verticale si le vecteur $(0, 0, 1)$ appartient à son plan tangent en tout point. Cependant, en utilisant cette définition, la projection de la faille sur le plan UV devient une ligne polygonale, chaque triangle étant projeté sur un seul segment de la ligne polygonale. Comme le montre la figure 2.10, cette contrainte a pour effet de regrouper les sommets en lignes verticales.

Il est important de noter que notre objectif est de calculer une déformation \mathcal{U} du maillage d'entrée sans altérer sa structure combinatoire. Lorsque nous rendons les failles verticales, nous souhaitons uniquement manipuler les positions des sommets. Par conséquent, imposer une verticalité stricte des failles conduirait probablement à une distorsion importante du maillage.

Pour répondre à cette préoccupation, nous proposons une définition moins forte de la verticalité. Selon cette définition, une faille est représentée par une ligne polygonale dans le plan UV qui satisfait aux conditions suivantes : chaque sommet de la faille est projeté sur la ligne polygonale, et la ligne polygonale elle-même correspond à la projection des arêtes inférieures. Ici, le terme « arêtes inférieures » fait référence aux arêtes du bord de la faille qui sont situées sous la faille.

En utilisant cette définition plus faible de la verticalité des failles, nous pouvons formuler notre problème comme étant de trouver \mathcal{U} avec une distorsion minimale, sous réserve de la condition suivante pour chaque sommet i appartenant à une faille :

$$U_i = \lambda_i U_{org(i)} + (1 - \lambda_i) U_{dest(i)} \quad (2.2)$$

Ici, λ_i prend des valeurs dans l'intervalle $[0, 1]$, et le sommet i est projeté sur le segment dont les extrémités sont les sommets $org(i)$ et $dest(i)$, aux coordonnées barycentriques $(\lambda_i, 1 - \lambda_i)$ (Fig. 2.11).

Pour assurer l'accessibilité de ce problème, nous déterminons d'abord $org(i)$, $dest(i)$, et λ_i (Section 2.3.2) en utilisant une approche heuristique qui approxime « les courbes qui seront transformées en lignes verticales (dans UVZ) » avec les lignes de courant d'un champ de vecteurs lisse. Ensuite (dans la section 2.3.3), nous calculons la carte \mathcal{U} avec une distorsion minimale, tout en considérant les contraintes imposées par les failles faiblement verticales définies par les valeurs de $org(i)$, $dest(i)$, et λ_i (2.2). La carte résultante \mathcal{U} est continue, linéaire à l'intérieur de chaque tétraèdre, et idéalement bijective. Elle peut donc être utilisée pour déformer le maillage tétraédrique d'entrée (via \mathcal{U}) et le maillage hexaédrique obtenu (via \mathcal{U}^{-1}).

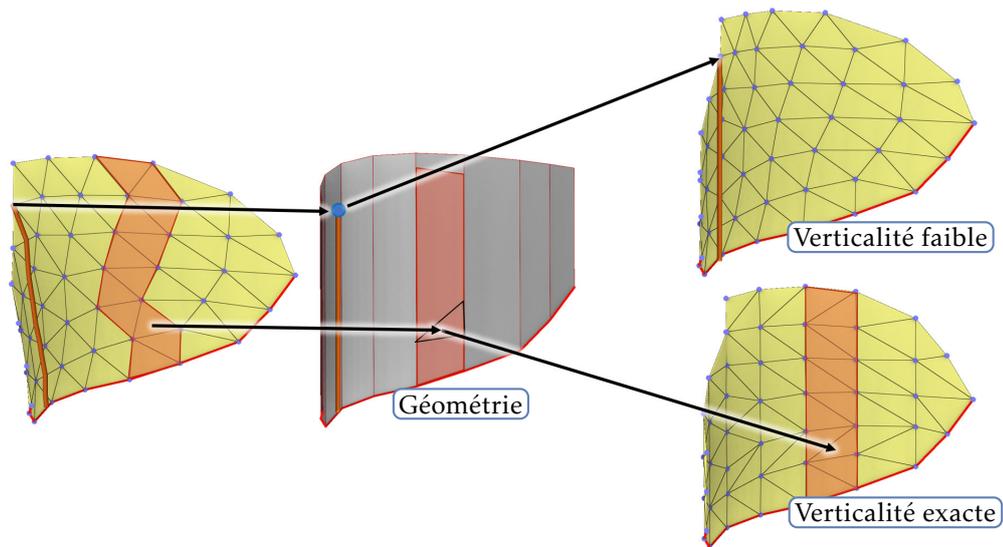


FIGURE 2.10 – Verticalité faible ou exacte. Pour rendre une faille (à gauche) verticale sans déplacer ses bords inférieurs (en rouge), la surface finale doit être située à l'intérieur de la surface grise (au milieu). La verticalité faible (en haut à droite) oblige seulement les sommets à être situés sur la géométrie verticale. La verticalité exacte (en bas à droite) oblige les triangles à être situés sur la géométrie verticale. Même dans cet exemple jouet, la verticalité exacte génère beaucoup de distorsion dans le maillage.

2.3.2 Déterminer $org(i)$, $dest(i)$ et λ_i

Idéalement, $org(i)$, $dest(i)$ et λ_i devraient être choisis pour minimiser la distorsion de U . Malheureusement, $org(i)$ et $dest(i)$ sont des variables qualitatives qu'il est très difficile d'optimiser par des méthodes numériques. Pour cette raison, nous préférons les déterminer à l'aide d'une approche heuristique. À cette fin, nous définissons un champ de vecteurs « gravité »

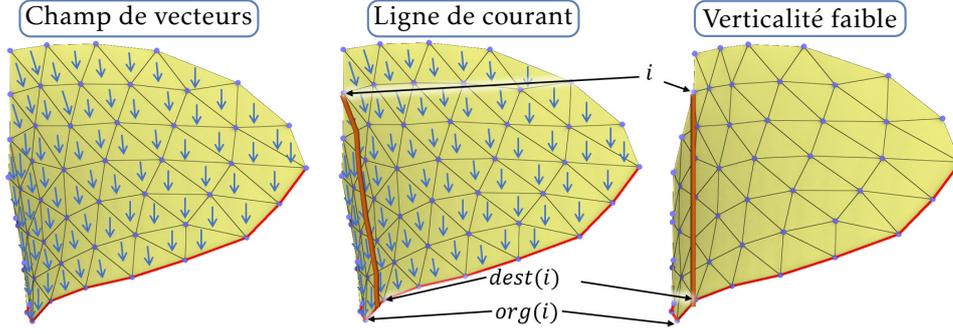


FIGURE 2.11 – Algorithme de verticalisation faible : nous calculons un champ de vecteurs 3D tangent aux failles (une restriction à la faille est donnée à gauche), nous traçons des lignes de courant à partir de chaque sommet i de la faille (milieu) pour déterminer où il quitte la faille ($org(i), dest(i), \lambda_i$), et nous trouvons une déformation minimale du modèle telle que les extrémités de la ligne de courant deviennent alignées dans la coordonnée W .

G dans l'espace XYZ . G est tangent aux failles, de sorte que pour chaque ligne de courant partant d'un sommet de faille i , la ligne de courant sort de la faille par une arête inférieure ($org(i), dest(i)$), en un point $H_i = \lambda_i X_{org(i)} + (1 - \lambda_i) X_{dest(i)}$.

Le champ de vecteurs G est constant sur chaque tétraèdre et est obtenu comme la solution du problème d'optimisation suivant :

$$\left\{ \begin{array}{l} \min \sum_{\text{tétraèdres adjacents } t, t'} \|G_t - G_{t'}\|^2 \\ G_t \cdot (0, 0, 1) = -1 \quad \text{pour tout tétraèdre } t \\ G_t \cdot n(f) = 0 \quad \text{si } t \text{ a une facette de faille } f \end{array} \right. \quad (2.3)$$

où G_t est la valeur constante du champ G sur le tétraèdre t , et $n(f)$ est la normale du triangle f . Cette formulation est basée sur les observations suivantes :

- Nous utilisons les lignes de courant de G pour approximer la préimage des lignes verticales (dans UVZ). Par conséquent, nous pensons que la minimisation de la variation de G à cette étape permettra de minimiser la distorsion de \mathcal{U} à la fin. Comme l'illustre la figure 2.12, il est important de définir G partout dans le volume, même si on ne l'utilise que pour tracer les lignes de courant situées sur les failles. En effet, cela permet de lisser le champ entre deux failles proches qui ne se touchent pas.
- Fixer la coordonnée Z de G à -1 est un moyen simple de forcer les lignes de courant à traverser le modèle de haut en bas. Comme la norme de G n'affecte pas les lignes de courant, cette contrainte ne fait que forcer le champ à pointer vers le bas.
- Le fait que le champ soit orthogonal au vecteur normal de la faille garantit simplement que les lignes de courant suivront la faille et s'en échapperont par un bord de la faille (plus précisément, un bord inférieur).

Ce problème d'optimisation est facile à résoudre, car il s'agit d'une fonction d'énergie quadratique soumise à des contraintes linéaires. Comme proposé dans [Bommes et al., 2012], nous pouvons transformer le problème en un problème sans contraintes en éliminant certaines variables. Cette transformation conduit à un problème standard de moindres carrés, qui est équivalent à la résolution d'un système d'équations linéaires. Une fois le champ de vecteurs G calculé, nous pouvons déplacer chaque sommet de faille i le long de la faille en suivant G

jusqu'à ce qu'il atteigne l'arête inférieure ($org(i), dest(i)$) avec des coordonnées barycentriques $\lambda_i, (1 - \lambda_i)$.

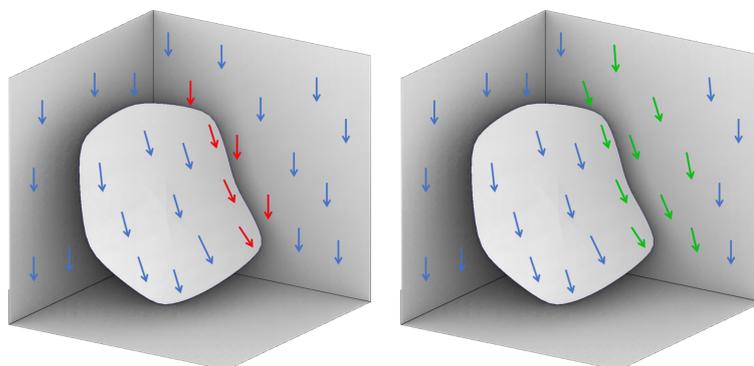


FIGURE 2.12 – Si le champ vectoriel G est optimisé uniquement sur les failles/frontières (à gauche), le maillage sera fortement déformé entre elles. C'est pourquoi nous préférons l'optimiser partout (à droite).

2.3.3 Optimisation de la déformation \mathcal{U}

La déformation \mathcal{U} est représentée par les coordonnées des sommets U_i et est linéaire sur chaque tétraèdre. Comme pour le champ de vecteurs G , nous obtenons \mathcal{U} en résolvant un problème d'optimisation :

$$\left\{ \begin{array}{l} \min \sum_t \|\nabla U_t - I_3\|^2 \\ (\lambda_i U_{org(i)} + (1 - \lambda_i) U_{dest(i)} - U_i) \cdot (1, 0, 0) = 0 \\ (\lambda_i U_{org(i)} + (1 - \lambda_i) U_{dest(i)} - U_i) \cdot (0, 1, 0) = 0 \end{array} \right. \quad (2.4)$$

Dans cette formulation, ∇U_t représente la matrice jacobienne de \mathcal{U} sur le tétraèdre t , I_3 est la matrice identité 3×3 , et $i, org(i)$ et $dest(i)$ sont des sommets. L'objectif du problème d'optimisation est de minimiser la distorsion de la carte tout en permettant une translation du domaine. Les contraintes de verticalité garantissent que les sommets de faille sont projetés sur leurs arêtes inférieures respectives, mais la géométrie des arêtes inférieures n'est pas prédéterminée. Elle est déterminée par le processus d'optimisation. Comme dans la section précédente, cette optimisation a une fonction objective quadratique et des contraintes linéaires. Le problème peut donc être résolu en le réduisant à un système d'équations linéaires.

2.4 Résultats

Dans cette section, nous évaluons les performances de notre algorithme à l'aide d'un ensemble de modèles synthétiques qui reflètent les difficultés rencontrées dans les données de terrain réelles.

La figure 2.13 présente les résultats de notre méthode sur des scénarios de référence provenant de [Pellerin et al., 2015]. Suite au processus de remaillage, les failles et les horizons sont représentés comme des sous-ensembles des faces des hexaèdres, formant des maillages quadrangulaires continus. Par conséquent, les limites des failles sont maintenant approximées par

des marches d'escalier (Fig. 2.13–bas). Comme prévu, les horizons du maillage hexaédrique se conforment avec précision aux horizons triangulés de part et d'autre des discontinuités verticales.

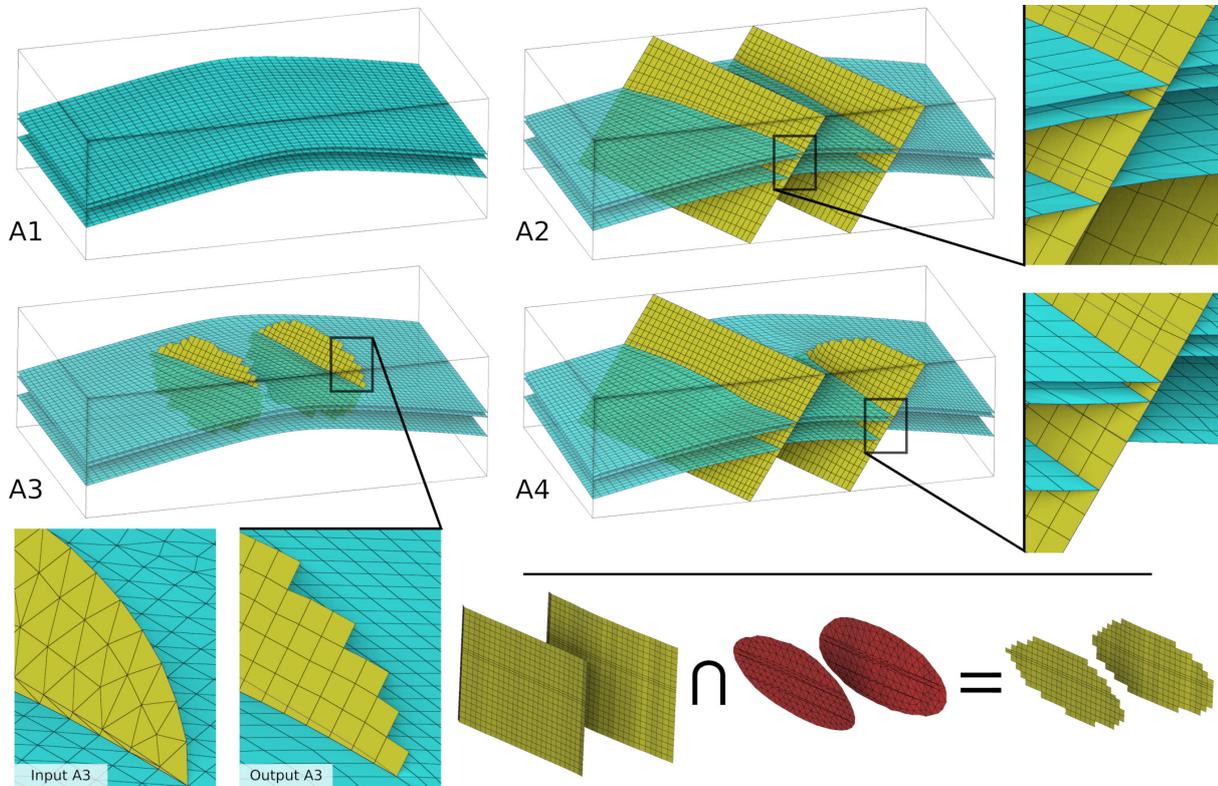


FIGURE 2.13 – Modèles de référence utilisés dans [Pellerin et al., 2015]. Les caractéristiques sont bien préservées (gros plan à droite). Les failles mourantes sont obtenues en traçant un rayon passant par le barycentre de chaque quadrilatère candidat et ayant sa normale comme direction, puis en vérifiant que ce rayon intersecte la faille triangulée (en bas).

Pour les discordances stratigraphiques, notre approche conforme crée des hexaèdres de volume nul qui représentent la partie « manquante » de chaque couche sédimentaire (Fig. 2.14). Cette représentation des « cellules mortes » répond aux contraintes d'entrée des simulateurs d'écoulement standard et peut être facilement supprimée si nécessaire.

Pour démontrer l'efficacité de notre méthode, nous avons effectué d'autres tests sur des réseaux de failles avec des intersections complexes entre les failles et les limites du domaine (Fig. 2.15). Grâce à la robustesse de notre algorithme de remaillage quadrangulaire, nous pouvons traiter des configurations de failles complexes au prix d'un petit nombre de sommets singuliers.

Nous fournissons également des exemples qui explorent les capacités de notre méthode de remaillage quadrangulaire avec des contraintes supplémentaires pour améliorer localement le maillage (Fig. 2.16) et l'affiner davantage (§2.4.2).

En outre, nous démontrons la robustesse du calcul de la déformation pendant l'étape de verticalisation en utilisant notre définition de la verticalité faible, en particulier sur un modèle avec une faille listrique (§2.4.1).

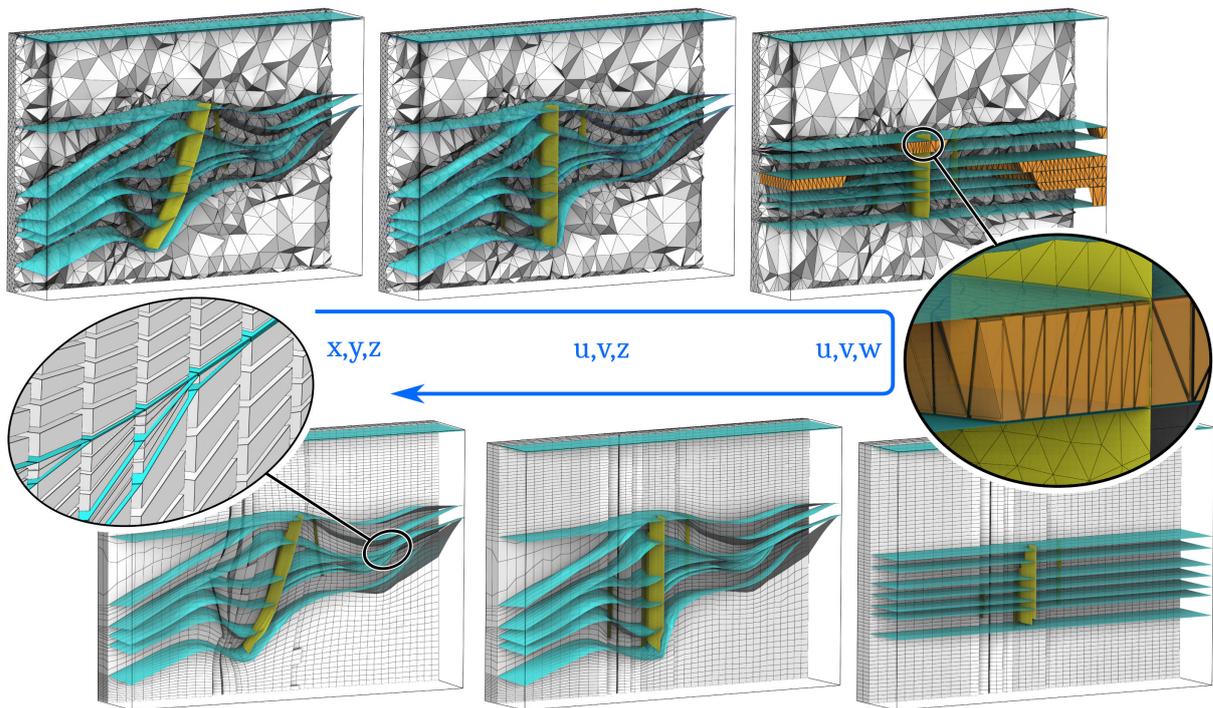


FIGURE 2.14 – Les discordances stratigraphiques données en entrée (érosion, en haut à gauche) sont compensées par l’ajout de cellules : elles permettent de calculer \mathcal{W} (gros plan à droite). Une fois la déformation inversée (en bas à gauche), le volume des hexaèdres correspondants tend vers zéro.

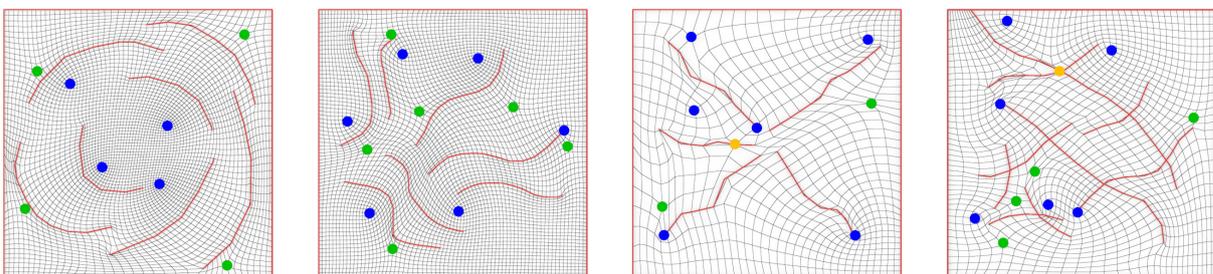


FIGURE 2.15 – Maillages quadrangulaires obtenus sur des tests avec des réseaux de failles générés artificiellement. Les points singuliers sont mis en évidence en couleur en fonction de leur valence.

2.4.1 Robustesse et efficacité

La déformation \mathcal{U} est obtenue en minimisant la distorsion sous des contraintes de verticalité faible imposées par les failles. Bien que cette approche produise une déformation valide sans tétraèdres retournés dans la plupart des cas, elle ne fournit aucune garantie. En effet, pour un test critique avec une faille listrique (Fig. 2.17), une déformation extrême est nécessaire pour atteindre une verticalité faible, ce qui entraîne des tétraèdres retournés. Pour résoudre ce problème, un algorithme de déformation plus robuste proposé par [Garanzha et al., 2021] peut être utilisé. Toutefois, en présence de failles quasi horizontales, la décomposition hexaédrique obtenue avec notre approche tend à être de mauvaise qualité.

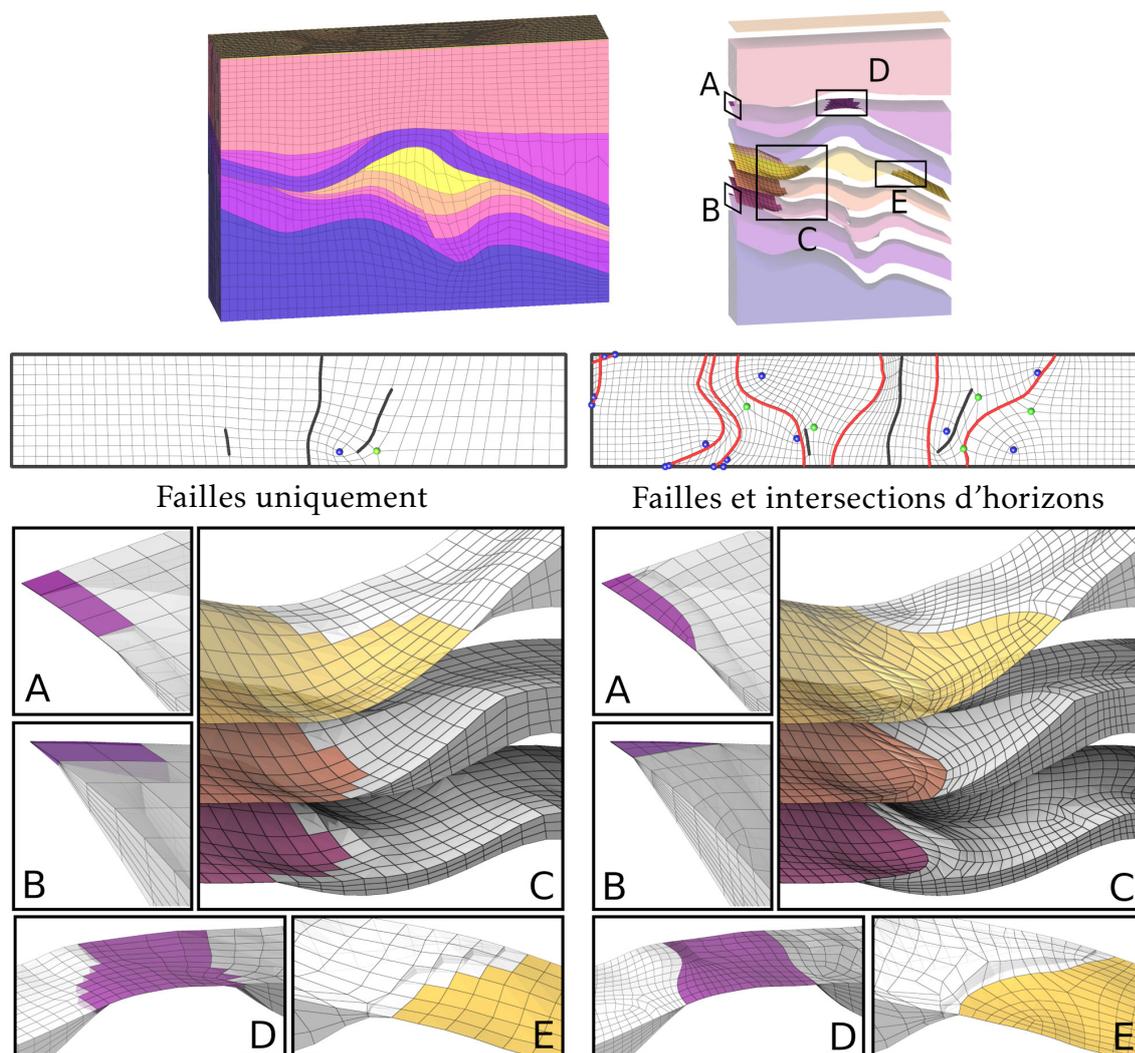


FIGURE 2.16 – Pour tester le remaillage quadrangulaire et améliorer l’approximation géométrique, nous essayons d’aligner les cellules avec les intersections d’horizons (jusqu’à présent approximées par des marches d’escalier, à gauche). Pour ce faire, nous introduisons un deuxième ensemble de courbes caractéristiques en entrée de l’algorithme de remaillage quadrangulaire (à droite). Le maillage quadrangulaire qui en résulte est plus complexe (20 points singuliers au lieu de 2), mais il permet de créer des éléments hexaédriques qui se conforment précisément aux courbes d’intersection des horizons (gros plans).

Les temps de calcul de notre approche sont raisonnables. Les résultats présentés ici ont été obtenus en moins de deux minutes pour des maillages d’entrée et de sortie de l’ordre de 100 000 éléments en utilisant un ordinateur portable équipé d’un processeur i5-11500H fonctionnant à 2,90GHz. Pour des maillages plus importants avec un million d’éléments, le temps de calcul varie de 10 à 15 minutes. L’un des avantages de notre approche est que le nombre de surfaces caractéristiques n’a qu’une influence minimale sur le temps d’exécution. Cependant, la phase la plus critique du processus est le calcul de la correspondance entre les espaces XYZ et UVZ . Le temps d’exécution dépend principalement de l’ampleur des distorsions nécessaire pour cette correspondance. Les fortes distorsions requièrent souvent l’utilisation de méthode d’optimisa-

tion plus complexe, comme [Garanzha et al., 2021]. Par exemple, le test critique de la figure 2.17 nécessite près de 10 minutes de calcul pour un maillage d'entrée avec 160k tétraèdres (et 40k hexaèdres en sortie).

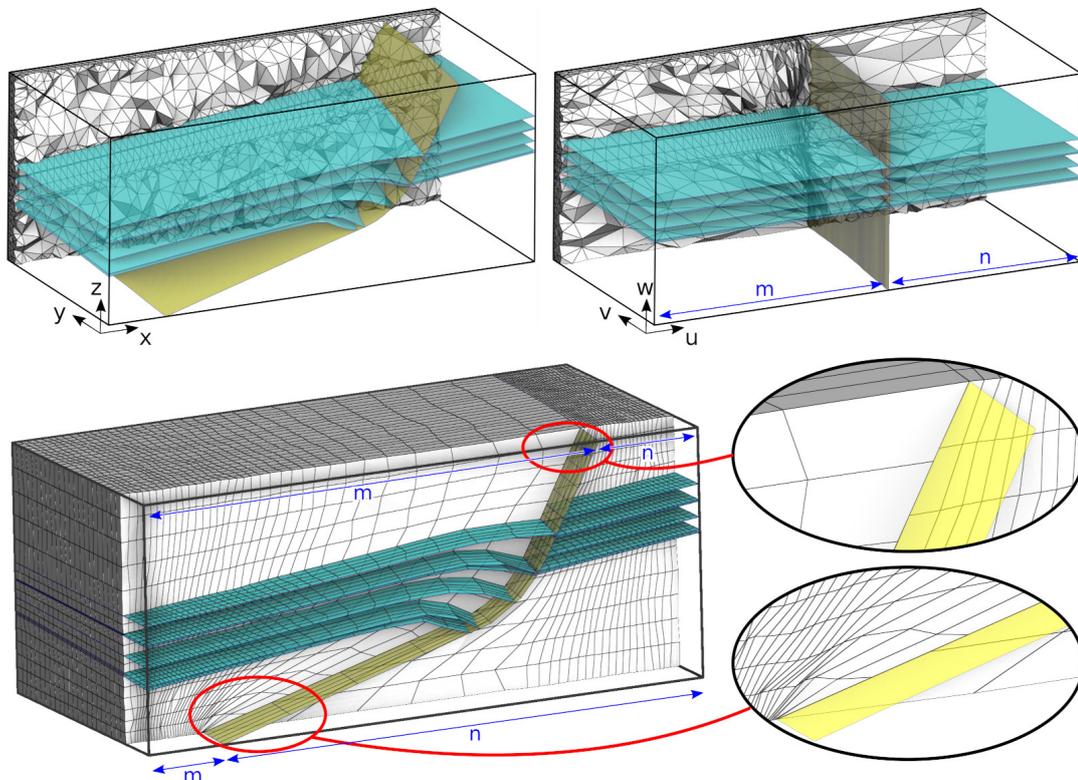


FIGURE 2.17 – Test de grande déformation : malgré une déformation importante (en haut), le maillage obtenu reste valide, sans aucun élément retourné. Bien qu'il y ait une variation notable dans la taille des éléments (gros plan), cette variation semble relativement douce. Notez que le résultat présenté ne tient pas compte de l'étape de lissage finale. Les valeurs de m et n indiquent le nombre d'éléments hexaédriques le long de l'axe X de part et d'autre de la faille.

2.4.2 Raffinement local

Parfois, une plus grande résolution de maillage peut être nécessaire à proximité de caractéristiques géologiques telles que des failles ou des objets artificiels tels que des puits de forage pour de meilleures études. Notre méthode permet d'y parvenir sans qu'il soit nécessaire d'affiner le maillage de manière globale.

Failles volumiques

Les failles géologiques ont souvent une certaine épaisseur, mais dans les grilles de réservoir, elles sont généralement représentées comme des surfaces. Cependant, dans les cas où les failles servent de couloirs d'écoulement, il est crucial de les représenter comme des entités volumétriques, ce qui permet de leur attribuer des attributs physiques plus réalistes. Notre algorithme répond à ce besoin en permettant de créer automatiquement des failles volumétriques dans le maillage. En pratique, nous y parvenons en introduisant des contraintes supplémen-

taires. Nous dupliquons la projection de la faille sur le plan UV et la déplaçons légèrement de part et d'autre de sa position initiale dans la direction normale (Fig. 2.18-b). En incorporant ces contraintes dans les données d'entrée, l'algorithme de maillage quadratique génère un maillage quadrangulaire avec une seule couche de quadrilatères entre chaque paire de projections de failles (Fig. 2.18-c). L'unique couche de quadrilatère peut être imposée durant l'étape de quantification expliquée plus en détail dans le chapitre 4. Après extrusion et application de la déformation inverse \mathcal{U}^{-1} , chaque faille correspond à une couche verticale lisse d'hexaèdres ayant une seule cellule d'épaisseur (Fig. 2.18-d).

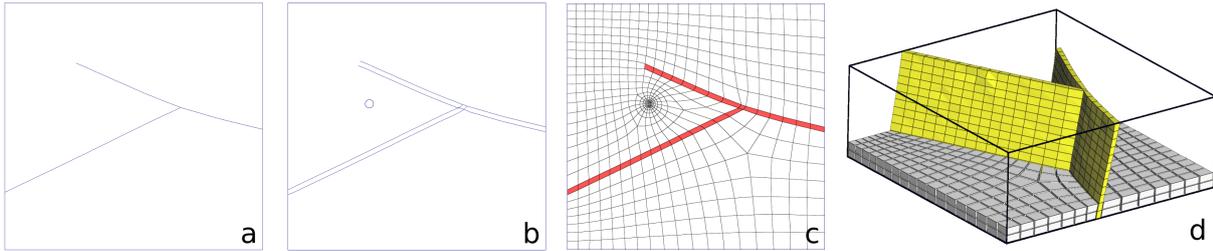


FIGURE 2.18 – De gauche à droite : les projections des failles sont dupliquées, donnant une représentation épaisse des bords inférieurs des failles. Après extrusion et application de \mathcal{U}^{-1} , on obtient une représentation volumétrique de la faille.

Puits de forage

En outre, il est important de tenir compte de la géométrie autour des puits et de procéder à des ajustements appropriés. Au lieu d'utiliser un seul point comme projection dans le plan UV , nous proposons d'utiliser une forme circulaire (Fig. 2.18-c). En utilisant une empreinte circulaire, le maillage résultant présentera une configuration plus fine autour du puits (figure 2.19).

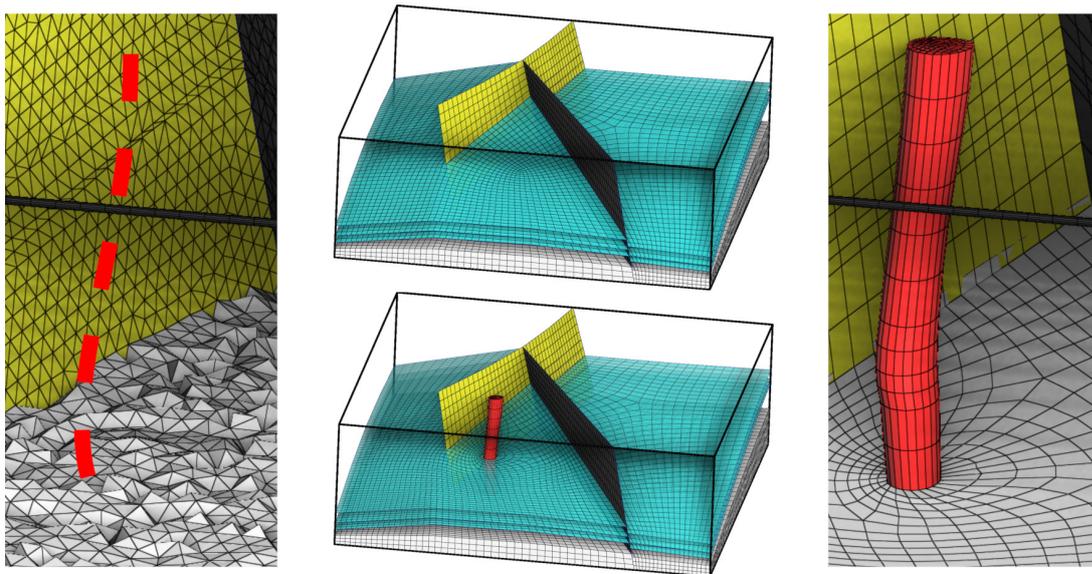


FIGURE 2.19 – De gauche à droite : un puits dans un maillage tétraédrique, un maillage hexaédrique classique et un maillage localement raffiné autour du puits (vue détaillée à droite).

2.4.3 Rétrocompatibilité avec les simulateurs historiques d'écoulement structurés

Il est important de noter que nos maillages hexaédriques sont semi-structurés et contiennent des arêtes singulières. Bien que ces maillages soient compatibles avec les simulateurs de grilles non structurées les plus récents, ils ne peuvent pas être facilement indexés à l'aide d'un simple tuple (i, j, k) dans un espace d'indexation unique.

Pour assurer la compatibilité avec les simulateurs d'écoulement structurés, nous proposons de partitionner notre maillage en sous-grilles structurées par blocs, comme le montre la figure 2.20—à gauche. Chaque sous-grille possède son propre espace d'indexation local, qui peut potentiellement se chevaucher avec d'autres sous-grilles. Afin de cartographier ces espaces d'indexation locaux dans un seul espace d'indexation global et de maintenir la compatibilité avec les simulateurs d'écoulement standards, il est important de minimiser le nombre d'index inoccupés dans l'espace d'indexation global (figure 2.20—droite). Ce problème d'empilement de pavés droits est connu pour être NP-complet [Korf, 2003], puisque le problème de *bin packing* s'y réduit. Pour y remédier, nous avons adopté une heuristique similaire à celle proposée dans [Lévy et al., 2002] pour l'empilement de textures.

Après le processus d'empilement des blocs, certaines paires d'hexaèdres qui sont géométriquement connectés se voient attribuer des indices (i, j, k) qui ne sont pas voisins immédiats dans l'espace d'indexation globale. Pour certains simulateurs d'écoulement standards, ces contacts géométriques doivent être explicitement définis comme des connexions de non-voisinage (Fig. 2.20—droite).

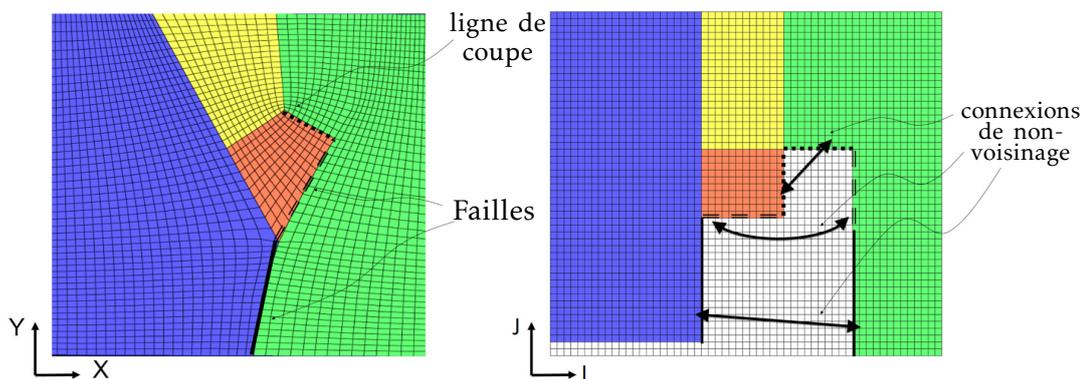


FIGURE 2.20 – Conversion d'un maillage quadrangulaire semi-structuré en une grille structurée.

2.4.4 Exemple d'application : simulation d'injection de fluide

Le maillage hexaédrique généré à l'aide de la méthode proposée est un excellent choix pour mener des études couplant écoulement et géomécanique, car il vérifie les critères des deux types de simulations.

La grille illustrée à la figure 2.5 est utilisée pour effectuer des simulations couplées d'injection de CO_2 à l'aide d'un simulateur à code ouvert [Gross and Mazuyer, 2021]. Le réservoir synthétique se compose de 8 couches de cellules et est situé entre deux couches de morts-terrains et de sous-terrains, respectivement. Au départ, l'ensemble du domaine est rempli d'eau et soumis à une pression initiale de $4 \cdot 10^6$ Pa. Pendant la simulation, nous injectons un fluide avec une fraction volumique CO_2/eau de 0.995/0.005 à une pression de 10^7 Pa. En ce qui

concerne les contraintes mécaniques, nous imposons une condition selon laquelle les bords du domaine ont un déplacement nul le long de l'axe parallèle à la normale de la surface pendant toute la durée de la simulation. Par exemple, les limites à $x = x_{min}$ et $x = x_{max}$, représentées par une surface dans le plan YZ, ont une condition de déplacement nul le long de l'axe X. Les propriétés détaillées se trouvent dans le tableau 2.1. Les résultats de la simulation, présentés dans la figure 2.21–gauche, montrent l'écoulement et les propriétés mécaniques. La saturation en CO₂ et la pression présentent une augmentation homogène autour du puits d'injection. La contrainte augmente également, avec des valeurs plus élevées observées dans les deux couches de couvertures en raison de leur module d'élasticité isostatique et de cisaillement plus élevé. Par conséquent, des valeurs de déplacement plus élevées sont observées dans la couche réservoir plus déformable, en particulier près de la zone du puits.

En outre, nous montrons des simulations similaires sur une grille avec un raffinement local autour du puits d'injection, comme décrit dans la section 2.4.2. Le schéma d'écoulement et la réponse mécanique sont similaires au cas sans raffinement. Le raffinement local fournit la résolution supplémentaire nécessaire pour étudier le comportement précis à proximité du puits, comme dans les études d'essais de puits, sans qu'il soit nécessaire de raffiner excessivement l'ensemble du maillage (Fig. 2.21–droite).

Régions	Couverture	Réservoir
Type de Roche	Schiste	Sable
Porosité	0.05	0.25
Perméabilité en X (m ²)	$1.00 \cdot 10^{-19}$	$1.00 \cdot 10^{-14}$
Perméabilité en Y (m ²)	$1.00 \cdot 10^{-19}$	$1.00 \cdot 10^{-14}$
Perméabilité en Z (m ²)	$1.00 \cdot 10^{-19}$	$1.00 \cdot 10^{-15}$
Module d'élasticité isostatique (Pa)	$1.00 \cdot 10^{10}$	$7.00 \cdot 10^8$
Module de cisaillement (Pa)	$1.60 \cdot 10^9$	$4.00 \cdot 10^8$
Pression initiale (Pa)	$4.00 \cdot 10^6$	
Transmissibilité des failles	0.001	

Temps de simulation	20 ans	
Pression du puits (Pa)		$1.0 \cdot 10^7$
CO ₂ injecté (t)		$4.5 \cdot 10^5$
Taux d'injection (t/an)		$2.25 \cdot 10^4$

TABLE 2.1 – Propriétés mécaniques et d'écoulement, suivit des résultats d'une simulation d'injection de CO₂ dans le réservoir.

Un deuxième modèle géologique avec un modèle de faille plus complexe (Fig. 2.15) est utilisé pour étudier l'injection de CO₂ le long des couloirs de faille. Nous présentons uniquement les résultats de la simulation de l'écoulement. La pression initiale et la composition du fluide sont identiques à travers le domaine. Initialement, le modèle est rempli d'eau dans un réservoir à une pression de $1.40 \cdot 10^7$ Pa. Les propriétés détaillées du simulateur sont spécifiées dans le tableau 2.2. Les résultats, présentés dans la figure 2.22, montrent la pression et la saturation en CO₂. Comme prévu, les failles, avec leur plus grande perméabilité, servent de corridor à l'écoulement du fluide injecté.

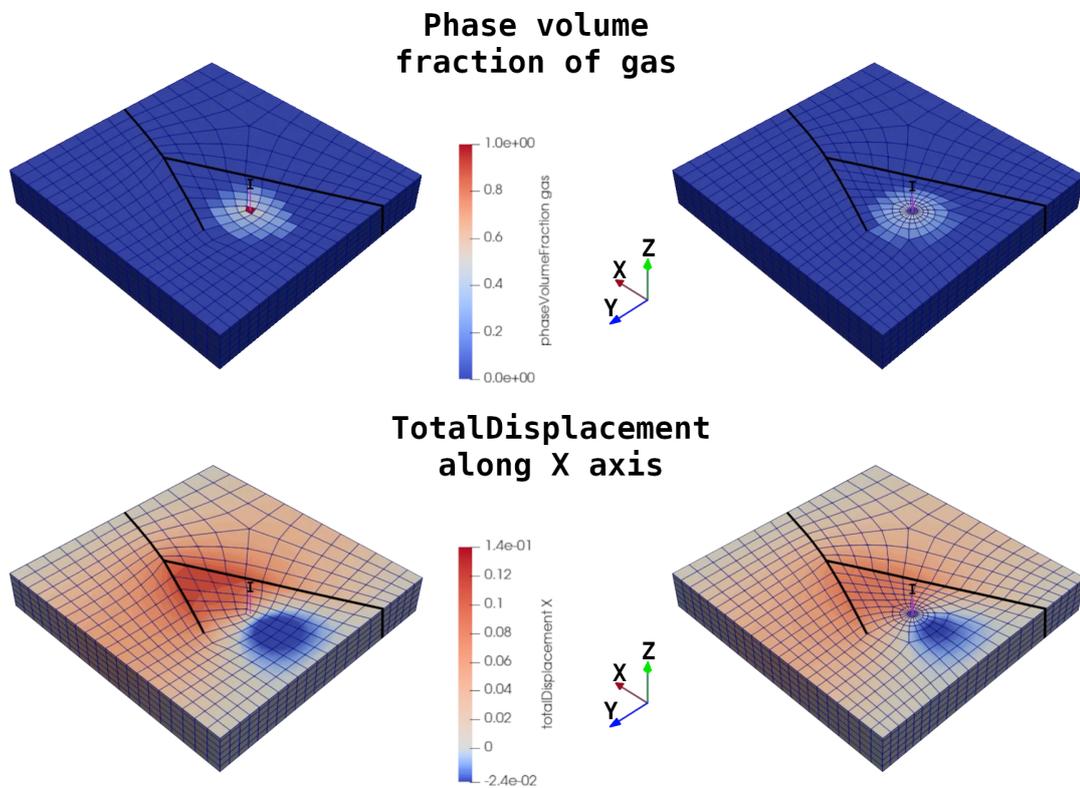


FIGURE 2.21 – Résultats de la simulation couplant écoulement et mécanique avec et sans raffinement local du puits. À noter que la vue est coupée au milieu dans la direction verticale.

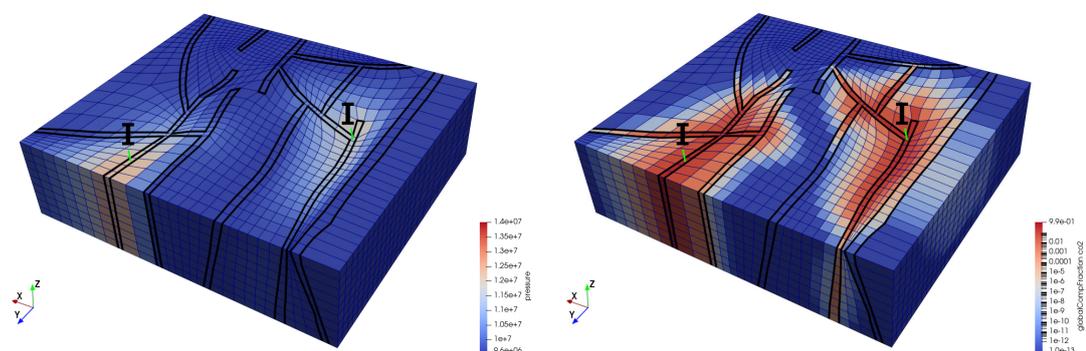


FIGURE 2.22 – Résultats d'une simulation d'injection de CO_2 à l'intérieur des failles servant de corridor.

Régions	Couverture	Réservoir
Type de Roche	Schiste	Sable
Porosité	0.05	0.20
Compressibilité (Pa^{-1})	$4.00 \cdot 10^{-10}$	$4.00 \cdot 10^{-7}$
Perméabilité en X (m^2)	$1.00 \cdot 10^{-15}$	$5.00 \cdot 10^{-14}$
Perméabilité en Y (m^2)	$1.00 \cdot 10^{-15}$	$5.00 \cdot 10^{-14}$
Perméabilité en Z (m^2)	$1.00 \cdot 10^{-15}$	$5.00 \cdot 10^{-14}$
Pression initiale (Pa)	$9.50 \cdot 10^6$	
Transmissibilité des failles	1.0	

Temps de simulation	10 ans	
Pression du puits (Pa)		$1.4 \cdot 10^7$
CO ₂ injecté (t)	$6.12 \cdot 10^7$	$3.77 \cdot 10^7$
Taux d'injection (t/an)	$5.36 \cdot 10^5$	$3.30 \cdot 10^6$

TABLE 2.2 – Propriétés d'écoulement, suivit des résultats d'une simulation d'injection de CO₂ dans le réservoir.

2.5 Limitations

Ce travail présente deux types de limitations : la première concerne la formulation de notre objectif, et la seconde les aspects algorithmiques pour atteindre cet objectif.

2.5.1 Formulation de l'objectif

Nos maillages hexaédriques sont utilisés pour discrétiser les caractéristiques physiques des modèles géologiques dans les simulations numériques. Alors que nous prenons en compte les critères classiques de qualité des maillages, tels que la déformation des cellules, nous devons également tenir compte des contraintes spécifiques à l'application qui jouent un rôle crucial dans la pertinence de nos résultats.

Dans notre implémentation actuelle, le maillage doit se conformer aux horizons. En outre, le maillage doit également être conforme aux failles, avec une coordonnée U ou V devant être constante le long de celles-ci. Nous autorisons la discontinuité de W à travers les failles, mais nous ne permettons pas de modéliser les déplacements latéraux des horizons de part et d'autre des failles. Si ces déplacements latéraux s'avèrent nécessaires, à l'avenir, nous pouvons facilement les autoriser en ne considérant les projections des failles non plus comme des courbes caractéristiques, mais comme des bords du domaine 2D à remailler avec des quadrilatères.

2.5.2 Limitations algorithmiques

De toute évidence, notre principale limitation est que le maillage final est contraint d'avoir une structure combinatoire d'un maillage quadrangulaire 2D extrudé dans la direction verticale. En pratique, la verticalisation des failles quasi horizontales peut produire un espace UVZ fortement déformé, ce qui se traduit par des maillages de mauvaise qualité, comme le montre la figure 2.17.

Notre méthode est également limitée par la robustesse des algorithmes de déformation de

l'espace XYZ en UVZ et de remaillage quadrangulaire. Heureusement, les travaux récents [Desobry et al., 2021b, Garanzha et al., 2021, Coudert-Osmont et al., 2023] que nous utilisons pour ces tâches sont assez robustes. De plus, les maillages tétraédriques que nous générerons en entrée ont des angles dièdres suffisamment bons (pas trop proches de 0 et π) pour éviter les retournements dans les déformations que nous calculons. La taille des tétraèdres, quant à elle, n'impacte pas la qualité du remaillage hexaédrique comme l'illustre la figure 2.14, permettant ainsi de correctement gérer les horizons proches les uns des autres.

Les limitations restantes sont dues aux différentes discrétisations des failles dans l'espace : un sous-ensemble de faces de tétraèdres et un sous-ensemble de faces d'hexaèdres. Pour les failles non mourantes et sans courbure dans l'espace UVW , la géométrie correspond parfaitement dans les deux représentations. Lorsqu'une faille possède de la courbure horizontale, celle-ci est généralement faible, mais engendre tout de même de légères différences géométriques entre les deux représentations.

2.6 Conclusion

Nous avons apporté une nouvelle approche pour la construction de maillages hexaédriques pour les réservoirs à l'aide d'une méthode fondée sur l'extrusion. La topologie de la grille extrudée est un maillage quadrangulaire généré par paramétrisation globale, ce qui permet au maillage de se conformer à toutes les courbes caractéristiques et tous les bords du domaine en introduisant des sommets singuliers. Cette méthode élargit considérablement la gamme des réseaux de failles qui peuvent être remaillés avec des cellules hexaédriques de manière structurée sans avoir recours à des approximations de failles en escalier.

Nous avons également pu montrer des résultats prometteurs de simulations couplant écoulement et géomécanique en utilisant un solveur open source conçu pour les grilles non structurées. En outre, notre maillage hexaédrique contenant des singularités peut aussi être utilisé dans les simulateurs d'écoulement standards après l'application d'un processus d'empilement (§2.4.3).

À l'avenir, nous prévoyons d'explorer diverses stratégies de raffinement local ou encore de réduction de résolution dans les régions éloignées des courbes caractéristiques, où la réduction du nombre de cellules peut améliorer l'efficacité des simulations.

Les deux prochains chapitres se concentrent sur des aspects précis de la génération de maillage quadrangulaire : le calcul d'un champ de repères, et la quantification de cartes sans coutures.

Chapitre 3

Champs de directions non orthogonales [Desobry et al., 2021a]

Ce chapitre présente les travaux réalisés pour la génération de champs de repères (2D et 3D). La plupart des méthodes de génération de champs de repères forçant l’orthogonalité des différentes directions ou à minima certaines symétries, nous explorons ici la possibilité de relaxer cette orthogonalité. Le contenu de ce chapitre a donné lieu à une publication dans le journal CAD et une présentation dans la conférence internationale SPM obtenant le second prix du meilleur article :

Desobry, D., Coudert-Osmont, Y., Corman, E., Ray, N., and Sokolov, D. (2021a). Designing 2d and 3d non-orthogonal frame fields. Computer-Aided Design, 139:103081

3.1 Introduction

Au sein de l’équipe Pixel nous nous sommes beaucoup intéressés au remaillage de modèles CAO. Pour ce type de modèle, il est fréquent de rencontrer des coins à angles aigus qui impliquent une forte non-conformité de la paramétrisation à construire. Malheureusement, la plupart des méthodes de construction de champs de repères impliquent une contrainte d’orthogonalité entre les différentes directions du champ empêchant alors d’obtenir les gradients des coordonnées de la paramétrisation souhaitée comme le montre la figure 3.5. Un champ de repères composés de n directions orthogonales dans un espace tangent de dimension n est appelé un champ de croix. Ces champs ont premièrement été introduits sur des surfaces pour faire des rendus non photoréalistes [Hertzmann and Zorin, 2000]. Puis ils ont été étendus à une classe plus grande de champ d’orientations (toujours 2D) avec des ensembles d’orientations invariants par rotation de $2\pi/N$ pour un N choisi [Palacios and Zhang, 2007, Ray et al., 2008]. Plusieurs travaux ont ensuite exploré d’autres façons de générer des champs de repères spécifiquement pour la génération de maillages quadrangulaires [Bommes et al., 2009], ou encore en redéfinissant la notion de champ lisse aboutissant à une énergie dont on est capable de calculer efficacement le minimum global [Knöppel et al., 2013]. Il existe une étude de l’état de l’art des générations de champs d’orientations datant de 2016 [Vaxman et al., 2016]. La section suivante présente quelques travaux qui ont été menés pour définir des champs d’orientations sans contraintes d’orthogonalité ou d’invariance par rotation.

3.1.1 Travaux précédents

En 2D, des travaux précédents ont commencé par relaxer cette contrainte d'orthogonalité en élaborant une nouvelle représentation des champs d'orientation. [Diamanti et al., 2014] propose de représenter un ensemble de vecteurs tangents (n-polyvecteur) $\{u_i\}_{1 \leq i \leq n} \in \mathbb{C}^n$ par le polynôme $P_u(z) = \prod_{i=1}^n (z - u_i)$ où les vecteurs sont individuellement représentés par des complexes dont les parties réelles imaginaires représentent les coordonnées du vecteur dans une base du plan tangent. Cette représentation permet de s'abstenir d'ordonner les vecteurs tangents, la multiplication complexe étant commutative. À noter qu'usuellement les orientations du champ d'orientations sont représentées par des vecteurs unitaires, mais dans cette représentation, les vecteurs peuvent avoir une norme quelconque. La distance entre deux n-polyvecteurs u et v est alors définie par la norme 2 de la différence des deux polynômes de représentation : $\|P_u - P_v\|_2$. Le problème de cette méthode est qu'elle ne se généralise pas aux variétés 3D, car il faudrait définir les polynômes de représentation sur une algèbre de nombres hypercomplexes non commutative. L'absence de commutativité fait alors perdre l'unicité du polynôme de représentation puisqu'il dépend de l'ordre des facteurs $(z - u_i)$. Une autre remarque que l'on peut soulever est la difficulté à donner un sens géométrique à la distance associée aux n-polyvecteurs, bien qu'en pratique elle soit très bonne. Dans le cas des 4-polyvecteurs représentant des repères, les auteurs donnent une interprétation géométrique. En revanche, dans le cas général, une telle interprétation est compliquée à trouver. La section 3.3.4 propose une comparaison avec notre méthode.

[Panozzo et al., 2014] s'appuie sur la décomposition polaire en proposant d'optimiser indépendamment un champ de repères orthogonaux qu'on qualifie généralement de champ de croix et un champ de matrices symétriques définies positives. Ces dernières sont utilisées pour déformer les directions du champ de croix de manière à obtenir des vecteurs tangents non orthogonaux et anisotropes. Étant donnée une croix de dimension 2, $\{u_1, -u_1, u_2, -u_2\}$, et une matrice symétrique définie positive W , il est possible d'obtenir un ensemble de directions non orthogonales et anisotropes $\{v_1 = Wu_1, -v_1, v_2 = Wu_2, -v_2\}$. Le champ de matrices de déformation W est calculé en interpolant harmoniquement et indépendamment les coefficients de la matrice en ayant fixé la matrice de déformation pour certaines faces. Contrairement à [Diamanti et al., 2014] cette méthode s'étend naturellement à la 3D lorsque l'on dispose d'une solution pour générer des champs de croix 3D.

Une approche assez similaire est donnée par [Jiang et al., 2015]. L'idée est de construire une nouvelle métrique sur une surface de telle sorte qu'un champ de croix pour cette nouvelle métrique satisfasse des contraintes prescrites en entrée. Puis un champ de croix est calculé en utilisant la nouvelle métrique et la connexion qu'elle induit. Les croix obtenues n'étant pas nécessairement des repères orthogonaux pour la métrique euclidienne, cette méthode permet de générer des champs de repères non orthogonaux. À nouveau, cette approche pourrait très bien s'étendre à la 3D.

[Liu et al., 2011] propose une méthode de génération de champs de directions conjuguées s'appuyant sur [Ray et al., 2008] et son optimisation de champs de croix dont les principales variables sont des angles. Deux vecteurs d'un plan tangent à une surface plongée dans \mathbb{R}^3 sont conjugués lorsqu'ils correspondent aux directions d'un quadrilatère planaire infiniment petit aligné à la surface. Bien que [Liu et al., 2011] présentent une méthode pour générer des champs de directions conjuguées, il suffit de supprimer une contrainte de leur optimisation pour produire des champs de repères non orthogonaux arbitraires. En revanche, le processus

d'optimisation proposé est non convexe, compliqué et ne semble à priori pas s'étendre naturellement à la 3D.

À notre connaissance, il n'existe pas de travaux au sujet de la conception de champs de repères 3D non orthogonaux.

3.1.2 Élaborer une méthode d'optimisation de champs d'orientation

Chaque tentative de définition d'une nouvelle méthode d'optimisation de champs d'orientation fait face à trois défis que l'on énumère ci-dessous en s'appuyant sur l'exemple d'une des premières publications sur les champs d'orientation 3D.

Le premier défi est de définir une représentation des champs d'orientation (un champ de polynômes, un couple métrique/champ de croix, ...). Ce choix est fortement impacté par la dimension, car certaines représentations ne s'étendent pas naturellement de la 2D à la 3D, notamment à cause de la non-commutativité du groupe $SO(3)$. Pour les maillages volumétriques on peut par exemple s'appuyer sur l'approche introduite dans [Huang et al., 2011] qui utilise les symétries des harmoniques sphériques pour des champs de croix. L'idée est de représenter une croix dont les branches sont alignées avec les colonnes d'une matrice de rotation $R \in SO(3)$ par une fonction f_R de la sphère S^2 de sorte que pour chacune des 24 symétries du cube M , on ait $f_{RM} = f_R$. Pour ce faire, la croix alignée avec les axes de l'espace euclidien est donnée par la fonction invariante par symétrie du cube :

$$h(t) = t_x^2 t_y^2 + t_y^2 t_z^2 + t_z^2 t_x^2 \quad (3.1)$$

La figure 3.1 propose une visualisation de cette fonction sur la sphère. Il est facile de vérifier que $h(M^T t) = h(t)$. Pour une croix arbitraire, la fonction de représentation est simplement une rotation de h donnée par :

$$f_R(s) = h(R^T s) \quad (3.2)$$

On peut alors vérifier que la propriété souhaitée est obtenue :

$$f_{RM}(s) = h(M^T R^T s) = h(R^T s) = f_R(s) \quad (3.3)$$

Il se trouve que ces fonctions sont décomposables dans la base orthonormée des harmoniques sphériques d'ordre 4 ($\{Y_4^m\}_{-4 \leq m \leq 4}$). Il est alors possible d'utiliser un vecteur $\hat{f}_R \in \mathbb{R}^9$ de dimension 9 comme représentation.

Le second challenge est de définir une notion de champ lisse. Pour cela, la tendance est de définir une énergie à minimiser. Le plus souvent, cette énergie s'appuie sur une distance entre deux ensembles d'orientations. Chez [Huang et al., 2011], la distance est celle associée à la norme L^2 sur $\mathcal{C}^0(S^2, \mathbb{R})$ ou \mathbb{R}^9 .

$$d(R_a, R_b)^2 = \|f_{R_b} - f_{R_a}\|_2^2 = \int_{S^2} (f_{R_b}(s) - f_{R_a}(s))^2 ds = \sum_{i=0}^8 (\hat{f}_{R_b}(i) - \hat{f}_{R_a}(i))^2 = \|\hat{f}_{R_b} - \hat{f}_{R_a}\|_2^2 \quad (3.4)$$

La décomposition dans la base des harmoniques sphériques prend alors tout son sens puisqu'elle permet d'obtenir une distance facilement calculable entre les croix.

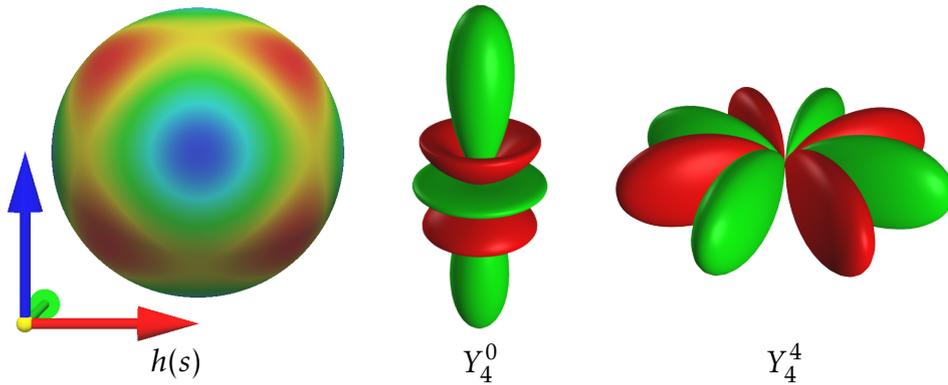


FIGURE 3.1 – Visualisation de la fonction h associée à la croix alignée avec les axes de l’espace euclidien (3.1). La couleur bleue indique de faibles valeurs (0) et le rouge des valeurs élevées (1/3). Si l’on omet la constante Y_0^0 qui n’affecte pas la distance entre deux croix dans la décomposition en harmoniques sphériques de h et à un facteur multiplicatif près on peut exprimer h comme $\sqrt{7}Y_4^0 + \sqrt{5}Y_4^4$. Les deux harmoniques qui interviennent sont représentées par des tracés sphériques où le rayon correspond à la valeur absolue de l’harmonique pour un couple d’angles polaire et azimut donné. La couleur, elle, indique le signe de l’harmonique (vert : positif, rouge : négatif). Images issues de [Huang et al., 2011]

Le dernier défi est de pouvoir être capable d’imposer des contraintes, comme le fait d’imposer une direction dans le champ d’orientation et de laisser les autres orientations libres. Il peut également être intéressant d’autoriser un certain contrôle sur le champ, comme son orthogonalité. Toujours pour l’exemple de [Huang et al., 2011], il est possible de fixer l’une des trois directions avec une contrainte linéaire qui repose sur un lemme démontré dans l’article qui postule qu’une croix est alignée avec l’axe z si et seulement si $\hat{f}(4)$ (le coefficient associée à l’harmonique Y_4^0) vaut $\sqrt{7}$.

3.1.3 Contributions

Notre contribution consiste à étendre la représentation des croix de [Huang et al., 2011] aux ensembles de directions non orthogonales pour des espaces tangents de dimension n toujours sous forme d’un polynôme de degré 4 sur l’hypersphère S^{n-1} . Notre méthode répond efficacement aux trois challenges de la section précédente. Dans notre publication, nous affirmons même que les méthodes de génération de champs non orthogonaux précédentes ne répondaient pas à ces trois points, notamment sur l’extension à la 3D. Mais toujours comme nous l’affirmons dans notre publication il paraît assez clair que la méthode [Panozzo et al., 2014] décrite pour des maillages surfaciques peut très bien s’étendre aux maillages volumiques puisque des méthodes de génération de champs de croix existent en volumique (ex : [Huang et al., 2011]) et la génération du champ de matrices symétriques positives s’étend très naturellement à la 3D. En effet, la preuve de la positivité des matrices interpolées harmoniquement ne dépend pas de la dimension. Il reste toutefois à noter qu’à notre connaissance il n’existait pas de travaux proposant concrètement une méthode de génération de champs de directions non orthogonales en 3D.

3.2 Champs de directions de dimension n en tant que polynômes sphériques

On considère un complexe simplicial de dimension n dont l'ensemble des cellules (les simplexes de dimension n) est \mathcal{C} . On cherche à construire un champ de directions échantillonné sur chacune des cellules tel qu'il est fréquent de faire dans les méthodes de génération de maillages quadrangulaire [Kälberer et al., 2007] ou hexaédriques [Nieser et al., 2011]. On note \mathcal{E} l'ensemble des paires de cellules adjacentes et $\partial\mathcal{C}$ l'ensemble des cellules de bord du complexe. Pour simplifier les explications, on se restreint aux domaines plats. C'est-à-dire que les sommets du complexe appartiendront à \mathbb{R}^n . Il est toutefois possible d'adapter ce qui suivra à des surfaces courbes en utilisant la connexion de Levi-Civita comme dans les précédents travaux [Diamanti et al., 2014, Ray et al., 2008].

3.2.1 Représentation d'un ensemble de directions

On appelle ensemble de N -directions un multiensemble composé de N vecteurs unitaires et leurs opposés $\{\{u_1, -u_1, \dots, u_N, -u_N\}\}$. Un champ de N -directions est l'assignation d'un ensemble de N -directions par cellule d'un complexe. Notre objectif sera de calculer un champ de N -directions le plus lisse possible tout en respectant certaines contraintes. Il nous reste à définir ce qu'est un champ lisse. Mesurer la similarité entre deux ensembles est une tâche difficile. En effet, si l'on cherche à mesurer des similarités entre les éléments de deux ensembles, il faut construire une correspondance entre les éléments des deux ensembles. Une représentation d'un ensemble de N -directions invariante par réordonnement des $(u_i)_{1 \leq i \leq N}$ et changement de signe simplifierait grandement la mesure de similarité. Inspiré par [Huang et al., 2011, Ray et al., 2016], on identifiera un ensemble de N -directions avec un polynôme sphérique.

Pour un ensemble d'une seule direction u , il paraît assez naturel de considérer le produit scalaire $p_u(s) = \langle u, s \rangle$ comme polynôme de représentation. Malheureusement, cette définition se heurte au problème d'invariance par changement de signe ($p_{-u} \neq p_u$). Pour y remédier il suffit d'élever ce polynôme à une puissance paire, i.e. $p_u(s) = \langle u, s \rangle^{2k}$ pour un $k \in \mathbb{N}^*$ que l'on choisira plus tard. Pour un ensemble de N -directions $\{\{u_i, -u_i \mid 1 \leq i \leq N\}\}$, il est alors possible de sommer les polynômes de représentation de chaque direction u_i :

$$p_u(s) = \sum_{i=1}^N \langle u_i, s \rangle^{2k} \quad (3.5)$$

La commutativité de la somme permet de garantir l'invariance par réordonnement. La figure 3.2 illustre la représentation polynomiale pour une unique direction et $k = 1$. Les repères 2D et 3D avec $k = 2$ illustrés dans les figures (3.3) et (3.9) sont particulièrement utiles pour les applications de paramétrisation.

3.2.2 Base orthonormée des ensembles de N -directions

L'ensemble des polynômes de représentation des ensembles de N -directions est noté $\mathbb{P}_{k,N}$:

$$\mathbb{P}_{k,N} = \left\{ p_u(s) = \sum_{i=1}^N \langle u_i, s \rangle^{2k} \mid u_i \in \mathbb{S}^{n-1} \right\} \quad (3.6)$$

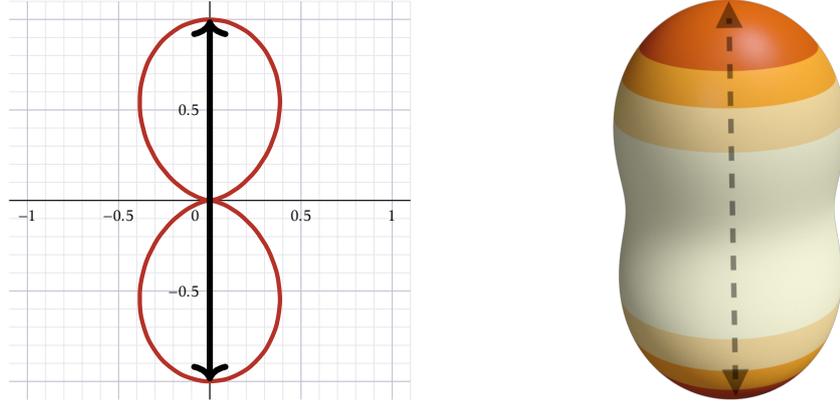


FIGURE 3.2 – Une direction unique (flèche noire) est représentée par un polynôme polaire (gauche) et sphérique (droite) de degré 2. La valeur du polynôme en chaque point du cercle est représentée par une variation du rayon en coordonnées polaires.

Cet ensemble contient uniquement des polynômes homogènes de degré $2k$. Ces derniers forment un espace vectoriel de dimension $L_{n,k} = \binom{2k+n-1}{2k}$ où n est la dimension de notre complexe simplicial et de l'espace euclidien dans lequel il est plongé.

En munissant l'espace des polynômes homogènes de degré $2k$, $\mathbb{R}_{2k}[X_1, \dots, X_n]$, du produit scalaire

$$\langle p, q \rangle = \int_{S^{n-1}} p(s)q(s) ds \quad (3.7)$$

on peut définir une base orthonormée de $\mathbb{R}_{2k}[X_1, \dots, X_n]$ que l'on note $Y_1, \dots, Y_{L_{n,k}}$. En 2D, on peut par exemple utiliser la base de Fourier et en 3D la base des harmoniques sphériques. Ainsi un ensemble de N -directions u peut être représenté par le vecteur $c(u) = (c_1(u) \ \cdots \ c_{L_{n,k}}(u))$ des coefficients de la décomposition de p_u dans la base Y :

$$p_u = \sum_{l=1}^{L_{n,k}} c_l(u) Y_l \quad (3.8)$$

Il est important de noter que cette nouvelle représentation $c(u)$ est également invariante par réordonnement et changement de signes des éléments de u .

Ce dernier vecteur est utile pour simplifier les calculs de distances entre deux polynômes de représentation selon la norme 2 sur S^{n-1} .

$$d(u_1, u_2)^2 = \|p_{u_1} - p_{u_2}\|_2^2 = \int_{S^{n-1}} (p_{u_1}(s) - p_{u_2}(s))^2 ds = \sum_{l=0}^{L_{n,k}} (c_l(u_1) - c_l(u_2))^2 = \|c(u_1) - c(u_2)\|_2^2 \quad (3.9)$$

Cette norme est utilisée, car le polynôme de représentation possède une forme similaire à l'ensemble de N -directions u donnant ainsi un sens géométrique clair à la distance associée (Fig. 3.3 et 3.9).

3.2.3 Champ de N -directions constant par morceaux

Un champ de N -directions est l'assignation d'un ensemble de N -directions par cellule d'un complexe. Maintenant que l'on a défini une distance entre deux ensembles de N -directions, on peut définir une notion de champ lisse. On considère qu'un champ est parfaitement lisse à la frontière entre deux cellules lorsque leurs ensembles de N -directions associés sont égaux. À l'inverse, plus la distance entre les deux ensembles est grande, moins le champ est considéré comme lisse. Cela nous amène à introduire une énergie de lissage s'apparentant à une énergie de Dirichlet discrète :

$$E_D(U) = \sum_{(i,j) \in \mathcal{E}} \|c(u_i) - c(u_j)\|_2^2 \quad (3.10)$$

La conception de champs de directions lisses interpolant des contraintes revient alors à minimiser l'énergie de lissage E_D tout en imposant des contraintes directement sur la représentation vectorielle des ensembles de N -directions de U . Cela rend les problèmes de conception de champ beaucoup plus accessibles qu'auparavant.

3.2.4 Degré du polynôme de représentation

Une propriété qu'il est naturel de souhaiter est que deux ensembles de N -directions différents aient des polynômes de représentation différents et donc une distance non nulle entre eux. C'est-à-dire que l'on souhaite que l'application $u \mapsto p_u$ soit injective sur l'ensemble des ensembles de N -directions. Cette propriété recherchée implique des contraintes sur le choix de k . [Shen et al., 2016] établissent de telles contraintes pour des ensembles de N -directions vérifiant certaines symétries. On peut entre autres réutiliser leur résultat vis-à-vis des $2N$ -RoSy (ensembles de N -directions en dimension $n = 2$ stables par rotation de π/N). Ce résultat montre que pour représenter de tels ensembles, il est nécessaire de choisir k supérieur ou égal à N . L'article se plaçant dans un cadre assez différent du nôtre, voici une démonstration de ce résultat dans notre formalisme.

Proposition 3.1. *Dans le cas $n = 2$, l'application $u \mapsto p_u$ est injective seulement si $k \geq N$.*

Démonstration. Pour $k = 0$, $p_u(s) = N$ indépendamment de u . L'injectivité implique donc $k > 0$. Supposons maintenant $k > 0$ et associons le plan réel \mathbb{R}^2 au plan complexe \mathbb{C} . On peut alors considérer l'ensemble de N -directions $u_\varphi = \{e^{i(j\pi/N + \varphi)}, -e^{i(j\pi/N + \varphi)} \mid 1 \leq j \leq N\}$ pour un certain $\varphi \in \mathbb{R}$. Son polynôme de représentation est donné par :

$$p_{u_\varphi}(e^{i\theta}) = \sum_{j=1}^N \cos(\theta - j\pi/N - \varphi)^{2k} \quad (3.11)$$

Intéressons-nous désormais à la dérivée de ce polynôme par rapport à θ . Pour alléger les cal-

culs, on utilisera $x_j = \theta - j\pi/N - \varphi$:

$$\begin{aligned}
 \partial_{\theta} p_{u_{\varphi}}(e^{i\theta}) &= 2k \sum_{j=1}^N \sin(x_j) \cos(x_j)^{2k-1} \\
 &= k \sum_{j=1}^N \sin(2x_j) \cos(x_j)^{2k-2} \\
 &= \frac{k}{2^{2k-1}} \sum_{j=1}^N (e^{2ix_j} - e^{-2ix_j})(e^{ix_j} + e^{-ix_j})^{2k-2} \\
 &= \frac{k}{2^{2k-1}} \sum_{j=1}^N \sum_{l=0}^{2k-2} \binom{2k-2}{l} (e^{2ix_j} - e^{-2ix_j}) e^{2i(k-1-l)x_j} \\
 &= \frac{k}{2^{2k-1}} \sum_{l=0}^{2k-2} \binom{2k-2}{l} \sum_{j=1}^N (e^{2i(k-l)x_j} - e^{2i(k-2-l)x_j}) \\
 &= \frac{k}{2^{2k-1}} \sum_{l=0}^{2k-2} \binom{2k-2}{l} \left(e^{2i(k-l)(\theta-\varphi)} \sum_{j=1}^N e^{2i(l-k)j\pi/N} - e^{2i(k-2-l)(\theta-\varphi)} \sum_{j=1}^N e^{2i(l+2-k)j\pi/N} \right)
 \end{aligned}$$

On remarque ensuite que si N ne divise pas $(l-k)$, alors $\sum_{j=1}^N e^{2i(l-k)j\pi/N}$ est nul. Or ici, $l-k$ varie de $-k$ à $k-2$. Donc pour $k < N$, la somme $\sum_{j=1}^N e^{2i(l-k)j\pi/N}$ est toujours nulle. De la même manière $l+2-k$ varie de $-k+2$ à k et donc pour $k < N$, la seconde somme est également nulle. On en conclut que :

$$k < N \Rightarrow \partial_{\theta} p_{u_{\varphi}}(e^{i\theta}) = 0 \quad (3.12)$$

Si $k < N$, alors $p_{u_{\varphi}}$ est constant. De plus $p_{u_{\varphi}}(e^{i\varphi})$ ne dépend pas de φ , donc pour tout $\varphi' \in \mathbb{R}$, $p_{u_{\varphi}} = p_{u_{\varphi'}}$ et donc l'application $u \mapsto p_u$ n'est pas injective. \square

Expérimentalement, il semble que $k \geq N$ soit également une condition suffisante pour l'injectivité. Intuitivement, cela vient du fait que l'ensemble de N -direction u_{φ} proposé dans la preuve est le pire cas que l'on puisse construire. Pour $n > 2$ il est toujours possible de considérer un ensemble de N -directions u dont les vecteurs u_i se trouvent tous dans un plan. Donc la condition $k \geq N$ est toujours nécessaire à l'injectivité et semble à nouveau expérimentalement suffisante.

Il est enfin important de noter que choisir une grande valeur de k augmente le nombre de coefficients de c et complexifie fortement la minimisation de l'énergie de lissage E_D (3.10). Dans la suite, nous nous intéressons aux deux cas $n = N = 2$ et $n = N = 3$. Nous avons décidé de travailler avec le plus petit degré possible, à savoir $k = 2$ en dimension 2. En dimension 3, nous utilisons également $k = 2$ malgré la condition nécessaire $k \geq N$. En effet l'injectivité semble être perdue uniquement pour les ensembles de N -directions ayant trois directions dans un même plan, or ce cas de figure n'est pas souhaitable dans l'optimisation d'un champ de directions pour la génération de maillages hexaédriques puisque cela engendrerait une paramétrisation dégénérée. Idéalement, nos ensembles de directions devraient avoir trois directions orthogonales, ce qui correspond aux *Cube-RoSy* chez [Shen et al., 2016] qui peuvent être représentés

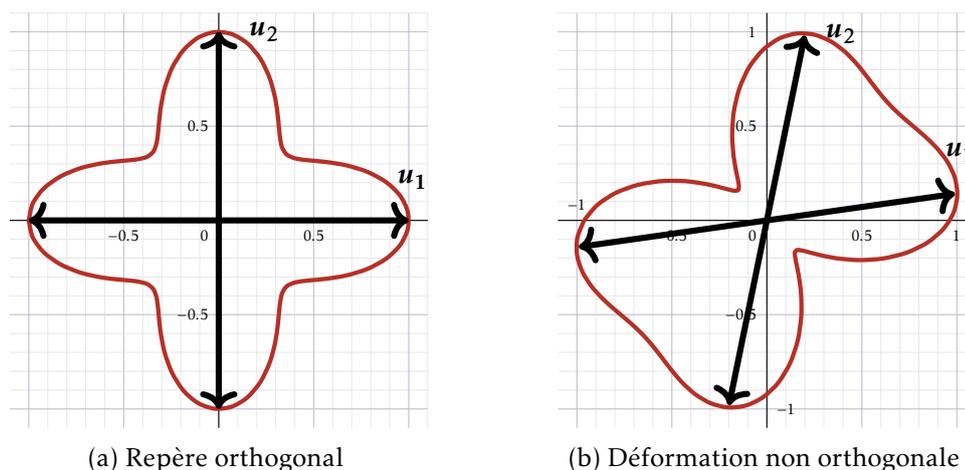


FIGURE 3.3 – Un ensemble de 2-directions $\{\{u_1, -u_1, u_2, -u_2\}\}$ (flèches noires) représenté par un polynôme polaire de degré 4.

avec $k = 2$. Les figures 3.3b et 3.9b montrent que des polynômes de degré 4 semblent suffire à correctement représenter les déformations non dégénérées d'une croix 2D ou 3D.

3.3 Champs de directions 2D

Dans cette section, on étudie le cas $n = 2$. Les polynômes de représentation $\mathbb{P}_{k,N}$ (3.6) sont alors définis sur le cercle unité et décomposés dans la base de Fourier. On se restreindra même à $N = 2$ afin de considérer des champs de repères. Nous verrons qu'avec ce choix de base et lorsque les deux directions sont orthogonales, notre représentation est exactement celle des champs de croix utilisés dans plusieurs articles [Palacios and Zhang, 2007, Knöppel et al., 2013, Diamanti et al., 2014].

3.3.1 Base de Fourier

Le polynôme de représentation p_u (3.5) est défini sur le cercle unité \mathbb{S}^1 . Il est naturel d'utiliser la base de Fourier comme base orthonormée pour décomposer notre signal périodique. Pour ce faire on paramétrise dans un premier temps notre ensemble de 2-directions $u = \{\{u_1, -u_1, u_2, -u_2\}\}$ par $u_1 = (\cos \alpha \ \sin \alpha)$ et $u_2 = (\cos \beta \ \sin \beta)$. On notera $c_{l,1}(\alpha, \beta)$ le coefficient associé à $t \mapsto \cos(lt)$ et $c_{l,2}(\alpha, \beta)$ le coefficient associé à $t \mapsto \sin(lt)$ dans la décomposition de $s \mapsto p_u(s)$ avec $s = (\cos t \ \sin t)$.

Développer l'expression de $\langle u_1, s \rangle^4$ permet de retrouver les coefficients $c_{l,1}$ et $c_{l,2}$ dans la base de Fourier :

$$\begin{aligned} \langle u_1, s \rangle^4 &= (\cos \alpha \cos t + \sin \alpha \sin t)^4 \\ &= \frac{3}{8} + \frac{1}{2} \cos(2\alpha) \cos(2t) + \frac{1}{2} \sin(2\alpha) \sin(2t) + \frac{1}{8} \cos(4\alpha) \cos(4t) + \frac{1}{8} \sin(4\alpha) \sin(4t) \end{aligned}$$

Définir les coefficients à un facteur multiplicatif près n'ayant aucun impact par la suite, on décide de les multiplier par 8 afin d'obtenir les expressions suivantes :

$$\begin{cases} c_{2,1}(\alpha, \beta) = 4\cos(2\alpha) + 4\cos(2\beta) \\ c_{2,2}(\alpha, \beta) = 4\sin(2\alpha) + 4\sin(2\beta) \\ c_{4,1}(\alpha, \beta) = \cos(4\alpha) + \cos(4\beta) \\ c_{4,2}(\alpha, \beta) = \sin(4\alpha) + \sin(4\beta) \end{cases} \quad (3.13)$$

Si l'on omet le terme constant de la décomposition qui est indépendant de u , notre repère est représenté par des coefficients uniquement aux fréquences $l = 2$ et $l = 4$. Un point important à noter est que le repère est orthogonal ($\beta \equiv \alpha + \pi/2 \pmod{\pi}$) si et seulement si les coefficients d'ordre deux sont nuls. Dans ce cas, le repère est uniquement représenté par les coefficients d'ordre 4 ($\cos(4\alpha), \sin(4\alpha)$) ou de manière équivalente ($\cos(4\beta), \sin(4\beta)$). On retrouve donc la représentation usuelle des champs de croix [Palacios and Zhang, 2007].

3.3.2 Conception de champs de repères pour le remaillage quadrangulaire

Dans cette section, nous allons construire un problème d'optimisation pour trouver le champ de directions le plus lisse interpolant des contraintes de direction prescrites par un utilisateur. Un paramètre permettra de contrôler l'orthogonalité du champ obtenu.

Variables Pour simplifier la résolution du problème d'optimisation, nous n'utilisons pas les angles α et β comme variables, ni leurs vecteurs unitaires associés u_1 et u_2 .

Si l'on cherche à obtenir un problème le plus simple possible, il semble naturel de considérer les coefficients $c_{l,m}$ en tant que variables. Le souci de cette approche est que le résultat du processus d'optimisation pourrait aboutir à des variables $\bar{c}_{l,m}$ qui ne correspondent pas du tout à une paire d'angles (α, β) (c'est-à-dire que \bar{c} pourrait être éloigné de la 2-variété définie par l'image de $(\alpha, \beta) \mapsto c(\alpha, \beta)$). Une manière de remédier à ce problème est de rajouter un terme de régularisation à l'énergie que l'on minimise et qui viserait à pénaliser la distance entre \bar{c} et la 2-variété \mathcal{V} définie par l'image de $(\alpha, \beta) \mapsto c(\alpha, \beta)$. Malheureusement, il est difficile de définir de tels termes de régularisation pour \bar{c} . Une piste serait de s'inspirer de [Palmer et al., 2020]. Si l'on arrive à démontrer que \mathcal{V} est une variété algébrique définie comme le noyau de polynômes de degré 2 il est possible de calculer la projection et la distance euclidienne entre un point $\bar{c} \in \mathbb{R}^4$ et \mathcal{V} par une optimisation semi-définie positive.

On propose donc d'utiliser comme variables les vecteurs $v_1 = (\cos 2\alpha \quad \sin 2\alpha)$ et $v_2 = (\cos 2\beta \quad \sin 2\beta)$ obtenus en doublant les angles α et β . On exprime alors les coefficients $c_{l,m}$ (3.13) en fonction de ces variables de la manière suivante :

$$\begin{cases} c_{2,1}(v) = 4v_{1x} + 4v_{2x} \\ c_{2,2}(v) = 4v_{1y} + 4v_{2y} \\ c_{4,1}(v) = 2v_{1x}^2 + 2v_{2x}^2 - 2 \\ c_{4,2}(v) = 2v_{1x}v_{1y} + 2v_{2x}v_{2y} \end{cases} \quad (3.14)$$

À noter que ces expressions ne sont pas définies de façon unique. Par exemple pour le coefficient $c_{4,1}$, il aurait été possible de considérer $v_{1x}^2 - v_{1y}^2 + v_{2x}^2 - v_{2y}^2$.

Pour ce qui est de la régularisation, on remarque dans un premier temps que (v_1, v_2) appartient à l'image de $(\alpha, \beta) \mapsto ((\cos 2\alpha \quad \sin 2\alpha), (\cos 2\beta \quad \sin 2\beta))$ si et seulement si $\|v_1\| = 1$ et $\|v_2\| = 1$. On peut donc définir l'énergie de régularisation suivante :

$$E_{\text{reg}}^{2D}(v) = (\|v_1\|^2 - 1)^2 + (\|v_2\|^2 - 1)^2$$

Par la suite, on associe un repère v_i par cellule $i \in \mathcal{C}$. On regroupe toutes ces valeurs au sein d'une matrice $v \in \mathbb{R}^{\#\mathcal{C} \times 4}$.

Énergie de lissage L'énergie de lissage E_D (3.10) définie en section 3.2.3 dépend uniquement des coefficients de la décomposition (3.14). On modifie légèrement cette énergie en ajoutant un paramètre λ qui contrôle l'importance des coefficients d'ordre 2 :

$$E_D^{2D}(v) = \lambda^2 \sum_{(i,j) \in \mathcal{E}} \sum_{m=1,2} (c_{2,m}(v_i) - c_{2,m}(v_j))^2 + \sum_{(i,j) \in \mathcal{E}} \sum_{m=1,2} (c_{4,m}(v_i) - c_{4,m}(v_j))^2 \quad (3.15)$$

Paramètre d'orthogonalité Comme mentionné dans la section 3.3.1, l'orthogonalité est directement liée aux coefficients de second ordre. Dans l'énergie de lissage, nous pondérons différemment les coefficients d'ordre 2 de sorte qu'un poids λ élevé mènera à des repères ayant la même différence d'angle entre leurs deux directions à travers toute la surface. On peut remarquer qu'un repère non orthogonal ne possède pas de symétries par rotations de $\pi/4$ mais uniquement par rotations de $\pi/2$. En conséquence, la présence d'une singularité interne au champ possédant un indice non multiple de $\pi/2$ forcera les ensembles de 2-directions à être orthogonales autour de la singularité et par suite, sur tout le maillage dans le cas où λ est élevé.

L'effet de λ sur le champ de repères et sur le maillage quadrangulaire est illustré Figure 3.4. La valeur $\lambda = 1$ correspond à la notion géométrique de lisse expliquée en section 3.2.3. Une valeur plus grande que 1 promeut l'orthogonalité ou à minima la constance de différence d'angle entre les directions dans le cas d'un champ ne possédant pas de singularité interne avec un indice non multiple de $\pi/2$ (ce qui peut arriver si l'on veut mailler un parallélogramme par exemple). Une valeur plus faible de λ pénalise peu des ensembles de 2-directions qui tendent à être dégénérés (c'est-à-dire lorsque les deux directions de l'ensemble se rapprochent). Une conséquence est que les singularités ont tendance à être poussées sur les bords du domaine.

Le choix de ce paramètre est dicté par l'application et est souvent le résultat d'un compromis. D'un côté, les repères orthogonaux mènent à des maillages quadrangulaires avec des cellules proches de carrés parfaits, mais en contrepartie ne peuvent pas représenter les coins à angle aigu omniprésents dans les maillages quadrangulaires. D'un autre côté, les repères non orthogonaux gèrent correctement les coins à angle aigu mais introduisent pour cela des torsions et dans les cas extrêmes suppriment les singularités de l'intérieur du domaine.

Contraintes de direction La contrainte la plus répandue pour la génération de champs de directions est d'imposer des directions sur certaines cellules. Dans notre cas, imposer ces contraintes est simple, il suffit de fixer les valeurs de v_1 et v_2 aux vecteurs correspondants aux angles souhaités. Si une seule direction doit être contrainte, il suffit d'uniquement imposer la valeur de v_1 . On peut remarquer que v_1 et v_2 ont des rôles parfaitement symétriques. Le choix du vecteur que l'on contraint n'a donc pas d'importance.

Pour nos résultats de maillages quadrangulaires, nous contraignons une direction à être perpendiculaire à chaque arête de bord du modèle. Si une cellule triangulaire du maillage d'entrée possède deux arêtes de bord alors les deux directions de la cellule sont contraintes.

Optimisation Notre schéma d'optimisation est fondé sur une méthode de diffusion MBO [Merriman et al., 1992] minimisant une énergie inspirée de celle de type Ginzburg-Landau in-

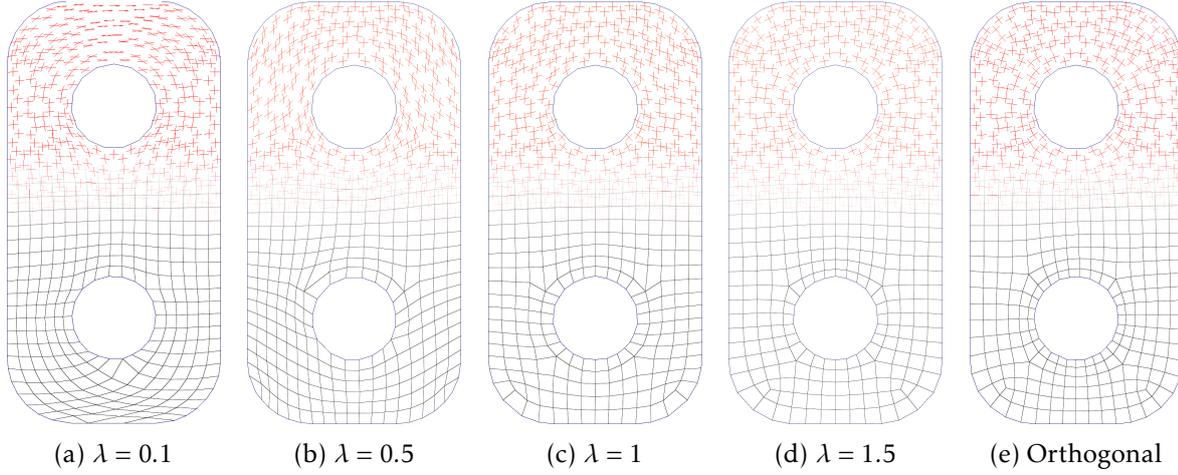


FIGURE 3.4 – Résultats de notre méthode de génération de champs de repères pour des valeurs croissantes du paramètre d’orthogonalité λ . Les faibles valeurs (à gauche) poussent les singularités hors du domaine alors que les grandes valeurs font tendre le champ vers celui obtenu par [Viertel and Osting, 2019] qui est orthogonal (à droite).

troducte dans [Viertel and Osting, 2019]. On cherche à minimiser l’énergie suivante :

$$E^{2D}(v) = E_D^{2D}(v) + \frac{1}{\epsilon} E_{\text{reg}}^{2D}(v) \quad (3.16)$$

Si l’on remplace le premier terme E_D^{2D} par une énergie de Dirichlet directement sur v

$$\tilde{E}^{2D}(v) = \tilde{E}_D^{2D}(v) + \frac{1}{\epsilon} E_{\text{reg}}^{2D}(v) = \sum_{(i,j) \in \mathcal{E}} \|v_i - v_j\|^2 + \frac{1}{\epsilon} E_{\text{reg}}^{2D}(v) \quad (3.17)$$

on obtient une discrétisation de l’énergie minimisée dans [Viertel and Osting, 2019]. La méthode MBO proposée pour trouver une approximation d’un minimiseur d’une telle énergie consiste en partir d’une initialisation $v^{(0)}$ et alterner des étapes de diffusion (i) et de renormalisation (ii) :

$$(i) \quad \tilde{v}^{(k+1)} = e^{\tau \Delta} v^{(k)} \quad ; \quad (ii) \quad v_{i,j}^{(k+1)} = \frac{\tilde{v}_{i,j}^{(k+1)}}{\|\tilde{v}_{i,j}^{(k+1)}\|}, \quad i = 1 \dots \#\mathcal{C}, \quad j = 1, 2 \quad (3.18)$$

Où $e^{\tau \Delta} v^{(k)}$ est le résultat de la diffusion suivante :

$$e^{\tau \Delta} v^{(k)} = w(\tau), \text{ avec } w \text{ tel que } \begin{cases} \partial_t w(t, x) = \Delta w(t, x), & x \in \mathcal{C} \\ w(t, x) = g(x), & x \in \partial \mathcal{C} \\ w(0, x) = v^{(k)}(x), & x \in \mathcal{C} \end{cases} \quad (3.19)$$

La linéarisation de l’exponentielle du laplacien résulte en un schéma d’Euler explicite

$$\tilde{v}^{(k+1)} = (I + \tau \Delta) v^{(k)} \quad (3.20)$$

que l’on peut transformer en un schéma d’Euler implicite :

$$(I - \tau \Delta) \tilde{v}^{(k+1)} = v^{(k)} \quad (3.21)$$

Algorithme 1 : Méthode de diffusion pour approximer la minimisation de l'énergie 2D (3.16)

Entrées : Trois paramètres $\epsilon > 0$, $\tau > 0$ et $\delta > 0$ ainsi qu'un premier champ de directions paramétré par les variables $v^{(0)}$ servant d'initialisation

1 **tant que** $k > 0$ et $\|v^{(k)} - v^{(k-1)}\| > \delta$ **faire**

2 (i) Minimiser la fonction de diffusion f_k^{2D} (3.23) avec L-BFGS

3 $\tilde{v}^{(k+1)} \leftarrow \arg \min_w f_k^{2D}(w)$

4 (ii) Re-normaliser $\tilde{v}^{(k+1)}$

5 **pour** $i \in \mathcal{C}$ **faire**

6
$$v_{i,1}^{(k+1)} \leftarrow \frac{\tilde{v}_{i,1}^{(k+1)}}{\|\tilde{v}_{i,1}^{(k+1)}\|}; \quad v_{i,2}^{(k+1)} \leftarrow \frac{\tilde{v}_{i,2}^{(k+1)}}{\|\tilde{v}_{i,2}^{(k+1)}\|}$$

La solution de cette équation linéaire en $\tilde{v}^{(k+1)}$ est le minimiseur de l'énergie suivante :

$$\tilde{f}_k(w) = \text{tr}(-w^T \Delta w) + \frac{1}{\tau} \|w - v^{(k)}\|^2 = \tilde{E}_D^{2D}(w) + \frac{1}{\tau} \|w - v^{(k)}\|^2 \quad (3.22)$$

Les directions du bord n'étant pas nécessairement toutes contraintes dans notre problème comme le suggère (3.19), nous proposons de rajouter notre terme de régularisation sur le bord du modèle. Puis pour minimiser l'énergie souhaitée (3.16), on remplace également l'énergie de Dirichlet par celle définie quelques paragraphes plus hauts (3.15). Finalement, notre étape de diffusion peut être résumée à minimiser l'énergie suivante :

$$f_k^{2D}(w) = E_D^{2D}(w) + \frac{1}{\epsilon} \sum_{i \in \partial \mathcal{C}} E_{\text{reg}}^{2D}(w_i) + \frac{1}{\tau} \|w - v^{(k)}\|^2 \quad (3.23)$$

Contrairement à \tilde{f}_k , cette nouvelle fonction f_k^{2D} ne peut être minimisée en résolvant une équation linéaire. Nous avons décidé de minimiser cette dernière avec une méthode quasi newtonienne, L-BFGS [Nocedal, 1980]. Notre processus d'optimisation est résumé par l'Algorithme 1.

3.3.3 Résultats

Nous évaluons l'usage de notre méthode de conception de champs de directions dans le contexte de la génération de maillages quadrangulaires. Tous les champs de repères sont contraints à avoir une direction tangente au bord du domaine. Le paramètre pondérant l'énergie de régularisation est choisi à $\epsilon = 0.1$. Dans toutes nos expériences, le paramètre de diffusion est fixé à $\tau = 2$. Enfin, le paramètre d'orthogonalité λ est ajusté en fonction de nos besoins, mais la valeur $\lambda = 1.5$ s'est avéré être un bon compromis dans la plupart des cas.

Le problème d'optimisation étant non convexe, une bonne initialisation est cruciale. Nous avons décidé d'initialiser avec un champ de repères orthogonaux [Palacios and Zhang, 2007]. Le choix de l'initialisation et son impact sur la solution de l'optimisation sont discutés pour le cas de la 3D dans la section 3.4.3, mais les conclusions s'appliquent également en 2D. Notre optimisation (Algorithme 1) requiert généralement une dizaine d'itérations pour converger.

Les maillages quadrangulaires sont extraits d'un champ de repères en utilisant des méthodes de l'état de l'art [Kälberer et al., 2007, Ebke et al., 2013].

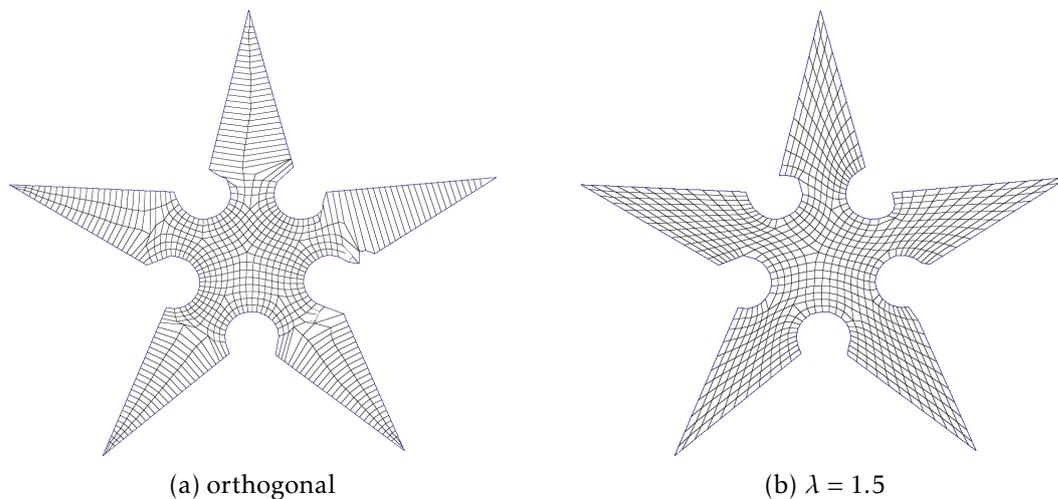


FIGURE 3.5 – Un champ de repères orthogonaux 3.5a ne peut pas être utilisé pour générer un maillage quadrangulaire d'un modèle possédant des coins à angle aigu. Notre champ de repères non orthogonaux 3.5b permet d'obtenir un maillage quadrangulaire de bonne qualité.

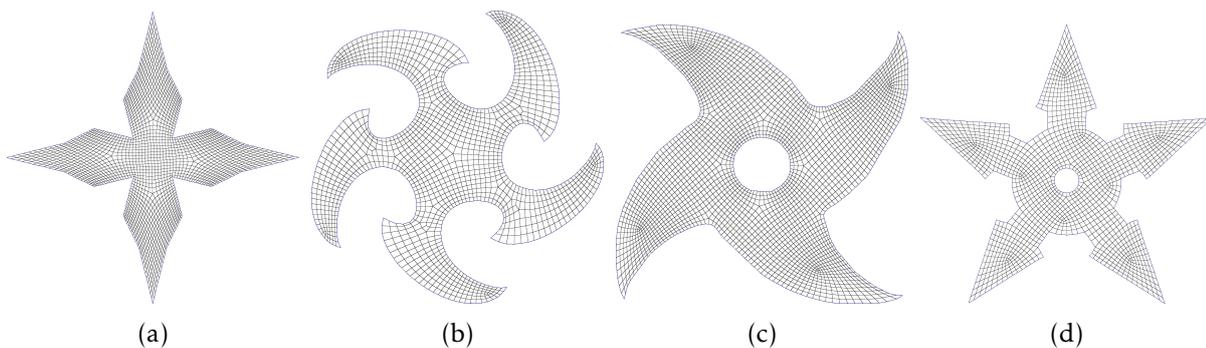


FIGURE 3.6 – Notre méthode permet la construction de maillages quadrangulaires de bonne qualité en ajustant le paramètre $\lambda = 1.5$

Remaillage quadrangulaire de modèles CAO La génération de maillages quadrangulaires de modèles CAO est rendue difficile par la présence de coins à angle aigu. En effet, il n'est pas possible de placer des cellules ayant une forme de carré quasi parfait en de tels coins. La figure 3.5 illustre ce problème. Le champ de repères orthogonaux 3.5a ne permet pas de satisfaire les contraintes des deux côtés des coins à angle aigu, rendant alors impossible d'obtenir un maillage quadrangulaire valide. Notre méthode permet de générer des maillages quadrangulaires anisotropes 3.5b qui satisfont très bien ces contraintes particulières.

La figure 3.6 montre une collection de maillages quadrangulaires générés avec nos champs de repères non orthogonaux. On peut constater que les contraintes de bord sont parfaitement satisfaites tout en ayant un faible nombre de singularités et des quadrilatères loin d'être dégénérés.

Temps d'exécution La table 3.1 montre les temps d'exécution de chaque étape de notre méthode pour chacun des modèles présentés précédemment. L'étape qui prend le plus de temps est le calcul du champ de repères non orthogonaux comme elle requiert 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'attendre la convergence de L-BFGS durant l'étape de diffusion est souvent non nécessaire et l'optimisation pourrait être arrêtée plus tôt. Il pourrait aussi être envisagé d'utiliser une méthode de Newton. Toutes les expérimentations ont été conduites sur un ordinateur portable avec un processeur Intel Core i5-10400H avec 4 cœurs cadencés à 2.6 GHz et 32 Gb de RAM.

Figure	Faces	Initialisation	Optimisation	Maillage
3.4	658	0.017	< 1	2
3.5b	1 282	0.026	< 1	3
3.6c	9 968	0.177	26	13
3.6a	18 430	0.355	66	30
3.6d	29 000	0.611	120	45
3.6b	36 047	0.775	161	46

TABLE 3.1 – Temps d'exécution (en secondes) des différentes étapes de notre méthode : initialisation avec un champ de croix [Palacios and Zhang, 2007], résolution de notre problème d'optimisation non convexe (Algorithme 1) et l'extraction de maillage quadrangulaire [Kälberer et al., 2007, Ebke et al., 2013].

3.3.4 Comparaison avec les champs de polyvecteurs [Diamanti et al., 2014]

Comme évoqué en section 3.1.1, [Diamanti et al., 2014] propose une représentation d'un repère par les coefficients d'un polynôme complexe dont les racines sont les représentations complexes des directions dans une base du plan tangent. Tout comme ce que nous proposons, leur représentation est invariante par réordonnement des directions. Toutefois, leur processus d'optimisation est assez différent puisque leurs variables sont les coefficients des polynômes permettant ainsi de minimiser leur énergie en résolvant un simple problème linéaire. Aucune régularisation n'est appliquée dans le but d'imposer les racines des polynômes interpolés à être unitaire.

Par la suite [Diamanti et al., 2015] a proposé une nouvelle énergie renforçant l'intégrabilité du champ de repère, c'est-à-dire de telle sorte que chaque direction du champ soit conservative/irrotationnelle (champ de vecteur dont le rotationnel est nul) afin que chaque direction corresponde aux gradients des coordonnées d'une paramétrisation du modèle. Cette énergie est minimisée en utilisant pour variables les directions u_1 et u_2 . Cela montre que leur énergie de lissage commune à [Diamanti et al., 2014] et [Diamanti et al., 2015] peut être minimisée avec différents jeux de variables. Nous proposons dans la suite de cette section de comparer leur énergie de lissage à la nôtre au sein de notre processus d'optimisation travaillant sur les variables v_1 et v_2 .

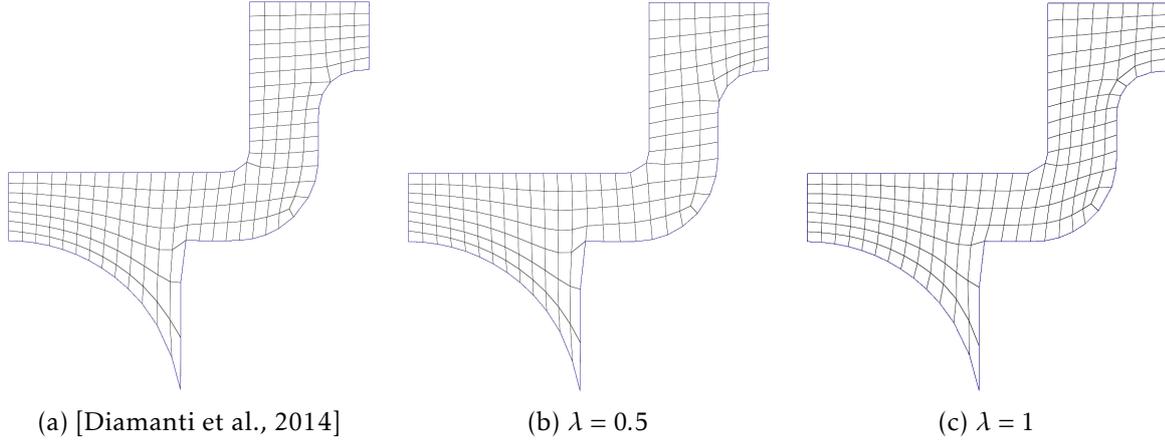


FIGURE 3.7 – La distance des polyvecteurs [Diamanti et al., 2014] donne des résultats similaires aux nôtres pour le choix $\lambda = 0.5$, notamment en termes de placement des singularités.

Distance entre repères Dans la formulation des polyvecteurs, un repère $u = \{u_1, -u_1, u_2, -u_2\}$ est représenté par le polynôme complexe :

$$p_u^{\text{poly}}(z) = (z^2 - u_1^2) \cdot (z^2 - u_2^2) = z^4 - (u_1^2 + u_2^2)z^2 + u_1^2 u_2^2 \quad (3.24)$$

En identifiant le plan \mathbb{R}^2 au plan complexe \mathbb{C} et donc u_i^2 à v_i , il est possible de réécrire ce polynôme avec nos notations par le quadruplet de coefficients suivant :

$$\begin{cases} c_{2,1}^{\text{poly}}(v) = v_{1x} + v_{2x} \\ c_{2,2}^{\text{poly}}(v) = v_{1y} + v_{2y} \\ c_{4,1}^{\text{poly}}(v) = v_{1x}v_{2x} - v_{1y}v_{2y} \\ c_{4,2}^{\text{poly}}(v) = v_{1x}v_{2y} + v_{1y}v_{2x} \end{cases} \quad (3.25)$$

À un facteur multiplicatif près, les coefficients d'ordre 2 sont les mêmes que les nôtres (3.14). En revanche les coefficients d'ordre 4 diffèrent. Plus précisément, lorsqu'on se restreint au cercle unité on, obtient les relations suivantes :

$$\begin{cases} c_2(\alpha, \beta) = 4 \cdot c_2^{\text{poly}}(\alpha, \beta) \\ c_4(\alpha, \beta) = 2 \cdot \cos(2\alpha - 2\beta) \cdot c_4^{\text{poly}}(\alpha, \beta) \end{cases} \quad (3.26)$$

Une première remarque est que dans le cas d'un champ de croix, la valeur du paramètre d'orthogonalité $\lambda = 0.5$ permet de reproduire la représentation des polyvecteurs. Une seconde remarque est que notre distance donne moins d'importance aux coefficients d'ordre 4 lorsque la différence d'angle entre les directions de nos repères est proche de $\pi/4$. Toutefois, en pratique, cette différence ne semble pas impacter significativement les champs de repères obtenus en utilisant ces différentes représentations. Il nous est possible de générer des champs de polyvecteurs en remplaçant c par c^{poly} dans notre optimisation. La figure 3.7 compare des maillages quadrangulaires obtenus à partir de ces deux représentations. On observe des résultats similaires pour $\lambda = 0.5$; Et comme on pouvait s'y attendre pour $\lambda = 1$, nous obtenons une singularité interne supplémentaire dans la partie supérieure droite du maillage.

La figure 3.8 illustre deux cas où les comportements d'une descente de gradient diffèrent lorsque l'on utilise notre distance ou celle de [Diamanti et al., 2014]. Sur chacune des deux figures on initialise nos variables à $v = v^{(\text{init})}$ correspondant au repère dont les deux directions

sont en rouge et l'on trace les trajectoires de v_1 et v_2 lors d'une descente de gradient minimisant $\lambda^2 \|c_2(v) - c_2(v^{(\text{obj})})\|^2 + \|c_4(v) - c_4(v^{(\text{obj})})\|^2$ dans le cas des trajectoires dont le dégradé de couleur va du rouge au bleu et en minimisant $\|c^{\text{poly}}(v) - c^{\text{poly}}(v^{(\text{obj})})\|^2$ dans le cas des trajectoires dont le dégradé de couleur va du jaune au cyan. $v^{(\text{obj})}$ correspond au repère dont les directions sont en bleu. En réalité, les trajectoires ne sont pas celles de v_1 et v_2 mais les trajectoires de leurs racines complexes qui correspondent réellement aux directions de nos repères puisque pour rappel v_i est le carré de la i -ème direction u_i . Dans les deux exemples, la correspondance entre les directions du repère initial (rouge) et celles du repère objectif (bleu) n'est pas la même entre les deux distances (celle de [Diamanti et al., 2014] et la nôtre). Dans le premier cas 3.8a, les deux méthodes proposent des correspondances qui semblent acceptables. La distance de [Diamanti et al., 2014] écarte les deux directions du repère initial tandis que notre distance mène à une rotation du repère. Dans le second exemple 3.8b, la trajectoire de [Diamanti et al., 2014] ne semble pas naturelle et peut paraître sous-optimal puisque les deux directions se croisent en parcourant in fine une "plus longue distance".

Les deux exemples de la figure 3.8 sont en réalité des cas relativement peu fréquents. En générant de façon uniforme des quadruplets d'angles $(\alpha_{\text{init}}, \beta_{\text{init}}, \alpha_{\text{obj}}, \beta_{\text{obj}})$ dans $[0, 2\pi[^4$ de sorte à construire v_{init} et v_{obj} et appliquer la descente de gradient décrite dans le paragraphe précédent, on peut faire une analyse statistique du comportement de la correspondance entre les directions du repère v_{init} et du repère v_{obj} . On obtient alors que la correspondance est identique entre notre distance et celle de [Diamanti et al., 2014] dans 88% des cas. De plus, lorsque la correspondance est différente, notre distance reproduit l'appariement entre les directions des repères qui correspond au plan de transport pour la distance de Wasserstein d'ordre 2 sur le cercle unité entre les distributions $\frac{1}{2}(\delta_{\alpha_{\text{init}}} + \delta_{\beta_{\text{init}}})$ et $\frac{1}{2}(\delta_{\alpha_{\text{obj}}} + \delta_{\beta_{\text{obj}}})$ dans 51% des cas. Ce dernier appariement me paraissant à titre personnel, le plus naturel, aucune des deux distances ne semblent significativement "meilleure" que l'autre.

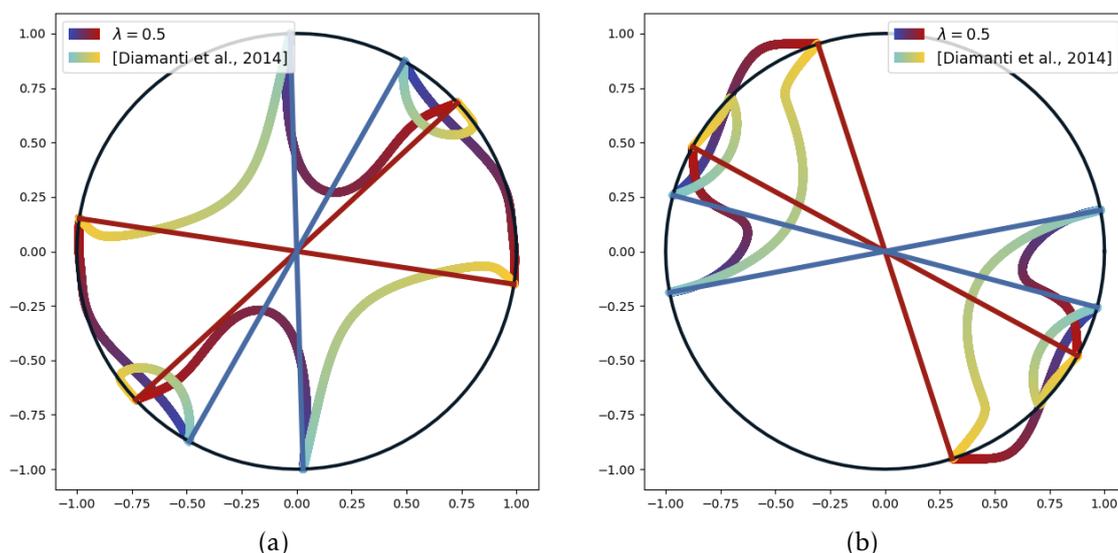


FIGURE 3.8 – Comparaison des trajectoires de v lors d'une descente de gradient minimisant la distance entre v (initialisé au repère rouge) et $v^{(\text{obj})}$ (repère bleu) en utilisant notre représentation avec $\lambda = 0.5$ (trajectoire bleu/rouge) et la représentation de [Diamanti et al., 2014] (trajectoire cyan/jaune).

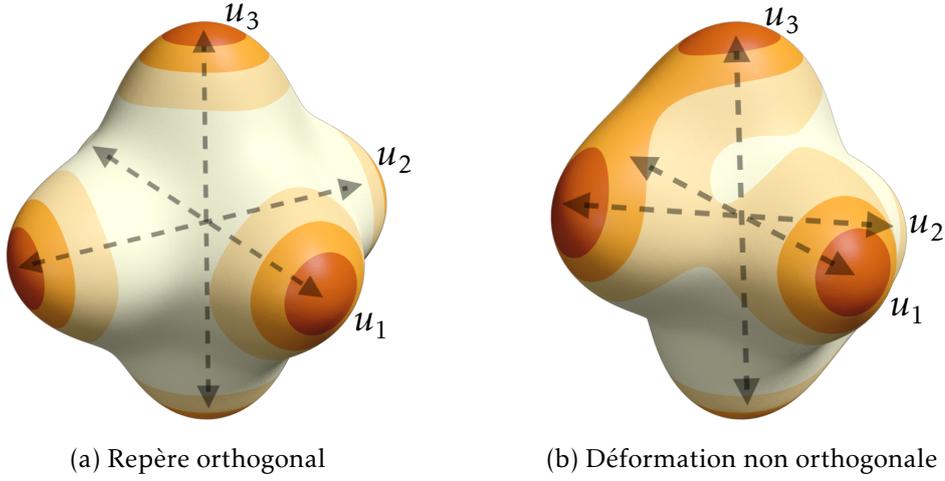


FIGURE 3.9 – Tracés sphériques des polynômes de représentation (3.27) d’ensembles de 3-directions

3.4 Champs de directions 3D

Dans le cas particulier des champs de repères 3D, nous considérons des ensembles de 3-directions $u = \{u_1, -u_1, u_2, -u_2, u_3, -u_3\}$. Comme nos directions sont 3D, nous n’utilisons plus la base de Fourier, mais la base des harmoniques sphériques pour décomposer les polynômes de $\mathbb{P}_{k,3}$ (3.6).

Comme dans de précédents travaux [Huang et al., 2011, Ray et al., 2016] nous choisissons d’utiliser des polynômes de degré 4 ($k = 2$). En effet c’est le plus petit degré permettant d’obtenir les mêmes symétries qu’une croix 3D. Toutefois, comme l’indique la proposition 3.1, pour des repères dégénérés, $k = 2$ ne permet pas l’injectivité de la représentation et donc la distance que l’on construit distinguera difficilement des repères différents lorsqu’ils seront proches d’être dégénérés. Nous avons décidé de garder des polynômes de degré 4, car pour de plus hauts degrés la complexité de l’optimisation augmente fortement (passant de 15 coefficients par repères pour $k = 2$, à 28 coefficients pour $k = 3$). Notre polynôme de représentation est alors le suivant :

$$p_u(s) = \langle u_1, s \rangle^4 + \langle u_2, s \rangle^4 + \langle u_3, s \rangle^4 \quad (3.27)$$

Son tracé sphérique est donné par la figure 3.9.

3.4.1 Base d’harmoniques sphériques

Les harmoniques sphériques Y_l^m constituent une base orthonormée des fonctions sphériques. Elles sont identifiées par deux indices : leur fréquence l , et m , l’indice de la fonction au sein de la bande de fréquence l ($-l \leq m \leq l$). Pour calculer la décomposition du polynôme $p_{\{u_1, -u_1\}} : s \mapsto \langle u_1, s \rangle^4$ dans cette base, on se propose d’utiliser une astuce que l’on trouve par exemple dans [Sloan, 2008] qui s’appuie sur les harmoniques zonales. Ces dernières sont des harmoniques qui possèdent une symétrie axiale. Il est très facile de calculer la décomposition d’une rotation d’une harmonique zonale. $p_{\{u_1, -u_1\}}$ possédant une symétrie autour de l’axe défini par u_1 , il nous suffit alors de calculer la décomposition pour une seule valeur de u_1 . Prenons comme valeur, le vecteur unitaire $e_z = (0, 0, 1)$ pointant dans la direction de l’axe Z. Un polynôme en z peut en effet facilement être décomposé en harmoniques sphériques, car sa décom-

position ne fait intervenir que les harmoniques d'indice 0 dans chaque bande de fréquence. On obtient alors par résolution d'un petit système linéaire à trois inconnus :

$$p_{\{e_z, -e_z\}} = \frac{2\sqrt{\pi}}{105} (21Y_0^0 + 12\sqrt{5}Y_2^0 + 8Y_4^0) \quad (3.28)$$

En notant Z_u^l l'harmonique zonale d'ordre l possédant une symétrie autour de l'axe u (exemple : $Z_{e_z}^l = Y_4^0$), on peut se servir de l'identité suivante :

$$Z_u^l = \sqrt{\frac{4\pi}{2l+1}} \sum_{m=-l}^l Y_l^m(u) Y_l^m \quad (3.29)$$

Ce qui nous donne :

$$p_{\{u_1, -u_1\}} = \frac{4\pi}{315} \left(63Y_0^0(u_1)Y_0^0 + 36 \sum_{m=-2}^2 Y_2^m(u_1)Y_2^m + 8 \sum_{m=-4}^4 Y_4^m(u_1)Y_4^m \right) \quad (3.30)$$

On peut à nouveau définir les coefficients de cette décomposition à un facteur multiplicatif près comme suit :

$$\begin{cases} c_{2,m}(u) &= \sum_{i=1}^3 Y_2^m(u_i) \\ c_{4,m}(u) &= \sum_{i=1}^3 Y_4^m(u_i) \end{cases} \quad (3.31)$$

Tout comme pour le cas 2D, la décomposition ne fait intervenir que des termes d'ordre 2 et 4. Le facteur multiplicatif entre les termes d'ordre 2 et 4 n'est pas respecté par rapport à (3.30), car nous allons à nouveau introduire un paramètre d'orthogonalité λ . Il est également possible de vérifier dans le cas où les vecteurs u_i seraient unitaires que le repère est orthogonal si et seulement si les coefficients d'ordre 2 sont nuls. Dans le cas orthogonal, on retrouve donc les propriétés basiques des méthodes de champs de croix [Huang et al., 2011, Ray et al., 2016] : à savoir que les polynômes de représentation des croix de degré 4 se décomposent avec uniquement des harmoniques sphériques d'ordre 4. Toutefois, nous donnons pour la première fois une relation simple et exacte entre la déformation du repère donnée par u et la décomposition du polynôme associé dans la base des harmoniques sphériques.

3.4.2 Conception de champs de repères pour le remailage hexaédrique

Contrairement au cas 2D, nous n'avons pas trouvé de changement de variables évidents pour simplifier les expressions des coefficients de (3.31). Nous proposons alors de garder les vecteurs u_i comme variables. Comme les travaux précédents [Huang et al., 2011, Ray et al., 2016, Palmer et al., 2020], calculer des champs de directions lisses requiert de résoudre des problèmes d'optimisation non convexes.

Énergie de lissage Tout comme pour le cas 2D (3.15), l'énergie de lissage E_D^{3D} est définie comme une énergie de Dirichlet avec un paramètre λ permettant de contrôler l'importance des coefficients d'ordre 2.

$$\begin{aligned} E_D^{3D}(u) &= \lambda^2 \sum_{(i,j) \in \mathcal{E}} \sum_{m=-2}^2 (c_{2,m}(u_i) - c_{2,m}(u_j))^2 \\ &\quad + \sum_{(i,j) \in \mathcal{E}} \sum_{m=-4}^4 (c_{4,m}(u_i) - c_{4,m}(u_j))^2 \end{aligned} \quad (3.32)$$

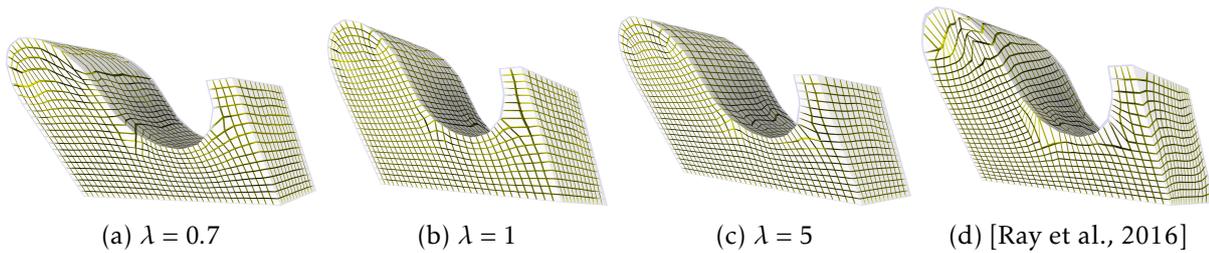


FIGURE 3.10 – Résultats de notre méthode de génération de champs de repères 3D pour des valeurs croissantes du paramètre d’orthogonalité λ . Les faibles valeurs (gauche) poussent les singularités hors du domaine alors que les grandes valeurs (droite) produisent un résultat proche du champ de croix produit par [Ray et al., 2016].

Paramètre d’orthogonalité Toujours comme dans le cas 2D, l’orthogonalité des repères est liée aux coefficients d’ordre 2. On pénalise alors les coefficients de la bande de fréquence $l = 2$ avec un poids λ^2 dans l’énergie (3.32). Augmenter λ^2 pénalise les changements d’angles entre les directions des repères favorisant ainsi les champs avec une différence d’angles constante. Comme les singularités impliquent une énergie plus faible lorsque les repères sont orthogonaux autour d’elles, une grande valeur de λ^2 force les repères à être orthogonaux à travers tout le champ.

La figure 3.10 illustre l’effet de λ sur la génération de maillages hexaédriques. Les petites valeurs produisent des paramétrisations similaires à celles des polycubes avec les singularités sur le bord du domaine alors que les grandes valeurs permettent d’approcher les champs de croix générés par [Ray et al., 2016].

Contraintes de direction Les contraintes de directions peuvent facilement être imposées en fixant u_1 , u_2 ou u_3 . On peut fixer qu’une seule direction, deux ou bien trois à la fois. Le choix de quelle variable est fixé n’a pas d’impact sur le résultat comme notre représentation est invariante par réordonnancement des directions. Cela est très pratique pour les contraintes de bords des paramétrisations globales. Dans toutes nos expérimentations, un repère associé à une cellule de bord à certaines de ces directions fixées aux normales du bord. Si un tétraèdre possède un, deux ou trois triangles de bord alors le repère associé doit satisfaire une, deux ou trois contraintes.

3.4.3 Remaillage hexaédriques de modèles CAO

Nous évaluons la qualité de nos champs de repères 3D par leur capacité à générer des maillages hexaédriques. Dans toutes nos expérimentations, les maillages hexaédriques sont extraits en utilisant des méthodes de l’état de l’art [Nieser et al., 2011, Lyon et al., 2016].

L’optimisation des champs de repères 3D est faite avec une méthode MBO de la même manière que pour les champs de repères 2D (algorithme 1). Nous alternons entre des étapes de diffusion avec les paramètres $\tau = 1$ et $\epsilon = 0.01$, et des étapes de renormalisation. Les champs de croix, eux, sont générés en utilisant [Ray et al., 2016].

Réussir à mailler correctement les coins à angles aigus est un défi classique des méthodes de remaillage hexaédrique. Leur présence est fréquente dans les modèles CAO et les champs de croix échouent souvent à produire des hexaèdres dans ces situations comme l’illustre par exemple la figure 3.11a. À l’inverse, le champ de repère obtenu par notre méthode introduit

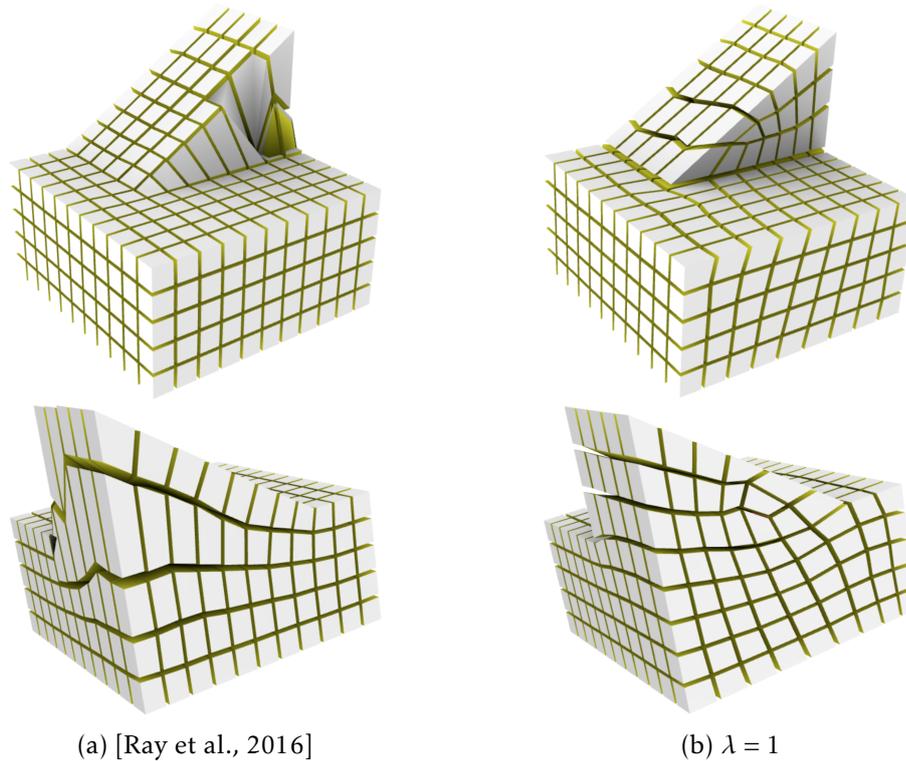


FIGURE 3.11 – Sur un modèle CAO comportant un coin à angle saillant, un champ de repère orthogonal 3D provoque un écrasement d’une partie du modèle (Fig. 3.11a). Notre méthode réussit à ajouter une singularité qui compense la pente (Fig. 3.11b).

une singularité rendant l’extraction d’un maillage hexaédrique possible (Fig. 3.11b). Pour produire de tels résultats, il est primordial que les repères soient entièrement contraints dans les coins saillants.

La figure 3.12 met également en lumière la nécessité de construire des champs de repères non orthogonaux pour capturer correctement les coins à angles aigus. Les champs de croix produisent des maillages hexaédriques avec des éléments énormément distordus, voire manquants, alors que notre méthode est capable de placer des éléments valides dans ces coins.

Dans la figure 3.13, un modèle oblique force le champ de croix à osciller pour satisfaire les contraintes de bord, tandis que notre champ de repères peut rester constant.

Initialisation L’énergie E_D^{3D} 3.32 est non convexe. Sa minimisation dépend alors fortement de son initialisation. La figure 3.14 compare les solutions de notre processus d’optimisation pour trois différentes initialisations. Expérimentalement, nous avons observé qu’une initialisation par un champ de croix aléatoire converge systématiquement vers une même solution (Fig. 3.14b). Partir d’un champ de croix lisse [Ray et al., 2016] requiert moins d’itérations pour converger, mais la solution est toujours la même Fig. 3.14a). Comme indiqué dans le tableau 3.2, cela réduit (parfois considérablement) le temps de calcul. Cependant, notre optimisation peut tomber dans des minimums locaux lorsque l’initialisation est faite à partir d’un champ de repères aléatoire incluant des repères presque dégénérés (Fig. 3.14c). Après la convergence, l’énergie de Dirichlet E_D^{3D} est plus élevée que pour la solution optimale. De plus, le champ obtenu contient des repères quasi dégénérés (entourés en rouge) qui rendent le résultat inuti-

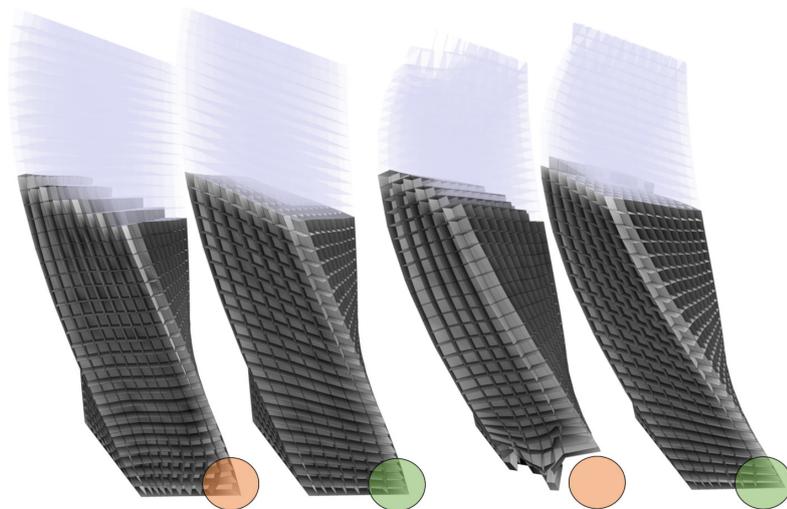


FIGURE 3.12 – Résultats sur un carré extrudé avec une torsion. Le premier résultat est obtenu à partir d’un champ de croix, tandis que le second est produit avec notre méthode. Les deux images suivantes ajoutent simplement plus de torsion lors de l’extrusion du modèle. Les coins saillants (mis en valeur par les disques) sont mieux capturés avec nos champs. On peut également noter que les couches d’hexaèdres à l’intérieur du volume restent parallèles aux extrémités.

lisible pour du remaillage hexaédrique.

Dans toutes nos autres expérimentations, notre initialisation est un champ de croix lisse obtenu avec [Ray et al., 2016].

Figure	Tétraèdres	Initialisation	Optimisation	Maillage
3.10	13 225	0.349	454	8.4
3.12	4 420	0.096	31	3.0
3.13	10 788	0.332	305	7.7
3.14a	862	0.016	22	0.7
3.14b	862	0.000	25	0.7
3.14c	862	0.000	26	—

TABLE 3.2 – Temps d’exécution en seconde pour chaque étape de notre méthode de conception de champs de repères 3D : initialisation avec un champ de croix lisse [Ray et al., 2016], optimisation non convexe, extraction de maillage hexaédrique [Nieser et al., 2011, Lyon et al., 2016].

Temps d’exécution La table 3.2 liste les temps d’exécution pour générer des maillages hexaédriques avec notre méthode sur les différents modèles 3D illustrés précédemment. À nouveau, l’étape prenant le plus de temps est le calcul du champ de repères non orthogonaux comme elle requiert de résoudre un problème d’optimisation non convexe. Toutes les expérimentations ont été conduites sur un ordinateur portable avec un processeur 4 cœurs, Intel Core i5-10400H cadencé à 2.6 GHz et 32 Gb de RAM. Tous les algorithmes sont implémentés en C++.

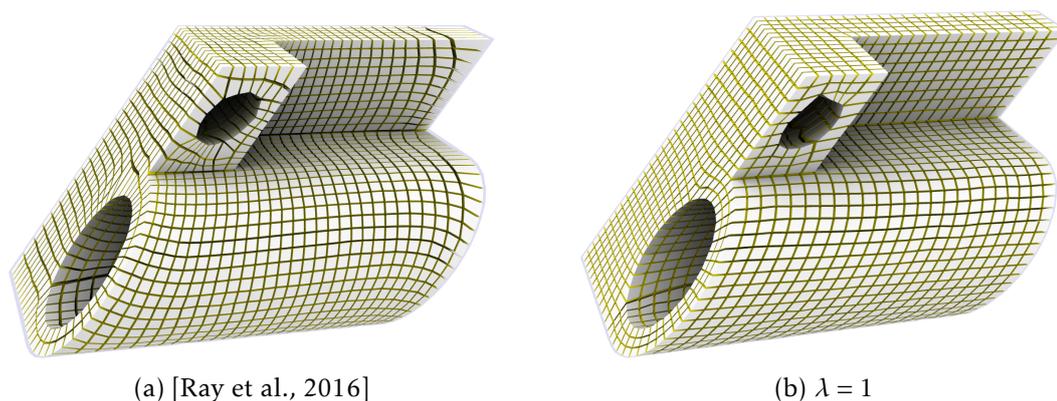


FIGURE 3.13 – Sur un modèle CAO ayant subi une transvection, un champ de croix 3D compense la déformation en créant des oscillations non nécessaires (Fig. 3.13a). Notre méthode produit des hexaèdres bien formés et réguliers (Fig. 3.13b).

3.5 Conclusion

Dans ce chapitre, nous avons introduit une représentation des champs de repères commune aux surfaces et aux volumes. Cette dernière permet de contraindre de manière indépendante autant de directions que nécessaire par repère, et permet également de contrôler de façon assez intuitive l’orthogonalité des repères.

Bien que notre processus d’optimisation soit non convexe et plus difficile que les méthodes standards de champs de croix, il autorise plus de flexibilité sur les contraintes que l’on peut appliquer, donnant ainsi de meilleurs résultats pour des problèmes de paramétrisation globale.

À l’avenir, il pourrait être intéressant de relaxer la contrainte imposant que les directions u_i soient unitaires. Cela a été fait pour les champs de croix, en 2D avec [Knöppel et al., 2013], puis en 3D avec la variété algébrique des repères Odeco [Palmer et al., 2020]. Dans les deux cas, il y est montré l’utilité d’une telle relaxation particulièrement autour des singularités, où il est souhaitable que la norme des directions u_i soit plus faible. De plus cette relaxation est nécessaire pour un autre objectif futur qui serait de générer des champs de repères intégrables, *i.e.* des champs dont les directions correspondent aux gradients de fonctions scalaires. L’intégrabilité est fondamentale pour garantir l’existence d’une carte de paramétrisation globale bijective ayant la topologie du champ. La génération de champs de directions intégrables a été étudiée pour les surfaces [Diamanti et al., 2015] mais il n’existe toujours pas de travaux à ce sujet pour les volumes. Avec notre représentation, nous pouvons avoir espoir que l’intégrabilité pourra être étendue à la 3D.

Malgré tout, un point noir de la méthode présentée est son temps de calcul très élevé comparé à d’autres méthodes. En pratique, dans l’équipe Pixel, nous préférons utiliser une publication de l’équipe survenue quelques mois plus tard [Desobry et al., 2021b] permettant de gérer correctement les coins à angles aigus en résolvant un simple problème linéaire.

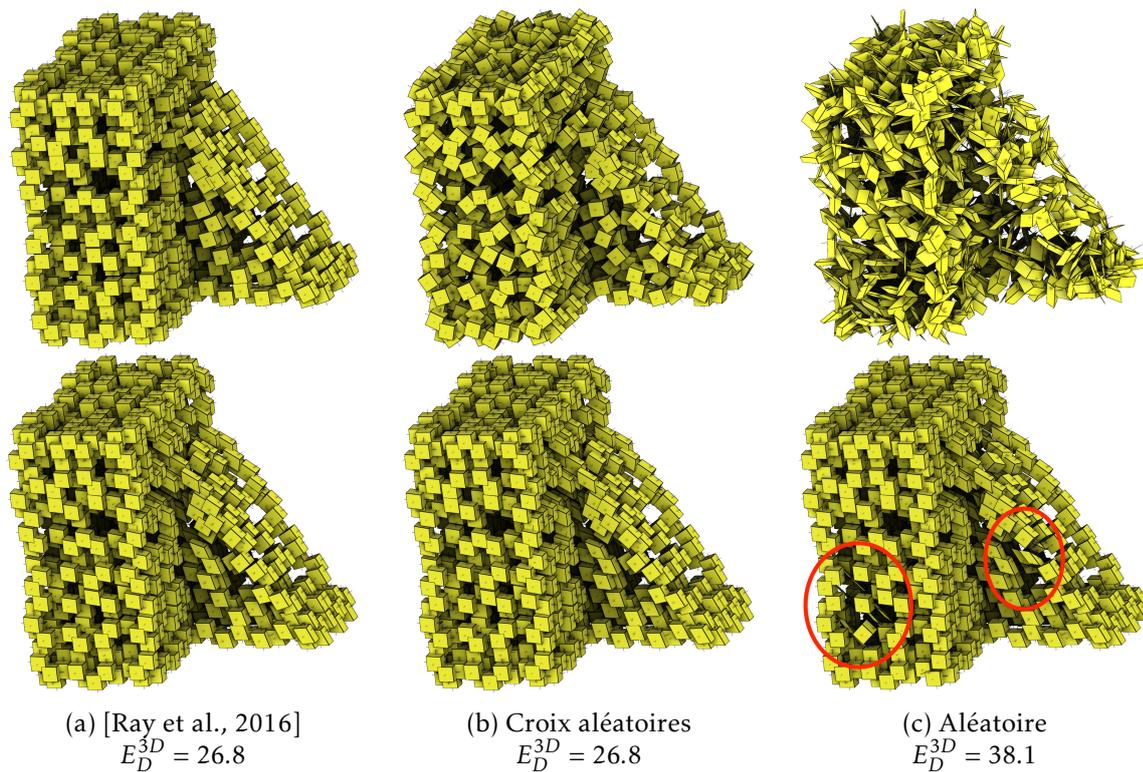


FIGURE 3.14 – Résultats de notre optimisation de champs de repères 3D pour trois différentes initialisations. Partir d'un champ de croix lisse [Ray et al., 2016] (Fig. 3.14a, haut) ou depuis un champ de croix aléatoire (Fig. 3.14b, haut) mène à la même solution (Fig. 3.14a-3.14b, bas). Une initialisation avec un champ de repères complètement aléatoire, incluant des repères quasi dégénérés, peut rester coincé dans un minimum local dégénéré résultant en une énergie plus élevée E_D^{3D} (Fig. 3.14c).

Chapitre 4

Quantification de cartes sans coutures [Coudert-Osmont et al., 2023]

Dans ce chapitre, nous proposons une nouvelle méthode de quantification de cartes sans coutures sur des surfaces. Les cartes préservant la grille unité sur les surfaces triangulées sont généralement obtenues en résolvant un problème faisant apparaître des variables à valeurs entières et réelles. Une stratégie répandue pour la résolution de ce type de problème est dans un premier temps, d’ignorer les contraintes entières, puis de renforcer ces contraintes dans un second temps par une étape dite de quantification. Les algorithmes de quantification actuels s’appuient sur une partition de la surface sous forme d’un maillage en T. Nous proposons à la place d’opérer sur une décimation de la surface triangulée initiale. Nous revendiquons que cela simplifie l’implémentation, et permet plus facilement de s’adapter à des contraintes, telles que les bords libres ou l’alignement à des réseaux de courbes caractéristiques complexes. Ce chapitre a donné lieu à une publication dans le journal *Computer Graphics Forum* et une présentation dans la conférence internationale *Eurographics* :

Coudert-Osmont, Y., Desobry, D., Heistermann, M., Bommès, D., Ray, N., and Sokolov, D. (2023). Quad mesh quantization without a t-mesh. Computer Graphics Forum, 43

4.1 Introduction

Les méthodes de remaillage quadrangulaire par paramétrisation globale consistent en la construction d’une carte de la surface vers $\mathbb{R}^2 \cong \mathbb{C}$ de sorte que l’image inverse de la grille unité par cette carte résulte en un maillage quadrangulaire de la surface. Une telle carte est appelée carte préservant la grille. Pour caractériser cette notion, nous utilisons une idée très simple : la grille unité est invariante par translation entière et rotation d’angle multiple de 90° . Une *carte préservant la grille* sur un maillage triangulaire est alors une carte linéaire par triangle, telle que les images d’une arête incidente à deux triangles soient identiques à une composition près avec une translation entière et une rotation d’un angle multiple de 90° (Fig. 4.1c). Similairement, on définira une *carte sans coutures* comme une carte telle que les images d’une arête incidente à deux triangles soient identiques à une composition près par une translation arbitraire et une rotation d’un angle multiple de 90° (Fig. 4.1b).

Notre défi est alors de produire pour une surface, une carte U préservant la grille avec le moins de distorsion possible. Cela est généralement formulé avec un problème d’optimisation, où l’on cherche à minimiser une fonction objectif $f(U)$ sous des contraintes de préservation de la grille. Il y a de nombreuses façons de définir la fonction objectif, mais la plupart inclue

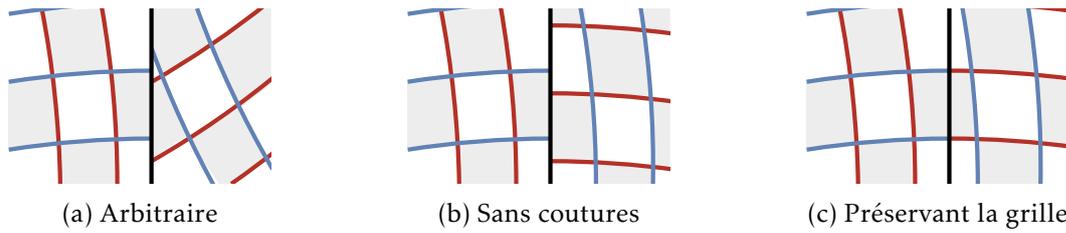


FIGURE 4.1 – Classification des cartes selon leur comportement le long de discontinuités (en noir).

une estimation de la distorsion. Par exemple, cela peut être la distance entre les colonnes de la jacobienne de U et les directions d'un champ de repères donné en entrée [Kälberer et al., 2007, Bommès et al., 2009]. Dans certains cas, il est préférable d'utiliser des fonctions plus élaborées comme [Garanzha et al., 2021] dont la minimisation aboutie plus souvent à une carte valide (c'est-à-dire avec un jacobien strictement positif).

4.2 Cartes préservant la grille

Dans cette section, nous définissons formellement les contraintes qui doivent être satisfaites par une carte U pour préserver la grille. Pour cela, nous utilisons une structure de demi-arêtes [Muller and Preparata, 1978] (Fig. 4.2a) et nous représentons la carte $U = (U_1 \cdots U_H)^\top$ par les coordonnées U_h à l'origine de la demi-arête h . H désigne le nombre total de demi-arêtes dans la triangulation en entrée. On décide de représenter les coordonnées U_h par un complexe plutôt qu'un vecteur de \mathbb{R}^2 de sorte qu'une rotation de 90° soit une multiplication par l'unité imaginaire i . Afin de caractériser les discontinuités autorisées pour U à travers les arêtes, nous faisons intervenir de nouvelles variables. Pour une demi-arête h , la fonction de transition entre sa face et sa face opposée est définie par un entier de Gauss t_h représentant la translation entière et une rotation r_h dans l'ensemble $\{1, i, i^2, i^3\}$ (Fig. 4.2b).

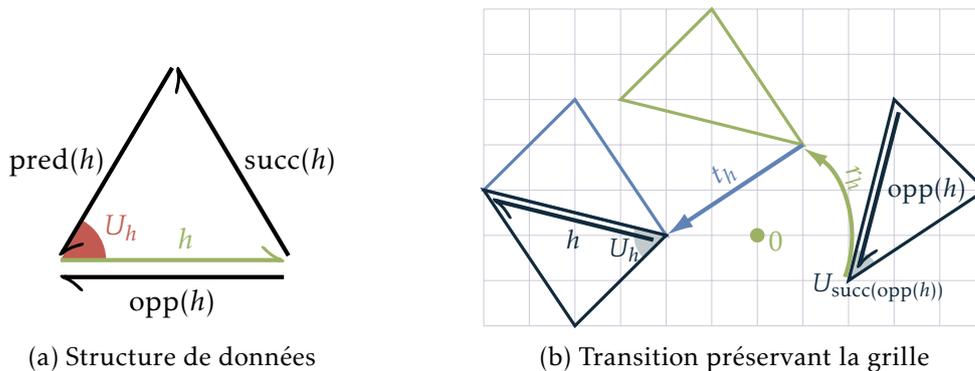


FIGURE 4.2 – Notations : Structure de demi-arêtes (gauche), et une fonction de transition préservant la grille à travers une demi-arête h donnée par une rotation r_h et une translation entière t_h (droite).

4.2.1 Formalisation (irréalisable)

Pour cette formulation, notre problème d'optimisation possède le jeu de variables suivant : les coordonnées de la carte $\{U_h\}_{h=1}^H$, les rotations $\{r_h\}_{h=1}^H$ et les translations $\{t_h\}_{h=1}^H$. On peut ensuite définir les conditions que doivent satisfaire nos variables pour que la carte U préserve la grille :

- La première condition porte sur la fonction de transition qui relie une demi-arête $h \in \{1, \dots, H\}$ à son opposée $\text{opp}(h)$

$$\begin{cases} U_h &= r_h U_{\text{succ}(\text{opp}(h))} + t_h \\ U_{\text{succ}(h)} &= r_h U_{\text{opp}(h)} + t_h \\ r_h &\in \{1, i, i^2, i^3\} \end{cases} \quad (4.1)$$

- La seconde condition est que certaines variables doivent être entières. Premièrement, les translations doivent être entières. Bien que cela suffise à obtenir une carte préservant la grille, dans le cas d'une application au remaillage quadrangulaire, il est également nécessaire que les singularités se trouvent à des coordonnées entières. En effet, lorsque les angles des coins de triangles (au sein de la carte) incidents à un sommet v ne somment pas à 360° , le sommet v est dit singulier et doit nécessairement avoir des coordonnées entières afin de produire un sommet irrégulier dans le maillage quadrangulaire final. Nous exprimons ces contraintes comme suit

$$\begin{cases} t_h \in \mathbb{Z}[i] & \forall h \in \{1, \dots, H\} \\ U_h \in \mathbb{Z}[i] & \text{si l'origine de } h \text{ est une singularité} \end{cases} \quad (4.2)$$

Où $\mathbb{Z}[i] = \{a + ib \mid a, b \in \mathbb{Z}\}$ est l'extension de \mathbb{Z} donnant les entiers de Gauss.

Finalement, la génération d'une carte préservant la grille peut être formulée comme le problème d'optimisation suivant :

$$\arg \min_{\{U_h, r_h, t_h\}_{h=1}^H} f(U) \quad \text{sous les contraintes (4.1) et (4.2)}. \quad (4.3)$$

4.2.2 Rendre le problème accessible, étape N°1 : champ de repères

Bien qu'élégant, le problème dans sa forme générale (4.3) ne semble pas abordable. Mélanger des variables à valeurs entières et réelles, et plus particulièrement, un grand nombre de variables entières rend le problème extrêmement difficile à résoudre. En pratique, la vaste majorité des tentatives pour résoudre ce problème réduit l'espace de recherche en décomposant le problème en deux parties. La première partie définit l'orientation désirée des quadrilatères (le champ de repères), et la seconde, produit une carte telle que les isovaleurs des coordonnées de la carte soient alignées avec le champ de repères.

L'idée est alors de réduire drastiquement le nombre de variables entières dans le problème (4.3) en calculant un champ de repères \mathcal{F} [Vaxman et al., 2016, Desobry et al., 2021b] sur la triangulation en entrée. Cela permet de fixer toutes les rotations $\{r_h\}$, de sorte qu'elles ne soient plus des variables de notre problème. On peut même aller plus loin et éliminer une large partie des translations $\{t_h\}$. En ayant calculé un champ de repères, nous avons également fixé les sommets singuliers. Il est alors possible de couper la surface en un disque topologique le long d'un ensemble de demi-arêtes appelé *graphe de coupure* $\mathcal{C} \subset \{1, \dots, H\}$. L'idée est d'obtenir un disque topologique avec toutes les singularités de \mathcal{F} situées sur le bord [Kälberer et al., 2007]. Cela permet de fixer toutes les translations t_h et rotation r_h à respectivement zéro et un, à travers

toutes les demi-arêtes qui n'appartiennent pas au graphe de coupure \mathcal{C} (Fig. 4.3). De cette façon, la condition $t_h \in \mathbb{Z}[\iota]$ est satisfaite pour toutes les demi-arêtes concernées. Formellement, cela revient à :

$$\begin{cases} t_h = 0 & \forall h \notin \mathcal{C} \\ r_h = 1 & \forall h \notin \mathcal{C} \\ r_h \text{ contraint par } \mathcal{F} & \forall h \in \mathcal{C} \end{cases} \quad (4.4)$$

Pour résumer, par la suite nous serons intéressés par le problème suivant :

$$\arg \min_U f(U) \quad \text{sous les contraintes (4.1), (4.2) et (4.4).} \quad (4.5)$$

À noter que les translations t_h pour $h \in \mathcal{C}$ ne sont plus des degrés de liberté dans cette formulation, car elles peuvent être exprimées à partir des coordonnées de carte $\{U_h\}_{h=1}^H$ et des rotations $\{r_h\}_{h=1}^H$.

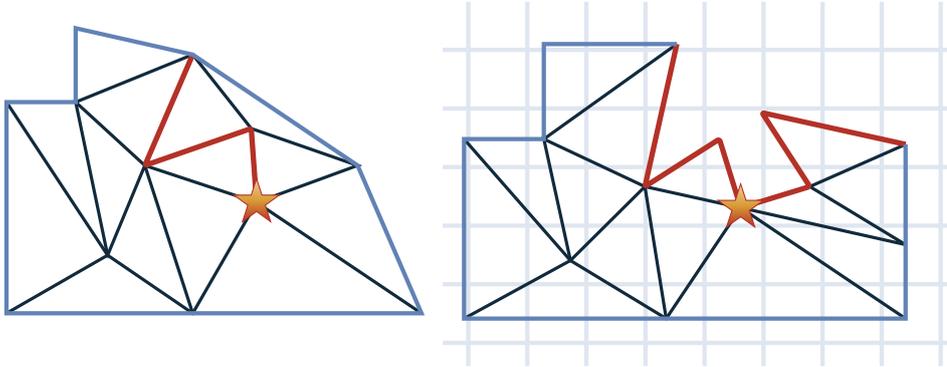


FIGURE 4.3 – Le maillage triangulaire (gauche) admet une carte sans coutures (droite) qui est discontinue à travers le graphe de coupure \mathcal{C} (arêtes rouges). Ici, les discontinuités sont caractérisées par une rotation de $\pi/2$ autour de la singularité (étoile).

4.2.3 Rendre le problème accessible, étape N°2 : quantification

Le problème d'optimisation (4.5) consiste à minimiser une fonction en étant sujet à des contraintes linéaires (discontinuités dans la carte, alignement à des courbes caractéristiques) et des variables entières (pour les singularités, mais aussi pour les courbes caractéristiques qui doivent avoir une coordonnée entière dans la carte afin d'être représentées par une succession d'arêtes dans le maillage quadrangulaire final). La difficulté avec un tel problème d'optimisation est qu'il requiert à la fois une optimisation combinatoire pour les variables entières, mais aussi une optimisation numérique pour les autres variables à valeurs réelles. Dans le cas des calculs de cartes préservant la grille, il est courant d'utiliser des algorithmes fondés sur la quantification pour séparer la partie combinatoire de la partie optimisation numérique en différentes étapes. Il est possible de se référer à l'algorithme 2 pour une description typique de cette méthode. Une carte sans coutures (Fig. 4.1b) est obtenue par une optimisation numérique sans considérer les contraintes entières. Ensuite, l'étape de quantification trouve un ensemble de contraintes affines qui permet de fixer tous les degrés de liberté entières. Finalement, ces nouvelles contraintes sont utilisées pour remplacer les anciennes contraintes d'entiereté dans une seconde optimisation numérique.

Dans ce chapitre, nous nous occupons de l'étape de quantification : étant donnée une carte sans coutures, nous cherchons des contraintes affines qui permettraient de rendre entiers les degrés de liberté qui doivent l'être tout en minimisant la distorsion que ces nouvelles contraintes imposeraient.

Algorithme 2 : Génération d'une carte préservant la grille

Entrées : Un maillage triangulaire représenté par des demi-arrêtes $\{h\}_{h=1}^H$, un champ de repères \mathcal{F} et un graphe de coupure \mathcal{C}
Sorties : Une carte préservant la grille U^{PG}

```

/* Calcul d'une carte sans coutures: ignore les contraintes entières */
1  $U^{\text{SC}} \leftarrow \underset{U}{\operatorname{argmin}} f(U)$  sous les contraintes (4.1) et (4.4)

/* Recherche de contraintes affines pour fixer les degrés de liberté entiers */
2  $[A, \omega] \leftarrow \text{quantifie}(\mathcal{F}, U^{\text{SC}})$ 

/* Calcul d'une carte préservant la grille: optimisation numériques sous
   contraintes affines */
3  $U^{\text{PG}} \leftarrow \underset{U}{\operatorname{argmin}} f(U)$  sous les contraintes  $AU = \omega$ , (4.1) et (4.4)

```

4.3 État de l'art de la quantification

Arrondissement Les premiers algorithmes de quantification, [Kälberer et al., 2007] et [Bommes et al., 2009], arrondissent simplement certaines variables à leur entier le plus proche. Cependant, l'existence et la qualité d'une carte valide sujette à ces contraintes ne sont pas garanties. En plus de cela, on peut également remarquer que [Kälberer et al., 2007] réduit son espace de recherche en arrondissant non pas à l'entier le plus proche, mais à l'entier pair le plus proche. La parité y est nécessaire, car les variables contraintes à être entières garantissent uniquement que la carte préserve la grille, mais pas que les singularités aient des coordonnées entières. L'arrondissement glouton, en alternant entre optimisation numérique et arrondissements de [Bommes et al., 2009] améliore les résultats, mais dans le cas général, de telles méthodes ne sont pas robustes.

Séparation et évaluation Il est possible de concevoir une méthode robuste avec un algorithme par séparation et coupe (*Branch-and-Cut*) [Pardalos and Resende, 2002] qui combine méthode de séparation et évaluation et méthode des plans sécants. Cependant, la fonction d'évaluation requiert une optimisation numérique coûteuse qui peut potentiellement être appelée un nombre de fois exponentiel en le nombre de variables entières. Pour rendre cette méthode raisonnable, [Bommes et al., 2013] propose de l'appliquer sur une décimation du maillage triangulaire puis de finalement propager le résultat vers le maillage initial. Cette approche est particulièrement lente, mais permet toutefois de produire des maillages quadrangulaires grossiers.

Quantized Global Parametrization (QGP) La méthode de référence pour la quantification [Campen et al., 2015] prend un autre point de vue. À partir d’une carte sans couture, elle procède en trois étapes :

1. *Construction d’un problème de substitution plus simple* (Algorithme 3, lignes 1 et 2). Un graphe de motocyclettes [Eppstein et al., 2008] est construit en traçant simultanément les courbes isovaleurs des coordonnées de la carte en partant de chaque singularité de la paramétrisation. Les courbes s’arrêtent lorsqu’elles en intersectent une autre, produisant alors une jonction en T. Elles forment alors un maillage en T, découpant et partitionnant la surface en des régions courbes de forme rectangulaire alignées avec les axes du domaine paramétrique.
2. *Optimisation des degrés de liberté entiers* (Algorithme 3, ligne 3). L’étape de quantification assigne des longueurs entières à chaque arc du graphe de motocyclettes, ou, de manière équivalente, détermine la taille de chaque région rectangulaire dans la carte. L’opération atomique de ce processus est d’augmenter/diminuer d’une unité, la longueur de chaque arc du graphe de motocyclette traversé par un cycle parcourant des régions rectangulaires en ressortant toujours par un arc parallèle et opposé à celui par lequel le cycle est rentré (Fig. 4.4). Un tel cycle est calculé avec l’algorithme de Dijkstra pour être le plus court possible selon des poids fabriqués pour favoriser des longueurs d’arcs proches des longueurs au sein de la carte sans couture. Les longueurs des arcs du graphe de motocyclettes sont initialisées à zéro, puis elles sont itérativement augmentées jusqu’à ce qu’il n’y ait plus de chemins de longueur zéro rejoignant deux singularités. Une fois que cette dernière condition est validée, QGP continue d’appliquer des opérations atomiques tant qu’elles ne violent pas le critère de validité, et n’augmente pas une énergie conçue pour favoriser de faibles différences entre les longueurs assignées par QGP et celles correspondantes dans la carte sans couture.
3. *Déduction des contraintes pour le problème initial* (Algorithme 3, ligne 4). Une carte préservant la grille peut alors être calculée en résolvant le problème de carte sans coutures avec des contraintes additionnelles fixant à des valeurs entières trouvées par la quantification, les différences de coordonnées le long de chemins reliant des paires de singularités.

Construits sur la base de QGP, des travaux plus récents permettent de borner la déviation angulaire entre le champ de repères et la carte préservant la grille [Lyon et al., 2021a], de supporter des bords libres [Lyon et al., 2019], ou même de déplacer/supprimer des singularités [Lyon et al., 2021b].

Algorithme 3 : Quantification — version maillage en T

Entrées : Un maillage triangulaire \mathcal{M} et une carte sans coutures U^{SC}
Sorties : Un système linéaire A, ω tel que $AU = \omega$ implique (4.2).
// Construction d’un problème de substitution plus simple
1 $\mathcal{M}^{\text{sub}} \leftarrow$ motocyclettes $(\mathcal{M}, U^{\text{SC}})$
2 $U^{\text{sub}} \leftarrow$ propagation de carte $(\mathcal{M}, \mathcal{M}^{\text{sub}}, U^{\text{SC}})$
// Optimisation des degrés de liberté entiers
3 $\{l_h\}_{h=1}^{H^{\text{sub}}} \leftarrow$ calcul des longueurs $(\mathcal{M}^{\text{sub}}, U^{\text{sub}})$
/ Déduction des contraintes pour le problème initial */*
4 $[A, \omega] \leftarrow$ contraintes $(\mathcal{M}, \mathcal{M}^{\text{sub}}, \{l_h\}_{h=1}^{H^{\text{sub}}})$

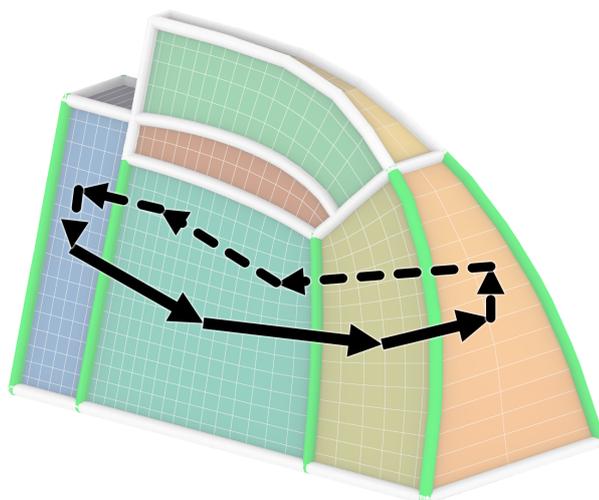


FIGURE 4.4 – Opération atomique de QGP : pour un cycle (noir) du graphe d’adjacence du partitionnement en régions rectangulaires, on augmente la longueur de chaque arc du graphe de motocyclettes qui est traversé par le cycle (vert).

Notre contribution La contribution principale de ce chapitre est l’adaptation du cadre de la méthode QGP à un proxy plus simple : au lieu de nous appuyer sur un maillage en T, nous proposons d’effectuer une décimation du maillage d’entrée et de directement travailler sur un maillage triangulaire. L’objectif est donc de disposer d’une méthode de quantification dont la robustesse peut être prouvée et qui produit des résultats similaires à ceux de QGP en termes de vitesse et de qualité. Notre intuition derrière ce changement est qu’un maillage triangulaire peut être un proxy plus polyvalent qu’un maillage en T. Dans la section 4.6, nous discutons de plusieurs extensions possibles qui pourraient bénéficier de notre adaptation.

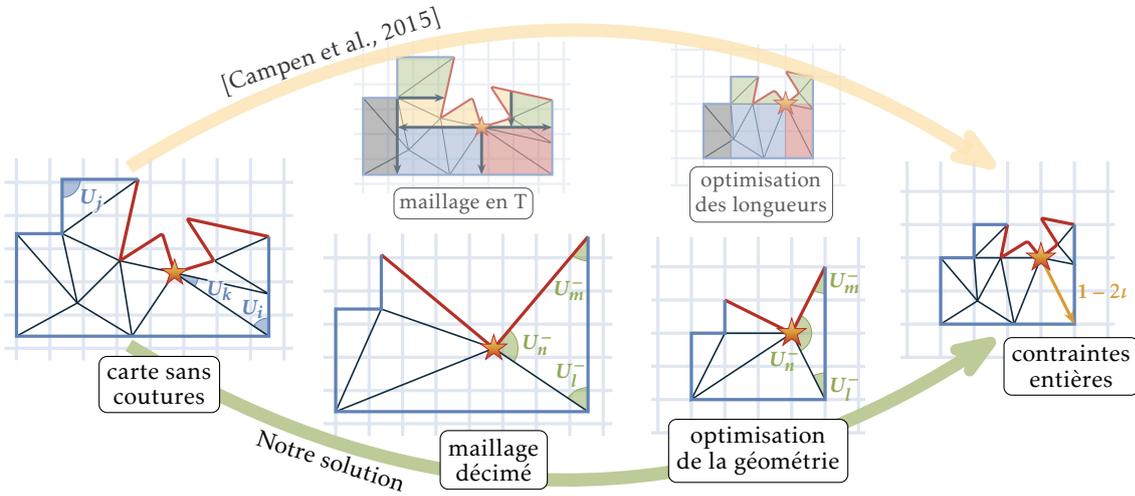
4.4 Notre quantification

Dans cette section, nous proposons une nouvelle méthode de quantification qui est une adaptation de QGP où le maillage en T est remplacé par un maillage triangulaire décimé (voir la différence entre les algorithmes 3 et 4). Notre algorithme suit les mêmes trois étapes que QGP ; la suite de cette section se concentre sur chacune de ces étapes. La figure 4.5 fournie une illustration de la méthode et ses différentes étapes :

1. La première étape (§4.4.1) décime le maillage triangulaire et produit une matrice D permettant d’obtenir les coordonnées de carte U_h^- sur le nouveau maillage décimé, lorsque multiplié par les coordonnées de la carte initiale U .
2. La seconde étape (§4.4.2) détermine les nouvelles géométries quantifiées (avec des entiers de Gauss) $\omega_h = U_{\text{succ}(h)}^- - U_h^-$ pour chaque demi-arête h du maillage décimé.
3. La dernière étape (§4.4.3) enchaîne les deux relations pour obtenir les contraintes qui forceront la carte à préserver la grille sur le maillage triangulaire initial.

Algorithme 4 : Quantification — version maillage décimé

Entrées : Un maillage triangulaire \mathcal{M} et une carte sans coutures U^{SC}
Sorties : Un système linéaire A, ω tel que $AU = \omega$ implique (4.2).
// Construction d'un problème de substitution plus simple (§4.4.1)
1 $\mathcal{M}^{sub}, D \leftarrow$ décimation (\mathcal{M}, U^{SC})
2 $U^{sub} \leftarrow D \cdot U^{SC}$
// Optimisation des degrés de liberté entiers (§4.4.2, Algorithme 5)
3 $\{\omega_h\}_{h=1}^{H^{sub}} \leftarrow$ calcul des géométries $(\mathcal{M}^{sub}, U^{sub})$
/ Déduction des contraintes pour le problème initial (§4.4.3) */*
4 $A_h \leftarrow (S^{sub} - I)D$



Étape	§4.4.1 Décimation	§4.4.2 Optimisation des degrés de liberté entiers	§4.4.3 Extraction des contraintes entières
Équation	$U_h^- = D_h U$	$U_{next(h)}^- - U_h^- = \omega_h$	$(D_{next(h)} - D_h)U = \omega_h$
Exemple	$\begin{cases} \dots \\ U_i^- = U_i \\ U_m^- = U_k - \iota(U_j - U_k) \\ U_n^- = U_k \\ \dots \end{cases}$	$\begin{cases} \dots \\ U_m^- - U_l^- = 4\iota \\ U_n^- - U_m^- = -1 - 2\iota \\ U_l^- - U_n^- = 1 - 2\iota \\ \dots \end{cases}$	$\begin{cases} \dots \\ U_k - \iota(U_j - U_k) - U_i = 4\iota \\ U_k - (U_k - \iota(U_j - U_k)) = -1 - 2\iota \\ U_i - U_k = 1 - 2\iota \\ \dots \end{cases}$

FIGURE 4.5 – Notre algorithme illustré sur le maillage de la figure 4.3. La méthode de référence [Campen et al., 2015] (chemin du dessus) génère un maillage en T, optimise ses longueurs d’arêtes et les convertit en contraintes entières. Au lieu de cela, nous proposons (chemin du bas) de décimer le maillage d’entrée, optimiser la géométrie de ses arêtes pour ensuite les convertir en contraintes entières. Le tableau montre comment les équations reliant ces étapes sont utilisées sur un triangle de l’exemple.

4.4.1 Un maillage décimé comme proxy

Nous décimons le maillage initial en utilisant l'algorithme proposé par [Bommes et al., 2013]. Nous basculons les arêtes qui ne respectent pas le critère de Delaunay dans l'espace de la carte sans couture, puis nous écrasons autant d'arêtes que possible, et nous répétons finalement le processus jusqu'à la convergence. Les conditions pour qu'une arête puisse être écrasée sont que cela ne supprime pas un sommet singulier, que le maillage reste localement homéomorphe au plan, et qu'aucun triangle ne soit inversé dans la carte sans coutures. L'écrasement des arêtes se fait en priorisant les arêtes les plus courtes.

Après chaque opération d'édition, les coordonnées des nouveaux triangles dans la carte sont copiées à partir des coordonnées des triangles qui précédaient l'opération. À noter que les discontinuités de la carte requièrent une attention particulière : avant d'appliquer une opération, certains triangles doivent être déplacés dans la carte afin de pousser les discontinuités en dehors de la région affectée par l'opération, comme illustré Figure 4.6. Une chose importante à remarquer ici est que déplacer les discontinuités et appliquer les opérations de basculement et d'écrasement d'arête produisent de nouvelles coordonnées qui peuvent être exprimées comme combinaisons linéaires des anciennes coordonnées de la carte. Nous enchaînons ces expressions linéaires au cours du processus de décimation de sorte à former une matrice D qui permet de convertir une carte sur le maillage initial en une carte sur le maillage décimé. Nous avons en particulier

$$U^{\text{sub}} = D \cdot U^{\text{SC}} \quad (4.6)$$

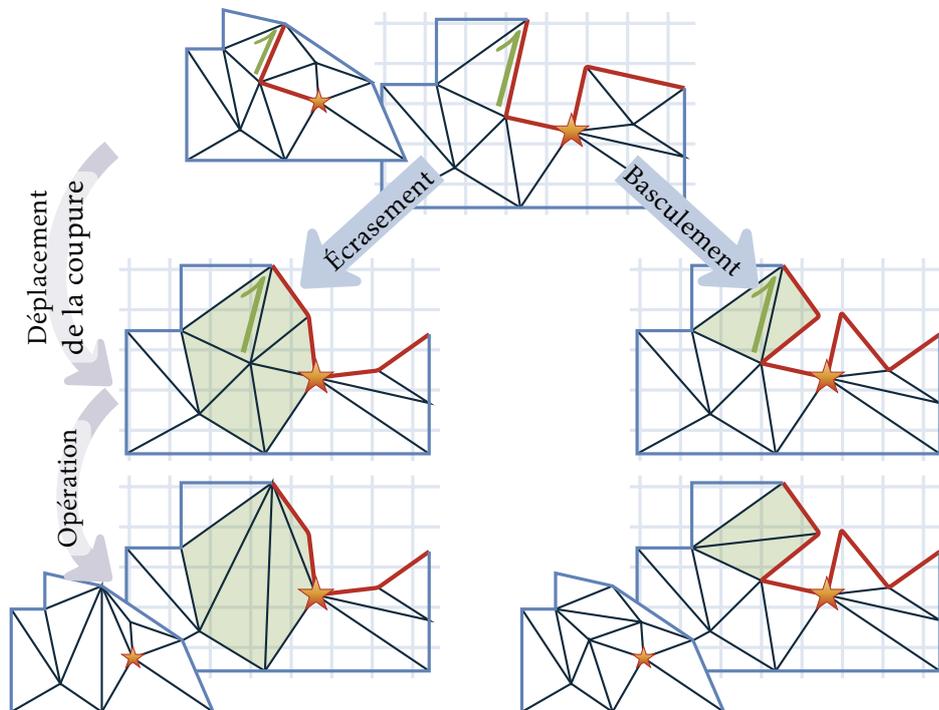


FIGURE 4.6 – Les opérations de décimation (écrasement et basculement d'arête) nécessitent certains changements au sein de la carte. Lorsqu'une arête affectée par l'opération se trouve au niveau d'une discontinuité, cette dernière est déplacée avant d'appliquer l'opération.

4.4.2 Optimisation des degrés de liberté entiers

Après avoir décimé le maillage d'entrée et calculé la carte sans coutures valide correspondante U^{sub} , nous devons calculer une carte préservant la grille U^- sur ce même maillage décimé. Pour cela, nous trouvons dans un premier temps une carte valide préservant la grille. Par valide, nous entendons qu'aucun triangle n'est dégénéré ou inversé. Puis nous optimisons cette carte. Laissons pour le moment de côté la façon dont nous initialisons U^- (nous retournerons sur ce point dans la sous-section Initialisation) et concentrons-nous sur l'optimisation.

Nous cherchons des coordonnées de cartes U^- qui sont des entiers de Gauss de sorte que la géométrie induite des arêtes soit la plus proche possible que celle induite par la carte sans coutures U^{sub} . C'est-à-dire que U^- doit minimiser :

$$E(\omega) = \sum_h \frac{\|\omega_h - \omega_h^{\text{sub}}\|^2}{\|\omega_h^{\text{sub}}\|^2} \quad (4.7)$$

où $\omega_h = U_{\text{succ}(h)}^- - U_h^-$ et $\omega_h^{\text{sub}} = U_{\text{succ}(h)}^{\text{sub}} - U_h^{\text{sub}}$ sont respectivement les géométries de la demi-arête h dans la carte préservant la grille U^- et dans la carte sans coutures U^{sub} sur le maillage décimé.

Il est plus pratique de manipuler ω_h plutôt que U_h^- . Les deux représentations sont équivalentes à une translation près de chaque triangle, puisque ω est la dérivée extérieure de la 0-forme U^- sur chaque triangle [Crane, 2023]. En particulier, ω est exacte, donc fermée, c'est-à-dire que sa dérivée extérieure est nulle et donc pour chaque triangle t :

$$\sum_{h \in t} \omega_h = 0. \quad (4.8)$$

De plus, pour cette nouvelle représentation, l'équation (4.1) devient :

$$\omega_h = -r_h \omega_{\text{opp}(h)}. \quad (4.9)$$

On note également que lorsque pour tout h , $\omega_h \in \mathbb{Z}[t]$, alors il est possible de trouver une carte quantifiée telle que $U_h^- \in \mathbb{Z}[t]$ pour toute demi-arête h . L'intégration de ω étant définie à une constante près, il suffit par exemple de choisir une demi-arête h par triangle et d'imposer $U_h^- = 0$. De cette façon, les translations à travers les discontinuités sont toutes entières et les coordonnées des singularités sont également entières, de sorte que U^- satisfasse (4.2) et préserve donc la grille. Dans le reste de cette section, nous utiliserons abusivement ω pour simplifier les expressions.

Comme pour QGP, notre optimisation (Algorithme 5) repose sur l'application d'opérations atomiques qui préservent la validité de la carte. Une initialisation valide ω est trouvée (lignes 1–8) en multipliant et arrondissant la carte sans coutures en entrée U^{sub} (à nouveau, nous rentrons dans les détails dans la sous-section Initialisation). Puis, un algorithme applique des opérations atomiques faisant réduire l'énergie (4.7) (lignes 9–16) jusqu'à ce que plus aucune opération ne puisse être trouvée. Comme pour QGP, cette stratégie atteint un minimum local de l'énergie, mais pas nécessairement un minimum global.

Nous présentons premièrement le cœur de la procédure d'optimisation en définissant et en expliquant comment calculer les opérations atomiques. Puis nous expliquerons l'initialisation de ω .

Algorithme 5 : Optimisation des degrés de liberté entiers

Entrées : Le maillage décimé \mathcal{M}^{sub} et sa carte sans couture U^{sub}
Sorties : ω la dérivée extérieure par triangle, d'une carte préservant la grille U^- .

/ Initialisation de ω (voir §4.4.2-Initialisation) */*

```

1  $n \leftarrow 0$ 
2 répéter
3   pour chaque  $h \in \{1, \dots, H^{\text{sub}}\}$  faire
4      $\omega_h \leftarrow U_{\text{succ}(h)}^{\text{sub}} - U_h^{\text{sub}}$ 
5      $\omega_h \leftarrow \arg \min_{a \in \mathbb{Z}[i] \setminus \{0\}} |a - 2^n \omega_h|$ 
6     rendre_fermée ( $\omega$ )
7      $n \leftarrow n + 1$ 
8 jusqu'à  $\omega$  est valide
   /* Modification de  $\omega$  pour minimiser  $E(\omega)$  */
9 répéter
10  pour chaque  $(h_0, a_0) \in \{1, \dots, H^{\text{sub}}\} \times \{1, i, i^2, i^3\}$  faire
11     $\alpha \leftarrow \text{trouver\_opération\_atomique}(\omega, h_0, a_0)$ 
12    si  $E(\omega + \alpha) < E(\omega)$  et  $\omega + \alpha$  est valide alors
13       $\omega \leftarrow \omega + \alpha$ 
14 jusqu'à plus aucune opération atomique trouvée

```

Notre opération atomique

Notre opération atomique édite la combinatoire du maillage quadrangulaire structuré qui sera produit par la carte préservant la grille que nous construisons. Pour l'illustrer, la figure 4.7 montre une opération qui insère un cycle de quadrilatères¹.

QGP manipule la combinatoire du maillage quadrangulaire en ajustant les longueurs $l_h \in \mathbb{N}$ des arêtes d'un maillage en T (Fig. 4.7–deuxième ligne). Après avoir trouvé un cycle de faces du maillage en T (illustré en noir) sortant toujours d'une région rectangulaire par une arête opposée à celle par laquelle il est rentré, QGP ajuste d'une unité de longueur toutes les arêtes traversées.

Alors que QGP n'altère pas la direction des arêtes, mais ne fait que les allonger (ou les raccourcir), il est possible d'effectuer exactement la même opération en utilisant un autre jeu de variables : on peut utiliser la géométrie des arêtes au lieu des longueurs (Fig. 4.7–troisième ligne). Dans ce cas, chaque arc h du maillage en T voit sa géométrie $\omega_h \in \mathbb{Z}[i]$ mise à jour par $\omega_h \leftarrow \omega_h + \alpha_h$, où $\alpha_h \in \{1, i, i^2, i^3\}$ est sélectionné pour augmenter (ou diminuer) la longueur d'une unité sans changer l'orientation de l'arc au sein de la carte.

Dans notre cas, nous faisons exactement cela, mais nous autorisons l'orientation des demi-arêtes du maillage décimé à changer au sein de la carte (Fig. 4.7–dernière ligne). Ainsi, notre opération atomique est représentée par un cycle orienté (en noir) du maillage dual, ainsi que par les valeurs α_h associées à chaque demi-arête traversée par ce cycle. Ici, les valeurs de α_h sont contraintes à préserver la fermeture de ω (4.8) et ses transitions sans coutures (4.9).

1. Nous montrons ici, l'insertion d'un cycle de quadrilatères, mais nos opérations atomiques, ainsi que celle de QGP, sont en réalité plus riche que cela. Voir la section 4.5.3 pour une illustration.

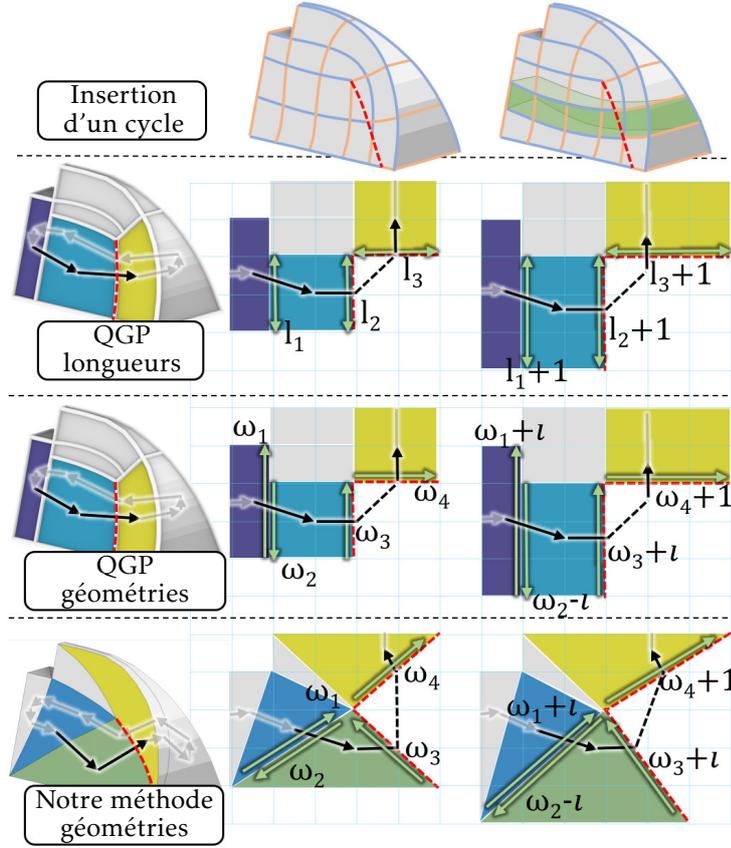


FIGURE 4.7 – L’insertion d’un cycle de quadrilatères (en vert, première ligne), en utilisant une opération atomique de QGP (seconde ligne), notre formalisme appliqué à un maillage en T (troisième ligne) et sur un maillage décimé (quatrième ligne).

Calcul des opérations atomiques

Nous représentons, une opération atomique comme un cycle du graphe dual \mathcal{G} du 4-revêtement (*quad covering*) [Kälberer et al., 2007] du maillage décimé (Fig. 4.8–gauche). Les sommets de \mathcal{G} sont définies par les couples (t, a) composé d’un triangle t de \mathcal{M}^{sub} et d’une valeur $a \in \{1, t, t^2, t^3\}$. Deux sommets (t, a) et (t', a') sont reliés par un arc si et seulement si t est adjacent à t' via une demi-arrête h appartenant à t et vérifiant $a = r_h a'$. On appelle alors cet arc (h, a) . Une opération atomique est donc une séquence d’arcs $\{(h_i, a_i) \mid i = 1, \dots, C\}$ formant un cycle de \mathcal{G} qui peut être appliqué pour mettre à jour ω via

$$\omega_{h_i} \leftarrow \omega_{h_i} + a_i \quad \text{et} \quad \omega_{\text{opp}(h_i)} \leftarrow \omega_{\text{opp}(h_i)} - r_{\text{opp}(h_i)} a_i \quad (4.10)$$

pour tout i dans $\{1, \dots, C\}$. On associera alors un vecteur α à chaque opération atomique de sorte que son application puisse s’écrire $\omega \leftarrow \omega + \alpha$.

Pour sélectionner une opération atomique à appliquer, l’algorithme 5, à la ligne 11, a besoin d’un cycle passant par un arc (h_0, a_0) . Nous calculons ce cycle en chaînant cet arc avec le chemin de coût minimal reliant ses deux extrémités dans \mathcal{G} . Ici le coût du chemin est la somme des poids de ses arcs que nous allons maintenant définir.

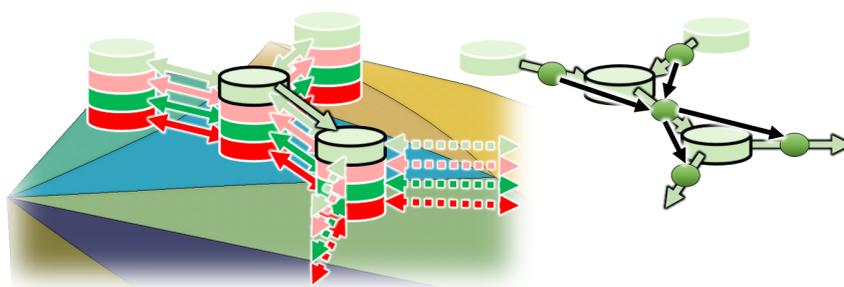


FIGURE 4.8 – Le graphe dual \mathcal{G} du 4-revêtement (gauche). Les sommets (cylindres) sont définis par un triangle t et une valeur $a \in \{1, i, i^2, i^3\}$ (la couleur du cylindre). Pour apporter plus de libertés dans la définition des coûts des chemins de \mathcal{G} , un nouveau graphe \mathcal{G}' est créé (droite) où chaque arc de \mathcal{G} est transformé en un sommet de \mathcal{G}' (petite sphère verte) auquel on ajoute deux arcs sortants et deux arcs entrants (flèches noires).

En fixant les poids des arcs à 1, on risque de trouver des opérations atomiques qui ne diminueront pas l'énergie ou rendront ω invalide. En conséquence, nous devons choisir soigneusement les poids pour améliorer la probabilité que l'opération atomique trouvée conserve la validité de ω et diminue $E(\omega)$. Idéalement, le coût du chemin pourrait être le gain d'énergie et les chemins cassant la validité devraient avoir un coût infini afin de ne pas les considérer. Malheureusement, cette définition du coût rend le problème de trouver le chemin le plus court difficile pour plusieurs raisons :

- La première est que le gain d'énergie peut être négatif. Il ne serait donc pas possible de trouver le plus court chemin avec un algorithme de Dijkstra ayant une complexité quasi linéaire en la taille de \mathcal{G} . Les solutions connues pour rechercher un chemin de coût minimal dans un graphe ayant des poids négatifs ont une complexité quadratique en la taille du graphe \mathcal{G} (exemple : l'algorithme de Bellman-Ford). Pour éviter cette hausse de complexité, nous choisissons de définir des coûts positifs par arête.
- La seconde difficulté est qu'attribuer un coût infini aux chemins cassant la validité ne peut pas être obtenu en sommant des poids par arc. Ce coût dépend du chemin dans sa globalité. Dans le cas le plus simple, où un chemin ne repasse jamais deux fois par un même triangle, la validité a posteriori d'un triangle t traversé par le chemin dépend à la fois de l'arc qui arrive à t et de l'arc qui part de t . Une solution pour remédier à ce problème est de construire un nouveau graphe \mathcal{G}' dont les sommets sont les arcs de \mathcal{G} . \mathcal{G}' correspond au *graphe dual* défini dans [Añez et al., 1996] (à ne pas confondre avec le sens usuel de graphe dual inversant sommets et faces d'un graphe plongé sur une surface). Deux sommets $v = (h, a)$ et $v' = (h', a')$ de \mathcal{G}' sont reliés par un arc si et seulement si l'arrivée de v est l'origine de v' dans \mathcal{G} et $h \neq \text{opp}(h')$ (Fig. 4.8–droite). Un arc $v \rightarrow v'$ de \mathcal{G}' encode toutes les modifications apportées au triangle associé à l'arrivée de v (l'origine de v'), de sorte qu'on puisse lui associer un coût infini si ladite modification rend le triangle invalide. Nous proposons donc de rechercher une opération atomique dans ce nouveau graphe. Cependant, il est tout à fait possible qu'une opération atomique passe plusieurs fois à travers un triangle. Par exemple, en modifiant une coordonnée lors du premier passage et en modifiant l'autre coordonnée pour le second passage. \mathcal{G}' n'est donc pas suffisant pour empêcher de considérer des opérations atomiques invalides et la vérification de validité à la ligne 12 de l'algorithme 5, est bien nécessaire.

Désormais, nous cherchons un cycle incluant le sommet (h_0, a_0) de \mathcal{G}' , c'est-à-dire un chemin partant de (h_0, a_0) et arrivant à (h_0, a_0) . Un arc $(h, a) \leftarrow (h', a')$ de \mathcal{G}' est noté (h, h', a) et le poids qui lui est assigné est noté $w_{h, h', a}$. Ce poids dépend de l'impact de l'ajout de la valeur a à ω_h sur l'énergie, et également de la validité du triangle auquel appartient h' après modification. Pour estimer l'impact sur l'énergie, nous calculons une valeur δ (illustré par la figure 4.9) comme le produit scalaire entre a et $\omega_h^{\text{sub}} - \omega_h$ (en associant \mathbb{C} à \mathbb{R}^2) :

$$\delta = \Re \epsilon (\bar{a} \cdot (\omega_h^{\text{sub}} - \omega_h)) \quad (4.11)$$

Finalement la valeur du poids $w_{h, h', a}$ dépend de trois cas :

- Si $\delta > 1/2$, alors ajouter a à ω_h diminue l'énergie. Pour favoriser cet arc, nous considérons le poids $w_{h, h', a} = (1/2 + \delta)^{-1}$. Cette valeur est faible (< 1), positive et décroît en fonction de δ (pour prendre en considération le carré dans la définition de l'énergie (4.7)).
- Si $\delta \leq 1/2$, alors ajouter a à ω_h augmente l'énergie. Pour éviter cet arc, nous considérons le poids $w_{h, h', a} = (1 - \delta)/\epsilon$. Cette valeur est toujours positive et décroissante en fonction de δ , mais elle est plus grande que la somme des poids des arcs diminuant l'énergie lorsque $\epsilon \rightarrow 0$.
- Finalement, si ajouter a à ω_h et $r_h a$ à $\omega_{h'}$ rend le triangle de h' invalide, $w_{h, h', a}$ est remplacé par $1/\epsilon^2$ qui est plus grand que la somme de tous les autres poids lorsque $\epsilon \rightarrow 0$. En conséquence, cette arête est considérée uniquement lorsqu'il n'existe pas de chemins passant seulement par des arcs des deux premiers types.

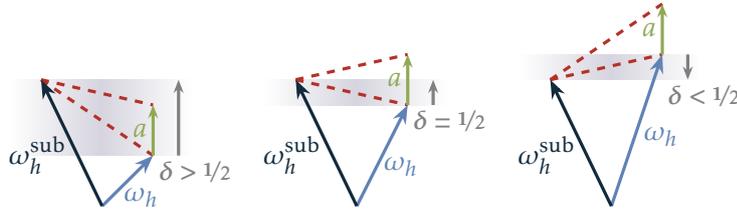


FIGURE 4.9 – δ mesure la projection de $\omega_h^{\text{sub}} - \omega_h$ le long de la droite dirigée par a . La contribution de h à l'énergie est proportionnelle au carré de longueur du segment rouge en traitillé (le segment le plus bas correspond à la contribution avant d'ajouter a , et celui du dessus après l'ajout). La comparaison de δ avec $1/2$ permet de savoir si l'opération augmente ou diminue l'énergie.

Pour résumer, le cycle contiendra uniquement des arcs qui décroissent l'énergie et qui n'invalident aucun triangle si cela est possible. Sinon, il peut contenir des arcs augmentant l'énergie. S'il n'existe toujours pas de telle solution, il peut contenir des arcs qui invalident un triangle dans l'espoir que le cycle repasse par ce même triangle en le rendant à nouveau valide. Pour trouver ce cycle qui est le plus court chemin reliant (h_0, a_0) à lui-même, nous utilisons l'algorithme de Dijkstra.

Initialisation

Initialiser ω par un vecteur nul pourrait presque être une bonne solution, mais malheureusement la carte correspondante n'est pas valide puisque tous les triangles seraient dégénérés. Pour QGP, la validité requiert simplement que toutes les longueurs soient strictement positives. Ce critère peut toujours être atteint en partant de longueurs toutes nulles et en n'appliquant que des opérations augmentant les longueurs. Cette jolie propriété n'étant a priori pas valable dans notre cas, nous avons opté pour une autre approche.

Si l'on prend en compte que la carte sans couture de départ est représentée avec des nombres à virgule flottante, la multiplier par 2^n pour une valeur de n suffisamment grande permet d'obtenir une carte dont toutes les coordonnées sont entières et préservant ainsi la grille. Cela certifie la possibilité de trouver une initialisation valide de ω .

En pratique n n'a pas besoin d'être si grand. Les 8 premières lignes de l'algorithme 5 montrent comment éviter de devoir considérer une trop grande valeur. On tente de multiplier la carte avec des valeurs croissantes de n . À chaque itération, la première étape est d'arrondir ω_h à l'entier de Gauss le plus proche de $2^n \omega_h^{\text{sub}}$ qui est non nul. La valeur de ω qui en résulte peut ne pas être fermée (4.8), c'est pourquoi la ligne 6 de l'algorithme 5 corrige cela. Pour ce faire, nous réutilisons le graphe dual \mathcal{G} du 4-revêtement. ω peut être représenté sur ce graphe en assignant la valeur $\omega_{h,a} = \Re(\bar{a} \cdot \omega_h)$ à chaque arc (h,a) de \mathcal{G} . L'équation (4.8) est alors équivalente à $\forall a \in \{1, i, i^2, i^3\}, \sum_{h \in t} \omega_{h,a} = 0$. Dans un premier temps, il est possible de montrer qu'après l'arrondi, $|\sum_{h \in t} \omega_{h,a}| \leq 1$ pour tout couple (t,a) . Supposons maintenant qu'un triangle t n'est pas valide. Il existe une valeur de a tel que $\sum_{h \in t} \omega_{h,a} \neq 0$. Quitte à considérer $-a$ au lieu de a , on peut supposer $\sum_{h \in t} \omega_{h,a} = -1$. On peut montrer qu'il existe nécessairement un autre sommet (t',a') dans la même composante connexe que (t,a) qui vérifie $\sum_{h' \in t'} \omega_{h',a'} = 1$. En effet, pour chaque arc (h,a) de \mathcal{G} , la valeur associée à l'arc opposé $(\text{opp}(h), r_{\text{opp}(h)}a)$ est opposée (c'est-à-dire $\omega_{h,a} = -\omega_{\text{opp}(h), r_{\text{opp}(h)}a}$). On en déduit que la somme des valeurs des arcs d'une composante connexe de \mathcal{G} est nulle et donc la somme de $\sum_{h \in t} \omega_{h,a}$ pour tous les sommets (t,a) d'une composante connexe est également nulle. D'où l'existence de (t',a') vérifiant $\sum_{h' \in t'} \omega_{h',a'} = 1$. Il se trouve qu'appliquer l'opération atomique associée à un chemin reliant (t,a) à (t',a') modifie ω de sorte à obtenir $\sum_{h \in t} \omega_{h,a} = \sum_{h' \in t'} \omega_{h',a'} = 0$ sans modifier les sommes sur les autres triangles. Il est alors possible de rendre ω fermé en appliquant une telle opération atomique à partir de chaque sommet (t,a) vérifiant $\sum_{h \in t} \omega_{h,a} = -1$ jusqu'à ce ω soit fermé. Une fois ω fermé, la carte associée peut ne pas être valide. Auquel cas nous augmentons n de 1 et réessayons.

4.4.3 Déduction des contraintes pour le problème initial

Par définition, $\omega_h = U_{\text{succ}(h)}^- - U_h^-$, et la carte préservant la grille sur le maillage décimé, U^- , est lié à une carte préservant la grille sur le maillage initial, U^{PG} , par l'équation $U^- = DU^{\text{PG}}$. En introduisant une matrice de permutation S^{sub} encodant l'opérateur succ sur \mathcal{M}^- (c.à.d. $S_{h,h'}^{\text{sub}} = \delta_{h', \text{succ}(h)}$), on obtient l'équation $\omega = (S^{\text{sub}} - I)DU^{\text{PG}}$ qui contraint les chemins reliant les singularités dans la carte U^{PG} à avoir certaines valeurs entières calculées dans ω . Avec cette dernière équation, U^{PG} est bien forcé à préserver la grille, mais les singularités ne sont pas contraintes à se trouver à des coordonnées entières. Il suffit alors de fixer la position d'une singularité à 0 pour que toutes les singularités aient des coordonnées entières. Finalement notre carte U^{PG} peut être calculé à la ligne 3 de l'algorithme 2.

4.4.4 Détails d'implémentation

Dans cette section, nous donnons quelques ingrédients d'implémentation que nous trouvons importants pour reproduire les résultats.

Optimisation numérique Comme dans la plupart des travaux précédents, nous générons une carte alignée à un champ de repères. Nous commençons par intégrer le champ de repères au sens des moindres carrés [Kälberer et al., 2007, Bommès et al., 2009], puis nous procédons à une seconde optimisation non linéaire renforçant la validité de la carte en utilisant [Garanzha et al., 2021]. Cette optimisation est réalisée une première fois pour produire la carte

sans coutures d'entrée, puis une seconde fois pour générer la carte préservant la grille en ajoutant les contraintes affines de quantification. Ces deux optimisations sont effectuées avec une matrice de réduction variable M , et un vecteur colonne B , tel que $U = MX + B$ satisfasse toutes les contraintes nécessaires pour tout vecteur X , et tel que $\ker(M) = \{0\}$. Cela permet de minimiser $X \mapsto f(MX + B)$ sans contraintes et de retrouver notre carte via $U = MX + B$.

Bords et courbes caractéristiques Les bords et courbes caractéristiques d'une surface doivent être représentés par un ensemble d'arêtes dans le maillage quadrangulaire final. Pour cela, la géométrie de chaque demi-arête h de bord ou caractéristique, doit être réelle ou imaginaire pur dans la carte U^{PG} . Cette contrainte est renforcée durant le calcul de la carte sans couture U^{SC} , et propagée durant l'étape de quantification en supprimant certains arcs (h, a) du graphe \mathcal{G} pour empêcher de modifier la partie imaginaire de la géométrie d'une demi-arête devant être réelle (ou inversement).

Pour un maillage possédant un bord, les opérations atomiques calculées pour minimiser $E(\omega)$, ne sont plus nécessairement des cycles. Il est également possible de considérer des chemins reliant deux arêtes de bord.

Degrés de liberté non entiers Remarquons que si nous calculons U^{PG} comme à la ligne 3 de l'algorithme 2, tous les sommets qui sont à la fois dans le maillage d'entrée et dans le maillage décimé sont contraints à avoir des coordonnées entières. Cependant, des sommets non singuliers peuvent persister dans le maillage décimé. Par exemple, un tore peut être paramétré sans singularités, mais sa décimation contient nécessairement des sommets pour préserver la topologie du maillage et pour pouvoir fixer les longueurs de ces cycles irréductibles à des valeurs entières. La contrainte $AU = \omega$ est donc trop forte. Pour retrouver les degrés de liberté perdus, nous définissons un vecteur creux Y dont la valeur Y_h est non nulle si et seulement si, l'origine de h est un sommet préservé dans la décimation et dont les coordonnées ne sont pas nécessairement entières. L'optimisation suivante permet alors de retrouver les degrés de liberté perdus :

$$U^{\text{PG}} \leftarrow \arg \min_U \min_Y f(U) \quad \text{tel que} \quad \begin{cases} A(U + Y) = \omega \\ U \text{ et } U + Y \text{ vérifient (4.1) et (4.4)} \end{cases} \quad (4.12)$$

Poids des arcs de \mathcal{G}' En pratique, nous ne représentons pas les poids des arêtes par des réels, mais par des 3-uplets. Ainsi la valeur $x/\epsilon^2 + y/\epsilon + z$ est représenté par le 3-uplet (x, y, z) et l'on munit l'ensemble des 3-uplets de l'ordre lexicographique.

4.5 Résultats

Les performances de notre algorithme ont été évaluées sur la même base de données de modèles CAO que [Reberol et al., 2021]. Elle contient 1000 modèles du jeu de données ABC [Koch et al., 2019] et les 114 modèles de MAMBO [Ledoux, 2019]. Ces modèles CAO possèdent beaucoup de courbes caractéristiques qui rendent difficile la production de cartes préservant la grille. Nous avons été capables de générer une carte sans couture valide avec les contraintes d'alignement aux bords et courbes caractéristiques pour 1111 modèles et sans les contraintes pour les 3 modèles restants. Sur chacune des entrées, nous avons effectué deux tests de quantification : un pour produire des cartes grossières préservant la grille (Fig. 4.10a), l'autre pour

produire des maillages quadrangulaires avec la résolution de l'entrée (Fig. 4.10b). Les résultats grossiers sont obtenus en multipliant U^{sub} par une très faible valeur avant d'appliquer la quantification.

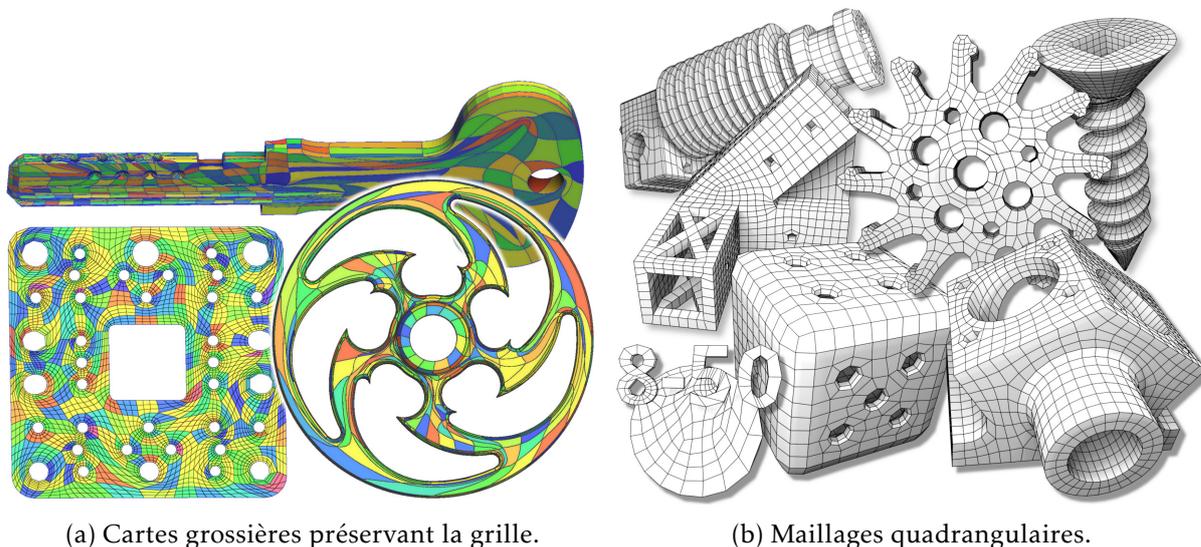


FIGURE 4.10 – Exemples de résultats obtenus sur la base de données. Des cartes grossières préservant la grille à gauche. Ces cartes segmentent la surface en régions rectangulaires, qui peuvent être ensuite subdivisées (modèle en bas à gauche). À droite, des maillages quadrangulaires produits avec notre algorithme.

4.5.1 Temps d'exécution

Nos temps d'exécution ont le même ordre de grandeur que ceux de QGP, et sont presque négligeables comparés au temps de calcul du processus entier de génération de maillages quadrangulaires. Plus précisément, nous avons comparé les temps d'exécution des deux méthodes sur notre base de données. Pour chaque méthode (Algorithme 3 et Algorithme 4), nous séparons la génération du problème de substitution (lignes 1 et 2) de l'étape de quantification (ligne 3). Malheureusement, les expérimentations pour les deux méthodes ont dû être effectuées sur deux ordinateurs différents. Pour notre méthode, nous avons utilisé un processeur Intel Core i5-11500H, et pour QGP nous avons utilisé un processeur AMD 7742.

Bien que l'algorithme de basculement d'arêtes pour rendre une triangulation Delaunay ait une complexité quadratique dans le pire cas [Hurtado et al., 1996], on observe en pratique que le nombre de basculements nécessaires est linéaire en la taille du maillage [Sharp et al., 2021]. La complexité de notre processus de décimation est ainsi empiriquement linéaire. La figure 4.11 compare les temps d'exécutions entre QGP et notre méthode. Le graphique de droite montre que notre décimation est plus rapide que la génération du graphe de motocyclettes. Inversement, la quantification de QGP semble avoir une meilleure complexité en la taille du problème de substitution. Ainsi lorsque notre maillage décimé possède plus de 100 triangles, QGP obtient de meilleur temps d'exécution. En effectuant une régression linéaire à l'échelle log-log on peut estimer la complexité de notre quantification à $\mathcal{O}((H^{\text{sub}})^{2.5})$ et la complexité de QGP à $\mathcal{O}((H^{\text{sub}})^{1.5})$. Où H^{sub} est le nombre de demi-arêtes du maillage décimé.

Dans le cas le plus difficile (Fig. 4.12), notre initialisation de ω requiert de multiplier la

carte sans coutures U^- par 64 avant d'arrondir pour obtenir une initialisation valide. En effet, pour une valeur strictement inférieure à 64, l'arrondissement rend le triangle rouge dégénéré, en plaçant la singularité (étoile) sur le bord de l'arête caractéristique (blanc). Dans ce scénario, notre algorithme a été 50 fois plus lent que QGP, rendant ainsi le processus entier de génération de maillages quadrangulaires 3 fois plus lent. Heureusement, il convient de noter que ce cas particulier est une valeur aberrante dans notre ensemble de données (Fig. 4.11).

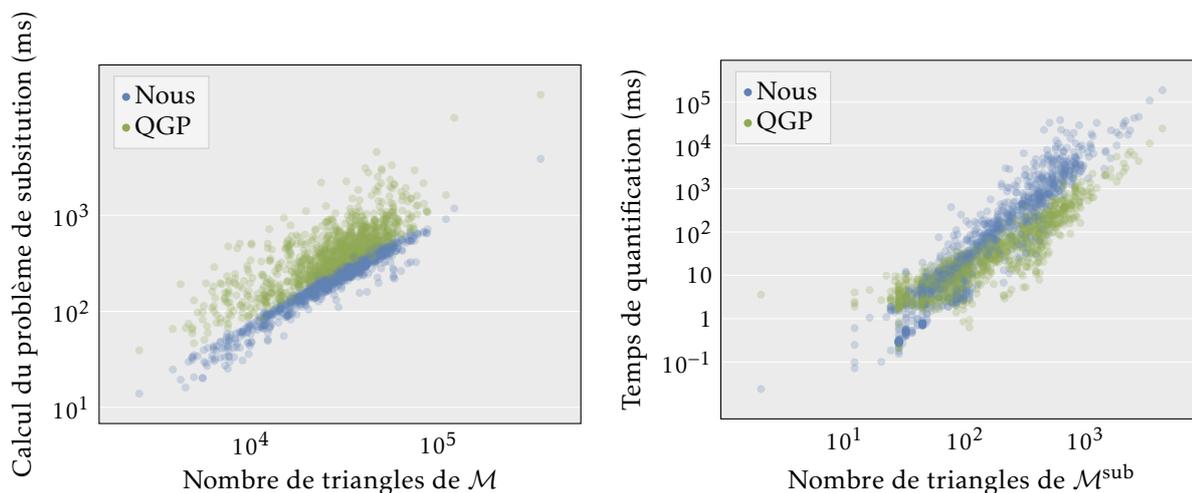


FIGURE 4.11 – Temps de construction du problème de substitution (gauche) et temps de quantification (droite). Chaque modèle de la base de données est à l'origine d'un point bleu pour notre méthode et d'un point vert pour QGP.

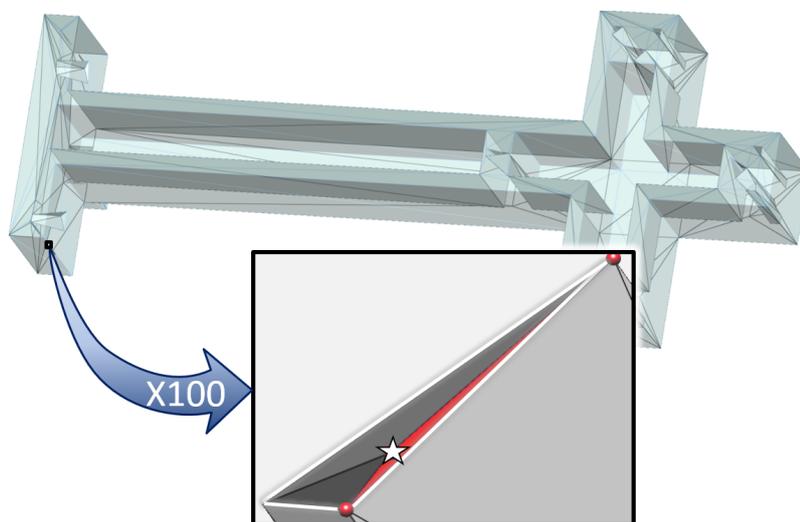


FIGURE 4.12 – Initialisation difficile de la quantification. Le triangle rouge de ce maillage décimé (visible dans le gros plan) nécessite une multiplication de la carte sans couture par un facteur de 64 pour éviter la dégénérescence de la carte due à la singularité (étoile) forcée à se trouver sur l'arête blanche.

4.5.2 Qualité

Il existe plusieurs critères pour évaluer la qualité d'un maillage quadrangulaire [Knupp, 2003], mais l'impact précis de la quantification est plutôt difficile à évaluer indépendamment des autres étapes (paramétrisation, extraction des quadrilatères, lissage, etc.).

En inspectant visuellement quelques modèles (Fig. 4.13 ou encore les documents complémentaires de notre publication), la qualité des maillages quadrangulaires obtenus avec QGP et notre méthode est très similaire.

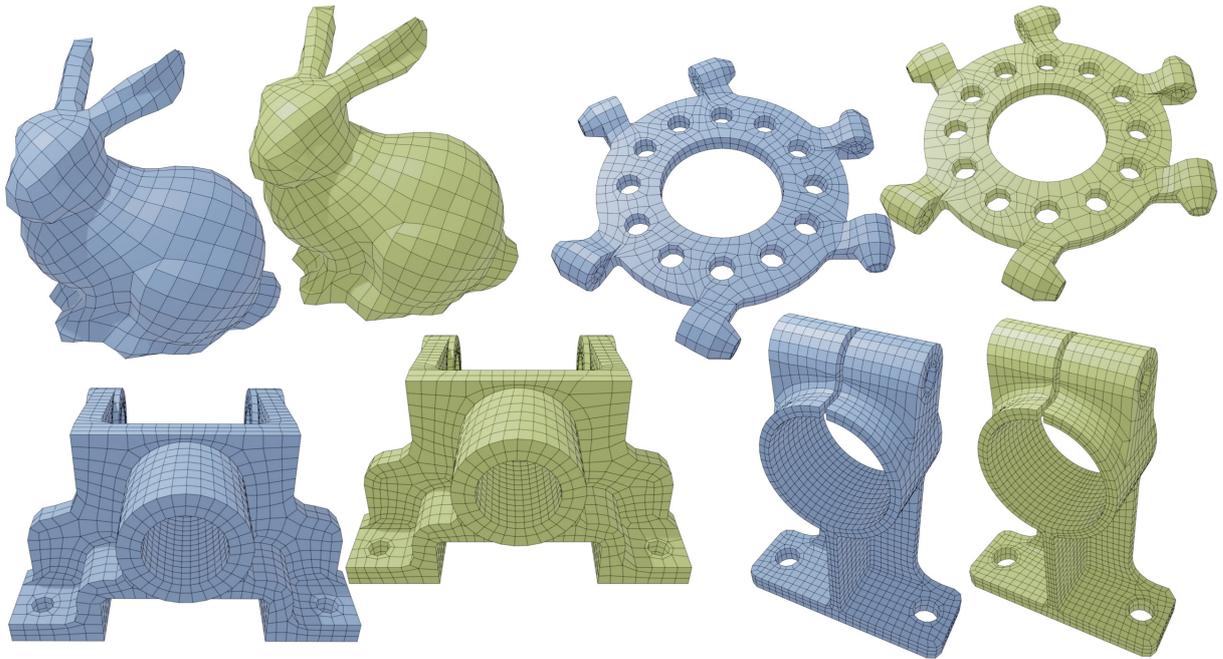


FIGURE 4.13 – Les maillages quadrangulaires produits avec notre méthode (bleu) et avec [Campen et al., 2015] (vert) présentent une qualité similaire.

Nous nous sommes également comparé à [Pietroni et al., 2021] (Fig. 4.14). Cette méthode possède une approche assez différente de celle présentée par l'algorithme 2, mais contient tout de même une étape de quantification similaire à QGP. Cette étape fixe des longueurs d'arêtes dans une décomposition de la surface en régions possédant de 3 à 6 côtés. Après quantification, des singularités peuvent être ajoutées à l'intérieur des régions. On observe à nouveau une qualité similaire entre les maillages produits par [Pietroni et al., 2021] et ceux produits par notre méthode. En revanche, une différence importante est que notre méthode (ou encore QGP) permet d'obtenir une topologie de maillage plus simple. En effet, notre méthode reproduit exactement la topologie (c'est-à-dire le nombre de singularités et leurs placements) du champ de repère calculé initialement, tandis que [Pietroni et al., 2021] ne respecte pas cette topologie.

Finalement, la figure 4.15, illustre nos résultats sur des modèles possédant des structures fines qui sont connus pour ajouter de la difficulté à l'étape de quantification.

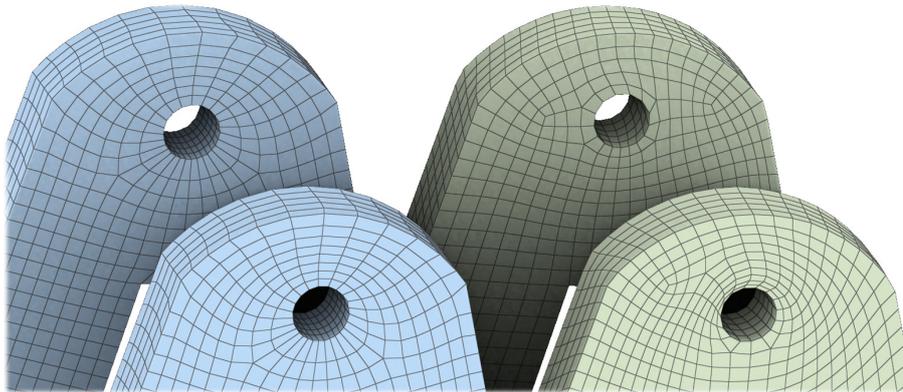


FIGURE 4.14 – Un maillage quadrangulaire extrait depuis une carte préservant la grille (produit avec QGP ou notre méthode, ici en bleu) possède exactement les mêmes singularités que le champ de repères en entrée. En revanche, la méthode robuste introduite par [Pietroni et al., 2021] a tendance à produire des maillages avec une structure plus complexe (vert).

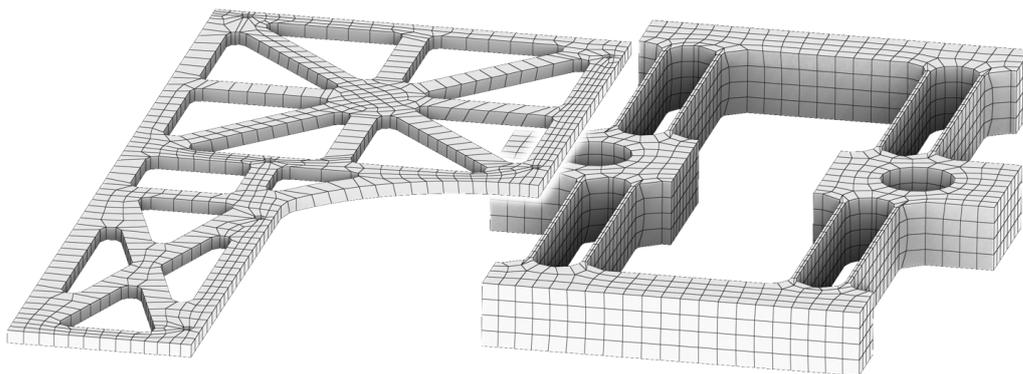


FIGURE 4.15 – Résultats sur des modèles possédant des structures fines.

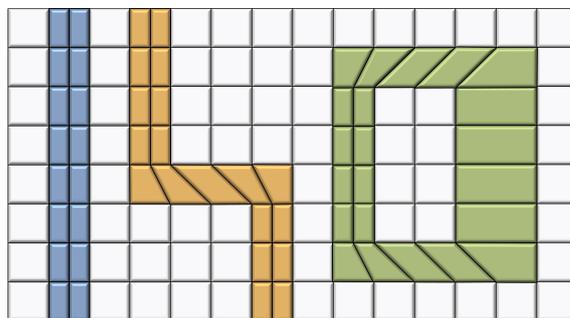


FIGURE 4.16 – Éditions de maillage quadrangulaire appliquées à une grille : insertion d'une corde (bleu), l'opération atomique de QGP peut décaler perpendiculairement l'insertion (orange), notre opération atomique peut également alterner entre insertion et suppression (vert).

4.5.3 Différences d'expressivité avec QGP

Opérations atomiques L'intuition derrière QGP et notre algorithme de quantification est de manipuler la structure combinatoire du maillage quadrangulaire final en appliquant une série d'opérations atomiques. La figure 4.7 illustre l'insertion d'un cycle de quadrilatères, mais notre opération atomique permet en réalité une plus grande liberté d'édition.

L'opération atomique de QGP est une simple insertion/suppression de cycle (Fig. 4.16–bleu) lorsqu'elle est appliquée à un graphe de motocyclettes sans jonctions en T. Lorsque des jonctions en T sont présentes, l'insertion peut glisser/se décaler perpendiculairement, produisant ainsi une modification du maillage quadrangulaire comme celle représentée en orange, Figure 4.16. Cela correspond à l'opération préservant la grille introduite par [Bommes et al., 2011] qui permettait de modifier directement un maillage quadrangulaire. Enfin, notre opération atomique, plus riche, permet également d'alterner entre insertion et suppression au sein d'une même opération (Fig. 4.16–vert).

Expressivité des cartes Comme illustré par l'exemple jouet Figure 4.17, notre maillage décimé permet de représenter des cartes valides pour lesquels un maillage en T requiert des arcs de longueur nulle (troisième colonne) et même des inversions (dernière colonne). Inversement, les maillages en T peuvent représenter de manière valide des cartes qui seraient invalides sur notre maillage décimé (voir figure 4.18). À noter, cependant, qu'un basculement d'arête durant la phase d'optimisation de ω/U^- pourrait résoudre ce problème. Bien que nous n'ayons pas expérimenté la modification du maillage décimé durant la quantification, il est tout à fait envisageable d'explorer cette possibilité dans de futurs travaux. En effet, le basculement d'une arête ne modifie que très localement ω et le graphe \mathcal{G} .

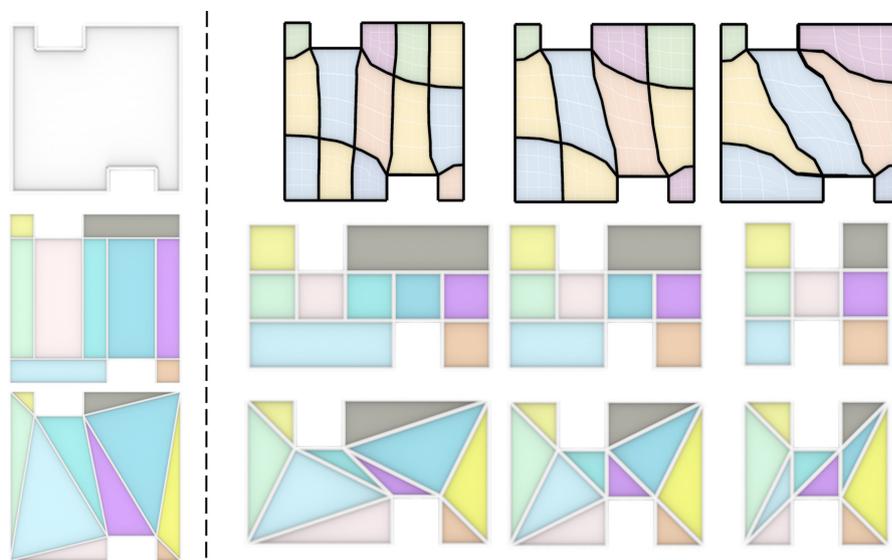


FIGURE 4.17 – À gauche : un modèle (en haut), accompagné de son graphe de motocyclettes (milieu) et de sa décimation (bas). À droite : 3 décompositions en blocs possibles (en haut), leurs cartes correspondantes sur le maillage en T (milieu) et sur le maillage décimé (bas). Une région du maillage en T doit être dégénérée dans la colonne du milieu, et des régions doivent être inversées dans la colonne de droite alors que les cartes sont toujours valides sur le maillage décimé.

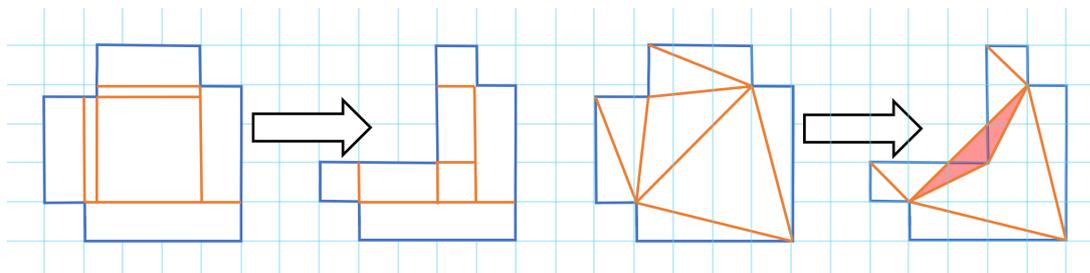


FIGURE 4.18 – Une carte sans couture déformée en une carte préservant la grille. La représentation de la carte sur un maillage en T est tout à fait valide (gauche) alors que le triangle rouge du maillage décimé (droite) est inversé et donc invalide.

4.6 Possibles extensions

Rappelons que nous avons modifié la méthode QGP pour manipuler une carte sur un maillage décimé en optimisant la géométrie de ses arêtes plutôt que de travailler sur un maillage en T en ajustant les longueurs de ses arêtes. Se fondant ainsi sur QGP, notre méthode est susceptible de supporter des extensions similaires, telles que la restriction de la déviation d'angle à un certain intervalle [Lyon et al., 2021a], ou l'ajout de singularités dans les faces du maillage décimé lorsqu'aucune quantification valide ne peut être trouvée. En plus de cela, notre représentation peut offrir de nouveaux avantages qui peuvent directement être exploités ou ouvrir de nouvelles opportunités de recherche.

4.6.1 Bords libres

Contrairement aux maillages en T, les arêtes du maillage décimé ne sont pas nécessairement alignées à un axe de la carte. Comme conséquence, si l'on ne force pas explicitement l'alignement des arêtes de bords, la carte préservant la grille que l'on construit est exempte de toutes conditions (Fig. 4.19). En utilisant des maillages en T [Lyon et al., 2019], un effort conséquent est nécessaire pour obtenir ce comportement : le maillage en T doit être raffiné et des arcs virtuels doivent être introduits.

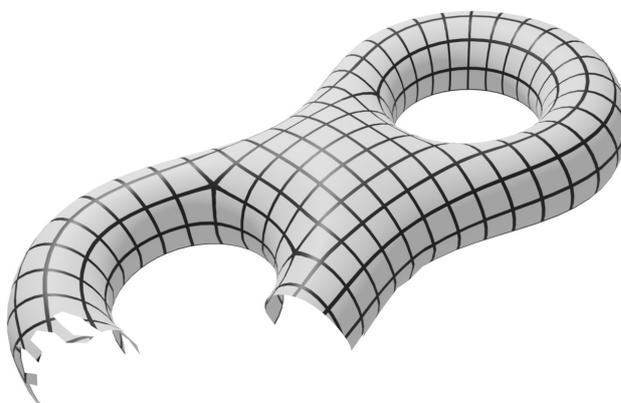


FIGURE 4.19 – Carte préservant la grille à bords libres obtenue avec notre algorithme de quantification.

4.6.2 Renforcement des contraintes de courbes caractéristiques durant la quantification

Le fait que les arêtes ne soient pas alignées avec les axes de la carte présente un autre avantage : nous pouvons prendre en charge les cartes sans coutures mal alignées avec les bords et courbes caractéristiques. Cette fonctionnalité est particulièrement importante pour les modèles CAO qui peuvent présenter tellement de contraintes liées à des courbes caractéristiques, qu'il est parfois difficile (voir impossible) de construire une carte sans coutures valide. Pour respecter le plus grand nombre possible de contraintes liées aux courbes caractéristiques, nous essayons de les appliquer lors de l'étape de quantification. Nous faisons cela dans un premier temps en changeant la partie réelle ou imaginaire de ω_h^{sub} à 0 pour les arêtes concernées (après initialisation de ω). Dans un second temps, lorsqu'une nouvelle contrainte est satisfaite durant l'optimisation de ω , les arcs de \mathcal{G} qui pourraient violer la contrainte sont supprimés. Un avantage direct de la gestion de l'alignement des courbes caractéristiques à l'étape de quantification est de faciliter la recherche d'une carte sans couture avec une distorsion acceptable (Fig. 4.20). S'il n'est pas possible de satisfaire toutes les contraintes, l'alignement des dernières courbes caractéristiques pourrait être obtenu par un post-traitement du maillage (Fig. 4.21).

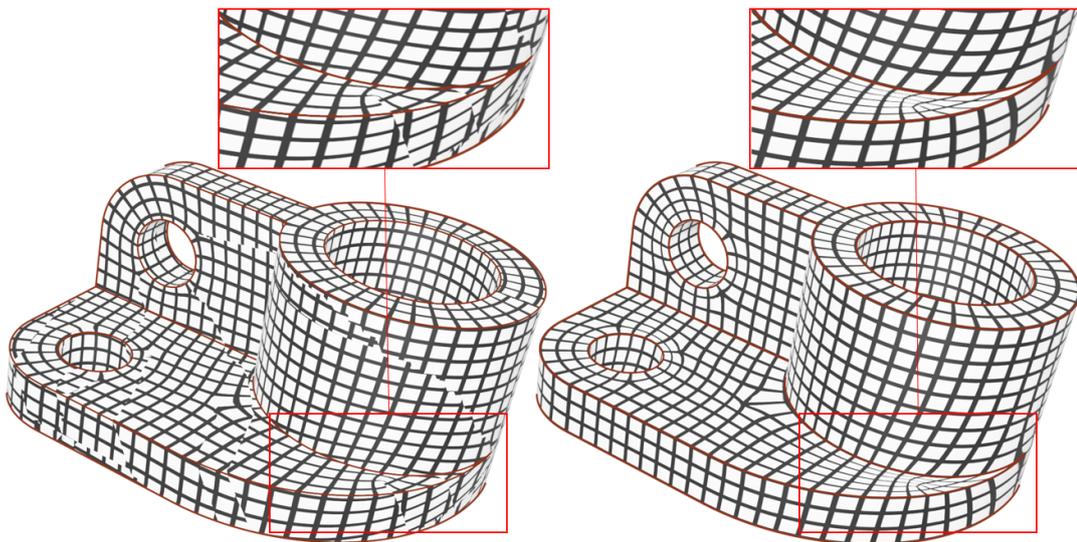


FIGURE 4.20 – Notre méthode peut avoir comme entrée une carte sans couture qui n'est pas parfaitement alignée avec les courbes caractéristiques (gauche). Ces contraintes sont ensuite renforcées durant l'étape de quantification (droite).

4.6.3 Faces dégénérées

Un autre avantage du non-alignement des arêtes avec les axes de la carte est que le maillage décimé permet de représenter un large éventail de structures combinatoires de maillages quadrangulaires sans faces dégénérées. À contrario, l'alignement des singularités, généralement désiré pour le remaillage quadrangulaire, nécessite avec la méthode QGP, de dégénérer certaines régions du maillage en T avec des longueurs d'arc nulles (Fig. 4.22). Ces régions dégénérées sont contre-intuitives si l'on considère que chaque région du maillage en T doit être remplacée par une grille régulière de quadrilatères; les régions dégénérées produiraient un espace vide. En réalité, la méthode QGP autorise des longueurs d'arêtes nulles. Dans ce cas,

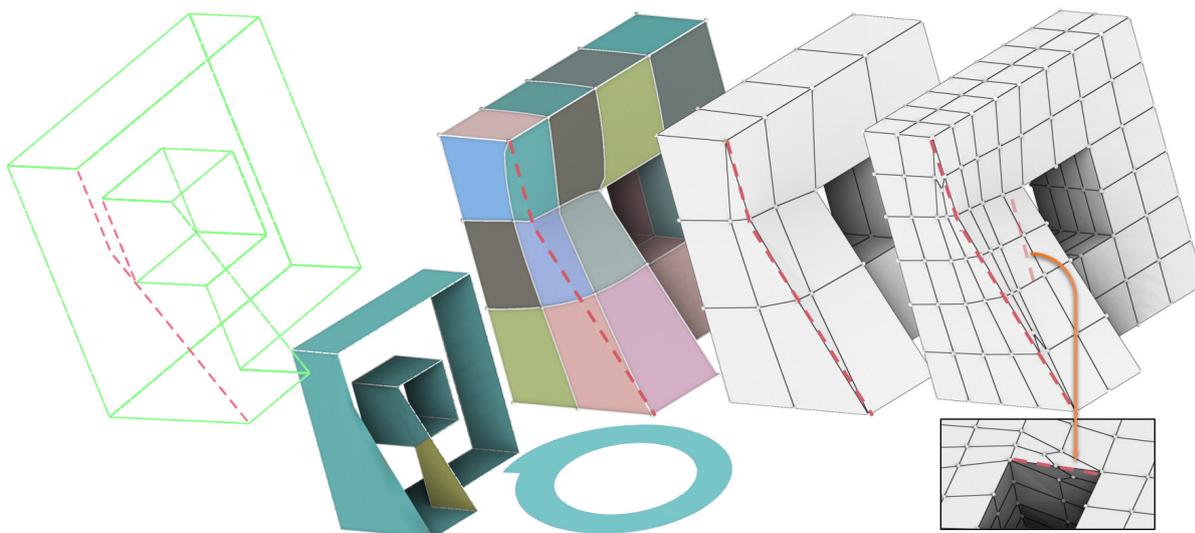


FIGURE 4.21 – Le champ de repères que nous calculons sur ce modèle est incompatible avec les courbes caractéristiques : deux sous-ensembles de faces bleus (en bas à gauche) sont topologiquement équivalents à la célèbre bande spirale de [Myles and Zorin, 2013]. Des contraintes d’alignements ne peuvent être satisfaites durant la quantification (traitillés rouges). Un maillage aligné à toutes les courbes caractéristiques peut alors être obtenu en imprimant les courbes caractéristiques au sein de la première carte préservant la grille puis en subdivisant toutes les arêtes en deux (droite).

un jeu de longueurs d’arcs $\{l_h\}$ est considéré valide s’il n’existe aucun chemin entre deux singularités de longueurs nulles. Comme nous allons le voir dans le prochain paragraphe, cela a des conséquences sur la suite du processus de génération de maillages quadrangulaires. À l’inverse, notre méthode n’a pas besoin de triangles dégénérés pour aligner les singularités.

Transfert de la carte préservant la grille au maillage initial La production d’une carte préservant la grille valide via une optimisation numérique accompagnée de contraintes affines (Algorithme 2–ligne 3) peut être extrêmement difficile. Parfois, cela est même impossible sans raffiner le maillage. S’il est possible de transférer la carte U^- vers le maillage triangulaire initial, cela semble une bien meilleure alternative. En considérant que le maillage décimé \mathcal{M}^{sub} est incorporé dans le maillage initial \mathcal{M} (via la carte sans couture), il n’est pas nécessaire de résoudre un problème d’optimisation complexe pour obtenir la carte préservant la grille finale. En effet, en imprimant les arêtes de \mathcal{M}^{sub} sur le maillage initial, il est possible de transférer U^- sur \mathcal{M} . Une stratégie similaire avait déjà été utilisée avec des maillages en T [Myles et al., 2014, Lyon et al., 2019], mais nécessitait un processus d’édition du maillage en T pour y faire disparaître les régions prescrites à une aire nulle par la quantification. À l’inverse, l’absence de triangles dégénérés dans notre quantification évite d’avoir recours à un processus fastidieux d’édition et de réincorporation du maillage décimé.

Faces dégénérées dans notre quantification Enfin, il est possible d’augmenter l’espace de recherche de notre quantification en autorisant des triangles dégénérés. Dans ce cas, chaque région connexe de triangles dégénérés est paramétrée par un segment de dimension 1. Chaque sommet incident à cette région est projeté sur ce segment. La carte U^- construite est alors consi-

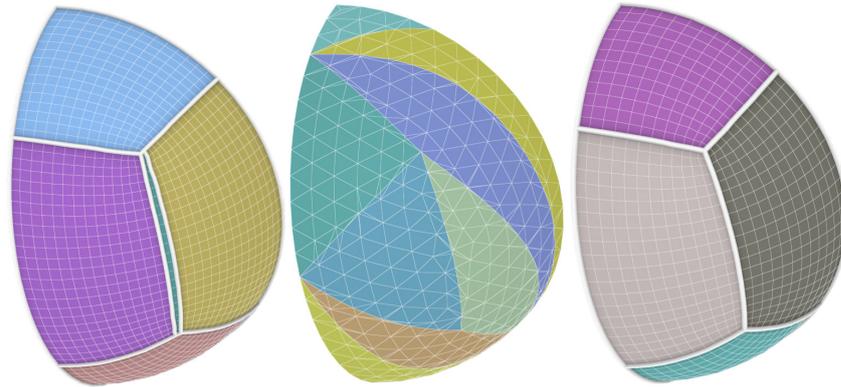


FIGURE 4.22 – Sans dégréner de régions, la quantification du maillage en T (à gauche) ne peut pas supprimer la région fine pour aligner les singularités (à droite). Notre maillage décimé (au milieu) n’a pas besoin de dégréner les faces pour le faire.

dérée valide si et seulement si, aucune paire de singularités n’est projetée sur un même point du segment. Si l’on ajoute des courbes caractéristiques, il faut également ajouter la condition qu’aucune singularité ne doit être projetée sur la projection d’une arête caractéristique. Bien sûr, cet agrandissement de l’espace de recherche permet de réduire l’énergie $E(\omega)$ (4.7) au prix de complexifier le transfert de la carte U^- au maillage initial comme expliqué dans le paragraphe précédent.

4.7 Conclusion

Ce travail a montré que les algorithmes de quantifications peuvent aussi bien s’appliquer à un maillage décimé qu’à un maillage en T. Un premier avantage est que cette nouvelle structure est plus courante, plus simple à générer et à manipuler. L’autre avantage est que ses arêtes ne sont pas nécessairement alignées avec les axes de la carte, ce qui rend la structure plus flexible. Bien que nous n’améliorions pas la robustesse ni la qualité de l’état de l’art, nous pensons que l’une des principales contributions de ce travail est son aspect esthétique, tout comme une preuve plus concise d’un théorème en mathématiques. En outre, une formulation plus simple ouvre la voie à de nouvelles opportunités de futurs travaux.

Conclusion

Dans cette thèse, trois contributions ont été présentées. La première décrit une méthode permettant de remailler hexaédriquement un domaine géologique en extrudant un maillage quadrangulaire, tandis que les deux autres contributions proposent de nouvelles méthodes pour résoudre des étapes spécifiques du remaillage quadrangulaire par paramétrisation globale.

La première contribution, en plus d'élargir la gamme des réseaux de failles pouvant être remaillés de manière structurée, montre des résultats prometteurs pour les simulations couplant écoulements et géomécanique. À l'avenir, la méthode de remaillage hexaédrique présentée spécifiquement pour des domaines géologiques pourrait en réalité être étendue à une plus grande variété de modèles, en particulier aux modèles CAO. Ce fut le travail effectué durant un stage en amont de cette thèse de doctorat [Coudert-Osmont, 2020]. En revanche, les travaux effectués durant ce précédent stage n'étaient pas suffisamment matures et mériteraient d'être approfondis.

Comme précisé dans le chapitre 3, la méthode de génération de champs de repères non orthogonaux présentée requiert une optimisation complexe et peu efficace. Pour cette raison nous préférons au sein, de l'équipe Pixel, l'algorithme spécialement conçu pour la 2D décrit dans [Desobry et al., 2021b]. Certes, cette dernière méthode ne s'étend pas naturellement en 3D, mais cela n'est absolument pas nécessaire grâce à l'approche 2.5D présentée dans cette thèse.

Le nouvel algorithme de quantification, quant à lui, est une contribution qui devrait pouvoir bénéficier à la communauté en raison de sa simplicité. À l'instar de la méthode QGP, il pourrait être extrêmement intéressant d'étendre cette approche à la 3D. C'est sans doute la continuité qui semble la plus naturelle des travaux présentés dans ce manuscrit. La décimation d'un maillage tétraédrique ne semble pas poser particulièrement plus de problèmes que la décimation d'un maillage triangulaire, et semble plus simple que la construction d'un complexe de motocyclettes 3D [Brückler et al., 2022b]. En revanche, de nombreux défis risquent d'apparaître durant l'optimisation des degrés de liberté entiers puisque les opérations atomiques ne seront plus des chemins unidimensionnels, mais des « nappes » à deux dimensions. Il faudra peut-être malheureusement se résoudre à utiliser des solveurs en nombres entiers génériques tout comme l'extension de QGP [Brückler et al., 2022a].

Bibliographie

- [Altair, 2024] Altair (2024). HyperMesh. <https://altairengineering.fr/hypermesh/>.
- [Ansys, 2024] Ansys (2024). Ansys Meshing. <https://www.ansys.com/products/meshing>.
- [Añez et al., 1996] Añez, J., De La Barra, T., and Pérez, B. (1996). Dual graph representation of transport networks. *Transportation Research Part B : Methodological*, 30(3) :209–216.
- [Bathie et al., 2022] Bathie, G., Berthe, G., Coudert-Osmont, Y., Desobry, D., Reinald, A., and Rocton, M. (2022). Pace solver description : Dreyfvs. *Leibniz International Proceedings in Informatics*.
- [Bedel et al., 2022] Bedel, A., Coudert-Osmont, Y., Martínez, J., Nishat, R. I., Whitesides, S., and Lefebvre, S. (2022). Closed space-filling curves with controlled orientation for 3d printing. In *Computer Graphics Forum*, volume 41, pages 473–492. Wiley Online Library.
- [Berthe et al., 2023] Berthe, G., Coudert-Osmont, Y., Dobler, A., Morelle, L., Reinald, A., and Rocton, M. (2023). Pace solver description : Touiuidth. In *IPEC 2023-18th International Symposium on Parameterized and Exact Computation*, volume 285, page 4.
- [Bommes et al., 2013] Bommes, D., Campen, M., Ebke, H., Alliez, P., and Kobbelt, L. (2013). Integer-grid maps for reliable quad meshing. *ACM Trans. Graph.*, 32(4) :98 :1–98 :12.
- [Bommes et al., 2011] Bommes, D., Lempfer, T., and Kobbelt, L. (2011). Global structure optimization of quadrilateral meshes. In *Computer Graphics Forum*, volume 30, pages 375–384. Wiley Online Library.
- [Bommes et al., 2009] Bommes, D., Zimmer, H., and Kobbelt, L. (2009). Mixed-integer quadrangulation. *ACM Trans. Graph.*, 28(3).
- [Bommes et al., 2012] Bommes, D., Zimmer, H., and Kobbelt, L. (2012). Practical mixed-integer optimization for geometry processing. In *Proceedings of the 7th international conference on Curves and Surfaces*, pages 193–206, Berlin, Heidelberg. Springer-Verlag.
- [Bonnet et al., 2021] Bonnet, E., Kim, E. J., Thomassé, S., and Watrigant, R. (2021). Twin-width i : Tractable fo model checking. *J. ACM*, 69(1).
- [Botella et al., 2016] Botella, A., Lévy, B., and Caumon, G. (2016). Indirect unstructured hex-dominant mesh generation using tetrahedra recombination. volume 20, pages 437–451.
- [Brückler et al., 2022a] Brückler, H., Bommes, D., and Campen, M. (2022a). Volume parametrization quantization for hexahedral meshing. *ACM Trans. Graph.*, 41(4).
- [Brückler et al., 2022b] Brückler, H., Gupta, O., Mandad, M., and Campen, M. (2022b). The 3d motorcycle complex for structured volume decomposition. In *Computer Graphics Forum*, volume 41, pages 221–235. Wiley Online Library.
- [Campen et al., 2015] Campen, M., Bommes, D., and Kobbelt, L. (2015). Quantized global parametrization. *ACM Trans. Graph.*, 34(6) :192 :1–192 :12.

- [Coreform, 2024] Coreform (2024). Cubit. <https://coreform.com/products/coreform-cubit/>.
- [Coudert et al., 2022] Coudert, Y., Eisemann, E., Marroquim, R., et al. (2022). Semi-automatic perspective lines from paintings. In *Eurographics Workshop on Graphics and Cultural Heritage*.
- [Coudert-Osmont, 2020] Coudert-Osmont, Y. (2020). FrameField 2.5D. Master’s thesis, Ecole Normale Supérieure Paris-Saclay. <http://perso.ens-lyon.fr/yoann.coudert-osmont/Documents/internship2020.pdf>.
- [Coudert-Osmont et al., 2023] Coudert-Osmont, Y., Desobry, D., Heistermann, M., Bommès, D., Ray, N., and Sokolov, D. (2023). Quad mesh quantization without a t-mesh. *Computer Graphics Forum*, 43.
- [Crane, 2019] Crane, K. (2019). The n-dimensional cotangent formula. <https://www.cs.cmu.edu/~kmc Crane/Projects/Other/nDCotanFormula.pdf>.
- [Crane, 2023] Crane, K. (2023). Discrete differential geometry : An applied introduction. <http://www.cs.cmu.edu/~kmc Crane/Projects/DDG/paper.pdf>.
- [Desobry et al., 2021a] Desobry, D., Coudert-Osmont, Y., Corman, E., Ray, N., and Sokolov, D. (2021a). Designing 2d and 3d non-orthogonal frame fields. *Computer-Aided Design*, 139 :103081.
- [Desobry et al., 2021b] Desobry, D., Protais, F., Ray, N., Corman, E., and Sokolov, D. (2021b). Frame fields for cad models. In *Advances in Visual Computing*, pages 421–434, Cham. Springer International Publishing.
- [Diamanti et al., 2014] Diamanti, O., Vaxman, A., Panozzo, D., and Sorkine-Hornung, O. (2014). Designing N -polyvector fields with complex polynomials. *Comput. Graph. Forum*, 33(5) :1–11.
- [Diamanti et al., 2015] Diamanti, O., Vaxman, A., Panozzo, D., and Sorkine-Hornung, O. (2015). Integrable PolyVector fields. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)*, 34(4) :38 :1–38 :12.
- [Ebke et al., 2013] Ebke, H., Bommès, D., Campen, M., and Kobbelt, L. (2013). Qex : robust quad mesh extraction. *ACM Trans. Graph.*, 32(6) :168 :1–168 :10.
- [Edmonds, 1965] Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17 :449–467.
- [Eppstein et al., 2008] Eppstein, D., Goodrich, M. T., Kim, E., and Tamstorf, R. (2008). Motorcycle Graphs : Canonical Quad Mesh Partitioning. *Computer Graphics Forum*.
- [Eymard et al., 2000] Eymard, R., Gallouët, T., and Herbin, R. (2000). Finite volume methods. In *Solution of Equation in \mathbb{R}^n (Part 3), Techniques of Scientific Computing (Part 3)*, volume 7 of *Handbook of Numerical Analysis*, pages 713–1018. Elsevier.
- [Fremming, 2002] Fremming, N. (2002). 3d geological model construction using a 3d grid.
- [Garanzha et al., 2021] Garanzha, V. A., Kaporin, I. E., Kudryavtseva, L. N., Protais, F., Ray, N., and Sokolov, D. (2021). Foldover-free maps in 50 lines of code. *CoRR*, abs/2102.03069.
- [Gringarten et al., 2009] Gringarten, E., Haouesse, A., Arpat, B., and Nghiem, L. (2009). Advantages of Using Vertical Stair Step Faults in Reservoir Grids for Flow Simulation. In *Proc. SPE Reservoir Simulation Conference*.
- [Gringarten et al., 2017] Gringarten, E., Lecuyer, J. D., Villarubias, E., Cosson, C., and Li, W.-C. (2017). Optimized Grids for Accurately Representing Geology in Geomechanical Simulations. In *Proc. SPE Annual Technical Conference and Exhibition*.

-
- [Gross and Mazuyer, 2021] Gross, H. and Mazuyer, A. (2021). GEOSX : A Multiphysics, Multilevel Simulator Designed for Exascale Computing. In *SPE Reservoir Simulation Conference*, page D011S010R007, On-Demand. SPE.
- [Hertzmann and Zorin, 2000] Hertzmann, A. and Zorin, D. (2000). Illustrating smooth surfaces. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 517–526, USA. ACM Press/Addison-Wesley Publishing Co.
- [Hoffman et al., 2003] Hoffman, K., Neave, J., and Klein, R. (2003). Streamlining the workflow from structure model to reservoir grid.
- [Hu and Zhang, 2016] Hu, K. and Zhang, Y. J. (2016). Centroidal voronoi tessellation based polycube construction for adaptive all-hexahedral mesh generation. *Computer Methods in Applied Mechanics and Engineering*, 305 :405–421.
- [Huang et al., 2011] Huang, J., Tong, Y., Wei, H., and Bao, H. (2011). Boundary aligned smooth 3d cross-frame field. *ACM Trans. Graph.*, 30(6) :1–8.
- [Hurtado et al., 1996] Hurtado, F., Noy, M., and Urrutia, J. (1996). Flipping edges in triangulations. In Whitesides, S., editor, *Proceedings of the Twelfth Annual Symposium on Computational Geometry, Philadelphia, PA, USA, May 24-26, 1996*, pages 214–223. ACM.
- [Jiang et al., 2015] Jiang, T., Fang, X., Huang, J., Bao, H., Tong, Y., and Desbrun, M. (2015). Frame field generation through metric customization. *ACM Trans. Graph.*, 34(4).
- [Kälberer et al., 2007] Kälberer, F., Nieser, M., and Polthier, K. (2007). Quadcover - surface parameterization using branched coverings. *Comput. Graph. Forum*, 26(3) :375–384.
- [Klemetsdal et al., 2017] Klemetsdal, Ø. S., Berge, R. L., Lie, K. A., Nilsen, H. M., and Møyner, O. (2017). Unstructured gridding and consistent discretizations for reservoirs with faults and complex wells. In *Proc. SPE Reservoir Simulation Conference*.
- [Knöppel et al., 2013] Knöppel, F., Crane, K., Pinkall, U., and Schröder, P. (2013). Globally optimal direction fields. *ACM Trans. Graph.*, 32(4).
- [Knupp, 2003] Knupp, P. M. (2003). Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elements in Analysis and Design*, 39(3) :217–241.
- [Koch et al., 2019] Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., and Panozzo, D. (2019). Abc : A big cad model dataset for geometric deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Korf, 2003] Korf, R. E. (2003). Optimal rectangle packing : Initial results. In Giunchiglia, E., Muscettola, N., and Nau, D. S., editors, *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS 2003), June 9-13, 2003, Trento, Italy*, pages 287–295. AAAI.
- [Ledoux, 2019] Ledoux, F. (2019). Mambo dataset. <https://gitlab.com/franck.ledoux/mambo>.
- [Lévy et al., 2002] Lévy, B., Petitjean, S., Ray, N., and Maillot, J. (2002). Least squares conformal maps for automatic texture atlas generation. In ACM, editor, *ACM SIGGRAPH conference proceedings*.
- [Li et al., 1995] Li, T., McKeag, R., and Armstrong, C. (1995). Hexahedral meshing using mid-point subdivision and integer programming. *Computer Methods in Applied Mechanics and Engineering*, 124(1) :171–193.
- [Liu et al., 2018] Liu, H., Zhang, P., Chien, E., Solomon, J., and Bommers, D. (2018). Singularity-constrained octahedral fields for hexahedral meshing. *ACM Trans. Graph.*, 37(4).

- [Liu et al., 2011] Liu, Y., Xu, W., Wang, J., Zhu, L., Guo, B., Chen, F., and Wang, G. (2011). General planar quadrilateral mesh design using conjugate direction field. *ACM Trans. Graph.*, 30(6) :1–10.
- [Livesu et al., 2021] Livesu, M., Pitzalis, L., and Cherchi, G. (2021). Optimal dual schemes for adaptive grid based hexmeshing. *ACM Transactions on Graphics (TOG)*, 41(2) :1–14.
- [Livesu et al., 2013] Livesu, M., Vining, N., Sheffer, A., Gregson, J., and Scateni, R. (2013). Polycut : Monotone graph-cuts for polycube base-complex construction. *Transactions on Graphics (Proc. SIGGRAPH ASIA 2013)*, 32(6).
- [Lopez et al., 2024] Lopez, D., Coudert-Osmont, Y., Desobry, D., Benedicto, A., Li, W.-C., Borgese, C., Ray, N., Sokolov, D., and Pellerin, J. (2024). 2.5D Hexahedral Meshing for Reservoir Simulations. *Mathematical Geosciences*.
- [Lyon et al., 2016] Lyon, M., Bommès, D., and Kobbelt, L. (2016). Hexex : Robust hexahedral mesh extraction. *ACM Transactions on Graphics*, 35(4).
- [Lyon et al., 2019] Lyon, M., Campen, M., Bommès, D., and Kobbelt, L. (2019). Parametrization quantization with free boundaries for trimmed quad meshing. *ACM Trans. Graph.*, 38(4).
- [Lyon et al., 2021a] Lyon, M., Campen, M., and Kobbelt, L. (2021a). Quad layouts via constrained t-mesh quantization. *Comput. Graph. Forum*, 40(2) :305–314.
- [Lyon et al., 2021b] Lyon, M., Campen, M., and Kobbelt, L. (2021b). Simpler quad layouts using relaxed singularities. *Computer Graphics Forum*, 40(5).
- [Mallet, 2002] Mallet, J.-L. (2002). *Geomodeling / Jean-Laurent Mallet*. Applied geostatistics series. Oxford University Press.
- [Mallison et al., 2013] Mallison, B., Sword, C., Viard, T., Milliken, W., and Cheng, A. (2013). Unstructured cut-cell grids for modeling complex reservoirs. *SPE Journal*, 19.
- [Maréchal, 2009] Maréchal, L. (2009). Advances in octree-based all-hexahedral mesh generation : handling sharp features. In *Proceedings of the 18th international meshing roundtable*, pages 65–84. Springer.
- [Marot et al., 2018] Marot, C., Pellerin, J., and Remacle, J.-F. (2018). One machine, one minute, three billion tetrahedra. *International Journal for Numerical Methods in Engineering*, 117 :967 – 990.
- [Merland et al., 2011] Merland, R., Lévy, B., and Caumon, G. (2011). Building PEBI Grids Conforming To 3D Geological Features Using Centroidal Voronoi Tessellations. In *Proc. IAMG*.
- [Merriman et al., 1992] Merriman, B., Bence, J. K., and Osher, S. (1992). *Diffusion generated motion by mean curvature*. Department of Mathematics, University of California, Los Angeles.
- [Micali and Vazirani, 1980] Micali, S. and Vazirani, V. (1980). An $O(\sqrt{|v|} |e|)$ algorithm for finding maximum matching in general graphs. pages 17–27.
- [Muller and Preparata, 1978] Muller, D. and Preparata, F. (1978). Finding the intersection of two convex polyhedra. *Theoretical Computer Science*, 7(2) :217–236.
- [Myles et al., 2014] Myles, A., Pietroni, N., and Zorin, D. (2014). Robust field-aligned global parametrization. *ACM Trans. Graph.*, 33(4).
- [Myles and Zorin, 2013] Myles, A. and Zorin, D. (2013). Controlled-distortion constrained global parametrization. *ACM Trans. Graph.*, 32(4).

-
- [Nieser et al., 2011] Nieser, M., Reitebuch, U., and Polthier, K. (2011). Cubecover- parameterization of 3d volumes. *Comput. Graph. Forum*, 30(5) :1397–1406.
- [Nocedal, 1980] Nocedal, J. (1980). Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151) :773–782.
- [Owen and Saigal, 2000] Owen, S. J. and Saigal, S. (2000). H-morph : an indirect approach to advancing front hex meshing. *International Journal for Numerical Methods in Engineering*, 49(1-2) :289–312.
- [Palacios and Zhang, 2007] Palacios, J. and Zhang, E. (2007). Rotational symmetry field design on surfaces. *ACM Trans. Graph.*, 26(3) :55–es.
- [Palagi and Aziz, 1994] Palagi, C. L. and Aziz, K. (1994). Use of voronoi grid in reservoir simulation. *SPE Advanced Technology Series*, 2 :69–77.
- [Palmer et al., 2020] Palmer, D., Bommès, D., and Solomon, J. (2020). Algebraic representations for volumetric frame fields. *ACM Trans. Graph.*, 39(2).
- [Panozzo et al., 2014] Panozzo, D., Puppo, E., Tarini, M., and Sorkine-Hornung, O. (2014). Frame fields : Anisotropic and non-orthogonal cross fields. *ACM Trans. Graph.*, 33(4).
- [Pardalos and Resende, 2002] Pardalos, P. M. P. M. and Resende, M. G. C. (2002 - 2002). *Handbook of applied optimization / edited by Panos M. Pardalos and Mauricio G.C. Resende*. Oxford University Press, Oxford ;.
- [Pellerin et al., 2015] Pellerin, J., Caumon, G., Julio, C., Mejia-Herrera, P., and Botella, A. (2015). Elements for measuring the complexity of 3d structural models : Connectivity and geometry. *Computers & Geosciences*, 76 :130–140.
- [Petersen, 1891] Petersen, J. (1891). Die Theorie der regulären graphs. *Acta Mathematica*, 15 :193 – 220.
- [Pietroni et al., 2022] Pietroni, N., Campen, M., Sheffer, A., Cherchi, G., Bommès, D., Gao, X., Scateni, R., Ledoux, F., Remacle, J., and Livesu, M. (2022). Hex-mesh generation and processing : A survey. *ACM Trans. Graph.*, 42(2).
- [Pietroni et al., 2021] Pietroni, N., Nuvoli, S., Alderighi, T., Cignoni, P., and Tarini, M. (2021). Reliable feature-line driven quad-remeshing. *ACM Trans. Graph.*, 40(4).
- [Price and Armstrong, 1997] Price, M. A. and Armstrong, C. G. (1997). Hexahedral mesh generation by medial surface subdivision : Part ii. solids with flat and concave edges. *International Journal for Numerical Methods in Engineering*, 40(1) :111–136.
- [Price et al., 1995] Price, M. A., Armstrong, C. G., and Sabin, M. (1995). Hexahedral mesh generation by medial surface subdivision : Part i. solids with convex edges. *International Journal for Numerical Methods in Engineering*, 38(19) :3335–3359.
- [Protais, 2022] Protais, F. (2022). *Maillage à dominante Polycube*. PhD thesis, Université de Lorraine.
- [Protais et al., 2022] Protais, F., Reberol, M., Ray, N., Corman, E., Ledoux, F., and Sokolov, D. (2022). Robust quantization for polycube maps. *Computer-Aided Design*, 150 :103321.
- [Ray et al., 2016] Ray, N., Sokolov, D., and Lévy, B. (2016). Practical 3d frame field generation. *ACM Trans. Graph.*, 35(6).
- [Ray et al., 2008] Ray, N., Vallet, B., Li, W. C., and Lévy, B. (2008). N-symmetry direction field design. *ACM Trans. Graph.*, 27(2).

- [Reberol et al., 2021] Reberol, M., Georgiadis, C., and Remacle, J. (2021). Quasi-structured quadrilateral meshing in gmsh - a robust pipeline for complex CAD models. *CoRR*, abs/2103.04652.
- [Remacle et al., 2012] Remacle, J.-F., Lambrechts, J., Seny, B., Marchandise, E., Johnen, A., and Geuzainet, C. (2012). Blossom-quad : A non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm. *International journal for numerical methods in engineering*, 89(9) :1102–1119.
- [Santoshini et al., 2018] Santoshini, S., Harris, S., Kashem, S., Levannier, A., Benabbou, A., Viard, T., and Macé, L. (2018). Depogrid : Next generation unstructured grids for accurate reservoir modeling and simulation.
- [Schneiders, 1996] Schneiders, R. (1996). A grid-based algorithm for the generation of hexahedral element meshes. *Engineering with computers*, 12 :168–177.
- [Schneiders and Debye, 1995] Schneiders, R. and Debye, J. (1995). Refining quadrilateral and brick element meshes. In Babuska, I., Henshaw, W. D., Oliger, J. E., Flaherty, J. E., Hopcroft, J. E., and Tezduyar, T., editors, *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, pages 53–65, New York, NY. Springer New York.
- [Schneiders et al., 1999] Schneiders, R., Schindler, R., and Weiler, F. (1999). Octree-based generation of hexahedral element meshes. *Proceedings of the 5th International Meshing Roundtable*.
- [Schöberl, 1997] Schöberl, J. (1997). Netgen an advancing front 2d/3d-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1 :41–52.
- [Sharp et al., 2021] Sharp, N., Gillespie, M., and Crane, K. (2021). Geometry processing with intrinsic triangulations.
- [Shen et al., 2016] Shen, Z., Fang, X., Liu, X., Bao, H., and Huang, J. (2016). Harmonic functions for rotational symmetry vector fields. *Comput. Graph. Forum*, 35(7) :507–516.
- [Si, 2015] Si, H. (2015). Tetgen, a delaunay-based quality tetrahedral mesh generator. 41(2).
- [Sloan, 2008] Sloan, P.-P. (2008). Stupid spherical harmonics (sh) tricks. *Game Developers Conference*.
- [Sokolov et al., 2016] Sokolov, D., Ray, N., Untereiner, L., and Lévy, B. (2016). Hexahedral-dominant meshing. *ACM Trans. Graph.*, 35(5).
- [Spatial Corp, 2024] Spatial Corp (2024). 3D Precise Mesh. <https://www.spatial.com/products/3d-precise-mesh>.
- [Tessael, 2024] Tessael (2024). <https://www.tessael.com/>.
- [Vaxman et al., 2016] Vaxman, A., Campen, M., Diamanti, O., Panozzo, D., Bommers, D., Hildebrandt, K., and Ben-Chen, M. (2016). Directional field synthesis, design, and processing. *Comput. Graph. Forum*, 35(2) :545–572.
- [Viertel and Osting, 2019] Viertel, R. and Osting, B. (2019). An approach to quad meshing based on harmonic cross-valued maps and the ginzburg–landau theory. *SIAM Journal on Scientific Computing*, 41(1) :A452–A479.
- [Wellmann and Caumon, 2018] Wellmann, F. and Caumon, G. (2018). Chapter one - 3d structural geological models : Concepts, methods, and uncertainties. volume 59 of *Advances in Geophysics*, pages 1–121. Elsevier.