AVERTISSEMENT

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

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

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

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Maillages hex-dominants : génération, simulation et évaluation

THÈSE

présentée et soutenue publiquement le 23 mars 2018

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Maxence Reberol

Composition du jury

Président :	Marie-Odile Berger	Directrice de recherche	Inria Grand-Est
Rapporteurs :	Pascal Frey	Professeur	Sorbonne Universités, UPMC
	Jean-François Remacle	Professeur	Université Catholique de Louvain
Examinateurs :	Dominique Bechmann	Professeure	Université de Strasbourg
	Raphaëlle Chaine	Professeure	Université de Lyon
Directeurs de thèse :	Bruno Lévy	Directeur de recherche	Inria Grand-Est
	Sylvain Lefebvre	Directeur de recherche	Inria Grand-Est

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Résumé

Maillages hex-dominants : génération, simulation et évaluation

Cette thèse s'intéresse à la génération, à l'utilisation et à l'évaluation des maillages hex-dominants, composés d'hexaèdres et de tétraèdres, dans la cadre de la simulation numérique par la méthode des éléments finis. Les éléments finis hexaédriques sont souvent préférés aux éléments tétraédriques car ils offrent un meilleur ratio entre précision et temps de calcul dans un certain nombre de situations. Cependant, si la génération automatique de maillages tétraédriques est aujourd'hui un domaine bien maîtrisé, ce n'est pas le cas de la génération de maillages hexaédriques alignés avec le bord, qui reste un problème largement ouvert. En l'absence de progrès significatifs, les approches actuelles se contentent de maillages hex-dominants afin de tirer parti des performances supérieures des hexaèdres et de la flexibilité géométrique des tétraèdres, qui rend possible le maillage automatique.

Dans une première partie, nous développons des algorithmes robustes pour la génération de maillages hex-dominants à partir de champs de directions, notamment pour l'isolement et le remplissage des régions difficiles à mailler (singularités et autres dégénérescences). Dans la seconde partie, nous essayons de déterminer dans quelles situations et dans quelle mesure les maillages hexaédriques, et hex-dominants générés précédemment, sont plus intéressants que les maillages tétraédriques. Ceci implique spécifiquement d'étudier plusieurs manières d'effectuer des simulations par éléments finis avec les maillages hybrides, dont une approche où nous utilisons des contraintes de continuité pour maillages non-conformes. Pour mesurer l'influence du maillage sur l'approximation des solutions, nous proposons une nouvelle méthode d'échantillonnage pour calculer très efficacement des distances globales entre solutions éléments finis définies sur des domaines compliqués.

Mots clés : simulation numérique, maillage, maillage hex-dominant, méthode des éléments finis, comparaison de maillages

Remerciements

Avant tout, je souhaite adresser mes premiers remerciements à mes deux directeurs de thèse, à qui je dois le plaisir d'avoir passé trois années passionnantes au sein de l'équipe ALICE. Merci Bruno de m'avoir guidé et de m'avoir toujours encouragé à comprendre en profondeur les différentes questions auxquelles je me suis intéressé, même lorsque celles-ci s'éloignaient des problématiques de la thèse. Merci Sylvain pour ton soutien et ces nombreuses discussions techniques très enrichissantes.

Je tiens également à remercier les membres du jury qui ont accepté d'évaluer ces travaux. Je suis très honoré de votre participation. Merci à Pascal Frey et Jean-François Remacle d'avoir rapporté ce manuscrit, et merci à Raphaëlle Chaine, Dominique Bechmann et Marie-Odile Berger d'avoir pris part au jury de soutenance.

Un grand merci aux membres d'ALICE avec qui j'ai eu le plaisir de partager ces dernières années : Nicolas, Dmitry, Laurent, Jonàs, Cédric, Erica, Jimmy, Noémie, Céline, etc et mes co-bureaux Dobrina et Samuel. Également merci à mes anciens co-thésards Jean et Jérémie ! Ces remerciements s'étendent aussi naturellement aux postdocs et stagiaires que j'ai eu l'occasion de côtoyer au cours de ces années.

Je remercie aussi toutes les personnes avec qui j'ai passé de très bons moments à Nancy. À commencer par l'ensemble des membres de l'équipe RING, avec une pensée particulière pour Gabriel, Marion et Antoine qui ont commencé et terminent en même temps cette épopée de thésard. Sans oublier la brigade de l'ambiance, Pierre et Jon, et la team ABZ qui n'a pas hésité à débarquer sur Nancy de temps en temps. Merci également à Florent sans qui je n'aurais sans doute pas eu la chance d'effectuer cette thèse. Je remercie enfin les doctorants rencontrés dans les Vosges, Florian, Anaelle, Anna-Livia, Yohann, toujours partants pour aller boire un verre.

Pour finir, un très grand merci à mes parents et à mes soeurs pour leur soutien inconditionnel tout au long de ces longues années d'études. Et bien sûr, merci Charline de m'avoir supporté, soutenu et accompagné toutes ces années !

Sommaire

Résumé **i**

Introduction **1**

Notations **5**

Partie I Fondements et état de l'art **6**

1 Maillages et simulations par la méthode des éléments finis **7**
1.1 Introduction de la méthode des éléments finis 7
1.2 Éléments finis polynomiaux locaux et approximation globale . . 16
1.2.1 Interpolation locale : espaces polynomiaux et bases de Lagrange 16
1.2.2 Espace d'approximation global 20
1.3 Influence du maillage 22
1.4 Variantes 24

2 État de l'art **29**
2.1 Génération de maillages hexaédriques et hex-dominants 29
2.1.1 Différentes approches pour le maillage hexaédrique 31
2.1.2 Les approches par champ de directions 39
2.2 Différents éléments finis pour différentes primitives 54
2.2.1 Différences entre éléments finis tétraédriques et hexaédriques 54
2.2.2 Éléments finis sur maillages hex-dominants 62
2.3 Méthodes pour comparer des solutions éléments finis 67
2.3.1 Méthode des solutions manufacturées 67
2.3.2 Comparaison de quantités d'intérêt 68
2.3.3 Calcul de distances entre solutions éléments finis 69
2.3.4 Techniques de visualisation de solutions éléments finis . . 71

Partie II Contributions **73**

3 Génération de maillages hex-dominants **74**
3.1 Maillage hex-dominant robuste par extraction de cavités 74
3.1.1 Extraction des hexaèdres et du bord des cavités 75
3.1.2 Maillage des cavités 83
3.1.3 Discussion et expériences numériques 84
3.2 Transformation des singularités volumiques en singularités linéiques 89
3.2.1 Approche par contraction 90
3.2.2 Approche par décomposition 91

4 Méthode des éléments finis et maillages hex-dominants **98**
4.1 Comparaisons de primitives sur des problèmes simples 2D 98
4.1.1 Problèmes scalaires elliptiques linéaires d'ordre deux . . . 98
4.1.2 Problèmes d'élasticité linéaire 102
4.1.3 Résumé des observations 104
4.2 Simulation éléments finis sur maillages hexa-tétraédriques 106
4.2.1 Conformité géométrique des maillages hexa-tétraédriques 106
4.2.2 Espace d'approximation continu 109
4.2.3 Validation et expériences numériques 112
4.3 Comparaisons de maillages sur des domaines compliqués 115
4.3.1 Problème de Poisson 116
4.3.2 Élasticité linéaire 118
4.3.3 Conclusions 120

5 Calcul de distances entre solutions éléments finis **123**
5.1 Approximation de la distance entre champs 124
5.1.1 Décomposition de la grille d'échantillonnage en tranches . 126
5.1.2 Échantillonnage du champ sur une grille 2D 127
5.1.3 Extension aux éléments courbes 129
5.2 Implémentation GPU avec OpenGL 131
5.2.1 Pipeline de rendu 132
5.2.2 Différence de textures 134
5.2.3 Réduction de texture 134
5.3 Validation et discussion de la méthode 135
5.3.1 Validation sur des cas analytiques de *bosses sinusoïdales* 135
5.3.2 Sensibilité aux paramètres pour des maillages et champs non-triviaux 139

5.3.3 Performances 141
5.3.4 Visualisation interactive et extraction d'information . . . 143
5.3.5 Limitations 144
5.3.6 Perspectives d'amélioration 146
5.3.7 Conclusion 148

Conclusion **149**

Bibliographie **152**

Introduction

Les logiciels de simulation numérique sont aujourd'hui utilisés dans tous les domaines de l'ingénierie et de la physique appliquée. Ils permettent de simuler précisément le comportement de phénomènes très variés, allant de l'interaction quantique entre particules aux mouvements gravitationnels des galaxies. Plus proches de l'échelle humaine, les méthodes numériques par éléments finis dominent le champ des simulations mettant en jeu des géométries complexes, construites grâce aux logiciels de conception assistée par ordinateur.

Les phénomènes physiques sont souvent modélisés par des équations aux dérivées partielles définies sur des domaines géométriques. La méthode des éléments finis [Tur+56 ; CO78] consiste à approcher numériquement les solutions des équations par des combinaisons de fonctions simples, souvent des polynômes, définies par morceaux sur les éléments géométriques simples d'un partitionnement du domaine d'étude : le maillage. Cette approche est utilisée quotidiennement par les ingénieurs des bureaux d'études s'intéressant à divers problèmes de conception contraints par des phénomènes de déformation mécanique, de diffusion thermique, d'interaction fluide-structure ou encore de propagation d'ondes électromagnétiques.

La taille, le type et la géométrie des éléments utilisés dans les maillages ont une influence significative sur la qualité des solutions numériques calculées. En dimension trois, les maillages volumiques sont généralement formés à partir de tétraèdres (quatre sommets et quatre faces triangulaires) ou d'hexaèdres (huit sommets et six faces quadrilatérales). Avec le développement de l'informatique, les simulations sont basées sur des modèles géométriques de plus en plus détaillés et il est indispensable d'utiliser des outils automatisés, ou au minimum semi-automatisés, pour générer les maillages volumiques. La génération automatique de maillages tétraédriques pour des géométries quelconques [GHS90] est aujourd'hui suffisamment mature pour être utilisée quotidiennement dans la plupart des logiciels de maillage et de simulation. Cependant, pour un certain nombre d'applications, il est plus intéressant d'utiliser des maillages hexaédriques (§2.2.1). Malgré des attentes industrielles fortes et deux à trois décennies de recherche (§2.1), la génération automatique de maillages hexaédriques alignés avec les bords des domaines est encore une question largement ouverte, pour laquelle il n'existe pas aujourd'hui de solution satisfaisante dans le cas général. En effet, la particularité des maillages hexaédriques est de posséder une structure interne très contraignante qui ne permet pas la réalisation d'opérations locales d'édition du maillage. Cette structure interne se traduit par des contraintes combinatoires d'adjacences entre éléments qui forment des surfaces traversant tout le domaine. Les mailleurs industriels actuels requièrent un grand nombre d'interventions de l'utilisateur qui doit fournir des informations topologiques aux algorithmes, par exemple en décomposant le modèle en blocs cubiques.

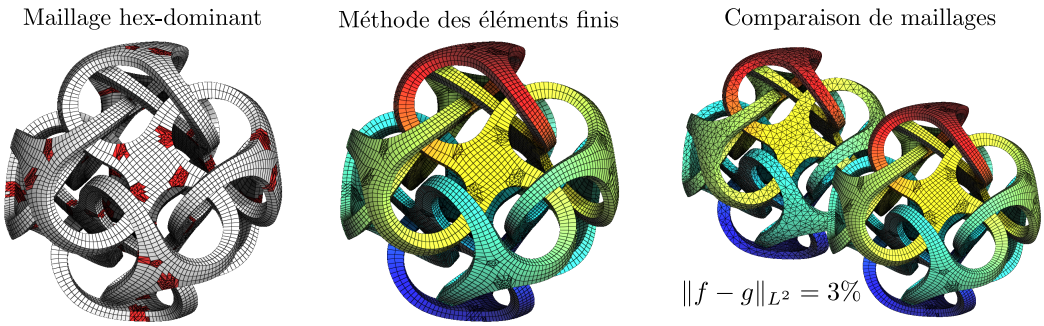


FIGURE 1 – Illustration des différents chapitres de la thèse : génération de maillages hex-dominants (gauche), simulations avec la méthode des éléments finis sur ces maillages (centre) et comparaisons de solutions numériques obtenues avec des maillages différents (droite). Modèle 3D par Bathsheba Grossman [2].

Maillages hex-dominants. La quête du maillage hexaédrique a connu récemment des progrès importants (§2.1.2) avec le développement des champs de directions qui permettent de capturer l'orientation locale des hexaèdres. Les champs de directions, composés de trois directions orthogonales, sont construits de manière à être alignés avec les bords et lisses à l'intérieur du volume. Ils servent de support aux algorithmes de construction de maillages hexaédriques, soit en guidant le positionnement des sommets (approches indirectes), soit en guidant la construction de paramétrisations à partir desquelles des hexaèdres peuvent être extraits. Cette évolution récente permet de générer automatiquement des maillages hybrides contenant une grande proportion d'hexaèdres et des tétraèdres dans les régions les plus difficiles à mailler. Dans cette thèse, nous proposons une méthode robuste pour les construire (§3.1). L'idée de ces maillages dits hex-dominants est de profiter à la fois de l'avantage de performance en simulation apporté par les hexaèdres et de la flexibilité géométrique des tétraèdres.

Simulations éléments finis. L'utilisation des maillages hex-dominants dans la méthode des éléments finis n'est pas sans difficulté car il faut construire un espace d'approximation qui réconcilie les fonctions polynomiales de différentes natures des hexaèdres et des tétraèdres. Les maillages hex-dominants conformes utilisent des pyramides (une face quadrilatérale et trois triangulaires) pour faire l'interface entre les hexaèdres et les tétraèdres. Il n'existe pas de fonctions polynomiales pour les pyramides compatibles avec celles des hexaèdres et des tétraèdres [Bed92]. Il est cependant possible d'utiliser des fonctions rationnelles ou des fonctions polynomiales composites (§2.2.2). Nous proposons une autre approche qui consiste à effectuer directement la simulation sur un maillage hexa-tétraédrique non-conforme (§4.2).

Évaluation. Avec la prolifération des maillages hex-dominants se pose la question de leur utilité concrète dans le cadre de la méthode des éléments finis. Mesurer précisément l'avantage offert par ces nouveaux maillages n'est pas évident car ils sont

2. Modèle *Metatron* : https://www.thingiverse.com/thing:1146870

développés pour des domaines aux géométries compliquées, pour lesquelles il n'existe pas de solutions analytiques. Il reste la possibilité de comparer avec des solutions de référence calculées sur des maillages très fins et d'observer la convergence de quantités d'intérêts ou de la solution approchée avec le raffinement du maillage. Le calcul de la distance entre deux solutions éléments finis définies sur des maillages volumiques distincts peut être difficile et couteux en temps de calcul, mais nous montrons que nous pouvons approcher très efficacement cette distance avec un échantillonnage régulier en exploitant des techniques d'informatique graphique (*Chapitre 5*).

En résumé, cette thèse s'intéresse aux différentes problématiques associées aux maillages hex-dominants : génération (*Chapitre 3*), utilisation et intérêt (*Chapitre 4*), et méthode d'évaluation (*Chapitre 5*).

Organisation du manuscrit

Partie I : Fondements et état de l'art.

Le *Chapitre 1* introduit la méthode des éléments finis à partir du problème de Poisson. Il est principalement dédié aux lecteurs qui ne sont pas familiers avec la méthode des éléments finis et la place des maillages dans celle-ci. Il permet également d'introduire les notions et le vocabulaire utilisés dans le reste du manuscrit.

Le *Chapitre 2* couvre les différents travaux de la littérature qui sont liés aux problématiques de cette thèse. Il comporte trois parties complémentaires. La première s'intéresse aux différentes techniques développées pour générer des maillages hexaédriques et hex-dominants (§2.1). La seconde se concentre sur la simulation par éléments finis avec maillages hexaédriques et hex-dominants, notamment sur leurs avantages rapportés dans la littérature et sur la manière de construire des espaces conformes avec des éléments de plusieurs natures mélangés dans un même maillage (§2.2). La dernière partie évoque différentes possibilités pour comparer des solutions éléments finis ainsi que les techniques de visualisation utilisées pour afficher des solutions d'ordres élevés (§2.3.3).

Partie II : Contributions.

Le *Chapitre 3* propose des contributions techniques au maillage hex-dominant à partir de paramétrisations alignées avec des champs de directions. Notamment, nous présentons une méthode robuste d'extraction des régions où la paramétrisation n'est pas de qualité suffisante pour extraire des hexaèdres. Une fois isolées, ces régions peuvent être remplies avec un maximum d'hexaèdres et des tétraèdres afin de former un maillage hex-dominant.

Le *Chapitre 4* explore les différences entre éléments finis pour tétraèdres et pour hexaèdres. Nous proposons la construction d'un espace d'approximation conforme d'ordre k pour les maillages non-conformes, basé sur l'utilisation des espaces d'interpolation locaux d'ordres k et $2k$, et de contraintes de continuité. Nous effectuons également des comparaisons de précision entre les différents types de maillages : comparaisons très simples sur le carré unité pour étudier l'influence de l'alignement des éléments, et comparaisons sur des modèles volumiques plus complexes, grâce à l'outil développé dans le *Chapitre 5*.

Le *Chapitre 5* propose un nouvel outil efficace de calcul de distances entre solutions éléments finis définies sur des maillages volumiques distincts. L'approche est basée sur un échantillonnage des champs sur une grille régulière très fine, effectué efficacement à partir d'algorithmes d'informatique graphique adaptés aux GPUs. Les techniques mises en oeuvre permettent de gérer des maillages avec des éléments courbes et des polynômes d'ordres arbitraires.

Publications associées à la thèse

Une partie des contributions présentées dans les chapitres 3, 4 et 5 a fait l'objet de communications écrites.

- L'outil de calcul de distances entre solutions éléments finis (*Chapitre 5*) est décrit dans l'article ci-dessous, accepté pour une publication imminente dans le journal *SIAM Journal on Scientific Computing*.

  Maxence Reberol *et Bruno* Lévy. *« Computing the distance between two finite element solutions defined on different 3D meshes on a GPU ». In :* SIAM Journal on Scientific Computing (accepted) *(2017).* <hal-01634176>

- La méthode robuste d'extraction des cavités des maillages hex-dominants (§3.1) est adaptée d'un article en attente de soumission.

  Nicolas Ray, *Dmitry* Sokolov, *Maxence* Reberol, *Franck* Ledoux *et Bruno* Lévy. *« Hexahedral Meshing : Mind the Gap ! » 2017.* <hal-01551603>

- La construction à l'ordre un de l'espace d'approximation conforme pour maillages est détaillée dans le rapport de recherche ci-dessous. Cette construction est étendue à un ordre k arbitraire dans ce manuscrit (§4.2).

  Maxence Reberol *et Bruno* Lévy. *« Low-order continuous finite element spaces on hybrid non-conforming hexahedral-tetrahedral meshes ». 2016. Research report.* <hal-01313285>

Notations

Conventions

- Les vecteurs et les champs vectoriels sont notés en gras, par exemple : $\mathbf{v}$
- Ω désigne le domaine du problème courant, $\Omega \subset \mathbb{R}^2$ ou $\Omega \subset \mathbb{R}^3$ dans ce manuscrit.
- $\partial\Omega$ désigne le bord du domaine Ω.
- Ω_h désigne le maillage courant de Ω (le type d'éléments utilisé dans le maillage sera précisé) et $\partial\Omega_h$ désigne le maillage associé du bord.
- Pour une fonction u solution d'un problème, u_h désigne l'approximation discrète calculée avec la méthode des éléments finis.
- δ_{ij} est le symbole de Kronecker : $\delta_{ij} = 1$ si $i = j$, $\delta_{ij} = 0$ sinon.

Opérateurs mathématiques

- grad, ou ∇ est l'opérateur gradient, par exemple en 2D : $\nabla f = \frac{\partial f}{\partial x}\mathbf{e}_x + \frac{\partial f}{\partial y}\mathbf{e}_y$.
- div$(\cdot)$ est l'opérateur divergence, par exemple en 2D : $\text{div}(\mathbf{f}) = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y}$.
- rot$(\cdot)$ est l'opérateur rotationnel défini en dimension trois :

$$\text{rot}(\mathbf{f}) = (\frac{\partial f_z}{\partial y} - \frac{\partial f_y}{\partial z})\mathbf{e}_x + (\frac{\partial f_x}{\partial z} - \frac{\partial f_z}{\partial x})\mathbf{e}_y + (\frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y})\mathbf{e}_z$$

- Δ est l'opérateur Laplacien, $\Delta = \text{div}(\text{grad}())$, par exemple en 2D :

$$\Delta f = \frac{\partial^2 f_x}{\partial x^2} + \frac{\partial^2 f_y}{\partial y^2}$$

Espaces

- $L^2(\Omega)$ est l'ensemble des fonctions de carré sommable sur Ω, *c.à-d.* $\int_\Omega |f|^2 < \infty$
- $H^1(\Omega)$ est l'ensemble des fonctions de carré sommable et dont le gradient est aussi de carré sommable.
- $H(rot, \Omega)$ est l'ensemble des fonctions de carré sommable et dont le rotationnel est aussi de carré sommable.
- $H(div, \Omega)$ est l'ensemble des fonctions de carré sommable et dont la divergence est aussi de carré sommable.
- $\mathbb{P}_k$ est l'ensemble des polynômes de degré inférieur ou égal à k.
- $\mathbb{Q}_k$ est l'ensemble des polynômes dont le degré associé à chaque variable est inférieur ou égal à k.
- V_h désigne un espace d'approximation de $H^1(\Omega)$ associé au maillage Ω_h.

Première partie

Fondements et état de l'art

---

1 Maillages et simulations par la méthode des éléments finis 7

1.1 Introduction de la méthode des éléments finis 7

1.2 Éléments finis polynomiaux locaux et approximation globale . . 16

1.3 Influence du maillage 22

1.4 Variantes 24

2 État de l'art 29

2.1 Génération de maillages hexaédriques et hex-dominants 29

2.2 Différents éléments finis pour différentes primitives 54

2.3 Méthodes pour comparer des solutions éléments finis 67

Chapitre 1

Maillages et simulations par la méthode des éléments finis

1.1 Introduction de la méthode des éléments finis

Dans cette section, nous décrivons la démarche de résolution d'un problème aux limites avec la méthode des éléments finis. L'objectif est de présenter les idées principales et non de fournir un exposé mathématique rigoureux, pour lequel le lecteur peut se reporter aux livres dédiés à la théorie des éléments finis, par exemple : [CO78 ; EG02 ; All05]. L'exposé qui suit s'inspire partiellement de ces ouvrages.

La démarche est la suivante : le problème aux limites, pour lequel on cherche une solution approchée, est transformé en une formulation faible, moins contraignante que le problème initial . La formulation faible est ensuite discrétisée sur un espace fonctionnel de dimension finie (méthode de Galerkine). L'espace de dimension finie est construit à partir de polynômes définis sur des éléments géométriques simples (*par ex.* tétraèdres) qui forment un maillage du domaine (méthode des éléments finis). Le problème discrétisé est alors réduit à un système linéaire, dont la matrice est creuse, que l'on peut résoudre efficacement en utilisant des solveurs adéquats.

Problème aux limites

Un problème aux limites est un système composé d'une ou plusieurs équations aux dérivées partielles définies à l'intérieur d'un domaine Ω et de conditions aux limites définies sur la frontière $\partial\Omega$ du domaine. Un des exemples les plus simples est le problème de Poisson :

$$\begin{cases} -\Delta u = f & \text{dans } \Omega \quad (1.1a) \\ u = u_d & \text{sur } \partial\Omega_D \quad (1.1b) \\ \nabla u \cdot \mathbf{n} = g & \text{sur } \partial\Omega_N \quad (1.1c) \end{cases}$$

où u est le champ scalaire inconnu, f un terme source, u_d la valeur du champ u imposée sur la partie $\partial\Omega_D$ de la frontière et g la composante normale du gradient de u imposée sur le reste de la frontière $\partial\Omega_N$. Ce problème aux limites est illustré sur la Figure 1.1.

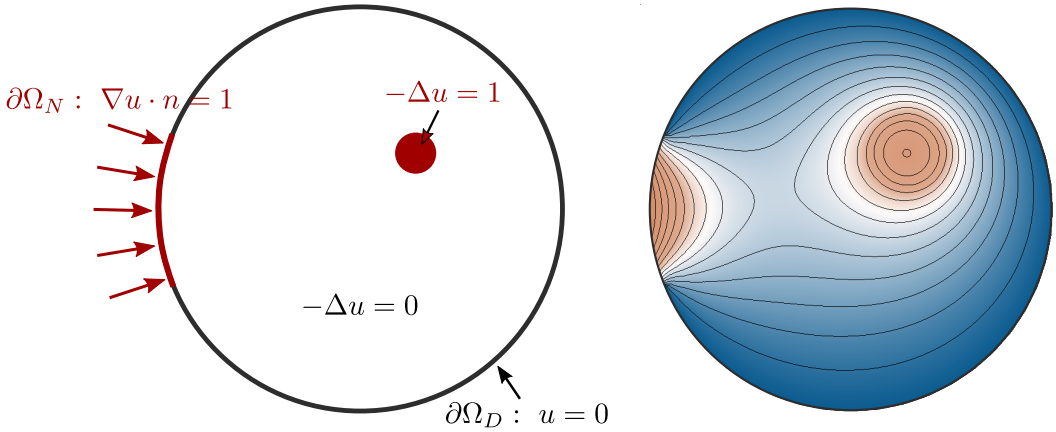


FIGURE 1.1 – Problème aux limites de Poisson. (gauche) Domaine et conditions aux limites. (droite) Solution du problème, avec les courbes iso-valeurs.

La condition aux limites (1.1b) qui impose la valeur de la solution sur le bord est appelée *condition aux limites de Dirichlet* et la condition sur le gradient (1.1c) est appelée *condition aux limites de Neumann*. D'autres conditions aux limites sont possibles (*par ex.* condition aux limites de Robin [3]) mais elles sont plus rares et nous ne les utiliserons pas dans ce manuscrit.

À noter que le problème de Poisson (1.1) est un problème aux limites très simple, il en existe d'autres plus compliqués qui mettent en jeu plusieurs champs comme le problème de Stokes (1.13, p. 24) composé de deux équations aux dérivées partielles couplées qui mettent en jeu deux champs inconnus (pression et vitesse). Dans cette thèse, nous étudions des problèmes aux limites et nous ne nous intéressons pas aux problèmes d'évolution qui introduisent d'autres difficultés liées à la discrétisation du temps.

Pour simplifier l'exposé qui suit, on se restreint au problème de Poisson homogène, c'est-à-dire avec des conditions de Dirichlet nulles : $u_d = 0$ (1.1b). Cette restriction n'est pas contraignante car pour un problème avec $u = u_d$ non nul, on peut se ramener à un problème homogène avec un changement de variable (*par ex.* voir [EG02, p. 104]).

Intégration par parties et formule de Green

Soient $u \in \mathcal{C}^2(\Omega)$ et $v \in \mathcal{C}^1(\Omega)$, alors on a l'égalité suivante, appelée formule de Green :

$$\int_\Omega \Delta u \; v = -\int_\Omega \nabla u \cdot \nabla v + \int_{\partial\Omega} (\nabla u \cdot \mathbf{n}) \; v$$

Il s'agit d'une formule d'intégration par parties. Elle est importante car la droite de l'égalité ne comporte que des dérivées partielles d'ordre un, contrairement à la partie gauche qui contient des dérivées d'ordre deux. Il est intéressant de remarquer que la partie droite a un sens mathématique si $\nabla u, \nabla v$ sont de carrés intégrables [4] (car le produit de deux fonctions carrées intégrables est intégrable). Cette condition

3. mélange de conditions de Dirichlet et de Neumann, de type $au + b\,\nabla u \cdot \mathbf{n} = g$ sur $\partial\Omega_R$
4. fonctions $\mathbf{f} \in L^2(\Omega, \mathbb{R}^n) = \{\mathbf{f} : \Omega \to \mathbb{R}^n \;/\; \int_\Omega \|\mathbf{f}\|^2 < \infty\}$, *c.à-d.* d'énergies finies

est plus faible que pour la partie de gauche de l'équation, où u doit avoir des dérivées secondes bien définies.

Formulation faible du problème aux limites

Pour obtenir la formulation faible du problème de Poisson précèdent, on multiplie l'équation (1.1a) par une fonction v, appelée fonction test :

$$-\int_\Omega \Delta u \; v = \int_\Omega f v$$

puis on applique la formule de Green à la partie gauche de l'égalité :

$$\int_\Omega \nabla u \cdot \nabla v - \int_{\partial\Omega} (\nabla u \cdot \mathbf{n}) \; v = \int_\Omega f v$$

Pour le moment, on n'a pas précisé de conditions sur la fonction test v, mais l'idée est d'utiliser le même espace fonctionnel pour la solution recherchée u et les fonctions tests. Comme on a $u = 0$ sur $\partial\Omega_D$, on impose $v = 0$ sur $\partial\Omega_D$, et on utilise la condition de Neumann (1.1c) sur $\partial\Omega_N$. L'équation ci-dessous se simplifie :

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v + \int_{\partial\Omega_N} g v \tag{1.2}$$

L'objectif de la formulation faible est de réduire les contraintes de régularité [5] sur la solution du problème. Par conséquent, on cherche à utiliser les espaces fonctionnels les plus larges possible où l'équation (1.2) a un sens. Pour que les intégrales soient définies, il faut $v, \nabla u, \nabla v, f, g \in L^2(\Omega)$, c'est-à-dire qu'elles soient de carré intégrable. Il est utile d'introduire l'espace de Sobolev H^1 :

$$H^1(\Omega) = \{u \in L^2(\Omega) \mid \nabla u \in L^2(\Omega)\} \tag{1.3}$$

Pour simplifier les notations, on peut également introduire $H^1_{\partial\Omega_D}$ l'espace des fonctions de H^1 dont la trace s'annule sur $\partial\Omega_D$:

$$H^1_{\partial\Omega_D}(\Omega) = \{u \in H^1(\Omega) \mid u = 0 \text{ sur } \partial\Omega_D\}$$

La *formulation faible* du problème de Poisson (1.1) avec des conditions de Dirichlet nulles s'écrit alors :

$$\text{Trouver } u \in H^1_{\partial\Omega_D}(\Omega) \text{ tel que } \forall v \in H^1_{\partial\Omega_D}(\Omega), \; \int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v + \int_{\partial\Omega_N} g v \tag{1.4}$$

Cette formulation est dite faible car elle moins contraignante que (1.1) : on ne cherche pas $u \in \mathcal{C}^2$ ni même $u \in \mathcal{C}^1$. La seule condition est que u possède des dérivées partielles carrées intégrables. La formulation faible (1.4) peut être vue comme un ensemble (infini) de conditions nécessaires : une pour chaque fonction test $v \in H^1_{\partial\Omega_D}(\Omega)$. À noter que si u est une solution forte du problème (1.1), on observe facilement que u est une solution faible de (1.4) par application de la formule de Green.

5. la régularité d'une fonction est l'ordre maximal des dérivées définies et continues, *par ex.* $\mathcal{C}^k(\Omega)$. Ici la notion est à interpréter au sens faible (celui des espaces de Sobolev), *c.à-d.* $f \in H^k(\Omega)$, voir [All05, chap. 4] pour la théorie associée.

Plus généralement, une formulation faible s'écrit sous la forme :

$$\text{Trouver } u \in V \text{ tel que } \forall v \in V, \quad a(u,v) = l(v)$$

où $a(\cdot,\cdot)$ est une forme bilinéaire et $l(\cdot)$ une forme linéaire. Cette formulation faible admet une solution et celle-ci est unique (*c.à-d.* que le problème est bien posé) si les hypothèses du théorème de Lax-Milgram sont vérifiées : V est un espace de Hilbert, $a(\cdot,\cdot)$ est continue et $a(\cdot,\cdot)$ est coercitive ($\exists \alpha > 0,\ a(u,u) \geq \alpha ||u||^2$).

Dans le cas de la formulation faible de Poisson qui nous sert d'exemple (1.4), les hypothèses du théorème de Lax-Milgram sont vérifiées donc on est certain de l'existence d'une solution unique. Une question plus délicate est de savoir si la solution faible obtenue est aussi solution forte de l'équation (1.1). La réponse dépend fortement de la régularité du terme source f, de la régularité du bord $\partial\Omega$ et de la condition aux limites g. La gestion des domaines non lisses ($\partial\Omega \notin \mathcal{C}^1$) et non convexes est particulièrement compliquée ; le lecteur peut se référer à [Gri11] pour plus d'informations.

À noter que le théorème de Lax-Milgram est uniquement une condition suffisante pour l'existence d'une unique solution à la formulation faible. Dans le cas général, il existe un théorème plus fort qui donne à la fois des conditions suffisantes et nécessaires : les conditions *inf-sup* (voir [EG02, p. 79]). Celles-ci s'appliquent également dans un cadre plus vaste et permettent de traiter le cas des formulations mixtes que nous évoquerons plus tard.

Discrétisation de la formulation faible

Dans la formulation faible (1.4), u et v appartiennent à $V = H^1_{\partial\Omega_D}$ qui est un espace fonctionnel de dimension infinie. La *méthode de Galerkine* consiste à proposer une version discrétisée du problème sur un sous-ensemble de dimension finie $V_h \subset V$:

$$\text{Trouver } u_h \in V_h \text{ tel que } \forall v_h \in V_h,\ \int_\Omega \nabla u_h \cdot \nabla v_h = \int_\Omega f v_h + \int_{\partial\Omega_N} g v_h \tag{1.5}$$

En notant $(\phi_i)_{i=1..n}$ une base de V_h, de dimension n, on peut écrire la décomposition de u_h sur cette base :

$$u_h = \sum_{i=j}^{n} u_j \phi_j$$

où les u_j sont n coefficients inconnus. Par linéarité, la formulation faible devient :

$$\forall v_h \in V_h,\ \sum_{j=1}^{n} u_j \int_\Omega \nabla\phi_j \cdot \nabla v_h = \int_\Omega f v_h + \int_{\partial\Omega_N} g v_h$$

Comme cette équation est linéaire et que $(\phi_i)_i$ est une base de V_h, elle est équivalente au système linéaire suivant :

$$\forall i \in [1,n],\ \sum_{j=1}^{n} u_j \int_\Omega \nabla\phi_i \cdot \nabla\phi_j = \int_\Omega f\phi_i + \int_{\partial\Omega_N} g\phi_i$$

qui est équivalent au système matriciel :

$$K\mathbf{x} = F \text{ de coefficients } k_{ij} = \int_\Omega \nabla\phi_i \cdot \nabla\phi_j,\ f_i = \int_\Omega f\phi_i + \int_{\partial\Omega_N} g\phi_i \tag{1.6}$$

La matrice K, mettant en jeu les gradients des fonctions de la base, est souvent appelée *matrice de rigidité*.

Naturellement, l'utilisation d'un sous-espace de dimension finie introduit une erreur d'approximation sur la solution obtenue via cette discrétisation (la solution exacte de la formulation faible étant un espace de dimension infinie). Deux résultats importants sur cette approximation peuvent être évoqués : l'erreur $u - u_h$ est orthogonale à V_h par rapport à la forme bilinéaire $a(.,.)$:

$$\forall v \in V_h,\ a(u - u_h, v) = 0$$

Ceci est une conséquence directe de $a(u,v) = l(v)$ et $a(u_h,v) = l(v)$ pour $v \in V_h$. Pour une forme bilinéaire $a(.,.)$ respectant les hypothèses du théorème de Lax-Milgram (continue et coercitive) s'applique également le lemme de Céa [Cea64] :

$$\exists C > 0 \text{ tel que } \|u - u_h\|_V \leq C \|u - v\|_V,\ \forall v \in V_h$$

qui veut dire que la solution approchée obtenue par la méthode de Galerkine est la meilleure approximation de la solution exacte u dans V_h à une constante près, qui ne dépend pas de la discrétisation V_h de V. Dans le cas du problème de Poisson, il s'agit de la meilleure approximation possible, mais ce n'est généralement pas le cas pour des équations aux dérivées partielles plus compliquées.

L'approche de Galerkine ci-dessus est dite conforme car elle utilise un sous espace de dimension finie V_h inclus dans V. Il existe des variantes, comme la méthode de Galerkine discontinue, où l'espace d'approximation n'est pas contenu dans V. Ceci change la formulation faible car certaines transformations utilisées ne sont plus valables. Par exemple la formule du Green, utilisable globalement dans H^1, n'est pas applicable directement pour des fonctions discontinues.

Lien avec la méthode des résidus pondérés

Note : Ce paragraphe est un aparté et n'est pas nécessaire à l'introduction des éléments finis, le lecteur peut passer directement au paragraphe suivant (**méthode des éléments finis**) s'il le souhaite.

Il est possible de faire le parallèle entre la méthode de Galerkine et la méthode des résidus pondérés. Dans cette dernière, on commence par définir une solution approchée $u_h = \sum_{i=1}^{n} u_i \phi_i$, formée par combinaison linéaire de n fonctions ϕ_i, et on cherche à minimiser la fonction résidu $R = -\Delta u_h - f$ associée à (1.1a) (ainsi que les résidus associés aux conditions aux limites). Afin de choisir les n coefficients u_i déterminants u_h, on introduit n fonctions poids w_i et on résout le système :

$$\forall i \in [1,n],\ \int_\Omega w_i\ (\Delta u_h + f) = 0$$

Si on choisit des fonctions ayant des poids identiques aux fonctions de la base de la définition de u_h (*c.à-d.* $\forall i \in [1,n],\ w_i = \phi_i$) et si on applique la formule de Green, on retrouve le système linéaire (1.6) associé à la formulation faible discrétisée (1.5).

En utilisant d'autres fonctions poids, on peut faire le lien avec d'autres méthodes numériques :

— Méthode de collocation : les fonctions poids sont basées sur des fonctions de Dirac associées à des points $\mathbf{x}_i$ du domaine

$$w_i(x) = \delta(\mathbf{x} - \mathbf{x}_i)$$

— Méthode des moindres carrés : les fonctions poids sont les dérivées du résidu R

$$w_i(x) = \frac{\partial R}{\partial u_i}$$

— Méthode des volumes finis : les fonctions poids sont définies égales à 1 sur un ensemble de cellules (partition ou maillage du domaine) et nulles ailleurs.

Le désavantage de l'approche par résidus pondérés est de travailler avec une version discrétisée des espaces fonctionnels et de ne pas avoir directement accès aux nombreuses garanties théoriques liées à l'analyse des formulations faibles (*par ex.* existence et unicité de la solution, nombreux résultats de régularité). Ceci dit, avec le choix adéquat de fonctions poids, elle est numériquement équivalente à la méthode de Galerkine.

Méthode des éléments finis

La méthode de Galerkine a permis de passer de la formulation faible (1.4) au système linéaire (1.6) en utilisant la décomposition sur une base $(\phi_i)_{i=1..n}$ de l'espace d'approximation de dimension finie V_h, avec $V_h \subset H^1_{\partial\Omega_D}$ dans notre cas. Mais nous n'avons pas encore décrit concrètement cet espace d'approximation V_h, et c'est là qu'intervient la méthode des éléments finis. Celle-ci propose de construire V_h comme une combinaison linéaire de fonctions simples (polynômes le plus souvent) définies localement sur des éléments géométriques simples d'une partition de Ω (*c.à-d.* maillage). L'intérêt est double : les fonctions ϕ_i ont un support compact (quelques éléments du maillage), ce qui implique une matrice K de (1.6) creuse et les fonctions ϕ_i étant simples (polynômes), les coefficients k_{ij}, f_i du système, qui sont des intégrales, pourront être calculés rapidement, avec des quadratures de Gauss par exemple.

En dimension deux, les éléments simples sont des triangles ou des quadrilatères. Comme la génération de maillages ne comportant que des quadrilatères n'est pas simple, il est également possible de mélanger les deux types pour former des maillages hybrides, également appelés quad-dominants (Figure 1.2). La méthode des éléments finis n'interdit pas d'utiliser des éléments plus compliqués (polygones quelconques), cependant la définition de fonctions d'interpolation devient beaucoup plus délicate dans ce cas, or l'intérêt premier de la méthode des éléments finis est d'utiliser des fonctions simples, c'est pourquoi les polygones composés de cinq arêtes et plus sont rarement utilisés.

En dimension trois, on utilise le plus souvent des tétraèdres et des hexaèdres. Dans certaines situations, les éléments finis basés sur hexaèdres sont plus performants et ceux-ci sont préférables. Cependant, la génération de maillages hexaédriques est un problème très difficile dans le cas général et est encore largement ouvert. Toutefois, ces dernières années il est devenu de plus en plus raisonnable de générer automatiquement des maillages hex-dominants (Figure 1.2), composés d'une majorité d'hexaèdres et de quelques tétraèdres.

Nous souhaitons maintenant construire l'espace de dimension finie V_h, composé de fonctions $f : \Omega \to \mathbb{R}$, à partir des éléments du maillage. Supposons que le domaine Ω est partitionné par un maillage Ω_h composé de n_e éléments $(K_i)_{i=1..n_e}$, et qu'à l'intérieur de chaque élément, les fonctions appartiennent à un espace d'interpolation polynomial $\mathcal{P}$. Alors l'espace fonctionnel V_h, composé de fonctions continues, peut

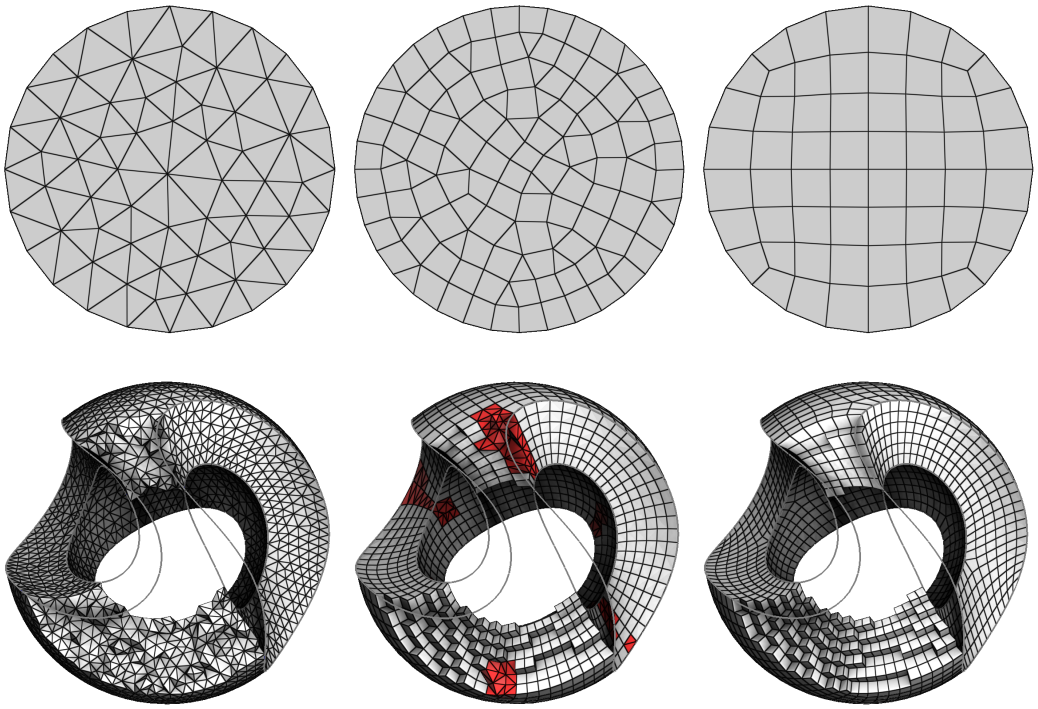


FIGURE 1.2 – Exemple de maillages. *Haut :* Maillages 2D d'un disque : triangulaire, quad-dominant et quadrilatéral. *Bas :* Maillages volumiques (3D) d'une sculpture : tétraédrique, hex-dominant (tétraèdres en rouge) et hexaédrique. Le maillage hexaédrique provient des données associées à [Li+12].

être défini par :

$$\Omega_h = \bigcup_{i=1}^{n_e} K_i, \qquad V_h = \{f \in \mathcal{C}^0(\Omega_h) \mid f_{|K_i} \in \mathcal{P},\ \forall K_i \in \Omega_h\} \tag{1.7}$$

La continuité des fonctions de V_h permet de pouvoir appliquer la formule de Green (intégration par parties), qui est utilisée pour produire la formulation faible (1.4). La description détaillée de V_h et des espaces locaux $\mathcal{P}$ est l'objet de la section 1.2. Nous supposons que $(\phi_i)_{i=1..n}$ est une base de V_h de fonctions à support compact : chaque ϕ_i est nulle sur la majorité du maillage, sauf sur son support, noté $Sup(\phi_i)$, composé de quelques éléments adjacents en pratique. Les indices des éléments du support de ϕ_i sont notés $\mathcal{S}_i$. Le support compact des fonctions peut être exploité pour réécrire les coefficients du système $K\mathbf{x} = F$ (1.6) :

$$k_{ij} = \int_{\Omega_h} \nabla\phi_i \cdot \nabla\phi_j = \int_{Sup(\phi_i)\cap Sup(\phi_j)} \nabla\phi_i \cdot \nabla\phi_j = \sum_{e\in\mathcal{S}_i\cap\mathcal{S}_j} \int_{K_e} \nabla\phi_i \cdot \nabla\phi_j$$

$$f_i = \int_{\Omega} f\phi_i + \int_{\partial\Omega_N} g\phi_i = \sum_{e\in S_i} \int_{K_e} f\phi_i + \sum_{b\in S_i\cap S(\partial\Omega_N)} \int_{\partial K_b} g\phi_i$$

Comme la plupart du temps $Sup(\phi_i) \cap Sup(\phi_j) = \emptyset$, la matrice K est creuse.

Pour des raisons d'efficacité, un code de calcul éléments finis n'itère pas sur les coefficients i, j du système mais sur les éléments, et pour chaque élément il calcule la

contribution de celui-ci à tous les coefficients affectés. Cette procédure de construction du système linéaire est appelée *assemblage*. Elle est résumée par l'algorithme 1. Dans la plupart des cas, l'assemblage du système linéaire est moins coûteux en temps de calcul que sa résolution.

Algorithme 1 Principe de la procédure d'assemblage de (1.6)

initialiser le système : $\mathbf{K} = \mathbf{0}, \mathbf{F} = \mathbf{0}$
pour chaque élément K_e de $[1, n_e]$ **faire**
  pour chaque degré de liberté i tel que $e \in S_i$ **faire**
    ajouter la contribution de K_e à f_i : $f_i = f_i + \int_{K_e} f\phi_i$
    pour chaque degré de liberté j tel que $e \in S_j$ **faire**
      ajouter la contribution de K_e à k_{ij} : $k_{ij} = k_{ij} + \int_{K_e} \nabla\phi_i \cdot \nabla\phi_j$
    fin pour
    si i sur le bord du maillage $\partial\Omega_{N,h}$ **alors**
      ajouter la contribution de K_e à f_i : $f_i = f_i + \int_{K_e \cap \partial\Omega_N} g\phi_i$
    fin si
  fin pour
fin pour

Intéressons nous maintenant à l'erreur d'approximation induite par la méthode les éléments finis. L'indice h de l'espace d'approximation V_h caractérise la taille des éléments du maillage, par exemple $h = max(diam(K_e))$ le plus grand diamètre des sphères englobantes minimales des éléments. Lorsque h tend vers 0, le nombre d'éléments dans le maillage tend vers l'infini et le sous-espace V_h tend vers V. En pratique, cela revient à raffiner le maillage de manière à obtenir une solution approchée de plus en plus précise. Il y a de nombreux travaux dont l'objectif est de borner l'erreur d'approximation commise, avec différentes normes et différentes hypothèses sur la régularité du domaine et de la solution. Sans entrer dans les détails, pour des problèmes linéaires elliptiques et en supposant que la solution est suffisamment régulière, il est possible d'aboutir à une convergence asymptotique de type [Cia91, chap. 3] :

$$\|u - u_h\|_{L^2} \leq Ch^{k+1}$$

avec h la taille caractéristique du maillage et k l'ordre d'interpolation des éléments finis utilisés (*par ex.* $k = 2$ pour des éléments quadratiques). La constante C dépend de la solution et de la qualité des éléments. De plus, cette borne est obtenue en supposant qu'avec le raffinement, la qualité géométrique des éléments ne se détériore pas. L'hypothèse sur la régularité de la solution est très importante : lorsqu'elle n'est pas vérifiée (*par ex.* domaine avec coins rentrants), la vitesse de convergence observée est souvent très inférieure h^{k+1}.

Aparté historique : La première apparition de l'idée d'approximer la solution d'une formulation variationnelle avec des fonctions linéaires par morceaux définies sur des triangles est évoquée en annexe de [Cou43] mais n'a pas été reprise. Il faut ensuite attendre les années cinquante pour que la méthode des éléments finis soit développée par des ingénieurs de Boeing [Tur+56]. Celle-ci est ensuite devenue un sujet d'étude considérable et est encore majoritairement utilisée dans les logiciels de simulation numérique car elle s'est révélée particulièrement efficace pour traiter toute

sorte d'équations aux dérivées partielles définies sur des domaines aux géométries compliquées. Le lecteur intéressé par l'histoire de la méthode des éléments finis peut se référer aux quelques pages très instructives de *The Origins of the Finite Element Method, C. A. Felippa* [Fel].

Résolution du système linéaire

La méthode des éléments finis permet de construire un système linéaire de type $AX = B$ dont la solution X est l'ensemble des coefficients associées à l'approximation $u_h \in V_h$ de la solution de la formulation faible. Cependant elle ne s'occupe pas de résoudre le système, or cette étape est très importante en pratique. Pour des problèmes 3D avec des géométries complexes, les systèmes sont souvent de taille importante, plusieurs centaines de milliers voire millions d'inconnues, ce qui pose des problèmes de résolution conséquents.

La résolution de systèmes linéaires est un domaine de recherche à part entière. Les quelques remarques qui suivent sont données à titre indicatif et sont très loin d'être exhaustives. Il y a deux grandes catégories de solveurs : les solveurs directs et les solveurs itératifs.

Les solveurs directs [Dav06], qui fonctionnent généralement par factorisation, permettent d'obtenir directement la solution exacte du système linéaire. Cependant ils sont souvent limités par la taille des systèmes qu'ils peuvent gérer. Disons approximativement qu'ils sont adaptés pour des systèmes de moins de quelques centaines de milliers d'inconnues. Il est possible d'aller plus loin en décomposant le problème en sous-domaines, mais ceci introduit d'autres difficultés liées à la parallélisation et à la communication entre les sous-domaines.

Comme leur nom l'indique, les solveurs itératifs cherchent une solution approchée $\mathbf{X}_k$ au système linéaire de manière itérative, en minimisant la norme du résidu $R_k = A\mathbf{X}_k - B$ jusqu'à ce que celui-ci soit inférieur à un critère d'arrêt ϵ. Pour les systèmes linéaires symétriques définis positifs (*par ex.* le problème de Poisson ci-dessus), l'algorithme itératif le plus populaire est le *gradient conjugué* [She+94]. Pour des systèmes plus compliqués, il faut utiliser d'autres solveurs, comme *GMRES* [SS86] par exemple. À noter que plus les problèmes deviennent importants en taille, plus leur conditionnement [6] est mauvais et les solveurs itératifs convergent très lentement, voire pas du tout. Il faut alors utiliser des pré-conditionneurs, qui sont une approximation de A^{-1}, afin de faciliter la convergence. Il existe de très nombreuses techniques, et les plus efficaces exploitent les connaissances à disposition sur l'équation aux dérivées partielles à résoudre ainsi que sur la géométrie du domaine. On peut notamment citer les méthodes *multi-grilles* [BHM00], *géométriques* si elles utilisent la géométrie du maillage ou *algébriques* si elles n'analysent que la matrice, qui construisent une hiérarchie de matrices A_i et des opérateurs de projection et prolongation pour passer d'une résolution à une autre. Dans les deux cas, l'idée est de commencer par capturer les variations spatiales basses fréquences de la solution, puis de finalement déterminer les variations plus localisées.

6. le conditionnement d'une matrice inversible est le rapport de ses valeurs singulières maximale et minimale : $\kappa = \frac{\sigma_{max}}{\sigma_{min}}$. Si la matrice est normale, c'est le ratio des valeurs propres max et min.

1.2 Éléments finis polynomiaux locaux et approximation globale

Précédemment, nous avons introduit V_h, l'espace d'approximation de dimension finie de H^1 (1.3, p. 9), qui est un espace de fonctions continues dont les restrictions sur les éléments d'un maillage Ω_h appartiennent à un espace polynomial $\mathcal{P}$:

$$V_h = \{f \in \mathcal{C}^0(\Omega_h) \mid f_{|K_i} \in \mathcal{P},\ \forall K_i \in \Omega_h\}$$

Nous commençons par caractériser l'interpolation locale à l'intérieur des éléments en définissant l'espace polynomial $\mathcal{P}$ pour les différents types d'éléments ainsi que les bases de Lagrange associées (§1.2.1). Dans un second temps, nous montrons comment la continuité des fonctions de V_h est réalisée en pratique et comment on peut définir la base globale $(\phi_i)_{i=1..n}$ de V_h à partir des interpolations locales (§1.2.2).

Afin d'éviter les redondances, nous utilisons le terme *simplexe* pour parler indistinctement des triangles en 2D et des tétraèdres en 3D, et le terme *cube* pour designer les quadrilatères en 2D et les hexaèdres en 3D.

1.2.1 Interpolation locale : espaces polynomiaux et bases de Lagrange

La méthode des éléments finis est une technique permettant de trouver les coefficients des fonctions d'interpolation sur chaque élément. Ci-dessous, nous décrivons les fonctions d'interpolation et leurs coefficients, d'abord en dimension un puis en dimensions supérieures.

Dimension un. On note $\mathbb{P}_k$ l'ensemble des polynômes réels de degré k :

$$\mathbb{P}_k = \{x \mapsto \sum_{0 \leq i \leq k} \alpha_i\ x^i,\ \alpha_i \in \mathbb{R}\}$$

La base canonique de $\mathbb{P}_k$ est l'ensemble des monômes de degrés inférieurs à k : $\{1., x, x^2, .., x^k\}$. Pour des raisons pratiques, il est parfois préférable d'utiliser la base interpolante de Lagrange $\{\mathcal{L}_0^k, ..., \mathcal{L}_k^k\}$ associée à $k+1$ points distincts $\{x_0, .., x_k\}$:

$$\mathcal{L}_i^k(x) = \prod_{0 \leq j \leq k,\ j \neq i} \frac{(x - x_j)}{(x_i - x_j)},\quad 0 \leq i \leq k \tag{1.8}$$

La propriété intéressante de cette base est que $\forall i, j \in [0, k],\ \mathcal{L}_i^k(x_j) = \delta_{ij}$, c'est-à-dire que $\mathcal{L}_i^k$ vaut 1 en x_i et 0 sur tous les autres points x_j associés à la base. On en déduit que les coefficients d'une fonction f de $\mathbb{P}_k$ dans la base de Lagrange sont les valeurs de la fonction f aux points $\{x_0, .., x_k\}$:

$$\forall f \in \mathbb{P}_k,\ \forall x \in \mathbb{R},\quad f(x) = \sum_{i=0}^{k} f(x_i)\mathcal{L}_i^k(x)$$

Les bases de Lagrange de $\mathbb{P}_1$, $\mathbb{P}_2$ et $\mathbb{P}_3$ construites avec un espacement équidistant des points de contrôle x_i, sur le segment $[0, 1]$, sont tracées sur la Figure 1.3.

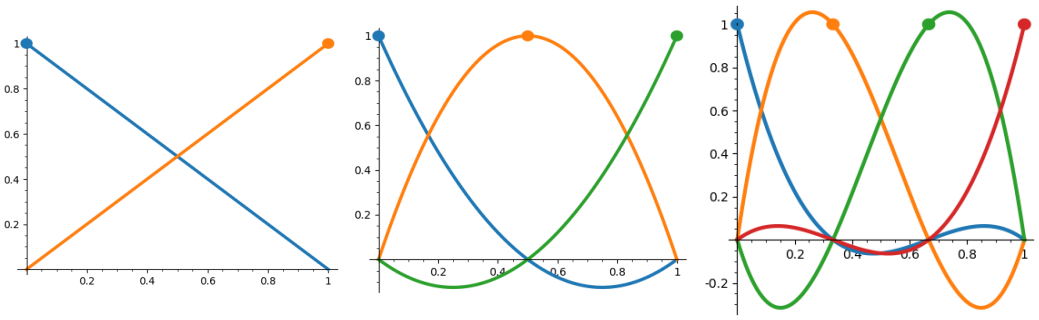


FIGURE 1.3 – Bases de Lagrange de $\mathbb{P}_k$ pour $k = 1$ (gauche, fonctions linéaires), $k = 2$ (milieu, fonctions quadratiques) et $k = 3$ (droite, fonctions cubiques). Les points représentent les couples $(x_i, 1)$ associés à chaque fonction de la base.

Si on souhaite approcher une fonction $f : [0, 1] \to \mathbb{R}$, on peut utiliser une fonction $f_h \in \mathbb{P}_k$ qui minimise l'erreur d'approximation (*par ex.* $\min_{f_h}(\|f - f_h\|_2)$), f_h pouvant être contrôlée par ses $k + 1$ valeurs aux points $(x_i)_{i=0..k}$. Pour une fonction f définie sur un domaine quelconque $[a, b]$, on peut décomposer $[a, b]$ en un ensemble d'intervalles $[I_i, I_{i+1}]$ et utiliser une fonction f_h définie sur chaque intervalle (en translatant les bases de Lagrange de $[0, 1]$ vers le sous-intervalle $[I_i, I_{i+1}]$).

Dimensions supérieures. Les idées sont similaires au cas 1D, si ce n'est que l'espace polynomial d'interpolation est différent pour les simplexes et les cubes. En dimension d, on introduit $\mathbb{P}_k^d$ l'espace des polynômes à d variables dont le degré maximal des monômes est k. Il s'agit simplement d'une généralisation de $\mathbb{P}_k$ aux dimensions supérieures. Pour le cas 2D, on a :

$$\mathbb{P}_k^{2D} = \{(x, y) \mapsto \sum_{0 \le i+j \le k} \alpha_{ij}\, x^i y^j,\ \alpha_{ij} \in \mathbb{R}\} \qquad \text{(cas 2D)} \tag{1.9}$$

$$\textit{par ex.} : \ \mathbb{P}_1 = \{(x, y) \mapsto \alpha_{00} + \alpha_{10}\, x + \alpha_{01}\, y,\ \alpha_{ij} \in \mathbb{R}\}$$
$$\mathbb{P}_2 = \{(x, y) \mapsto \alpha_{00} + \alpha_{10}\, x + \alpha_{01}\, y + \alpha_{11}\, xy + \alpha_{20}\, x^2 + \alpha_{02}\, y^2,\ \alpha_{ij} \in \mathbb{R}\}$$

Pour simplifier les notations, nous notons seulement $\mathbb{P}_k$ lorsque la dimension de l'espace d n'est pas ambigüe. Comme $\mathbb{P}_k$ est de dimension $\binom{d+k}{k}$, il faudra autant de coefficients pour déterminer entièrement un polynôme de degré k à d variables. Si on introduit $(\lambda_j)_{j=0..d}$ les $d+1$ coordonnées barycentriques associées au simplexe T, on peut former un ensemble Σ_T de $\binom{d+k}{k}$ points équi-répartis définis par [All05, p. 177] :

$$\Sigma_T = \{\mathbf{x} \in T \text{ tel que } \lambda_j(\mathbf{x}) \in \{0, \frac{1}{k}, .., \frac{k-1}{k}, 1\},\ 0 \le j \le d\}$$

Ces points sont illustrés sur la gauche de la Figure 1.4 en dimension deux. La propriété importante est que tout polynôme de $\mathbb{P}_k$ (en dimension d) est entièrement déterminé par ses valeurs sur les points de Σ_T, appelés degrés de liberté.

Une autre propriété très importante est que la restriction $f_{|F}$ d'un polynôme $f \in \mathbb{P}_k^d$ sur une face F du simplexe T est un polynôme de $\mathbb{P}_k^{d-1}$, entièrement déterminé par les degrés de liberté sur la face ($\mathbf{x} \in \Sigma_T \cap F$). Ceci implique que lorsque deux

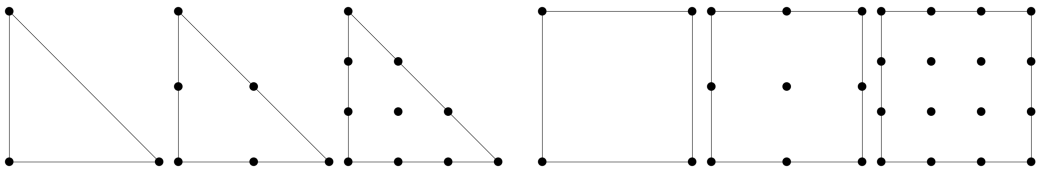


FIGURE 1.4 – Degrés de liberté (*c.à-d.* points de contrôle Σ_T, Σ_Q) des bases de Lagrange de $\mathbb{P}_k$ (gauche) et $\mathbb{Q}_k$ (droite) en dimension deux, pour $k = 1, 2, 3$ (de gauche à droite).

simplexes T_1, T_2 sont adjacents sur une face commune $F_c = T_1 \cap T_2$, le polynôme défini par morceaux sur les deux simplexes est continu si et seulement si les deux restrictions f_{T_1}, f_{T_2} ont les mêmes valeurs sur les degrés de liberté de la face commune F_c. Ceci permet de réduire le problème de continuité, d'une fonction polynomiale par morceaux, à l'égalité des degrés de liberté sur les interfaces.

Le cas des cubes (quadrilatères et hexaèdres) est différent car il est impossible d'à la fois (a) définir des degrés de liberté équi-répartis sur les arêtes et faces, et (b) d'utiliser $\binom{d+k}{k}$ degrés de liberté pour définir une base de $\mathbb{P}_k$. À la place, on utilise $\mathbb{Q}_k$ l'ensemble des polynômes dont le degré associé à chaque variable est inférieur à k. En dimension deux :

$$\mathbb{Q}_k = \{(x,y) \mapsto \sum_{0 \le i,j \le k} \alpha_{ij}\ x^i y^j,\ \alpha_{ij} \in \mathbb{R}\} \quad \text{(cas 2D)} \tag{1.10}$$

$$\begin{aligned} \textit{par ex. :}\ \mathbb{Q}_1 &= \{(x,y) \mapsto \alpha_{00} + \alpha_{10}\ x + \alpha_{01}\ y + \alpha_{11}\ xy,\ \alpha_{ij} \in \mathbb{R}\} \\ \mathbb{Q}_2 &= \{(x,y) \mapsto \alpha_{00} + \alpha_{10}\ x + \alpha_{01}\ y + \alpha_{11}\ xy \\ &\qquad + \ \alpha_{20}\ x^2 + \alpha_{02}\ y^2 + \alpha_{21}\ x^2 y + \alpha_{12}\ xy^2 + \alpha_{22}\ x^2 y^2,\ \alpha_{ij} \in \mathbb{R}\} \end{aligned}$$

Cet espace peut également être vu comme l'ensemble des polynômes formés par produit de d polynômes 1D (un pour chaque variable). Comme cet espace est de dimension $(k+1)^d$, on peut définir un ensemble de degrés de liberté Σ_Q pour un cube $Q = [0,1]^d$ par :

$$\Sigma_Q = \{\mathbf{x} \in Q \text{ tel que } x_j \in \{0, \frac{1}{k}, .., \frac{k-1}{k}, 1\},\ 1 \le j \le d\}$$

Les valeurs d'un polynôme $f \in \mathbb{Q}_k$ aux degrés de liberté Σ_Q, illustrés sur la Figure 1.4 en dimension deux, déterminent entièrement le polynôme. Comme pour le cas des simplexes, la restriction $f_{|F}$ sur une face est un polynôme de $\mathbb{Q}_k^{d-1}$, entièrement déterminé par ses valeurs aux degrés de liberté sur la face. Et de la même façon, on a la continuité d'un polynôme défini par morceaux sur les interfaces si les valeurs des degrés de liberté sont identiques sur l'interface.

Comme pour le cas 1D, nous pouvons construire explicitement des bases interpolantes, dites de Lagrange, de $\mathbb{P}_k$ et $\mathbb{Q}_k$. Les bases respectives, notées $(\mathcal{N}_i)_i$ et $(\mathcal{Q}_i)_i$, vérifient :

$$\forall \mathbf{x}_i, \mathbf{x}_j \in \Sigma_T,\ \mathcal{N}_i(\mathbf{x}_j) = \delta_{ij} \ \text{ et } \ \forall \mathbf{x}_i, \mathbf{x}_j \in \Sigma_Q,\ \mathcal{Q}_i(\mathbf{x}_j) = \delta_{ij}$$

c'est-à-dire que la i-ème fonction de la base vaut 1 sur le i-ème degré de liberté et 0 sur tous les autres. Après calculs, on peut exprimer, avec les polynômes de Lagrange

1D, les fonctions des bases en fonction des coordonnées barycentriques [7] pour $\mathbb{P}_k$ et des coordonnées classiques pour $\mathbb{Q}_k$. En dimension deux :

$$\begin{aligned} \mathcal{N}_{ijm}(\lambda_0, \lambda_1, \lambda_2) = & \quad \mathcal{L}_i^i(\lambda_0)\, \mathcal{L}_j^j(\lambda_1)\, \mathcal{L}_m^m(\lambda_2) & \text{avec } i+j+m=k \\ \mathcal{Q}_{ij}(x, y) = & \quad \mathcal{L}_i^k(x)\, \mathcal{L}_j^k(y) & \text{avec } 0 \leq i, j \leq k \end{aligned}$$

Les expressions suivent le même schéma en dimensions supérieures.

Géométries définies par transformation. La définition de l'interpolation $\mathbb{P}_k$ dans les simplexes est valable pour tout simplexe non dégénéré (volume non nul) car nous avons utilisé les coordonnées barycentriques, qui sont bien définies pour tous les simplexes définis comme étant l'enveloppe convexe de leurs sommets. Tout simplexe peut également être défini comme étant la transformation affine, ou linéaire par abus de langage, d'un simplexe de référence (*par ex.* le triangle (0,0), (1,0), (0,1)). La situation est plus délicate pour les cubes car dans les maillages, la géométrie des cubes est généralement non linéaire, par exemple la face d'un hexaèdre est composée de quatre points, qui en général ne forment pas un plan. En pratique, la géométrie des quadrilatères est définie par une fonction bilinéaire et celle des hexaèdres par une fonction trilinéaire [8], c'est-à-dire des polynômes de $\mathbb{Q}_1$. L'approche classique en éléments finis est de définir la géométrie des éléments comme étant la transformation d'éléments de référence. Par exemple en dimension trois : tétraèdre de référence $(0,0,0), (1,0,0), (0,1,0), (0,0,1)$ et cube unité $[0,1]^3$, illustrés sur la Figure 1.5. Pour un élément E, on peut écrire :

$$E = \mathcal{M}_e(\hat{E})$$

avec $\mathcal{M}_e$ la transformation et $\hat{E}$ l'élément de référence, Ces transformations (*mapping* en anglais) doivent être bijectives. Pour chaque élément du maillage, on définit alors la fonction d'interpolation sur l'élément de référence puis on la transforme vers l'élément réel. Formellement, on définit la fonction d'interpolation f par ses valeurs aux points $\mathbf{x}$ d'un élément E (défini par une transformation $\mathcal{M}_e$) par une composition :

$$\forall \mathbf{x} \in E,\ f(\mathbf{x}) = \hat{f}(\mathcal{M}_e^{-1}(\mathbf{x})) \text{ avec } \hat{f} \in \mathcal{P}$$

où $\mathcal{P} = \mathbb{P}_k,\ \mathcal{M}_e \in \mathbb{P}_1$ si E est un simplexe et $\mathcal{P} = \mathbb{Q}_k,\ \mathcal{M}_e \in \mathbb{Q}_1$ si E est un cube. La fonction $\hat{f}$, définie dans l'élément de référence [9] , est entièrement déterminée par ses valeurs aux degrés de liberté de l'élément de référence associé. Ceci détermine également la fonction f dans E car la transformation $\mathcal{M}_e$ est bijective. À première vue, cette réécriture peut sembler inutilement compliquée, mais elle a de nombreux avantages :

- on peut définir une interpolation $f = \hat{f} \circ \mathcal{M}^{-1}$ d'ordre k dans un hexaèdre trilinéaire à partir de $\hat{f} : [0,1]^3 \to \mathbb{R},\ \hat{f} \in \mathbb{Q}_k$, alors que $f \notin \mathbb{Q}_k$ ($\mathcal{M}^{-1}$ n'est pas polynomiale) !
- on peut utiliser des géométries d'ordres supérieures (*par ex.* tétraèdres quadratiques) avec le même formalisme ;
- on peut réécrire les calculs d'intégrales avec un changement de variable pour que le domaine d'intégration soit toujours l'élément de référence, de sorte qu'on ait jamais à calculer la transformation inverse $\mathcal{M}^{-1}$.

7. sur le triangle de référence $(0,0), (1,0), (0,1)$, on a : $\lambda_0 = 1 - x - y,\ \lambda_1 = x,\ \lambda_2 = y$
8. bi-affine et tri-affine pour être exact
9. espace paramétrique dans le vocabulaire de l'informatique graphique

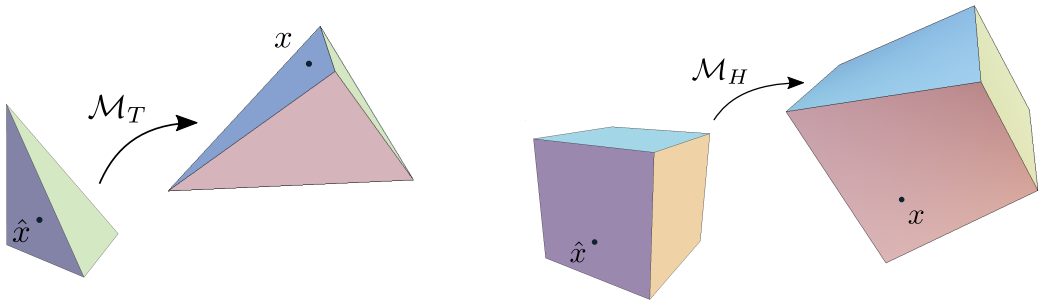


FIGURE 1.5 – Transformations de l'élément de référence vers un élément du maillage.

Intégration par changement de variable et quadratures. Les coefficients du système linéaire (1.6) peuvent être reformulées à partir des transformations géométriques introduites précédemment. Pour chaque élément K_e, on utilise la définition par composition $u = \hat{u} \circ \mathcal{M}_e^{-1}$ de l'interpolation et on applique le changement de variables $\mathbf{x} = \mathcal{M}_e(\hat{\mathbf{x}})$ aux contributions k_{ij}^e, f_i^e de l'élément K_e aux coefficients k_{ij}, f_i du système linéaire (1.6). Pour transformer les gradients, il faut penser à utiliser le théorème de dérivation des fonctions composées, *c.à-d.* $D(f \circ g) = Df \circ Dg$. Après calculs, on arrive à :

$$f_i^e = \int_{K_e} f(\mathbf{x})\phi_i(\mathbf{x})\, d\mathbf{x} = \int_{\hat{K}} f(\mathcal{M}_e(\hat{\mathbf{x}}))\ \hat{\phi}_i(\hat{\mathbf{x}})\ |det(J_{\mathcal{M}_e}(\hat{\mathbf{x}}))|\ d\hat{\mathbf{x}}$$

$$k_{ij}^e = \int_{K_e} \nabla\phi_i(\mathbf{x}) \cdot \nabla\phi_j(\mathbf{x})\, d\mathbf{x} = \int_{\hat{K}} (\nabla\hat{\phi}_i(\hat{\mathbf{x}}) J_{\mathcal{M}_e}^{-1}(\hat{\mathbf{x}})) \cdot (\nabla\hat{\phi}_j(\hat{\mathbf{x}}) J_{\mathcal{M}_e}^{-1}(\hat{\mathbf{x}}))\ |det(J_{\mathcal{M}_e}(\hat{\mathbf{x}}))|\ d\hat{\mathbf{x}}$$

avec $J_{\mathcal{M}_e}$ la matrice jacobienne associée à la transformation $\mathcal{M}_e$. Celle-ci est constante par élément pour les transformations affines (tétraèdres ou grilles régulières).

En pratique, le calcul des intégrales est remplacé par des sommes finies en utilisant les quadratures de Gauss. L'intégrale d'une fonction f sur un élément K est approximée par une somme pondérée :

$$\int_K f(\mathbf{x})d\mathbf{x} = \sum_{i=1}^{N} w_i f(\mathbf{x}_i)$$

Les quadratures doivent comporter suffisamment de points d'échantillonnage pour que l'erreur d'intégration numérique soit négligeable devant l'erreur de discrétisation des éléments finis [BW76].

Remarque. Pour les quadrilatères et hexaèdres, il existe également d'autres familles d'éléments finaux polynomiaux [Zie+13, p. 167], appelés éléments de sérendipité $\mathbb{Q}'_k$, qui contiennent tous les polynômes de degrés inférieurs à k mais moins que $\mathbb{Q}_k$: $\mathbb{P}_k \subset \mathbb{Q}'_k \subset \mathbb{Q}_k$. Ceux-ci sont parfois utilisés pour réduire de le nombre de degrés de liberté d'un problème, mais nous ne les utilisons pas dans cette thèse.

1.2.2 Espace d'approximation global

Dans la partie précédente, nous avons décrit comment une fonction pouvait être approximée par interpolation à l'intérieur d'un élément. L'objectif de cette section

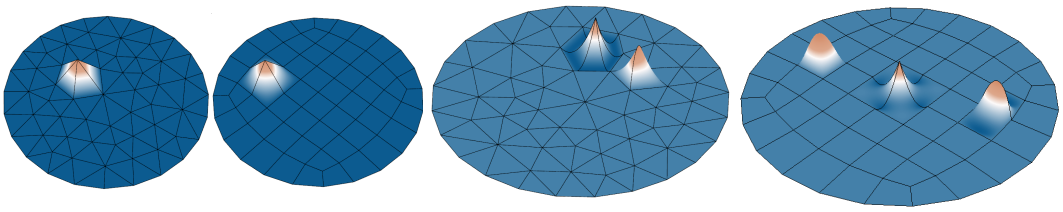


FIGURE 1.6 – Exemples de fonctions ϕ_i de la base globale de l'espace d'approximation par éléments finis. *Gauche :* ordre un, bases des fonctions linéaires par triangles et bilinéaires par quads. *Droite :* ordre deux, bases des fonctions quadratiques et bi-quadratiques. Il y a deux types de fonctions dans la base triangulaire : celles associées aux sommets du maillage et celles associées aux milieux des arêtes. Pour la base quadrilatérale, il y a ces deux types de fonctions et une fonction *bulle* associée aux centres des quadrilatères.

est de construire l'approximation globale sur tout le maillage. L'expression (1.7) de l'espace d'approximation dans le cas d'un maillage de simplexes (V_h^T) et dans le cas d'un maillage de cubes (V_h^H) devient :

$$V_h^T = \{f \in \mathcal{C}^0(\Omega_h) \mid f_{|K_e} = \hat{f}_e \circ \mathcal{M}_e^{-1},\ \hat{f}_e \in \mathbb{P}_k,\ \forall K_e \in \Omega_h\} \tag{1.11}$$

$$V_h^H = \{f \in \mathcal{C}^0(\Omega_h) \mid f_{|K_e} = \hat{f}_e \circ \mathcal{M}_e^{-1},\ \hat{f}_e \in \mathbb{Q}_k,\ \forall K_e \in \Omega_h\} \tag{1.12}$$

avec $(\mathcal{M}_e)_e$ les transformations bijectives associées à chaque élément du maillage. Nous avons vu précédemment que la continuité de f entre deux éléments adjacents K_{e_1} et K_{e_2} est équivalente à l'égalité des degrés de liberté des deux éléments sur la face commune. Pour simplifier le problème, notons $\mathbf{x}_i$ les n_{ddl} degrés de liberté (unique) du maillage Ω_h. Ceux-ci sont l'ensemble des degrés de liberté locaux (Σ_T, Σ_Q) transformés vers l'espace réel par les transformations $\mathcal{M}_e$, avec fusion des degrés de liberté qui se retrouvent au même endroit (*c.à-d.* sur les interfaces entre éléments). La *fusion*, équivalente à une contrainte d'égalité, des degrés de liberté sur les interfaces garantit la continuité des fonctions globales, définies par morceaux. Pour chaque élément K_e, on a une assignation discrète Ξ_e des degrés de liberté locaux vers les degrés de liberté globaux. Pour les hexaèdres avec interpolation $\mathbb{Q}_k$: $\Xi_e : [1, .., (k+1)^3] \to [1, .., n_{ddl}]$.

Base de V_h. La base interpolante $(\phi_i)_{i=1..n_{ddl}}$ de V_h, de dimension n_{ddl}, peut être définie par :

$$\forall i, j \in [1, n_{ddl}], \quad \phi_i \in V_h \ \text{ et } \ \phi_i(\mathbf{x}_j) = \delta_{ij}$$

À l'ordre un ($k = 1$), chaque fonction ϕ_i vaut 1 sur le i-ème sommet du maillage et 0 partout ailleurs. Ces fonctions sont souvent appelées *fonctions chapeaux*. Elles sont illustrées pour l'ordre un et deux, en dimension deux, sur la Figure 1.6.

La solution obtenue avec la méthode des éléments finis sera une combinaison linéaire des fonctions de cette base. La précision de la solution dépendra donc de la richesse des variations qui pourront être représentées avec cette ensemble de fonctions. Des exemples de solutions éléments finis $\mathbb{P}_1$ et $\mathbb{P}_2$ sont illustrés sur la figure 1.7. On peut observer l'effet du maillage triangulaire sur la solution $\mathbb{P}_1$, qui est linéaire par triangle. Les discontinuités sur les gradients aux interfaces entre triangle sont très

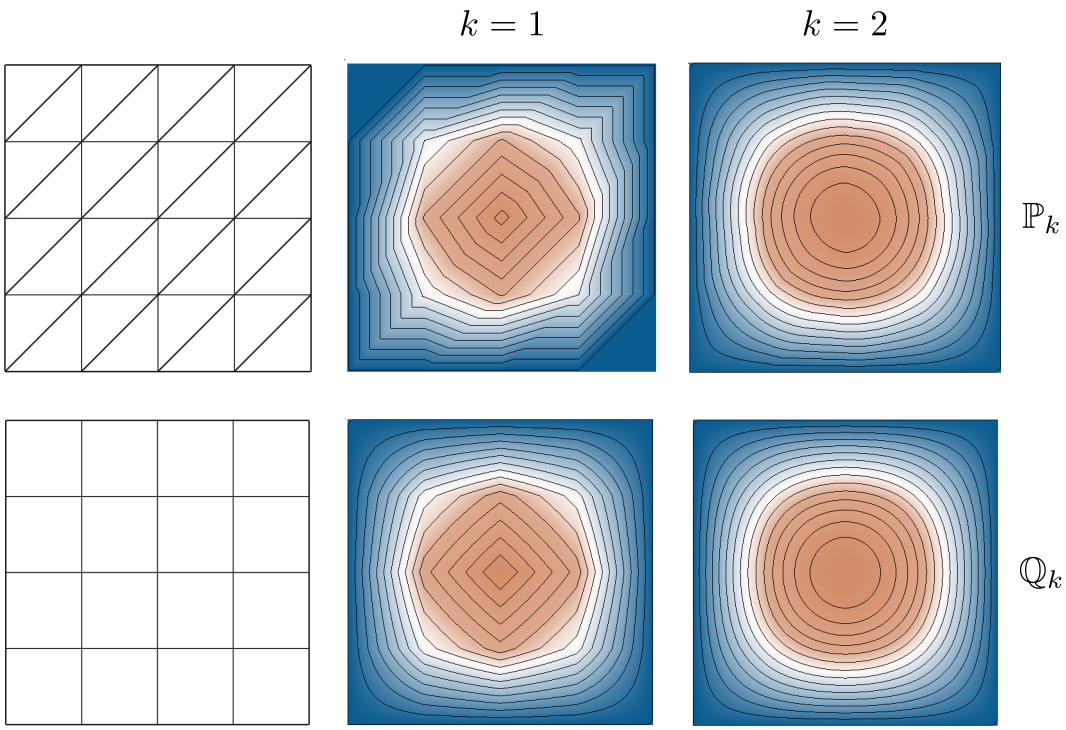


FIGURE 1.7 – Solutions éléments finis de $-\Delta u = 2\pi^2 \sin(\pi x) \sin(\pi y)$ dans Ω, $u = 0$ sur $\partial\Omega$. *Gauche :* Maillage triangulaire et quadrilatéral. *Milieu :* Éléments d'ordre un. *Droite :* Éléments d'ordre deux.

marquées. La solution quadratique par éléments ($\mathbb{P}_2$) est bien plus lisse et il est plus délicat d'observer visuellement l'effet du maillage dans ce cas précis. Ceci s'explique par le fait qu'un polynôme d'ordre deux est une bien meilleure approximation d'une fonction sinusoïdale que des polynômes d'ordre un.

Remarque. Nous avons présenté la base interpolante de Lagrange, avec une répartition équidistante des points de contrôle dans les éléments. Cependant, il est possible d'utiliser d'autres répartitions plus intéressantes (*par ex.* points de Gauss-Lobatto pour les cubes) qui ont des propriétés numériques plus avantageuses : meilleure stabilité numérique, meilleur conditionnement de la matrice de rigidité ou encore orthogonalité de la base. De plus, il est parfois intéressant d'utiliser des bases non interpolantes, comme les bases hiérarchiques de Legendre qui permettent de manipuler plus facilement des polynômes avec des ordres différents dans le même espace d'approximation (p-FEM) ou les bases de Bernstein-Bézier qui permettent des factorisations pour les ordres élevés.

1.3 Influence du maillage

Le maillage en tant que partition du domaine et support des polynômes d'interpolation est indispensable à la méthode des éléments finis. Ils permettent notamment de générer des systèmes linéaires creux que l'on peut résoudre efficacement. Mais il y a un autre aspect qu'il ne faut pas perdre de vue : l'espace d'approximation dépend

fortement des propriétés du maillage. Nous évoquons ci-dessous des conséquences liées au choix du type de maillage (simplexe ou cube). Celles-ci seront détaillées dans l'état de l'art (*Chapitre 2*) ainsi que dans les essais expérimentaux du *Chapitre 4*.

Différents polynômes d'interpolation. Les fonctions d'interpolation sont basées sur des produits de polynômes 1D ($\mathbb{Q}_k$) pour les d-cubes tandis que des polynômes multivariés ($\mathbb{P}_k$) sont utilisés pour les simplexes. À l'ordre k, les polynômes de $\mathbb{Q}_k$ ont des termes supplémentaires, que l'on appelle termes croisés dans la suite de ce manuscrit. Ces termes permettent une meilleure approximation des fonctions lisses, comme on peut l'observer sur la comparaison entre $\mathbb{P}_1$ et $\mathbb{Q}_1$ sur la Figure 1.7. De plus, un certain nombre de modifications spécifiques à certaines applications (*par ex.* mécanique), comme l'utilisation de quadratures réduites ou de modes de déplacement incompatibles, sont couramment mises en oeuvre pour les quadrilatères et hexaèdres dans les logiciels commerciaux.

Optimisation calculatoire de l'assemblage. Le calcul des coefficients (intégrales) repose sur des quadratures, c'est-à-dire des sommes pondérées. Lorsque les fonctions à intégrer sont des produits de polynômes uni-variés, comme c'est le cas pour $\mathbb{Q}_k$, il est possible de réduire le nombre d'opérations en factorisant des termes communs. Cette technique, dite de *somme-factorisation*, est particulièrement utile pour les polynômes d'ordres élevés et est une des raisons pour laquelle les maillages hexaédriques sont appréciés. Cette structure tensorielle rend également plus facile le développement de solveurs *sans matrices*, où les coefficients du système sont calculés à la volée lors de la résolution.

Maillages structurés par blocs. Les maillages tétraédriques sont fondamentalement non structurés (*par ex.* valence des sommets très variable), il faut stocker toutes les informations de connectivité. Les maillages hexaédriques ont naturellement plus de régularité. Les maillages hexaédriques structurés sont combinatoirement équivalents à des grilles régulières. Dans ce cas, de nombreuses optimisations de calcul sont possibles, par exemple on peut connaitre à l'avance la largeur de bande des matrices creuses. Pour des géométries complexes, il n'est souvent pas possible d'utiliser des maillages globalement structurés, mais il est parfois possible de se ramener à des maillages structurés par blocs, pour lesquels ces optimisations peuvent être appliquées par bloc. Cette remarque ne s'applique pas à tous les maillages hexaédriques, mais la sous-catégorie des maillages structurés (par blocs) est très intéressante pour des applications hautes performances.

Raffinement local. Comme les maillages hexaédriques sont plus ou moins structurés, il n'est généralement pas possible de leur appliquer des opérations de raffinement locales et de conserver un maillage conforme. Raffiner un hexaèdre engendre des modifications qui se propagent à travers une grande partie du modèle. Sur ce point, les maillages tétraédriques sont beaucoup plus flexibles, il existe d'ailleurs de nombreuses techniques de raffinement adaptatif pour ces derniers. Le raffinement adaptatif est particulièrement intéressant dans le cadre de la méthode des éléments finis puisqu'il

offre la possibilité d'augmenter la précision de la solution approchée tout en impactant peu le temps de calcul. Cette approche est parfois appelée h-FEM, car h caractérise la taille des éléments, en opposition à l'approche p-FEM où la précision de la solution est améliorée en augmentant le degré des polynômes d'interpolation. Les méthodes hp-FEM combinent ces deux outils.

Alignement avec les bords. Un autre avantage souvent mis en avant des maillages hexaédriques est qu'ils peuvent être alignés avec le bord des modèles 3D, ce qui correspond souvent également aux particularités des champs solutions. Par exemple dans une simulation de mécanique des fluides, le fluide va s'écouler en longeant des bords du domaine. Il est alors possible de construire des couches d'éléments hexaédriques très anisotropiques le long des bords. Ceci permet de capturer efficacement les variations importantes des champs dans les couches limites de problèmes avec interaction fluide-structure.

Nombre d'éléments et nombre de faces. Il est également intéressant de noter que les maillages hexaédriques ont besoin de moins d'éléments que les maillages tétraédriques pour couvrir le même volume. Par exemple il faut au minimum cinq tétraèdres pour mailler un cube unité. De la même manière, il y a moins d'arêtes et de faces intérieures dans les maillages hexaédriques. Dans la méthode des éléments finis classique que nous avons présentée précédemment, cela n'est pas très important car le temps de calcul dépend plutôt du nombre de degrés de liberté qui est similaire dans les deux cas. Cependant, cette différence peut se révéler plus significative pour d'autres variantes, comme la méthode de Galerkine discontinue où il faut calculer des flux entre éléments, ou encore les méthodes d'hybridation où les inconnues du système linéaire sont uniquement les degrés de liberté sur les faces intérieures et extérieures.

1.4 Variantes

L'exposé précèdent est une courte introduction à la méthode des éléments finis sur un problème simple. Mais comme la méthode des éléments finis a été appliquée avec succès depuis une cinquantaine d'années, elle a connu de nombreux développements dans des directions très différentes, afin de s'adapter au mieux aux spécificités des différents problèmes à résoudre. Dans les paragraphes suivants, nous évoquons très brièvement certains de ces développements afin que le lecteur ait connaissance de leur existence et puisse s'y référer s'il le souhaite.

Formulations mixtes Un grand nombre de problèmes physiques fait intervenir plusieurs champs couplés. Par exemple en mécanique des fluides, il y a généralement deux champs inconnus : la vitesse du fluide et la pression. Pour résoudre un tel problème avec la méthode des éléments finis, il est généralement intéressant de discrétiser les différents champs. Pour donner une idée des problématiques mises en jeu, nous évoquons brièvement le cas du problème de Stokes, qui est une modélisation de l'écoulement d'un fluide visqueux incompressible où les effets inertiels sont négligeables. Pour simplifier le problème, on suppose que la vitesse du fluide $\mathbf{u}$ est nulle sur le bord (condition de Dirichlet homogène).

$$\begin{cases} -\Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{dans } \Omega \quad (1.13a) \\ -\text{div}(\mathbf{u}) = 0 & \text{dans } \Omega \quad (1.13b) \\ \mathbf{u} = \mathbf{0} & \text{sur } \partial\Omega \quad (1.13c) \end{cases}$$

La première équation lie la vitesse $\mathbf{u}$ et la pression p tandis que la seconde équation retranscrit l'incompressibilité du fluide. Pour construire une formulation faible mixte, on multiplie la première équation par une fonction test $\mathbf{v}$ et la seconde équation par une autre fonction test q, puis on applique la formule d'intégration par partie, ce qui donne :

$$\begin{cases} \displaystyle\int_\Omega \nabla \mathbf{u} \ \nabla \mathbf{v} - \int_\Omega p \ \text{div}(\mathbf{v}) = \int_\Omega \mathbf{f}\mathbf{v} & (1.14a) \\ \displaystyle -\int_\Omega \text{div}(\mathbf{u}) \ q = 0 & (1.14b) \end{cases}$$

Pour que ces équations soient définies, il faut $\mathbf{u}, \mathbf{v} \in H^1(\Omega)$ et $p, q \in L^2(\Omega)$. Ceci dit, il n'est pas possible d'utiliser des éléments finis arbitraires de H^1 et L^2 pour trouver une solution approchée de la formulation faible. Cette situation est très différente du problème de Poisson présenté initialement, où tout sous-espace de H^1 pouvait être utilisé. Dans le cas de la formulation mixte, il faut respecter des conditions de stabilité qui sont données par la condition *inf-sup* et qui s'appliquent directement aux espaces discrets. Par exemple, l'utilisation d'éléments linéaires $\mathbb{P}_1$ pour la vitesse et $\mathbb{P}_1$ pour la pression est instable et ne permet pas de trouver de solution. Des exemples de paires compatibles sont $\mathbb{P}_k/\mathbb{P}_{k-1}$ pour $k \geq 2$, dits éléments de Taylor-Hood.

Comme les formulations mixtes sont très courantes en pratique, elles sont l'objet d'une grande partie de la littérature sur les éléments finis et il existe de nombreuses paires d'éléments stables vérifiant les conditions *inf-sup*.

Espaces $H(div)$, $H(rot)$ Jusqu'à maintenant nous avons parlé des éléments finis $\mathbb{P}_k$ et $\mathbb{Q}_k$, utilisés pour construire un sous-espace de dimension fini de l'espace H^1. Cet espace est adéquat pour la formulation faible du problème de Poisson (1.4, p. 9) car elle met en jeu le gradient de la fonction inconnue u et des fonctions tests v. Pour d'autres problèmes, et formulations faibles associées, il est possible d'utiliser des espaces moins contraignants. En pratique, H^1 impose la continuité des fonctions d'interpolation éléments finis. Mais pour certains problèmes mettant en jeu des champs vectoriels, il est plus intéressant de se contenter de la continuité de la composante normale ou de la composante tangentielle aux interfaces entre éléments. Prenons l'exemple du problème de Darcy, qui est une modélisation de l'écoulement d'un fluide en milieu poreux :

$$\begin{cases} \alpha \ \mathbf{u} + \nabla p = \mathbf{f} & \text{dans } \Omega \\ \text{div}(\mathbf{u}) = 0 & \text{dans } \Omega \\ \mathbf{u} \cdot \mathbf{n} = 0 & \text{sur } \partial\Omega \end{cases}$$

La formulation faible associée (après multiplication par $\mathbf{v}, q$ et intégration par parties) est :

$$\begin{cases} \displaystyle\int_\Omega \alpha \mathbf{u} \cdot \mathbf{v} + \int_\Omega p \ \mathrm{div}(\mathbf{v}) = \int_\Omega \mathbf{f} \cdot \mathbf{v} \quad \text{dans } \Omega \\ \displaystyle\int_\Omega q \ \mathrm{div}(\mathbf{u}) = 0 \quad \text{dans } \Omega \end{cases}$$

avec la condition $\mathbf{v} \cdot \mathbf{n} = 0$ sur la fonction test $\mathbf{v}$. Cette formulation faible est définie si $\mathbf{u}, \mathbf{v} \in \mathbf{L}^2$ et $\mathrm{div}(\mathbf{u}), \mathrm{div}(\mathbf{v}) \in L^2$. Cet espace de fonctions vectorielles est noté $\mathbf{H}(div)$:

$$\mathbf{H}(div, \Omega) = \{\mathbf{u} \in \mathbf{L}^2(\Omega) \mid \mathrm{div}(\mathbf{u}) \in L^2(\Omega)\}$$

De manière similaire, on peut définir $\mathbf{H}(rot)$:

$$\mathbf{H}(rot, \Omega) = \{\mathbf{u} \in \mathbf{L}^2(\Omega) \mid \mathrm{rot}(\mathbf{u}) \in L^2(\Omega)\}$$

qui est un espace couramment rencontré dans les formulations faibles de problèmes d'électromagnétisme.

Pour construire des discrétisations de $H(div), H(rot)$, d'autres éléments finis que $\mathbb{P}_k, \mathbb{Q}_k$ sont utilisés. On peut citer les éléments de Raviart-Thomas qui sont souvent utilisés pour $H(div)$ ainsi que les éléments de Nédélec pour $H(rot)$.

Calcul extérieur pour éléments finis. Le calcul extérieur pour la méthode des éléments finis, *Finite Element Exterior Calculus* en anglais, introduit par [AFW06], lie la théorie du calcul extérieur pour les formes différentielles et les discrétisations par éléments finis. Cette nouvelle formalisation a permis une grande unification et standardisation des concepts d'analyse numérique utilisés pour étudier la méthode des éléments finis. En plus de permettre une meilleure compréhension des concepts existants, elle est à l'origine du développement de nouveaux éléments finis stables par construction pour les formulations mixtes. On se contente ici d'évoquer une version simplifiée de l'idée principale. Le complexe de De Rham (séquence supérieure du diagramme ci-dessous) exprime l'idée que les images d'un espace par un opérateur sont dans l'espace à leur droite dans la séquence. Par exemple les gradients de fonctions de H^1 sont des fonctions de $H(rot)$. De plus cette séquence est exacte, *c.à-d.* le noyau de rot est l'image de $grad$ et le noyau div est l'image de rot.

$$\begin{array}{ccccccc} H^1 & \xrightarrow{grad} & H(rot) & \xrightarrow{rot} & H(div) & \xrightarrow{div} & L^2 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \\ V_h^1 & \xrightarrow[grad]{} & V_h^r & \xrightarrow[rot]{} & V_h^d & \xrightarrow[div]{} & V_h^s \end{array}$$

L'idée du calcul extérieur en éléments finis est que les sous-espaces discrets, c'est-à-dire les espaces d'approximation éléments finis sur la ligne inférieure du diagramme, doivent également former une séquence exacte, et les projections des espaces continus vers les discrets doivent être bornées. Lorsque le diagramme commute, c'est-à-dire que la discrétisation de la dérivée d'une fonction est égale à la dérivée de la fonction discrétisée, alors les éléments finis des espaces discrets sont stables et ils peuvent être utilisés dans des formulations mixtes, car ils vérifient les conditions *inf-sup* par construction. Cette approche systématique de la construction d'éléments finis stables

est très intéressante en pratique. Elle a même donné lieu à une classification sous forme de *tableau périodique des éléments finis* [AL14].

La théorie complète du calcul extérieur pour éléments finis est plus générale (et compliquée) que l'explication ci-dessus, qui est très approximative. Ceci dit, nous avons pensé qu'il est important de la mentionner vu son rôle de plus en plus prépondérant dans l'analyse numérique de la méthode des éléments finis. Le lecteur intéressé peut se reporter à [AFW10] pour une introduction détaillée.

Méthode de Galerkine discontinue. Précédemment, nous avons regardé comment les éléments $\mathbb{P}_k$ et $\mathbb{Q}_k$ pouvaient être utilisés pour discrétiser H^1. Ceci impliquait la continuité des fonctions aux interfaces entre éléments, qui est réalisée en pratique par l'égalité des degrés de liberté sur les faces communes. Une autre approche [HW08 ; Riv08] consiste à ne pas imposer cette continuité entre éléments, mais à pénaliser les sauts de valeur au niveau des faces. Cette approche est dite non conforme car l'espace d'approximation V_h n'est plus un sous-espace de H^1. En particulier, il n'est plus possible d'appliquer l'intégration par partie sur l'ensemble du domaine, mais élément par élément. La formulation faible du problème de Poisson (1.4) s'écrit alors :

$$\sum_{K\in\Omega_h} \int_K \nabla u \cdot \nabla v - \sum_{K\in\Omega_h} \int_{\partial K} (\nabla u \cdot \mathbf{n})\ v = \int_\Omega fv + \int_{\partial\Omega_N} gv \tag{1.17}$$

Le second terme de la gauche de l'équation correspond à des intégrales sur les faces intérieures. Il y a différentes manières de transformer cette formulation faible qui conduisent à différentes méthodes numériques. Le lecteur peut se référer à [HW08 ; Riv08] pour plus d'informations.

L'inconvénient premier des méthodes de Galerkine discontinu est que le nombre de degrés de liberté augmente significativement par rapport à l'approche continue, mais pour certaines applications le gain de précision et de stabilité est largement favorable. On peut également évoquer une évolution récente, la méthode de Galerkine discontinue hybridable (HDG) [CGL09], où le nombre d'inconnus du système linéaire est réduit aux degrés de liberté des faces intérieures et extérieures.

Éléments finis non polynomiaux : NURBS et analyse iso-géométrique Avant d'être maillés avec des tétraèdres ou des hexaèdres, les modèles 3D sont généralement construits dans des logiciels de CAO (conception assistée par ordinateur) qui utilisent souvent des NURBS (B-splines rationnelles non uniformes) pour représenter les surfaces. L'idée de l'analyse iso-géométrique [HCB05] est d'utiliser des NURBS pour représenter la géométrie ainsi que les fonctions d'interpolation de l'approximation par éléments finis. Les avantages de cette approche sont que la géométrie de l'objet utilisée pour la simulation est exacte (pas d'approximation des bords) et que l'étape de maillage tétraédrique ou hexaédrique n'est plus nécessaire.

Cependant, dans la plupart des logiciels de CAO, seuls les bords des objets sont paramétrés par des NURBS surfaciques alors que la simulation a besoin de NURBS volumiques. Techniquement, le problème de générer une paramétrisation du volume de l'objet avec des NURBS volumiques à partir de la surface extérieure est plus ou moins équivalent au problème de générer un maillage hexaédrique, c'est-à-dire que c'est un problème très difficile.

Éléments finis sur géométries polyédrales D'autres formulations en cours de développement permettent de gérer des maillages avec des éléments polyédriques avec un nombre quelconque de faces, on peut notamment citer la méthode des éléments virtuels (VEM) [Bei+13] et les méthodes hybrides d'ordres élevés (HHO) [DE15] qui ont fait l'objet de nombreux travaux de recherche récemment. L'idée principale consiste à ne pas formuler explicitement les fonctions d'interpolations mais à les définir comme étant des solutions de problèmes locaux.

Chapitre 2

État de l'art

Les maillages hexaédriques sont particulièrement recherchés pour la simulation par éléments finis, pour les raisons évoquées en introduction (§1.3) et détaillées dans la suite (§2.2.1). Toutefois, la génération automatique de maillages hexaédriques (§2.1) est un problème difficile et les techniques actuelles sont loin d'être assez matures pour un usage industriel systématique.

Une alternative plus accessible est de générer (§2.1) et d'utiliser des maillages hex-dominants, composés de plusieurs types d'éléments, dont une majorité d'hexaèdres et des tétraèdres. Cependant, comme différents types de fonctions sont utilisés pour différents types d'éléments, il est nécessaire de trouver un moyen de les réconcilier. Différentes approches sont possibles (§2.2.2), la plupart reposent sur l'utilisation de pyramides qui permettent de faire l'interface entre hexaèdres et tétraèdres.

Générer des maillages n'est pas une fin en soi. Pour qu'ils servent au mieux les intérêts de la simulation numérique, il est nécessaire de quantifier leur influence sur les solutions numériques. Pour des simulations réalistes sans solutions analytiques, mesurer l'erreur n'est pas évident. Une possibilité est d'utiliser une solution de référence très précise mais nous verrons que cela pose un certain nombre de difficultés techniques (§2.3.3).

2.1 Génération de maillages hexaédriques et hex-dominants

La génération de maillages hexaédriques est un problème difficile dont les enjeux industriels sont importants. Par conséquent, elle a fait l'objet de nombreux travaux de recherche ces deux dernières décennies. Dans cette thèse, nous faisons le choix de traiter séparément les techniques récentes qui reposent sur les champs de directions de celles, souvent antérieures, qui ne les utilisent pas.

Les approches sans champs de directions (§2.1.1) sont également couvertes par trois états de l'art [Owe98 ; SJ07 ; SRR14] auxquels le lecteur peut se reporter pour une description détaillée. Les deux premières revues sont antérieures aux approches par champs de directions et la troisième les évoque brièvement sans entrer dans les détails.

La seconde catégorie de méthodes (§2.1.2) repose sur l'utilisation de champs de directions qui permettent de guider le maillage à l'intérieur du domaine, de manière à ce qu'il soit aligné avec les bords. Celles-ci sont plus récentes et permettent de traiter des configurations géométriques plus complexes. Cependant, elles aboutissent

le plus souvent à des maillages hex-dominants qui ne sont pas exclusivement composés d'hexaèdres. Les techniques par champs de directions sont notamment couvertes par l'état de l'art de la thèse de N. Kowalski [Kow13] et dans l'article [Arm+15]. Un des objectifs de cette section est de proposer une description complémentaire et de l'étoffer avec des résultats plus récents.

Nous traitons simultanément la génération de maillages hexaédriques et de maillages hex-dominants. Ce choix est justifié par le fait que la présence d'autres éléments (tétraèdres) est souvent le résultat d'un échec de la technique de maillage hexaédrique dans une configuration géométrique spécifique.

Structure et singularités. Pour chaque méthode de maillage, il est intéressant de regarder la structure des singularités produites. Les singularités d'un maillage quadrilatéral sont les sommets de valences différentes de quatre. Pour des maillages quadrilatéraux, les valences des sommets singuliers sont souvent soit 3, soit 5. Pour des maillages hexaédriques, les singularités sont les arêtes de valences différentes de quatre, qui en pratique sont également soit 3, soit 5. Dans un maillage hexaédrique conforme, les arêtes singulières forment des lignes qui apparaissent et disparaissent sur les bords du maillage, et qui peuvent éventuellement se rejoindre à l'intérieur. Un maillage qui ne comporte pas de singularités est dit structuré, il peut être vu comme la déformation d'une grille régulière. Ceux-ci sont les plus intéressants pour la simulation numérique car ils permettent de nombreuses optimisations numériques (voir §2.2.1.2, p. 2.2.1.2). Malheureusement de nombreux modèles ne peuvent pas être maillés sans singularités (*par ex.* un triangle en 2D requiert un point singulier au centre). Pour les maillages non-structurés, l'étude des singularités est intéressante car elle donne une information sur la structure du maillage, et particulièrement à quel point le maillage est *loin* d'un maillage structuré. Nous utiliserons régulièrement deux exemples simples, le prisme triangulaire et le tétraèdre de la Figure 2.1, pour illustrer le comportement des différentes méthodes de maillage hexaédrique.

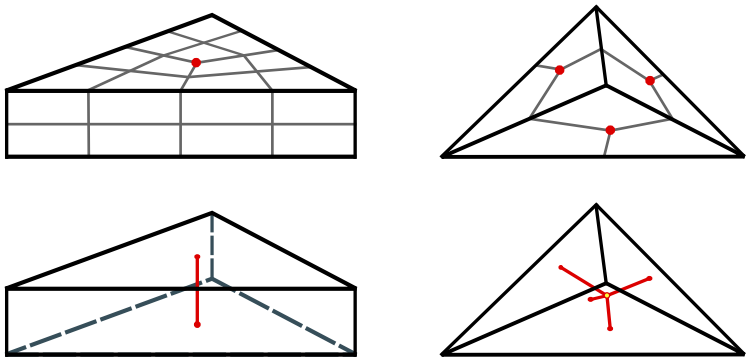


FIGURE 2.1 – Exemples de maillages hexaédriques et leurs singularités (en rouge). *Gauche :* le maillage du *prisme* contient une ligne de singularité. *Droite :* le maillage du *tétraèdre* contient quatre lignes de singularité qui se rejoignent en un noeud intérieur.

Qualité des maillages hexaédriques. Avant d'aller plus loin, il peut être utile de définir ce que nous entendons par un *bon* maillage hexaédrique, car il est toujours

possible d'en générer un *médiocre* (voir §2.1.1.1). Le maillage est un support de calcul pour une méthode de simulation numérique, un *bon* maillage est donc celui qui permet de minimiser l'erreur d'approximation de la solution pour un coût de calcul donné, mais ce critère est extrêmement dépendant du problème aux limites à résoudre. Afin de créer des techniques de maillage génériques, nous utilisons plutôt des critères géométriques, indépendant des problèmes de simulation, qui se révèlent utiles *en général*. Un critère de validité nécessaire pour la simulation par éléments finis est que le jacobien de la transformation (trilinéaire) qui transforme le cube unité en un hexaèdre soit strictement positif. Une mesure classique de la qualité d'un hexaèdre [Knu99] est le jacobien mis à l'échelle (*scaled jacobian* en anglais), qui varie entre 0 et 1 pour les éléments valides. Il s'agit du minimum du déterminant de la matrice jacobienne divisé par la norme des trois vecteurs colonnes. L'objectif est d'avoir des hexaèdres dont le jacobien mis à l'échelle tend vers 1, ce qui correspond à un pavé (tous les angles sont droits). Un autre critère plus global est de minimiser le nombre de singularités, car un trop grand nombre de celles-ci conduit à des éléments déformés. D'un autre côté, un nombre insuffisant de singularités conduit également à un maillage avec des éléments très déformés. Il faut donc trouver un bon compromis.

2.1.1 Différentes approches pour le maillage hexaédrique

Dans les paragraphes suivants, nous nous intéressons aux techniques de maillage qui ne reposent pas sur la construction de champs de directions. Ceci recouvre des approches très variées : décomposition manuelle et transformation de grilles régulières, extrusion de maillages quadrangulaires, remplissage par superposition d'octree, propagation depuis le bord, résolution d'un problème topologique, déformation cubique du domaine ou encore utilisation de diagrammes de Voronoï.

2.1.1.1 Maillage tétraédrique décomposé en hexaèdres

La génération automatique de maillages tétraédriques est aujourd'hui mature [GHS90 ; Si15a] et fonctionne de manière satisfaisante dans la très grande majorité des situations. De plus, tout tétraèdre peut être décomposé en quatre hexaèdres (droite de la Figure 2.1). Par conséquent, il est possible de possible de construire un maillage entièrement hexaédrique pour toutes les géométries en générant et décomposant un maillage tétraédrique.

Malheureusement, les résultats obtenus avec cette approche ne sont pas satisfaisants : les hexaèdres sont tous de très mauvaise qualité et le réseau de singularités du maillage est extrêmement compliqué car quatre singularités se rejoignent dans chaque tétraèdre initial et bien plus (une douzaine) sur les sommets d'origine.

2.1.1.2 Approches semi-automatique

Décomposition et transformation L'une des approches les plus anciennes exploite une décomposition du modèle en blocs "cubiques". Ensuite pour chaque bloc, un maillage structuré rectangulaire est déformé pour recouvrir la région du modèle 3D, tout en s'assurant que les interfaces entre régions sont conformes [Whi+95]. Le procédé est illustré sur la gauche de la Figure 2.2. Cette approche par *sous-transformations* est loin d'être automatique puisqu'il faut que le modèle soit manuellement décomposé en régions cubiques. Par construction, il n'y a aucune singularité

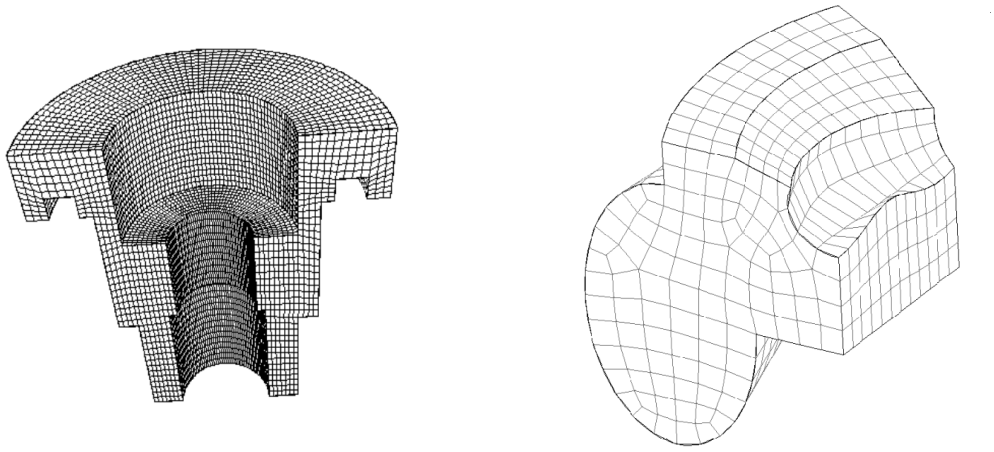


FIGURE 2.2 – Techniques de maillage hexaédrique semi-automatique. *Gauche :* Par sous-transformations, source : [Whi+95]. *Droite :* Par balayage, source : [Bla96].

possible à l'intérieur de chaque bloc, mais il est théoriquement possible d'en placer manuellement lors de la décomposition du modèle en blocs. Combinatoirement, ceci revient à fournir un maillage hexaédrique grossier du problème, dont les hexaèdres sont les blocs, et où la transformation trilinéaire d'un hexaèdre est remplacée par une transformation potentiellement plus précise.

Balayage Une autre possibilité est d'extruder un maillage quadrilatéral (2D) entre deux surfaces, la première étant *source* et la seconde *cible*. Ces deux surfaces sont manuellement définies par l'utilisateur. Le maillage quad est extrudé de manière à remplir le volume entre les deux surfaces (*par ex.* droite de la Figure 2.2). L'intérêt principal est que le maillage quadrilatéral peut être relativement complexe ; par exemple il peut contenir des points singuliers qui génèreront des lignes singulières dans le maillage hexaédrique. Il est ainsi possible de mailler le *prisme* de la Figure 2.1, mais pas le *tétraèdre*. Cette technique, appelé *balayage* ou *sweeping* en anglais, a été introduite par BLACKER [Bla96] et KNUPP [Knu98]. Elle a notamment été améliorée afin de gérer des situations avec plusieurs surfaces sources et plusieurs cibles. Les maillages obtenus sont structurés dans la direction d'extrusion. Pour plus de détails et références, le lecteur peut se référer à RUIZ-GIRONÉS et al. [RRS09]. Cette approche a des limitations pratiques : (a) l'utilisateur doit définir les surfaces sources et cibles, fournir un maillage quadrangulaire des sources et surtout (b) le modèle géométrique doit pouvoir se prêter à une telle décomposition.

Les techniques de maillage par *sous-transfomations* et *balayage* ne sont pas idéales parce qu'elles requièrent de bien décomposer les modèles, et éventuellement d'assigner des contraintes (e.g. surfaces sources et cibles) manuellement. Tous les modèles ne s'y prêtent pas, mais quand c'est le cas ces approches peuvent être très intéressantes car elles génèrent des hexaèdres de bonnes qualités. Elles sont notamment employées industriellement dans des logiciels de maillage semi-automatique comme *Trelis*[10]-

10. http://www.csimsoft.com/trelis, version commerciale de CUBIT

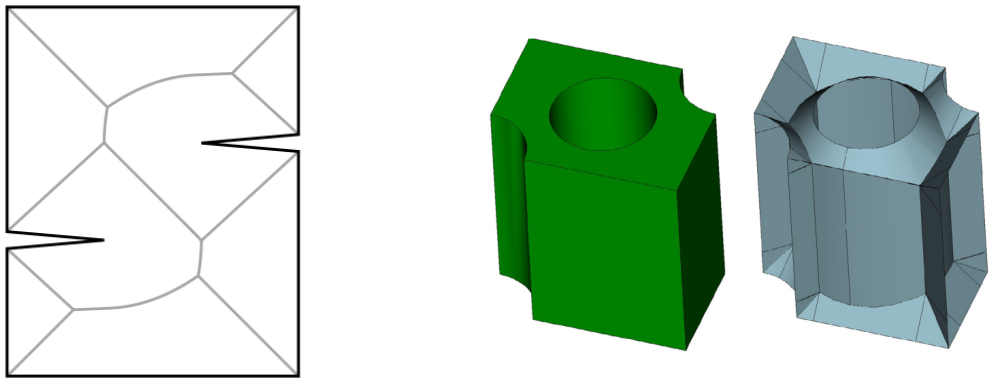


FIGURE 2.3 – Axes médians d'un objet en 2D (gauche, source : [FAR16]) et en 3D (droite, source : [Qua16]). Ils peuvent servir de guide pour une décomposition automatique du domaine.

CUBIT[11] ou *GridPro*[12].

Automatisation de la décomposition en blocs Différentes techniques ont été développées pour tenter de produire automatiquement la décomposition du modèle d'entrée en blocs qui peuvent être maillés par *transformation* ou par *balayage*.

L'axe médian du modèle est l'ensemble des points équidistants à plusieurs bords du modèle, illustré sur la Figure 2.3. Cet ensemble est également appelé *squelette* car il peut être vu comme l'ossature de l'objet. Le squelette permet de construire une décomposition de l'objet [PAS95 ; PA97], cependant celle-ci n'est pas forcément composée de cubes (*par ex.* elle peut contenir des parties triangulaires) et elle a tendance à produire des cubes *topologiques* sans se soucier de critères géométriques. Ceci dit, avec certaines modifications [FAR16], cette approche est utilisée avec succès en dimension deux sur un certain nombre de cas et avec fait toujours l'objet de recherches en dimension trois, *par ex.* [Qua16].

Une autre approche [She+99] propose d'utiliser le diagramme de Voronoï immergé pour guider la décomposition. La décomposition obtenue contient certains blocs qui peuvent être maillés avec les techniques précédentes, mais pas tous et il n'y a pas de garantie sur le type de blocs obtenus. Cette approche semble notamment fonctionner de manière satisfaisante sur les modèles contenant de nombreuses arêtes vives et peu de surfaces courbes.

Comme ces approches de décomposition ne sont pas robustes, elles peuvent être intégrées dans un logiciel semi-interactif afin d'aider l'utilisateur, tout en lui laissant le soin de gérer manuellement les configurations difficiles.

2.1.1.3 Superposition d'une grille régulière

Il existe une classe de mailleurs hexaédriques basée sur l'utilisation de grilles structurées régulières, superposées aux modèles 3D à remplir. Le plus simple est de prendre une grille cubique et d'assigner tous les cubes strictement à l'intérieur de la frontière au maillage hexaédrique [Sch96]. La partie plus délicate est de mailler le

11. https://cubit.sandia.gov/
12. http://www.gridpro.com/

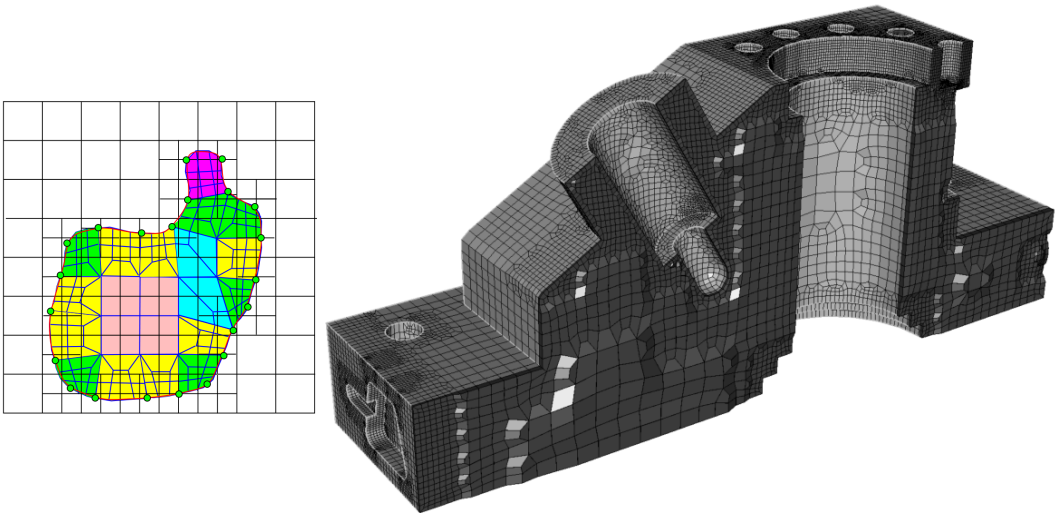


FIGURE 2.4 – Maillage par superposition d'octree. *Gauche :* Illustration du principe en 2D, source : [ZB06]. *Droite :* Exemple de maillage hexaédrique généré, source : [Mar09].

bord : il faut produire des hexaèdres qui respectent la surface d'entrée et les cubes intérieurs. De nombreuses heuristiques sont possibles, on peut notamment citer SU et al. [SLS04]. Cette méthode peut facilement être adaptée pour extraire un maillage hexaédrique d'un modèle défini par une iso-valeur dans un champ volumique d'une grille régulière [ZB06].

Deux axes principaux d'amélioration ont été développés :

— utilisation d'une grille octree au lieu d'une grille conforme [SSW96 ; Mar01]. Ceci permet de mailler plus précisément l'intérieur des modèles 3D. En contrepartie, il faut transformer les interfaces non-conformes de l'octree en un maillage hexaédrique conforme, ce qui est réalisé par l'utilisation de patrons de transition (visibles sur la Figure 2.4) ;

— insertion de couches tampons le long des bords afin d'éviter de générer des éléments de trop mauvaises qualités lors de la projection des sommets de l'octree intérieur vers le bord [ISS09 ; Mar09].

L'avantage principal de cette approche par superposition est sa robustesse. Il est possible d'extraire automatiquement des maillages entièrement hexaédriques pour toutes sortes de géométries (*par ex.* droite de la Figure 2.4). De fait, elle est utilisée par des mailleurs industriels comme *MeshGems-Hexa*[13] *-Hexotic*[14] et *Bolt*[15].

Cependant, les maillages obtenus sont très loin d'être idéaux : les hexaèdres sont alignés avec la grille utilisée et non avec les bords de l'objet, les éléments proches du bord sont souvent de qualité médiocre (très déformés) et il y a de nombreuses singularités dans le maillage, au niveau des changements de résolution de l'octree.

13. http://www.meshgems.com/volume-meshing-meshgems-hexa.html, version commerciale de *Hexotic*
14. https://www.rocq.inria.fr/gamma/
15. http://www.csimsoft.com/boltoverview

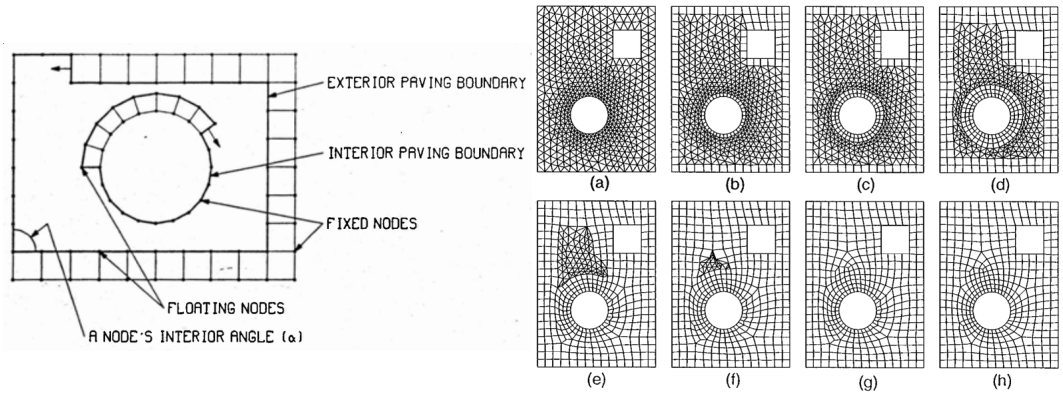


FIGURE 2.5 – Illustration en 2D de techniques de maillage par avancée de fronts. *Gauche :* Pavage, source : [BS91]. *Droite :* Q-Morph, source : [OS00].

2.1.1.4 Approches frontales géométriques

Il existe des approches qui ajoutent des hexaèdres de manière itérative, en commençant par les bords. Elles sont dites frontales car nous pouvons visualiser la progression d'un front entre les hexaèdres déjà insérés et le reste du volume à remplir. Nous citons ci-dessous deux approches qui utilisent des critères géométriques.

Approche par plâtrage Le *plâtrage* [Can92 ; BM93], *plastering* en anglais, est une généralisation en 3D de la méthode de *pavage* développé pour le maillage quadrangulaire de surfaces [BS91]. L'idée, illustrée en 2D sur la Figure 2.5, est de partir des bords de l'objet et d'insérer progressivement des éléments en se déplaçant vers l'intérieur. Cette approche a deux inconvénients importants : elle nécessite un maillage du bord (quadrangulaire en 3D) et elle échoue autour de l'axe médian du modèle, où les fronts se rejoignent. STATEN et al. [Sta+09] remédie partiellement à ces défauts en proposant une technique de *plâtrage non-contraint* qui part toujours du bord mais ne nécessite pas de maillage quadrangulaire. Ceci dit, les résultats ne sont toujours pas satisfaisants parce que la structure globale du maillage n'est pas prise en compte. Le plus souvent, il reste d'importants vides dans le modèle qui peuvent être complétés par des tétraèdres et des pyramides afin de produire un maillage hex-dominant.

Approche indirecte : H-Morph Afin d'éviter certaines difficultés liées à la génération successive des hexaèdres dans l'approche par plâtrage, OWEN et al. [OS00] proposent une approche indirecte qui repose sur l'utilisation d'un maillage tétraédrique. Un front se propage à l'intérieur du maillage et combine les tétraèdres en hexaèdres. Un des intérêts est qu'à tout moment le maillage est un maillage hexa-tétraédrique valide. Cependant, il reste toujours de nombreuses configurations dans lesquelles la recombinaison échoue.

Les deux approches présentées reposent sur des décisions locales basées sur des critères géométriques et elles ne permettent pas de générer des maillages entièrement hexaédriques pour des géométries légèrement compliquées. Les problèmes principaux apparaissent dans les régions où les différents fronts se rejoignent.

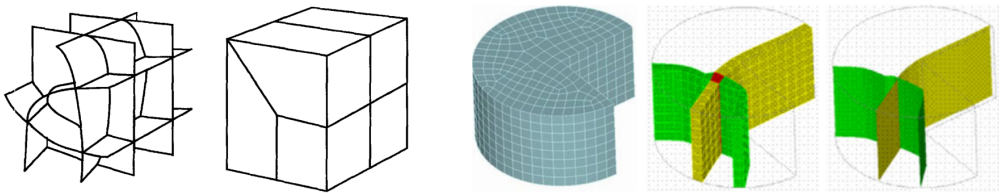


FIGURE 2.6 – *Gauche :* STC d'un maillage hexaédrique simple (source : [Mur+97]). *Droite :* deux feuilles (en vert et jaune) du STC d'un maillage hexaédrique (source : [LS10]).

2.1.1.5 Approches topologiques basées sur le maillage dual

Au lieu de construire directement le maillage hexaédrique, il est possible de travailler à la construction de son maillage dual. La notion de **Spatial Twist Continuum** (abrégé en STC) [Mur+97] a été introduite afin de caractériser la connectivité des maillages hexaédriques. Il s'agit d'une représentation de la connectivité à partir de surfaces, ou *feuilles*, et d'intersections de feuilles. Visuellement, ces feuilles peuvent être vues comme le dual du maillage hexaédrique, mais la géométrie (position des points) n'a pas d'importance. L'intérêt est que les feuilles sont des variétés de dimension deux valides alors que les surfaces formées par les faces d'hexaèdres n'en sont pas : elles se séparent au niveau des singularités. Dans le STC, les hexaèdres sont représentés par des noeuds, les faces par des lignes et les points par des volumes. Cette équivalence, illustrée sur la Figure 2.6, permet de développer des algorithmes de maillage qui se concentrent sur la topologie et non sur la géométrie. Pour une explication plus détaillée du STC, le lecteur peut se reporter à LEDOUX et al. [LS09] qui proposent une spécification précise des maillages hexaédriques et de leur dual.

Un résultat théorique intéressant, obtenu en utilisant le STC, est que tout volume, topologiquement équivalent à une boule, dont le bord est maillé avec un nombre pair de quadrilatères admet un maillage hexaédrique [Mit96]. Cependant, cette preuve d'existence n'est pas constructive et il n'est pas possible d'en déduire un algorithme systématique.

En pratique, le premier algorithme de maillage utilisant le *Spatial Twist Continuum* est le **Whisker Weaving** [TBM96], où le STC est construit par avancement de fronts. Le point de départ est un maillage quadrilatéral du bord composé d'un nombre pair d'éléments. Sur ce maillage du bord, nous pouvons énumérer toutes les *bandes* de quads qui forment une boucle (*c.à-d.* ensemble de quads adjacents par leurs arêtes opposées, qui revient sur lui même). Ces *bandes* sont les bords de surfaces, ou *feuilles* (*sheets* en anglais), qui traversent le volume et qui contraignent le maillage hexaédrique à avoir une *tranche* d'hexaèdres. Ensuite, de manière itérative, des points du STC (intersection de trois feuilles, c'est-à-dire des hexaèdres dans l'espace primal) sont insérés à l'intérieur du volume. Une fois le STC complet construit, celui-ci est transformé en maillage hexaédrique, avec lissage pour essayer d'obtenir des éléments avec des géométries satisfaisantes.

Cependant en pratique, certaines feuilles sont très compliquées et il n'est pas toujours possible de construire un maillage hexaédrique complet du volume avec l'algorithme frontal du *whisker-weaving*. De plus, il est difficile de construire des hexaèdres géométriquement valides (dont le jacobien est positif). L'algorithme initial du

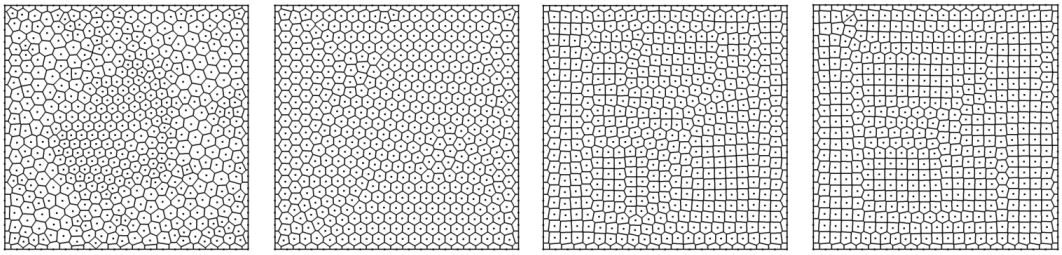


FIGURE 2.7 – Diagrammes de Voronoï barycentriques avec des différentes normes L^p. Source : [Car13].

whisker-weaving a connu de nombreuses améliorations, par exemple en supprimant les auto-intersections des bandes de quadrilatères du bord [FM99] ou en accordant d'avantage d'importance à des critères géométriques [LW08].

Une autre approche de maillage basée sur le STC est l'**élimination de cycles**, introduite par MÜLLER-HANNEMANN [Mül99]. Le point de départ est toujours un maillage quadrilatéral, avec un nombre pair d'éléments, pour lequel on a identifié les *bandes*. Dans l'approche par élimination, ces bandes sont supprimées successivement jusqu'à ce qu'il ne reste plus qu'un seul cube. Ensuite, en partant de ce cube, des feuilles sont ajoutées par extrusion dans l'ordre inverse de suppression des bandes. Comme pour le *whisker-weaving*, il faut ensuite reconstruire la géométrie du maillage à partir du STC. L'approche initiale est limitée aux volumes convexes, ou quasi-convexes, mais elle a été modifiée afin de gérer des géométries concaves [Kre+14].

Le dual du maillage hexaédrique, ou STC, est un outil intéressant pour comprendre la topologie des maillages hexaédriques. Il permet notamment d'observer comment les modifications locales ont tendance à se propager sur de longues distances dans le maillage (via les *feuilles*). De fait, les approches topologiques bénéficient de cette *vue d'ensemble* mais il demeure un grand nombre d'inconvénients : (a) le maillage quadrilatéral du bord doit être défini à priori, ce qui sur-contraint le problème, et (b) elles ignorent, ou ne considèrent pas assez, la géométrie des modèles, ce qui conduit à des maillages de très mauvaises qualités qui ne sont pas forcément adaptés pour effectuer des simulations numériques, (c) les résultats contiennent de nombreuses singularités.

On peut également remarquer que les algorithmes en question sont particulièrement complexes à comprendre et à mettre en oeuvre. En particulier, afin d'obtenir des maillages satisfaisants, les auteurs des approches topologiques ont développé de nombreuses techniques de modifications des STC [LS10 ; Gao+17b], ce qui est bien plus difficile, car non local, que des opérations équivalentes d'édition des maillages tétraédriques.

2.1.1.6 Approches par diagramme de Voronoï et normes L^∞

Le diagramme de Voronoï barycentrique (CVT) d'un ensemble de points est un diagramme de Voronoï tel que les sommets de chaque cellule sont situés aux barycentres de celles-ci. Ce diagramme, illustré sur la gauche de la Figure 2.7, peut être construit avec l'algorithme itératif de Lloyd qui consiste à minimiser une énergie

basée sur la norme euclidienne L^2. Lévy et al. [LL10] ont montré qu'en utilisant une norme L^p de degré supérieure (L_p-CVT avec $p \to \infty$), les sommets se déplacent de sorte que les cellules de Voronoï soient carrées, ou cubiques en dimension trois, comme illustré sur la droite de la Figure 2.7. Pour que les cellules soient alignées avec les bords, il faut fournir une matrice d'anisotropie à l'algorithme en tout point du domaine, par exemple en utilisant la normale du bord le plus proche [LL10].

Cette première étape permet de positionner les sommets du nouveau maillage. Les éléments, triangles en 2D ou tétraèdres en 3D, sont générés par triangulation de Delaunay et sont transformés en quadrilatères ou hexaèdres avec les algorithmes classiques de recombinaison des approches indirectes (*par ex.* [YS03]). Avec cette technique, il est possible de former des maillages entièrement quadrilatéraux en 2D mais cela fonctionne moins bien en 3D où il reste toujours une portion significative de tétraèdres.

Les inconvénients majeurs de cette approche sont qu'elle très lente et qu'elle converge vers des minima locaux (énergie minimisée non convexe) [Car13]. En dimension deux, Remacle et al. [Rem+13] proposent une approche frontale, similaire à L^p-CVT, basée sur la norme L^∞, mais plus rapide car elle ne construit pas le diagramme de Voronoï. Cependant, celle-ci n'a pas été étendue à la dimension trois à notre connaissance.

2.1.1.7 Polycubes

Il existe une approche de *transformation géométrique* du domaine, développée récemment par la communauté d'informatique graphique et dénommée *polycube*, qui peut être exploitée pour le problème du maillage hexaédrique. L'idée est de déformer un modèle 3D de manière à ce que les bords soient alignés avec les trois plans du repère (xy, xz, yz). Le modèle déformé est alors maillé avec une grille régulière, puis ce maillage est transformé vers la géométrie initiale (transformation inverse de la déformation initiale). Ce procédé a été introduit pour appliquer des textures sur des objets [Tar+04] et a été adapté pour générer des maillages hexaédriques [GSZ11]. Il est illustré sur la Figure 2.8.

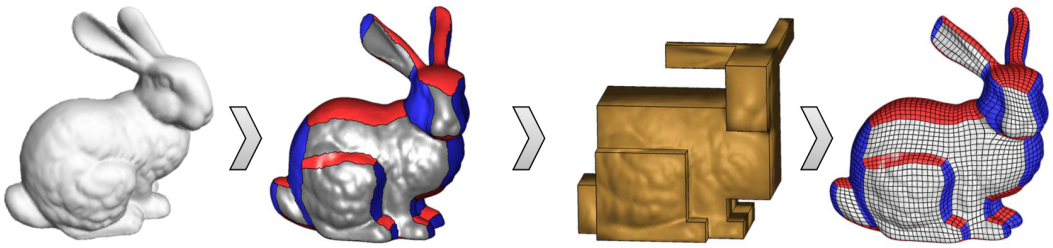


Figure 2.8 – Maillage hexaédrique par *polycube*. Étapes : étiquetage, polycube, paramétrisation. Source : http://www.cs.ubc.ca/~sheffa/hexing/hexing.htm

Une étape particulièrement importante de la construction du polycube est l'étiquetage de la surface : à chaque triangle du bord est associé une des six directions de référence ($\pm\mathbf{e}_x, \pm\mathbf{e}_y, \pm\mathbf{e}_z$). Cet étiquetage doit à la fois être fidèle à la surface d'entrée et permettre des transformations avec peu de distorsion [Liv+13]. L'étape suivante

consiste à déformer le modèle initial vers le polycube, celle-ci est généralement réalisée de manière itérative en minimisant une énergie qui contient l'alignement des normales avec les axes et un terme de régularisation qui mesure la distorsion induite par la transformation géométrique.

L'approche initiale dépend fortement de l'orientation du modèle par rapport aux axes de référence. Il est possible d'orienter le modèle de manière optimale [Hua+14], mais le polycube résultant a toujours des biais directionnels.

Comme le maillage hexaédrique final est obtenu par la déformation continue d'une grille régulière (le polycube), il ne contient aucune singularité interne. Le résultat est donc un maillage entièrement structuré. On peut faire le rapprochement avec la méthode semi-automatique par *décomposition et transformation*. Le polycube peut être vu comme une manière automatique de réaliser la décomposition en blocs et de calculer les transformations associées à chacun de ces blocs, tout en maintenant la conformité aux interfaces entre blocs. Ceci dit, il ne peut pas gérer tous les cas automatiquement, par exemple un pavé défini le long d'un demi-cercle peut être maillé en un seul bloc par la méthode de transformation (un étage du modèle à gauche de la Figure 2.2) tandis que les critères géométriques de l'approche par polycube attribueront différentes étiquettes à la partie courbe.

En pratique, les maillages hexaédriques obtenus sont de bonne qualité lorsque les modèles se prêtent à un maillage sans singularité interne. Les modèles à bord lisses, qui ne comportent pas d'arêtes vives, s'y prêtent particulièrement bien, comme c'est souvent le cas en informatique graphique. Cependant cette approche ne permet pas de traiter toutes les géométries de manière satisfaisante, et par exemple elle ne permet pas de mailler directement les modèles *prisme* et *tétraèdre* de la Figure 2.1.

Certes la déformation du maillage vers un polycube n'est pas toujours possible, mais nous pourrions imaginer une approche hybride qui déforme le modèle d'entrée le plus possible et qui utilise ensuite un mailleur hexaédrique par *superposition de grilles* pour mailler le modèle déformé. L'alignement partiel avec les bords devrait aider ce dernier à produire des hexaèdres de qualités satisfaisantes. La transformation inverse vers la géométrie initiale permettrait également d'avoir des hexaèdres mieux alignés avec le bord par rapport à l'approche par grille seule.

2.1.2 Les approches par champ de directions

Dans cette seconde partie de l'état l'art sur les méthodes de maillage hexaédrique, nous nous intéressons aux approches utilisant des champs de directions. Ce terme peut recouvrir différentes notions, dans ce manuscrit nous l'utilisons uniquement pour designer un champ d'objets constitués de directions orthogonales. En dimension deux, on peut également utiliser la dénomination *champs de croix* (Figure 2.9). L'intérêt des champs de directions est de permettre de représenter les propriétés d'orientations des quadrilatères en dimension deux et des hexaèdres en dimension trois. Par exemple, le cube est invariant par toutes les combinaisons de rotations de 90 degrés. Ces nouvelles approches ont initialement été développées par la communauté d'informatique graphique pour le maillage quadrilatéral [Ray+08 ; BZK09] et hexaédrique (ou hex-dominant) [NRP11].

Généralement, les méthodes de maillage basées sur les champs de directions fonctionnent en deux temps : (1) génération d'un champ de directions volumique à l'intérieur du modèle et (2) génération d'hexaèdres, guidée par le champ de directions.

Ces deux problèmes peuvent être traités indépendamment, ce que nous allons faire dans la suite de l'exposé en détaillant les différentes représentations et techniques de génération des champs de directions (§2.1.2.1), puis en montrant comment ceux-ci sont utilisés pour générer des maillages (§2.1.2.2). Le lecteur particulièrement intéressé par les champs de directions le contexte du maillage peut également se référer à thèse de N. Kowalski [Kow13] qui se concentre sur ce sujet.

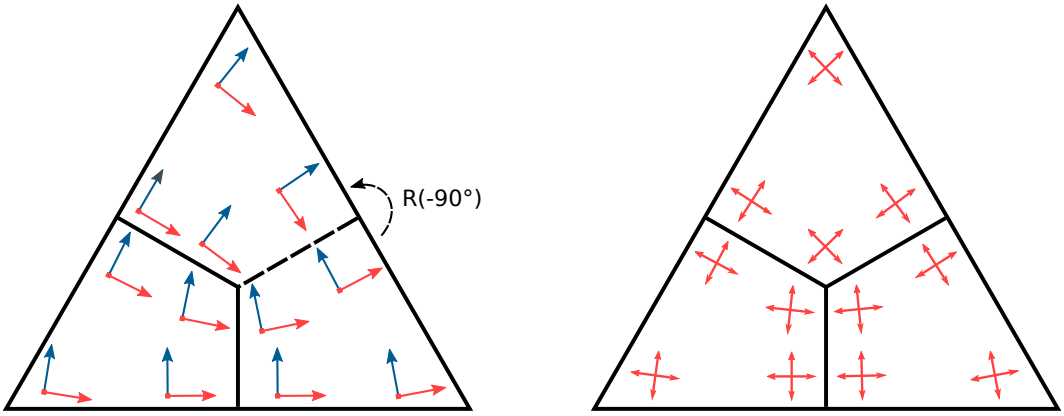


FIGURE 2.9 – Maillage quadrilatéral d'un triangle et représentation de l'orientation des éléments. Gauche : deux champs vectoriels orthogonaux, discontinuité nécessaire (sur l'arête en pointillés). Droite : champ de croix avec transitions continues sur les arêtes, mais présence d'une singularité sur le point central.

2.1.2.1 Champs de directions

En maillage, les champs de directions sont utilisés car ils encodent l'orientation des éléments. On pourrait penser qu'il suffit d'utiliser des champs vectoriels : l'orientation d'un carré est peut être prescrite par deux directions orthogonales. Cependant, arrive fréquemment que les champs vectoriels échangent de rôle, ce qui correspond à des discontinuités dans les champs vectoriels associés. On peut observer ce phénomène sur l'exemple simple de la Figure 2.9. En dimension deux, l'orientation des carrés est invariante par quatre rotations (de $0, 90, 180$ et 270 degrés) et en dimension trois, l'orientation des cubes est invariante par 24 rotations (combinaisons de rotations de 90 degrés selon les trois axes). Comme l'objectif est de construire un champ d'*orientations* afin de guider les algorithmes de maillage, il faut définir un problème dont la solution est ce champ d'*orientations*. Utiliser directement des champs vectoriels orthogonaux discontinus pour définir le problème est particulièrement compliqué car toutes les opérations mathématiques doivent être définies *à toute rotation de* 90 *degrés près*, ce qui conduit à des problèmes mixtes réels-entiers nécessitant des stratégies de résolution très spécifiques, *par ex.* approche multi-résolutions [Gao+17b]. Une alternative est d'utiliser des objets mathématiques qui possèdent intrinsèquement ces invariances : les champs de directions.

Définitions Nous reprenons les définitions de champs de directions de KOWALSKI [Kow13] :

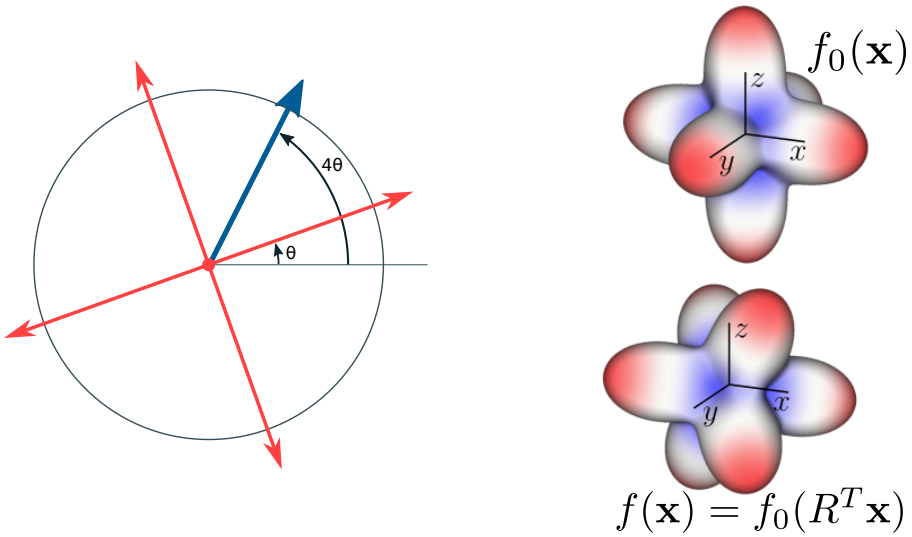


FIGURE 2.10 – *Gauche :* représentation d'une croix (rouge) par un vecteur de représentation $\mathbf{f} = (\cos(4\theta), \sin(4\theta))$ (bleu). *Droite :* représentation de la frame de référence F_0 et d'une frame quelconque F par des fonctions harmoniques sphériques, f_0 et f, dont l'amplitude sur la sphère est représentée en relief et en couleur (source : [RS15]).

— en dimension deux, une *croix* $\mathbf{F}$ est définie par un ensemble de quatre vecteurs $\{\mathbf{u}, -\mathbf{u}, \mathbf{v}, -\mathbf{v}\}$ unitaires et orthogonaux :

$$\mathbf{F} \iff \{\mathbf{u}, -\mathbf{u}, \mathbf{v}, -\mathbf{v} \in \mathbb{R}^2\} \text{ tels que } \|\mathbf{u}\| = \|\mathbf{v}\| = 1 \text{ et } \mathbf{u} \cdot \mathbf{v} = 0$$

— en dimension trois, une *frame* $\mathbf{F}$ est définie par six vecteurs $\{\mathbf{u}, -\mathbf{u}, \mathbf{v}, -\mathbf{v}, \mathbf{w}, -\mathbf{w}\}$ unitaires et orthogonaux :

$$\mathbf{F} \iff \{\mathbf{u}, -\mathbf{u}, \mathbf{v}, -\mathbf{v}, \mathbf{w}, -\mathbf{w} \in \mathbb{R}^3\} \text{ tels que } \|\mathbf{u}\| = \|\mathbf{v}\| = \|\mathbf{w}\| = 1,\ \mathbf{u} \cdot \mathbf{v} = 0 \text{ et } \mathbf{w} = \mathbf{u} \wedge \mathbf{v}$$

Représentation des champs de croix (2D)

Même si les définitions ci-dessus sont invariantes par rotations de 90 degrés, elles utilisent toujours des champs vectoriels. Il est important de trouver une représentation *pratique* de ces objets. Comme souligné par KOWALSKI [Kow13], pour construire des problèmes générant des champs de directions pertinents (*par ex.* le plus lisse possible), il serait intéressant que la représentation utilisée permette facilement de (a) calculer une distance entre deux *croix* afin de définir une énergie et (b) réaliser des interpolations, ce qui permet de discrétiser le champ sur un maillage.

En dimension deux, une représentation intuitive d'une croix est l'angle θ entre l'une des directions et l'axe des abscisses (gauche de la Figure 2.10). Cet angle peut également être vu comme l'angle de la rotation qui transforme le repère unité vers la croix en question. Toutes les croix peuvent être représentées par cet angle, c'est-à-dire que l'espace des croix est de dimension un.

Malheureusement cette représentation n'est pas unique : θ et $\theta + k\frac{\pi}{2},\ k \in \mathbb{N}$ représentent la même croix. Pour avoir une représentation unique, on peut restreindre

l'angle θ à l'intervalle $[0, \frac{\pi}{2}[$. Cependant, cette représentation ne permet pas de calculer facilement (sans modulo) des distances ou d'effectuer des interpolations : les deux croix $\theta_1 = 0$ et $\theta_2 = 89°$ sont très proches mais la norme euclidienne et l'interpolation d'angle ne peuvent pas être utilisées directement. Une alternative intéressante [PZ07] est d'utiliser une représentation à deux composantes exploitant la périodicité des fonctions sinusoïdales : $\mathbf{f} = (\cos(4\theta),\ \sin(4\theta))$. Le vecteur unité $\mathbf{f}$ parcourt le cercle unité lorsque θ parcourt $[0, \frac{\pi}{2}[$ (gauche de la Figure 2.10). Avec cette représentation, appelée **vecteur de représentation** ou *representation-vector* en anglais, nous retrouvons bien l'intuition que deux croix proches ont deux représentations (*c.à-d.* vecteurs) proches. Pour calculer une distance ou effectuer une interpolation, on peut le faire le long du cercle, ou bien utiliser les opérations euclidiennes sur les vecteurs de représentation $\mathbf{f}$. Ceci dit, l'interpolation entre deux vecteurs unitaires (non parallèles) ne produit pas un vecteur unitaire et il faudra projeter le vecteur interpolé sur le cercle unité pour représenter à nouveau une croix.

Les vecteurs de représentation sont très utiles en pratique car (a) ils permettent de définir facilement une énergie quadratique pour minimiser les variations du champ de croix à partir de la distance euclidienne et (b) le champ de vecteurs de représentation peut être discrétisé sur un maillage comme tout champ vectoriel continu.

Représentation des champs de directions (3D)

En dimension trois, les *frames* peuvent être représentées par une matrice de rotation $R \in SO(3)$. Formellement, l'espace des rotations $SO(3)$ est le groupe des matrices ($R \in \mathbb{R}^{3\times3}$) orthogonales de déterminant 1. Comme les *frames* sont invariantes par les 24 permutations d'axes du repère, en notant $A = \{\pm\mathbf{e}_x, \pm\mathbf{e}_y, \pm\mathbf{e}_z\}$, il est intéressant d'introduire le groupe des symétries cubiques (ou de permutations d'axes) :

$$O = \{R_0 \in SO(3) \mid R_0\mathbf{x} \in A,\ \forall \mathbf{x} \in A\}$$

À chaque *frame* correspond 24 matrices de rotation, et si l'on connait une des matrices, il est possible d'obtenir toutes les autres en la multipliant par les matrices de permutation $R_0 \in O$.

Une matrice 3D est constituée de neuf coefficients, ce qui est un nombre élevé de paramètres. Mais les matrices de $SO(3)$ peuvent être exprimées en n'utilisant que trois paramètres, les **angles d'Euler** α, β, γ, qui caractérisent les rotations successives autour de chaque axe du repère :

$$R = R_x(\alpha)R_y(\beta)R_z(\gamma)$$

Cependant, la représentation des matrices de rotation par angles d'Euler souffre du problème du *blocage de cadran* : dans certaines situations (*par ex.* $\beta = 0$), certaines variations de la rotation ne peuvent pas être représentées par des variations de α, γ. Toutes les paramétrisations à trois variables de $SO(3)$ ont des problèmes similaires [Stu64] et pour s'en affranchir, il est nécessaire d'utiliser plus de variables. Nous évoquons deux autres représentations utilisées dans le cadre des champs de directions :

— Kowalski [Kow13] introduit la représentation sous formes de **quaternions** (4 variables). L'utilisation de quaternions est un choix courant en informatique graphique car toute matrice de rotation peut être représentée par un quaternion unitaire de type :

$$\mathbf{q} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} + \mathbf{z},\ (x, y, z, w) \in \mathbb{R}^4 \text{ avec } w^2 + x^2 + y^2 + z^2 = 1$$

Pour être plus précis, il existe un revêtement double des quaternions vers les matrices de rotation : chaque matrice de rotation peut être représentée par un quaternion $\mathbf{q}$ et son opposé $-\mathbf{q}$, et une petite variation de la matrice de rotation est équivalente à une petite variation des deux quaternions. Comme il y a 24 matrices de rotation possibles pour une frame, cela implique qu'il y a 48 quaternions unitaires pour représenter une même frame. Afin de pouvoir calculer des distances et des interpolations, Kowalski [Kow13] compare le quaternion-rotation $\mathbf{q}_i\mathbf{q}_j^{-1}$, d'un quaternion $\mathbf{q}_i$ à un second $\mathbf{q}_j$, aux 48 quaternions $\mathbf{g}_k$ qui représentent les 24 permutations $R_0 \in O$. Si $\mathbf{q}_i\mathbf{q}_j^{-1}$ est proche d'une des permutations R_0, alors cela signifie que les deux frames associées à $\mathbf{q}_i$ et $\mathbf{q}_j$ sont proches. Formellement, la distance est définie par :

$$d_{ij} = \min_{\mathbf{g}_k}(1 - |\mathbf{g}_k(\mathbf{q}_i\mathbf{q}_j^{-1})|)$$

De la même manière pour l'interpolation, il faut d'aborder transformer l'un des quaternions avec la permutation $\mathbf{g}_k$ qui le rapproche le plus de l'autre, ensuite effectuer une interpolation classique des coefficients et finalement renormaliser le quaternion interpolé (car celui-ci n'est pas unitaire).

— Huang et al. [Hua+11] introduit une représentation des *frames* par des fonctions polynomiales : les harmoniques sphériques de degré quatre. Celles-ci sont des polynômes à 9 coefficients qui contiennent les mêmes invariances que les *frames* et qui permettent de les représenter de manière *unique*. La base des harmoniques sphériques de degré 4 est notée $(Y_4^l)_{l=-4..4}$ (formules explicites à [GB96, p. 236]). En reprenant les explications et notations de Ray et al. [RS15] et Solomon et al. [SVB17], on part de la fonction f_0 qui représente la frame alignée avec les axes du repère (Figure 2.10) :

$$f_0(\mathbf{x}) = \sqrt{\frac{7}{12}}Y_4^0(\mathbf{x}) + \sqrt{\frac{5}{12}}Y_4^4(\mathbf{x})$$

Pour chaque frame, on note $\mathbf{a}$ les coefficients dans la base sphérique harmonique, ce qui donne pour la frame de référence :

$$\mathbf{a}_0 = (0, 0, 0, 0, \sqrt{\frac{7}{12}}, 0, 0, 0, \sqrt{\frac{5}{12}})$$

Une frame quelconque F, obtenue par rotation R de la frame de référence, peut être représentée par la fonction $f(\mathbf{x}) = f_0(R^T\mathbf{x})$. Ces deux représentations fonctionnelles sont illustrées sur la droite de la Figure 2.10. Les coefficients $\mathbf{a} \in \mathbb{R}^9$ de f peuvent être obtenus par la multiplication par une matrice $W_4(R) \in \mathbb{R}^{9x9}$, dite matrice de Wigner-D de degré 4 associée à la rotation R [AA87], des coefficients $\mathbf{a}_0$ de la frame de référence. On peut ainsi caractériser l'espace des coefficients $\mathbf{a}$ représentant les *frames* de manière unique [SVB17] :

$$\Gamma = \{\mathbf{a} \in \mathbb{R}^9,\ \mathbf{a} = W_4(R)\mathbf{a}_0,\ R \in SO(3)\} \tag{2.1}$$

La contrainte sur les coefficients pour avoir $\mathbf{a} \in \Gamma$ est compliquée car il ne suffit pas d'un critère de type $\|\mathbf{a}\| = 1$ comme pour les vecteurs de représentation ou les quaternions. Pour s'en accommoder, Huang et al. [Hua+11] utilise les

coefficients $\mathbf{a}$ uniquement pour calculer une énergie, et leur solveur (non-linéaire) optimise par rapport aux angles d'Euler via les matrices de Wigner-D, tandis que Ray et al. [RS15] propose un algorithme itératif pour projeter un vecteur $\mathbf{v} \in \mathbb{R}^9$ vers le plus proche vecteur $\mathbf{a} \in \Gamma$ représentant une frame.

En résumé : il existe une paramétrisation simple et unique des croix par les vecteurs de représentation, mais il n'y a pas d'équivalent en dimension trois. Pour représenter les *frames* (trois degrés de liberté), on peut utiliser au choix : (1) une paramétrisation à trois variables de $SO(3)$, les angles d'Euler, qui contient des discontinuités et singularités, ou (2) une paramétrisation lisse à 4 variables contraintes (toujours de $SO(3)$), les quaternions unitaires, qui contient beaucoup de redondances (48 quaternions par frame), ou (3) une paramétrisation fonctionnelle de $SO(3)/O$ (l'espace des frames) par les harmoniques sphériques, à 9 variables contraintes, qui a l'avantage d'être unique mais dont les contraintes sont compliquées.

Caractérisation des singularités

De la même manière que les champs vectoriels, les champs de directions peuvent contenir des singularités, c'est-à-dire des parties du domaine où les champs de directions ne sont pas définis. Ce phénomène arrive fréquemment lorsqu'on veut générer des champs de directions continus. L'exemple le plus simple, en dimension deux, est illustré sur la droite de la Figure 2.9. Le champ de croix est bien défini dans chaque triangle, et il est continu le long des arêtes entre triangles, mais il n'est pas défini sur le sommet central. Une caractérisation précise des singularités en dimension deux est possible en utilisant l'indice de Poincaré d'un champ vectoriel $\mathbf{v}$ autour d'un point singulier $\mathbf{x}$, qui est donné par la circulation du champ le long d'une courbe fermée entourant $\mathbf{x}$:

$$\text{index}(\mathbf{x}) = \frac{1}{2\pi} \oint d\theta \text{ avec } \theta = \arctan(\frac{v_y}{v_x})$$

En appliquant cette formule au vecteur de représentation d'un champ de croix, on peut calculer l'indice des singularités. En pratique, les singularités sont d'indices $+1$ (la croix a subi une rotation de $+90°$ le long du cycle) ou d'indices -1 (rotation de $-90°$). Si on mesure la rotation θ de la croix, au lieu de 4θ la rotation du vecteur de représentation, on note les indices sous formes de fractions : $\pm\frac{1}{4}$. Le théorème de Poincaré-Hopf lie la somme des indices de Poincaré du champ vectoriel 2D à la caractéristique d'Euler du domaine. Ce résultat peut être généralisé aux champs de croix [Ray+08 ; Fog+17].

La situation est plus compliquée en dimension trois et, à notre connaissance, il n'existe pas de classification précise des singularités des champs de directions. Ceci dit, on sait que les singularités forment un graphe, composé de courbes [NRP11], qui peuvent se rejoindre à l'intérieur du domaine. Il est possible de les détecter en suivant la variation continue de la matrice de rotation, associée à la *frame*, le long d'une courbe fermée [SVB17]. Comme le champ de directions est supposé continu, la matrice de rotation R_0 obtenue après le parcours de la courbe contractée [16] appartiendra à l'ensemble des permutations d'axes O. Si c'est l'identité, alors il n'y a pas

16. si la courbe est trop grande, elle peut contenir plusieurs singularités qui s'annulent, en pratique il faut utiliser une courbe de la taille minimale possible avec la discrétisation choisie

de ligne singulière à l'intérieur de la courbe, et si R_0 est une permutation des axes, alors il y a une singularité.

Il est important de remarquer que les singularités des maillages quadrilatéraux et hexaédriques, *c.à-d.* valence des sommets différente de 4 et des arêtes différente de 4, correspondent à des singularités du champ de directions associé. Cependant l'inverse n'est pas toujours vrai, certaines singularités des champs de directions sont plus complexes et ne peuvent pas être maillées [Li+12 ; VSL16].

Discrétisation des champs de directions

Une fois qu'une représentation est choisie pour les champs de directions, il faut la discrétiser sur un maillage. De nombreux choix sont possibles, chacun affectant notamment la manière dont apparaissent les singularités :

— une frame constante par tétraèdre [NRP11]. Les singularités sont des arêtes du maillage. Cette discrétisation est probablement la plus courante ;

— une frame par sommet [KLF14 ; RSL16 ; Gao+17b]. Les singularités sont volumiques (ensemble de tétraèdres) ;

— une frame par face de tétraèdre [Hua+11], linéaire à l'intérieur (éléments finis de Crouzeix-Raviart). Les singularités sont des lignes (ensemble d'arêtes) ;

— une frame par triangle du bord [SVB17], le champ de directions volumique est défini par interpolation via la méthode des éléments frontières. Les singularités ne sont pas accessibles directement mais peuvent être reconstruites par échantillonnage aléatoire dans le volume avec des triangles.

La discrétisation constante par tétraèdres est particulièrement pratique pour la gestion du graphe de singularités, qui est représenté par des arêtes dans le maillage. Cependant certaines singularités peuvent être non maillables, c'est pourquoi il est préférable de restreindre les singularités possibles [Li+12]. Les discrétisations avec interpolation linéaire du champ de directions permettent une plus grande richesse d'approximation et les algorithmes itératifs d'optimisation de champs convergent beaucoup plus vite [RS15], cependant la gestion des singularités est plus difficile : il faut extraire le graphe du volume. De plus, avoir des singularités qui ne sont pas des composantes du maillage (arêtes) peut compliquer les phases ultérieures de maillage (*par ex.* paramétrisation par intégration du champ de croix).

Remarque. Le cas des surfaces immergées dans un espace 3D introduit de nouvelles difficultés car il faut prendre en compte la courbure de la surface. Le lecteur intéressé peut se référer à état de l'art récent [Vax+16], qui s'intéresse aux différentes techniques de construction et de manipulation des champs N-directionnels sur les surfaces, dont les champs de croix sont un cas particulier ($N = 4$).

Génération de champs de directions

L'objectif est de construire un champ de directions lisse, c'est-à-dire avec des variations locales faibles des *frames* (*c.à-d.* maillage hexaédrique localement structuré), tout en respectant les orientations prescrites par le bord du domaine. Plus on se rapproche d'une singularité, plus les *frames* varient fortement et l'énergie associée à la variation tend vers l'infini. Or dans la plupart des configurations géométriques, nous avons vu qu'il est nécessaire d'introduire des singularités dans le maillage hexaédrique. C'est là toute la difficulté de la génération de champs de directions : on

souhaite un champ de directions lisse, avec un certain nombre de singularités (*c.à-d.* pas du tout lisse localement), mais pas trop.

Dans la littérature, il existe deux types d'approches pour la génération de champs de directions lisses en dimension trois : (1) génération d'un champ de croix sur le bord puis propagation frontale vers l'intérieur tout en minimisant les variations des rotations et (2) minimisation d'une énergie caractérisant la variation du champ.

Nous commençons par évoquer les contraintes à respecter sur le bord, qui sont similaires dans les deux approches, et nous décrivons ensuite deux approches de génération du champ de directions volumique.

Contraintes. Le but des champs de directions est de permettre la génération de maillages dont les éléments sont alignés avec le bord, ce qui se traduit par des directions imposées sur les *frames* du bord. En dimension trois, il y a deux possibilités :

— sur un bord plat, une direction est fixée : celle qui correspond à la normale du bord. Les deux autres directions sont libres, la *frame* peut tourner autour de la normale à la surface ;

— le long d'une arête vive, les trois directions sont fixées : deux directions sont fixées aux normales de part et d'autre de l'arête, et la troisième direction n'est plus libre car les trois directions sont forcément orthogonales.

Le cas d'un coin du domaine correspond à trois arêtes vives qui se rejoignent, et dont les contraintes sur les directions sont compatibles. En pratique, les arêtes vives ne forment pas que des angles droits mais couvrent toute l'amplitude des variations de normales possible. Il est donc nécessaire d'utiliser certains critères géométriques pour distinguer une variation lisse, associée à la même direction de *frame*, d'une arête vive qui mettra en jeu deux directions orthogonales. Le critère le plus simple est un seuil sur l'angle diédral (*par ex.* $\alpha > 45°$) mais il faut être prudent car cela peut introduire des effets inattendus sur certains modèles.

Champ de directions par minimisation d'énergie. Dans le cas continu, en introduisant une transformation $\tau : \Omega \to SO(3)/O$, on peut décrire la régularité (*smoothness* en anglais) d'un champ de directions sans singularité par une énergie [SVB17] :

$$E[\tau] = \int_\Omega \|d\tau\|_2^2 \, dV$$

Il y a deux difficultés quasiment insurmontables avec cette énergie du problème continu : (a) elle n'est pas définie lorsqu'il y a des singularités car $\|d\tau\|_2^2 \to \infty$ autour de celles-ci et (b) l'espace $\tau : \Omega \to SO(3)/O$ n'est pas du tout linéaire et il n'est pas possible de se *déplacer* facilement sur cette variété, ce qui est plus ou moins nécessaire pour envisager l'usage d'un solveur numérique. En pratique, ces difficultés sont évitées en (a) travaillant directement avec une énergie discrète, d'un champ de directions discrétisé avec une énergie finie, et (b) en relâchant la contrainte d'appartenance à $SO(3)/O$.

En utilisant la représentation par harmoniques sphériques (*c.à-d.* vecteur $\mathbf{a}$ à 9 coefficients) et en discrétisant avec une *frame* par sommet, Ray et al. [RSL16] propose de définir l'énergie discrète en sommant les différences sur chaque arête ij du maillage tétraédrique :

$$E = \sum_{ij} \|\mathbf{a}_i - \mathbf{a}_j\|_2^2$$

Pour que le résultat soit un champ de directions, il faut $\mathbf{a} \in \Gamma$ (définie par (2.1)) et il faut également des contraintes sur les coefficients pour assurer l'alignement avec le bord. Le problème de minimisation de E est ensuite résolu en deux étapes :

1. Initialisation : l'énergie est minimisée avec un solveur moindre carrés sans utiliser la contrainte $\mathbf{a} \in \Gamma$, puis chaque *frame* $\mathbf{a}$ de la solution est projetée sur Γ avec un algorithme non linéaire. Une autre initialisation simple consiste à copier les *frames* du bord vers l'intérieur de manière frontale [Li+12] ;
2. Lissage : le champ de directions est lissé de manière successive avec un solveur itératif de type L-BFGS (enchainement de linéarisation et projection sur Γ). Le problème de minimisation d'énergie est fortement non-convexe et possède de nombreux minima locaux, d'où l'importance capitale de l'étape d'initialisation.

Le lecteur intéressé peut également se référer à l'article initial de Huang et al. [Hua+11], dont [RSL16] est une amélioration, et à Solomon et al. [SVB17] qui proposent de résoudre le problème de génération du champs avec la méthode des éléments frontières (*c.à-d.* sans discrétisation volumique) et en ajustant certaines contraintes supplémentaires sur le bord, ce qui permet de mieux gérer les géométries sphériques (moins de singularités). On peut également citer Li et al. [Li+12] qui restreignent les singularités du champ de directions à 10 configurations, au lieu de 24, de manière à n'accepter que celles qui ont une équivalence dans un maillage hexaédrique et Jiang et al. [Jia+14] qui proposent des techniques de modifications des singularités de manière à éliminer certaines configurations non-intégrables.

Champ de directions par propagation depuis le bord. Lorsque le champ de directions est représenté par des quaternions, il n'est pas possible de construire le champ directement par une minimisation d'énergie quadratique, car la distance entre deux frames ne peut pas être approximée par la distance euclidienne des coefficients. Comme pour chaque *frame*, il y a 48 quaternions possibles, il faudrait ajouter des variables pour chaque choix de permutation, ce qui compliquerait considérablement le problème.

Pour contourner cette difficulté, Kowalski [Kow13] et Kowalski et al. [KLF16] proposent de construire un premier champ de directions avec un algorithme basé sur la propagation d'un front partant des bords. La seconde étape consiste à lisser le champ de directions avec un algorithme itératif emprunté à Li et al. [Li+12], similaire à l'approche de lissage des harmoniques sphériques présentée précédemment, si ce n'est que celle-ci utilise les matrices de rotation comme représentation.

L'algorithme de propagation fonctionne de la manière suivante :

— on part d'un certain nombre de frames déjà assignées (les bords initialement)
— pour chaque sommet sans frame assignée, appartenant à un tétraèdre dont les autres sommets ont une frame assignée, une nouvelle frame candidate calculée par interpolation est assignée
— dans la liste globale des candidats, on conserve celui qui minimise une mesure de la variation du champ de frames actuel (frames déjà assignées et candidat courant)
— on recommence tant que tous les sommets ne sont pas assignés

L'ordre d'ajout des nouvelles *frames* à un impact très fort sur la position des singularités, c'est pourquoi l'étape du choix du candidat minimisant la variation du champ

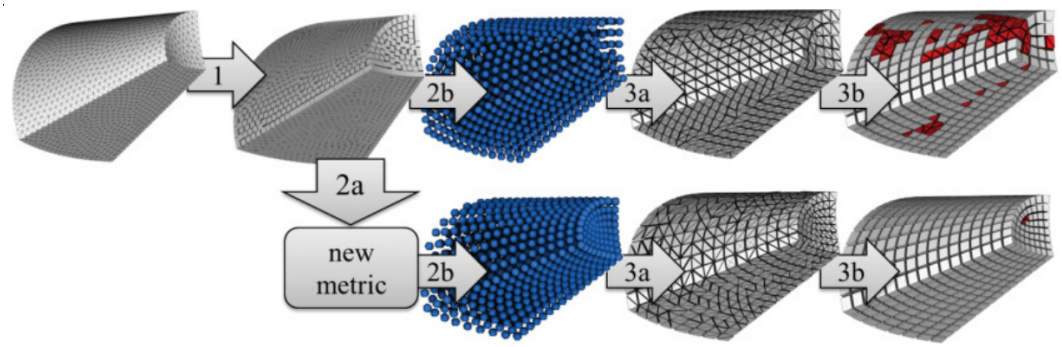


FIGURE 2.11 – Étapes successives du maillage hex-dominant indirect. (1) Génération du champ de directions. (2) Placement de points. (3a) Tétraédrisation des points. (3b) Recombinaison des tétraèdres en hexaèdres. Source : [Sok+16].

est importante car elle permet un ordre d'insertion moins arbitraire.

Le résultat de cette étape d'initialisation est un champ très *rigide* avec de fortes variations là où les fronts se sont rejoints. L'étape de lissage permet de répartir la variation du champ de manière plus homogène dans le volume, tout en conservant la même topologie du graphe de singularités.

Remarque. Au lieu d'essayer de générer le champ de directions de manière automatique, il est possible de le construire plus ou moins manuellement. C'est notamment l'approche suivie dans le premier article de maillage hexaédrique utilisant des champs des directions [NRP11], où les champs étaient construits à partir de *metamesh*, *c.à-d.* des maillages hexaédriques très grossiers englobant le domaine.

2.1.2.2 Génération d'hexaèdres à partir de champs de directions

Le champ de directions sert de guide pour l'étape de maillage hexaédrique. Différentes approches, classées en deux catégories, sont possibles : approches directes où l'on construit directement des hexaèdres et les approches indirectes où l'on commence par placer des points, permettant de générer un maillage tétraédrique, duquel sont formés des hexaèdres par recombinaison.

Approches indirectes

Construire directement des hexaèdres à partir d'un champ de directions n'est pas aisé car il faut prendre en compte les voisinages. L'approche indirecte, introduite en trois dimensions par BAUDOUIN et al. [Bau+14], propose de commencer par placer des points de manière régulière en se servant du champ de directions. Avec ces points, on peut construire un maillage tétraédrique par triangulation de Delaunay contrainte [GHS90]. La dernière étape consiste à combiner les tétraèdres voisins en hexaèdres. Les différentes étapes sont illustrées sur la Figure 2.11.

Placement des points. Afin d'espérer obtenir une proportion maximale d'hexaèdres, de bonnes qualités si possible, il est nécessaire de placer les sommets de manière régulière, ce qui est réalisé grâce à l'utilisation de champs de directions. BAUDOUIN

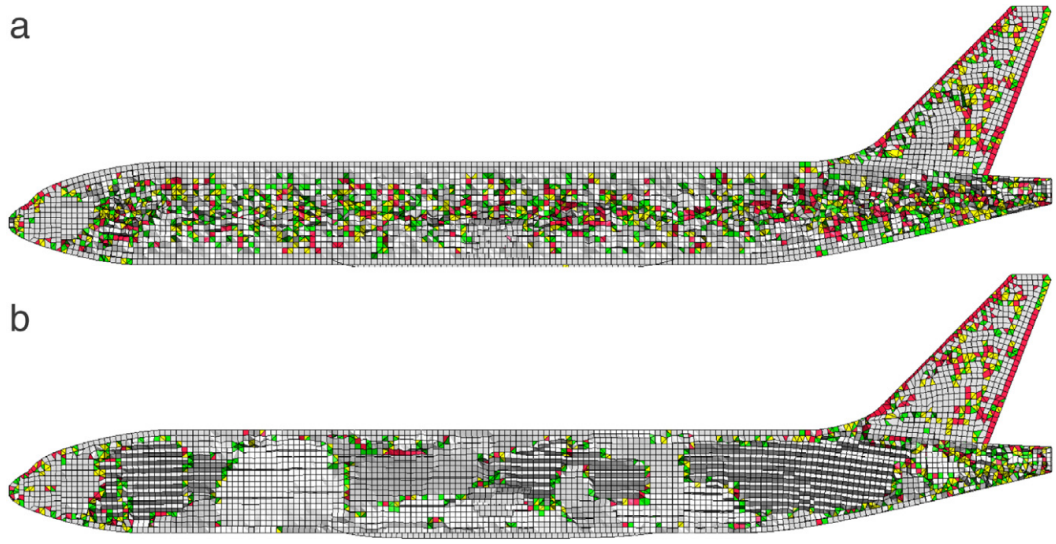


FIGURE 2.12 – Exemple de maillages hex-dominants construits par approche indirecte frontale. (a) : placement des points de manière purement frontale. (b) : les points sont placés prioritairement là où le champ de directions est lisse. Source : [Ber+16].

et al. [Bau+14] utilisent une approche frontale : de manière itérative, des points sont insérés avec des algorithmes *d'empaquetage*, où à chaque point est associée une boite (dont l'orientation est donnée par le champ de directions), et les différentes boites ne doivent pas contenir les points voisins. Une amélioration possible [Ber+16] consiste à insérer prioritairement des points dans les régions où le champ de directions est le plus lisse, plutôt que d'ajouter des points couche par couche en partant du bord. Les deux approches sont illustrées sur la Figure 2.12. Une autre manière de placer les points est d'utiliser une paramétrisation globale [Sok+16], issue de la résolution d'un problème mixte réels-entiers, où les points sont placés à l'intersection d'iso-valeurs de trois champs scalaires u, v, w. Nous évoquerons plus en détail la construction des paramétrisations dans la description des approches directes.

Combinaison des tétraèdres en hexaèdres. Une fois un maillage tétraédrique généré à partir des points judicieusement placés, il faut former des hexaèdres à partir des tétraèdres. Pour ce faire, BAUDOUIN et al. [Bau+14] utilisent l'algorithme de Yamakawa-Shimada [YS03] qui assemble des ensembles de 5, 6 ou 7 tétraèdres adjacents en un hexaèdre à partir de patrons combinatoires prédéfinis. Seuls les hexaèdres respectant des critères géométriques de qualité sont conservés. Toujours en utilisant des patrons prédéfinis, BOTELLA et al. [BLC14] et SOKOLOV et al. [Sok+16] proposent d'autres algorithmes de recombinaison couvrant un ensemble plus vaste de configurations en exploitant le formalisme sous forme de graphes introduit par MESHKAT et al. [MT00]. Plus récemment, une autre technique plus efficace et prenant en compte des combinaisons à plus de sept tétraèdres a été proposée par PELLERIN et al. [PJR17].

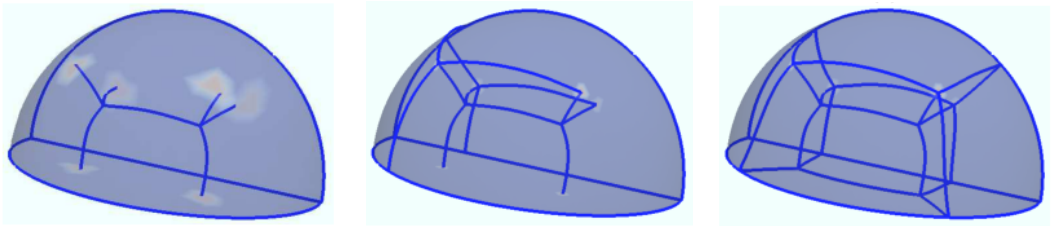


FIGURE 2.13 – Exemple de décomposition en blocs à partir du graphe de singularités du champ de directions. Gauche : point de départ (graphe de singularités). Milieu : tracé des séparatrices sur la surface en cours. Droite : décomposition en blocs finale. Source : [Kow13].

Les approches indirectes permettent de générer des maillages hex-dominants avec une grande proportion d'hexaèdres, mais le placement itératif de points semble toujours produire des régions non structurées, là où les points ont été ajoutés en dernier et autour des singularités du champ de directions, où il est très difficile de produire des hexaèdres.

Une approche alternative [Gao+17b], que l'on peut qualifier d'indirecte car reposant sur la modification d'un maillage tétraédrique, consiste à construire des maillages hex-dominants en déplaçant les sommets du maillage tétraédrique initial et en modifiant itérativement la topologie des cellules avec des opérations locales de fusion et de subdivision. Plus précisément, un champ de directions, construit à partir d'une méthode multigrille, est utilisé pour générer un champ de *positionnement* (inspiré de [Jak+15], semblable à une paramétrisation locale). Les sommets du maillage tétraédrique sont assignés à des positions du champ (sommets souhaités du maillage hexaédrique). Les arêtes et faces des tétraèdres contenant des sommets qui sont déplacés vers la même position finale sont progressivement supprimées à partir d'heuristiques, en maintenant un maillage valide au fur et à mesure des opérations. Le maillage final est composé d'hexaèdres et de polyèdres quelconques, qui peuvent éventuellement être tétraédrisés.

Approches directes

Les approches directes se passent d'un maillage intermédiaire et essaient de construire directement un maillage hexaédrique. Nous évoquons deux sous-catégories : celles où l'on construit une décomposition en blocs hexaédriques du domaine, chaque bloc étant maillé par transformation dans un second temps et celles où l'on construit une paramétrisation du domaine, de laquelle les hexaèdres sont extraits.

Décomposition du domaine en blocs. Il s'agit d'une approche que l'on pourrait qualifier de semi-directe dans le sens où elle consiste surtout à produire des blocs cubiques. L'étape suivante de remplissage des blocs fait appel aux techniques classiques de maillage structuré par blocs (*par ex.* sous-transformations). La première étape est de construire un ensemble de lignes fermées du modèle, telles que les lignes puissent être groupées en patchs quadrilatérales. KOWALSKI et al. [KLF14] partent du graphe de singularités du champ de directions, et des arêtes vives du bord, et

trace ensuite des lignes séparatrices[17] sur le champ de croix surfacique de manière à relier les extrémités des singularités connectées au bord. Ce procédé est illustré sur la Figure 2.13. Comme Kowalski [Kow13] utilise une discrétisation avec une *frame* par sommet, le graphe de singularités est initialement volumique (ensemble de tétraèdres). Le graphe linéique est reconstruit en interpolant le vecteur de représentation des croix, obtenues en sélectionnant les deux directions instables des *frames* le long des singularités linéiques. Là où les singularités se rejoignent (trois directions instables), un point est sélectionné au centre des tétraèdres singuliers. Lors du tracé des courbes séparatrices, celles-ci peuvent ne pas se rejoindre exactement, mais s'éviter de peu. Ceci est contourné en les fusionnant à partir de critères géométriques. Sur certains modèles, l'ensemble de lignes (singularités et séparatrices) correspond à une décomposition en blocs cubiques (*par ex.* Figure 2.13), et dans ce cas on peut en déduire un maillage hexaédrique. Cependant ce n'est pas toujours le cas, et certains modèles ne peuvent pas être décomposés en blocs cubiques avec cette approche.

Paramétrisations. L'idée des paramétrisations est de construire une transformation du domaine à mailler vers un espace paramétrique, où le domaine transformé peut être maillé par une grille régulière. La transformation inverse de cette grille produit le maillage hexaédrique du domaine, aligné avec les bords. La force de cette approche est que la transformation peut admettre certaines discontinuités, ce qui permet de gérer des singularités. Le processus est illustré sur la Figure 2.14. La transformation est typiquement définie par trois valeurs u_i, v_i, w_i, les coordonnées paramétriques, au niveau de chaque sommet $\mathbf{x}_i$ du maillage tétraédrique (ou pour chaque coin de tétraèdre au niveau des discontinuités). On peut directement visualiser les trois champs scalaires u, v, w associés dans le maillage du modèle initial. Le futur maillage hexaédrique correspond à un ensemble d'iso-valeurs dans ces trois champs, qui peuvent échanger leur rôle le long de certaines faces.

Nieser et al. [NRP11] introduisent la méthode **CuberCover** pour générer une paramétrisation dans $\mathbb{R}^3$ d'un modèle, sans prescrire les bords du domaine paramétrique, à partir d'un champ de directions. Il s'agit d'une généralisation de l'approche *QuadCover* [KNP07] développée en 2D. On cherche trois champs scalaires u, v, w (discontinus) tels que :

— Dans chaque tétraèdre T du maillage Ω_h, les gradients des champs u, v, w soient les plus alignés possible avec les directions $\mathbf{F}_0, \mathbf{F}_1, \mathbf{F}_2$ du champ de directions. Par exemple, on minimise : $\int_T \|\nabla u - \mathbf{F}_1\|_2^2$; de même pour v et w. La correspondance entre le champ et la direction n'a pas d'importance : ∇u pourrait tout aussi bien être aligné avec $\mathbf{F}_0$ ou $\mathbf{F}_2$. Conceptuellement, la paramétrisation u, v, w est l'intégrale (au sens des moindres carrés) du champ de directions ;

— Deux tétraèdres adjacents doivent avoir deux paramétrisations *grille-compatible*, c'est-à-dire qu'il existe une transformation, de type $\mathcal{M} = R\mathbf{x} + \mathbf{t}$, composée d'une rotation $R \in O$ (groupe des 24 permutations d'axes) et une translation entière $\mathbf{t} \in \mathbb{Z}^3$, qui permet de coller les deux tétraèdres sur leur face commune dans l'espace paramétrique. Cette contrainte sur les discontinuités dans l'espace paramétrique permet d'assurer la conformité des parties de maillage hexaédrique extraites des deux tétraèdres ;

17. les séparatrices sont des lignes, partant des singularités, tangentes au champ de directions.

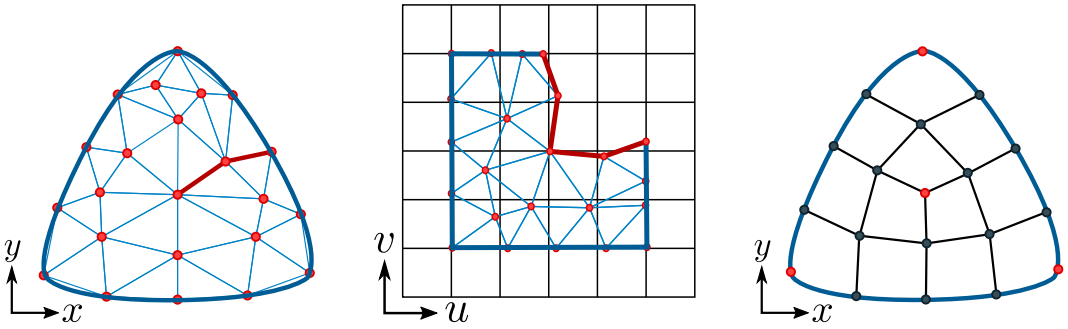


FIGURE 2.14 – Principe du maillage quadrangulaire par paramétrisation. *Gauche :* maillage triangulaire du domaine. *Milieu :* visualisation de l'objet dans le domaine paramétrique, avec discontinuité de la paramétrisation sur le trait rouge. Les arêtes du bord de l'objet ont des coordonnées entières et le saut associé à la discontinuité est *grille-compatible*. *Droite :* maillage quadrangulaire obtenu par transformation de la grille régulière vers le domaine initial, ou de manière équivalente par extraction des iso-valeurs entières dans les coordonnées u_i, v_i associées aux sommets du maillage triangulaire.

— Sur le bord du modèle $\partial\Omega_h$, les paramétrisations u, v, w ont des valeurs entières, ce qui correspondra à des faces d'hexaèdres ;

— Les paramétrisations u, v, w ont des valeurs entières le long des singularités, sauf la direction stable le long de la ligne de singularité qui peut conserver une valeur réelle. Ceci assure que les singularités du champ de directions seront représentées par des arêtes du maillage hexaédrique.

Cette formulation du problème introduit a priori trois types d'inconnus : les valeurs aux coins des tétraèdres (u_i, v_i, w_i), les rotations R aux interfaces entre tétraèdres et les translations entières $\mathbf{t}$. On peut déduire les rotations entre tétraèdres adjacents à partir du champ de directions, mais les translations entières sont de vraies inconnues du problème, qui est maintenant un problème mixte réél-entier, qui peut être résolu de manière efficace en utilisant un solveur développé pour le problème de quadrangulation [BZK09]. L'approche par paramétrisation de type *CubeCover* permet de produire des maillages entièrement hexaédriques pour un certain nombre de géométries, y compris des configurations compliquées (Figure 2.15, cf [Li+12]). Cependant elle n'est pas infaillible et il reste de nombreux cas d'échec. Par exemple, cette approche ne permet pas de gérer les rampes (Figure 3.7, p. 85), où la surface du bord se sépare en deux.

Avec *CuberCover*, un bloc du domaine paramétrique sera toujours maillé par une grille régulière, or ce bloc cubique peut avoir des faces de tailles très différentes. Techniquement, cela arrive quand la paramétrisation possède une forte divergence. Pour remédier à cette difficulté, SOKOLOV et al. [Sok+16] proposent d'utiliser une **paramétrisation périodique globale**, généralisation du cas 2D [Ray+06], autorisant la création de singularités dans la paramétrisation. Cette approche a des similitudes avec *CubeCover*, à la différence que la contrainte de *grille-compatible* est relâchée partiellement. En conséquence, la paramétrisation obtenue n'est pas valide partout mais elle possède moins de divergence. Les nouvelles singularités ne pourront pas

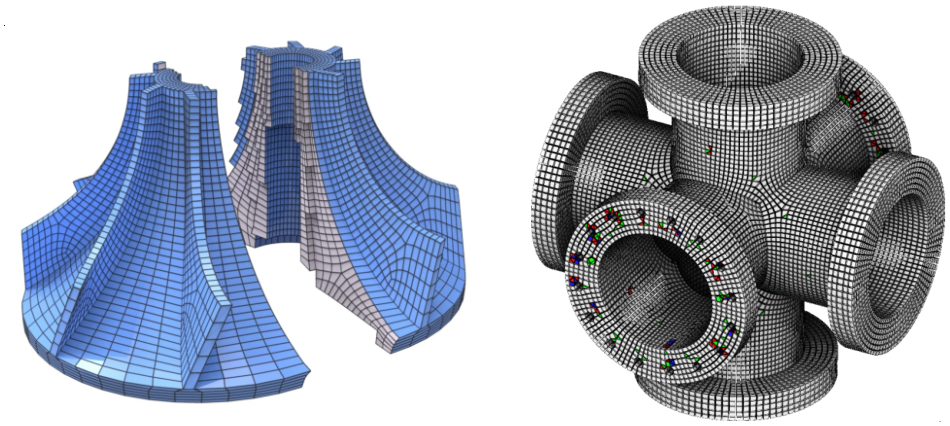


FIGURE 2.15 – *Gauche :* Maillage entièrement hexaédrique généré par *CubeCover* [Li+12]. *Droite :* Maillage hex-dominant généré par *PGP* [Sok+16].

être maillés par des hexaèdres. Cette approche permet de générer des maillages hex-dominants (avec des tétraèdres là où la paramétrisation n'est pas valide) avec des éléments de bonnes qualités (peu déformés). Un résultat sur un modèle compliqué est illustré sur la droite de la Figure 2.15.

Extraction d'hexaèdres à partir d'une paramétrisation. Avec une paramétrisation u, v, w respectant les contraintes de *CubeCover* et telle que le volume de chaque tétraèdre dans le domaine paramétrique est positif (pas de retournement), on peut extraire les hexaèdres à partir des iso-valeurs entières : chaque sommet du maillage hexaédrique correspond à l'intersection de trois iso-valeurs des trois champs scalaires, chaque face correspond à l'iso-valeur d'un champ (aux transformations grille-compatible près) et chaque arête correspond à l'intersection de deux isos.

Cependant, avec les deux approches de paramétrisation citées précédemment, les paramétrisations sont produites en minimisant une énergie, et rien ne garantit que la paramétrisation obtenue est valide. Au contraire, il est courant d'observer des tétraèdres avec un volume négatif dans le domaine paramétrique. EBKE et al. [Ebk+13] et LYON et al. [LBK16] décrivent des techniques robustes d'extraction de quadrilatères et d'hexaèdres qui permettent de rattraper un certain nombre de défauts locaux dans les paramétrisations. Et là où la paramétrisation est localement invalide, on peut laisser des trous qui seront rebouchés avec des tétraèdres dans une étape ultérieure.

Remarque. On peut réinterpréter les techniques de *polycube* avec le formalisme des paramétrisations ci-dessus. La différence fondamentale est que les approches polycubes construisent une paramétrisation continue du domaine. Par conséquent, il n'y a pas de singularités dans le domaine, ce qui correspond à un champ de directions dans lequel les différentes directions n'échangent jamais de rôle. Ceci dit, il existe une variation récente des polycubes [Fan+16] où la paramétrisation peut être discontinue, mais toujours sans rotations. Ceci permet de traiter une gamme plus large de modèles, toujours sans introduire de singularités internes.

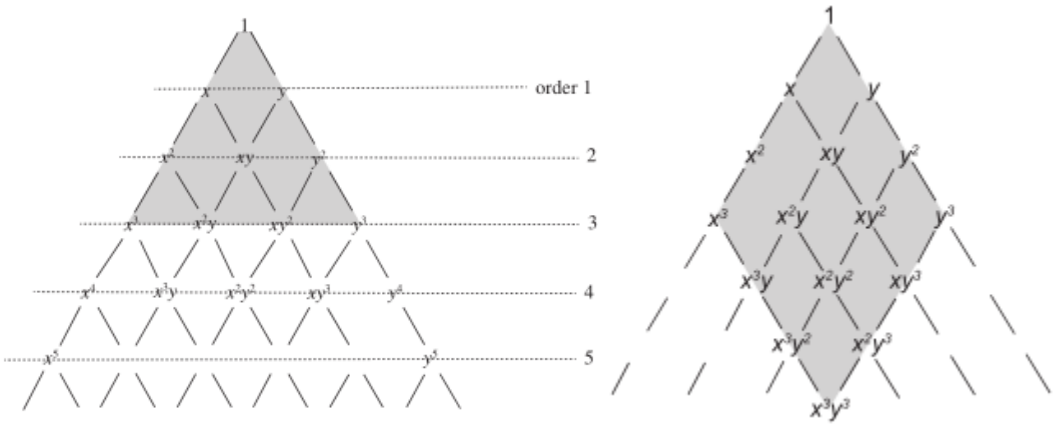


FIGURE 2.16 – Termes polynomiaux à deux variables illustrés sur le triangle de Pascal, pour $\mathbb{P}_3$ (gauche) et $\mathbb{Q}_3$ (droite). Source : adapté de [Zie+13].

2.2 Différents éléments finis pour différentes primitives

Du point de vue de la méthode des éléments finis, les fonctions d'interpolation sont différentes pour les tétraèdres et les hexaèdres (§1.2.1). Dans cette section, nous nous intéressons aux conséquences (§2.2.1) de ces différences, qui expliquent pourquoi les hexaèdres sont parfois plus intéressants que les tétraèdres, et nous regardons quelles sont les possibilités pour construire des espaces d'approximation pour les maillages hex-dominants qui combinent plusieurs types de fonctions d'interpolation (§2.2.2).

2.2.1 Différences entre éléments finis tétraédriques et hexaédriques

Nous avons vu dans le chapitre d'introduction (§1.2.1, p. 16) que l'espace polynomial d'interpolation des hexaèdres ($\mathbb{Q}_k$) est différent de celui des tétraèdres ($\mathbb{P}_k$). Le premier permet une meilleure interpolation car il contient plus de termes polynomiaux et sa structure tensorielle permet un certain nombre d'optimisations (§2.2.1.1) qui ne sont pas possibles, ou plus difficiles à mettre en oeuvre, avec les tétraèdres. De plus, au niveau global, les codes de calcul utilisant des maillages hexaédriques structurés, ou structurés par blocs, peuvent mettre en place des stratégies d'optimisation exploitant cette régularité (§2.2.1.2). Ces effets combinés font que les hexaèdres sont souvent plus intéressants pour des simulations éléments finis, *c.à-d.* qu'ils offrent un meilleur ratio précision pour temps de calcul (§2.2.1.3). Cependant, les maillages tétraédriques peuvent être modifiés localement, ce qui leur permet de s'adapter plus facilement aux particularités des solutions (§2.2.1.4).

2.2.1.1 Avantages des éléments finis hexaédriques

Termes polynomiaux supplémentaires. $\mathbb{P}_k$ contient tous les polynômes de degrés inférieurs à k alors que $\mathbb{Q}_k$ contient les polynômes de degrés inférieurs à k selon chaque variable :

$$\mathbb{P}_k = \text{Vect}(x^i y^j z^l,\ 0 \leq i+j+l \leq k) \text{ et } \mathbb{Q}_k = \text{Vect}(x^i y^j z^l,\ 0 \leq i,j,k \leq k)$$

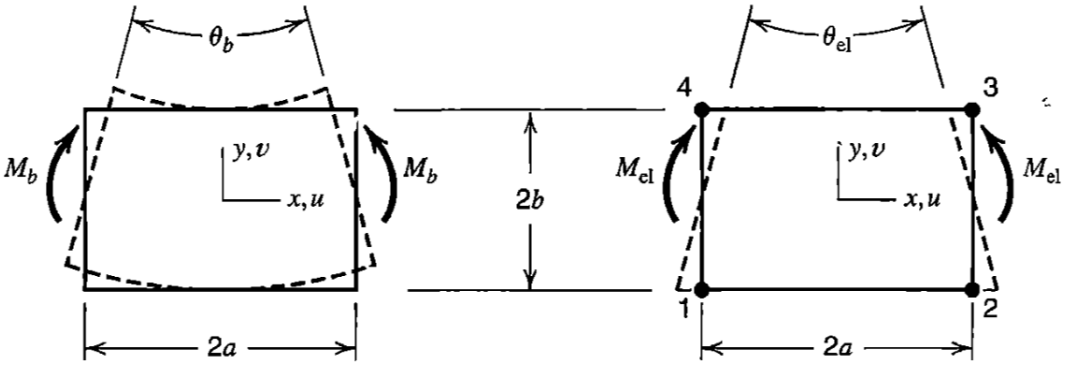


FIGURE 2.17 – *Gauche :* flexion pure d'un bloc rectangulaire. *Droite :* déformation d'un élément fini quadrilatéral $\mathbb{Q}_1$ en flexion. La flexion pure ne peut pas être représentée avec des arêtes droites. Source : [Coo+02, p. 99].

Pour un ordre k fixé, l'espace $\mathbb{Q}_k$ sur les hexaèdres contient des termes polynomiaux supplémentaires, que nous appelons termes croisés (illustrés sur la Figure 2.16). Un polynôme de $\mathbb{Q}_k$ peut avoir un degré $d \times k$, avec d la dimension de l'espace, mais il est incomplet, c'est pourquoi l'ordre d'interpolation est uniquement k. En pratique, ces termes supplémentaires permettent une meilleure approximation de la solution à approcher. Cet effet est visible sur les solutions du problème de Poisson présenté en introduction sur la Figure 1.7 (p. 22). L'augmentation de la précision peut être de l'ordre de 30% à temps de calcul constant, mais elle dépend de l'alignement des éléments avec la solution. Nous explorerons plus en détail ce phénomène dans le *Chapitre 4.1*.

Blocages. En élasticité linéaire et avec la formulation en déplacement, les triangles et tétraèdres linéaires ($\mathbb{P}_1$) sont particulièrement connus pour leurs mauvaises performances. Le maillage *bloque*, c'est-à-dire qu'il ne se déforme pas autant qu'il devrait. Ceci est dû au fait que (a) la contrainte σ est constante à l'intérieur des éléments car elle est liée linéairement aux gradients constants des composantes du déplacement, et (b) l'élément contient des modes de déformation parasites qui *absorbent* de l'énergie (*shear locking* en anglais). Un autre phénomène, le blocage volumique, intervient avec les matériaux quasi-incompressibles (coefficient de Poisson qui tend vers 0.5) [Coo+02, p. 93].

Les éléments $\mathbb{Q}_1$ souffrent également de ces phénomènes de blocage mais ils sont moins prononcés que pour les $\mathbb{P}_1$ car les termes croisés sur l'interpolation du déplacement (xy en 2D et xy, yz, xz, xyz en 3D) permettent de représenter partiellement certaines variations linéaires de la déformation ϵ et de la contrainte σ à l'intérieur des éléments. Concrètement, les éléments $\mathbb{P}_1$ et $\mathbb{Q}_1$ ne peuvent pas représenter la déformation en flexion pure illustrée sur la Figure 2.17.

Les phénomènes de *blocage* peuvent être subtils et sont assez spécifiques aux problèmes de mécanique, même s'il existe des difficultés analogues mais moins marquées dans d'autres domaines impliquant des champs vectoriels. Toutefois, pour les études de mécanique, ceux-ci ont parfois des conséquences très importantes sur les résultats. Le lecteur intéressé peut se reporter aux livres se concentrant sur la résolution de

problèmes mécaniques par la méthode des éléments finis [Mac93 ; Coo+02 ; Bat06].

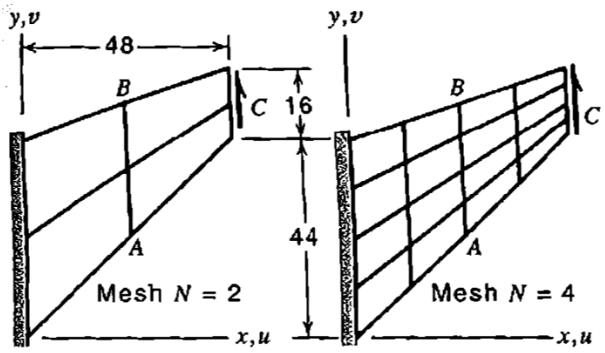
Intégrations réduites. L'intégration réduite consiste à utiliser moins de points de quadratures que nécessaire pour intégrer un polynôme. La sous-intégration de l'élément $\mathbb{Q}_1$, où l'on utilise un seul point de Gauss au centre, est particulièrement intéressante car elle permet de supprimer l'absorption parasite d'énergie évoquée dans le paragraphe précèdent (*shear locking*). Cependant, elle introduit d'autres modes de déformations parasites (*hourglass effect* en anglais). Ceux-ci peuvent être compensés en utilisant la technique *d'intégration sélective* : utilisation de quadratures complètes pour les composantes normales de la déformation et utilisation de la quadrature réduite (un point) pour les composantes en cisaillement, mais ceci introduit un biais directionnel [18][Coo+02, p. 223]. Ces optimisations, possibles à priori pour les quadrilatères et hexaèdres, et non pour les simplexes, sont utilisées dans les logiciels de simulation commerciaux (*par ex.* Nastran, ABAQUS), mais elles sont utilisées différemment suivant le type de simulation mécanique (statique ou dynamique, schéma temporel implicite ou explicite), les matériaux (quasi-incompressibles ou non) et l'ordre des polynômes (un ou deux en pratique en mécanique).

Modes incompatibles et éléments hybrides. Une autre manière d'éviter le blocage des éléments d'ordre un, qui résistent à la flexion (*shear locking*), est d'ajouter des degrés de liberté aux éléments $\mathbb{Q}_1$. Par exemple si on ajoute les polynômes x^2, y^2 au quadrilatère $\mathbb{Q}_1$, ceci permet de représenter tous les déplacements d'ordre deux (arêtes courbes de la flexion pure) et donc les variations linéaires de la déformation et de la contrainte. Ces nouvelles fonctions sont également appelées fonctions *bulles*. Cette modification de l'élément $\mathbb{Q}_1$ le rend incompatible : perte de la continuité du déplacement entre deux éléments adjacents le long de l'arête, mais la continuité au niveau des sommets partagés demeure. Cette modification des déplacements, introduite sous la forme de la brique de *Wilson* [Iro72], améliore fortement les résultats pour les parallélogrammes (transformations affines) mais ne fonctionne pas du tout pour des éléments quadrilatéraux quelconques. Pian et al. [PT86] ont montré qu'elle est très similaire à la formulation hybride *assumed stress element* introduite par Pian [Pia64] pour les éléments affines, et indique comment elle peut être modifiée pour fonctionner avec des géométries non-affines en ajoutant des contraintes numériques. Le résultat est un élément qui se comporte très bien en flexion, comme illustré sur le problème de la Figure 2.18, où il est quasiment deux fois plus précis que l'élément $\mathbb{Q}_1$. Le lecteur intéressé peut se référer au livre [PW05] qui se consacre entièrement à ces questions.

Les techniques d'ajout de modes incompatibles et de formulations hybrides ont surtout été développées pour les éléments quadrilatéraux et hexaèdres d'ordre un. Elles sont très intéressantes en pratique car elles permettent d'obtenir des performances équivalentes aux éléments d'ordre deux mais pour un coût bien inférieur. Il existe aussi des formulations hybrides (*assumed stress*) pour les simplexes mais celles-ci semblent moins intéressantes car (beaucoup) plus coûteuses.

Remarque. Les techniques de sous-intégration et d'ajout de modes incompatibles sont délicates à utiliser car elles induisent des *crimes variationnels* : les hypothèses

18. la matrice de rigidité calculée dépend de l'orientation de l'élément



	Mesh	N = 2	N = 4
QM6	v_C	0.884	0.967
	σ_A	0.840	0.978
	σ_B	0.788	0.926
Q4	v_C	0.498	0.769
	σ_A	0.558	0.830
	σ_B	0.457	0.753

FIGURE 2.18 – Problème de cisaillement : déformation d'une planche (coefficient de Poisson $\nu = 0.333$), fixée à gauche, par une force surfacique constante orientée vers le haut (C, côté droit). Deux résolutions de maillage quadrilatéral sont utilisées : 2×2 (gauche) et 4×4 (droite). Les résultats numériques (tableau à droite) sont : le déplacement v_c selon y en C, la contrainte normale σ_A maximale en A et la contrainte normale minimale σ_B en B. Les valeurs sont normalisées (valeurs exactes à 1.). Q4 est l'élément trilinéaire ($\mathbb{Q}_1$) et QM6 est l'élément $\mathbb{Q}_1$ amélioré avec deux modes incompatibles. Source : [Coo+02, p. 221].

des théorèmes apportant des garanties théoriques ne sont plus respectées. Par exemple, avec les modes incompatibles, les déplacements ne sont plus continus et donc il ne s'agit plus d'une méthode de Galerkine conforme. Comme elles sont très spécialisées, elles ne sont généralement pas implémentées dans les codes de calcul open-source, développés par des équipes de recherche qui se concentrent sur des approches plus génériques. Les méthodes hybrides sont basées sur une approche variationnelle du problème qui ressemble beaucoup aux formulations mixtes, utilisant plusieurs champs inconnus. La différence est qu'avec les méthodes hybrides, les nouveaux champs (*par ex.* déformation ϵ en élasticité) sont éliminés au moment de l'assemblage et il ne reste que des inconnus d'un seul type (*par ex.* déplacement) dans le système linéaire final.

Optimisation de l'assemblage par factorisation-somme. Lorsque l'on utilise des éléments d'ordres élevés ($k \geq 3$), comme c'est souvent le cas pour l'étude de la propagation d'ondes, le nombre de degrés de liberté augmente très rapidement et la construction selon la méthode classique des matrices locales, associées à chaque élément, devient très couteuse. Pour que le coût ne soit pas prohibitif, les codes de calculs reposent sur la technique de factorisation-somme, introduite par ORSZAG [Ors80], pour calculer les coefficients. Elle requiert que la fonction à intégrer par quadratures puisse s'écrire sous forme tensorielle. Par exemple en dimension deux, si on souhaite intégrer une fonction $f(x, y) = f_1(x)f_2(y)$ avec une quadrature 2D $(w_{ij}, (x_i, x_j))_{ij}$ formée par produit de deux quadratures 1D $(w_i, x_i)_i$, on peut factoriser la double somme comme ci-dessous :

$$c = \sum_{i=1..n} \sum_{j=1..n} w_{ij} f(x_i, y_j) = \sum_{i=1..n} w_i f_1(x_i) \left(\sum_{j=1..n} w_j f_2(y_j) \right)$$

Cette technique s'applique directement aux éléments $\mathbb{Q}_k$, car ils ont une construction tensorielle naturelle. Elle permet de réduire le nombre d'opérations de $\mathcal{O}(k^{3d})$ à

$\mathcal{O}(k^{2d+1})$, avec k l'ordre des polynômes et d la dimension de l'espace. Il est possible de mettre en place une factorisation similaire pour les simplexes mais pas avec les bases de Lagrange interpolantes. Ainsi, Karniadakis et al. [KS] construit une base tensorielle pour les simplexes par dégénérescence d'un cube, mais celle-ci n'est pas symétrique par rotation. Une autre possibilité proposée plus récemment est d'utiliser les bases de Bernstein-Bézier [AAD11].

Éléments finis spectraux. La méthode des éléments spectraux (SEM), introduite pour les quadrilatères et les hexaèdres, consiste à utiliser des polynômes d'ordre élevé avec la même répartition de Gauss-Lobatto des degrés de liberté et des points de quadrature [MGS01]. L'avantage de cette approche est que la matrice de masse est diagonale. La méthode des éléments spectraux a été étendue aux simplexes mais il n'est pas possible, ou intéressant, de prendre les mêmes points pour les degrés de liberté et les points de quadrature. Par conséquent, la construction d'éléments spectraux performants pour les tétraèdres n'est pas aussi évidente que pour les cubes. Différentes approches sont possibles [PR10], chacune ayant ses avantages et inconvénients.

2.2.1.2 Optimisations dues à la structure du maillage

Dans la partie précédente, nous avons évoqué des propriétés des éléments hexaédriques qui s'appliquent sur tous les types de maillages hexaédriques. Lorsque le maillage hexaédrique est structuré (grille déformée), des optimisations à un niveau plus global sont également possibles. Dans le cas des maillages structurés par blocs, ces optimisations peuvent être appliquées sur chacun des blocs.

Connectivité régulière. Dans un maillage hexaédrique structuré, les sommets, arêtes, faces appartiennent respectivement à huit, quatre et deux éléments. Cette régularité implique, avec la bonne numérotation, une largeur de bande constante dans la matrice du système linéaire, et il est possible d'utiliser des algorithmes très optimisés de produits matrices-vecteurs qui exploitent pleinement les capacités de calculs parallèles des CPUs et GPUs [TAB99 ; TBK06]. La structure régulière permet également, avec une numérotation adéquate, de ne pas stocker les adjacences entre éléments du maillage car celles-ci peuvent être calculées implicitement, ce qui évite des accès mémoires coûteux.

Solveur multigrille géométrique. Les solveurs multigrilles partent du système linéaire à résoudre et construisent une séquence de systèmes linéaires de plus en plus grossiers, de manière à approximer le système initial. Ceci permet de résoudre d'abord les basses fréquences de la solution puis les plus hautes. Cette approche est très efficace mais elle nécessite de construire une séquence de systèmes linéaires de tailles différentes, et de communiquer entre les différents niveaux avec des matrices de projection (prolongation et restriction). Les solveurs multigrilles *géométriques* exploitent la connaissance du maillage pour construire cette séquence. Avec des maillages structurés, cette construction est relativement simple, par exemple on peut construire un maillage plus grossier en ne conservant qu'un sommet sur deux. Ces méthodes sont particulièrement employées en mécanique des fluides [WO01 ; Tur+10].

2.2.1.3 Différences de performances dans la littérature

Nous nous intéressons ci-dessous aux différences de précision et de performances, entre maillages entièrement hexaédriques, hex-dominants et tétraédriques, reportées dans la littérature pour différents cas d'applications couvrant différents domaines de la physique et de l'ingénierie. Les performances sont comprises au sens du ratio entre précision et temps de calcul. La manière d'effectuer des simulations éléments finis avec des maillages hex-dominants est détaillée dans (§2.2.2) et les techniques utilisées pour mesurer les différences font l'objet de (§2.3).

En mécanique. Cifuentes et al. [CK92] comparent la précision des hexaèdres trilinéaires ($\mathbb{Q}_1$), des tétraèdres linéaires ($\mathbb{P}_1$) et quadratiques ($\mathbb{P}_2$) sur des problèmes de flexion, cisaillement et torsion de domaines cubiques (barres), avec une modélisation d'élasticité linéaire. Les conclusions sont que les tétraèdres linéaires sont très mauvais, et que les hexaèdres trilinéaires et les tétraèdres quadratiques sont équivalents en termes de précision et de temps de calcul. Benzley et al. [Ben+95] étendent l'étude de la déformation élastique de la barre au cas élasto-plastique et au comportement dynamique. Conformément à [CK92], les hexaèdres trilinéaires sont équivalents aux tétraèdres quadratiques, y compris en dynamique. En élasto-plasticité, les hexaèdres sont encore plus avantageux : les tétraèdres $\mathbb{P}_2$ sont plus mauvais que les $\mathbb{Q}_1$. Les deux études précédentes utilisent des maillages tétraédriques formés par décomposition des hexaèdres en six tétraèdres, ce qui produit des maillages tétraédriques beaucoup plus réguliers que des maillages réels. Sur des modèles compliqués de pièces mécaniques, Wang et al. [WNR04] comparent les performances des éléments (améliorés) tétraédriques et hexaédriques de *Ansys* sur divers problèmes de mécanique (déformation non-linéaire, plastique, contact). Ils concluent que les tétraèdres $\mathbb{P}_2$ apportent toujours des résultats satisfaisants et que les hexaèdres trilinéaires améliorés peuvent être suffisants mais qu'ils doivent être utilisés avec précaution.

On peut également citer Danielson et al. [DAW16], qui se concentrent sur le développement d'éléments prismatiques pour des simulations mécaniques non-linéaires avec schéma temporel explicite. L'une des expériences, sur l'impact d'un cylindre contre un mur, compare des éléments hexaédriques et tétraédriques d'ordre deux. Les auteurs indiquent que la simulation avec tétraèdres est 20 fois plus lente.

En biomécanique, Bourdin et al. [Bou+07] étudient la déformation d'un rein suite à un impact, avec un modèle hyper-élastique et un modèle visqueux de matériau. Ils observent que les écarts de précision entre hexaèdres et tétraèdres sont négligeables devant d'autres facteurs (tétraèdres environ 10% plus rigides que les hexaèdres) mais le pas de temps nécessaire pour les tétraèdres est plus faible, ce qui rend les calculs 1.75 fois plus lents. Tadepalli et al. [TEC11] étudient la déformation hyper-élastique d'un pied en contact avec le sol, en utilisant des éléments finis hexaédriques et tétraédriques d'ordre un et deux, améliorés avec des composantes hybrides pour la pression [19]. Les conclusions sont que les hexaèdres permettent des prédictions de contraintes plus consistantes (oscillations dues au maillage pour les tétraèdres), particulièrement lorsqu'il y a des phénomènes de friction. Ceci dit, les deux études précédentes n'étudient les résultats que pour un seul niveau de résolution, sans raffinement du maillage, et n'utilisent pas de solution de référence pour effectuer leurs

19. En notations Abaqus : C3D4H, C3D10H, C3D10MH pour les tétraèdres et C3D8H pour les hexaèdres

comparaisons, qui sont très qualitatives (allure visuelle de la distribution de pression, etc). Shoykhet et al. [Sho+04] étudient la déformation hyper-élastique de l'aorte abdominale avec des maillages tétraédriques, hexaédriques et hex-dominants, et compare les résultats à une solution de référence en calculant [20] l'erreur L^2. Les éléments finis utilisés sont également des éléments améliorés d'Abaqus. Les auteurs concluent que les maillages hexaédriques sont les plus intéressants et permettent de converger beaucoup plus rapidement mais il n'y a pas beaucoup de détails dans l'article. Ramos et al. [RS06] étudient la déformation élastique d'un modèle de fémur avec différentes résolutions de maillages, en observant le déplacement et la contrainte de Von Mises en cinq points d'échantillonnages. Les différences entre ordre un et deux sont faibles (moins de 10%), et en comparant à la solution analytique d'un modèle simplifié, les auteurs concluent que les tétraèdres linéaires sont les plus proches de la solution théorique. Cependant cette dernière conclusion n'a pas beaucoup de sens car le domaine géométrique de la solution analytique n'a que très peu de similarités avec les maillages du fémur utilisés.

En électromagnétisme. La thèse [Ber10] effectue des comparaisons entre maillages tétraédriques, hexaédriques et hex-dominants (avec pyramides). Sur un problème d'Helmholtz sur domaine sphérique, les éléments tétraédriques mettent de deux à quatre fois plus de temps pour atteindre le même niveau de précision (différence en norme L^2 avec une solution de référence) que les maillages hexaédriques et hex-dominants, pour des ordres polynomiaux variant de deux à quatre. Sur un problème de propagation d'ondes dans un piano, résolu avec la méthode de Galerkine discontinue, le maillage hex-dominant est environ quatre fois plus performant que le maillage tétraédrique.

La thèse [Dur13] s'intéresse aux simulations d'électromagnétismes sur maillages hexa-tétraédriques (interfaces non-conformes) avec la méthode de Galerkine discontinue et évolution temporelle. Les résultats numériques montrent que les maillages hybrides offrent un meilleur ratio entre précision et temps de calcul, compris généralement entre deux et quatre en fonction des problèmes. Fahs [Fah15] effectue des simulations similaires mais avec des non-conformités plus génériques (les faces de tétraèdres et d'hexaèdres sont dans le même plan mais ne partagent pas d'arêtes) et obtient des résultats similaires : les maillages hybrides (majorité d'hexaèdres) sont deux à huit fois plus performants, ce qui s'explique principalement par la possibilité d'utiliser un pas de temps plus large. Ces gains sont également confirmés par Zhao et al. [ZCY16] sur des simulations similaires, à la différence que les interfaces entre hexaèdres et tétraèdres sont assurées par des pyramides.

Sevilla et al. [SHM14] étudient en détails les gains apportés par l'utilisation de maillages hybrides pour la résolution temporelle des équations de Maxwell avec la méthode de Galerkine discontinue (le cas des méthodes précédentes). Cet article confirme que l'utilisation de maillages hybrides peut apporter un gain de performance significatif, allant jusqu'à un facteur quinze. Cependant le gain est fortement dépendant de la proportion d'hexaèdres et il est en grande partie du au fait que les hexaèdres considérés ont des transformations affines, régions structurées qui maillent le milieu de propagation, ce qui permet de pre-calculer des termes et évitent de couteux calculs de quadratures.

20. la méthode de calcul de l'erreur n'est pas indiquée

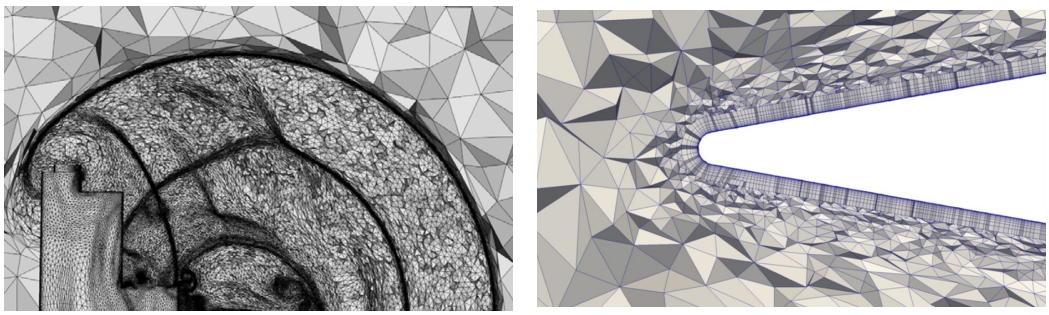


FIGURE 2.19 – Adaptation de maillages. *Gauche :* maillage tétraédrique anisotrope adaptatif pour ondes de chocs, source : [AL16]. *Droite :* maillage hybride avec maillage hexaédrique structuré de la couche limite , source : [Sah+16].

Dans les travaux évoqués précédemment, les maillages hybrides sont construits manuellement et comportent de grandes régions structurées, ainsi des obstacles maillés avec des tétraèdres. Cette configuration particulière permet un certain nombre d'optimisations qui ne sont pas applicables aux maillages hexaédriques et hex-dominants d'objets complexes, dont ceux sur lesquels ce manuscrit se concentre.

2.2.1.4 Adaptation du maillage aux champs

Sur des problèmes difficiles, *par ex.* fortes non linéarités et mécanique des fluides, il est souvent nécessaire d'adapter le maillage aux particularités du problème et des solutions. Nous évoquons deux techniques couramment employées : raffinement adaptatif et maillage des couches limites.

Raffinement adaptatif. Une manière efficace d'améliorer le ratio précision par temps de calcul est d'utiliser des éléments plus petits là où l'erreur est la plus importante, ou du moins dans les régions où les champs varient fortement. Comme les maillages tétraédriques sont entièrement non-structurés et peuvent être modifiés avec des opérations locales, il est possible d'adapter fortement le maillage au champ [Bak97], y compris en utilisant des tétraèdres très anisotropiques [FA05 ; Rem+05 ; LDA10]. Ces techniques ont été fortement étudiées et développées ces deux dernières décennies [AL16] et elles sont aujourd'hui implémentées et utilisables via de nombreuses librairies et logiciels. L'adaptation pour la simulation d'une onde de choc est illustrée sur la gauche de la Figure 2.19.

La génération de maillages hexaédriques conformes, adaptés aux variations internes du champ, est beaucoup plus difficile voire impossible. En exploitant des maillages hybrides, l'article de QIN et al. [QL06] et la thèse HARRIS [Har13] explorent la génération de maillages structurés partiels, où les blocs structurés suivent les régions des écoulements avec de fortes variations (*par ex.* ondes de chocs). Le reste du maillage utilise des quadrilatères non structurés et des triangles en dimension deux, des hexaèdres, prismes et pyramides en dimension trois. Ces développements sont intéressants mais très loin d'être aussi matures que leurs équivalents pour maillages tétraédriques.

Maillage anisotropique des couches limites. En mécanique des fluides, particulièrement pour les fluides visqueux, il est très important de modéliser avec un maillage très raffiné le comportement du fluide au contact d'un obstacle car des variations très fortes des champs ont lieu dans cette couche limite. Comme la couche limite est généralement *localement plane*, les maillages structurés sont particulièrement adaptés pour la mailler, surtout que les hexaèdres peuvent être très anisotropes tout en conservant des angles droits. Ces considérations ont donné lieu au développement de techniques de génération maillages hybrides structurés dans la couche limite et non-structuré ailleurs [Sah+08 ; Iba+16 ; Sah+16], utilisés par des logiciels de CFD industriels comme *elsA* [Pui+11], développé par l'ONERA. Le maillage d'une couche limite est illustré sur la droite de la Figure 2.19.

2.2.1.5 Conclusions sur les différences entre hexaèdres et tétraèdres

Les maillages hexaédriques étant bien plus difficiles à générer que les maillages tétraédriques, il faut qu'ils apportent un gain important pour la simulation numérique afin de justifier l'effort de maillage. Cependant, le champ des applications de la méthode des éléments finis est très vaste et chaque application a ses spécificités, par conséquent il est difficile voire impossible d'apporter une réponse catégorique et générique à cette question. En se basant sur les articles effectuant des comparaisons dans la littérature, nous distinguons quelques tendances :

— Pour la mécanique, qui utilise majoritairement des éléments d'ordres un et deux, il semble que les hexaèdres sont intéressants car ceux d'ordre un peuvent être modifiés (modes incompatibles, hybride, intégration sélective) afin d'atteindre des performances équivalentes aux tétraèdres quadratiques tout en conservant un coût de calcul bien inférieur ;

— Pour la propagation d'ondes électromagnétiques, souvent basée sur l'utilisation d'éléments spectraux (ordres élevés), l'optimisation des calculs numériques semble plus simple et efficace pour les éléments hexaédriques qui ont naturellement une structure tensorielle ;

— Pour la mécanique des fluides mettant en jeu des variations très fortes et très localisées, le maillage adaptatif évoluant dans le temps est nécessaire et les tétraèdres ont largement l'avantage de par leur flexibilité. Les hexaèdres sont plus intéressants pour mailler les couches limites le long des obstacles, dont la géométrie est connue à l'avance.

Un argument souvent mis en avant dans les articles de maillages hexaédriques par champs de directions est que ceux-ci sont plus performants car ils sont alignés avec les bords des objets. Cet argument est peut être vrai mais nous n'avons pas trouvé d'étude quantitative le confirmant.

2.2.2 Éléments finis sur maillages hex-dominants

Il y a plusieurs cas d'utilisations de maillages comportant des éléments de nature différente dans la littérature. Souvent, les différents types sont associés à des régions distinctes du problème : les hexaèdres sont utilisés pour les régions *régulières* (e.g. milieu de propagation) et les tétraèdres pour les parties géométriquement compliquées (e.g. obstacle). Le lien entre les deux types d'éléments est soit assuré par

des interfaces non conformes, soit par des éléments d'interface comme les pyramides et prismes. Du point de vue de la topologie des maillages, cette situation est assez différente des maillages étudiés dans cette thèse, où les interfaces entre éléments différents sont réparties aléatoirement dans le maillage. Cependant, pour la méthode des éléments finis, la question est la même : comment lier les espaces fonctionnels définis sur les hexaèdres et ceux définis sur les autres éléments. Nous étudions deux catégories d'approches possibles :

— avec un maillage conforme : définir des fonctions d'interpolation dans les éléments d'interface (faces triangulaires et quadrilatères) qui soient compatibles avec les fonctions des hexaèdres et celles des tétraèdres (§2.2.2.1) ;

— avec un maillage non-conforme hexa-tétraédrique : modifier les éléments et/ou ajouter des contraintes aux interfaces non-conformes de manière à ce que la méthode des éléments finis fonctionne toujours (§2.2.2.2).

Remarque. D'autres idées plus exotiques sont envisageables. Par exemple, en s'inspirant des méthodes *overset* [BBS85], on pourrait superposer des grilles régulières autour des régions non hexaédriques et les coupler faiblement avec le maillage hexaédrique partiel. Ceci dit, le fonctionnement du code de simulation serait très diffèrent d'un code d'éléments finis pour maillages classiques.

2.2.2.1 Maillages hex-dominants conformes

La manière la plus logique de construire un maillage hybride est d'utiliser des éléments compatibles les uns avec les autres. Si un maillage contient des hexaèdres (faces quadrilatérales) et des tétraèdres (faces triangulaires), alors il faut utiliser des éléments d'interface comme la pyramide et le prisme qui contiennent les deux types de faces (Figure 2.20). Il est alors possible de construire un maillage hybride conforme, c'est-à-dire un maillage où l'intersection entre deux éléments est soit le vide, soit un point, soit une arête, soit une face commune aux deux éléments. En pratique, il est rare de pouvoir n'utiliser que des prismes comme éléments d'interfaces, car ceux-ci ont une direction principale : celle qui lie les deux triangles. La plupart des maillages hex-dominants conformes utilisent majoritairement des pyramides comme éléments d'interfaces.

En fonction de l'espace de la solution de la formulation faible continue (H^1, $H(rot)$, $H(div)$ ou L^2), nous souhaitons que le champ discrétisé sur le maillage respecte respectivement la continuité du champ, de la composante tangentielle, de la composante normale, ou aucune, aux interfaces entre éléments. Comme dans le reste de ce manuscrit, nous nous concentrons surtout sur les éléments finis pour H^1 : continuité du champ aux interfaces. Ceci dit, les idées évoquées s'appliquent de manière similaire aux éléments finis de $H(rot)$ et $H(div)$.

L'objectif est de définir des éléments finis pour la pyramide et pour le prisme tels que les restrictions sur les faces triangulaires soient conformes avec $\mathbb{P}_k$ et les restrictions sur les faces quadrilatérales soient conforme avec $\mathbb{Q}_k$.

Le cas du **prisme** est relativement simple. De la même manière que l'élément fini de $\mathbb{Q}_k$ est formé par un produit tensoriel de trois polynômes 1D définis sur trois arêtes orthogonales, on peut former un élément fini pour prismes par produit d'un polynôme de degré k défini sur le triangle ($\mathbb{P}_k^{2D}$) et d'un polynôme 1D de degré k

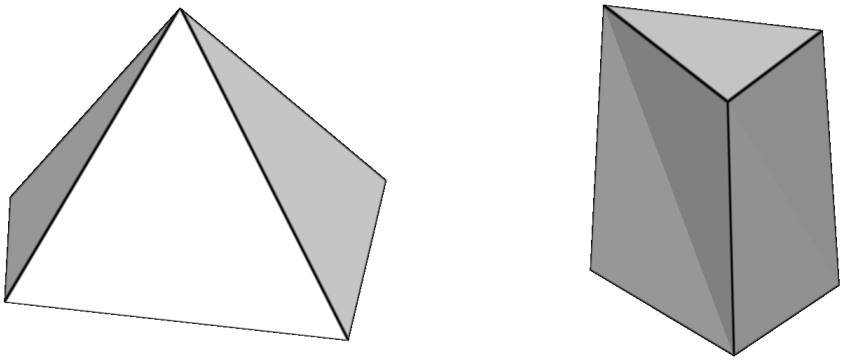


FIGURE 2.20 – Éléments d'interface. *Gauche :* pyramide (4 faces triangulaires et 1 face quadrilatérale). *Droite :* prisme (2 faces triangulaires et 3 faces quadrilatérales).

défini sur une arête orthogonale [EG02, p. 34]. Formellement, l'espace d'interpolation d'ordre k des prismes est :

$$\mathcal{P}_{prisme} = \mathbb{P}_k^{2D} \oplus \mathbb{P}_k^{1D}$$

L'ensemble des degrés de liberté est celui du triangle extrudé $k+1$ fois dans la direction orthogonale. Ces degrés de liberté correspondent naturellement à ceux des hexaèdres sur les faces quadrilatérales et à ceux des tétraèdres sur les faces triangulaires. Avec un espacement non équidistant des points de contrôle (*par ex.* Gauss-Lobatto), il faut faire attention que l'espacement sur les arêtes soit le même pour les hexaèdres et tétraèdres.

La construction d'éléments finis sur les **pyramides** est plus compliquée car il est impossible de construire un *polynôme* dont la restriction sur les faces triangulaires est un polynôme d'ordre k ($\mathbb{P}_k^{2D}$ pour la compatibilité avec tétraèdres adjacents) et la restriction sur la face quad un produit de deux polynômes univariés d'ordre k ($\mathbb{Q}_k^{2D}$ pour la compatibilité avec hexaèdres adjacents), voir [Bed92 ; Wie97]. Deux approches pour contourner cette difficulté ont été développées :

— construire un élément fini pyramidal à partir de fonctions rationnelles ;

— décomposer artificiellement la pyramide en tétraèdres et définir des polynômes sur ceux-ci, c'est-à-dire définir un élément fini pyramidal *polynomial par morceaux*, ou également appelé *composite*.

Le lecteur intéressé par les pyramides peut également se référer à la thèse de BERGOT [Ber10] qui explore le sujet en détail. Dans la suite, nous nous contentons d'évoquer les grandes lignes sans entrer dans les détails théoriques et techniques.

Pyramide : fonctions rationnelles. L'utilisation de fonctions de formes rationnelles pour les pyramides, afin de satisfaire les exigences de continuité avec les éléments adjacents, remonte à BEDROSIAN [Bed92], COULOMB et al. [CZM97] et ZGAINSKI et al. [Zga+96] pour l'ordre un et deux. Cependant les fonctions d'ordre deux proposées ont des défauts et ne sont pas complètement conformes avec les hexaèdres $\mathbb{Q}_k$ (voir [Ber10]).

Les bases de fonctions rationnelles ont été étendues à l'ordre arbitraire k par Bergot et al. [BCD09] et Bergot [Ber10] pour l'espace H^1. L'espace d'interpolation H^1 défini sur la pyramide de référence est :

$$\mathcal{P}_{pyr} = \mathbb{P}_k^{3D} \oplus \sum_{0 \leq i \leq k-1} \left(\frac{xy}{1-z} \right)^{k-i} \mathbb{P}_k^{2D}$$

où le terme à gauche assure la continuité avec les tétraèdres et le terme rationnel à droite permet d'être conforme avec la face quadrilatérale ($\mathbb{Q}_k^{2D}$) lorsque $z \to 0$. Les formules explicites des fonctions de la base de cet espace rationnel sont assez complexes car il faut différencier les arêtes qui touchent seulement les triangles des arêtes qui touchent les triangles et le quadrilatère.

Pour gérer les espaces $H(rot), H(div)$, et surtout les formulations mixtes qui mélangent des champs appartenant à différents espaces, Nigam et al. [NP11] construisent des éléments finis stables sur la pyramide pour $H^1, H(rot), H(div), L^2$, c'est-à-dire dont le diagramme de De Rham continu-discret est commutatif. La construction repose sur l'utilisation d'une pyramide *infinie* (le sommet de la pyramide est à l'infini), qui est ensuite transformée vers la pyramide de référence classique. Cependant, ceux-ci n'ont une convergence optimale qu'avec des pyramides dont la géométrie est affine [BD13], c'est-à-dire avec une base quadrilatérale plane, ce qui n'est jamais le cas dans un maillage hex-dominant généré automatiquement. Fuentes et al. [Fue+15] introduisent une construction cohérente et unifiée d'éléments finis pour les hexaèdres, tétraèdres, prismes et pyramides. Le développement des pyramides repose sur les travaux de Nigam et al. [NP11] et l'introduction de coordonnées barycentriques [21]. Il n'est pas clair que ces constructions convergent de manière optimale ou non pour des pyramides non-affines.

Dans la cadre de la méthode de Galerkine discontinue, Chan et al. [CW16] ont construit une base orthogonale de $\mathcal{P}_{pyr}$, qualifiée de *semi-nodale* car elle n'est pas complètement interpolante. Les matrices de masse construites avec cette base sont diagonales, ce qui permet une implémentation GPU efficace de simulations avec schémas temporels explicites, fonctionnant sur des maillages hybrides [Cha+16].

Pyramide : fonctions polynomiales composites La seconde approche repose sur une décomposition de la pyramide en deux ou quatre tétraèdres sur lesquels sont définies des bases polynômiales. Les fonctions rationnelles sont remplacées en utilisant des polynômes par morceaux.

Wieners [Wie97] utilisent la décomposition en deux tétraèdres (Figure 2.21) pour introduire des fonctions de formes compatibles avec les tétraèdres et hexaèdres pour l'ordre un et deux. Afin d'éviter l'anisotropie arbitraire introduite par la décomposition en deux tétraèdres (choix d'une diagonale), Liu et al. [Liu+04 ; Liu+11] proposent de décomposer la pyramide en quatre tétraèdres afin de former des fonctions de formes symétriques.

Pour des ordres arbitraires, Bluck et al. [BW07] construisent une base continue pour H^1 et Marais et al. [MD08] proposent une construction pratique de bases pour H(div) et H(rot) où les tétraèdres de la pyramide sont équipés de bases d'ordres supérieurs et contraints à l'interface hexa-tétraédrique.

21. se comportant comme des coordonnées barycentriques, ce n'en est pas strictement

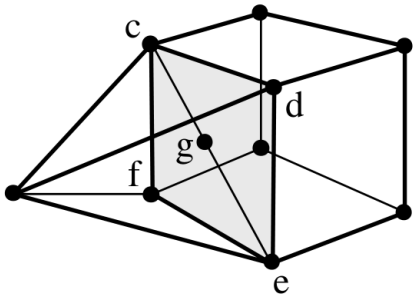


FIGURE 2.21 – Interface non-conforme entre deux tétraèdres et un hexaèdre. Source : [OCS97].

Alternativement, AINSWORTH et al. [ADS16] proposent une base construite avec les polynômes de Bernstein-Bézier, valable pour H^1 et un ordre arbitraire. L'extension de cette formulation à une séquence exacte (diagramme commutatif) d'espaces pour $H(div)$, $H(rot)$, L^2 a été établie par AINSWORTH et al. [AF17] pour l'ordre un. À notre connaissance, il n'y pas encore d'équivalent à un ordre arbitraire.

L'utilisateur de pyramides a le choix entre ces deux types d'espaces : fonctions rationnelles ou polynomiales par morceaux. La seconde introduit des discontinuités supplémentaires sur les dérivées à l'intérieur de l'élément, mais en contrepartie il n'y a pas besoin de se soucier des difficultés liées aux fonctions rationnelles (quadratures). Comme les deux approches convergent de manière optimale avec l'ordre choisi, ce choix ne devrait pas avoir beaucoup d'importance sur la précision de la solution, ni sur le temps de calcul, étant donné que le nombre d'éléments pyramidaux dans un maillage hex-dominant est censé être faible.

2.2.2.2 Maillages hex-dominants non conformes

Plutôt que de joindre les hexaèdres et les tétraèdres avec des pyramides, il est possible d'effectuer directement des simulations éléments finis sur le maillage non-conforme. Dans la suite, nous nous restreignons aux cas où chaque interface non-conforme est constituée de deux faces de tétraèdres qui se connectent à une face quadrilatérale, comme sur la Figure 2.21.

Le premier article [Dew+93] s'intéressant à cette idée, dans le cadre de la méthode de Galerkine classique avec continuité H^1, explore numériquement plusieurs configurations et conclut que la meilleure approche est d'utiliser des tétraèdres $\mathbb{P}_2$ et des hexaèdres $\mathbb{Q}_1$, en imposant que le degré de liberté au centre de la face (milieu de l'arête des deux tétraèdres) soit contraint à la moyenne des quatre valeurs sur les sommets du quad, c'est-à-dire $u_g = \frac{1}{4}(u_c + u_d + u_e + u_f)$ sur la Figure 2.21. Ce n'est pas étonnant que ce soit la seule solution viable vu que les autres essais ne respectent pas la continuité, ni n'ajoutent d'autres contraintes pour pénaliser la discontinuité. Cependant, cette solution fonctionne uniquement pour les transformations affines, c'est-à-dire les cas où les faces quadrilatérales sont planes. Nous développerons ce point dans la suite (§4.2). À notre connaissance, cette idée n'a pas été très populaire pour H^1 puisqu'elle a été rapidement abandonnée au profit d'approches basées sur la pyramide [OCS97].

En électromagnétisme, *c.à-d.* pour $H(rot)$, plusieurs articles se sont penchés sur

le problème de connecter directement des hexaèdres et des tétraèdres. Pour l'ordre un, Degerfeldt et al. [DR06] proposent de traiter les interfaces non-conformes hexa-tétraédriques avec la méthode de pénalisation de Nitsche [Nit71], de manière analogue à la méthode de Galerkine discontinue. Cette approche permet d'assurer la continuité faible de la composante tangentielle du champ vectoriel aux interfaces hybrides. Elle a été étendue à des ordres polynomiaux quelconques par Winges et al. [WR16]. Une autre approche [MD08] propose d'augmenter l'ordre des polynômes sur les tétraèdres et d'ajouter des contraintes pour obtenir la continuité de la composante tangentielle. Toujours en électromagnétisme, mais cette fois dans le cadre de la méthode de Galerkine discontinue, Durochat [Dur13], Durochat et al. [DLS13] et Fahs [Fah15] proposent un couplage faible des composantes tangentes entre hexaèdres et tétraèdres, cohérent avec le couplage faible naturel au niveau de toutes les faces internes.

Ces articles ne s'intéressent pas à la conformité géométrique entre hexaèdres trilinéaires et tétraèdres, ce qui implique que les formulations développées ne fonctionnent que pour des interfaces hexaèdres-tétraèdres planes, *c.à-d.* des transformations affines.

Remarque. Les fonctions polynomiales composites sur la pyramide, évoquées dans la partie précédente, sont plus ou moins équivalentes aux approches [Dew+93 ; MD08] qui construisent un espace fonctionnel conforme au niveau des interfaces non-conformes *1 hexaèdre-2 tétraèdres*. La différence concerne plutôt l'implémentation : soit les contraintes sont prises en compte au moment de l'assemblage (pyramide composite), soit ce sont des contraintes sur les variables du système (contraintes linéaires).

2.3 Méthodes pour comparer des solutions éléments finis

L'un des intérêts principaux de la méthode des éléments finis est de pouvoir résoudre des problèmes avec des domaines géométriques compliqués grâce à l'utilisation de maillages. Cependant lorsque le domaine est compliqué, il est difficile de quantifier l'impact du maillage sur la précision de l'approximation car il n'existe généralement pas de solution analytique, à moins de créer un problème complètement artificiel avec la méthode des solutions manufacturées (§2.3.1). Sans solution analytique, il reste la possibilité de construire une solution de référence à partir d'un maillage très raffiné ou d'éléments finis d'ordres élevés, puis de comparer les différents résultats obtenus avec différents maillages à cette référence. Il est possible de comparer des quantités d'intérêts spécifiques au problème (§2.3.2) ou d'effectuer une comparaison globale, par exemple une distance entre les différents champs scalaires ou vectoriels (§2.3.3).

Dans le *Chapitre 5*, nous proposons une méthode efficace de calcul de distances entre solutions éléments finis, proche des techniques de visualisation détaillées à la fin de cet état de l'art (§2.3.4).

2.3.1 Méthode des solutions manufacturées

Pour une équation aux dérivées partielles (ou un système d'équations) et un domaine fixé, l'idée de la méthode des solutions manufacturées [SK00 ; Roa02] est de choisir une solution analytique et de l'injecter dans l'équation de manière à calculer

le terme source et les conditions aux limites. On peut ensuite résoudre le problème et essayer de retrouver la solution analytique.

Par exemple pour l'équation de Poisson $-\Delta u = f$ sur le domaine $[0,1]^2$, avec conditions de Dirichlet u_b sur le bord, on peut choisir la solution :

$$u(x,y) = \cos(2\pi x)\cos(2\pi y)$$

et l'injecter dans le problème, pour obtenir : $f(x,y) = 4\pi^2\cos(2\pi x)\cos(2\pi y)$ dans Ω et $u_b = 0$ sur $\partial\Omega$.

À partir de la solution analytique u et de la solution approchée u_h, l'erreur d'approximation peut être calculée, dans la norme souhaitée, à partir de quadratures. Par exemple pour l'erreur L^2 :

$$\|u - u_h\|_2^2 = \int_\Omega (u-u_h)^2 = \sum_{K\in\Omega_h}\sum_i w_i (u(\mathcal{M}_K(\hat{\mathbf{x}}_i)) - \hat{u}_K(\hat{\mathbf{x}}_i))^2$$

avec $(w_i, \hat{\mathbf{u}}_i)_i$ les poids et points de quadrature dans l'élément de référence, $\mathcal{M}_K$ la transformation associée à l'élément K et $\hat{u}_K$ l'interpolation éléments finis associée à K, définie dans l'espace de référence $\hat{K}$.

Cette approche est très utile parce qu'elle permet de tracer des courbes de convergence (erreur en fonction du raffinement) avec lesquelles il est possible de :

— vérifier qu'un code de calcul fonctionne correctement ;

— vérifier qu'un espace d'approximation par éléments finis se comporte conformément aux attentes (*par ex.* convergence optimale avec des solutions analytiques lisses) ;

— développer des benchmarks mettant en jeu certains types de difficultés. Par exemple, Mitchell [Mit13] propose une liste de problèmes construits pour éprouver les algorithmes de raffinement adaptatif de maillages.

Il est également possible de comparer la précision d'approximation obtenue avec différents maillages, mais les résultats obtenus ne sont pas forcément représentatifs de situations réelles car :

— les solutions choisies sont souvent excessivement lisses, contrairement aux solutions de problèmes obtenus avec des géométries compliquées (*par ex.* coins rentrants) ;

— le terme source est uniformément présent et varie beaucoup : il influence fortement la solution. Pour des problèmes d'ingénierie classique, le terme source est plutôt soit absent, soit constant (*par ex.* gravité), et les variations de la solution sont plutôt dues à une propagation de l'effet des conditions aux limites vers l'intérieur du domaine.

2.3.2 Comparaison de quantités d'intérêt

Lorsqu'il n'y a pas de solution analytique pour un problème particulier, on peut observer la variation de quantités d'intérêt par rapport au raffinement du maillage.

Dans le cadre de simulations d'ondes électromagnétiques, Bergot et al. [BCD09 ; BD13] étudient l'*erreur de dispersion*, basée sur l'erreur de phase, pour différents espaces d'interpolation définis pour les pyramides.

En mécanique des structures, Cifuentes et al. [CK92] comparent les performances de tétraèdres et hexaèdres sur un modèle de poutre, et une modélisation d'élasticité linéaire, en observant le déplacement en un point et la fréquence fondamentale de résonance (équivalente à la première valeur propre). Benzley et al. [Ben+95] étendent cette étude aux valeurs propres suivantes, et s'intéresse aux déplacements mesurés en un point (extrémité de la poutre) pour des simulations non linéaires d'elasto-plasticité. Sur un problème de torsion avec modélisation par elasto-plasticité, Owen et al. [OS14] comparent des maillages hexaédriques en observant l'intégrale du couple appliqué sur le support rigide et la répartition de la contrainte de Von Mises dans un plan de coupe du modèle.

Pour des simulations biomécaniques non linéaires, Bourdin et al. [Bou+07] et Tadepalli et al. [TEC11] étudient respectivement des modèles d'un rein et d'un pied. Ils comparent les performances de maillages tétraédriques et hexaédriques en mesurant les valeurs maximales des champs de contraintes, ainsi qu'en étudiant visuellement les distributions de déplacement et de contraintes. Pour la simulation du fémur, à priori par élasticité linéaire, Ramos et al. [RS06] comparent les valeurs de déplacement, de contrainte de Von Mises et déformation en cinq points d'échantillonnage répartis dans le modèle.

En mécanique des fluides, des coefficients plus spécifiques sont étudiés comme le coefficient de trainée ou le nombre de Mach.

Pour comparer différents indicateurs de qualité pour les hexaèdres et faire des corrélations entre ceux-ci, Gao et al. [Gao+17a] utilisent notamment la valeur propre non nulle minimale λ_{min} du système linéaire comme approximation de la précision d'un maillage, en reprenant la méthodologie de Shewchuk [She02], et le conditionnement $\lambda_{max}/\lambda_{min}$ qui est lié à la vitesse de convergence des solveurs itératifs [Saa03].

2.3.3 Calcul de distances entre solutions éléments finis

Lorsque l'utilisateur a en tête une application bien précise et est intéressé par une certaine quantité, par exemple la contrainte maximale en mécanique pour vérifier des critères de sécurité, l'observation directe des quantités d'intérêt avec le raffinement du maillage est sans aucun doute l'approche la plus pragmatique et la plus utile. Mais pour des comparaisons plus générales et moins spécifiques aux applications, il peut être intéressant de calculer l'erreur commise du point de vue de normes globales comme les normes L^p ou H^1. Cette mesure permet également de se situer par rapport aux vitesses de convergence optimales, qui sont justement exprimées avec ces normes.

Lorsque la solution analytique n'est pas connue, il est possible d'utiliser une solution de référence calculée sur un maillage très fin. L'erreur d'approximation est alors approchée par la distance à la solution de référence. Dans la suite, nous nous intéressons à cette question : comment calculer une distance entre solutions éléments finis ?

Supposons que l'on veut calculer la distance $\|f - g\|$ entre deux solutions éléments finis définis sur deux maillages $\mathcal{A}$ et $\mathcal{B}$. Contrairement au cas où la solution analytique est connue, il n'est pas possible de calculer directement cette distance par quadrature car les éléments de $\mathcal{A}$ et de $\mathcal{B}$ sont différents. Nous présentons ci-dessous deux possibilités : utiliser un échantillonnage du domaine, sorte de quadrature globale, ou exprimer les deux fonctions f et g sur un maillage commun de manière à

pouvoir définir des quadratures locales.

Les deux approches de calcul de distance ci-dessous sortent légèrement du cadre de l'état de l'art car elles n'ont pas été décrites dans la littérature, mais nous estimons qu'elles sont suffisamment intuitives et naturelles pour qu'elles soient présentées dans cette section.

Échantillonnage du domaine La distance entre les champs f et g peut être approximée en évaluant leurs valeurs sur un ensemble $(\mathbf{x}_i)_{i=1..n}$ d'échantillons dans le domaine Ω. Pour la norme L^2 :

$$(\|f-g\|_2)^2 = \int_\Omega |f-g|^2 \approx \sum_{i=1..n} \frac{1}{n}(f(\mathbf{x}_i) - g(\mathbf{x}_i))^2$$

Avec l'approche par morceaux des éléments finis, la valeur de f en $\mathbf{x}_i$, échantillon contenu dans l'élément K_e, est définie par composition de l'interpolation dans l'espace de référence $\hat{f}_e$ et de l'inverse de la transformation $\mathcal{M}_e$:

$$f(\mathbf{x}_i) = f_{|K_e}(\mathbf{x}_i) = \hat{f}_e(\mathcal{M}_e^{-1}(\mathbf{x}_i))$$

Donc pour chaque échantillonnage, il faut : (1) identifier l'élément qui le contient et (2) inverser la transformation. Même s'il est possible d'accélérer l'identification des éléments avec des structures de données adéquats (*par ex.* arbre AABB), l'évaluation reste coûteuse quand elle doit être effectuée sur des millions d'échantillons, pour les deux maillages d'entrée. C'est particulièrement le cas lorsque les transformations ne sont pas linéaires, *par ex.* transformations trilinéaires des hexaèdres, car il faut utiliser des algorithmes itératifs pour l'inversion.

Dans notre expérience, cette approche est viable en dimension deux avec quelques dizaines ou centaines de milliers d'échantillons, mais le coût devient prohibitif en dimension trois lorsqu'il est nécessaire d'utiliser des (dizaines, centaines de) millions d'échantillons sur des maillages importants (centaines de milliers d'éléments ou plus).

Approches par maillage commun Une autre approche possible pour déterminer la distance entre deux champs est de projeter les deux sur le même espace et de calculer la distance sur celui-ci. Deux possibilités simples sont : (a) projeter un champ de son espace d'approximation vers celui associé à l'autre maillage ou (b) projeter les deux champs sur un troisième maillage, et espace fonctionnel associé, construit de manière à représenter les deux champs initiaux. Dans les deux cas il y a transfert de données, ou projection, d'un maillage à un autre.

Les maillages qui nous intéressent sont non-structurés et non-emboités l'un dans l'autre. Dans la littérature, le transfert de données entre de tels maillages apparait dans différents contextes et sous différentes notations. Par exemple : projection de Galerkin (i.e. interpolation au sens faible) dans le contexte des éléments mixtes [Geu+99], transfert de solution après maillage adaptatif pour des problèmes dépendant du temps [Ala16 ; BBP14] ou comme composant de solvers multi-grilles exploitant une hiérarchie de maillages non-emboités [PPM93 ; FPO97].

Dans le cas général, projeter un champ sur un autre espace d'approximation n'est pas toujours souhaitable car l'espace d'arrivée n'est pas forcément adapté pour représenter fidèlement le champ initial : les deux maillages peuvent avoir des éléments de

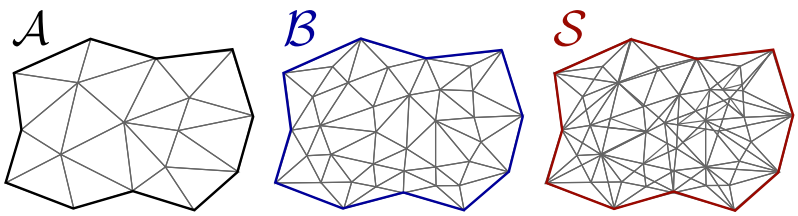


FIGURE 2.22 – Deux maillages $\mathcal{A}$ et $\mathcal{B}$ du même domaine. $\mathcal{S}$ est le *super-maillage* de $\mathcal{A}$ et $\mathcal{B}$. En 3D et avec des éléments courbes, le calcul de *super maillages* est très difficile, c'est pourquoi nous le remplaçons par une approche basée sur un échantillonnage de $\mathcal{A}$ et $\mathcal{B}$.

géométrie très différente, des polynômes d'ordres différents voire des discontinuités difficiles à projeter. Dans la littérature, ce problème a été résolu en construisant un maillage auxiliaire composé de l'intersection des deux maillages initiaux, comme illustré en 2D sur la Figure 2.22. Calculer précisément et efficacement des intersections entre maillages 3D est difficile et a principalement été étudié pour des maillages tétraédriques, sous différents noms : maillage rendez-vous [PHS98 ; SWP13], maillage raffiné commun [JH04], super-maillage [Far+09 ; MS11] ou maillage intersection [AL16]. À noter que pour conserver des performances raisonnables, il est souvent possible de calculer ces intersections localement lors de l'exécution d'un algorithme sans les stocker. L'avantage de cette approche est qu'il est possible de définir un *super-espace* [Far11] dans lequel la différence des champs $f - g$ peut être représentée exactement, puis la distance peut être calculée par quadratures. Les méthodes évoquées ont surtout été développées pour effectuer des projections préservant certaines quantités (e.g. masse, énergie) importantes dans le cadre de la simulation numérique. Cependant, ces méthodes sont compliquées à mettre en oeuvre et il est très difficile de gérer des éléments courbes (transformations non-affines) avec ces approches.

2.3.4 Techniques de visualisation de solutions éléments finis

L'approche que nous présentons dans le chapitre (§5) échantillonne les deux champs à comparer sur une grille régulière 3D, décomposée en *tranches*, c'est-à-dire en grilles régulières 2D. Évaluer les champs sur une telle grille est analogue à effectuer le rendu graphique d'une tranche d'une solution éléments finis, où il faut calculer la valeur au niveau de chaque pixel d'une grille avant d'appliquer une carte de couleurs. Dans ce chapitre, nous nous intéressons aux diverses techniques utilisées en informatique graphique pour visualiser des solutions éléments finis. Celles-ci sont intéressantes car, exploitant la puissance des cartes graphiques (GPU), elles sont très efficaces pour traiter de grandes quantités de triangles. Dans notre contexte, nous nous intéressons plus particulièrement aux méthodes permettant de visualiser des polynômes d'ordres élevés ainsi que des géométries courbes (e.g. tétraèdres quadratiques)

Historiquement, les cartes graphiques étaient uniquement optimisées pour interpoler des valeurs (correspondant à des couleurs) linéairement dans des triangles. Pour cette raison, une première approche pour visualiser des solutions éléments finis non linéaires est de construire une approximation linéaire par morceaux en raffinant le maillage. Ce raffinement peut être optimal vu que la solution éléments finis à ap-

proximer est connue [Leo+98 ; Rem+06]. Ensuite pour visualiser la tranche d'une solution volumique, on peut utiliser un algorithme d'intersection de type *marching tetrahedra* [AK91 ; TPG99]. Nous n'utilisons pas cette première approche car elle implique une étape de raffinement adaptatif et elle conduit à des maillages raffinés très gros, ce qui n'est pas pratique si les maillages en entrée sont déjà imposants.

Également conçue pour les premières générations de cartes graphiques, une approche originale [Haa+03] est de combiner des images pre-calculées (textures) représentant chacune une fonction de la base des polynômes. Cependant cette approche nécessite un prétraitement important puisqu'il faut générer les textures pour toutes les fonctions des bases polynomiales et elle n'est pas adaptée pour le cas de maillages 3D tranchés par des plans 2D.

Avec l'introduction de *shaders* programmables dans les pipelines de rendu graphique (OpenGL et DirectX), il est plus efficace d'évaluer directement les fonctions de forme des éléments finis à chaque pixel [BH04 ; NHK11]. Lors de l'utilisation d'un plan de coupe, l'élément contenant le pixel est habituellement trouvé via un lancer de rayons et les coordonnées de référence (nécessaires à l'évaluation des fonctions de forme) sont obtenues en inversant les transformations avec une méthode itérative de type Newton-Raphson. Cette approche est coûteuse lorsque l'on considère des dizaines de millions d'échantillons (grille 3D complète), particulièrement quand les transformations ne sont pas affines. Afin d'éviter les inversions de transformation, il a été proposé d'approximer les fonctions de forme le long des rayons en utilisant des projections L^2 [NK06] sur des polynômes 1D ou encore l'alternative de Üffinger et al. [ÜFE10].

Dans notre cas (§5), nous souhaitons seulement échantillonner les champs sur des tranches planes, par conséquent nous n'avons pas besoin d'une approche aussi flexible que le lancer de rayons et nous pouvons nous contenter de l'algorithme d'intersection *marching tetrahedra* [AK91] et de la rastérisation fournie par OpenGL pour interpoler les coordonnées de référence. Cette restriction permet de simplifier grandement les algorithmes, et les implémentations associées, par rapport à des outils de visualisation plus génériques.

Deuxième partie

Contributions

---

3 Génération de maillages hex-dominants **74**
3.1 Maillage hex-dominant robuste par extraction de cavités 74
3.2 Transformation des singularités volumiques en singularités linéiques 89

4 Méthode des éléments finis et maillages hex-dominants **98**
4.1 Comparaisons de primitives sur des problèmes simples 2D 98
4.2 Simulation éléments finis sur maillages hexa-tétraédriques 106
4.3 Comparaisons de maillages sur des domaines compliqués 115

5 Calcul de distances entre solutions éléments finis **123**
5.1 Approximation de la distance entre champs 124
5.2 Implémentation GPU avec OpenGL 131
5.3 Validation et discussion de la méthode 135

Chapitre 3

Génération de maillages hex-dominants

Ce chapitre est consacré à la génération de maillages hex-dominants à partir de paramétrisations dont les gradients sont alignés avec un champ de directions. Nous utilisons les techniques connues de construction de champs de directions et de paramétrisations, qui sont détaillées dans l'état de l'art (§2.1.2).

La première section (§3.1) propose une méthode robuste de maillage hex-dominant lorsque la paramétrisation n'est pas parfaite, ce qui est très souvent le cas sur des géométries compliquées. Elle est basée sur l'article

> *Nicolas* Ray*, Dmitry* Sokolov*, Maxence* Reberol*, Franck* Ledoux *et Bruno* Lévy*. « Hexahedral Meshing : Mind the Gap ! » 2017*

auquel j'ai collaboré mais dont je ne suis pas le principal contributeur.

La seconde section (§3.2) explore deux techniques pour transformer les singularités volumiques du champ de directions, obtenus lorsqu'on utilise un échantillonnage sur les sommets du maillage, en singularités linéiques. Cette transformation permet de mailler de manière entièrement hexaédrique les alentours de certaines singularités tout en profitant des avantages d'un échantillonnage par sommet du champ de directions.

3.1 Maillage hex-dominant robuste par extraction de cavités

Avec les approches de maillage hexaédrique par paramétrisation, comme *CubeCover* ou *PGP*, la paramétrisation est souvent localement défectueuse sur les modèles compliqués : non définie, volumes nuls ou négatifs dans l'espace paramétrique, ou possédant trop de distorsion. Ces régions où l'on ne peut pas extraire directement des hexaèdres sont appelées **cavités** dans la suite. Elles sont généralement maillées avec des tétraèdres, afin de finalement produire un maillage hex-dominant. La construction et le maillage des cavités ne sont généralement pas détaillés dans la littérature, or cette étape n'est pas complètement triviale.

Nous présentons une méthode qui permet de générer de manière *robuste* des surfaces fermées, sans auto-intersections, contenant les cavités (§3.1.1). Par robuste, nous entendons que l'extraction de cavités fonctionne sur tous les modèles et permet

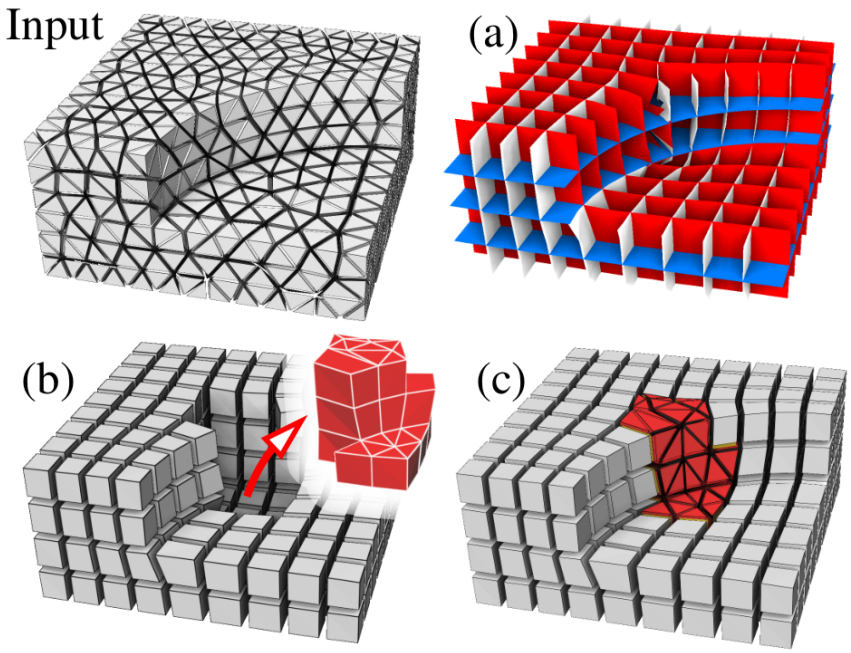


FIGURE 3.1 – Étapes de maillage hex-dominant à partir d'une paramétrisation. (a) Iso-valeurs entières de la paramétrisation. (b) Hexaèdres extraits de la paramétrisation et maillage quad-dominant de la cavité. (c) Maillage hex-dominant final après maillage de la cavité.

toujours d'en déduire un maillage hex-dominant, la solution la moins satisfaisante étant de reproduire le maillage d'entrée entièrement tétraédrique. Dans un second temps (§3.1.2), nous décrivons différentes manières de mailler l'intérieur des cavités, en essayant d'insérer un maximum d'hexaèdres.

3.1.1 Extraction des hexaèdres et du bord des cavités

La surface entourant une cavité est un maillage quad-dominant : les quadrilatères sont les faces des hexaèdres adjacents à la cavité et les triangles sont générés à partir du bord du maillage tétraédrique initial (support de la paramétrisation). Un exemple d'extraction et de maillage d'une cavité est illustré sur la Figure 3.1.

Conceptuellement, le volume des cavités C peut être exprimé comme étant la différence booléenne entre le volume tétraédrique initial T et le maillage hexaédrique H extrait de la paramétrisation :

$$C = T - H$$

Géométriquement, cette différence n'a pas vraiment de sens car les deux volumes T et H ont des bords différents. C'est la principale difficulté du problème : le bord des hexaèdres, noté ∂H, n'est pas compatible avec le bord triangulaire ∂T initial. Cette incompatibilité est illustrée sur la gauche de la Figure 3.2. Afin d'y remédier, nous proposons de générer un maillage quad-dominant du bord de l'objet, noté ∂Q_d (droite de la Figure 3.2), compatible avec les hexaèdres extraits. Le volume Q_d, défini comme l'intérieur de la surface fermée ∂Q_d, permet de définir géométriquement le volume des cavités : $C = Q_d - H$.

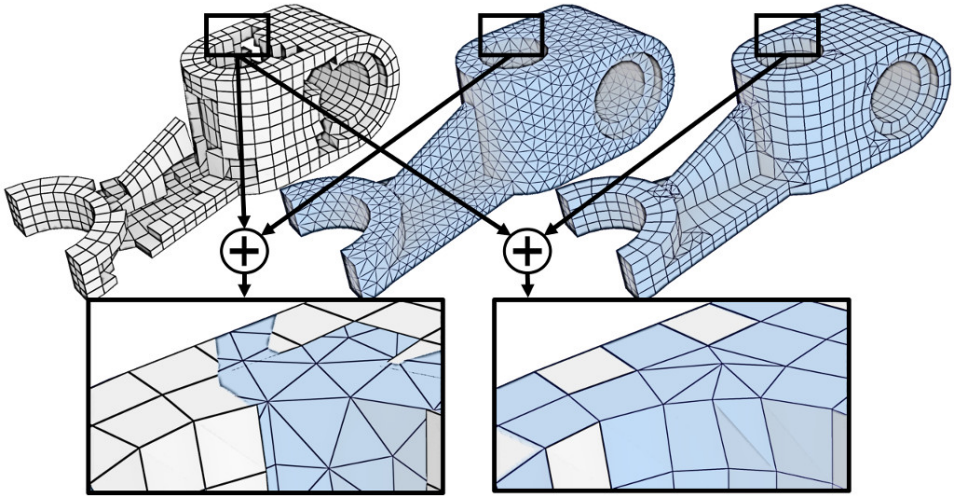


FIGURE 3.2 – Incompatibilité initiale entre les hexaèdres extraits et les bords du maillage tétraédrique. *Gauche :* Hexaèdres extraits de la paramétrisation. *Milieu :* Bord triangulaire du maillage d'entrée. *Droite :* Maillage quad-dominant conforme.

Pour éviter de travailler avec des volumes, dont les intersections sont compliquées à gérer, on peut se ramener à un problème surfacique en remarquant que le bord ∂C des cavités peut être obtenu en retirant les portions communes de ∂C et ∂Q_d à l'union des deux :

$$\partial C = (\partial Q_d \cup \partial H) - (\partial Q_d \cap \partial H)$$

Il faut bien faire attention que cette description n'a de sens que parce que ∂Q_d et H sont compatibles, c'est-à-dire que toute face de ∂H intersectant ∂Q_d forme soit un sommet, soit une arête, soit un quad de ∂Q_d. De plus, comme les surfaces sont compatibles, la différence ci-dessus sera purement combinatoire. Ainsi, toute la difficulté de l'extraction du bord des cavités est reportée sur la génération du maillage quad-dominant ∂Q_d compatible avec H (§3.1.1.1). Avec cette approche, l'extraction des hexaèdres H (§3.1.1.2) et des bords des cavités (§3.1.1.3) sont des étapes simples.

Les différentes étapes et maillages mis en jeu sont :

Entrées du problème d'extraction

- Un maillage tétraédrique T muni d'une paramétrisation u, v, w, linéaire par éléments (échantillonnage sur les coins des tétraèdres).
- Une liste de valeurs booléennes, une par face des tétraèdres, qui indique si la paramétrisation est valide ou non. Cette valeur peut être utilisée pour exclure les régions où la paramétrisation est invalide ou trop déformée.
- Une distance de Hausdorff maximale ϵ à respecter entre le bord initial ∂T de l'objet et le bord $\partial\Omega_h$ du maillage hex-dominant Ω_h généré.

Objets intermédiaires

- Un maillage quad-dominant du bord ∂Q_d compatible avec les iso-valeurs entières de la paramétrisation sur le bord et respectant la distance de Hausdorff.

Sorties

— Un maillage H composé d'hexaèdres, extraits dans les régions valides de la paramétrisation.

— Un maillage quad-dominant ∂C délimitant les cavités, conforme avec les hexaèdres de H.

Robustesse. Tous les maillages surfaciques sont des surfaces fermées sans auto-intersections, de manière à ce que les volumes intérieurs puissent être maillés par triangulation de Delaunay contrainte [GHS90 ; Si15b]. Cette garantie est obtenue en partant d'un maillage valide et en s'assurant que chaque opération conserve la validité, quitte à revenir en arrière dans certaines situations (stratégie de *roll-back*).

Distance de Hausdorff entre surfaces. La semi-distance de Hausdorff, ou distance de Hausdorff orientée, d'un objet A à un objet B est définie par :

$$d(A,B) = \sup_{x\in A}(\inf_{y\in B} \|x-y\|_2)$$

La version symétrique de cette distance est :

$$d_s(A,B) = \max(\ d(A,B),\ d(B,A)\ )$$

Nous utilisons cette dernière dans la suite du chapitre. Cette mesure est employée couramment pour contrôler l'erreur introduite par les algorithmes de simplification des surfaces triangulées [CRS98 ; ASE02].

3.1.1.1 Génération du maillage quad-dominant ∂Q_d du bord ∂T de l'objet

Le maillage quad-dominant ∂Q_d doit :

— être compatible avec les hexaèdres potentiels de la paramétrisation. La restriction de la paramétrisation (définie sur T) sur le bord ∂T est une paramétrisation surfacique, dont les iso-valeurs définissent des quadrilatères dans les régions valides. Par construction, ces quadrilatères sont des bords potentiels de l'ensemble des hexaèdres H (qui sera extrait plus tard) ;

— former une surface fermée sans auto-intersections. Comme le maillage est quad-dominant, cette condition est vérifiée pour tous les maillages triangulaires possibles obtenus en décomposant chaque quad en deux triangles ;

— respecter la distance de Hausdorff ϵ prescrite. Cette condition est vérifiée sur chaque décomposition possible des quads en triangles.

Pour générer ∂Q_d, nous procédons en quatre étapes successives, illustrées sur la Figure 3.3 :

1. Extraction du bord triangulaire ∂T et de sa paramétrisation surfacique (Figure 3.3.1.).
2. Création d'arêtes, à partir des iso-valeurs, qui sont potentiellement des arêtes d'hexaèdres (Figure 3.3.2.).

3. Création de patchs de triangles qui correspondent potentiellement à des quads (Figure 3.3.3.).
4. Regroupement des triangles d'un même patch en un quad lorsque c'est possible. Cette étape s'assure que le critère de Hausdorff est respecté. (Figure 3.3.4.)

Nous détaillons ci-dessous les étapes de génération de ∂Q_d. À noter que tous les calculs sont effectués avec des nombres flottants à précision double.

1. Extraction de la paramétrisation 2D. La restriction 2D de la paramétrisation sur un triangle est considérée valide si la paramétrisation 3D est valide sur le tétraèdre et si une des trois coordonnées est entière sur tout le triangle (dans la direction normale au triangle). Dans ce cas, la paramétrisation surfacique est définie par les valeurs des champs scalaires associés aux deux autres directions.

Sur la Figure 3.3.Input., les iso-valeurs entières constantes sont représentées en rouge et les triangules bleus correspondent à des tétraèdres pour lesquels la paramétrisation n'est pas valide. La paramétrisation valide est représentée en rouge sur la Figure 3.3.1., et le triangle pointé par la flèche correspond à un triangle rejeté car aucun des trois champs scalaires n'était entier sur celui-ci.

La paramétrisation surfacique hérite du caractère, ou non, *grille-compatibible* de la paramétrisation volumique.

2. Création d'arêtes. L'objectif est de créer des arêtes dans le maillage quad-dominant ∂Q_d qui correspondent à de potentielles arêtes du bord ∂H du (futur) maillage hexaédrique H. Celles-ci sont forcément associées à des iso-valeurs entières de la paramétrisation. Pour faire apparaitre les arêtes dans ∂Q_d, on subdivise les triangles de ∂Q_d qui sont traversés par une iso-valeur entière. Cette étape est réalisée en deux temps : division des arêtes (création de sommets) puis division des faces par des arêtes qui relient les nouveaux sommets ayant une même iso-valeur.

2.a. Création de sommets. En pratique, pour chaque arête de ∂Q_d, on choisit un polygone qui la contient. On construit ensuite un nouveau sommet pour toutes les valeurs (paramétriques) entières incluses dans l'arête. Les positions géométriques sont calculées par interpolation linéaire. Les sommets sont ensuite insérés dans l'arête par ordre croissant de leur distance avec l'une des extrémités. La création de sommets est illustrée sur la gauche de la Figure 3.3.2 (sommets en bleus). Facultativement, on peut arrondir les valeurs de paramétrisation des sommets qui sont à 10^{-10} d'une valeur entière. Ceci permet de rattraper certaines imprécisions dues à des erreurs numériques dans le calcul initial des paramétrisations.

2.b. Division des faces. Pour chaque paire de nouveaux sommets ayant une iso-valeur commune, une nouvelle arête est créée. Cette arête peut être traversée par des iso-valeurs de l'autre champ scalaire. Il faut donc lui appliquer la procédure 2.a. afin de la diviser également. Ce processus (2.a.-2.b.) est appliqué itérativement jusqu'à ce qu'il n'y ait plus de sommets ou d'arêtes à créer. Afin que l'algorithme termine, il faut s'assurer que chaque division de face vérifie deux conditions : (1) la nouvelle arête ne connecte pas deux sommets consécutifs du polygone à diviser (pour ne pas créer un polygone à deux sommets), et (2) dans l'espace paramétrique, les deux ensembles de sommets, associés à chaque côté de l'arête de coupe, sont situés dans les deux demi-plans opposés créés par la coupe (à part les sommets de l'arête de coupe qui sont à cheval). Les faces résultantes (polygonales) sont finalement triangulées. Le

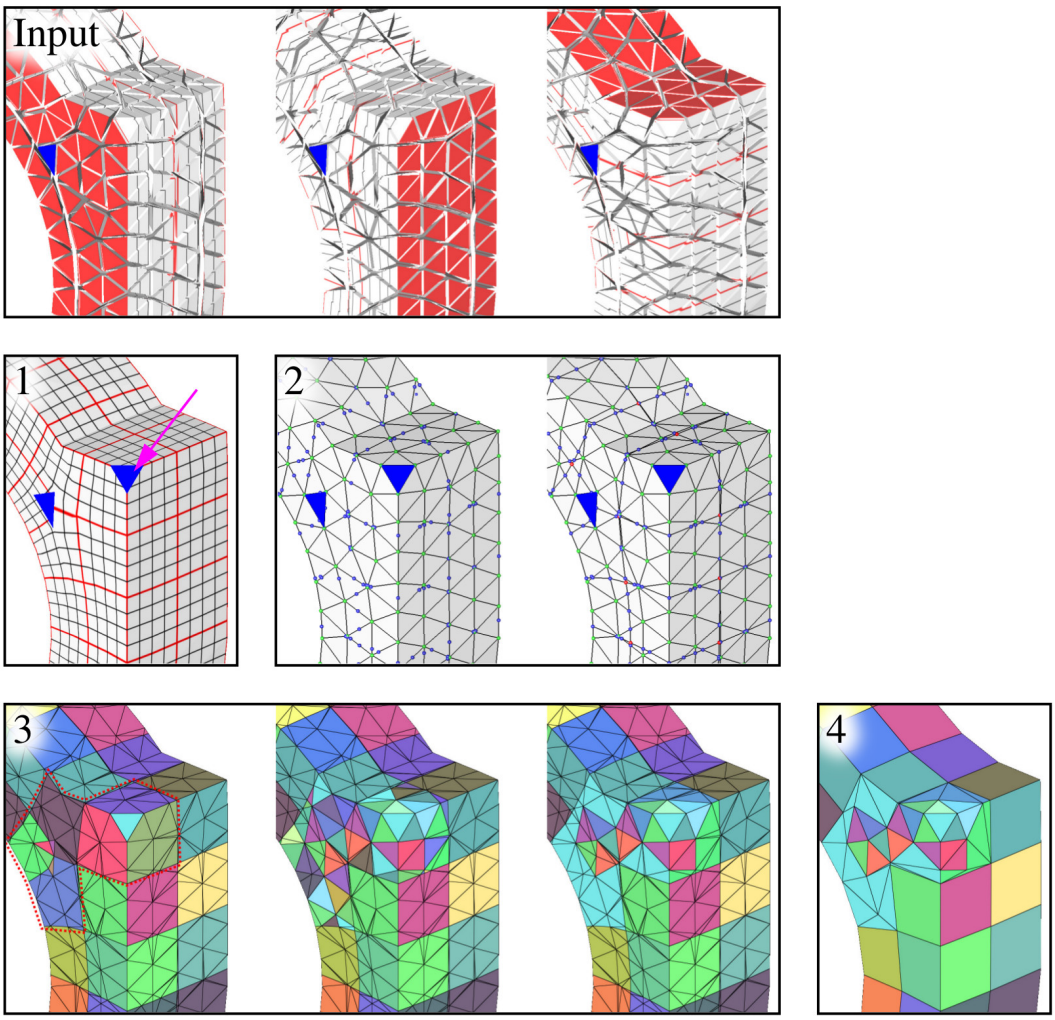


FIGURE 3.3 – Étapes de génération d'un maillage quad-dominant du bord. **(1)** Restriction surfacique de la paramétrisation 3D. **(2)** Création de sommets et arêtes correspondants aux iso-valeurs entières. **(3)** Regroupement des faces en patchs en fonction des iso-valeurs et conservation des patchs correspondants à des quads. **(4)** Simplification du maillage, c'est-à-dire transformation des patchs de quads en éléments quads et suppression des arêtes ajoutées dans les patchs non conservés. Le résultat est un maillage hex-dominant conforme.

résultat est illustré sur la droite de la Figure 3.3.

L'algorithme de division des faces termine toujours car pour une face donnée, un nombre fini de sommets correspondant à des iso-valeurs entières est ajouté. Lorsqu'une nouvelle arête relie deux de ces iso-valeurs, la face est divisée en deux. Chacune des deux parties contient un nombre strictement inférieur de sommets grâce aux conditions (1) et (2). Ensuite, ces triangles sont divisés si besoin, puis ses éventuelles subdivisions, etc. Comme le nombre de sommets dans les sous-triangles est strictement décroissant, la création de sommets l'est aussi et elle converge vers zéro.

Robustesse et retour en arrière. Avant la division des faces, le maillage ∂Q_d ne contenait pas d'auto-intersections. À ce stade, nous vérifions que tous les nouveaux triangles créés par divisions n'introduisent pas d'auto-intersections, ce qui peut très bien arriver, car toutes les opérations précédentes ont été menées avec des nombres flottants. Si une auto-intersection est détectée, le triangle d'origine est marqué comme étant non valide et le maillage triangulaire initial est restauré, *c.à-d.* $\partial Q_d = \partial T$, puis le procédé complet relancé, jusqu'à obtenir un maillage quad-dominant valide.

3. Création de patchs. L'objectif de cette étape est de regrouper les triangles générés par l'étape précédente, notamment pour faire ressortir les quads qui pourraient être compatibles avec le maillage hexaédrique. Pour commencer, un patch indépendant est créé pour chaque triangle de ∂Q_d. Ensuite nous appliquons les étapes suivantes :

— *identification des quads :* si deux triangles adjacents ne sont pas séparés par une arête correspondante à une valeur entière, alors leurs deux patchs sont fusionnés (gauche de la Figure 3.3.3). On conserve les patchs dont les extrémités sont composées de quatre lignes iso-valeurs et quatre coins intersections d'iso-valeurs (milieu de la Figure 3.3.3). Il est également possible de rejeter les quads dont la géométrie est trop déformée ;

— *identification des triangles :* les patchs de deux triangles adjacents, qui n'appartiennent pas à des patchs de quads, sont fusionnés si les deux triangles proviennent du même triangle initial de ∂T ;

— *regroupement des triangles :* les patchs de deux triangles adjacents sont fusionnés s'ils ne sont pas séparés par une iso-valeur et si un des triangles est adjacent à un patch de quad. (droite de la Figure 3.3.3). Ce regroupement de patchs est utile pour la future simplification du maillage.

4. Simplification du maillage. L'objectif de cette étape est de remplacer les patchs par des faces dans le maillage ∂Q_d. Pour chaque sommet qui n'est pas un coin de quadrilatère :

— si il touche un unique patch, le sommet est supprimé ainsi que les triangles le contenant, puis le trou est triangulé (*par ex.* avec l'algorithme de triangulation de poids minimum) ;

— si il touche deux patchs, le sommet est supprimé ainsi que les triangles le contenant, puis le trou est triangulé avec la contrainte qu'une des arêtes doit suivre la frontière entre les deux patchs.

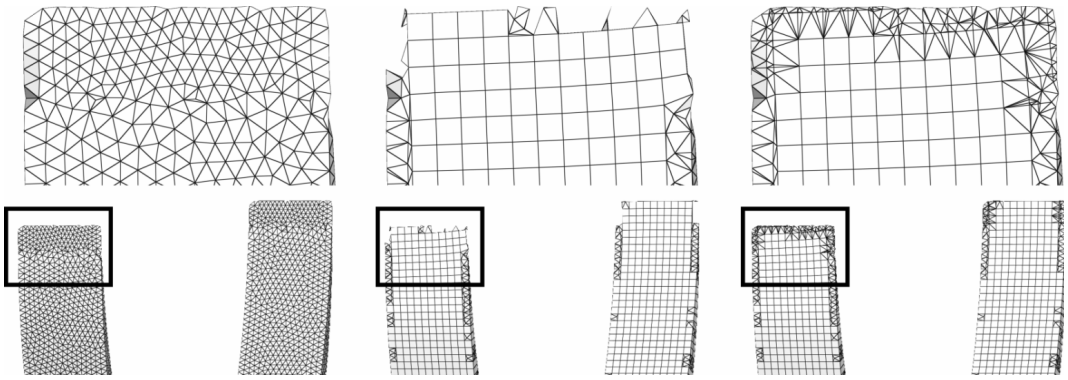


FIGURE 3.4 – Le maillage quad-dominant est calculé à partir de la surface d'entrée à gauche, avec (milieu) et sans (droite) contrôle de la distance de Hausdorff symétrique.

La dernière étape de *3. Création de patchs* (*regroupement des patchs de triangles*) permet de faire apparaitre la seconde configuration. En pratique, elle permet de modifier la géométrie de la surface afin de faire fusionner (géométriquement et localement) la surface de quads et la surface de triangles le long des interfaces.

A l'issue de ce processus, la plupart des patchs de quads sont représentés par deux triangles et la plupart des triangles originaux de ∂T qui ne sont pas dans des patchs de quads sont représentés par un triangle unique. Les patchs des quads composés de deux triangles sont fusionnés en quads.

Vérification et retour en arrière. Au cours de la simplification, les indices des sommets supprimés sont stockés sur les nouvelles faces. Ensuite, nous vérifions qu'il n'y a pas d'auto-intersections dans le maillage surfacique généré, en testant les deux décompositions possibles pour les quads. S'il y a intersection de deux triangles, la surface avant simplification est restaurée et les sommets associés aux triangles incriminés sont bloqués, puis la simplification recommence.

Contrôle de la distance de Hausdorff. Si un triangle de ∂Q_d est plus loin que ϵ de la surface d'entrée ∂T, alors le sommet associé à ce triangle est bloqué et la simplification est relancée. Symétriquement, si un triangle de ∂T est trop loin de ∂Q_d, alors les trois sommets associés au triangle de ∂T dans ∂Q_d sont bloqués, de manière à ce que ceux-ci ne soient pas supprimés lors de la nouvelle simplification. En pratique nous ne calculons pas exactement la distance de Hausdorff, mais nous échantillonnons les triangles de sorte que chaque sommet du triangle soit à moins de $\epsilon/2$ d'un échantillon. Si la distance entre un échantillon et l'autre surface est supérieure à $\epsilon/2$, la distance de Hausdorff peut être supérieure à ϵ et le triangle est considéré comme étant trop loin. Le résultat de ce contrôle de la distance est illustré sur la Figure 3.4.

3.1.1.2 Extraction du maillage hexaédrique H

Nous adoptons la même technique d'extraction que [NRP11] : pour chaque tétraèdre, nous calculons l'intersection avec la grille unitaire dans le domaine paramétrique. Le résultat de cette étape peut comporter des configurations non désirables. Pour les supprimer, nous proposons les filtres suivants :

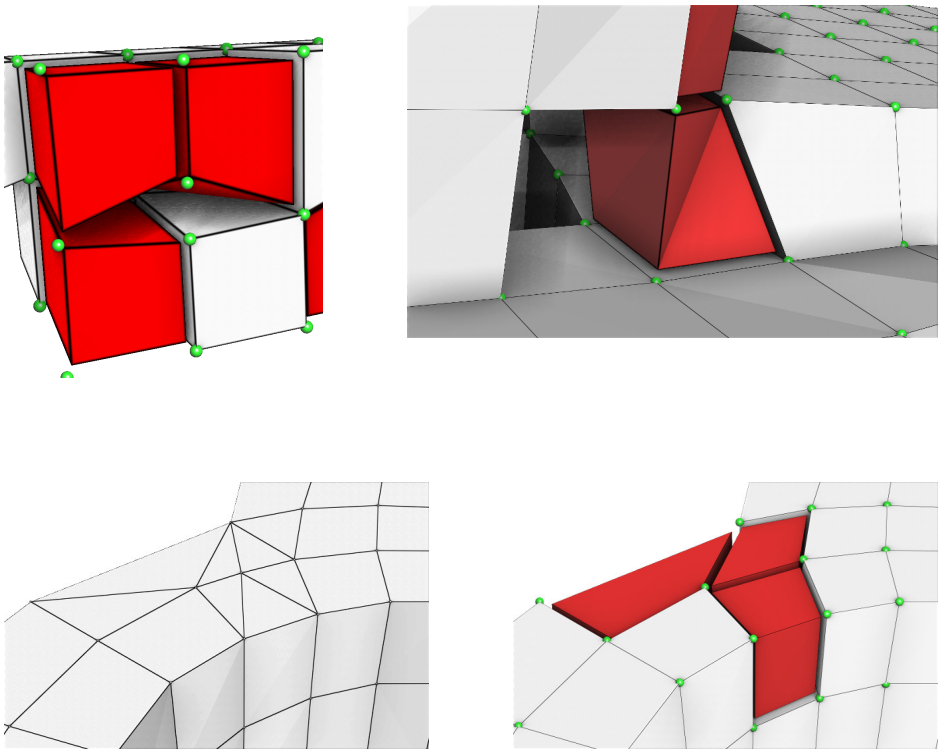


FIGURE 3.5 – Hexaèdres non conservés. *Haut gauche :* deux hexaèdres partagent trois sommets. *Haut droit :* hexaèdre de trop mauvaise qualité. *Bas :* lorsque le maillage quad-dominant (gauche) n'est pas compatible avec les hexaèdres extraits (droite), les hexaèdres sont rejetés.

— les sommets voisins plus proches qu'une certaine distance (*par ex.* 10^{-8}) sont fusionnés ;

— les hexaèdres dont l'intersection avec un autre hexaèdre n'est pas un sommet, une arête ou un quad sont supprimés. On considère qu'un quad intersecte une autre face si l'un des triangles de ses deux décompositions possibles l'intersecte ;

— les hexaèdres dont l'intersection avec le maillage quad-dominant ∂Q_d n'est pas un sommet, une arête ou un quad sont supprimés. Ce filtre assure que les maillages ∂Q_d et H sont compatibles ;

— *facultatif :* les hexaèdres dont la géométrie est trop déformée, *par ex.* dont le jacobien mis à l'échelle est inférieur à 0.5, sont supprimés.

La configuration où deux hexaèdres partagent trois sommets (gauche de la Figure 3.5) est rejetée par le second filtre car il y a une décomposition des quads en triangles, où l'intersection des deux éléments est un triangle unique, ce qui n'est ni une arête ni un quad. Le filtre sur la qualité des éléments est illustré sur la droite de la Figure 3.5 qui montre un hexaèdre trop déformé. Le bas de la Figure 3.5 illustre un cas où les quads de la surface ne sont pas compatibles avec les bords des hexaèdres. Conformément au troisième filtre, ces hexaèdres sont supprimés.

Remarque. Théoriquement, il est possible d'utiliser une méthode d'extraction plus robuste comme *HexEx* [LBK16], cependant il faut prendre d'importantes précautions

car *HexEx* modifie la géométrie. Par conséquent, le maillage hexaédrique extrait de la paramétrisation n'est plus compatible avec la paramétrisation surfacique que nous avons utilisée.

3.1.1.3 Extraction des bords des cavités ∂C

Le maillage quad-dominant ∂Q_d, compatible avec les hexaèdres H, permet d'exprimer le bord des cavités sous la forme :

$$\partial C = (\partial Q_d \cup \partial H) - (\partial Q_d \cap \partial H)$$

En pratique, il suffit d'ajouter toutes les faces de ∂Q_d et de ∂H dans un même maillage et de supprimer les faces dupliquées. Généralement il y a plusieurs cavités, la surface ∂C est alors composée d'un ensemble de plusieurs surfaces fermées.

Nous disposons maintenant d'un ensemble d'hexaèdres H et d'un maillage quad-dominant Q_d du bord ∂C des cavités. Il ne reste plus qu'à remplir les cavités pour former un maillage hex-dominant Ω_h du maillage tétraédrique initial T.

3.1.2 Maillage des cavités

L'objectif est maintenant de produire un maillage hex-dominant de chaque partie fermée connexe des cavités C. Il s'agit d'un problème de maillage hex-dominant **partiellement contraint** : il faut absolument que les éléments intérieurs soient conformes avec les quadrilatères du maillage quad-dominant ∂C.

Pour générer un maillage hexa-tétraédrique non-conforme (§2.2.2.2, p. 66), nous autorisons la jonction où deux tétraèdres intérieurs se connectent sur une face quadrilatérale d'hexaèdre. Dans ce cas, l'approche la plus simple consiste à trianguler ∂C (division des quads en deux triangles) et à tétraédriser le volume avec un mailleur par triangulation de Delaunay contrainte comme *TetGen* [Si15b], qui respecte les triangles du bord.

À partir d'un maillage tétraédrique de la cavité, il est possible d'augmenter la proportion d'hexaèdres dans le maillage hex-dominant en **recombinant** des tétraèdres en hexaèdres. Les méthodes de recombinaison, principalement développées pour les approches indirectes, ont été évoquées dans l'état de l'art (§2.1.2.2, p. 48). Dans les expériences numériques qui suivent, nous utilisons l'algorithme de recombinaison de [Bau+14].

Une autre approche est de considérer le problème de maillage de la cavité comme un problème de maillage hexaédrique contraint classique et d'utiliser les approches existantes (§2.1.1, p. 31). La première étape est de mailler ∂Q_d avec un maillage entièrement quadrilatéral, ce qui peut parfois être réalisé avec la technique frontale de pavage [BS91 ; WK97]. À partir des quads du domaine, il est possible de générer des *feuilles* intérieures avec l'algorithme de **whisker-weaving** [TBM96]. Cette approche permet d'augmenter de manière très importante la proportion importante d'hexaèdres mais elle n'est pas aussi robuste que l'approche par recombinaison et échoue sur certains modèles, lorsqu'il n'est pas possible de générer un bord quadrilatéral ou de mailler avec l'algorithme *whisker-weaving* .

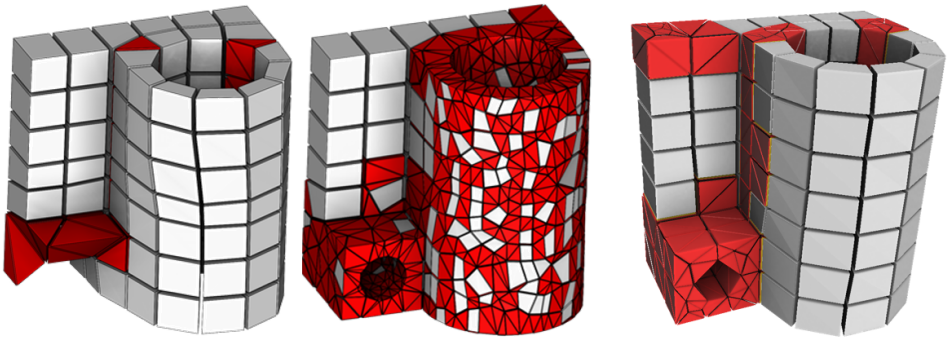


FIGURE 3.6 – Le bord du maillage hex-dominant produit par [Sok+16] dépend d'un paramètre global qui peut empêcher de capturer certaines parties du modèle (gauche) ou introduire trop de tétraèdres inutilement (milieu). Notre méthode ne souffre pas de ce défaut (droite).

3.1.3 Discussion et expériences numériques

Nous commençons par comparer qualitativement nos résultats avec certains travaux précédents de la littérature (§3.1.3.1). Ensuite nous analysons différents résultats, dont des statistiques globales, extraites de l'application de notre approche à une base de données de plus de 200 modèles (§3.1.3.2). Les résultats détaillés sont disponibles dans les fichiers accompagnant l'article [Ray+17].

3.1.3.1 Comparaison avec des travaux précédents

L'approche présentée s'inscrit définitivement dans le cadre des approches de maillage hexaédrique par champs de directions, et plus particulièrement parmi celles qui utilisent une paramétrisation, *c.à-d.* une forme d'intégration du champ de directions par des champs scalaires. L'avantage de la paramétrisation est qu'elle permet d'espérer capturer la topologie du maillage hexaédrique, notamment les singularités, car il s'agit d'un problème global et non d'un ensemble de décisions locales.

Comme le point d'entrée est une paramétrisation, qui peut être localement défectueuse, notre approche est assez flexible et peut aussi bien se brancher sur la paramétrisation produite par *CuberCover* [NRP11] que celle produite par *paramétrisation globale périodique (PGP)* [Sok+16].

Par rapport à [Sok+16] : la surface d'entrée du modèle est bien mieux respectée, via la construction du maillage intermédiaire quad-dominant qui respecte une distance de Hausdorff prescrite. La Figure 3.6 illustre bien la différence entre les deux approches.

Par rapport à *CubeCover* [NRP11] et ses extensions, dont l'extraction robuste d'hexaèdres *HexEx* [LBK16], notre contrôle de la distance Hausdorff permet de ne pas se faire piéger des paramétrisations trompeuses, notamment en cas de présence de rampes [22]. Le cas est illustré sur la gauche de la Figure 3.7. Notre approche, droite de la Figure 3.7, introduit des tétraèdres là où la paramétrisation ne permet pas de respecter les détails géométriques de la surface d'entrée.

22. surface qui se sépare en deux, via deux arêtes vives formant un *triangle* (cf Figure 3.7)

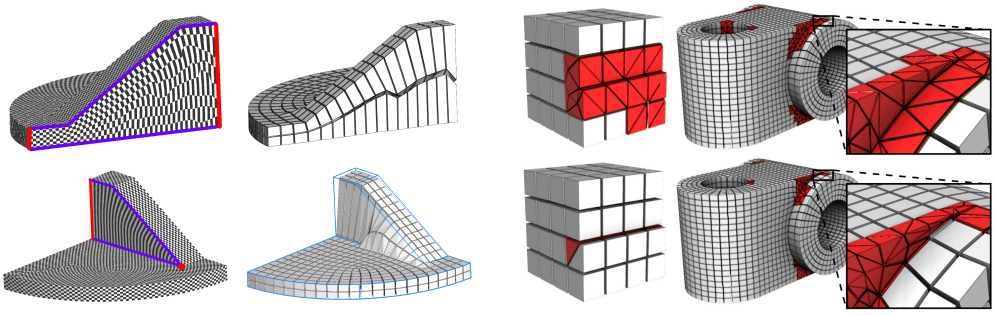


FIGURE 3.7 – Échecs de la paramétrisation en présence de rampes. *Gauche :* la paramétrisation via *CubeCover* impose une même iso-valeur pour la surface plane (quart de cercle), la partie supérieure de la rampe et le plateau supérieur. Cette paramétrisation est trompeuse car elle ne correspond pas à un maillage hexaédrique satisfaisant du modèle. *HexEx* arrive tout de même à extraire des hexaèdres (droite) mais ceux-ci sont très déformés et la géométrie d'entrée n'est pas respectée. *Droite :* gestion d'une rampe et respect des détails géométriques avec notre approche. La figure illustre également la différence entre le remplissage de cavité par tétraédrisation (haut) et par pavage et *whisker-weaving* (bas).

Concernant la paramétrisation, notre implémentation peut utiliser *PGP*, ou *CubeCover* sans gestion des singularités linéiques. En effet, contrairement à l'article original [NRP11] qui utilise des *frames* constantes par tétraèdre, nous discrétisons le champ de directions avec une *frame* par sommet. Par conséquent, les singularités sont volumiques dans notre paramétrisation, et les tétraèdres singuliers sont marqués comme non valides en entrée de notre méthode d'extraction. Cette différence explique que nous ayons un grand nombre de tétraèdres autour des singularités et que nous ne pouvons pas produire de maillages entièrement hexaédriques dans certaines situations où [NRP11 ; Li+12] y arrivent. Ceci dit, avec des paramétrisations valides sur tout le domaine, notre approche extrait également un maillage entièrement hexaédrique (puisque nous utilisons la même technique d'extraction hexaédrique que [NRP11]).

3.1.3.2 Application à une base de données de modèles

Les résultats ci-dessus sont obtenus à partir de l'application de notre approche à plus de 200 modèles. Il y a aussi bien des modèles très simples (prisme, tétraèdre, sphère, cylindre) que des modèles compliqués (pièces mécaniques avec des centaines de détails géométriques). Dans l'ensemble, les modèles correspondent surtout à des pièces mécaniques conçues par CAO [23], composées de nombreuses arêtes vives et il y a très peu de modèles à bords lisses (comme le lapin de Stanford ou l'Armadillo). Cette distinction est importante car le maillage hexaédrique des objets à nombreuses arêtes vives est significativement plus difficile que celui des objets à bords lisses, pour lesquels les méthodes *PolyCube* et *CubeCover* ont beaucoup de succès.

23. conception assistée par ordinateur

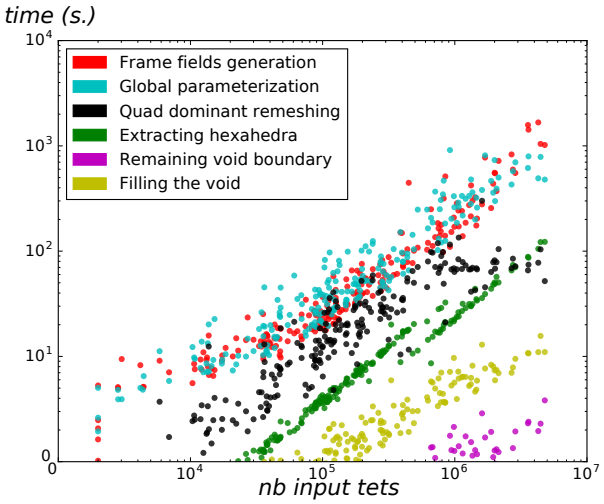


FIGURE 3.8 – Temps de calcul (en secondes) de chaque étape du processus de maillage, en fonction du nombre de tétraèdres dans le maillage d'entrée (échelle log-log).

Temps de calculs. Les temps de calcul, regroupés sur la Figure 3.8, montrent que la procédure complète, du maillage tétraédrique au maillage hex-dominant final, prend en général quelques minutes pour des maillages d'entrée composés de centaines de milliers de tétraèdres et jusqu'à une demi-heure pour ceux composés de dizaines de millions de tétraèdres.

Il est intéressant de remarquer que les étapes les plus coûteuses sont la génération du champ de directions et la paramétrisation du volume. Les étapes, présentées dans ce chapitre, de maillage quad-dominant du bord, d'extractions d'hexaèdres et des cavités, de remplissage de ces dernières, sont beaucoup plus rapides. Ceci s'explique principalement par le fait que les deux premières étapes mettent en jeu des systèmes linéaires et mixtes réels-entiers, ou une succession de systèmes, dont la résolution avec des solveurs itératifs peut prendre un temps important, particulièrement pour les gros systèmes.

Proportion volumique d'hexaèdres. La proportion de volume d'hexaèdres extraits, pour différents champs de directions et différentes paramétrisations, est illustrée sur la Figure 3.9. Ces valeurs ne tiennent pas compte de l'étape de remplissage hex-dominant des cavités. Nous constatons que pour les approches les plus performantes, *c.à-d.* champs de directions lissés et *CuberCover* ou *PGP* avec correction, la grande majorité des modèles a plus de $80-90\%$ de son volume couvert par des hexaèdres. Cette proportion pourrait encore être augmentée en gérant mieux les singularités, par exemple en utilisant des singularités linéiques comme [NRP11] ou avec des techniques performantes de remplissage des cavités qui peuvent exploiter les connaissances sur le graphe de singularité.

Remplissage des cavités. Avec notre implémentation de *CuberCover* et un champ de directions lissé, le tableau 3.1 montre les gains obtenus sur la proportion d'hexaèdres en utilisant la méthode de recombinaison des tétraèdres en hexaèdres [Bau+14],

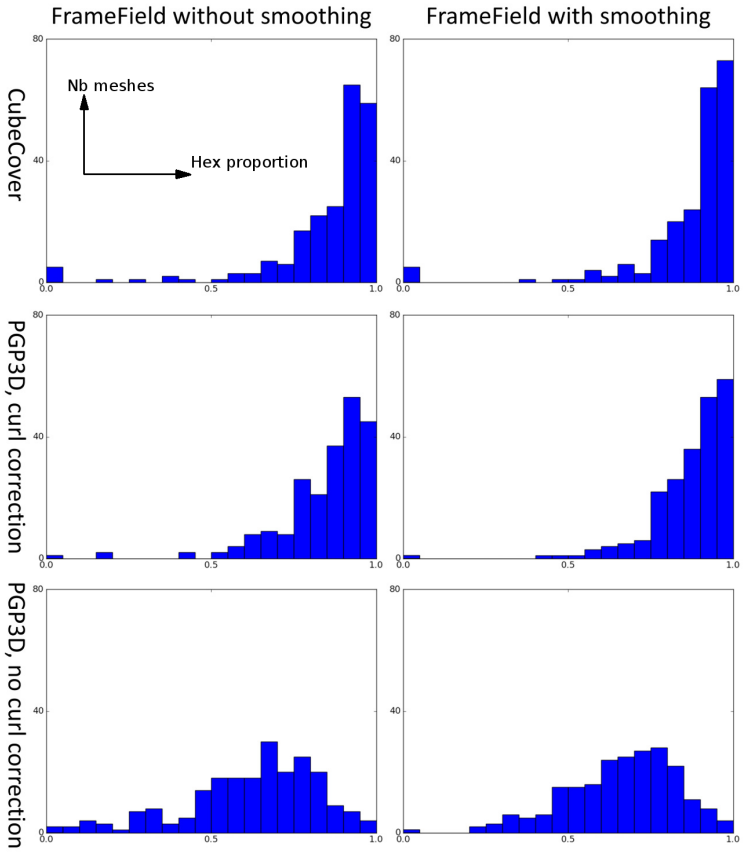


FIGURE 3.9 – Proportion du volume rempli par des hexaèdres pour différents champs de directions (avec et sans lissage) et différentes paramétrisations (*PGP* sans et avec correction du rotationnel, *CubeCober* sans singularités linéiques).

par rapport à une simple tétraédrisation des cavités. Nous montrons également que ces résultats sont globalement meilleurs que ceux obtenus dans l'article [Sok+16].

Il est possible d'obtenir une proportion volumique d'hexaèdres encore supérieure en utilisant l'approche *pavage et whisker-weaving* évoquée précédemment. Les résultats sont illustrés sur la Figure 3.10. Cependant, cette approche est moins robuste et de nombreux cas d'échecs ont été observés sur la base de données.

3.1.3.3 Conclusion

Nous pensons que l'approche robuste d'extraction des cavités, contrôlée par une distance de Hausdorff, est un élément important pour le développement d'un mailleur hex-dominant robuste. Par robuste, nous entendons un mailleur qui puisse générer un maillage hex-dominant pour toute géométrie d'entrée valide, *c.à-d.* surface fermée sans auto-intersections, quitte à produire un maillage avec beaucoup de tétraèdres si les étapes de maillage hexaédrique (champs de directions, paramétrisation) n'ont pas fonctionné de manière satisfaisante. Notre approche permet également de supprimer les hexaèdres invalides ou trop déformés selon des critères géométriques. Ces garanties sont nécessaires pour envisager l'usage régulier d'un mailleur hex-dominant dans un contexte industriel.

modèle	notre approche +TetGen	notre approche +Carrier-Baudoin	PGP3D
fusee	92.9	95	94.6
CV745	92.3	94.4	91.6
propeller	85.1	89.4	91.1
cubo	88.8	91.8	89
cylinder	98.9	99.3	90.9
corner	99.3	99.7	99.5
fandisk	75.5	82.6	77.8
impeller	73.3	81.8	81.1
bunny	89.8	92.2	88.6
bone	70	81.9	82.54
fertility	79.9	86.4	78.4

TABLE 3.1 – Proportion du volume couvert par des hexaèdres (en %). Amélioration de la proportion d'hexaèdres via recombinaison [Bau+14] (milieu) par rapport à une tétraédrisation des cavités (gauche). La colonne de droite reporte les résultats de l'article original sur *PGP* [Sok+16] à titre de comparaison.

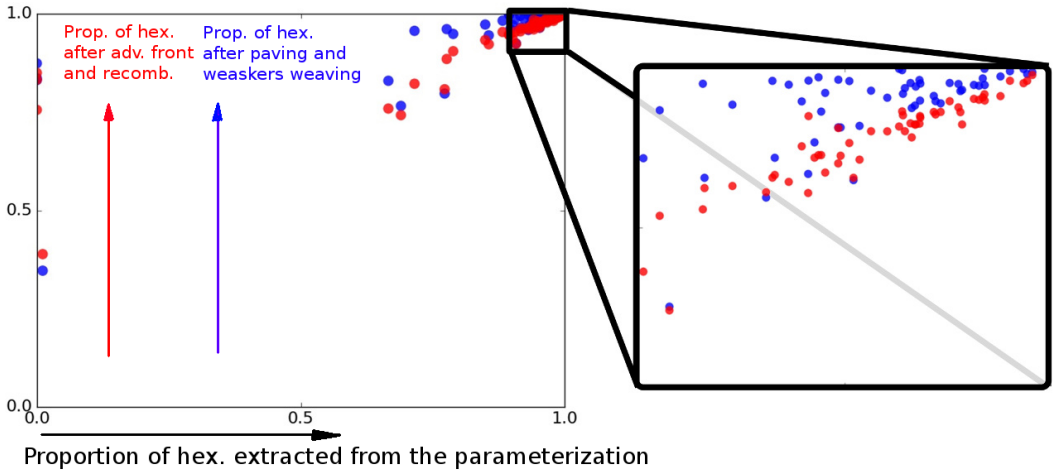


FIGURE 3.10 – Comparaison des méthodes de remplissage des cavités. Les résultats de l'approche par recombinaison sont affichés en rouge et ceux l'approche par *pavage et whisker-weaving* en bleu.

3.2 Transformation des singularités volumiques en singularités linéiques

Lorsque les champs de directions sont échantillonnés sur les sommets (et non sur les éléments), les singularités discrétisées sont volumiques. Elles sont formées de tétraèdres, que nous qualifions de *singuliers*. Or le graphe de singularités d'un champ de directions continu est composé de lignes, qui peuvent se rejoindre en des noeuds (§2.1.2,p. 39). Dans un tétraèdre singulier, la paramétrisation n'est pas définie et il n'est pas possible d'extraire des parties d'hexaèdres, ce qui conduit à des maillages hex-dominants avec un grand nombre de tétraèdres autour des singularités. Ce problème n'existe pas avec l'approche par échantillonnage constant des *frames* sur les éléments, car les singularités sont alors localisées sur des arêtes, ce qui ne perturbe pas la paramétrisation qui est continue par éléments (cas de *CubeCover* [NRP11]). Cependant l'échantillonnage par sommets a ses avantages : plus de degrés de liberté et meilleure convergence de l'optimisation non-linéaire de lissage du champ [RSL16].

L'objectif de cette section est de proposer une solution technique permettant de transformer le maillage afin de placer les singularités sur des arêtes lorsque le champ de directions est échantillonné sur les sommets. Le but est de pouvoir définir une paramétrisation sur tout le domaine et donc de pouvoir potentiellement extraire un maillage entièrement hexaédrique.

Vocabulaire Afin de simplifier la description de notre méthode, nous adoptons le vocabulaire suivant emprunté à [Kow13] :

- Une direction est **stable** sur un triangle si en faisant le tour, la multiplication des matrices de permutation permettant de passer d'une *frame* à une autre le long des arêtes laisse inchangeante la direction ;
- Une face (triangulaire) est **singulière** si l'une ou plus des directions est instable ;
- Une tétraèdre est **2-singulier** si deux directions sont instables. Par définition, un tétraèdre 2-singulier possède une direction stable (comme il y a trois directions orthogonales) ;
- Une tétraèdre est **3-singulier** si aucune des directions n'est stable.

De plus il n'existe pas de tétraèdre *1-singulier* car si une direction *permute* sur un triangle, elle permute forcement avec une autre direction qui est elle aussi instable [Kow13]. Par conséquent, comme à l'intérieur du maillage, une face appartient forcement à deux tétraèdres, les tétraèdres singuliers forment des *enchainements* qui ne peuvent ni apparaitre ni disparaitre à l'intérieur du volume. Cependant, plusieurs *enchainements* de tétraèdres singuliers peuvent se rejoindre à l'intérieur grâce aux tétraèdres 3-singuliers. Nous introduisons de nouvelles notations :

- Une **chaine singulière** est l'ensemble maximal de tétraèdres 2-singuliers qui peuvent être regroupés par adjacence. Une chaine se termine sur un bord ou sur un tétraèdre 3-singulier.
- Un **noeud singulier** est l'ensemble maximal de tétraèdres 3-singuliers qui peuvent être regroupés par adjacence.
- Le **graphe de singularité volumique** est l'union des chaines et des noeuds singuliers.

Par construction, tout tétraèdre 2-singulier du maillage appartient à une chaine singulière et tout tétraèdre 3-singulier du maillage appartient à un noeud singulier.

Comme la présence de chaines ou de noeuds singuliers empêche de construire une paramétrisation valide recouvrant tout le domaine, nous souhaitons remplacer le graphe de singularité volumique par un graphe de singularité linéique. Plus précisément, nous souhaitons remplacer les chaines singulières par des lignes et les noeuds singuliers par des points. Pour conserver la structure de données de maillage tétraédrique conforme avec échantillonnage sur les sommets, le graphe de singularité linéique peut être représenté par un graphe de singularité volumique (*c.à-d.* composé de tétraèdres) dont tous les tétraèdres sont de volume nul. C'est la technique utilisée dans la suite. Pour produire des tétraèdres de volume nul, nous avons étudié deux approches, illustrées en 2D sur la Figure 3.11 :

— **Contraction :** les tétraèdres singuliers des chaines sont comprimés le long de lignes géométriques. Les sommets du maillage sont déplacés de manière à former des tétraèdres de volume nul dans les chaines singulières. De la même manière, les noeuds singuliers sont réduits à des points en déplaçant les sommets.

— **Décomposition :** les tétraèdres des chaines singulières sont décomposés en sous-tétraèdres de manière à faire apparaitre une série d'arêtes contenant la singularité.

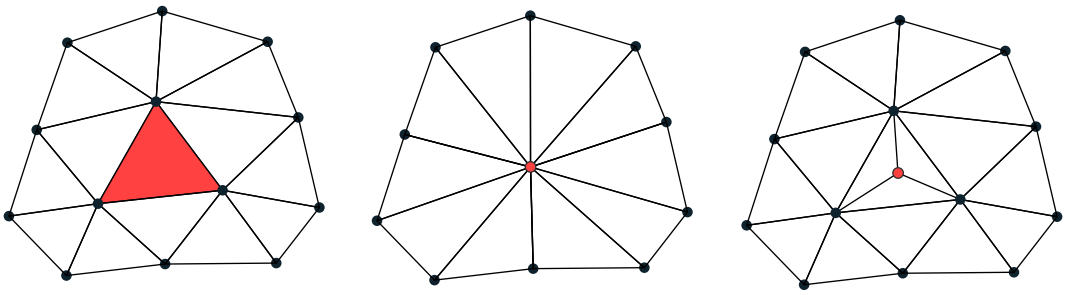


FIGURE 3.11 – Transformation des singularités volumiques en singularités linéiques, illustration du principe en 2D. Deux approches possibles : contraction ou décomposition. *Gauche :* maillage initial, triangle singulier en rouge. *Centre :* contraction du triangle en un point, ses sommets sont fusionnés. *Droite :* décomposition du triangle singulier en trois triangles réguliers.

3.2.1 Approche par contraction

Considérons une chaine singulière de tétraèdres, immergée à l'intérieur d'un maillage. L'objectif est de déplacer les sommets de la chaine de manière à former une ligne, c'est-à-dire une chaine de volume nulle. Cette transformation doit s'effectuer de manière à ce que : (1) la nouvelle chaine linéique soit contenue dans le volume de la chaine initiale et (2) tous les tétraèdres adjacents à la chaine conservent un volume strictement positif (pas d'inversion de tétraèdres dans le maillage). L'avantage de cette approche est qu'il n'y a pas à changer la connectivité du maillage.

La seconde condition (2) s'avère très difficile à réaliser en pratique et fréquemment il n'y a pas de solution. Pour chaque sommet à déplacer, il est possible de construire un volume dans lequel le sommet peut être déplacé sans inverser les tétraèdres qui le contiennent. Cet espace est formé par intersection de demi-plans associés aux faces ne contenant pas le sommets des *tétraèdres contenant le sommet*.

De la même manière, les noeuds volumiques sont transformés en un point géométrique en s'assurant de ne pas retourner les tétraèdres adjacents.

Pour explorer les solutions, nous avons construit un algorithme récursif qui déplace les sommets les uns après les autres, avec retour en arrière lorsque la branche en cours aboutit à une impasse, *c.à-d.* lorsqu'un tétraèdre est retourné.

Cet algorithme fonctionne sur des cas relativement simples mais il est très fréquent de ne pas trouver de solution. Dans ce cas, il faut relâcher le problème en autorisant quelques inversions (légères) de tétraèdres adjacents, qui sont ensuite corrigés en déplaçant des sommets plus loin dans le maillage lors d'une étape de nettoyage ultérieure.

Les essais expérimentaux que nous avons conduits ont donné des résultats plutôt décevants. Lorsque l'algorithme fonctionne, la ligne de singularité obtenue est souvent de mauvaise qualité (angles locaux importants), ce qui conduit à un champ de directions localement très déformé, et par conséquent à une paramétrisation de mauvaise qualité.

En bref, cette approche est un échec : elle est lourde à mettre en oeuvre, coûteuse en calculs et ne donne pas de résultats satisfaisants. Nous retenons de cette expérience qu'il vaut mieux éviter de dégrader successivement le maillage initial en déplaçant des sommets car cela conduit à des situations de plus en plus difficiles à gérer.

3.2.2 Approche par décomposition

Plutôt que de modifier la position des sommets dans le maillage, il est possible d'en ajouter des nouveaux et de former de nouveaux tétraèdres. L'avantage de cette approche est qu'elle est majoritairement combinatoire, et par conséquent nous ne rencontrerons pas les problèmes géométriques (*par ex.* introduction de volumes négatifs) de l'approche précédente. Le principe est illustré en dimension deux sur la droite de la Figure 3.11 et en dimension trois sur le tétraèdre de la Figure 3.12.

Décomposition des tétraèdres Pour chaque tétraèdre de la chaine singulière, il y a deux faces singulières. Un nouveau sommet est inséré au centre de chacune et la connectivité est modifiée afin de remplacer le tétraèdre singulier par cinq nouveaux tétraèdres. Cette décomposition est illustrée au centre de la Figure 3.12.

Comme dans notre maillage, le champ de directions est échantillonné sur les sommets, il n'est pas possible de représenter une singularité sur une arête avec des tétraèdres conformes dont les volumes sont strictement positifs. Afin de conserver notre structure de données, nous avons préféré dupliquer géométriquement les sommets et créer des tétraèdres de volumes nuls, permettant de stocker la singularité. Plus précisément, nous insérons trois nouveaux sommets par face singulière, trois nouveaux tétraèdres-arêtes (volume nul) qui lient les nouveaux points et sept nouveaux tétraèdres-triangles (volume nul) pour effectuer les jonctions entre les cinq tétraèdres volumiques. Cette décomposition est illustrée sur la droite de la Figure 3.12. Elle est certes techniquement compliquée (quinze tétraèdres au lieu d'un) mais elle permet

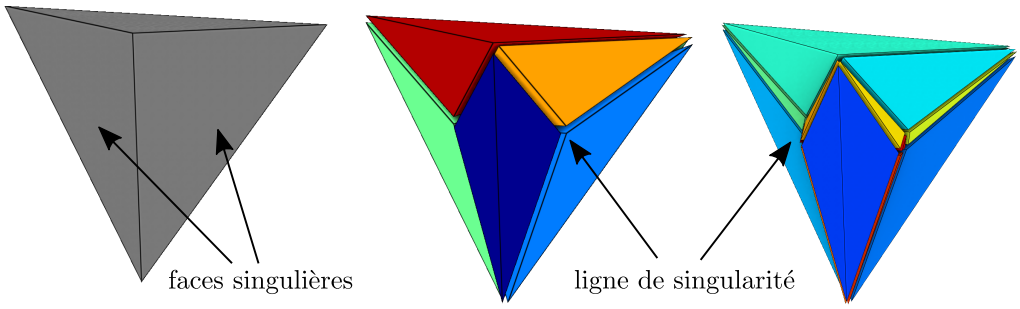


FIGURE 3.12 – Transformation des singularités volumiques en singularités linéiques par décomposition. *Gauche :* maillage initial, singularité volumique correspondant à deux faces singulières. *Centre :* décomposition du tétraèdre en cinq tétraèdres réguliers. *Droite :* décomposition en quinze tétraèdres : les cinq volumiques plus dix tétraèdres à volume nul permettant de créer une discontinuité dans le champ de directions. Les tétraèdres de volume nul sur la ligne de singularité sont singuliers.

de conserver un maillage conforme et de représenter des discontinuités du champ de directions au niveau de singularités géométriquement linéiques.

La description combinatoire de la décomposition est détaillée sur la partie haute de la Figure 3.13. Chaque nouveau sommet (trois par face) reçoit une copie d'une des frames associées aux quatre sommets initiaux afin de rendre non-singulier les tétraèdres à volume non-nuls et les tétraèdres-triangles à volumes nuls. L'assignation des frames est représentée par les flèches en rouge sur la Figure 3.13.

Jonction des tétraèdres décomposés Afin de joindre les deux décompositions de deux tétraèdres successifs d'une chaine singulière, il faut ajouter de nouveaux tétraèdres à volumes nuls. Pour une face commune à deux tétraèdres 2-singuliers adjacents, il y aura six nouveaux sommets à la même position géométrique : trois pour la face vue par chaque tétraèdre.

Les deux décompositions sont liées par trois tétraèdres-triangles liant les tétraèdres à volume non nul, trois tétraèdres-points liant les six sommets et six tétraèdres-arêtes liant les trois arêtes internes de l'interface. La combinatoire de la jonction est détaillée sur la partie basse de la Figure 3.13.

Contraintes à ajouter dans le solveur mixte réels-entiers Dans l'approche *CubeCover*, la paramétrisation est calculée grâce au solveur mixte réels-entiers, à partir du champ de directions. Dans l'approche que nous utilisons initialement, *c.à-d. CubeCover sans singularités* avec échantillonnage sur les sommets du champ de directions, les contraintes sur le solveur sont :

— sur les bords, les variables sur les coins des tétraèdres (u, v ou w) sont entières dans la direction normale au bord. Ceci permet que des faces d'hexaèdres extraits (une coordonnée entière dans la paramétrisation) soient sur le bord du modèle ;

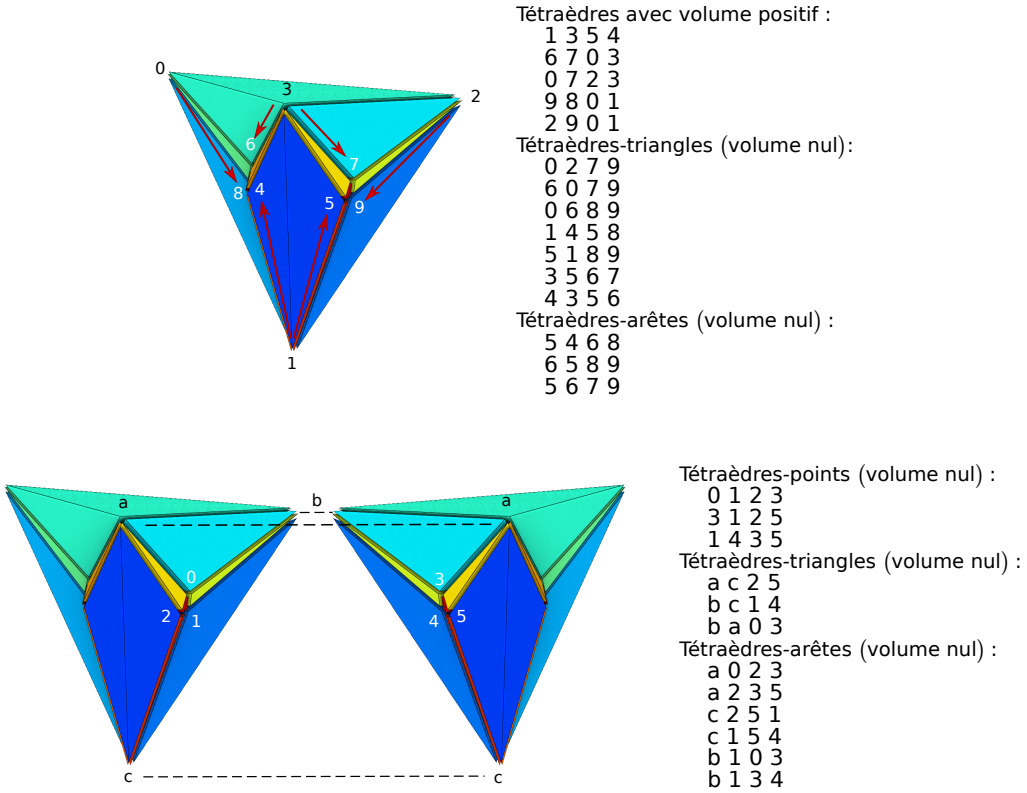


FIGURE 3.13 – *Haut :* Décomposition d'un tétraèdre 2-singulier en quinze tétraèdres, dont dix à volumes nuls. Les nouveaux sommets (d'indices 5 à 9) reçoivent une copie d'une frame d'un des sommets d'origine (flèches en rouge). *Bas :* Tétraèdres assurant la jonction conforme entre deux décompositions de tétraèdres adjacents.

— sur les arêtes du bord, les variables de la direction entière (condition précédente) sont identiques ;

— sur toutes les faces appartenant à deux tétraèdres adjacents, la discontinuité entre les deux est *grille-compatible*, c'est-à-dire que les paramétrisations des deux côtés sont liées par une permutation d'axes et une translation entière. Ceci permet d'autoriser certains types de discontinuités dans la paramétrisation ;

— sur toutes les faces triangulaires du maillage, le champ vectoriel (u, v, w) associé à la paramétrisation est une *1-forme* close, *c.à-d.* $\mathrm{rot}((u, v, w)) = \mathbf{0}$.

Ces contraintes sont suffisantes car le champ de directions à *intégrer* ne comporte pas de singularités (elles sont exclues). Dans la nouvelle version, avec des singularités dans des volumes nuls, il faut ajouter les contraintes suivantes :

— les variables sur le graphe de singularité linéique (volume nul) sont entières dans les deux directions instables (perpendiculaires à la ligne de singularité). Ceci permet de créer des arêtes dans le maillage hexaédrique le long de la singularité. Au niveau des noeuds singuliers (réduits à un point géométrique), les trois champs scalaires sont entiers ;

— les variables des coins appartenant à une arête de longueur nulle sont *grille-compatibles*.

Le reste de la formulation du problème mixte est inchangé. C'est là tout l'intérêt de conserver un maillage conforme, quitte à avoir des éléments de volumes nuls.

Positionnement des nouveaux points. En première approche, nous avons inséré des nouveaux points aux barycentres des faces singulières. Ceci produit des lignes de singularités non lisses et très dépendantes du maillage. Par exemple sur la droite de la Figure 3.14, on peut observer que la ligne de singularité est de mauvaise qualité même pour un modèle très simple.

Il est possible d'améliorer le positionnement des nouveaux points en observant les valeurs des frames sur les sommets, et en localisant la singularité par interpolation. Dans les tétraèdres 2-singuliers, nous pouvons utiliser l'approche de Kowalski [Kow13] : transformer les frames en croix en utilisant la direction stable, puis les représenter en utilisant les vecteurs de représentation, interpoler linéairement ce champ vectoriel dans les triangles et localiser l'endroit où il s'annule.

Une autre possibilité est d'ignorer le champ de directions et de positionner les nouveaux points dans le volume singulier de manière à obtenir des lignes de singularités lisses.

Gestion des noeuds singuliers. Les chaines singulières se rejoignent en noeuds singuliers. Pour que le maillage soit conforme avec les décompositions des chaines présentées précédemment, il faut également subdiviser les noeuds.

Lorsqu'un noeud est composé d'un unique tétraèdre, on peut construire une nouvelle décomposition qui insère des nouveaux sommets sur les trois, ou quatre, faces adjacentes à des chaines, avec également un nouveau sommet au centre du tétraèdre, où les singularités linéiques peuvent se rejoindre.

Malheureusement en pratique, les noeuds singuliers sont souvent composés de quelques tétraèdres, typiquement deux ou trois, et ils ont des configurations assez diverses. L'utilisation de décompositions qui subdivisent trois ou quatre faces introduit plusieurs sommets intérieurs où les singularités se rejoignent dans le même noeud volumique. Cette configuration ne pourra pas donner lieu à un maillage hexaédrique.

Pour construire une décomposition *maillable*, il faudrait décomposer le noeud entier (*c.à-d.* les tétraèdres 3-singulier adjacents) afin de faire apparaitre un unique sommet central où les singularités linéiques se rejoignent. Cette tâche est particulièrement compliquée car il y a beaucoup de configurations différentes et des problèmes géométriques à gérer car les noeuds ne sont pas convexes. Par manque de temps, et également parce que cette approche a de nombreux inconvénients, nous n'avons pas développé ces décompositions.

Lorsque le maillage contient des noeuds singuliers, nous nous contentons de décomposer les chaines singulières et de laisser les noeuds inchangés. Ceci produit des maillages tétraédriques non-conformes (au niveau des interfaces chaines-noeuds). Cependant, notre *workflow* fonctionne tout de même avec ces interfaces non-conformes car les noeuds singuliers restants sont exclus de la paramétrisation. Un exemple de graphe de singularité mixte est illustré sur la Figure 3.15.

Position géométrique des chaines singulières. Lorsque que les chaines singulières sont proches du bord du modèle, il n'y a pas la place pour insérer des hexaèdres entre la ligne singulière obtenue par décomposition et le bord. Cette situation est

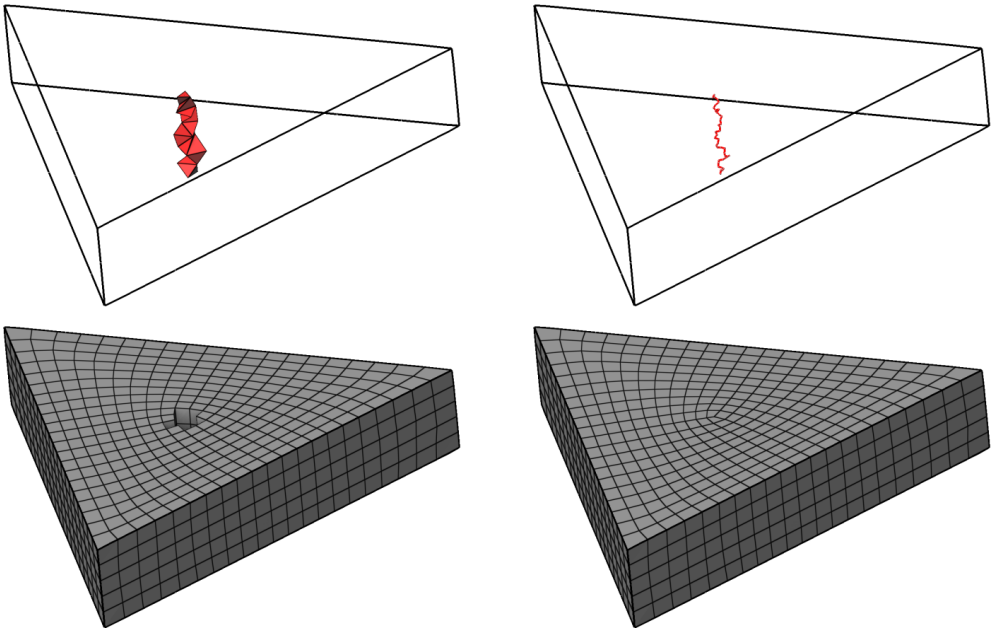


FIGURE 3.14 – *Haut :* Graphes de singularités. *Bas :* Maillages hexaédriques extraits de la paramétrisation. *Gauche :* Échantillonnage par sommet initial. *Droite :* Échantillonnage par sommet avec graphe de singularité de volume nul, obtenu par décomposition.

courante car les singularités ont souvent tendance a être proches et parallèles aux bords.

Pour atténuer cet effet, il est possible de modifier la fonction d'optimisation utilisée pour générer les champs de directions : le coût des variations de frames proche du bord est augmenté par rapport au coût loin de ceux-ci. Cette modification fait que naturellement les singularités apparaitront plutôt à l'intérieur.

Cette optimisation permet d'améliorer légèrement la qualité des hexaèdres extraits, mais elle n'a pas une grande influence car la topologie est inchangée. De plus, la géométrie des hexaèdres extraits était déjà améliorée par une étape finale de lissage.

Expériences numériques. Le cas le plus simple à traiter est celui d'un triangle extrudé, illustré sur la Figure 3.14. Il contient une unique singularité linéique droite qui traverse le volume. Notre méthode de décomposition permet de transformer la singularité volumique en singularité linéique (haut de la Figure 3.14), ce qui conduit à une paramétrisation valide sur tout le domaine, de laquelle nous extrayons un maillage entièrement hexaédrique (bas droit de la Figure 3.14). Cet exemple simple confirme que notre approche fonctionne sur des chaines singulières qui traversent le volume de part en part.

Le graphe de singularité d'un modèle 3D de tétraèdre est légèrement plus compliqué : il contient quatre chaines singulières qui se rejoignent en un noeud au centre du modèle. Cet exemple est illustré sur la Figure 3.14. Notre méthode de décomposition

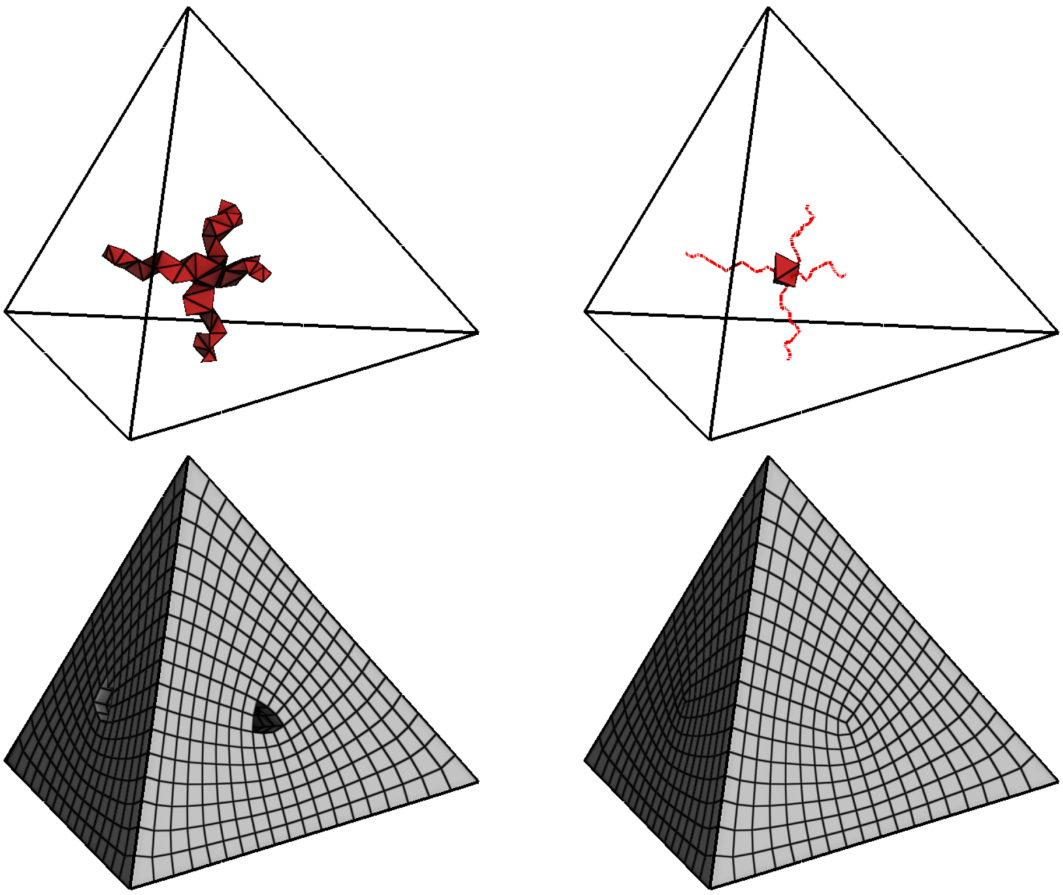


FIGURE 3.15 – *Haut :* Graphes de singularités. *Bas :* Maillages hexaédriques extraits de la paramétrisation. *Gauche :* Échantillonnage par sommet initial. *Droite :* Échantillonnage par sommet avec graphe de singularité de volume nul, obtenu par décomposition. Le maillage en bas à droite n'est pas entièrement hexaédrique, il contient des tétraèdres en son centre.

transforme le graphe de singularité volumique en un graphe contenant des lignes, associées aux chaines singulières, et le noeud volumique initial composé de deux tétraèdres. Grâce à cette décomposition, l'algorithme *CubeCover modifié* génère une paramétrisation valide autour des chaines singulières linéiques et nous pouvons y extraire des hexaèdres, ce qui augmente la proportion d'hexaèdres dans le maillage hex-dominant final.

Discussion L'objectif initial était de développer une technique simple pour mailler de manière entièrement hexaédrique les singularités linéiques (arêtes de valence trois ou cinq en général) tout en utilisant une discrétisation par sommet des champs de directions. Ceci permet notamment de reproduire les résultats de l'état de l'art [Li+12] obtenus avec une discrétisation constante par tétraèdre.

Les résultats sont mitigés : la technique par décomposition fonctionne sur les chaines singulières mais nous n'avons pas réussi à gérer correctement les noeuds

singuliers.

Si la méthode se voulait simple à la base, l'implémentation s'avère particulièrement technique car il y a de (trop) nombreux tétraèdres à volumes nuls à gérer afin de conserver un maillage conforme. Principalement pour cette raison, nous n'avons pas persévéré dans cette direction.

Une alternative plus élégante serait d'utiliser une discrétisation continue par morceaux du champ de directions, avec des discontinuités possibles le long d'arêtes. Le nouveau problème consisterait à :

1. extraire un réseau d'arêtes du maillage tétraédrique *similaire* au graphe de singularités volumique ;
2. autoriser des discontinuités du champ de directions sur ce réseau d'arêtes ;
3. projeter le champ de directions initial sur le nouvel espace de discrétisation de manière à ce que le nouveau champ de directions soit régulier sur tous les tétraèdres et que les singularités soient réduites au réseau d'arêtes présélectionné.

Pour mettre en oeuvre cette idée, il faudrait développer des outils pour manipuler localement les champs de direction. Notamment, il faut pouvoir effectuer des projections ou des interpolations de champ de directions qui n'introduisent pas de nouvelles singularités parasites.

Chapitre 4

Méthode des éléments finis et maillages hex-dominants

L'objectif de ce chapitre est d'évaluer l'intérêt des maillages hex-dominants pour la simulation par éléments finis. Les maillages hex-dominants sont supposés être intéressants car ils contiennent une majorité d'hexaèdres, dont les performances peuvent être supérieures aux éléments tétraédriques. Nous commençons par effectuer des comparaisons simples entre triangles et quadrilatères (§4.1) afin de compléter les comparaisons évoquées dans l'état de l'art (§2.2.1.1). Nous construisons ensuite un espace d'approximation éléments finis continu (§4.2) adapté aux maillages hexa-tétraédriques non-conformes générés automatiquement avec des techniques du *Chapitre 3*. Finalement, sur des domaines avec des géométries compliquées, nous comparons des résultats obtenus avec des maillages hex-dominants, hexaédriques et tétraédriques (§4.3) en utilisant l'outil de calcul de distances développé dans le *Chapitre 5*.

4.1 Comparaisons de primitives sur des problèmes simples 2D

Dans cette section, nous comparons les performances de maillages quadrilatérales, éventuellement quad-dominants, et de maillages triangulaires. Nous étudions en particulier l'influence sur la précision de l'alignement des éléments avec les variations des champs solutions. Les méthodes mises en oeuvre sont tout ce qu'il y a de plus classique, mais nous pensons que les résultats sont intéressants car ils ne sont pas forcément très intuitifs à première vue.

4.1.1 Problèmes scalaires elliptiques linéaires d'ordre deux

Les trois premiers problèmes sont définis sur le domaine carré unité ($\Omega = [0,1]^2$) et possèdent une solution analytique connue. Avec la méthode des solutions manufacturées (§2.3.1, p. 67), nous calculons le terme source et les conditions aux limites associés. Pour chaque problème, nous calculons l'erreur en norme L^2 obtenue avec différentes résolutions de maillages, et des éléments finis de Lagrange d'ordres un et deux. Les maillages du carré unité sont de quatre types :

— quadrilatérales réguliers alignés avec le domaine, notés $\mathbb{Q}_k$;

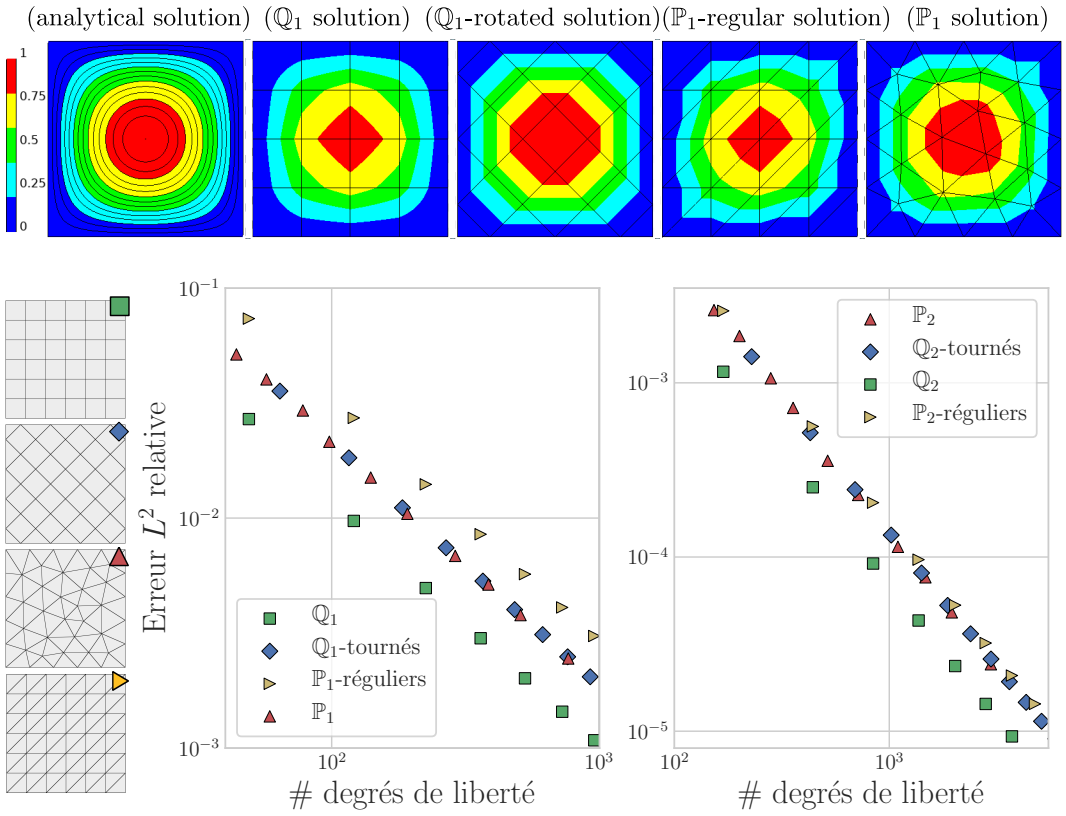


FIGURE 4.1 – Problème de la bosse sinusoïdale (4.1). *Haut :* Solution analytique et exemples de solutions d'ordre un pour différents maillages. *Bas :* Erreur relative L^2 en fonction du nombre de degrés de liberté pour différents maillages avec des éléments d'ordre un (gauche) et d'ordre deux (droite).

— quadrilatérales réguliers tournés de 45 degrés, notés $\mathbb{Q}_k$-tournés. Ces maillages sont en réalité quad-dominants car des triangles sont utilisés pour mailler les quatre bords du carré unité ;

— triangulaires réguliers, obtenus par décomposition des quadrilatères en deux triangles, notés $\mathbb{P}_k$-réguliers ;

— triangulaires aléatoires, obtenus par triangulation de Delaunay, notés $\mathbb{P}_k$.

Problème de la bosse sinusoïdale Le problème de la bosse sinusoïdale, *sine bump* en anglais, est un problème de Poisson très simple qui est souvent utilisé comme validation d'implémentations. Le problème est défini par :

$$\begin{cases} -\Delta u = f & \text{dans } \Omega \\ u = 0 & \text{sur } \partial\Omega \end{cases} \tag{4.1}$$

avec la solution analytique : $u = \sin(2\pi x)\sin(2\pi y)$.

Ce problème est résolu avec différents maillages quadrangulaires et triangulaires et des éléments de Lagrange d'ordres un et deux. L'allure des solutions et la convergence en norme L^2 avec le raffinement sont indiquées sur la Figure 4.1.

Si on se contente d'observer les quadrilatères $\mathbb{Q}_k$ et les triangles $\mathbb{P}_1$-réguliers, obtenus par division en deux des quads, on observe que les éléments $\mathbb{Q}_k$ sont environ 30% plus précis que les éléments $\mathbb{P}_k$-réguliers. Cette différence est due aux termes croisés supplémentaires de $\mathbb{Q}_k$, dont on peut observer l'intérêt sur l'allure de la solution $\mathbb{Q}_1$ en regardant les quatre coins : les termes croisés xy permettent d'approximer plus précisément la courbure du produit de sinus. Avec des quadrilatères tournés de 45 degrés, les termes croisés ne participent plus à l'approximation, comme on peut le voir sur l'allure de $\mathbb{Q}_1$-tournés. En effet, même si $\mathbb{Q}_k$ contient des termes croisés de degrés supérieurs à k, l'interpolation sur les arêtes est un polynôme de degré k au plus (car l'arête appartient à $\mathbb{P}_k^{1D}$). Par conséquent, les termes croisés ne peuvent améliorer l'interpolation qu'à l'intérieur des éléments et non uniformément sur tout le domaine.

En observant l'allure de la solution $\mathbb{P}_1$-réguliers, on voit clairement qu'il y a un biais directionnel dû à la division des quadrilatères selon une diagonale. Si on utilise un maillage triangulaire avec une répartition aléatoire ($\mathbb{P}_1$), le biais directionnel est supprimé et l'approximation s'en trouve améliorée.

De cet exemple très simple, nous retenons que l'orientation des éléments influence la solution, y compris avec le raffinement. Par exemple, les quadrilatères tournés de 45 degrés ne sont pas meilleurs que les triangles aléatoires, contrairement aux quadrilatères alignés.

Problème de l'arctangente Le problème qui suit est une version légèrement modifiée du problème *wave front* répertorié dans [Mit13]. La solution analytique contient un *front*, parallèle à l'axe des ordonnées, sur lequel la solution varie fortement. L'objectif est d'étudier l'influence ou non de l'alignement des éléments avec le front. L'équation aux dérivées partielles est légèrement plus compliquée que celle du problème de Poisson car il y a un terme linéaire u supplémentaire. Ceci dit, il s'agit toujours d'une équation linéaire elliptique du second ordre qui admet des solutions relativement lisses. Le problème est défini par :

$$\begin{cases} -\Delta u + u = f & \text{dans } \Omega \\ u = u_a & \text{sur } \partial\Omega \end{cases} \tag{4.2}$$

avec la solution analytique :

$$u_a = \arctan(\alpha(r - r_0))\ y(1-y), \quad r = \sqrt{(x-x_0)^2 + (y-y_0)^2)}$$

et les valeurs numériques :

$$\alpha = 20, \quad (x_0, y_0) = (-0.437, 0.4), \quad r_0 = 1$$

Le terme $y\ (1-y)$ permet d'annuler la solution sur les bords $y = 0$ et $y = 1$. Nous l'utilisons car sinon, d'après notre expérience, la discrétisation des conditions aux limites a trop d'influence sur la solution et il est difficile de séparer les causes dans l'analyse des résultats.

Les solutions approchées obtenues avec des éléments finis d'ordres un et deux, pour différentes résolutions, sont illustrées sur la Figure 4.2. L'élément intéressant est que les résultats ne sont pas très conformes à l'intuition, du moins à celle de l'auteur. Sur ce problème précis, nous pouvons faire les observations suivantes :

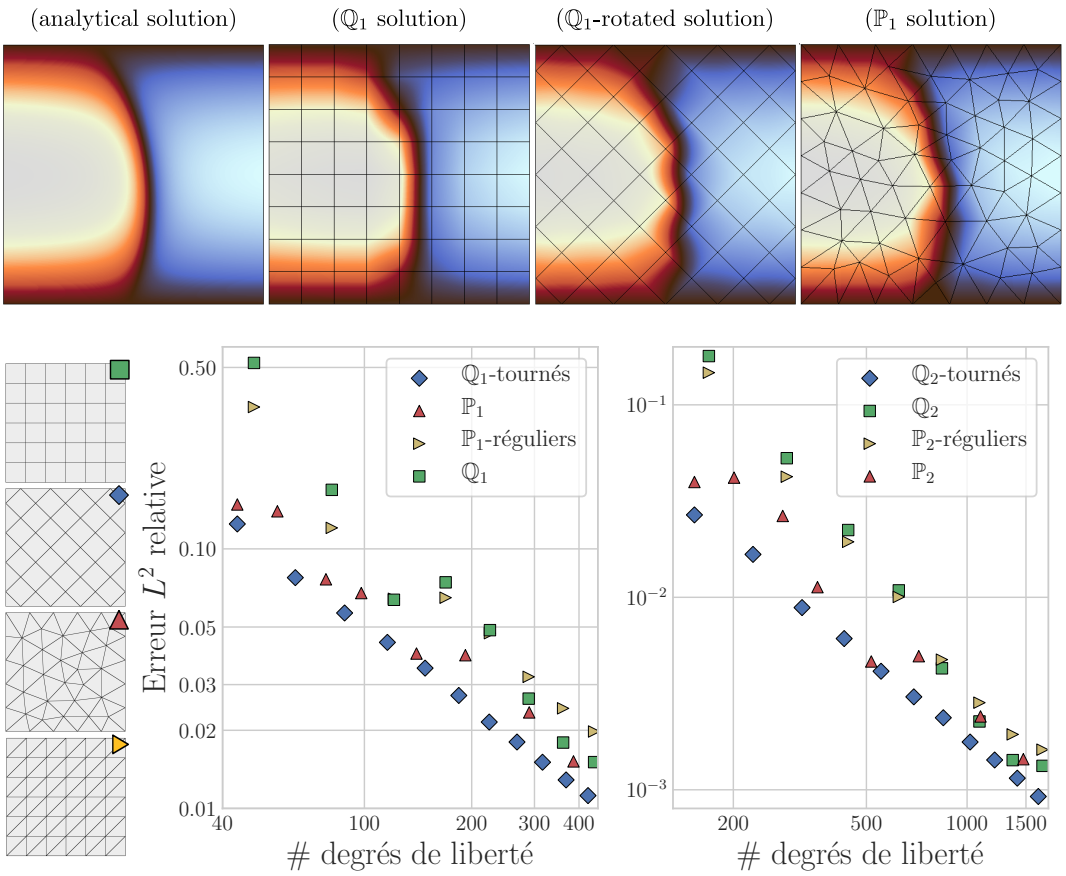


FIGURE 4.2 – Problème de l'arctangente (4.2). *Haut :* Solution analytique et exemples de solutions d'ordre un pour différents maillages. *Bas :* Erreur relative L^2 en fonction du nombre de degrés de liberté pour différents maillages avec des éléments d'ordre un (gauche) et d'ordre deux (droite).

— Les éléments finis $\mathbb{Q}_k$-tournés, c'est-à-dire ceux dont la diagonale est alignée avec le bord et non les éléments, sont les plus précis. Cet effet est dû aux termes polynomiaux croisés (e.g. xy for $\mathbb{Q}_1$) qui sont alignés avec la variation de la solution. Sur la troisième vignette (haut de la Figure 4.2), on peut voir les oscillations le long du front qui sont dues aux termes croisés.

— Les éléments $\mathbb{Q}_k$, et dans une moindre mesure $\mathbb{P}_k$, sont sensibles au positionnement des éléments par rapport au front, comme on peut le voir sur les oscillations des courbes de convergence. Ils sont plus précis quand les arêtes des éléments sont placées plus ou moins au milieu du front. Les éléments $\mathbb{Q}_k$-tournés ne sont pas influencés par ce phénomène parce que les arêtes ne sont jamais parallèles au front.

— Les éléments $\mathbb{P}_k$ sont plus précis que les éléments $\mathbb{Q}_k$ alignés avec le bord. C'est sans doute l'observation la plus étonnante de cette expérience. L'explication est que (a) les termes croisés des éléments $\mathbb{Q}_k$ n'apportent pas de précision supplémentaire (cf $\mathbb{P}_k$-régulier) et (b) les arêtes des triangles, d'orientations aléatoires, sont plus souvent aptes à capturer les variations du front en moyenne. Dans le cas du maillage quadrilatéral aligné avec le bord, ça n'arrive que pour des

raffinements spécifiques.

— Les trois types maillages convergent plus ou moins vers le même niveau de précision (celui des $\mathbb{Q}_k$-tournés) avec le raffinement, à mesure que le front est bien capturé (plusieurs éléments à l'intérieur de celui-ci). Lorsque cette situation est atteinte, c'est l'ordre des polynômes d'interpolation qui contrôle l'erreur, sachant que la solution est $\mathcal{C}^\infty$.

Les observations faites sur cet exemple sont représentatives des autres expériences que nous avons menées avec différentes solutions analytiques fabriquées avec la méthode des solutions manufacturées. Nous retenons de cet exemple que les idées *les éléments quadrilatéraux (et plus généralement d-cubes) sont meilleurs que les éléments triangulaires* et *l'alignement des éléments avec les caractéristiques de la solution est avantageux* ne sont pas évidentes et qu'elles nécessitent des conditions particulières pour s'appliquer.

Un gros défaut des deux expériences précédentes, bosse sinusoïdale et arctangente, est que la solution est principalement contrôlée par le terme source f, or (a) la discrétisation de ce terme source est dépendante des éléments utilisés et (b) les dérivées sont plus ou moins prescrites en tout point du domaine. C'est le cas pour la plupart des problèmes construits avec la méthode des solutions manufacturées, or ceux-ci sont majoritaires dans la littérature lorsqu'il s'agit de construire et d'analyser des courbes de convergence, puisque la solution analytique connue permet de calculer l'erreur.

Problème de Helmholtz sinusoïdal Le problème suivant est toujours un problème linéaire simple d'ordre deux dont la solution analytique est connue, mais sans terme source :

$$\begin{cases} -\Delta u - (4\pi)^2 u = 0 & \text{dans } \Omega \\ u = \sin(4\pi x) & \text{sur } \partial\Omega \end{cases} \tag{4.3}$$

avec la solution analytique : $u_a = \sin(4\pi x)$. L'allure des solutions et les erreurs L^2 commises avec les différents maillages sont renseignées sur la Figure 4.3.

Comme pour le problème de l'arctangente, on observe que les quadrilatères tournés de 45 degrés sont les plus précis : les termes croisés, associés aux diagonales des quadrilatères, sont alignés avec la variation de la solution (axe des abscisses). À l'ordre un, les triangles sont légèrement meilleurs que les quadrilatères alignés avec le bord car ils ont plus de flexibilité grâce aux arêtes (discontinuité du gradient) qui se trouvent plus ou moins parallèles à la variation. À l'ordre deux, les quadrilatères alignés avec le bord et les triangles sont équivalents en termes de précision.

4.1.2 Problèmes d'élasticité linéaire

Le système d'équations de l'élasticité linéaire est une modélisation simple et classique du comportement en petite déformation des milieux continus. Soit un objet de domaine Ω à l'équilibre, fixé sur le bord $\partial\Omega_D$, soumis à une force volumique $\mathbf{f}$ dans Ω et à une force surfacique $\mathbf{g}$ sur le bord $\partial\Omega_N$. La loi de comportement du matériau

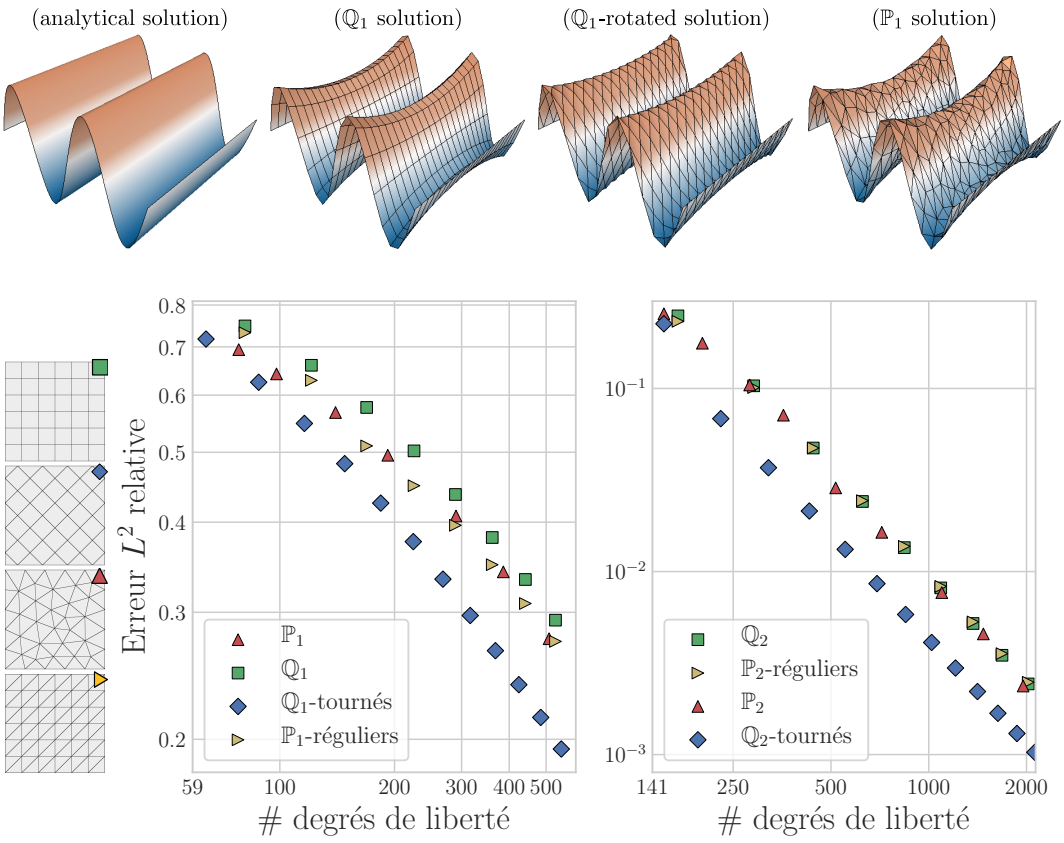


FIGURE 4.3 – Problème de Helmholtz sinusoïdal (4.3). *Haut :* Solution analytique et exemples de solutions d'ordre un pour différents maillages. *Bas :* Erreur relative L^2 en fonction du nombre de degrés de liberté pour différents maillages avec des éléments d'ordre un (gauche) et d'ordre deux (droite).

est donnée par la loi de Hooke, dont les coefficients de Lamé sont λ et μ. Le champ de déplacement $\mathbf{u}$, solution du problème, est gouverné par le système :

$$\nabla \cdot \sigma(\mathbf{u}) + \mathbf{f} = 0 \qquad \text{dans} \quad \Omega \tag{4.4}$$

$$\sigma(\mathbf{u}) = \lambda(\nabla \cdot \mathbf{u})\mathcal{I} + \mu(\nabla \cdot \mathbf{u} + \nabla \cdot \mathbf{u}^T) \qquad \text{dans} \quad \Omega \tag{4.5}$$

$$\mathbf{u} = 0 \qquad \text{sur} \quad \partial\Omega_D \tag{4.6}$$

$$\sigma(\mathbf{u}) \cdot \mathbf{n} = g \qquad \text{sur} \quad \partial\Omega_N \tag{4.7}$$

où $\mathbf{n}$ est la normale extérieure au domaine et $\mathcal{I}$ la matrice d'identité. La formulation faible en déplacement associée au problème est :

$$\forall \mathbf{v} \in (H_0^1)^3, \ \int_\Omega \lambda \, \nabla \cdot \mathbf{u} \, \nabla \cdot \mathbf{v} \ + \ 2 \, \mu \, \epsilon(\mathbf{u}) : \epsilon(\mathbf{v}) \, \mathrm{d}x = \int_\Omega \mathbf{f} \cdot \mathbf{v} \, \mathrm{d}x + \int_{\partial\Omega_N} \mathbf{g} \cdot \mathbf{v} \, \mathrm{d}s \tag{4.8}$$

où $\epsilon(\mathbf{u}) = \frac{1}{2}(\nabla \cdot \mathbf{u} + \nabla \cdot \mathbf{u}^T)$ et $H_0^1 = \{v \in H^1 \ / \ v = 0 \text{ sur } \Omega_D\}$. Pour plus de détails sur la construction de la formulation faible, le lecteur peut se reporter à [EG02].

Extension et cisaillement d'un carré unité. Nous discutons simultanément du cas d'extension et du cas de cisaillement car nous obtenus des résultats similaires

avec les deux expériences.

Le premier cas (haut de la Figure 4.4) est constitué d'une plaque (carré unité) attachée sur la partie inférieure, *c.à-d.* conditions de Dirichlet nulle, soumise à une traction uniforme sur la partie supérieure, *c.à-d.* conditions de Neumann constante. Pour le second cas (bas de la Figure 4.4), la plaque est fixée sur la gauche et une traction uniforme dirigée vers le bas est appliquée sur la partie droite.

Les erreurs L^2 sur les champs de déplacement calculés avec différents maillages sont reportées sur les courbes de la Figure 4.4. Dans les deux cas, nous observons que :

— les maillages triangulaires obtenus par décomposition de quadrilatères en deux triangles sont très mauvais. Dans le cas de l'extension, nous pouvons même observer un déplacement horizontal complètement parasite qui est dû au biais directionnel dans le maillage triangulaire ;

— à l'ordre un, les maillages quadrilatéraux, quadrilatéraux tournés et triangulaires (répartition aléatoire) ont des précisions équivalentes ;

— à l'ordre deux, les maillages triangulaires (répartition aléatoire) sont les plus précis, suivi des éléments quadrilatéraux tournés et finalement des quadrilatéraux normaux (dont les arêtes sont alignées avec l'effort).

Ainsi, nous n'observons pas d'avantage des quadrilatères sur les triangles avec des éléments finis classiques $(\mathbb{P}_k, \mathbb{Q}_k)$ sur ces deux problèmes très simples d'élasticité linéaire. Cependant, nous n'avons pas eu l'occasion de comparer les performances des éléments quadrilatéraux modifiés comme les éléments hybrides ou ceux avec des modes incompatibles (§2.2.1.1, p. 54), or ceux-ci sont largement conseillés et utilisés dans les logiciels de simulation numérique commerciaux.

4.1.3 Résumé des observations

Des observations effectuées sur les problèmes 2D elliptiques simples présentées précédemment, nous pouvons retenir quelques tendances sur les différences de précision entre les différents types de maillages :

— les maillages avec des triangles répartis aléatoirement se comportent bien mieux que ceux avec des triangles obtenus par décomposition de quadrilatères. Ceci s'explique par le fait que les triangles réguliers ont des directions redondantes qui introduisent un biais directionnel sur l'approximation ;

— l'alignement des éléments (arêtes des quadrilatères) avec les variations de la solution n'est pas forcément optimal. Au contraire, nous avons observé dans la majorité des cas des erreurs plus faibles avec des éléments quadrilatéraux tournés de 45 degrés, c'est-à-dire dont les diagonales sont alignées avec les variations de la solution ;

— l'erreur L^2 peut être minimisée par des solutions qui oscillent localement et dont l'allure visuelle n'est à priori pas très satisfaisante. C'est le cas des solutions $\mathbb{Q}_1$-tournés sur la Figure 4.2 et sur la figure Figure 4.3 : visuellement, elles ont l'air moins intéressantes que les solutions $\mathbb{Q}_1$, qui sont lisses, mais pourtant plus précises.

Comme les problèmes étudiés sont extrêmement simples, nous ne prétendons pas que ces remarques se généralisent à des problèmes plus compliqués. Cependant, pour

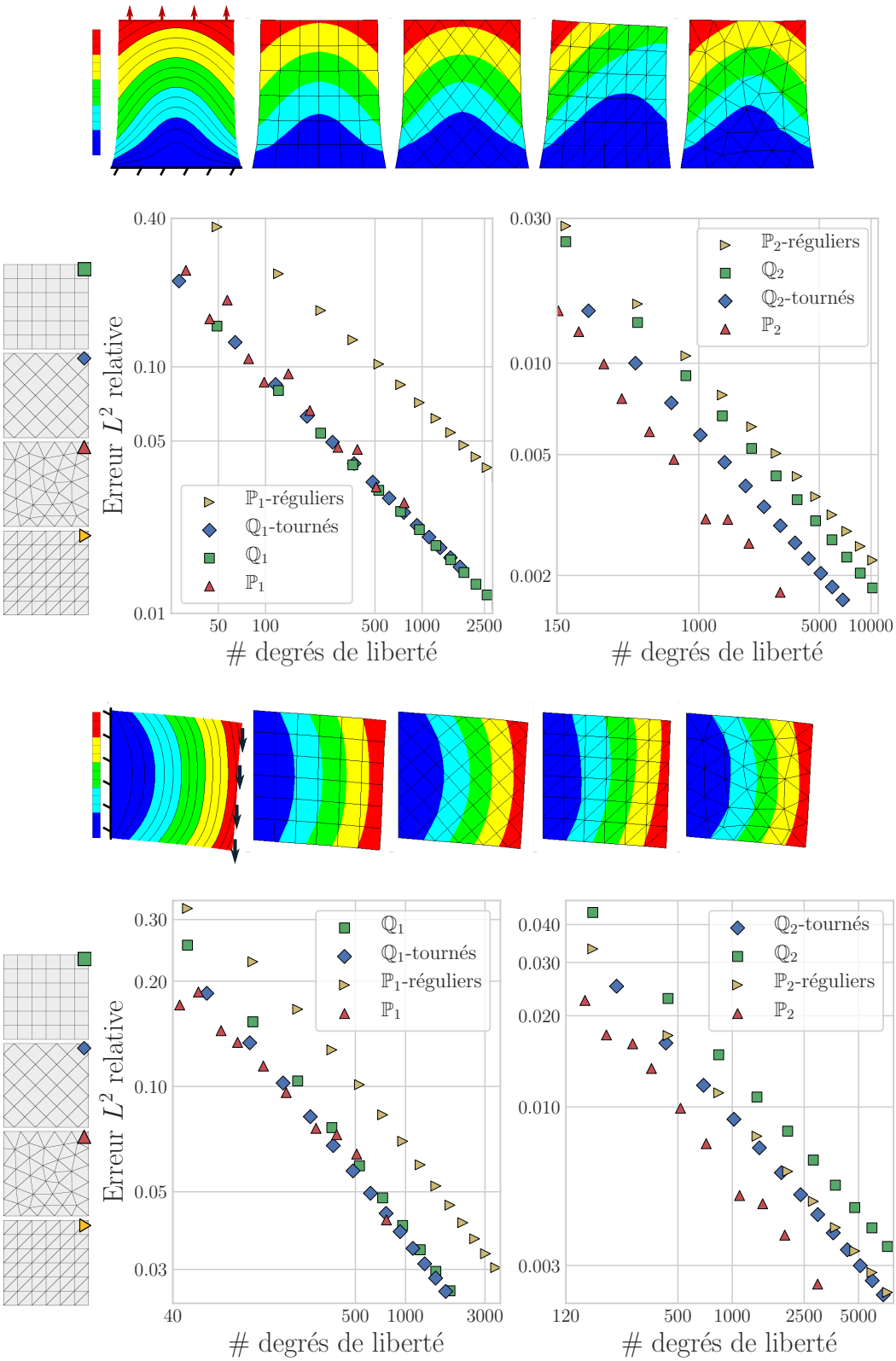


FIGURE 4.4 – Problèmes d'élasticité linéaire : extension (moitié haute) et cisaillement (moitié basse) du carré unité. *Courbes :* Erreur relative, en norme L^2, en fonction du nombre de degrés de liberté. Éléments finis d'ordre un (gauche) et d'ordre deux (droite).

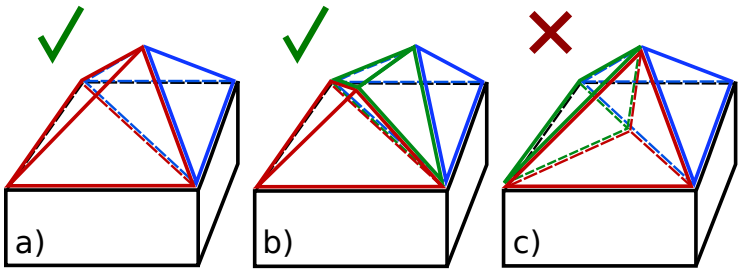


FIGURE 4.5 – Exemples de jonctions non-conformes entre hexaèdres et tétraèdres autorisées ou non.

la suite, il apparait comme très important de comparer les maillages hexaédriques (ou hex-dominants) à de vrais maillages tétraédriques et non à des décompositions d'hexaèdres en tétraèdres. De plus, l'idée qu'il est avantageux d'avoir des éléments hexaédriques alignés avec le bord, ou avec les variations de la solution, n'est pas évidente et mériterait d'être approfondie en détaillant les phénomènes numériques en jeu.

4.2 Simulation éléments finis sur maillages hexa-tétraédriques

Dans cette section, nous proposons de construire un espace d'approximation continu ($V_h \subset H^1$) pour les maillages hexa-tétraédriques non-conformes. Par rapport à la littérature existante sur le sujet (§2.2.2.2, p. 66), notre proposition tient compte des transformations trilinéaires pour la géométrie des hexaèdres et ne se restreint pas à des interfaces planes entre hexaèdres et tétraèdres.

Contexte historique. Nous avons proposé la construction pour l'ordre un dans l'article

> *Maxence* REBEROL *et Bruno* LÉVY. *« Low-order continuous finite element spaces on hybrid non-conforming hexahedral-tetrahedral meshes ». 2016*

soumis en février 2016 à la conférence *Solid and Physical Modeling (SPM)*, mais non retenu. L'article [ADS16], publié à la fin du mois de février 2016, propose la construction d'un espace très similaire en utilisant les bases de Bernstein-Bézier et la décomposition des pyramides en deux tétraèdres, *c.à-d.* pyramides composites dans l'état de l'art (§2.2.2.1, p. 63). Notre proposition, [RL16] pour l'ordre un et ci-dessous pour un ordre arbitraire, est légèrement plus flexible car nous ne supposons pas que les tétraèdres des interfaces non-conformes forment une pyramide. Ceci permet de gérer le cas au centre de la Figure 4.5, contrairement à AINSWORTH et al. [ADS16] qui se limitent au cas sur la gauche de la figure.

4.2.1 Conformité géométrique des maillages hexa-tétraédriques

Nous souhaitons pouvoir effectuer des simulations avec la méthode des éléments finis sur les maillages hex-dominants produits avec le mailleur décrit dans le *Chapitre 3*.

Par défaut, ceux-ci sont des maillages hexa-tétraédriques où l'interface non-conforme est restreinte à deux faces triangulaires qui se connectent sur une face quadrilatérale d'hexaèdre. Les tétraèdres à l'interface ne peuvent pas toujours être recombinés en pyramides (gauche de la Figure 4.5) car il arrive très fréquemment de tomber sur la configuration au centre de la Figure 4.5, où un tétraèdre est connecté à la face quadrilatérale sur sa diagonale uniquement. Il est possible d'ajouter des contraintes sur le mailleur pour former des pyramides aux interfaces, mais ceci conduit à une proportion plus faible d'hexaèdres, ce que nous voulons éviter.

Formellement, les maillages pris en charge par notre approche sont spécifiés par la *Définition 1*. Chaque maillage est composé d'un ensemble de points (information géométrique) et d'un ensemble d'éléments définis par leurs sommets et leurs faces (information combinatoire).

Définition 1. *Spécification combinatoire des maillages hexa-tétraédriques.*

Le maillage hybride $\mathcal{M}$ est composé d'un ensemble de sommets $\mathcal{P}$, définis par leurs coordonnées, d'un ensemble de tétraèdres, définis par leurs quatre sommets, et d'un ensemble d'hexaèdres, définis par leurs huit sommets et leurs six faces (chacune définie par quatre sommets). La connectivité entre éléments est restreinte aux cas combinatoires suivants :

— deux tétraèdres partagent 0, 1, 2 ou 3 sommets ;

— deux hexaèdres partagent 0, 1, 2 ou 4 sommets. Quand ils partagent 2 sommets, il s'agit d'une arête commune. Quand ils partagent 4 sommets, il s'agit d'une face commune ;

— un hexaèdre et un tétraèdre partagent 0, 1, 2 ou 3 sommets. Quand ils partagent 3 sommets, il existe un autre tétraèdre qui partage également 3 sommets avec l'hexaèdre et 2 ou 3 sommets avec le tétraèdre. Dans cette configuration, la face de l'hexaèdre est connectée à 2 faces de tétraèdres. Quand ils partagent 2 sommets, il s'agit d'une arête commune ou d'une diagonale de face d'hexaèdre.

Cette définition est principalement combinatoire et elle ne spécifie pas la géométrie des éléments. Si nous utilisons les transformations naturelles, affines pour les tétraèdres et trilinéaires pour les hexaèdres, les interfaces entre hexaèdres et tétraèdres ne sont pas géométriquement conformes lorsque les quatre sommets de la jonction ne sont pas coplanaires. Par conséquent, le maillage obtenu ne forme pas une partition satisfaisante de l'espace. Il existe des endroits où les éléments se superposent, comme illustré sur la gauche de la Figure 4.7. Les faces des tétraèdres affines sont des plans (déterminés par trois sommets) tandis que les faces hexaèdres sont des surfaces quadratiques (paramétrisation bilinéaire).

Il est possible de retrouver la continuité géométrique des interfaces entre éléments en utilisant des transformations quadratiques (Figure 4.6) pour les tétraèdres connectés à un hexaèdre par une face ou par une arête sur la diagonale du quadrilatère. La nouvelle géométrie des tétraèdres aux interfaces non-conformes est illustrée sur la figure droite de la Figure 4.7.

Plus précisément, nous considérons que les transformations quadratiques des tétraèdres sont déterminées par la position des quatre sommets et des six milieux d'arêtes, en utilisant la base interpolante de Lagrange de $\mathbb{P}_2$ (Figure 4.6). Pour obtenir la conformité géométrique, il faut que les degrés de liberté de la transformation quadratique sur l'interface soient sur la face quadrilatérale bilinéaire. Formellement :

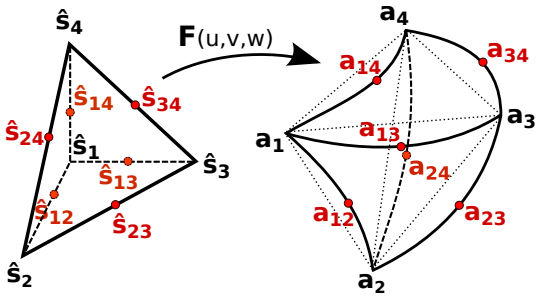


FIGURE 4.6 – Transformation quadratique du tétraèdre de référence

Proposition 1. *Conformité géométrique aux interfaces non-conformes.*

Soit un tétraèdre avec géométrie quadratique connecté par trois sommets a_1, a_2, a_4 à une face quadrilatérale a_1, a_2, a_3, a_4 d'un hexaèdre à géométrie trilinéaire, avec $a_2 - a_4$ la diagonale du quadrilatère. On note a_{12}, a_{14}, a_{24} les degrés de liberté de la transformation quadratique situés aux milieux des arêtes $(1-2)$, $(1-4)$ et $(2-4)$ (Figure 4.7.b.).

Si $a_{12} = \frac{1}{2}(a_1 + a_2)$, $a_{14} = \frac{1}{2}(a_1 + a_4)$ et $a_{24} = \frac{1}{4}(a_1 + a_2 + a_3 + a_4)$, alors la face quadratique $(1-2-4)$ du tétraèdre et la face bilinéaire $(1-2-3-4)$ de l'hexaèdre coïncident géométriquement.

Démonstration. Dans les bases de Lagrange de $\mathbb{P}_k$ et $\mathbb{Q}_k$, les restrictions des polynômes aux faces sont entièrement déterminées par les degrés de liberté sur les faces. La géométrie de la face quadratique $(1-2-4)$ est définie par un polynôme de $\mathbb{P}_2^{2D}$ entièrement déterminé par six points : les trois sommets et les trois milieux d'arêtes. La géométrie de la face quadrilatérale est déterminée par un polynôme $\mathbb{Q}_1^{2D}$. Or $\mathbb{Q}_1^{2D} \subset \mathbb{P}_2^{2D}$, donc on peut réécrire le polynôme en utilisant une base interpolante de $\mathbb{P}_2^{2D}$: les six points de contrôle associés au triangle quadratique $(1-2-4)$. Par la définition combinatoire du maillage, on sait déjà que les deux polynômes partagent les trois sommets a_1, a_2 et a_4. Il suffit d'ajouter les contraintes $a_{12} = \frac{1}{2}(a_1 + a_2)$, $a_{14} = \frac{1}{2}(a_1 + a_4)$ et $a_{24} = \frac{1}{4}(a_1 + a_2 + a_3 + a_4)$ pour que les deux polynômes, paramétrisation de la face triangulaire et paramétrisation de la face quadrilatérale, soient identiques. □

De la même manière, si un tétraèdre est connecté à quadrilatère uniquement sur sa diagonale, il faut utiliser une transformation quadratique et assigner le degré de liberté, au milieu de l'arête, à la position centrale de la face quadrilatérale (bilinéaire) : la moyenne des quatre sommets.

Nous identifions trois types d'éléments géométriques présents dans le maillage hexa-tétraédrique, avec la terminologie empruntée à AINSWORTH et al. [ADS16] :

— les hexaèdres, dont les transformations sont trilinéaires ($\mathcal{M} \in \mathbb{Q}_1^3$) ;

— les tétraèdres ordinaires, dont les transformations sont affines ($\mathcal{M} \in \mathbb{P}_1^3$)). Ceux-ci ne sont pas connectés aux hexaèdres par des faces ou par une diagonale de quad ;

— les tétraèdres d'interface, dont les transformations sont quadratiques ($\mathcal{M} \in \mathbb{P}_2^3$). Ceux-ci sont connectés à au moins un hexaèdre par une face ou par une diagonale de quad. Les points de contrôle de l'arrête associée à la diagonale d'un quad sont déplacés sur le milieu du quad (moyenne des quatre sommets).

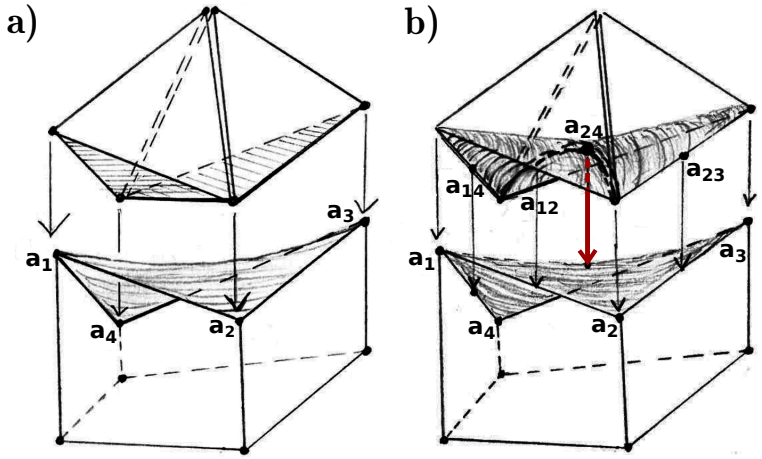


FIGURE 4.7 – Jonction non-conforme entre un hexaèdre et deux tétraèdres. a) Transformations tétraédriques affines. b) Transformations tétraédriques quadratiques.

Il est important de remarquer que le maillage contient des arêtes non conformes aux interfaces entre hexaèdres et tétraèdres. Ces arêtes sont des arêtes de tétraèdres d'interface mais ce ne sont pas des arêtes d'hexaèdres, elles correspondent à la diagonale des faces quadrilatères. Lorsque la face associée de l'hexaèdre n'est pas plane, la géométrie de ces arêtes est paramétrée par un polynôme d'ordre deux.

Remarque. À notre connaissance, au moment de l'écriture de REBEROL et al. [RL16], il n'y avait pas d'article avec maillages hexa-tétraédriques (§2.2.2.2, p. 66) qui s'était intéressé à la conformité géométrique des interfaces non-conformes, même si l'idée ci-dessus est très naturelle. Publié simultanément, AINSWORTH et al. [ADS16] utilise également des transformations quadratiques des tétraèdres de la pyramide pour assurer la conformité. Cependant ceux-ci sont exprimés dans la base de Bernstein-Bézier au lieu de la base de Lagrange dans notre cas. En analyse iso-géométrique, ZENG et al. [ZC15] propose également des contraintes pour imposer la conformité géométrique entre des tétraèdres avec géométrie de Bézier et des hexaèdres formés à partir de B-splines extrudés.

4.2.2 Espace d'approximation continu

Nous souhaitons maintenant former un espace d'approximation continu, noté $\mathcal{H}yb_k$, afin d'approximer les fonctions de H^1 et de pouvoir réaliser des simulations avec la méthode des éléments finis. Nous rappelons les définitions des espaces d'approximations pour les maillages tétraédriques et hexaédriques introduites précédemment (1.2.2, p. 1.2.2) :

$$V_h^T = \{f \in \mathcal{C}^0(\Omega_h) \mid f_{|K_e} = \hat{f}_e \circ \mathcal{M}_e^{-1},\ \hat{f}_e \in \mathbb{P}_k,\ \forall K_e \in \Omega_h\}$$
$$V_h^H = \{f \in \mathcal{C}^0(\Omega_h) \mid f_{|K_e} = \hat{f}_e \circ \mathcal{M}_e^{-1},\ \hat{f}_e \in \mathbb{Q}_k,\ \forall K_e \in \Omega_h\}$$

avec les transformations géométriques $\mathcal{M}_e \in \mathbb{P}_1$ pour les maillages tétraédriques et $\mathcal{M}_e \in \mathbb{Q}_1$ pour les maillages hexaédriques.

Dans notre situation, maillages hexa-tétraédriques avec des tétraèdres ordinaires (géométrie affine) et des tétraèdres d'interface (géométrie quadratique), nous définissons un espace d'approximation qui tient compte des trois types d'éléments :

Définition 2. *Espace d'approximation continu $\mathcal{H}yb_k$ d'ordre k :*

$$\begin{aligned} Hyb_k = \{ f \in C^0 \mid & f_{|H} = \hat{f}_H \circ \mathcal{M}_H^{-1},\ \hat{f}_H \in \mathbb{Q}_k, \ \text{ si } H \text{ hexaèdre} : \mathcal{M}_H \in \mathbb{Q}_1 \\ & f_{|T} = \hat{f}_T \circ \mathcal{M}_T^{-1},\ \hat{f}_T \in \mathbb{P}_k, \ \text{ si } T \text{ tétraèdre ordinaire} : \mathcal{M}_T \in \mathbb{P}_1 \\ & f_{|I} = \hat{f}_I \circ \mathcal{M}_I^{-1},\ \hat{f}_I \in \mathbb{P}_{2k}, \ \text{ si } I \text{ tétraèdre d'interface} : \mathcal{M}_I \in \mathbb{P}_2 \} \end{aligned}$$

Il est nécessaire d'utiliser des polynômes d'ordre $2k$ dans les tétraèdres d'interface car les restrictions des polynômes $\mathbb{Q}_k$ des hexaèdres sur les interfaces non-conformes sont des polynômes de degré $2k$ (produit de deux polynômes 1D de degré k)[24]. Formellement, la continuité aux interfaces non-conformes peut être démontrée de la même manière que pour la continuité géométrique. Les polynômes ($\mathbb{Q}_k^{2D}$) des hexaèdres sur les faces sont réécrits dans la base de Lagrange ($\mathbb{P}_{2k}^{2D}$) des restrictions des polynômes des tétraèdres d'interface sur leurs faces et les degrés de liberté sont choisis identique. En pratique, il est plus simple d'assurer la continuité en utilisant des contraintes linéaires entre les degrés de liberté des différents éléments.

L'introduction des éléments $\mathbb{P}_{2k}$ n'ajoute pas, à part involontairement, de nouvelles variables non-contraintes par rapport à un espace (non continu) $\mathbb{Q}_k - \mathbb{P}_k$. Les degrés de liberté supplémentaires introduits par $\mathbb{P}_{2k}$ sont contraints par continuité avec les hexaèdres $\mathbb{Q}_k$ et les tétraèdres ordinaires $\mathbb{P}_k$ adjacents. Les nouveaux degrés de liberté contraints sont illustrés en vert sur la Figure 4.8 pour l'ordre un. D'une certaine façon, nous avons introduit des degrés de liberté artificiels uniquement pour *réparer* la continuité.

Lorsqu'une arête, ainsi que face pour $k \geq 2$ et volume intérieur pour $k \geq 3$, n'appartient qu'à des éléments d'interfaces, de nouveaux degrés de libertés sont introduits et ceux-ci ne sont pas contraints par la continuité. Ils ne servent ni à assurer la continuité, ni à assurer la convergence d'ordre k. Par conséquent, il serait possible de les supprimer en ajoutant la condition :

$f_{|E} \in \mathbb{P}_k$ pour chaque entité E n'appartenant pas géométriquement à un hexaèdre.

Ceci dit, ces contraintes supplémentaires ne sont pas forcément très utiles en termes de performances car la proportion de tétraèdres d'interface est censée être faible.

Bases de fonctions. Afin de réaliser une implémentation de cet espace d'approximation, il peut être utile de définir une base $(\phi_i)_{i=1..N}$ de l'espace $\mathcal{H}yb_k$, où les fonctions ϕ_i sont associées aux réels degrés de liberté (non contraints). Cependant, $\mathcal{H}yb_k$ étant construit comme l'union d'espaces plus gros avec des contraintes de continuité, la construction explicite d'une base peut s'avérer très fastidieuse et d'une utilité pratique limitée compte tenu de la complexité des formules. Dans Reberol et al. [RL16], nous avons présenté une définition explicite d'une base construite à partir de combinaisons de polynômes de Lagrange pour l'espace d'ordre un $\mathcal{H}yb_1$, mais nous n'avons pas développé d'expressions explicites pour les ordres supérieurs.

24. à une bijection près. Les fonctions dans l'espace réel ne sont pas exactement des polynômes lorsque les transformations ne sont pas affines, mais l'identification peut être faite car les deux transformations bijectives sont identiques sur l'interface non-conforme (par conformité géométrique des transformations).

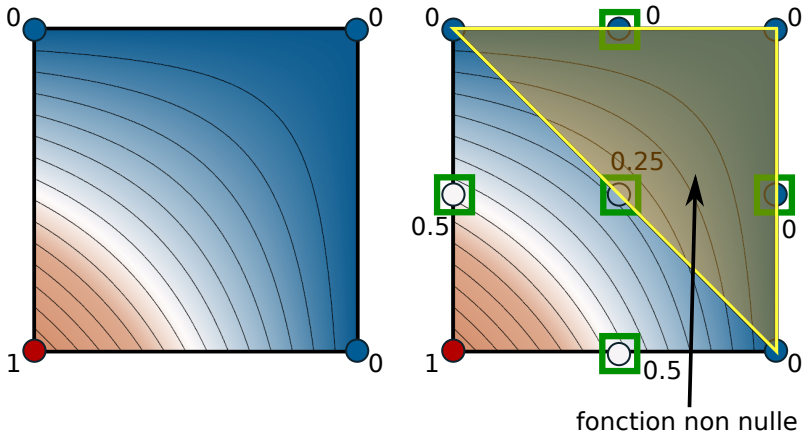


FIGURE 4.8 – Interface non-conforme entre une face d'hexaèdre $\mathbb{Q}_1$ (gauche) et deux faces triangulaires de tétraèdres $\mathbb{P}_2$ (droite). Les degrés de liberté sur les milieux d'arêtes sont contraints (carrés verts) de manière à assurer la continuité. La fonction interpolée (carte de couleur) associée au sommet en bas à gauche est non nulle sur tout l'intérieur du quadrilatère, y compris sur le triangle supérieur droit (zone jaune), qui est associé à un tétraèdre qui ne contient pas le degré de liberté non nul.

Une des difficultés principales de cette approche est que le support des fonctions de la base n'est pas restreint aux éléments contenant le degré de liberté, mais peut s'étaler sur d'autres éléments au niveau des interfaces non-conformes. Par exemple, la fonction de base associée au sommet d'une face quadrilatérale est non nulle sur l'ensemble de l'intérieur de la face (gauche de la Figure 4.8). Par continuité, cette fonction doit être non nulle sur les deux triangles associés à la face quadrilatérale (droite de la Figure 4.8), y compris sur le triangle dont tous les degrés de liberté non-contraints sont nuls. L'expression des fonctions de la base devient particulièrement compliquée lorsqu'il faut gérer les tétraèdres qui ne sont connectés aux hexaèdres que par la diagonale.

La seconde difficulté apparait pour des ordres supérieurs à trois car la répartition spatiale des degrés de liberté de Gauss-Lobatto des polynômes d'ordre k n'est pas un sous-ensemble des degrés de liberté des polynômes d'ordre $2k$. Par conséquent, il y a beaucoup de contraintes linéaires entre variables à exprimer et elles ne peuvent pas être trivialement supprimées du système.

Implémentation par matrices de contraintes. L'approche que nous avons préférée suivre s'inspire des techniques employées avec des maillages non conformes (*par ex.* AMR) : un espace sans contraintes de continuité est construit et un système de contraintes est formé de manière à garantir la continuité C^0 des fonctions aux interfaces non conformes. De ce système de contraintes est déduit une matrice de prolongation ainsi qu'une matrice de restriction, qui permettent de transformer le système linéaire calculé initialement.

Pour construire le système linéaire non contraint, on réalise l'assemblage sur chacun des éléments. Sur chaque entité (arête, face) partagée par deux éléments de types différents, on utilise des degrés de liberté différents. Il en résulte un système

linéaire de type :

$$A\mathbf{v}_{nc} = \mathbf{b}$$

avec $\mathbf{v}_{nc}$ l'ensemble des variables non contraintes du problème. Comme il y a plusieurs jeux de degrés de liberté sur les entités partagées, ce système n'impose pas la continuité aux interfaces entre éléments de types différents.

Pour chaque arête et face partagées par deux jeux de degrés de liberté, on exprime les contraintes linéaires nécessaires à la continuité : la valeur de la variable contrainte $v_{nc,i}$ (degré de liberté artificiel de $\mathbb{P}_{2k}$) est égale à la combinaison linéaire des variables non contraintes $v_{m,j}$ (de $\mathbb{P}_k$ ou $\mathbb{Q}_k$) pondérées par la valeur de leurs fonctions de formes à la position $\mathbf{x}_{nc,i}$ du degré de liberté contraint :

$$v_{nc,i} = \sum_j v_{m,j}\ \phi_j(\mathbf{x}_{nc,i})$$

Avec toutes ces contraintes, on peut former une matrice de prolongation P rectangulaire qui permet de lier les variables non contraintes aux variables libres :

$$\mathbf{v}_{nc} = P\ \mathbf{v}_{libre}$$

Le système linéaire contraint qui correspond à l'espace d'approximation continu $\mathcal{H}yb_k$ peut alors être écrit sous la forme :

$$P^T AP\ \mathbf{v}_{libre} = P^T\ \mathbf{b}$$

Nous avons réalisé cette implémentation en modifiant la librairie d'éléments finis *MFEM* [Mfe17]. Celle-ci gère nativement les maillages tétraédriques et les maillages hexaédriques, mais pas les maillages contenant plusieurs types d'éléments ni plusieurs ordres polynomiaux dans le même maillage.

4.2.3 Validation et expériences numériques

Afin de valider notre approche et notre implémentation, nous analysons l'erreur de la solution approchée obtenue sur des problèmes où l'on connait la solution analytique (§4.2.3.1). La seconde partie (§4.2.3.2) met en oeuvre une expérience montrant l'importance des transformations quadratiques pour les tétraèdres d'interface, car sans elles l'approximation ne converge pas correctement vers la solution avec le raffinement du maillage.

Des résultats numériques sur des maillages plus réalistes sont traités dans la section suivante (§4.3), où l'on compare différents maillages sur des géométries plus compliquées.

4.2.3.1 Validation par la méthode des solutions manufacturées

L'expérience qui suit est l'analogue en dimension trois du problème de la bosse sinusoïdale introduit précédemment ((4.1), p. 99) pour comparer différents maillages. Le domaine est le cube unité ($\Omega = [0,1]^3$), l'équation est celle de Poisson $-\Delta u = f$ et la solution analytique est : $u(x,y,z) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$.

Nous utilisons trois types de maillages : entièrement hexaédriques, entièrement tétraédriques (décomposition des hexaèdres en six tétraèdres) et hexa-tétraédriques.

Les maillages hexa-tétraédriques sont formés à partir de décompositions d'hexaèdres en six tétraèdres et en quinze tétraèdres. La seconde décomposition permet d'introduire la configuration où un tétraèdre est connecté à un hexaèdre uniquement sur la diagonale d'une face. Les maillages hexa-tétraédriques comportent une proportion volumique d'environ 20% de tétraèdres, ce qui est pessimiste par rapport aux proportions dans les maillages hex-dominants générés automatiquement, où la proportion est plutôt de l'ordre de $5-10\%$. Dans tous les maillages, les points à l'intérieur sont décalés aléatoirement d'une amplitude allant jusqu'à $d = 10\%$ de la taille des arêtes, de sorte que les faces des hexaèdres ne soient pas planes.

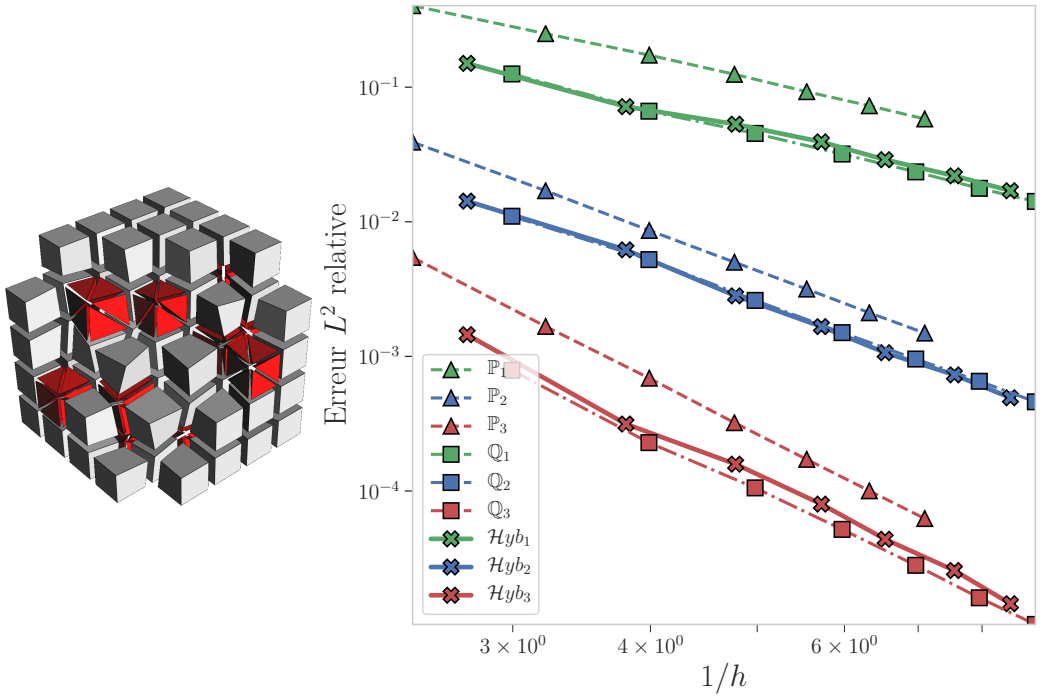


FIGURE 4.9 – Validation sur la *bosse sinusoïdale*. *Gauche :* Exemple de maillage hex-tet du cube unité, avec déplacement des points intérieurs. *Droite :* Convergence de l'approximation éléments finis avec différents maillages ($\mathbb{P}_k, \mathbb{Q}_k$ et $\mathcal{H}yb_k$), pour les ordres un, deux et trois, en fonction de l'inverse de la taille moyenne des arêtes ($1/h$).

Les résultats de simulation sont affichés sur la Figure 4.9. On observe notamment que l'erreur de la base $\mathcal{H}yb_k$ est proche de celle des maillages hex $\mathbb{Q}_k$, ce qui est encourageant puisque cela indique que l'introduction de tétraèdres dans le maillage hybride ne dégrade pas significativement le gain par rapport aux maillages tétraédriques. Sur cet exemple simple (non représentatif de problèmes réels), les maillages hexa-tétraédriques sont environ $2-3$ fois plus précis que les maillages tétraédriques pour le même nombre de degrés liberté.

Nous pouvons également observer la convergence optimale de la solution prévue par la théorie : convergence en $1/h^{k+1}$ pour la norme L^2, sachant que la solution est lisse. Dans l'exemple ci-dessus, nous avons utilisé la longueur moyenne des arêtes comme approximation de la taille caractéristique h des éléments.

4.2.3.2 Importance des transformations quadratiques

D'un point de vue numérique, l'utilisation de transformations quadratiques pour les tétraèdres d'interface (connectés aux faces bilinéaires des hexaèdres) pourrait être considérée comme superflue comme il y a déjà d'autres sources d'erreurs (*par ex.* discrétisation, quadratures).

Pour vérifier cette hypothèse, nous avons effectué les mêmes simulations en remplaçant les transformations quadratiques par des transformations affines. Les résultats sont reportés sur la Figure 4.10. On observe que cette modification introduit des erreurs significatives, qui deviennent dominantes avec l'augmentation de la précision : non-convergence vers la solution analytique avec le nombre de degrés de liberté. L'erreur introduite est d'autant plus importante que les faces bilinéaires sont non planaires : un décalage aléatoire des points intérieurs du cube jusqu'à $d = 20\%$ de la taille d'arête introduit une perte de convergence bien plus importante qu'un décalage de moins de $d = 10\%$.

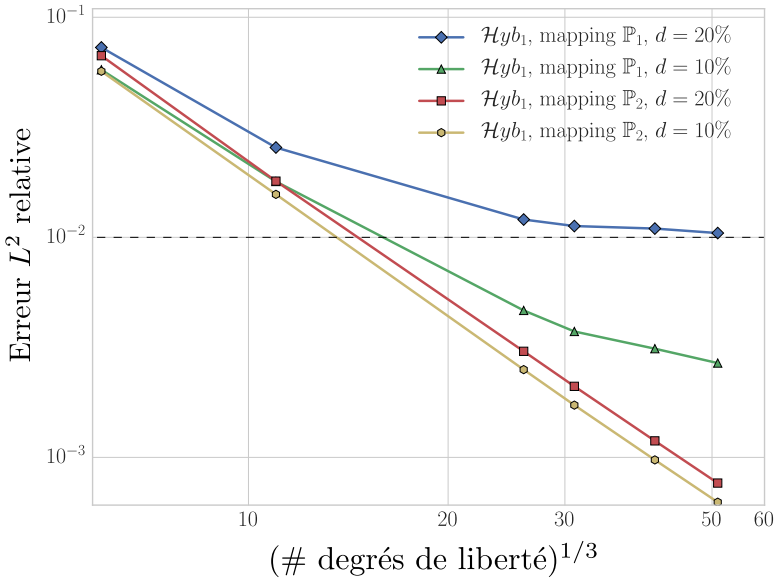


FIGURE 4.10 – Problème analytique du *sine bump* (Poisson). Influence des transformations quadratiques pour les tétraèdres d'interface. Les points à l'intérieur du cube sont déplacés aléatoirement d'une amplitude variant jusqu'à $d = 10\%$ et $d = 20\%$ de la longueur des arêtes. Lorsque des transformations affines sont utilisées pour les tétraèdres d'interface, la convergence est perdue, *c.à-d.* que la méthode des éléments finis ne converge pas vers la bonne solution.

Nous retenons de cette expérience qu'il est important d'utiliser des transformations quadratiques pour les tétraèdres d'interface. C'est d'autant plus vrai que les faces d'hexaèdres sont non-planaires. Or il est courant que les mailleurs hex-dominants génèrent des hexaèdres avec des faces fortement non-planaires.

4.2.3.3 Conclusions

La construction d'un espace d'approximation continu pour maillages hexa-tétraédriques, que nous venons de présenter, fonctionne correctement : erreur d'approximation équi-

valente à celle des maillages hexaédriques et tétraédriques et convergence optimale avec le raffinement. Elle pourrait probablement être étendue aux espaces $H(div)$ et $H(rot)$ sans trop de difficultés.

Cependant, nous pensons qu'elle n'a pas un grand intérêt pratique car sa mise en oeuvre pose un certain nombre de difficultés techniques que nous avons découvert au fur et à mesure de son développement, à savoir :

— la structure de données du maillage non-conforme hex-tet, particulièrement avec des tétraèdres qui peuvent se connecter à la diagonale d'une face d'hexaèdre, n'est pas courante. Dans la plupart des codes de calcul, il faut modifier les structures gérant le maillage ;

— il faut pouvoir gérer des polynômes d'ordres différents (k et $2k$) en même temps, ainsi que deux jeux de degrés de liberté distincts sur les mêmes entités géométriques (*par ex.* arêtes des interfaces non-conformes). Seuls les codes prévus pour les maillages non-conformes (*par ex.* hp-FEM) permettent de le faire sans trop de modifications ;

— il faut appliquer des matrices de prolongation et de restriction au système pour passer des variables libres aux variables contraintes, ce qui a un coût non négligeable pour des gros systèmes linéaires. Cette approche rend également difficile le développement d'approches *sans matrices*, où les coefficients sont calculés à la volée.

Pour ces raisons, nous pensons qu'il est préférable d'utiliser des éléments finis pyramidaux. Ceux-ci s'intègrent plus facilement dans un code calcul : maillage conforme, assemblage classique, continuité naturelle via les degrés de liberté des faces (pas de contraintes entre les variables). En contrepartie, il faut contraindre (a) le mailleur hex-dominant à générer des pyramides, et donc à interdire la configuration au centre de la Figure 4.5, et (b) gérer les difficultés associées aux pyramides (fonctions rationnelles, quadratures spécifiques).

4.3 Comparaisons de maillages sur des domaines compliqués

Dans la première partie (§4.1) du chapitre, nous avons observé des différences de précision entre maillages triangulaires et quadrilatérales, avec différentes orientations, sur des problèmes simples définis sur le carré unité. Dans cette partie, nous réalisons des comparaisons sur des modèles 3D plus compliqués.

L'objectif est de comparer les performances, en termes de précision et de temps de calcul, de différents types de maillages :

— tétraédriques construits à partir de *mmgs* [Mmg17] (remaillage surfacique) et *TetGen* [Si15b] (maillage tétraédrique) ;

— hex-dominants (hexa-tétraédriques) construits avec notre approche [Ray+17] (*Chapitre 3*) ;

— hexaédriques alignés avec le bord à partir de décompositions régulières d'un maillage hexaédrique existant (pour le modèle *hanger* [Liv+15]) ;

— hexaédriques-octree construits avec le logiciel *Bolt*[25] , basé sur la méthode *octree* [Mar09].

Pour les maillages tétraédriques et hexaédriques, nous utilisons les espaces polynomiaux standards $\mathbb{P}_k$ et $\mathbb{Q}_k$ (détaillés dans le *chapitre 1*) et l'implémentation associée dans la librairie *mfem* [Mfe17]. Sur les maillages hexa-tétraédriques non-conformes, nous utilisons les espaces conformes $\mathcal{H}yb_k \subset H^1$ que nous avons introduits précédemment (§4.2) et dont nous avons réalisé l'implémentation en modifiant *mfem*.

Pour calculer l'erreur associée à chaque approximation obtenue avec la méthode des éléments finis, nous calculons la distance à une solution de référence en utilisant la méthode d'échantillonnage efficace que nous développons dans le *Chapitre 5*. La solution de référence est en général au moins un ordre de grandeur plus précise que les solutions observées, typiquement nous utilisons des solutions $\mathbb{P}_4$ calculées sur un maillage très fin.

Nous présentons deux types de courbes de convergence : l'erreur en fonction de la résolution du maillage et l'erreur en fonction du temps de calcul.

— La résolution est caractérisée par h_{max}, la longueur maximale des arêtes du maillage. Il s'agit d'une approximation grossière du diamètre maximal des boules englobant les éléments, qui apparaissent dans les propositions d'estimations d'erreurs. Ce paramètre de résolution a peu de sens dans le cas de maillages comportant des éléments de tailles différentes (maillage adaptatif) et il n'est pas forcément représentatif du coût de calcul non plus. Nous nous en servons surtout pour se faire une première idée du comportement de la séquence de maillages et pour observer l'influence de l'ordre des polynômes sur les erreurs, à résolution constante.

— Le temps de calcul est décrit par la durée de convergence du solveur itératif. Nous n'incluons pas le temps d'assemblage parce que celui-ci est généralement court devant le temps de résolution du système linéaire et parce que notre assemblage dans le cas des maillages hexa-tétraédriques est particulièrement lent (la recherche et l'application des contraintes n'est pas optimisée).

Remarque. En dimension trois et avec des modèles non-triviaux, le choix du solveur itératif est fondamental pour converger et obtenir des performances correctes. D'après notre expérience, l'utilisation du gradient conjugué avec un pré-conditionneur basique (Jacobi, Gauss-Seidel) ne permet pas de résoudre des problèmes d'élasticité linéaire avec des polynômes d'ordre deux ou plus.

Nous avons eu le plus de succès avec le solveur multigrille algébrique *BoomerAMG* [HY02] de la librairie *hypre* [FY02]. En particulier, il est possible de spécifier que la solution est un champ vectoriel à plusieurs composantes dans le cas de l'élasticité, ce qui est très important pour que le solveur converge efficacement.

4.3.1 Problème de Poisson

À la différence des premiers essais (§4.1), nous étudions un problème de Poisson avec terme source nul mais avec des conditions aux limites de Dirichlet et de Neumann :

25. http://www.csimsoft.com/boltoverview

$$\begin{cases} -\Delta u = 0 & \text{dans } \Omega \\ u = u_D & \text{sur } \partial\Omega_D \\ \nabla u \cdot \mathbf{n} = u_N & \text{sur } \partial\Omega_N \end{cases} \tag{4.9}$$

Ce type de problème est particulièrement simple pour la méthode des éléments finis car la solution est très lisse par effet du Laplacien. La solution correspond plus ou moins à un gradient constant localement ($\Delta u = 0$), qui est discrétisé fidèlement par des fonctions polynomiales par morceaux.

Metatron

Problème. Pour effectuer une première comparaison, on utilise le modèle *metatron*[26] dont la géométrie est particulièrement compliquée (Figure 4.11). La valeur de la fonction est fixée sur une extrémité (étoile inférieure) et un flux entrant non nul est imposé sur une autre extrémité (étoile supérieure).

Pour les différents types de maillages, nous construisons des séquences avec des résolutions de plus en plus fines, où les éléments sont de tailles homogènes dans le modèle et où le bord du maillage respecte une distance de Hausdorff faible ($\epsilon < 0.5\%$) avec le modèle de référence.

Les conditions aux limites ne peuvent pas être discrétisées de manière identique pour les différents maillages car ils possèdent des éléments surfaciques différents sur le bord. Pour le flux non nul, nous définissons $\partial\Omega_N$ comme étant l'intersection du bord $\partial\Omega$ avec une sphère S de rayon r_0 et nous utilisons une amplitude décroissante radiale de type $(1 - r/r_0)$ de sorte que le flux soit nul sur le bord de la sphère S. Ceci permet que les différences de discrétisation de $\partial\Omega_N$ dues aux maillages aient un impact minimal.

La solution de référence est calculée sur un maillage tétraédrique très fin avec des polynômes de degré quatre. Elle contient environ vingt millions de degrés de liberté, soit dix à vingt fois plus que les solutions observées les plus fines ($\mathbb{P}_3$, $\mathbb{Q}_2$-octree) sur les courbes.

Résultats. Les résultats de convergence obtenus avec des maillages tétraédriques ($\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3$), un maillage hexaédrique-octree ($\mathbb{Q}_1, \mathbb{Q}_2$-octree) et des maillages hexa-tétraédriques ($\mathcal{H}yb_1, \mathcal{H}yb_2, \mathcal{H}yb_3$) sont présentés sur la Figure 4.11.

Sur les courbes à gauche (précision en fonction de la résolution), on observe bien qu'augmenter l'ordre des polynômes permet d'améliorer la précision, même si le gain est très faible pour le maillage hexaédrique-octree et pour le passage de l'ordre deux à trois pour les maillages hexa-tétraédriques. Ces courbes ne permettent pas réellement de comparer les différents types de maillages entre eux car la résolution est approchée par la taille maximale des arêtes, qui n'est pas forcément comparable entre les différents types de primitives.

Les courbes à droite (précision en fonction du temps de résolution) sont plus intéressantes pour comparer les performances des différents espaces d'approximation. On peut observer que les maillages hexa-tétraédriques ont de bonnes performances : ils offrent un meilleur ratio entre précision et temps de calcul. Il est également intéressant de noter que les performances du maillage hexaédrique-octree sont très

26. https://www.thingiverse.com/thing:1146870

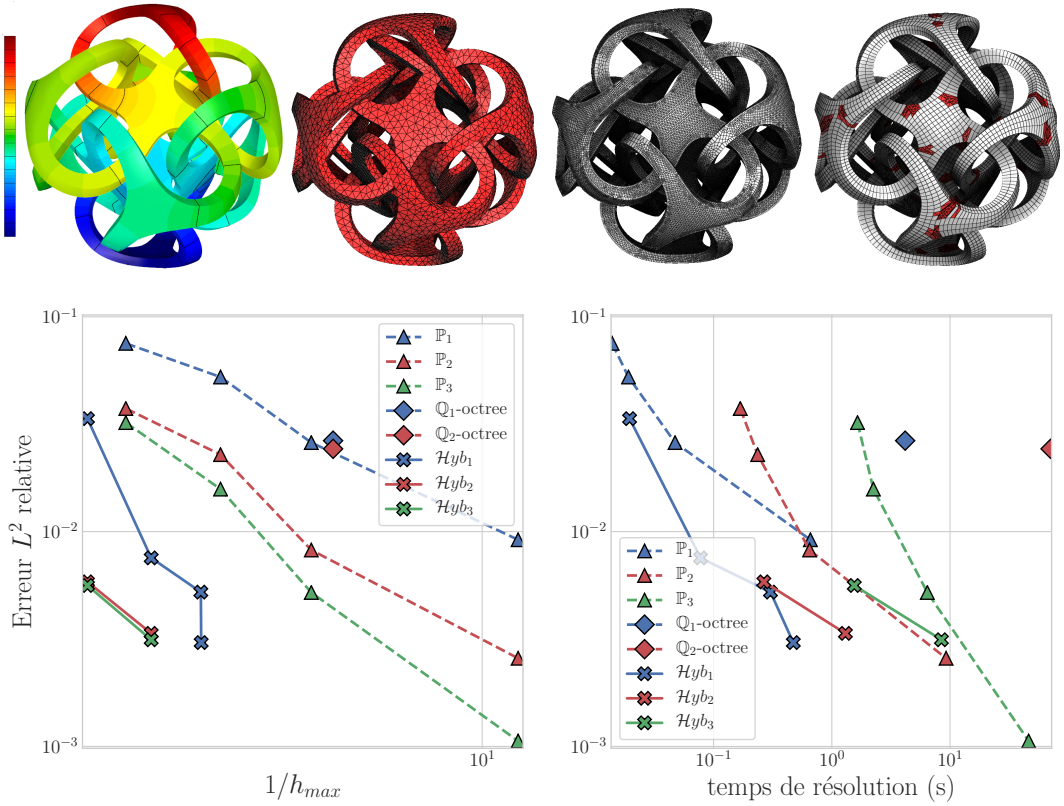


FIGURE 4.11 – Problème de Poisson sur le modèle *metatron* (conditions de Dirichlet nulles sur l'étoile basse et de Neumann non-nulles sur l'étoile haute). *Haut :* Solution et exemples de maillage tétraédrique, hexaédrique (*octree*) et hex-dominant. *Bas-gauche :* Convergence de l'erreur relative (en norme L^2) en fonction de la taille maximale (h_{max}) des arêtes des maillages. *Bas-droit :* Convergence en fonction du temps de calcul du solveur numérique (multigrille algébrique *BoomerAMG*, réduction de 10^{-8} du résidu).

mauvaises, ce qui s'explique par la complexité géométrique du modèle : les hexaèdres (initialement alignés avec la boite englobante) sont très déformés.

Limitations. Pour les ordres deux et trois, nous n'avons que deux maillages hexa-tétraédriques qui ont permis d'effectuer les calculs car il reste des problèmes d'implémentations dans la numérotation des degrés de liberté pour les ordres élevés qui apparaissent dans certaines configurations combinatoires que nous n'avons pas réussi à isoler et corriger. Pour les maillages hexaédriques-octree, nous n'avions qu'un maillage disponible pour la comparaison.

4.3.2 Élasticité linéaire

Modèle du pont

Problème. Nous commençons par un problème d'élasticité constitué d'un modèle très simplifié de pont ou d'arche, illustré sur la Figure 4.12. Une pression uniforme est appliquée sur la partie supérieure et les parties inférieures sont encastrées.

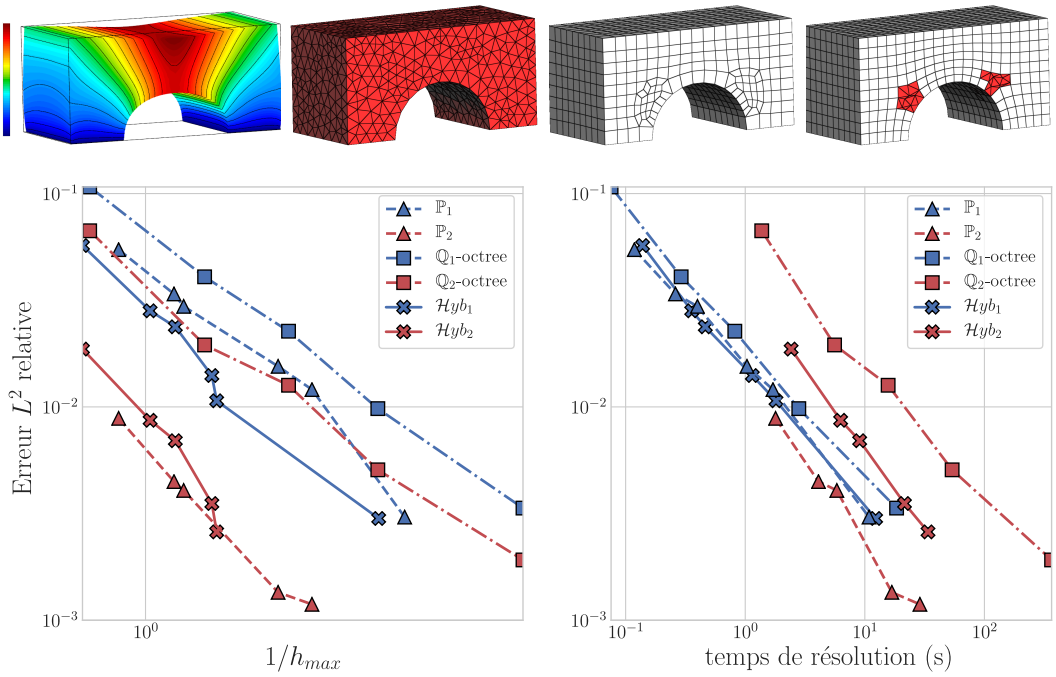


FIGURE 4.12 – Problème d'élasticité linéaire sur le modèle *pont* (fixé au sol et pression uniforme sur le tablier). *Haut :* Solution et exemples de maillage tétraédrique, hexaédrique (*octree*) et hex-dominant. *Bas-gauche :* Convergence de l'erreur relative (en norme L^2) sur le déplacement en fonction de la taille maximale (h_{max}) des arêtes des maillages. *Bas-droit :* Convergence en fonction du temps de calcul du solveur numérique (multigrille algébrique *BoomerAMG*, réduction de 10^{-8} du résidu).

Nous comparons des maillages tétraédriques, hexaédriques-octree et hexa-tétraédriques avec des polynômes d'ordres un et deux. Nous n'avons pas montré les résultats obtenus avec des polynômes d'ordre trois car ceux-ci n'apportent que très peu de précision supplémentaire par rapport à l'ordre deux mais coûtent beaucoup plus cher.

Résultats. Les résultats sont rapportés sur la Figure 4.12. Sur les courbes à gauche (précision en fonction de la résolution), on observe que le passage à l'ordre deux permet une amélioration de la précision, très significative (un ordre de grandeur) pour les tétraèdres et dans une moindre mesure pour les maillages hexaédriques et hex-dominants. Les différences entre maillages sont moins marquées sur les courbes de performance à droite : tous les maillages avec des polynômes d'ordre un ont un ratio équivalent entre précision et temps de calcul. Les tétraèdres quadratiques sont les plus intéressants mais le gain est faible par rapport aux éléments d'ordre un. On peut également remarquer que le passage à l'ordre deux détériore les performances des maillages hexaédriques et hex-dominants, pour lesquels l'ordre un semble le plus intéressant.

Modèle du *hanger*

Problème. Le second problème a pour domaine le modèle *hanger*, dont le maillage

entièrement hexaédrique est emprunté à Livesu et al. [Liv+15]. L'objet est encastré sur le cylindre à gauche et une traction uniforme est appliquée sur le cylindre à droite. La solution en déplacement et les différents types de maillages utilisés sont illustrés sur le haut de la Figure 4.13.

Résultats. Les courbes de la Figure 4.13 montrent l'erreur L^2 en fonction de la résolution et du temps de résolution du système linéaire, pour des solutions éléments finis d'ordres un et deux. Nous avons également effectué les calculs à l'ordre trois mais ne les avons pas affichés car la précision est quasiment identique à celle de l'ordre deux, avec cependant des temps de calculs bien supérieurs. Sur les courbes, nous pouvons observer les points suivants :

— à l'ordre un, les maillages hexaédriques et hex-dominants (hexa-tétraédriques) sont significativement plus performants que les maillages tétraédriques ;

— les maillages hexa-tétraédriques ont un comportement proche de celui des maillages entièrement hexaédriques ;

— les solutions d'ordre deux sont plus intéressantes que celles d'ordre un, à part pour le maillage hexaédrique-octree ;

— à l'ordre deux, les maillages hexaédriques et tétraédriques ont des performances similaires.

Limitations des comparaisons. Les éléments hexaédriques utilisés sont des éléments de Lagrange basiques ($\mathbb{Q}_1, \mathbb{Q}_2$). Or dans le cadre de la mécanique, il est conseillé d'utiliser des éléments améliorés, avec des modes incompatibles ou des variables hybrides (§2.2.1). Malheureusement ceux-ci sont uniquement disponibles dans des logiciels commerciaux et nous n'avons pas eu le temps de les intégrer à ce comparatif (acquisition des licences, conversion des formats de fichier pour calcul de distances, etc).

4.3.3 Conclusions

Nous avons mené quelques comparaisons de différents types maillages volumiques (tétraédriques, hexaédriques, hexaédriques-octree et hexa-tétraédriques) sur des problèmes simples (Poisson et élasticité linéaire) définis sur des domaines relativement compliqués. Nous en déduisons que :

— l'espace d'approximation finis hexa-tétraédrique $\mathcal{H}yb_k$, introduit précédemment (§4.2), permet d'obtenir des performances proches de celles des éléments hexaédriques ($\mathbb{Q}_k$) ;

— les éléments finis hexaédriques standards $\mathbb{Q}_k$ n'apportent pas un gain très significatif par rapport aux éléments tétraédriques $\mathbb{P}_k$ sur des problèmes simples ;

— lorsque les domaines sont compliqués, la vitesse de convergence n'est pas optimale par rapport à l'ordre polynomiale. Nous observons plutôt des vitesses de convergence similaires pour les différents ordres.

Cependant, les problèmes étudiés ne sont pas représentatifs de la complexité des phénomènes physiques pour lesquels les ingénieurs utilisent habituellement des logiciels de simulation numérique. Il serait intéressant d'effectuer des comparaisons

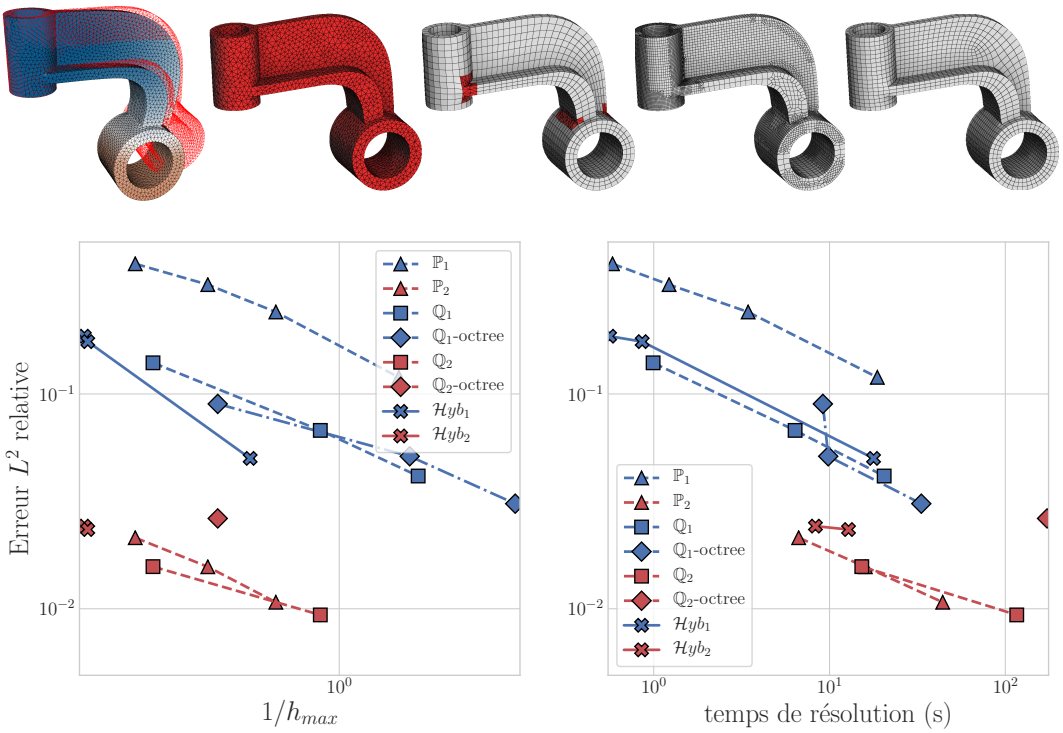


FIGURE 4.13 – Problème d'élasticité linéaire sur le modèle *hanger* (fixé sur le cylindre à gauche et traction uniforme exercée sur le cylindre à droite). *Haut :* Solution et exemples de maillage tétraédrique, hex-dominant, hexaédrique-octree et hexaédrique. *Bas-gauche :* Convergence de l'erreur relative (en norme L^2) sur le déplacement en fonction de la taille maximale (h_{max}) des arêtes des maillages. *Bas-droit :* Convergence en fonction du temps de calcul du solveur numérique (multigrille algébrique *BoomerAMG*, réduction de 10^{-8} du résidu).

similaires sur des problèmes comprenant des non-linéarités ou une dépendance temporelle. De plus, il faudrait utiliser des éléments finis adaptés à chaque problème, comme les hexaèdres avec modes incompatibles ou hybrides pour la mécanique.

Nous souhaitons également souligner que la mise en place du protocole de comparaison ci-dessus n'est pas sans difficultés :

— il faut pouvoir générer des séquences de maillages avec différentes résolutions. C'est assez simple avec les mailleurs tétraédriques actuels (*par ex.* [Mmg17]), mais c'est une tâche plus difficile avec les autres types car il n'y a pas de mailleur hexaédrique automatique (avec alignement aux bords) et les mailleurs hex-dominants ont du mal à capturer précisément les bords du domaine à des résolutions grossières en conservant une proportion significative d'hexaèdres ;

— la solution de référence doit être au moins un ordre de grandeur plus précise que les solutions comparées. Le coût de calcul associé peut être prohibitif ;

— les conditions aux limites (ou autre paramètre du problème) doivent rester plus ou moins constants avec le raffinement des maillages et l'augmentation de l'ordre d'approximation. Dans le cas contrainte, l'erreur mesurée sera en grande partie due à la précision de l'approximation de ces paramètres. Les courbes de

convergence construites avec la méthode des solutions manufacturées sont très affectées par ce phénomène.

Chapitre 5

Calcul de distances entre solutions éléments finis

Dans ce chapitre, nous présentons une méthode efficace qui permet d'évaluer la distance entre deux solutions éléments finis, c'est-à-dire des champs (scalaires ou vectoriels) discrétisés sur des maillages volumiques. Cette distance globale peut notamment être utilisée pour quantifier l'erreur d'approximation d'une méthode numérique en calculant la distance à une solution de référence très précise. Il s'agit d'une des approches évoquées dans l'état de l'art (§2.3, p. 67) pour comparer des solutions éléments finis lorsqu'il n'y a pas de solution analytique. Dans cette thèse, cet outil est utilisé pour étudier l'influence des maillages sur les approximations calculées avec la méthode des éléments finis (§4.3, p. 115).

Concrètement, notre méthode échantillonne les deux champs sur une grille de points régulièrement espacés et calcule une approximation de la distance à partir de ces échantillons. Les solutions éléments finis sont définies par morceaux par des compositions de type

$$f_{|K}(\mathbf{x}) = \hat{f}_K \circ \mathcal{M}_K^{-1}(\mathbf{x}) \tag{5.1}$$

avec $\hat{f}_K$ la fonction d'interpolation de l'élément K du maillage et $\mathcal{M}_K$ sa transformation géométrique. Utiliser directement cette formule pour calculer une distance par échantillonnage (§2.3.3, p. 69) requiert d'effectuer deux opérations coûteuses : trouver l'élément K qui contient $\mathbf{x}$ et inverser la transformation. Notre méthode propose d'échantillonner les champs sans jamais effectuer ces deux opérations. À la place, nous interpolons directement les coordonnées de référence, *c.à-d.* $\hat{\mathbf{x}} = \mathcal{M}_K^{-1}(\mathbf{x})$, en chaque point d'une grille régulière en exploitant des techniques d'informatique graphique (intersection tétraèdre-plan et rastérisation).

D'un certain point de vue, il s'agit d'une approche *brute-force* qui exploite la puissance de calcul des GPUs pour échantillonner massivement les champs à comparer. L'avantage de cette technique est qu'elle est suffisamment précise et très rapide : elle permet de traiter des maillages de tailles importantes (millions d'éléments) en quelques secondes, y compris dans le cas où ceux-ci sont équipés d'éléments finis d'ordres élevés.

Le contenu de ce chapitre a fait l'objet de l'article ci-dessous, publié par le journal *SIAM Journal on Scientific Computing* :

Maxence Reberol *et Bruno* Lévy. *« Computing the distance between*

two finite element solutions defined on different 3D meshes on a GPU ».
In : SIAM Journal on Scientific Computing (accepted) *(2017)*

Organisation du chapitre Échantillonner directement les champs sur une (large) grille régulière 3D n'est pas raisonnable en termes de consommation mémoire, par conséquent nous décomposons le calcul de distance (§5.1) en tranches parallèles (§5.1.1). Pour chaque tranche, les deux champs sont échantillonnés sur une grille 2D (§5.1.2, Figure 5.1). L'environnement de programmation graphique OpenGL permet d'implémenter efficacement le traitement des tranches en exploitant toute la puissance des cartes graphiques (§5.2).

L'approximation de la distance calculée par notre méthode converge bien vers la distance calculée avec des quadratures pour les cas où la solution analytique est connue (§5.3.1) et converge avec le nombre d'échantillons vers une valeur stable dans les autres cas (§5.3.2). Les performances de notre approche sont mesurées sur des cas variés et montrent que l'on obtient des valeurs précises en moins d'une seconde pour des problèmes standards (§5.3.3). Cette approche graphique permet de produire facilement des visualisations intéressantes et interactives des différences entre solutions éléments finis (§5.3.4). L'approche passe bien à l'échelle et converge sur de gros problèmes (plusieurs millions d'éléments avec des polynômes d'ordre 5 et 7) en moins d'une minute (§5.3.3). Toutefois cette méthode comporte des limitations liées à l'échantillonnage régulier ainsi qu'à notre implémentation GPU (§5.3.5). Certaines de ses limitations pourraient être surmontées et d'autres améliorations seraient envisageables (§5.3.6).

L'implémentation (§5.2) de notre méthode est open-source afin d'être facilement utilisable ou modifiable [Reb17].

5.1 Approximation de la distance entre champs

L'objectif de notre méthode est d'approximer la distance L^p entre deux champs f et g définis sur deux maillages distincts. La définition que nous utilisons est la norme L^p de la différence sur le domaine commun aux deux maillages :

$$\begin{aligned} \|f-g\|_{L^p} &= \left(\int_{\mathcal{A}\cap\mathcal{B}} (f(\mathbf{x})-g(\mathbf{x}))^p d\mathbf{x}\right)^{\frac{1}{p}} && \text{si } p<\infty \\ &= \max_{\mathbf{x}\in\mathcal{A}\cap\mathcal{B}} (f(\mathbf{x})-g(\mathbf{x})) && \text{si } p=\infty \end{aligned}$$

Nous proposons d'approcher cette norme par un échantillonnage sur une grille 3D de points $(\mathbf{x}_i)_{i=1..N}$ régulièrement espacés :

$$d_{L^p,h}(f,g) = \Big(\sum_{i=1..N} h^3 (f(\mathbf{x}_i)-g(\mathbf{x}_i))^p\Big)^{1/p}$$

où h est l'écartement entre deux points voisins et N le nombre de points à l'intérieur du domaine commun $\mathcal{A}\cap\mathcal{B}$.

$d_{L^p,h}(f,g)^p$ peut être vu comme la somme de Riemann associée à l'intégrale. Pour des fonctions bornées et continues par morceaux (toujours le cas dans nos applications), cette somme converge vers l'intégrale avec le nombre d'échantillons :

$$d_{L^p,h}(f,g) \xrightarrow[h\to 0]{} \|f-g\|_{L^p}$$

Remarques :

- Les points d'échantillonnage $(\mathbf{x}_i)_{i=1..N}$ sont également les centres de cellules d'une grille cubique régulière. Notre approximation peut alors être interprétée comme la distance entre les projections des deux champs f et g sur un espace d'approximation formé par le maillage cubique équipé de fonctions constantes par cellules.
- Les solutions éléments finis sont souvent obtenues à partir d'équations avec des phénomènes de diffusion importants (ex : équation de la chaleur, élasticité). Par conséquent les solutions sont suffisamment lisses pour pouvoir espérer une convergence rapide de l'approximation.
- L'approximation par échantillonnage régulier est naïve et n'est pas optimale d'un point de vue mathématique mais l'intérêt est qu'elle peut être implémentée de manière très efficace, ce qui nous permet d'utiliser un échantillonnage dense (millions à milliards de points).
- Cette approximation implique que le bord de $\mathcal{A} \cap \mathcal{B}$ est également approximé par des petits cubes (voxels).
- La grille régulière a une orientation (donnée par ses trois axes principaux) qui aura une influence sur l'approximation.

Des réponses quantitatives aux remarques précédentes sont données dans la partie *validation et discussion* (§5.3).

Définition du problème :
Entrées :

- Deux champs scalaires ou vectoriels. Chaque champ (*par ex.* f) est défini par le maillage volumique d'un domaine (*par ex.* $\mathcal{A}$) composé d'éléments $(K_i, i \in [1, n_{\mathcal{A}}])$. Pour chaque élément K, il y une transformation directe $\mathcal{M}_K$ qui définit la géométrie ($K = \mathcal{M}_K(\hat{R})$) et une fonction d'interpolation $\hat{f}_K$ dont le domaine est $\hat{R}$ (la primitive de référence). Le champ est défini par morceaux :

$$f(\mathbf{x}) = f_{|K}(\mathbf{x}) = \hat{f}_K \circ \mathcal{M}_K^{-1}(\mathbf{x}) \text{ si } \mathbf{x} \in K$$

- Une grille d'échantillonnage déterminée par un espacement h entre points voisins et par une orientation (rotation par rapport aux axes de références $\mathbf{x}, \mathbf{y}, \mathbf{z}$)

Sortie :

- La distance approximée : $d_{L^p,h}(f,g) = (\sum_{i=1..N} h^3 (f(\mathbf{x}_i) - g(\mathbf{x}_i))^p)^{1/p}$

Nous nous limitons aux champs définis par morceaux (éléments du maillage) et par des fonctions composées de type *interpolation* $\circ$ *transformation inverse*. Cette restriction n'est pas très contraignante car c'est la définition classique des solutions éléments finis et la plupart des champs définis sur des maillages volumiques peuvent se réécrire sous cette forme si nécessaire. Ceci dit, cette approche exclut par exemple les champs de type *meshless* ou *boundary element method*.

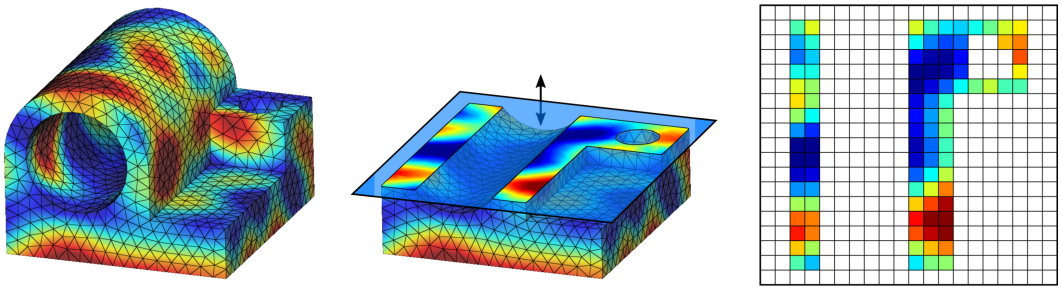


FIGURE 5.1 – *Gauche :* champ scalaire sur un maillage tétraédrique. *Milieu :* une tranche. *Droite :* échantillons du champ aux centres des pixels de la tranche.

5.1.1 Décomposition de la grille d'échantillonnage en tranches

Nous décomposons la grille cubique (ou grille de voxels) en un empilement de grilles de pixels (2D) que nous appelons tranches. L'approximation de la distance s'écrit alors sous la forme d'une double somme :

$$d_{L^p,h}(f,g)^p = h^3 \sum_{k=1..n_s} \sum_{i=1..N_k} (f(\mathbf{x}_i) - g(\mathbf{x}_i))^p$$

où n_s est le nombre de tranches et les $(\mathbf{x}_i)_{i=1..N_k}$ sont les points d'échantillonnage associés à la tranche k.

Les sommes intérieures associées à chaque tranche peuvent être calculées indépendamment. Pour calculer la somme globale, il suffit d'ajouter les contributions de chaque tranche. L'avantage de cette approche est qu'il n'est pas nécessaire de stocker les échantillons de f et g d'une tranche à l'autre. Ceci permet d'utiliser des grilles 3D contenant des milliards d'échantillons tout en conservant une consommation mémoire réduite (les valeurs sur la tranche courante).

Pour une tranche donnée k, nous calculons indépendamment les valeurs de f et g en chaque échantillon, ce qui produit deux listes de valeurs $(f_i = f(\mathbf{x}_i))_{i=1..N_k}, (g_i = g(\mathbf{x}_i))_{i=1..N_k}$. Nous calculons ensuite la somme des différences et stockons cette contribution. Cette étape d'*échantillonnage des champs* est illustrée par la Figure 5.1.

L'algorithme 2 global résume les grandes étapes de notre approche.

Algorithme 2 Calcul de distance

calcul des dimensions de la grille d'échantillonnage n_x, n_y, n_s à partir de h
pour chaque tranche k in $[1, n_s]$ **faire**
  calcul des échantillons $(f_i)_{i=1..N_k}$ ▷ sous-section 5.1.2
  calcul des échantillons $(g_i)_{i=1..N_k}$ ▷ sous-section 5.1.2
  calcul de la contribution $c_k^p = \sum_{i=1..N_k} (f(\mathbf{x}_i) - g(\mathbf{x}_i))^p$
fin pour
calcul de la distance globale $d_h^p(f,g) = (h^3 \sum_{k=1..n_s} c_k^p)^{1/p}$

5.1.2 Échantillonnage du champ sur une grille 2D

Les données d'entrée sont un champ f défini sur un maillage $\mathcal{A}$ et une grille régulière de points, notés $(\mathbf{x}_{ij})_{ij}$. Le champ restreint à un élément est défini par une fonction composée de la forme :

$$f_{|K}(\mathbf{x}) = \hat{f}_K \circ \mathcal{M}_K^{-1}(\mathbf{x}) \tag{5.2}$$

où $\mathcal{M}_K$ est la transformation directe de la primitive de référence vers l'élément $K = \mathcal{M}_K(\hat{K})$. Nous nommons $\hat{\mathbf{x}} = \mathcal{M}_K^{-1}(\mathbf{x})$ les coordonnées dans la primitive de référence. La fonction d'interpolation $\hat{f}_K$ associée à l'élément K a pour domaine $\hat{K}$ (élément de référence). Nous notons $P(\mathbf{x}) = 0$ l'équation du plan $\mathbf{P}$ associé à la tranche courante.

Le résultat de l'algorithme est une liste d'échantillons $(f_{ij})_{ij}$ qui sont les valeurs du champ f aux points $(\mathbf{x}_{ij})_{ij}$. Pour les points qui ne sont pas à l'intérieur de $\mathcal{A}$, nous stockons simplement une valeur *non définie*.

Lorsque $\mathcal{A}$ est un maillage tétraédrique, nous échantillonnons le champ avec l'algorithme 3 ci-dessous.

Algorithme 3 Échantillonnage 2D d'un champ défini sur un maillage tétraédrique

```
pour chaque tétraèdre K tel que K ∩ P ≠ ∅ faire
    interpoler x̂ aux sommets de K ∩ P                ▷ marching tetrahedra
    pour chaque point x_ij à l'intérieur de K ∩ P faire
        interpoler linéairement x̂ pour calculer x̂_ij à x_ij      ▷ rastérisation
        évaluer l'échantillon f_ij = f̂_K(x̂_ij)                   ▷ évaluation du champ
    fin pour
fin pour
```

Nous expliquerons plus tard (§5.1.3) comment étendre cette algorithme aux maillages composés d'éléments à faces courbes comme les hexaèdres trilinéaires.

5.1.2.1 Algorithme d'intersection : *marching tetrahedra*

Pour chaque tétraèdre, il faut calculer l'intersection avec le plan de coupe. Nous utilisons l'algorithme *marching tetrahedra* [AK91] dont nous donnons ci-dessous une brève description.

Historiquement, l'algorithme *marching cube* [LC87] a été développé pour extraire une surface associée à une iso-valeur dans un champ scalaire défini sur un maillage volumique composé de cubes. Cet algorithme a été adapté pour effectuer une extraction similaire dans des maillages tétraédriques [DK91 ; GH95] (ceux-ci pouvant provenir d'une décomposition d'un maillage hexaédrique). Le cas de l'intersection avec un plan est un cas particulier où le champ scalaire correspond à l'équation du plan, de type $ax + by + cz + d = 0$.

Soient un plan défini par $P(\mathbf{x}) = 0$ et un tétraèdre défini par ses quatre sommets $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4$. Si on s'intéresse aux signes des $P(\mathbf{a}_i)$, *c.à-d.* aux valeurs du champ scalaire aux sommets, on peut observer qu'il y a 16 configurations possibles. Deux correspondent à une absence d'intersection (les quatre valeurs de même signe), les quatorze autres peuvent être réduites (par symétrie) à trois configurations où l'intersection est soit un triangle, soit un quadrilatère (*c.à-d.* deux triangles). Les positions

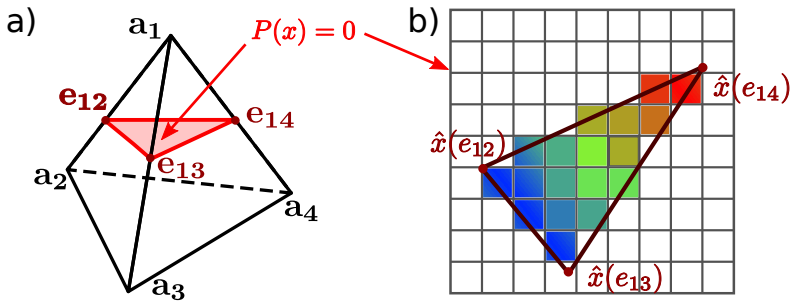


FIGURE 5.2 – *Gauche :* intersection entre un plan et un tétraèdre (*marching tetrahedra*). *Droite :* rastérisation d'un triangle et interpolation des coordonnées de référence $\hat{\mathbf{x}}$ aux centres des pixels (symbolisés par les couleurs des pixels).

des sommets de l'intersection sont calculées par interpolation linéaire. Si l'intersection contient un sommet $\mathbf{e}_{ij}$ sur l'arête $i-j$, sa position est donnée par $\mathbf{e}_{ij} = (1-\alpha)\mathbf{a}_i + \alpha\mathbf{a}_j$ avec $\alpha = \frac{P(\mathbf{a}_i)}{P(\mathbf{a}_i)-P(\mathbf{a}_j)}$. Si un champ (scalaire ou vectoriel) est attaché aux sommets, on peut calculer ses valeurs aux nouveaux sommets $\mathbf{e}_{ij}$ en utilisant la même interpolation linéaire.

La gauche de la Figure 5.2 illustre cet algorithme d'intersection dans notre cas, où l'on interpole les coordonnées de références $\hat{\mathbf{x}}$. Par exemple pour l'arête $1-2$:

$$\mathbf{e}_{12} = (1-\alpha)\mathbf{a}_1 + \alpha\mathbf{a}_2$$
$$\hat{\mathbf{x}}_{12} = (1-\alpha)\hat{\mathbf{x}}_1 + \alpha\hat{\mathbf{x}}_2$$

avec $\alpha = \frac{P(\mathbf{a}_1)}{P(\mathbf{a}_1)-P(\mathbf{a}_2)}$ et $\hat{\mathbf{x}}_1 = (0,0,0)$, $\hat{\mathbf{x}}_2 = (0,1,0)$ les coordonnées dans l'espace de référence associées aux sommets $\mathbf{a}_1, \mathbf{a}_2$ du tétraèdre.

Optimisation : restriction à un ensemble de candidats Il est couteux d'appeler l'algorithme *marching tetrahedra* pour chaque tétraèdre du maillage (pour une tranche donnée) alors que la plupart des tétraèdres ne sont pas intersectés par le plan. Nous proposons une optimisation qui consiste à trier spatialement les éléments et à maintenir une liste de tétraèdres candidats.

Ayant en tête une implémentation GPU, nous souhaitons que les candidats forment un bloc *consécutif* dans la liste globale des tétraèdres du maillage, de sorte que la liste des candidats soit décrite par deux indices c_{bas}, c_{haut}.

Une proposition simple est de trier les tétraèdres par leur position minimale le long de l'axe perpendiculaire aux tranches. En supposant que les tranches sont perpendiculaires à l'axe z, on trie les tétraèdres selon leurs coordonnées z minimale, notée z_{min}. Alors lorsque que la position z_k du plan de coupe augmente à z_{k+1}, l'intervalle $[c^k_{bas}, c^k_{dernier}]$ est mis à jour en $[c^{k+1}_{bas}, c^{k+1}_{dernier}]$ tel que :

— de tous les tétraèdres entièrement en dessous du plan de coupe z_{k+1}, c^{k+1}_{bas} est celui avec le plus grand z_{min}

— de tous les tétraèdres entièrement au-dessus du plan de coupe z_{k+1}, c^{k+1}_{haut} est celui avec le plus petit z_{min}

Cet ensemble contient des tétraèdres qui ne sont pas intersectés par le plan de coupe mais nous avons constaté en pratique que cette optimisation est suffisante pour la

plupart des maillages et permet un gain de vitesse très significatif (facteur 10 à 100 voire 1000 suivant les maillages car elle permet d'éviter un grand nombre d'accès mémoires sur le GPU).

5.1.2.2 Rastérisation des triangles

En informatique graphique, rastériser un triangle consiste à déterminer les pixels à l'intérieur et à interpoler linéairement des propriétés aux centres de ceux-ci. Dans notre cas, l'*entrée* de la rastérisation est un triangle produit par l'algorithme d'intersection plan-tétraèdre présenté ci-dessus. Il faut déterminer quels centres de pixels (points d'échantillonnage) sont à l'intérieur du triangle et interpoler linéairement les coordonnées de références (définies sur les trois sommets du triangle) aux points. Ce processus est illustré sur la droite de la Figure 5.2.

Dans notre implémentation GPU (§5.2), cette étape est réalisée automatiquement et très efficacement par OpenGL. En effet, comme la rastérisation est une étape décisive du pipeline de rendu graphique sur GPU, elle est implémentée au niveau matériel et est très optimisée.

5.1.2.3 Échantillonnage : évaluation du champ

À chaque point d'échantillonnage $\mathbf{x}_{ij}$ à l'intérieur du triangle rasterisé (de l'élément K), nous évaluons le champ pour obtenir f_{ij} avec la formule d'interpolation des éléments finis :

$$f_{ij} = \sum_{k=1..N_K} f_{K,k}\ \hat{\phi}_k(\hat{\mathbf{x}}_{ij}) \tag{5.3}$$

Pour les éléments de Lagrange, les $(f_{K,k})_{k=1..N_K}$ sont les coefficients associés aux N_K degrés de liberté de l'élément K, $(\hat{\phi}_k)_{k=1..N_K}$ sont les fonctions de formes de Lagrange associées à l'élément de référence $\hat{K}$ et $\hat{\mathbf{x}}_{ij}$ sont les coordonnées de référence calculées lors de la rastérisation.

Cette étape d'évaluation peut être généralisée à toute situation où il est possible de calculer l'échantillon à partir de coefficients et de coordonnées de référence :

$$f_{ij} = \hat{f}_K(\hat{\mathbf{x}}_{ij})$$

Ceci inclut les fonctions d'interpolation d'éléments finis vectoriels comme ceux de Nédélec ou de Raviart-Thomas.

5.1.3 Extension aux éléments courbes

Jusqu'à maintenant nous avons considéré des tétraèdres dont les transformations sont affines, ce qui garantit que les étapes *marching tetrahedra* et *rastérisation* sont exactes. Dans cette section, nous étendons notre approche aux éléments dont les transformations ne sont pas affines, par exemple : hexaèdres (transformations trilinéaires), tétraèdres quadratiques ou pyramides (transformations rationnelles).

Dans la formule d'évaluation du champ (5.3), il faut déterminer l'élément K qui contient le point et obtenir les coordonnées de référence $\hat{\mathbf{x}}$ pour calculer l'échantillon $f(\mathbf{x}_i)$. Lorsque la transformation $\mathcal{M}$ est affine, ces deux étapes sont simples (intersection tétraèdre-plan et interpolation linéaire), mais la situation est plus compliquée avec des transformations non linéaires. Nous évoquons deux approches possibles :

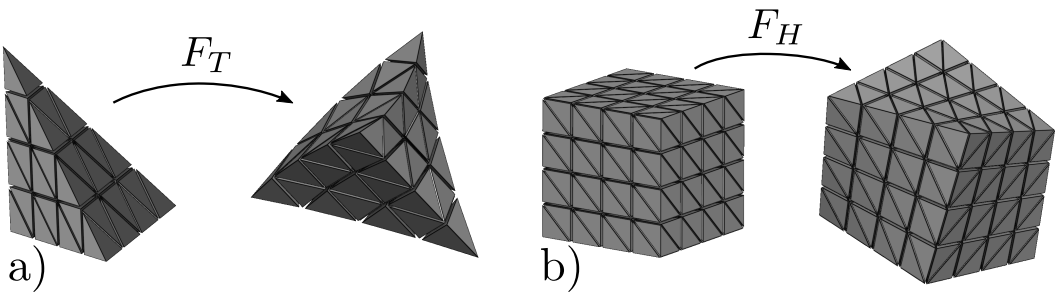


FIGURE 5.3 – Approximation linéaire par morceaux des transformations (décomposition en tétraèdres). (a) Tétraèdre quadratique avec deux niveaux de subdivision. (b) Hexaèdre trilinéaire avec deux niveaux de subdivision.

— approche exacte : (a) Construire une boite englobante de l'élément et tester s'il y a intersection entre cette boite et le plan. (b) Si oui, inverser la transformation avec une méthode itérative pour calculer les coordonnées de référence $\hat{\mathbf{x}} = \mathcal{M}^{-1}(\mathbf{x})$. Si $\hat{\mathbf{x}}$ est dans la primitive de référence, alors cela veut dire que le point $\mathbf{x}$ est bien dans K et le champ peut être évalué avec (5.3).

— approche par subdivision : Tous les éléments du maillage dont les transformations sont non-affines sont subdivisés en dizaines de petits tétraèdres. L'algorithme précédent, développé pour les transformations affines, est alors appliqué à tous les sous-tétraèdres.

La seconde approche n'est pas exacte : elle est équivalente à approximer les transformations par des fonctions affines par morceaux. Mais l'avantage est qu'elle ne nécessite ni la construction de boites englobantes ni l'inversion des transformations. Nous avons choisi la seconde approche dans cette thèse, sachant que dans notre implémentation nous pouvons réaliser la subdivision des éléments efficacement sur le GPU.

Dans le nouvel algorithme 4, la primitive de référence $\hat{K}$ est décomposée en sous tétraèdres qui sont transformés par $\mathcal{M}_K$ (illustré sur la Figure 5.3) :

Algorithme 4 Échantillonnage du champ avec éléments courbes

pour chaque élément K **faire**
  subdiviser K en tétraèdres T's ▷ *subdivision*
  pour chaque sous-tétraèdre T tel que $T \cap \mathbf{P} \neq \emptyset$ **faire**
    interpoler $\hat{\mathbf{x}}$ aux sommets de $T \cap \mathbf{P}$ ▷ *marching tetrahedra*
    pour chaque point $\mathbf{x}_{ij}$ à l'intérieur de $T \cap \mathbf{P}$ **faire**
      interpoler linéairement $\hat{\mathbf{x}}$ pour calculer $\hat{\mathbf{x}}_{ij}$ à $\mathbf{x}_{ij}$ ▷ *rastérisation*
      calculer $f_{ij} = \hat{f}_K(\hat{\mathbf{x}_{ij}})$ ▷ *évaluation du champ*
    fin pour
  fin pour
fin pour

Cet algorithme est quasiment identique au précédent si ce n'est qu'il y a une subdivision supplémentaire. Plus formellement, la nouvelle formule d'évaluation est :

$f_{ij} = \hat{f}_K(\tilde{\mathcal{M}}_K{}^{-1}(\mathbf{x}_{ij}))$, où $\tilde{\mathcal{M}}_K$ est l'approximation linéaire par morceaux de la transformation $\mathcal{M}_K$.

Subdivision des éléments Pour toute primitive de référence (principalement tétraèdre, cube et pyramide), n'importe quelle subdivision composée de sous-tétraèdres de formes et tailles similaires conviendra. Dans notre implémentation, nous subdivisons récursivement les tétraèdres et les hexaèdres. Le nombre, ou niveau, de subdivision *sl* est le paramètre qui contrôle la précision de l'approximation.

Pour les *hexaèdres*, notre subdivision de base consiste à découper un cube ou huit cubes. Quand tous les niveaux de subdivisions ont été appliqués, les cubes sont finalement décomposés en six tétraèdres. Par exemple, un niveau deux de subdivision produit 384 sous-tétraèdres, comme illustré sur la Figure 5.3. À la fin, chaque hexaèdre est décomposé en $8^{sl} \times 6$ sous-tétraèdres. Cette décomposition introduit une direction diagonale sur les faces. Deux hexaèdres adjacents devraient posséder la même direction diagonale sur leurs faces communes pour que la subdivision (globale) soit cohérente. Dans le cas contraire, le maillage subdivisé n'est pas parfait et il peut comporter des petits trous ou chevauchements. Il est possible de trouver des subdivisions cohérentes en utilisant des décompositions en 5 tets et en 6 tets et en étudiant les configurations locales, cependant cela exigerait un prétraitement et l'implémentation GPU des subdivisions serait plus compliquée. En pratique, on observe que ce défaut de cohérence dans les subdivisions a un impact négligeable sur les calculs et nous l'ignorons.

Pour les *tétraèdres*, notre subdivision de base est d'appliquer une décomposition en quatre tétraèdres et un octaèdre, puis de décomposer l'octaèdre en quatre nouveaux tétraèdres. À chaque niveau de subdivision, chaque tétraèdre devient ainsi huit sous-tétraèdres. À la fin, la subdivision produit 8^{sl} sous-tétraèdres, soit par exemple 64 sous tétraèdres pour le niveau deux comme illustré sur la Figure 5.3.

Dans les maillages hybrides composés d'éléments de différents types, il est préférable d'utiliser des subdivisions compatibles, c'est-à-dire qu'à chaque sous-tétraèdre sur le bord d'une décomposition correspond un sous-tétraèdre dans la décomposition de l'élément adjacent. Les incompatibilités de diagonales peuvent être ignorées si le niveau de subdivision est suffisamment élevé, par contre il faut s'assurer que les subdivisions produisent des tailles cohérentes telles que les nouveaux sommets des décompositions de deux éléments adjacents se correspondent.

L'approximation géométrique des éléments courbes introduit une erreur sur le calcul de la distance finale. Cette erreur décroit avec le niveau de subdivision, et le niveau de subdivision adéquat dépend de l'application : une subdivision de niveau un peut être suffisante pour une première estimation de la distance tandis qu'il est nécessaire d'utiliser des niveaux supérieurs pour des calculs précis (e.g. distance à une solution de référence lors d'une analyse de convergence éléments finis). Ce point est discuté plus en détail dans la section avec des expériences numériques (§5.3).

5.2 Implémentation GPU avec OpenGL

Les cartes graphiques (GPU) sont extrêmement efficaces pour exécuter des algorithmes parallélisés, où tous les processus exécutent les mêmes instructions en même temps. De plus les cartes graphiques ont des circuits spécialisés pour rastériser des

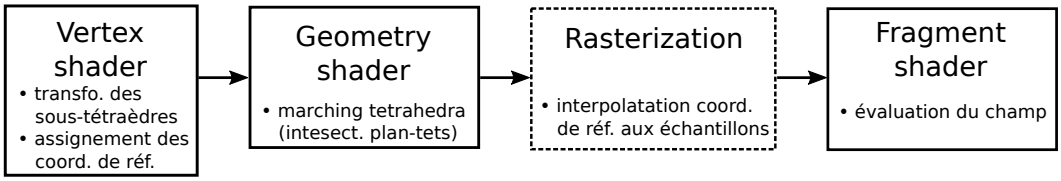


FIGURE 5.4 – Pipeline de rendu OpenGL utilisé pour échantillonner les champs.

triangles et effectuer des interpolations linéaires dans ceux-ci. L'objectif de notre implémentation est d'exploiter au maximum ces spécificités.

Notre implémentation est calquée sur les algorithmes 2 et 4. La version adaptée à l'environnement OpenGL est résumée par l'algorithme 5. Les différentes étapes sont discutées dans la suite de cette partie.

Algorithme 5 Aperçu du fonctionnement de l'implémentation OpenGL

transfert des coefficients des maillages et champs de CPU vers GPU
pour chaque tranche k de $[1, n_s]$ **faire**
  rendu du champ f dans la texture T_f ▷ *pipeline de rendu*
  rendu du champ g dans la texture T_g ▷ *pipeline de rendu*
  calcul de la texture de différence $T_d = T_f - T_g$ ▷ *différence de textures*
  réduction de la texture T_d vers la k-ième colonne de la texture T_N ▷ *réduction*
fin pour
transfert de la texture T_N du GPU vers le CPU
calcul de la distance globale à partir des valeurs de T_N

5.2.1 Pipeline de rendu

Le *pipeline de rendu* est l'étape qui permet d'échantillonner les champs sur les grilles de pixels. Dans la suite, nous utiliserons les termes *échantillonnage* et *rendu* de manière interchangeable. Les sous-étapes du pipeline de rendu sont résumées sur la Figure 5.4.

Au lieu d'un rendu classique à l'écran, nous utilisons OpenGL de manière à effectuer des rendus dans des textures, *c.à-d.* des grilles rectangulaires, en les attachant à des *framebuffer object*. La taille des textures est fixée à $n_x * n_y$ (e.g. 1000x800). En pratique, ces dimensions sont calculées à partir du nombre d'échantillons maximal souhaité. Également, au lieu de produire des couleurs RGBA, nous configurons OpenGL de manière à produire des nombres flottants (32 bits), ce qui se traduit par l'utilisation de textures de type GL_FLOAT.

Le *pipeline de rendu* est appelé pour chaque tranche via un appel à la fonction glDrawElementsInstanced() [27] avec la primitive GL_LINES_ADJACENCY en argument afin de spécifier que les points sont regroupés quatre par quatre en entrée du *geometry shader*.

En appelant glDrawElementsInstanced(), nous dessinons plusieurs instances d'un groupe d'éléments. Dans notre cas, les instances correspondent aux éléments du

27. OpenGL 4.5 Reference Pages https://www.opengl.org/sdk/docs/man4/

maillage à trancher et le groupe d'éléments est toujours un groupe de tétraèdres (sommets regroupés par quatre) et correspond à la décomposition de la primitive de référence (e.g. cube, pyramide). Pour un maillage tétraédrique avec géométries affines, ce groupe est réduit à un seul tétraèdre.

Dans la terminologie graphique (OpenGL, DirectX, etc), les opérations effectuées dans les étapes successives du pipeline de rendu sont spécifiées dans des *shaders*. Chaque shader effectue un nombre limité de tâches. Dans notre implémentation, la répartition des tâches est la suivante :

a) Le *vertex shader* a pour entrée des tétraèdres (les décompositions). Il applique la transformation directe (élément de référence vers espace réel) associée à l'instance en cours. Des coordonnées dans l'élément de référence (e.g. $(0, 0, 0)$ pour le premier sommet) sont associées aux sommets (elles seront utilisées dans le *geometry shader*). Les coefficients déterminant la transformation sont lus via mécanisme OpenGL de *vertex attribute* et changent pour chaque instance.

b) Le *geometry shader* est appelé une fois pour chaque tétraèdre et exécute l'algorithme de *marching tetraedre* (intersection plan tétraèdre). Il produit 0, 1 ou 2 triangles avec les coordonnées de référence interpolées à leurs sommets.

c) L'étape de rastérisation est exécutée automatiquement par OpenGL. Elle interpole linéairement les coordonnées de référence aux centres des pixels de la grille qui sont à l'intérieur du triangle considéré.

d) Le *fragment shader* est responsable du rendu final. Il est appelé pour chaque pixel émis par la rastérisation. Dans notre approche, il reçoit les coordonnées de référence du pixel et il évalue le champ en appliquant la formule d'interpolation. Cette formule est un polynôme dont les coefficients sont récupérés via un mécanisme détaillé dans le paragraphe **Évaluation des champs**.

Si le champ est composé de plusieurs types d'éléments finis (primitive géométrique différente ou fonction d'interpolation différente), nous appelons le pipeline de rendu une fois pour chaque type. Ainsi lors d'un appel de rendu, toutes les instances utilisent la même formule de transformation et d'interpolation (dont les coefficients varient).

Évaluation des champs Il serait possible de stocker les coordonnées de référence obtenues au niveau du *fragment shader* dans une texture, de la télécharger sur le CPU et d'effectuer les interpolations sur le CPU. Cependant les transferts de données entre le CPU et le GPU sont lents, et transférer deux textures de tailles importantes à chaque tranche prend beaucoup plus de temps que le rendu initial. Il est donc nécessaire d'évaluer complètement les champs sur le GPU : transférer (une seule fois à l'initialisation) les coefficients associés aux fonctions d'interpolation du CPU vers le GPU et y accéder dans le *fragment shader*. Prenons l'exemple d'un maillage composé d'un million d'éléments, avec dix degrés de liberté (*par ex.* tétraèdres $\mathbb{P}_2$), et un champ vectoriel de dimension trois (*par ex.* déplacement en élasticité). Cela correspond à 30 millions de valeurs (32 bits). Une telle quantité ne tient pas directement ni dans une texture, ni dans un *uniform buffer object*, ni dans un *buffer texture*. Une possibilité serait de diviser les données en de nombreuses textures et d'accéder à la texture voulue au moment adéquat, mais cela compliquerait l'implémentation. À la place nous avons choisi d'utiliser les *Shader Storage Buffer Objects (SSBO)* qui ont été introduits avec OpenGL 4.3. car leur taille maximale est celle de la mémoire GPU.

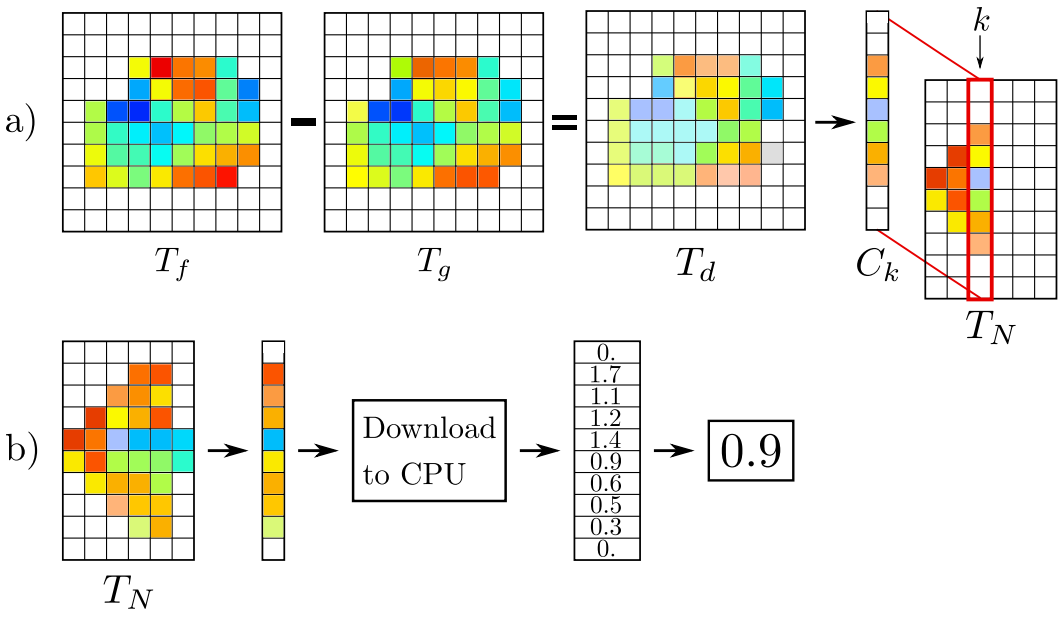


FIGURE 5.5 – Calcul de distance à partir des échantillons. (a) Traitement de la k-ème tranche. (b) Réduction finale des contributions de chaque tranche pour calculer la distance globale.

Malheureusement cela implique que l'ordinateur utilisé supporte OpenGL 4.3, ce qui n'est pas le cas des ordinateurs sous OSX à l'heure actuelle.

La *sortie* du pipeline de rendu est une texture (*c.à-d.* un tableau 2D) avec les valeurs du champ (ou valeurs *non-défini* au niveau des pixels qui ne sont pas à l'intérieur du domaine). Si le champ est scalaire, le format de la texture est GL_RED (une valeur par pixel). Si le champ est vectoriel, nous utilisons respectivement les formats GL_RG, GL_RGB, GL_RGBA pour les dimensions deux, trois et quatre. Pour des dimensions supérieures, il faut utiliser un rendu sur plusieurs textures (*multiple render targets*).

5.2.2 Différence de textures

Comme les deux textures T_f, T_g associées aux champs f et g sont disponibles sur la mémoire GPU, nous pouvons facilement calculer la texture différence $T_d = T_f - T_g$ en dessinant un quadrilatère et en appelant un nouveau *fragment shader* sur chaque pixel. Ce *fragment shader* exécute simplement une soustraction (composante par composante si le champ est vectoriel).

5.2.3 Réduction de texture

Les temps de transfert entre CPU et GPU étant lents, ils peuvent dégrader significativement les performances de notre approche. Il faut donc traiter la texture de différence T_d directement sur le GPU.

Les contributions d'une tranche à la distance globale sont stockées dans une texture notée T_N. T_N est une texture de taille $n_y * n_s$ dont chaque colonne C_k est

une contribution calculée à partir de la k-ème texture de différence (T_d). La j-ème valeur du vecteur colonne C_k est la somme des valeurs calculée le long de la j-ème ligne de T_d. Ce processus est illustré sur la Figure 5.5.a. Pour chaque tranche, nous calculons un vecteur contribution au lieu d'une seule valeur car la réduction en une seule valeur n'est pas optimale du point de vue de l'utilisation du GPU.

Quand le champ est vectoriel, nous combinons les différentes composantes en prenant la racine des carrés moyennés. Nous appliquons également la puissance p correspondante à la norme L^p souhaitée, ou nous prenons le maximum des différences pour la norme L^∞. Pour calculer la distance globale, il faut également connaître le nombre de pixels à l'intérieur du domaine pour chaque tranche. Ce nombre est stocké comme une composante de la texture T_N qui est multivaluée (GL_RGBA).

Finalement, la texture T_N est transférée sur le CPU afin de calculer la distance globale (Figure 5.5.b.). Il est possible de réduire T_N en une texture 1D directement sur le GPU (avant le transfert) mais le gain (sur le temps de calcul total) est faible car l'opération est réalisée une seule fois.

Pour des raisons d'efficacité, nous calculons plusieurs distances en même temps lors de la réduction qui remplit la texture T_N : celle-ci peut stocker quatre valeurs par pixels (dont une est prise par le nombre d'échantillons dans le domaine). Notre implémentation calcule les distances L^1, L^2, L^∞ mais ceci peut être facilement modifié.

Remarque. Il est entièrement possible d'utiliser d'autres stratégies de réduction pour calculer les contributions. Mais il faut garder en tête que les textures ne sont ni carrés ni des puissances de deux, ce qui rend le problème non trivial.

De plus, il serait possible d'effectuer directement la différence des textures vers une colonne sans former explicitement la texture T_d. Nous l'avons conservée car elle facilite fortement le débogage et permet une visualisation interactive.

5.3 Validation et discussion de la méthode

Dans cette partie, nous validons la méthode d'approximation de distance sur deux problèmes analytiques simples (§5.3.1) et nous explorons l'influence des paramètres d'approximation (nombre d'échantillons et niveau de subdivision) sur des cas d'applications réalistes (§5.3.2). Dans la partie suivante (§5.3.3) nous nous intéressons aux performances (convergence, temps de calcul). En plus de la rapidité, un autre avantage de notre approche graphique est la possibilité de visualiser les résultats intermédiaires et de se servir interactivement de l'outil (§5.3.4). Dans la dernière partie, nous évoquons les limitations inhérentes à notre approche graphique et implémentation GPU (§5.3.5), ainsi que des perspectives d'amélioration (§5.3.6).

5.3.1 Validation sur des cas analytiques de *bosses sinusoïdales*

Lorsque la solution analytique d'un problème est connue, il est possible de calculer l'erreur associée à la solution éléments finis en transformant les points de quadrature dans l'espace réel et en évaluant à la fois la solution analytique et la solution calculée (voir méthode des solutions manufacturées §2.3.1, p. 67). Cette procédure est rapide car tous les calculs sont directs, *c.à-d.* sans transformation à inverser. Dans les validations suivantes, nous utilisons cette erreur (calculée à partir de quadratures

d'ordre élevé) comme référence et nous la comparons à la distance (à une solution de référence) calculée avec notre approche.

Nous commençons par étudier le problème de la *bosse sinusoïdale* (*sine bump* en anglais). On rappelle qu'il s'agit d'un problème de Poisson simple de la forme $-\Delta u = f$ avec $f = 3\pi^2 \sin(\pi x)\sin(\pi y)\sin(\pi z)$, avec des conditions de Dirichlet nulles sur les bords du domaine qui est le cube unité $[0,1]^3$. La solution analytique est $u = \sin(\pi x)\sin(\pi y)\sin(\pi z)$. Nous résolvons ce problème avec des éléments finis linéaires par morceaux $\mathbb{P}_1$ définis sur des tétraèdres et des éléments trilinéaires $\mathbb{Q}_1$ définis sur des hexaèdres. Les solutions numériques sont illustrées sur la gauche de la Figure 5.6. Pour évaluer l'influence des transformations non-affines, nous avons construit les maillages de sorte que les faces des hexaèdres ne soient pas planes (quatre sommets non-alignés). Les erreurs de référence (en pointillés sur la figure) sont calculées en utilisant des quadratures d'ordres élevés. Il est intéressant de souligner que pour obtenir une erreur L^∞ précise, nous avons du utiliser des quadratures associées à des polynômes d'ordre supérieur à 20 (milliers de points par élément) car la solution analytique n'est pas polynomiale.

Pour le maillage tétraédrique, les erreurs reportées sur le haut de la Figure 5.6 montrent que notre approximation de la distance converge rapidement vers la bonne valeur avec le nombre d'échantillons. Comme l'on pouvait s'y attendre, la convergence de la distance L^∞ est plus lente car elle est uniquement déterminée par le point où les deux champs diffèrent le plus, or un échantillonnage uniforme n'est pas une stratégie optimale ni efficace pour trouver ce point. Un autre aspect à considérer est que les champs (produits de sinusoïdes) sont très lisses, et donc le nombre d'échantillons nécessaires pour ce cas simple n'est probablement pas pertinent pour des problèmes plus compliqués.

Pour le maillage hexaédrique, les faces intérieures ne sont pas planes car les transformations définissant la géométrie des éléments sont des fonctions trilinéaires. Dans notre approche, nous approximons ces géométries en les décomposant en sous-tétraèdres. Sur le bas de la Figure 5.6, on peut observer l'influence du niveau de subdivision sur notre calcul de la distance L^∞. Nous n'indiquons que la norme L^∞ car il s'agit du cas le plus défavorable, mais le comportement est similaire dans une moindre mesure pour les autres normes. Sur cet exemple, nous constatons que la décomposition d'un hexaèdre en six tétraèdres n'est clairement pas suffisante car elle produit une distance significativement fausse. Par contre le niveau deux de subdivision (384 sous-tétraèdres) permet d'approcher suffisamment la distance exacte. Un seul niveau de subdivision (48 sous-tétraèdres) peut suffire pour une approximation plus grossière de la distance.

Le second problème analytique que nous utilisons pour valider notre approche est un problème d'élasticité linéaire que nous résolvons avec le formulation faible standard en déplacement. Il est très similaire au problème précèdent : le domaine est le cube unité, les conditions aux limites sont celles de Dirichlet homogène (déplacement nul sur les bords) et la solution analytique est un produit de fonctions sinusoïdales dont les composantes sont :

$$\forall i \in [0,2],\ u_i = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$$

Ce problème est résolu avec des maillages tétraèdres et hexaédriques de différentes résolutions, équipés de fonctions éléments finis de Lagrange d'ordre un à trois. Nous

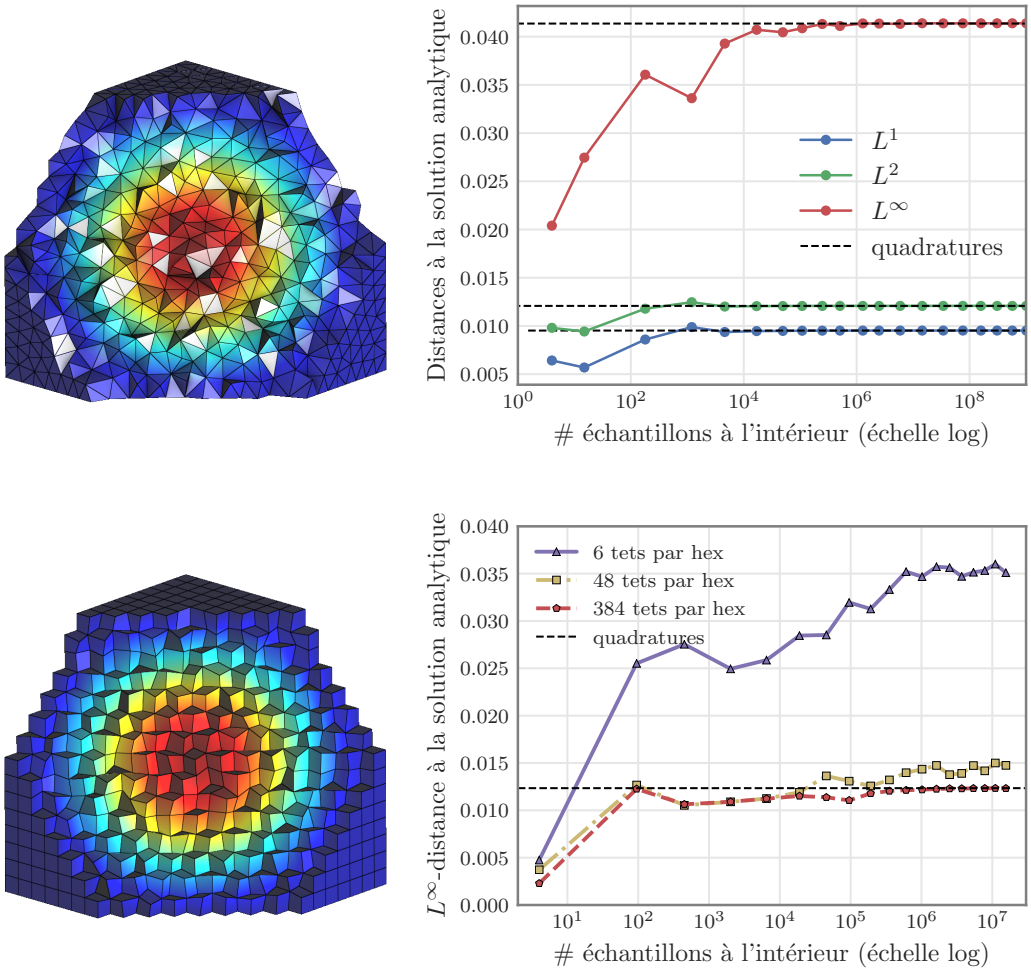


FIGURE 5.6 – Distances à la solution analytique. *Gauche :* Solutions $\mathbb{P}_1$ et $\mathbb{Q}_1$ définis sur des maillages tétraédriques et hexaédriques du cube unité. *Haut-droit :* Convergence de la distance approchée avec le nombre d'échantillons dans le domaine, pour la solution $\mathbb{P}_1$. *Bas-droit :* Influence du niveau de subdivision appliqué aux hexaèdres sur la distance L^∞. Les distances de référence (en pointillés) sont calculées avec des quadratures d'ordres élevés.

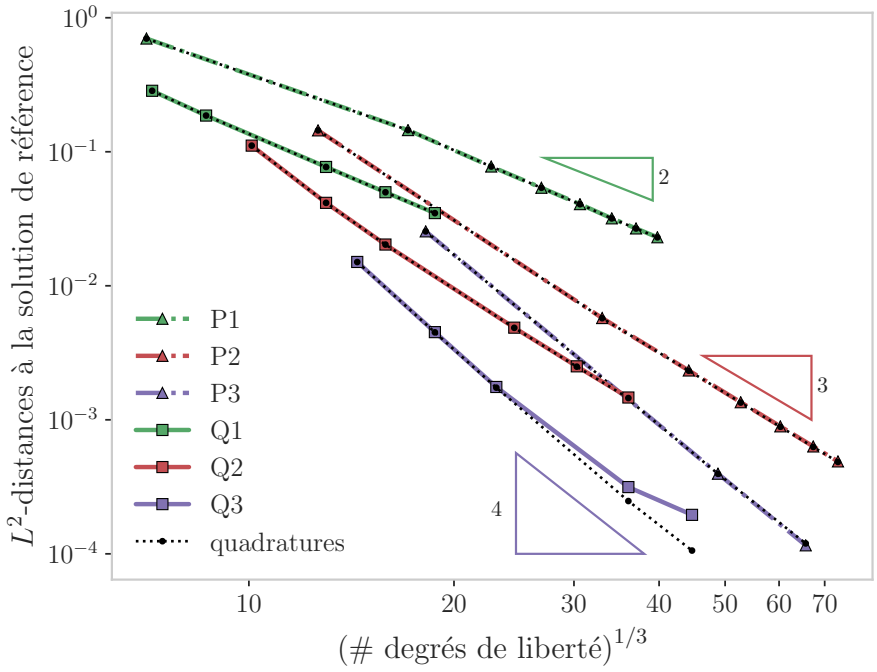


FIGURE 5.7 – Problème d'élasticité linéaire *sine bump*. Convergence en norme L^2 des solutions éléments finis de Lagrange vers la solution de référence en fonction du nombre de degrés de liberté. La solution de référence est $P4$ calculée sur un maillage fin. La distance relative L^2 est la distance divisée par la norme L^2 de la solution. Les courbes en pointillés correspondent à l'erreur d'approximation L^2, calculée par quadrature avec la solution analytique. La perte de convergence de la courbe $Q3$ est due à une décomposition insuffisante des transformations trilinéaires dans le calcul de distance.

calculons également une solution de référence d'ordre quatre sur un maillage très fin. Pour chaque espace d'approximation (maillage et élément fini associé), nous calculons la distance L^2 à la solution de référence avec notre approche ainsi que l'erreur d'approximation par rapport à la solution analytique avec des quadratures d'ordres élevées (e.g. 20). Pour le calcul de distance, nous utilisons 500^3 échantillons ainsi qu'un niveau trois de subdivision pour les transformations des hexaèdres. Les résultats sont présentés sur la Figure 5.7 sous forme de courbes de convergence éléments finis. Comme le domaine est cubique et que les éléments sont relativement isotropes, la racine cubique du nombre de degrés de liberté est approximativement proportionnelle à la taille caractéristique h des éléments. Pour l'élasticité linéaire, l'erreur optimale L^2 associée à la solution éléments finis d'ordre k est de type $\|u - u_h\|_{L^2} < Ch^{k+1}$ pour une solution $\mathcal{C}^\infty$, ce qui est le cas ici.

On observe que, dans la plupart des cas, la distance à la solution de référence calculée avec notre approche correspond à l'erreur d'approximation calculée avec la solution analytique par quadratures. Cependant, la distance calculée n'est plus exacte pour les maillages hexaédriques fins équipés d'éléments d'ordre trois ($\mathbb{Q}_3$). Cet écart est dû à l'approximation linéaire par morceaux des transformations trilinéaires des hexaèdres dans le calcul de distance : lorsque l'approximation par éléments finis

devient très précise (erreur relative inférieure à 10^{-3}), l'erreur d'approximation des transformations trilinéaires devient dominante. En pratique, cet effet peut être contré en augmentant le niveau de subdivision des hexaèdres afin de repousser le problème à un niveau de précision supérieur.

À partir de ces deux cas analytiques, nous concluons que notre calcul de distance approché se comporte bien puisque : (1) il converge vers les valeurs attendues lorsque le nombre d'échantillons augmente, (2) les éléments courbes peuvent être approximés de manière satisfaisante (avec une décomposition suffisamment fine). Concernant le second point, il faut bien être conscient que les mesures de distance de précision élevée nécessitent des décompositions importantes. Si la décomposition n'est pas adaptée, la distance calculée pourra être significativement fausse.

5.3.2 Sensibilité aux paramètres pour des maillages et champs non-triviaux

En pratique, l'utilisateur doit choisir un nombre d'échantillons (ou de manière équivalente un espacement entre échantillons voisins), et un niveau de subdivision si les éléments sont non-affines. Nous étudions maintenant deux exemples afin de savoir comment choisir ces paramètres et quelle est leur influence sur le calcul de distance.

Le premier cas d'étude est un problème d'élasticité linéaire défini sur la pièce mécanique dénotée *hanger*. Le modèle 3D, les résultats de la simulation par éléments finis, ainsi que des exemples de maillages utilisés sont illustrés sur la Figure 5.8. Pour le moment, nous nous concentrons sur la convergence du calcul de distance en fonction des paramètres (nombre d'échantillons et niveau de subdivision). Sur la Figure 5.9.a., nous étudions la distance entre une solution $\mathbb{P}_3$ calculée sur un maillage grossier (86k tets) et une autre solution $\mathbb{P}_2$ calculée sur un maillage plus fin (960k tets). On observe que l'approximation de la distance a convergé en moins de dix millions d'échantillons. Ce nombre plutôt faible d'échantillons nécessaires peut s'expliquer par l'aspect lisse du champ de déplacement.

Sur la Figure 5.9.b., on calcule la distance entre une solution $\mathbb{P}_2$ définie sur un maillage tétraédrique (209k éléments) et une solution $\mathbb{Q}_2$ définie sur un maillage hexaédrique (36k hexs), pour lequel nous utilisons différents niveaux de subdivision lors du calcul de la distance L^∞. On observe que le niveau un est suffisant, ce qui correspond à approximer la géométrie des hexaèdres par 48 sous-tétraèdres. Il est important de remarquer qu'une décomposition en seulement six tétraèdres produit de mauvais résultats.

Les exemples précédents considèrent uniquement des champs lisses (problème de Poisson ou d'élasticité linéaire), or il parait assez logique que notre approche par échantillonnage régulier fonctionne bien sur ce type de cas. Dans les exemples suivants, nous regardons le comportement de notre calcul sur des champs plus irréguliers. Pour générer des champs aléatoires tout en contrôlant la corrélation spatiale, nous utilisons l'algorithme du bruit de Perlin [Per85].

Les résultats expérimentaux de la figure Figure 5.10 montrent que la convergence de notre calcul de distance est plus lente pour des champs avec des variations hautes fréquences, particulièrement pour la norme L^∞. Ce comportement est logique, mais il est intéressant de noter que la convergence est toujours suffisamment rapide pour des applications concrètes, même avec des hautes fréquences. Ceci peut s'expliquer

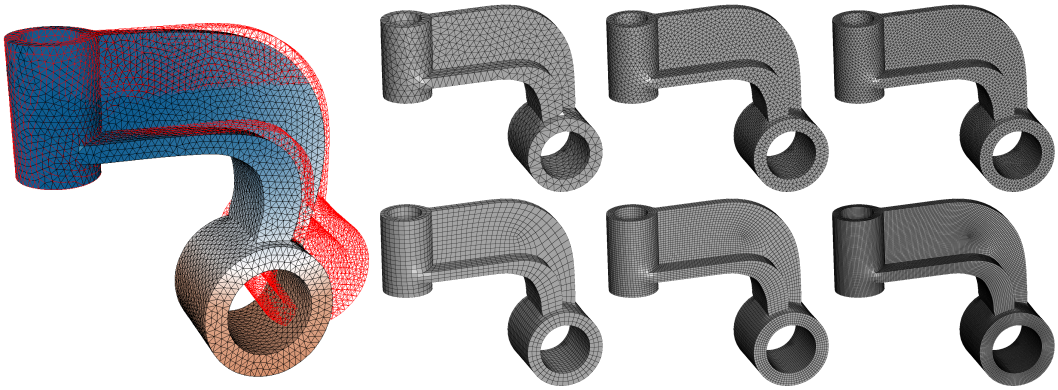


FIGURE 5.8 – Problème d'élasticité linéaire sur le modèle *hanger*. *Gauche :* Solution éléments finis, maillage non déformé en rouge, amplitude du déplacement en couleurs. *Droite :* Exemples de maillages utilisés.

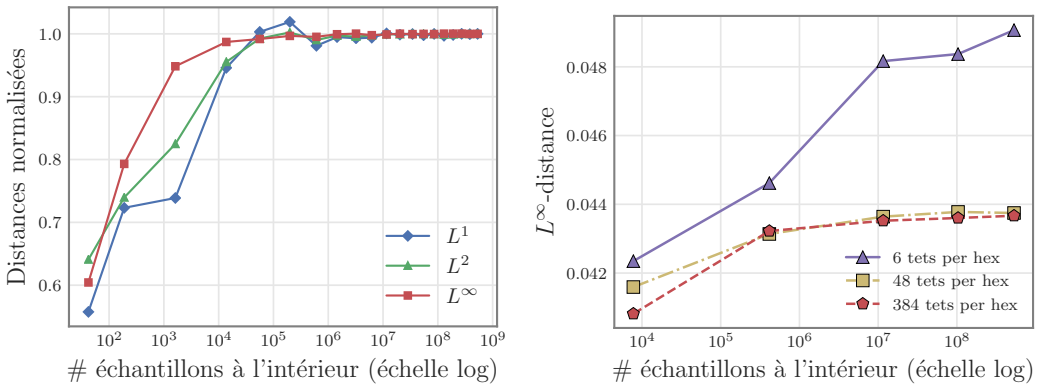


FIGURE 5.9 – *hanger*. Problème d'élasticité linéaire, déplacements illustrés sur la Figure 5.8. *Gauche :* convergence des distances L^1, L^2, L^∞. Les distances sont normalisées par leur valeur la plus précise afin de pouvoir afficher les trois normes sur la même échelle. *Droite :* influence du niveau de subdivision sur la distance L^∞ calculée sur le maillage hexaédrique du *hanger*.

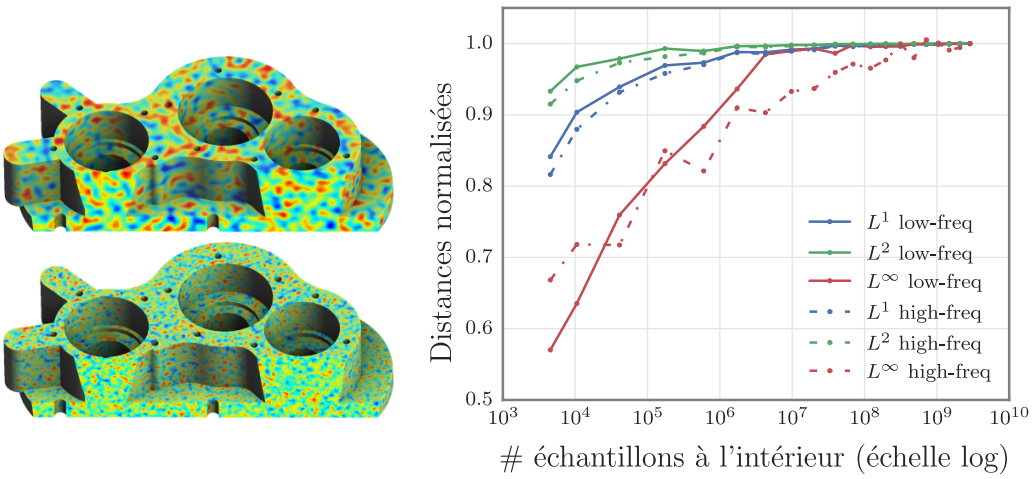


FIGURE 5.10 – *Gauche :* modèle *carter* coupé, avec des bruits de Perlin respectivement basse et haute fréquence. *Droite :* pour les deux champs, les distances calculées entre les deux solutions éléments finis (P_1 sur le maillage fin et P_2 sur le maillage grossier)

par le fait que les distances calculées sont globales (à l'exception de la norme L^∞), et que la différence *moyenne* des deux champs est capturée assez rapidement. Lors de nos essais et expériences numériques, nous avons observé qu'il est très rare d'avoir besoin de plus d'un milliard d'échantillons à l'intérieur du domaine pour converger. Il est sans aucun doute possible de mettre en défaut notre approche en construisant des cas pathologiques, mais ceux-ci ne sont pas des résultats typiques de simulations éléments finis. En pratique, pour s'assurer que l'échantillonnage est suffisant, nous calculons la distance plusieurs fois en augmentant le nombre d'échantillons jusqu'à ce que la valeur calculée soit stable.

Comme notre approche utilise une grille régulière pour évaluer la distance, elle est sensible à l'orientation de la boite englobante dans laquelle est définie la grille. Nous étudions l'influence de cette orientation sur la distance calculée pour les simulations précédentes (bosse sinusoïdale et élasticité linéaire sur le *hanger*), en affichant les variations sur la distance calculée pour des rotations d'axes aléatoires. Sur la Figure 5.11, on observe que l'influence de la rotation diminue avec le nombre d'échantillons, comme l'on pouvait s'y attendre. Pour des champs suffisamment échantillonnés, on considère que l'influence de la rotation est négligeable.

5.3.3 Performances

Comme notre approche exploite des opérations optimisées pour les GPUs (*par ex.* rastérisation de triangles), le calcul de distance prend généralement quelques secondes avec une carte graphique de moyenne gamme (e.g. Nvidia 960) et moins d'une seconde avec une carte graphique haut de gamme (e.g. Nvidia 1080GTX). Dans cette partie, nous reportons les temps de calcul obtenus sur un ensemble de maillages et de champs variés, dont ceux présentés précédemment. Le temps de calcul dépend de nombreux facteurs : nombre d'échantillons dans le domaine, nombre d'éléments

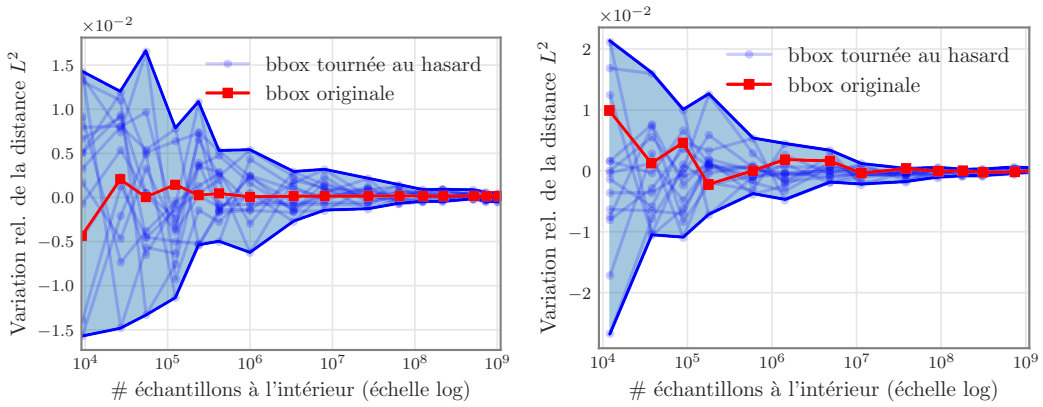


FIGURE 5.11 – Influence de l'orientation des axes en fonction du nombre d'échantillons dans le domaine. *Gauche :* Bosse sinusoïdale sur le cube. *Droite :* Problème d'élasticité linéaire sur le modèle *hanger*.

dans le maillage, niveau de subdivision si éléments non-affines, dimension du champ (scalaire ou vectoriel). Le temps nécessaire pour obtenir une bonne approximation de la distance variera donc significativement pour chaque modèle, mais l'important est que pour des maillages standards (jusqu'à quelques millions d'éléments), nous obtenons toujours une approximation précise en moins de quelques secondes.

Les temps de calcul rapportés sur la Figure 5.13 sont obtenus avec une *Nvidia Geforce GTX 1080*. Chaque paire de champs est décrit dans le tableau à gauche de la Figure 5.13. Pour les maillages hexaédriques, nous utilisons toujours le niveau deux de subdivision (chaque hexaèdre est décomposé en 384 tets), ce qui est un choix sûr d'après nos discussions précédentes. Les temps de calculs ne prennent pas en compte les temps de transfert du disque dur vers la mémoire vive (car ils dépendent de la présence ou non de SSD, du cache, etc) mais ils incluent les temps de transfert de la mémoire vive vers la mémoire GPU. Comme nous sommes principalement intéressés par le temps minimal requis pour obtenir une bonne approximation de la distance, nous affichons la convergence des distances L^2, L^∞ en fonction des temps de calcul sur la droite de la Figure 5.13.

En général, il faut moins d'une seconde pour calculer précisément une distance en utilisant jusqu'à un milliard d'échantillons à l'intérieur du domaine, pour des maillages tétraédriques de taille moyenne (entre 100k et 1M d'éléments). Le calcul sur les maillages hexaédriques est plus lent, à cause des subdivisions en 384 tétraèdres, mais cela reste du même ordre de grandeur. La convergence de la distance est souvent atteinte en moins d'une seconde pour la distance L^2, y compris pour des maillages assez imposants (cf les résultats sur la Figure 5.13). Pour les gros maillages, comme #5 et #6 illustrés sur la Figure 5.12, il faut plus d'échantillons pour converger en norme L^∞, mais il s'agit toujours d'un calcul prenant au plus quelques secondes.

Ces faibles temps de calcul permettent de calculer des dizaines de distances en quelques minutes, ce qui est très utile pour effectuer des analyses de convergence, comme celles présentées dans la dernière partie du *Chapitre 4*.

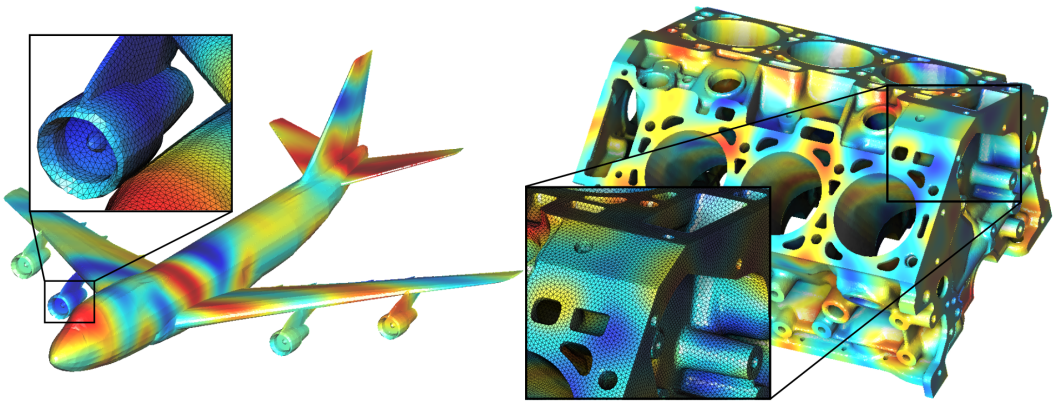


FIGURE 5.12 – *Gauche :* Bruit de Perlin sur modèle *747*. *Droite :* Bruit de Perlin sur modèle *40 heads*. Les maillages tétraédriques sont mis en avant sur les zooms.

Passage à l'échelle. Dans l'exemple qui suit, nous calculons la distance entre deux champs définis sur deux maillages tétraédriques du modèle *40 heads*, illustré sur la Figure 5.12. Ce modèle comporte de nombreux détails géométriques et est composé de beaucoup de vide (uniquement 20% de la boite englobante est solide). Le champ est un bruit de Perlin [Per85] projeté sur deux espaces éléments finis : le premier est composé de 2 905k tétraèdres munis de polynômes de degré 5 (56 degrés de liberté) et le second est composé de 2 305k tétraèdres munis de polynômes de degré 7 (120 degrés de liberté). Ces espaces comportent respectivement 64M et 146M degrés de liberté (notre implémentation fonctionne comme si les champs étaient discontinus aux interfaces entre éléments). Cet exemple implique l'utilisation de larges grilles d'échantillonnage (jusqu'à 3752x3067x2200) et consomme $3GB$ de mémoire GPU.

Les convergences des distances L^1, L^2 et L^∞ en fonction du nombre d'échantillons à l'intérieur du domaine et en fonction du temps de calcul sont indiquées sur la Figure 5.14. On observe que la convergence est plus lente qu'avec les polynômes d'ordres plus faibles étudiés précédemment, ce qui est logique car il faut beaucoup plus d'échantillons pour approximer les intégrales. L'évaluation des polynômes dans le *fragment shader* est également plus lente (beaucoup plus d'opérations */+ que nous n'avons pas cherché à optimiser). Toutefois les temps de calcul nécessaires restent tout à fait raisonnables (de l'ordre d'une minute) compte tenu de la taille des champs.

Pour des gros maillages (dizaines de millions d'éléments), notre approche peut être adaptée en ajoutant une étape de décomposition des maillages. Dans ce dernier cas, le temps calcul pourra prendre quelques minutes pour obtenir des valeurs précises, mais ce sera toujours des ordres de grandeur plus rapide que le temps requis pour effectuer les simulations éléments finis en premier lieu.

5.3.4 Visualisation interactive et extraction d'information

L'objectif initial de notre méthode est de calculer des distances globales, mais il est intéressant d'exploiter le fait que notre *pipeline de rendu* calcule la différence entre les deux champs à chaque échantillon.

Dans notre implémentation, la mémoire GPU contient les valeurs de f, g et $f - g$

model	dim	# cells	basis
#1 : joint (fig. 5.1)	1	10 k	P_2 (tet)
		65 k	P_1 (tet)
#2 : hanger (fig. 5.8)	3	1 739 k	P_1 (tet)
		343 k	P_2 (tet)
#3 : hanger (fig. 5.8)	3	36 k	Q_2 (hex)
		290 k	Q_1 (hex)
#4 : carter (fig. 5.10)	1	1 210 k	P_1 (tet)
		373 k	P_2 (tet)
#5 : 747 (fig. 5.12)	1	1 385 k	P_1 (tet)
		577 k	P_2 (tet)
#6 : 40heads (fig. 5.12)	1	2 905 k	P_1 (tet)
		2 350 k	P_2 (tet)

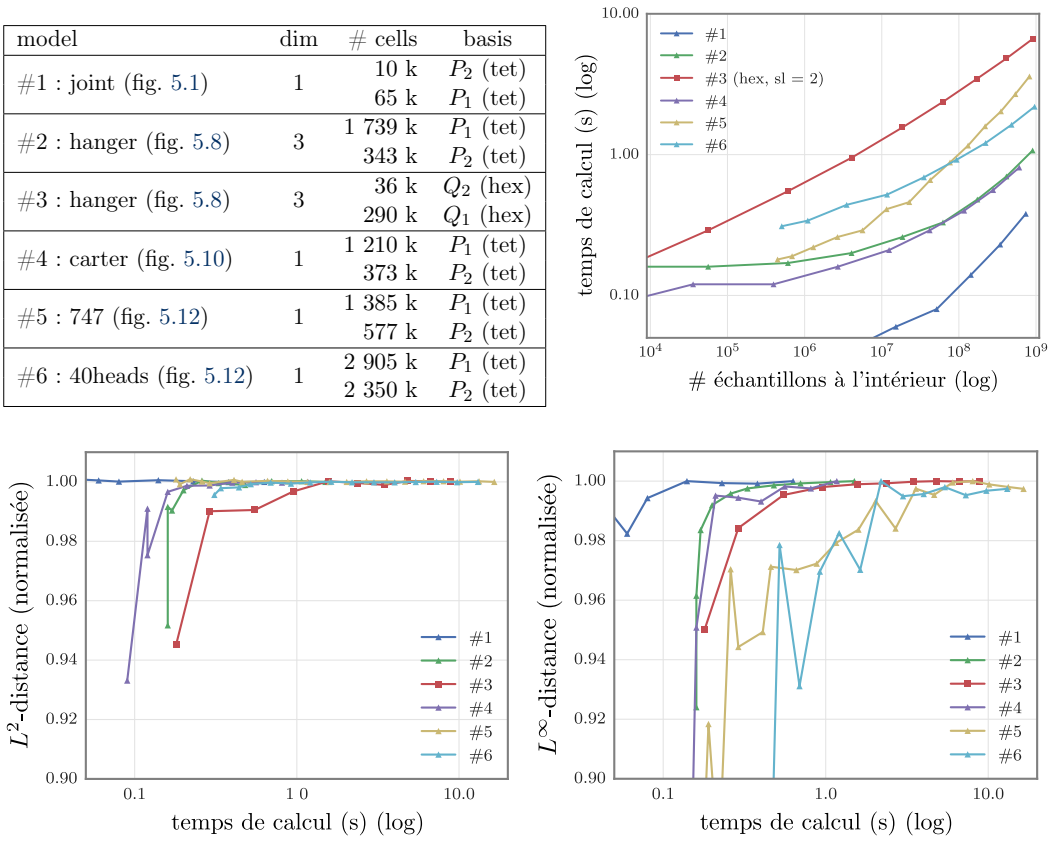


FIGURE 5.13 – *Haut-gauche :* Description des pairs de champs. *Haut-droit :* Évolution des temps de calcul avec le nombre d'échantillons. *Bas :* Distances L^2, L^∞ calculées en fonction du temps de calcul (qui est lié au nombre d'échantillons par les courbes en haut à droite). Pour visualiser les différents cas sur les mêmes figures, la distance L^2 est normalisée par la valeur calculée avec le plus grand nombre d'échantillons et la valeur L^∞ est normalisée par la plus grande valeur obtenue pour chaque modèle.

dans des textures pour chaque tranche. Il est très simple de visualiser graphiquement ces textures, comme sur la Figure 5.15. Cette visualisation est particulièrement intéressante pour comprendre comment l'ordre des polynômes, la géométrie des éléments, ou toute autre propriété, affectent la solution éléments finis. Les textures peuvent également être exportées comme images, ou encore il est possible d'exporter la voxelisation complète pour une analyse plus approfondie dans un autre logiciel avec des fonctionnalités avancées de visualisation (e.g. iso-valeurs, transparence) comme ParaView.

5.3.5 Limitations

Notre calcul de distance souffre de limitations inhérentes à l'utilisation d'un échantillonnage régulier et à son implémentation sur GPU.

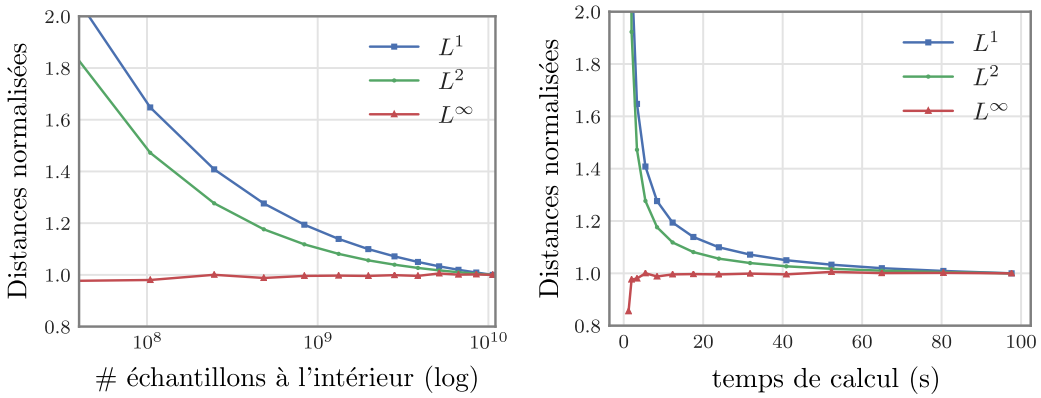


FIGURE 5.14 – Performances sur de gros maillages tétraédriques (2 905k et 2 350k tets) équipés de polynômes d'ordres élevés (5 et 7). Convergence de la distance en fonction du nombre d'échantillons dans le domaine (gauche) et du temps de calcul (droite). Le modèle *40 heads* utilisé est illustré sur la Figure 5.12.

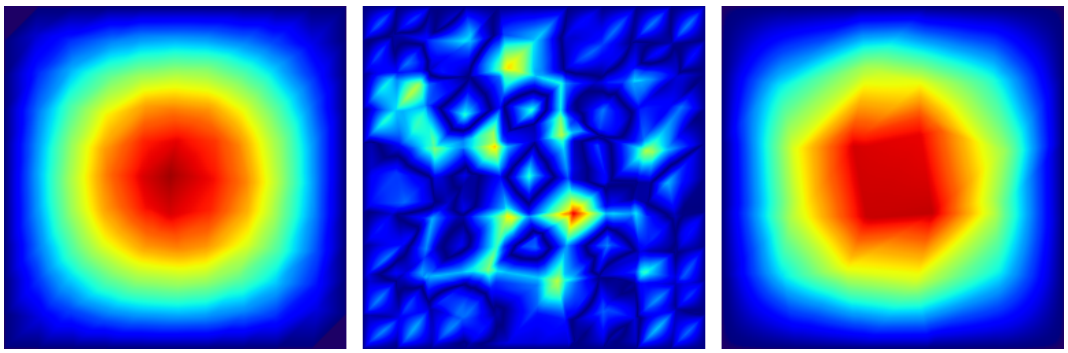


FIGURE 5.15 – *Gauche :* tranche de la solution $\mathbb{Q}_1$ du problème *sinus bump*. *Droite :* tranche de la solution $\mathbb{P}_1$. *Milieu :* différence absolue des deux échantillonnages.

Échantillonnage régulier. La distance calculée est déterminée par les valeurs du champ aux points d'échantillonnage régulièrement espacés. Il est facile d'imaginer des cas où cet échantillonnage rate complètement les valeurs intéressantes des champs étudiés. Ce problème ne devrait cependant pas arriver pour des fonctions suffisamment lisses, comme c'est le cas des solutions éléments finis composées de polynômes. Cependant, les potentiels utilisateurs doivent avoir à l'esprit que cette approche ne capture pas efficacement les phénomènes hautes fréquences.

L'échantillonnage régulier est affecté par l'orientation de la boite englobante (axes de la grille). Ceci a notamment un impact sur l'approximation par voxels des bords du maillage. De la même façon, lors de l'étude de champs discontinus aux interfaces entre éléments (e.g. Galerkine discontinu), l'échantillonnage des interfaces sera affecté par l'orientation de la grille. En pratique ces défauts sont assez négligeables dès que l'échantillonnage est suffisamment fin (plusieurs millions d'échantillons).

Approximation par subdivision des éléments courbes. Les transformations non-affines sont approximées par des transformations affines par morceaux (*c.à-d.* par une décomposition en tétraèdres). L'erreur due à cette approximation dépend du niveau de subdivision. L'exemple de la Figure 5.7 montre que cette erreur peut devenir dominante par rapport à la distance calculée entre solutions éléments finis proches. Une solution à ce problème est d'augmenter le niveau de subdivision, cependant un niveau supérieur à trois peut ralentir significativement le temps de calcul.

Limitations liées à l'implémentation sur GPU. Notre approche est rapide parce qu'elle repose sur des algorithmes de visualisation pour lesquels les cartes graphiques sont optimisées. Il faut donc posséder une carte graphique suffisamment puissante. En pratique, ceci exclut les cartes graphiques intégrées aux processeurs ou cartes mères. Une seconde limitation est que notre implémentation exploite des fonctionnalités d'OpenGL 4.3, version qui n'est pas forcément disponible pour tous les systèmes d'exploitation (actuellement non supporté par OSX).

Notre implémentation utilise des nombres flottants d'une précision de 32 bits parce qu'OpenGL ne permet pas d'effectuer la rastérisation avec une précision double. Cependant il est tout à fait possible de re-implémenter notre approche avec OpenCL ou CUDA pour profiter à la fois de la puissance des GPUs et d'une précision 64 bits, au prix de temps de calcul rallongés. D'après notre expérience avec ce code, la précision limitée à 32-bits n'a jamais posé de problèmes mais ce pourrait être le cas sur de très gros maillages.

Un autre aspect limitant des cartes graphiques est la quantité de mémoire disponible. Dans notre implémentation, les deux maillages ainsi que les coefficients associés aux champs sont transférés (une seule fois à l'initialisation) sur GPU. Dans notre cas (Nvidia 1080 GTX), la limite mémoire est 8 GB, ce qui est suffisant pour traiter des maillages importants (dizaines à centaines de millions d'éléments). Si les maillages et champs sont trop gros, il faut les décomposer.

Pour une tranche donnée, nous calculons tous les échantillons. Pour des gros maillages, cela peut impliquer d'utiliser et de stocker des grilles de pixels assez importantes (e.g. $5k \times 5k$), ce qui n'est pas forcément adapté pour les GPUs qui sont optimisés pour des grilles dont la taille est de l'ordre de la résolution des écrans. Une possibilité est de décomposer les tranches en sous-tranches (équivalent à effectuer des zooms) et de traiter ces sous-tranches indépendamment. Cette décomposition pourrait être particulièrement utile pour gérer des modèles imposants composés de nombreux et/ou gros vides (e.g. micro-structures).

5.3.6 Perspectives d'amélioration

Les limitations liées à l'échantillonnage régulier sont difficilement évitables car c'est principalement cette régularité qui nous a permis de développer une approche efficace au problème de calcul de distances. Ceci dit, il est fortement possible d'améliorer notre implémentation afin qu'elle soit plus précise et plus à même de traiter des problèmes plus imposants. Un autre axe d'amélioration important concerne l'expérience utilisateur : comment cet outil peut-il être adapté pour répondre au mieux aux besoins des utilisateurs ?

Approximation des éléments courbes. Pour des calculs de distance haute précision, notre calcul de distance peut souffrir d'erreurs trop importantes introduites par l'approximation des transformations non-affines. Une amélioration pratique qui nous parait prometteuse serait d'utiliser la décomposition comme une première approximation des coordonnées de référence, puis d'appliquer quelques itérations d'un algorithme de type Newton-Raphson sur la transformation non-affine afin de converger vers des coordonnées de référence beaucoup plus précises. Comme un niveau de décomposition élevé (e.g. niveau trois) est très couteux en temps de calcul, il devrait être possible d'implémenter une solution rapide combinant un niveau un de subdivision comme initialisation et quelques itérations. L'inconvénient de cette amélioration est une augmentation de la complexité du code ainsi qu'une perte de flexibilité : il est nécessaire d'utiliser le gradient de la transformation non-affine, qui est spécifique à chaque type de transformation.

Optimisation et passage à l'échelle. Une première possibilité pour s'affranchir des problèmes de limitation mémoire et précision (passer de 32 à 64 bits) est d'implémenter notre approche sur CPU. Nous pensons que cette solution est viable mais il faut être conscient que pour obtenir des performances décentes, un gros travail d'optimisation sera nécessaire (parallélisation, instructions AVX, etc). En termes d'efforts, cette implémentation CPU sera probablement du même ordre qu'une implémentation GPU.

Concernant l'implémentation GPU, nous envisageons deux améliorations techniques majeures, toutes deux réalisables par un prétraitement CPU des maillages :

— Décomposer les maillages (et champs) en sous-parties et les traiter indépendamment. Ceci nécessite que les décompositions des deux maillages soient compatibles, c'est-à-dire qu'elles couvrent une même sous-grille d'échantillonnage. Cette amélioration permettrait de consommer moins de mémoire GPU et donc de pouvoir traiter des plus gros problèmes.

— Pour chaque tranche, décomposer la grille d'échantillonnage en sous-grilles (*par ex.* la grille k devient 16 sous-grilles). Pour chaque sous-grille, déterminer à l'avance si elle intersecte des tétraèdres, et si oui lesquels. L'objectif de cette amélioration est d'éviter le plus de calculs d'intersections inutiles par le GPU et de permettre une couverture en pixels plus importante par tranche (mais en plusieurs fois au lieu d'une).

Fonctionnalités supplémentaires. Dans un premier temps, il est possible de modifier légèrement l'implémentation actuelle pour prendre en compte des normes différentes : par exemple pour la norme H^1, il faudrait évaluer les gradients des champs aux points d'échantillonnage et les stocker dans les textures 2D afin que ceux-ci soient utilisés par le calcul de contribution lors de l'étape de réduction. De la même manière, il est relativement simple d'ajouter tout calcul de quantité qui peut être approché à partir d'échantillons sur une grille régulière. Il peut également être intéressant de calculer des informations plus spécifiques : extraire les échantillons où la distance est supérieure à un seuil donné, calculer la distance par élément et extraire ceux qui diffèrent le plus, etc. Toutes ces informations peuvent être intégrées dans un système de visualisation afin de permettre à l'utilisateur d'explorer interactivement la

différence entre les champs, en lui permettant également d'agrandir certaines zones, etc.

En somme, l'échantillonnage très rapide des champs permet d'envisager de nombreuses applications d'analyses qui dépassent le cadre restreint du calcul de distances globales.

5.3.7 Conclusion

Nous avons présenté une méthode rapide de calcul de distances entre champs discrétisés sur des maillages volumiques (e.g. solutions éléments finis), basée sur un échantillonnage régulier. Cette méthode est une réponse à un problème rencontré au cours de cette thèse : comment comparer quantitativement les solutions éléments finis obtenus sur des maillages hex-dominants à celles obtenues sur des maillages tétraédriques ? Les comparaisons réalisées à partir de cet outil sont présentées dans la dernière partie du *Chapitre 4*.

L'approche que nous proposons repose sur deux paramètres : le nombre d'échantillons et le niveau de subdivision des éléments courbes. Tout utilisateur de cette approche doit vérifier que les valeurs choisies sont appropriées pour l'application considérée. Par exemple, un niveau de subdivision insuffisant peut déformer les vitesses de convergence observées lors d'une analyse de convergence éléments finis. Toutefois, comme le calcul de la distance est rapide, il est facile de lancer le même calcul avec des paramètres supérieurs et de vérifier que les distances calculées sont identiques.

L'approche développée étant plus générale que notre problème initial, et étant flexible car reposant sur des algorithmes simples, nous espérons qu'elle pourra servir à d'autres applications. Afin de faciliter sa réutilisation, son implémentation est accessible librement [Reb17].

Conclusion

Dans cette thèse nous avons étudié différentes problématiques associées à la génération (*Chapitre 3*), à l'utilisation (*Chapitre 4*), à l'évaluation (*Chapitres 4 et 5*) des maillages hexaédriques et hex-dominants :

- développement d'un algorithme robuste d'extraction de cavités pour le maillage hex-dominant (§3.1, [Ray+17]) ;
- exploration de techniques pratiques pour produire des singularités linéiques dans les champs de directions échantillonnés sur les sommets afin d'augmenter la proportion d'hexaèdres dans les maillages hex-dominants (§3.2) ;
- comparaisons de solutions éléments finis simples calculées sur des maillages triangulaires et quadrangulaires avec différentes orientations des éléments (§4.1) ;
- développement d'un espace approximation conforme ($\mathcal{H}yb_k \subset H^1$) d'ordre k pour les maillages hexa-tétraédriques à partir de polynômes $\mathbb{P}_k$, $\mathbb{P}_{2k}$, $\mathbb{Q}_k$ et de contraintes de continuité (§4.2, [RL16]) ;
- développement d'une méthode efficace de calcul de distances entre solutions éléments finis définies sur des maillages distincts, utilisable avec des ordres polynomiaux quelconques et des éléments courbes (*Chapitre 5, [RL17]*) ;
- comparaisons de solutions éléments finis obtenues avec des maillages tétraédriques, hexaédriques et hex-dominants (hexa-tétraédriques) sur des problèmes de Poisson et d'élasticité linéaire (§4.3) grâce à l'outil de calcul de distances.

En conclusion, nous donnons notre perception sur ces contributions et sur l'état de l'art associé. Nous évoquons également des perspectives possibles pour la génération de maillages entièrement hexaédriques.

Génération de maillages hex-dominants.

Pour envisager une adoption courante des maillages hex-dominants (alignés avec les bords des objets) dans un contexte industriel, il est nécessaire de proposer un mailleur robuste qui fonctionne sur toutes géométries, quitte à fournir des maillages composés de nombreux tétraèdres. Avec l'algorithme d'extraction et de maillage des cavités (§3.1), nous avons montré qu'il est possible de combiner les différents ingrédients actuels de l'état de l'art pour produire un tel mailleur hex-dominant robuste.

De plus, il nous parait très utile de combiner des approches très différentes au sein d'un même mailleur *modulaire* afin de profiter des points forts de chaque technique. Par exemple, si l'algorithme de maillage contraint (maillage quadrilatéral donné en entrée) du *whisker weaving* n'est pas assez flexible pour produire des résultats satisfaisants sur des modèles compliqués, il est parfait pour mailler l'intérieur de certaines cavités.

Méthode des éléments finis et maillages hex-dominants.

Pour construire des espaces d'approximation conformes (*par ex.* H^1) sur les maillages hex-dominants, nous pouvons soit utiliser des pyramides et définir des éléments finis sur celles-ci, soit ajouter des contraintes de conformité aux maillages hexa-tétraédriques non-conformes (§4.2). Bien que nous ayons développé la seconde approche dans cette thèse, nous pensons qu'elle n'est pas pratique dans un contexte industriel car elle exige trop de modifications par rapport à un code d'éléments finis standard. Avec le recul, notre estimons est qu'il est préférable d'utiliser des éléments finis pyramidaux construits à partir de fonctions rationnelles, par exemple : [BCD09 ; Fue+15].

Les techniques actuelles sont suffisamment matures pour générer et exploiter les maillages hex-dominants dans le cadre de la méthode des éléments finis.

Intérêt des maillages hexaédriques et hex-dominants.

S'il est aujourd'hui possible de générer automatiquement des maillages hex-dominants, cela reste significativement plus difficile que de générer des maillages tétraédriques. Est-ce que l'avantage numérique offert par ces maillages justifie l'effort ?

Dans l'état de l'art sur la génération de maillages hexaédriques, il est généralement admis que ceux-ci sont supérieurs aux maillages tétraédriques. Cependant à notre connaissance il existe très peu d'études comparatives rigoureuses sur le sujet, et celles-ci ne sont pas forcément représentatives de la réalité. Une grande partie génère des maillages tétraédriques en décomposant les hexaèdres en six tétraèdres, or ces maillages réguliers se comportent beaucoup moins bien que des maillages non-structurés avec répartition aléatoire des éléments (§4.1).

Les comparaisons que nous avons menées sur des équations simples (Poisson, élasticité linéaire) avec des éléments finis classiques ($\mathbb{P}_k, \mathbb{Q}_k$) ne mettent pas en évidence un avantage décisif des maillages hexaédriques et hex-dominants. Cependant, nous pensons que ces comparaisons basiques ne sont pas représentatives d'applications avancées, pour lesquelles il est tout à fait possible que les hexaèdres soient bien plus intéressants en termes de précision pour un temps de calcul donné.

Il serait important de mener des investigations poussées sur les différences entre maillages tétraédriques et hexaédriques sur des problèmes avec des équations avancées et des géométries compliquées. Dans cette direction, nous avons développé une méthode efficace de calcul de distances (*Chapitre 5*) qui peut être utilisée pour estimer l'erreur globale associée aux différentes solutions. Certes, les erreurs en normes L^p ou H^1 ne sont pas forcément les quantités les plus pertinentes à étudier pour un problème spécifique, mais elles ont l'avantage d'être génériques et assez neutres pour envisager une comparaison systématique sur un grand nombre de problèmes.

Cependant, même avec un outil de calcul de distances efficace, notre expérience a montré qu'il est difficile de mettre en oeuvre des études de convergence avec le raffinement pour différents types de maillages parce que des phénomènes secondaires influencent l'erreur, comme la précision sur le maillage des bords ou la discrétisation des termes sources et des conditions aux limites.

Vers des maillages entièrement hexaédriques.

Générer automatiquement des maillages entièrement hexaédriques alignés avec le

bord, pour des géométries quelconques, reste un problème très difficile avec encore de nombreux défis à surmonter.

Nous sommes convaincus que les champs de directions sont un élément important de la solution, cependant il reste des défauts à corriger. La difficulté n'est pas tellement de les rendre plus lisses, mais de les contraindre plus spécifiquement aux nécessités du maillage hexaédrique. Viertel et al. [VSL16] ont montré que même les champs de directions respectant les contraintes proposées par Li et al. [Li+12] ne correspondent pas toujours à des maillages hexaédriques réalisables. Il faut s'assurer que les singularités linéiques sont parallèles avec la direction stable du champ de directions local, or ce n'est pas garanti par les techniques de génération actuelles.

L'approche par paramétrisation *CubeCover* [NRP11 ; Li+12] est celle qui permet d'obtenir les meilleurs résultats en termes de maillages entièrement hexaédriques. Cependant elle n'est pas suffisante car dans sa formulation courante, elle est incapable de gérer des configurations de types *rampes* (Figure 3.7, p. 85), fréquentes dans les pièces mécaniques.

Plus généralement, la tendance actuelle est de générer un champ de directions puis de l'utiliser pour produire un maillage hexaédrique (avec une approche indirecte ou par paramétrisation). Cette approche séquentielle manque de flexibilité : il serait intéressant de pouvoir modifier le champ de directions, et particulièrement son graphe de singularités, en fonction des besoins des étapes de maillage. L'idéal serait de construire une sorte de couplage entre ces deux étapes distinctes.

Une piste prometteuse pour la suite, à notre avis, serait de combiner les champs de directions et la représentation duale sous forme de feuilles (STC) des maillages hexaédriques. Ces deux structures sont complémentaires : les champs de directions caractérisent l'orientation locale des hexaèdres (information géométrique) et le STC incorpore les contraintes combinatoires globales qui se propagent dans tout le volume (information topologique). Cependant, le couplage de ces deux idées est sans aucun doute une tâche difficile, ces objets étant de natures très différentes.

Bibliographie

[AA87] Michael A Morrison et Gregory A Parker. « A Guide to Rotations in Quantum Mechanics ». In : *Aust. J. Phys.* 40.4 (1987), p. 465. ISSN : 0004-9506. DOI : 10.1071/ph870465.

[AAD11] Mark Ainsworth, Gaelle Andriamaro et Oleg Davydov. « Bernstein–Bézier Finite Elements of Arbitrary Order and Optimal Assembly Procedures ». In : *SIAM J. Sci. Comput.* 33.6 (jan. 2011), p. 3087-3109. ISSN : 1064-8275, 1095-7197. DOI : 10.1137/11082539x.

[ADS16] Mark Ainsworth, Oleg Davydov et Larry L. Schumaker. « Bernstein-Bézier finite elements on tetrahedral–hexahedral–pyramidal partitions ». In : *Computer Methods in Applied Mechanics and Engineering* 304 (juin 2016), p. 140-170. ISSN : 0045-7825. DOI : 10.1016/j.cma.2016.01.021.

[AF17] Mark Ainsworth et Guosheng Fu. « A lowest-order composite finite element exact sequence on pyramids ». In : *Computer Methods in Applied Mechanics and Engineering* 324 (sept. 2017), p. 110-127. ISSN : 0045-7825. DOI : 10.1016/j.cma.2017.05.030.

[AFW06] Douglas N. Arnold, Richard S. Falk et Ragnar Winther. « Finite element exterior calculus, homological techniques, and applications ». In : *ANU* 15 (mai 2006), p. 1. ISSN : 0962-4929, 1474-0508. DOI : 10.1017/s0962492906210018.

[AFW10] Douglas N. Arnold, Richard S. Falk et Ragnar Winther. « Finite element exterior calculus : From Hodge theory to numerical stability ». In : *Bull. Amer. Math. Soc.* 47.2 (jan. 2010), p. 281-354. ISSN : 0273-0979. DOI : 10.1090/s0273-0979-10-01278-4.

[AK91] Doi Akio et Akio Koide. « An efficient method of triangulating equivalued surfaces by using tetrahedral cells ». In : *IEICE Transactions on Information and Systems* 74.1 (1991), p. 214-224.

[AL14] Douglas N Arnold et Anders Logg. « Periodic table of the finite elements ». In : *SIAM News* 47.9 (2014), p. 212.

[AL16] Frédéric Alauzet et Adrien Loseille. « A decade of progress on anisotropic mesh adaptation for computational fluid dynamics ». In : *Computer-Aided Design* 72 (mar. 2016), p. 13-39. ISSN : 0010-4485. DOI : 10.1016/j.cad.2015.09.005.

[Ala16] Frédéric Alauzet. « A parallel matrix-free conservative solution interpolation on unstructured tetrahedral meshes ». In : *Computer Methods in Applied Mechanics and Engineering* 299 (fév. 2016), p. 116-142. ISSN : 0045-7825. DOI : 10.1016/j.cma.2015.10.012.

[All05] Grégoire Allaire. *Analyse numérique et optimisation : Une introduction à la modélisation mathématique et à la simulation numérique*. Editions Ecole Polytechnique, 2005.

[Arm+15] Cecil G. Armstrong, Harold J. Fogg, Christopher M. Tierney et Trevor T. Robinson. « Common Themes in Multi-block Structured Quad/Hex Mesh Generation ». In : *Procedia Engineering* 124 (2015), p. 70-82. ISSN : 1877-7058. DOI : 10.1016/j.proeng.2015.10.123.

[ASE02] N. Aspert, D. Santa-Cruz et T. Ebrahimi. « MESH : Measuring errors between surfaces using the Hausdorff distance ». In : *Proceedings. IEEE International Conference on Multimedia and Expo*. T. 1. IEEE. IEEE, 2002, p. 705-708. ISBN : 0780373049. DOI : 10.1109/icme.2002.1035879.

[Bak97] Timothy J. Baker. « Mesh adaptation strategies for problems in fluid dynamics ». In : *Finite Elements in Analysis and Design* 25.3-4 (avr. 1997), p. 243-273. ISSN : 0168-874X. DOI : 10.1016/s0168-874x(96)00032-7.

[Bat06] Klaus-Jürgen Bathe. *Finite element procedures*. Klaus-Jurgen Bathe, 2006.

[Bau+14] Tristan Baudouin, Jean-François Remacle, Emilie Marchandise, François Henrotte et Christophe Geuzaine. « A frontal approach to hex-dominant mesh generation ». In : *Advanced Modeling and Simulation in Engineering Sciences* 1.1 (2014), p. 8. ISSN : 2213-7467. DOI : 10.1186/2213-7467-1-8.

[BBP14] Philippe Bussetta, Romain Boman et Jean-Philippe Ponthot. « Efficient 3D data transfer operators based on numerical integration ». In : *Int. J. Numer. Meth. Engng* 102.3-4 (nov. 2014), p. 892-929. ISSN : 0029-5981. DOI : 10.1002/nme.4821.

[BBS85] J. Benek, P. Buning et J. Steger. « A 3-D chimera grid embedding technique ». In : *7th Computational Physics Conference*. American Institute of Aeronautics et Astronautics, juil. 1985. DOI : 10.2514/6.1985-1523.

[BCD09] Morgane Bergot, Gary Cohen et Marc Duruflé. « Higher-order Finite Elements for Hybrid Meshes Using New Nodal Pyramidal Elements ». In : *J Sci Comput* 42.3 (nov. 2009), p. 345-381. ISSN : 0885-7474, 1573-7691. DOI : 10.1007/s10915-009-9334-9.

[BD13] Morgane Bergot et Marc Duruflé. « High-order optimal edge elements for pyramids, prisms and hexahedra ». In : *Journal of Computational Physics* 232.1 (jan. 2013), p. 189-213. ISSN : 0021-9991. DOI : 10.1016/j.jcp.2012.08.005.

[Bed92] G. Bedrosian. « Shape functions and integration formulas for three-dimensional finite element analysis ». In : *Int. J. Numer. Meth. Engng.* 35.1 (juil. 1992), p. 95-108. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/nme.1620350106.

[Bei+13] L. Beirão Da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini et A. Russo. « Basic Principles Of Virtual Element Methods ». In : *Math. Models Methods Appl. Sci.* 23.01 (jan. 2013), p. 199-214. ISSN : 0218-2025, 1793-6314. DOI : 10.1142/s0218202512500492.

[Ben+95] Steven E Benzley, Ernest Perry, Karl Merkley, Brett Clark et Greg Sjaardama. « A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elasto-plastic analysis ». In : *Proceedings, 4th International Meshing Roundtable*. T. 17. Sandia National Laboratories Albuquerque, NM. 1995, p. 179-191.

[Ber+16] P.-E. Bernard, J.-F. Remacle, N. Kowalski et C. Geuzaine. « Frame field smoothness-based approach for hex-dominant meshing ». In : *Computer-Aided Design* 72 (mar. 2016), p. 78-86. ISSN : 0010-4485. DOI : 10.1016/j.cad.2015.10.003.

[Ber10] Morgane Bergot. « Éléments finis d'ordre élevé pour maillages hybrides-Application à la résolution de systèmes hyperboliques linéaires en régimes harmonique et temporel ». Thèse de doct. Université Paris Dauphine-Paris IX, 2010.

[BH04] M. Brasher et R. Haimes. « Rendering planar cuts through quadratic and cubic finite elements ». In : *IEEE Visualization 2004*. IEEE. IEEE Comput. Soc, 2004, p. 409-416. ISBN : 0780387880. DOI : 10.1109/visual.2004.91.

[BHM00] William L. Briggs, Van Emden Henson et Steve F. McCormick. *A Multigrid Tutorial, Second Edition*. Society for Industrial et Applied Mathematics, jan. 2000. ISBN : 9780898714623, 9780898719505. DOI : 10.1137/1.9780898719505.

[Bla96] Ted Blacker. « The cooper tool ». In : *5th International Meshing Roundtable, SAND 95-2130, Sandia National Laboratories*. Citeseer. 1996.

[BLC14] A.B. Botella, B.L. Lévy et G.C. Caumon. « Indirect Unstructured Hex-dominant Mesh Generation Using Tetrahedra Recombination ». In : *ECMOR XIV - 14th European Conference on the Mathematics of Oil Recovery*. EAGE Publications BV, sept. 2014. ISBN : 9789073834941. DOI : 10.3997/2214-4609.20141857.

[BM93] Ted D. Blacker et Ray J. Meyers. « Seams and wedges in plastering : A 3-D hexahedral mesh generation algorithm ». In : *Engineering with Computers* 9.2 (juin 1993), p. 83-93. ISSN : 0177-0667, 1435-5663. DOI : 10.1007/bf01199047.

[Bou+07] Xavier Bourdin, Xavier Trosseille, Philippe Petit et Philippe Beillas. « Comparison of tetrahedral and hexahedral meshes for organ finite element modeling : An application to kidney impact ». In : *20th International technical conference on the enhanced safety of vehicle, Lyon*. 2007.

[BS91] Ted D. Blacker et Michael B. Stephenson. « Paving : A new approach to automated quadrilateral mesh generation ». In : *Int. J. Numer. Meth. Engng.* 32.4 (sept. 1991), p. 811-847. issn : 0029-5981, 1097-0207. doi : 10.1002/nme.1620320410.

[BW07] M. J. Bluck et S. P. Walker. « Polynomial basis functions on pyramidal elements ». In : *Commun. Numer. Meth. Engng.* 24.12 (nov. 2007), p. 1827-1837. issn : 1069-8299. doi : 10.1002/cnm.1070.

[BW76] Klaus-Jürgen Bathe et Edward L Wilson. *Numerical methods in finite element analysis*. T. 197. Prentice-Hall Englewood Cliffs, NJ, 1976.

[BZK09] David Bommes, Henrik Zimmer et Leif Kobbelt. « Mixed-integer quadrangulation ». In : *Transactions on Graphics (TOG)* 28.3 (juil. 2009), p. 1. issn : 0730-0301. doi : 10.1145/1531326.1531383.

[Can92] SCOTT Canann. *Plastering : A new approach to automated, 3-D hexahedral mesh generation*. Rapp. tech. Sandia National Labs., Albuquerque, NM (United States), 1992.

[Car13] Tristan Carrier Baudouin. « Hexahedral-dominant mesh generation ». Thèse de doct. 2013.

[Cea64] Jean Cea. « Approximation variationnelle des problèmes aux limites ». In : *Annales de l'institut Fourier* 14.2 (1964), p. 345-444. issn : 0373-0956. doi : 10.5802/aif.181.

[CGL09] Bernardo Cockburn, Jayadeep Gopalakrishnan et Raytcho Lazarov. « Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems ». In : *SIAM Journal on Numerical Analysis* 47.2 (2009), p. 1319-1365.

[Cha+16] Jesse Chan, Zheng Wang, Axel Modave, Jean-Francois Remacle et T. Warburton. « GPU-accelerated discontinuous Galerkin methods on hybrid meshes ». In : *Journal of Computational Physics* 318 (août 2016), p. 142-168. issn : 0021-9991. doi : 10.1016/j.jcp.2016.04.003.

[Cia91] Philippe G Ciarlet. « Basic error estimates for elliptic problems ». In : *Handbook of numerical analysis* 2 (1991), p. 17-351.

[CK92] A.O. Cifuentes et A. Kalbag. « A performance study of tetrahedral and hexahedral elements in 3-D finite element structural analysis ». In : *Finite Elements in Analysis and Design* 12.3-4 (déc. 1992), p. 313-318. issn : 0168-874X. doi : 10.1016/0168-874x(92)90040-j.

[CO78] Philippe G. Ciarlet et J. T. Oden. « The Finite Element Method for Elliptic Problems ». In : *J. Appl. Mech.* 45.4 (1978), p. 968. issn : 0021-8936. doi : 10.1115/1.3424474.

[Coo+02] Robert D Cook, David S Malkus, Michael E Plesha et Robert J Witt. *Concepts and applications of finite element analysis*. Wiley New York, 2002.

[Cou43] R. Courant. « Variational methods for the solution of problems of equilibrium and vibrations ». In : *Bull. Amer. Math. Soc.* 49.1 (jan. 1943), p. 1-24. ISSN : 0002-9904. DOI : 10.1090/s0002-9904-1943-07818-4.

[CRS98] Paolo Cignoni, Claudio Rocchini et Roberto Scopigno. « Metro : Measuring error on simplified surfaces ». In : *Computer Graphics Forum*. T. 17. 2. Wiley Online Library. 1998, p. 167-174.

[CW16] Jesse Chan et T. Warburton. « Orthogonal Bases for Vertex-Mapped Pyramids ». In : *SIAM J. Sci. Comput.* 38.2 (jan. 2016), A1146-A1170. ISSN : 1064-8275, 1095-7197. DOI : 10.1137/15m1011408.

[CZM97] J.-L. Coulomb, F.-X. Zgainski et Y. Marechal. « A pyramidal element to link hexahedral, prismatic and tetrahedral edge finite elements ». In : *IEEE Trans. Magn.* 33.2 (mar. 1997), p. 1362-1365. ISSN : 0018-9464. DOI : 10.1109/20.582509.

[Dav06] Timothy A. Davis. *Direct Methods for Sparse Linear Systems*. Society for Industrial et Applied Mathematics, jan. 2006. ISBN : 9780898716139, 9780898718881. DOI : 10.1137/1.9780898718881.

[DAW16] Kent T. Danielson, Mark D. Adley et T. Neil Williams. « Second-order finite elements for Hex-Dominant explicit methods in nonlinear solid dynamics ». In : *Finite Elements in Analysis and Design* 119 (oct. 2016), p. 63-77. ISSN : 0168-874X. DOI : 10.1016/j.finel.2016.02.008.

[DE15] Daniele A. Di Pietro et Alexandre Ern. « A hybrid high-order locking-free method for linear elasticity on general meshes ». In : *Computer Methods in Applied Mechanics and Engineering* 283 (jan. 2015), p. 1-21. ISSN : 0045-7825. DOI : 10.1016/j.cma.2014.09.009.

[Dew+93] Donald L Dewhirst, Peter M Grinsell, John R Tucker et Alok Mahajan. « Joining tetrahedra to hexahedra ». In : *Proceedings of MSC World Users' Conference*. MSC Software. 1993.

[DK91] Akio Doi et Akio Koide. « An efficient method of triangulating equi-valued surfaces by using tetrahedral cells ». In : *IEICE TRANSACTIONS on Information and Systems* 74.1 (1991), p. 214-224.

[DLS13] Clément Durochat, Stéphane Lanteri et Claire Scheid. « High order non-conforming multi-element discontinuous Galerkin method for time domain electromagnetics ». In : *Applied Mathematics and Computation* 224 (nov. 2013), p. 681-704. ISSN : 0096-3003. DOI : 10.1016/j.amc.2013.08.069.

[DR06] D. Degerfeldt et T. Rylander. « A brick-tetrahedron finite-element interface with stable hybrid explicit–implicit time-stepping for Maxwell's equations ». In : *Journal of Computational Physics* 220.1 (déc. 2006), p. 383-393. ISSN : 0021-9991. DOI : 10.1016/j.jcp.2006.05.016.

[Dur13] Clément Durochat. « High order non-conforming multi-element discontinuous Galerkin method for time-domain electromagnetics ». Theses. Université Nice Sophia Antipolis, 2013.

[Ebk+13] Hans-Christian Ebke, David Bommes, Marcel Campen et Leif Kobbelt. « QEx ». In : *Transactions on Graphics (TOG)* 32.6 (nov. 2013), p. 1-10. ISSN : 0730-0301. DOI : 10.1145/2508363.2508372.

[EG02] Alexandre Ern et Jean-Luc Guermond. *Eléments finis : Théorie, applications, mise en oeuvre*. T. 36. Springer Science & Business Media, 2002.

[FA05] P.J. Frey et F. Alauzet. « Anisotropic mesh adaptation for CFD computations ». In : *Computer Methods in Applied Mechanics and Engineering* 194.48-49 (nov. 2005), p. 5068-5082. ISSN : 0045-7825. DOI : 10.1016/j.cma.2004.11.025.

[Fah15] Hassan Fahs. « High-order discontinuous Galerkin method for time-domain electromagnetics on non-conforming hybrid meshes ». In : *Mathematics and Computers in Simulation* 107 (2015), p. 134-156.

[Fan+16] Xianzhong Fang, Weiwei Xu, Hujun Bao et Jin Huang. « All-hex meshing using closed-form induced polycube ». In : *Transactions on Graphics (TOG)* 35.4 (juil. 2016), p. 1-9. ISSN : 0730-0301. DOI : 10.1145/2897824.2925957.

[Far+09] P.E. Farrell, M.D. Piggott, C.C. Pain, G.J. Gorman et C.R. Wilson. « Conservative interpolation between unstructured meshes via supermesh construction ». In : *Computer Methods in Applied Mechanics and Engineering* 198.33-36 (juil. 2009), p. 2632-2642. ISSN : 0045-7825. DOI : 10.1016/j.cma.2009.03.004.

[Far11] P.E. Farrell. « The addition of fields on different meshes ». In : *Journal of Computational Physics* 230.9 (mai 2011), p. 3265-3269. ISSN : 0021-9991. DOI : 10.1016/j.jcp.2011.01.028.

[FAR16] Harold J. Fogg, Cecil G. Armstrong et Trevor T. Robinson. « Enhanced medial-axis-based block-structured meshing in 2-D ». In : *Computer-Aided Design* 72 (mar. 2016), p. 87-101. ISSN : 0010-4485. DOI : 10.1016/j.cad.2015.07.001.

[Fel] Carlos Felippa. *The Origins of the Finite Element Method*.

[FM99] N. T. Folwell et S. A. Mitchell. « Reliable Whisker Weaving via Curve Contraction ». In : *Engineering with Computers* 15.3 (sept. 1999), p. 292-302. ISSN : 0177-0667, 1435-5663. DOI : 10.1007/s003660050024.

[Fog+17] Harold J. Fogg, Liang Sun, Jonathan E. Makem, Cecil G. Armstrong et Trevor T. Robinson. « A simple formula for quad mesh singularities ». In : *Procedia Engineering* 203 (2017), p. 14-26. ISSN : 1877-7058. DOI : 10.1016/j.proeng.2017.09.781.

[FPO97] Y.T. Feng, D. Perić et D.R.J. Owen. « A non-nested Galerkin multigrid method for solving linear and nonlinear solid mechanics problems ». In : *Computer Methods in Applied Mechanics and Engineering* 144.3-4 (mai 1997), p. 307-325. ISSN : 0045-7825. DOI : 10.1016/s0045-7825(96)01183-8.

[Fue+15] Federico Fuentes, Brendan Keith, Leszek Demkowicz et Sriram Nagaraj. « Orientation embedded high order shape functions for the exact sequence elements of all shapes ». In : *Computers & Mathematics with Applications* 70.4 (août 2015), p. 353-458. ISSN : 0898-1221. DOI : 10.1016/j.camwa.2015.04.027.

[FY02] Robert D Falgout et Ulrike Meier Yang. « hypre : A library of high performance preconditioners ». In : *International Conference on Computational Science*. Springer. 2002, p. 632-641.

[Gao+17a] Xifeng Gao, Jin Huang, Kaoji Xu, Zherong Pan, Zhigang Deng et Guoning Chen. « Evaluating Hex-mesh Quality Metrics via Correlation Analysis ». In : *Computer Graphics Forum*. T. 36. 5. Wiley Online Library. 2017, p. 105-116.

[Gao+17b] Xifeng Gao, Wenzel Jakob, Marco Tarini et Daniele Panozzo. « Robust hex-dominant mesh generation using field-guided polyhedral agglomeration ». In : *Transactions on Graphics (TOG)* 36.4 (juil. 2017), p. 1-13. ISSN : 0730-0301. DOI : 10.1145/3072959.3073676.

[GB96] Christiane Görller-Walrand et Koen Binnemans. « Rationalization of crystal-field parametrization ». In : *Handbook on the physics and chemistry of rare earths* 23 (1996), p. 121-283.

[Geu+99] C. Geuzaine, B. Meys, F. Henrotte, P. Dular et W. Legros. « A Galerkin projection method for mixed finite elements ». In : *IEEE Trans. Magn.* 35.3 (mai 1999), p. 1438-1441. ISSN : 0018-9464. DOI : 10.1109/20.767236.

[GH95] A. Gueziec et R. Hummel. « Exploiting triangulated surface extraction using tetrahedral decomposition ». In : *IEEE Trans. Visual. Comput. Graphics* 1.4 (1995), p. 328-342. ISSN : 1077-2626. DOI : 10.1109/2945.485620.

[GHS90] P.L. George, F. Hecht et E. Saltel. « Fully automatic mesh generator for 3D domains of any shape ». In : *IMPACT of Computing in Science and Engineering* 2.3 (sept. 1990), p. 187-218. ISSN : 0899-8248. DOI : 10.1016/0899-8248(90)90012-y.

[Gri11] Pierre Grisvard. *Elliptic Problems in Nonsmooth Domains*. Society for Industrial et Applied Mathematics, jan. 2011. ISBN : 9781611972023, 9781611972030. DOI : 10.1137/1.9781611972030.

[GSZ11] James Gregson, Alla Sheffer et Eugene Zhang. « All-Hex Mesh Generation via Volumetric PolyCube Deformation ». In : *Computer Graphics Forum* 30.5 (août 2011), p. 1407-1416. ISSN : 0167-7055. DOI : 10.1111/j.1467-8659.2011.02015.x.

[Haa+03] Bernard Haasdonk, Mario Ohlberger, Martin Rumpf, Alfred Schmidt et Kunibert G Siebert. « Multiresolution visualization of higher order adaptive finite element simulations ». In : *Computing* 70.3 (2003), p. 181-204.

[Har13] Matthew Harris. « Flow feature aligned mesh generation and adaptation ». Thèse de doct. University of Sheffield, 2013.

[HCB05] T.J.R. Hughes, J.A. Cottrell et Y. Bazilevs. « Isogeometric analysis : Cad, finite elements, NURBS, exact geometry and mesh refinement ». In : *Computer Methods in Applied Mechanics and Engineering* 194.39-41 (oct. 2005), p. 4135-4195. ISSN : 0045-7825. DOI : 10.1016/j.cma.2004.10.008.

[Hua+11] Jin Huang, Yiying Tong, Hongyu Wei et Hujun Bao. « Boundary aligned smooth 3D cross-frame field ». In : *Proceedings of the 2011 SIGGRAPH Asia Conference on - SA '11*. ACM Press, 2011. ISBN : 9781450308076. DOI : 10.1145/2024156.2024177.

[Hua+14] Jin Huang, Tengfei Jiang, Zeyun Shi, Yiying Tong, Hujun Bao et Mathieu Desbrun. « L1-Based Construction of Polycube Maps from Complex Shapes ». In : *Transactions on Graphics (TOG)* 33.3 (juin 2014), p. 1-11. ISSN : 0730-0301. DOI : 10.1145/2602141.

[HW08] Jan S. Hesthaven et Tim Warburton. *Nodal Discontinuous Galerkin Methods*. Springer New York, 2008. ISBN : 9780387720654, 9780387720678. DOI : 10.1007/978-0-387-72067-8.

[HY02] Van Emden Henson et Ulrike Meier Yang. « BoomerAMG : A parallel algebraic multigrid solver and preconditioner ». In : *Applied Numerical Mathematics* 41.1 (avr. 2002), p. 155-177. ISSN : 0168-9274. DOI : 10.1016/s0168-9274(01)00115-5.

[Iba+16] Daniel A. Ibanez, E. Seegyoung Seol, Cameron W. Smith et Mark S. Shephard. « Pumi ». In : *ACM Trans. Math. Softw.* 42.3 (mai 2016), p. 1-28. ISSN : 0098-3500. DOI : 10.1145/2814935.

[Iro72] Bruce Irons. *An assumed-stress version of the Wilson 8-node isoparametric brick*. University College of Swansea. Centre for Numerical Methods in Engineering, 1972.

[ISS09] Yasushi Ito, Alan M. Shih et Bharat K. Soni. « Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates ». In : *Int. J. Numer. Meth. Engng* 77.13 (mar. 2009), p. 1809-1833. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/nme.2470.

[Jak+15] Wenzel Jakob, Marco Tarini, Daniele Panozzo et Olga Sorkine-Hornung. « Instant field-aligned meshes ». In : *Transactions on Graphics (TOG)* 34.6 (oct. 2015), p. 1-15. ISSN : 0730-0301. DOI : 10.1145/2816795.2818078.

[JH04] Xiangmin Jiao et Michael T. Heath. « Common-refinement-based data transfer between non-matching meshes in multiphysics simulations ». In : *Int. J. Numer. Meth. Engng.* 61.14 (2004), p. 2402-2427. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/nme.1147.

[Jia+14] Tengfei JIANG, Jin HUANG, Yuanzhen WANG, Yiying TONG et Hujun BAO. « Frame Field Singularity Correctionfor Automatic Hexahedralization ». In : *IEEE Trans. Visual. Comput. Graphics* 20.8 (août 2014), p. 1189-1199. ISSN : 1077-2626. DOI : 10.1109/tvcg.2013.250.

[KLF14] N. KOWALSKI, F. LEDOUX et P. FREY. « Block-structured Hexahedral Meshes for CAD Models Using 3D Frame Fields ». In : *Procedia Engineering* 82 (2014), p. 59-71. ISSN : 1877-7058. DOI : 10.1016/j.proeng.2014.10.373.

[KLF16] N. KOWALSKI, F. LEDOUX et P. FREY. « Smoothness driven frame field generation for hexahedral meshing ». In : *Computer-Aided Design* 72 (mar. 2016), p. 65-77. ISSN : 0010-4485. DOI : 10.1016/j.cad.2015.06.009.

[KNP07] Felix KÄLBERER, Matthias NIESER et Konrad POLTHIER. « QuadCover - Surface Parameterization using Branched Coverings ». In : *Computer Graphics Forum* 26.3 (sept. 2007), p. 375-384. ISSN : 0167-7055, 1467-8659. DOI : 10.1111/j.1467-8659.2007.01060.x.

[Knu98] Patrick M KNUPP. « Next-Generation Sweep Tool : A Method for Generating All-Hex Meshes on Two-and-One-Half Dimensional Geometries ». In : *IMR*. 1998, p. 505-513.

[Knu99] Patrick M KNUPP. *Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities, Part II-A framework for volume mesh optimization and the condition number of the Jacobian matrix*. Rapp. tech. Sandia National Labs., Albuquerque, NM (US) ; Sandia National Labs., Livermore, CA (US), 1999.

[Kow13] Nicolas KOWALSKI. « Domain partitioning using frame fields : Applications to quadrilateral and hexahedral meshing ». Thèse de doct. Paris 6, 2013.

[Kre+14] Michael KREMER, David BOMMES, Isaak LIM et Leif KOBBELT. « Advanced Automatic Hexahedral Mesh Generation from Surface Quad Meshes ». In : *Proceedings of the 22nd International Meshing Roundtable*. Springer International Publishing, 2014, p. 147-164. ISBN : 9783319023342, 9783319023359. DOI : 10.1007/978-3-319-02335-9_9.

[KS] GE KARNIADAKIS et SJ SHERWIN. *Spectral/hp element methods for CFD. 1999*.

[LBK16] Max LYON, David BOMMES et Leif KOBBELT. « HexEx ». In : *Transactions on Graphics (TOG)* 35.4 (juil. 2016), p. 1-11. ISSN : 0730-0301. DOI : 10.1145/2897824.2925976.

[LC87] William E. LORENSEN et Harvey E. CLINE. « Marching cubes : A high resolution 3D surface construction algorithm ». In : *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87*. ACM Press, 1987. ISBN : 0897912276. DOI : 10.1145/37401.37422.

[LDA10] A. Loseille, A. Dervieux et F. Alauzet. « Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations ». In : *Journal of Computational Physics* 229.8 (avr. 2010), p. 2866-2897. ISSN : 0021-9991. DOI : 10.1016/j.jcp.2009.12.021.

[Leo+98] Andrea O Leone, Paola Marzano, Enrico Gobbetti, Riccardo Scateni et Sergio Pedinotti. « Discontinuous finite element visualization ». In : *Proceedings 8th International Symposium on Flow Visualization*. 1998.

[Li+12] Yufei Li, Yang Liu, Weiwei Xu, Wenping Wang et Baining Guo. « All-hex meshing using singularity-restricted field ». In : *Transactions on Graphics (TOG)* 31.6 (nov. 2012), p. 1. ISSN : 0730-0301. DOI : 10.1145/2366145.2366196.

[Liu+04] Liping Liu, Kevin B Davies, Kewei Yuan et M Krizek. « On symmetric pyramidal finite elements ». In : *Dynamics of Continuous Discrete and Impulsive Systems Series B* 11 (2004), p. 213-228.

[Liu+11] Liping Liu, Kevin B. Davies, Michal Křížek et Li Guan. « On Higher Order Pyramidal Finite Elements ». In : *Adv. Appl. Math. Mech.* 3.02 (avr. 2011), p. 131-140. ISSN : 2070-0733, 2075-1354. DOI : 10.4208/aamm.09-m0989.

[Liv+13] Marco Livesu, Nicholas Vining, Alla Sheffer, James Gregson et Riccardo Scateni. « PolyCut ». In : *Transactions on Graphics (TOG)* 32.6 (nov. 2013), p. 1-12. ISSN : 0730-0301. DOI : 10.1145/2508363.2508388.

[Liv+15] Marco Livesu, Alla Sheffer, Nicholas Vining et Marco Tarini. « Practical hex-mesh optimization via edge-cone rectification ». In : *Transactions on Graphics (TOG)* 34.4 (juil. 2015), 141 :1-141 :11. ISSN : 0730-0301. DOI : 10.1145/2766905.

[LL10] Bruno Lévy et Yang Liu. « LpCentroidal Voronoi Tessellation and its applications ». In : *Transactions on Graphics (TOG)* 29.4 (juil. 2010), p. 1. ISSN : 0730-0301. DOI : 10.1145/1778765.1778856.

[LS09] Franck Ledoux et Jason Shepherd. « Topological and geometrical properties of hexahedral meshes ». In : *Engineering with Computers* 26.4 (oct. 2009), p. 419-432. ISSN : 0177-0667, 1435-5663. DOI : 10.1007/s00366-009-0144-3.

[LS10] Franck Ledoux et Jason Shepherd. « Topological modifications of hexahedral meshes via sheet operations : A theoretical study ». In : *Engineering with Computers* 26.4 (2010), p. 433-447.

[LW08] Franck Ledoux et Jean-Christophe Weill. « An Extension of the Reliable Whisker Weaving Algorithm ». In : *Proceedings of the 16th International Meshing Roundtable*. Springer Berlin Heidelberg, 2008, p. 215-232. ISBN : 9783540751021. DOI : 10.1007/978-3-540-75103-8_13.

[Mac93] Richard MacNeal. *Finite Elements : Their Design and Performance*. CRC Press, 1993.

[Mar01] Loic Marechal. « A New Approach to Octree-Based Hexahedral Meshing. » In : *IMR*. 2001.

[Mar09] Loïc Maréchal. « Advances in Octree-Based All-Hexahedral Mesh Generation : Handling Sharp Features ». In : *Proceedings of the 18th International Meshing Roundtable*. Springer Berlin Heidelberg, 2009, p. 65-84. ISBN : 9783642043185, 9783642043192. DOI : 10.1007/978-3-642-04319-2_5.

[MD08] N. Marais et D.B. Davidson. « Conforming arbitrary order hexahedral/tetrahedral hybrid discretisation ». In : *Electron. Lett.* 44.24 (2008), p. 1384. ISSN : 0013-5194. DOI : 10.1049/el:20081861.

[Mfe17] Mfem. *MFEM : Modular finite element methods*. mfem.org. 2017.

[MGS01] J.M. Melenk, K. Gerdes et C. Schwab. « Fully discrete hp-finite elements : Fast quadrature ». In : *Computer Methods in Applied Mechanics and Engineering* 190.32-33 (mai 2001), p. 4339-4364. ISSN : 0045-7825. DOI : 10.1016/s0045-7825(00)00322-4.

[Mit13] William F. Mitchell. « A collection of 2D elliptic problems for testing adaptive grid refinement algorithms ». In : *Applied Mathematics and Computation* 220 (sept. 2013), p. 350-364. ISSN : 0096-3003. DOI : 10.1016/j.amc.2013.05.068.

[Mit96] Scott A Mitchell. « A characterization of the quadrilateral meshes of a surface which admit a compatible hexahedral mesh of the enclosed volume ». In : *Annual Symposium on Theoretical Aspects of Computer Science*. Springer. 1996, p. 465-476.

[Mmg17] Mmg. *Mmg platform : Robust, open-source and multidisciplinary software for remeshing*. https://www.mmgtools.org/. 2017.

[MS11] Sandeep Menon et David P. Schmidt. « Conservative interpolation on unstructured polyhedral meshes : An extension of the supermesh approach to cell-centered finite-volume variables ». In : *Computer Methods in Applied Mechanics and Engineering* 200.41-44 (oct. 2011), p. 2797-2804. ISSN : 0045-7825. DOI : 10.1016/j.cma.2011.04.025.

[MT00] Sia Meshkat et Dafna Talmor. « Generating a mixed mesh of hexahedra, pentahedra and tetrahedra from an underlying tetrahedral mesh ». In : *Int. J. Numer. Meth. Engng.* 49.12 (sept. 2000), p. 17-30. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/1097-0207(20000910/20)49:1/2<17::aid-nme920>3.3.co;2-l.

[Mül99] M. Müller-Hannemann. « Hexahedral Mesh Generation by Successive Dual Cycle Elimination ». In : *Engineering with Computers* 15.3 (sept. 1999), p. 269-279. ISSN : 0177-0667, 1435-5663. DOI : 10.1007/s003660050022.

[Mur+97] Peter Murdoch, Steven Benzley, Ted Blacker et Scott A. Mitchell. « The spatial twist continuum : A connectivity based method for representing all-hexahedral finite element meshes ». In : *Finite Elements in Analysis and Design* 28.2 (déc. 1997), p. 137-149. ISSN : 0168-874X. DOI : 10.1016/s0168-874x(97)81956-7.

[NHK11] B. Nelson, R. Haimes et R. M. Kirby. « GPU-Based Interactive Cut-Surface Extraction From High-Order Finite Element Fields ». In : *IEEE Trans. Visual. Comput. Graphics* 17.12 (déc. 2011), p. 1803-1811. ISSN : 1077-2626. DOI : 10.1109/tvcg.2011.206.

[Nit71] Joachim Nitsche. « Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind ». In : *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*. T. 36. 1. Springer. 1971, p. 9-15.

[NK06] B. Nelson et R.M. Kirby. « Ray-tracing polymorphic multidomain spectral/hp elements for isosurface rendering ». In : *IEEE Trans. Visual. Comput. Graphics* 12.1 (jan. 2006), p. 114-125. ISSN : 1077-2626. DOI : 10.1109/tvcg.2006.12.

[NP11] Nilima Nigam et Joel Phillips. « Numerical integration for high order pyramidal finite elements ». In : *ESAIM : M2AN* 46.2 (oct. 2011), p. 239-263. ISSN : 0764-583X, 1290-3841. DOI : 10.1051/m2an/2011042.

[NRP11] M. Nieser, U. Reitebuch et K. Polthier. « CubeCover- Parameterization of 3D Volumes ». In : *Computer Graphics Forum* 30.5 (août 2011), p. 1397-1406. ISSN : 0167-7055. DOI : 10.1111/j.1467-8659.2011.02014.x.

[OCS97] Steven J Owen, Scott A Canann et Sunil Saigal. « Pyramid elements for maintaining tetrahedra to hexahedra conformability ». In : (1997).

[Ors80] Steven A Orszag. « Spectral methods for problems in complex geometries ». In : *Journal of Computational Physics* 37.1 (août 1980), p. 70-92. ISSN : 0021-9991. DOI : 10.1016/0021-9991(80)90005-4.

[OS00] Steven J. Owen et Sunil Saigal. « H-Morph : An indirect approach to advancing front hex meshing ». In : *Int. J. Numer. Meth. Engng.* 49.1-2 (2000), p. 289-312. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/1097-0207(20000910/20)49:1/2<289::aid-nme934>3.0.co;2-l.

[OS14] Steven J. Owen et Tim R. Shelton. « Evaluation of grid-based hex meshes for solid mechanics ». In : *Engineering with Computers* 31.3 (juil. 2014), p. 529-543. ISSN : 0177-0667, 1435-5663. DOI : 10.1007/s00366-014-0368-8.

[Owe98] Steven J Owen. « A survey of unstructured mesh generation technology ». In : *IMR*. 1998, p. 239-267.

[PA97] Mark A Price et Cecil G Armstrong. « Hexahedral mesh generation by medial surface subdivision : Part II. Solids with flat and concave edges ». In : *International Journal for Numerical Methods in Engineering* 40.1 (1997), p. 111-136.

[PAS95] M. A. Price, C. G. Armstrong et M. A. Sabin. « Hexahedral mesh generation by medial surface subdivision : Part I. Solids with convex edges ». In : *Int. J. Numer. Meth. Engng.* 38.19 (oct. 1995), p. 3335-3359. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/nme.1620381910.

[Per85] Ken PERLIN. « An image synthesizer ». In : *SIGGRAPH Comput. Graph.* 19.3 (juil. 1985), p. 287-296. ISSN : 0097-8930. DOI : 10.1145/325165.325247.

[PHS98] S. PLIMPTON, B. HENDRICKSON et J. STEWART. « A Parallel Rendezvous Algorithm for Interpolation Between Multiple Grids ». In : *Proceedings of the IEEE/ACM SC98 Conference*. SC '98. San Jose, CA : IEEE, 1998, p. 1-8. ISBN : 081868707X. DOI : 10.1109/sc.1998.10032.

[Pia64] T. H. H. PIAN. « Derivation of element stiffness matrices by assumed stress distributions ». In : *AIAA Journal* 2.7 (juil. 1964), p. 1333-1336. ISSN : 0001-1452, 1533-385X. DOI : 10.2514/3.2546.

[PJR17] Jeanne PELLERIN, Amaury JOHNEN et Jean-Francois REMACLE. « Identifying combinations of tetrahedra into hexahedra : A vertex based strategy ». In : *Procedia Engineering* 203 (2017), p. 2-13. ISSN : 1877-7058. DOI : 10.1016/j.proeng.2017.09.779.

[PPM93] J. PERAIRE, J. PEIRO et K. MORGAN. « Multigrid solution of the 3-D compressible euler equations on unstructured tetrahedral grids ». In : *Int. J. Numer. Meth. Engng.* 36.6 (mar. 1993), p. 1029-1044. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/nme.1620360610.

[PR10] Richard PASQUETTI et Francesca RAPETTI. « Spectral element methods on unstructured meshes : Which interpolation points ? » In : *Numer Algor* 55.2-3 (mai 2010), p. 349-366. ISSN : 1017-1398, 1572-9265. DOI : 10.1007/s11075-010-9390-0.

[PT86] Theodore H. H. PIAN et Pin TONG. « Relations between incompatible displacement model and hybrid stress model ». In : *Int. J. Numer. Meth. Engng.* 22.1 (jan. 1986), p. 173-181. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/nme.1620220112.

[Pui+11] Guillaume PUIGT, Michel GAZAIX, M. MONTAGNAC, Jean-francois BOUSSUGE, Marie-Claire LE PAPE, Marta de la LLAVE PLATA, Claude MARMIGNON et Vincent COUAILLIER. « Development of a new hybrid compressible solver inside the CFD elsA software ». In : *20th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics et Astronautics, juin 2011, p. 3379. ISBN : 9781624101489. DOI : 10.2514/6.2011-3379.

[PW05] Theodore PIAN et Chang-Chun WU. *Hybrid and Incompatible Finite Element Methods*. Chapman et Hall/CRC, nov. 2005. ISBN : 9781584882763, 9780203487693. DOI : 10.1201/9780203487693.

[PZ07] Jonathan PALACIOS et Eugene ZHANG. « Rotational symmetry field design on surfaces ». In : *Transactions on Graphics (TOG)* 26.99 (juil. 2007), p. 55. ISSN : 0730-0301. DOI : 10.1145/1239451.1239506.

[QL06] Ning QIN et Xueqiang LIU. « Flow feature aligned grid adaptation ». In : *Int. J. Numer. Meth. Engng* 67.6 (2006), p. 787-814. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/nme.1648.

[Qua16] William Roshan QUADROS. « LayTracks3D : A new approach for meshing general solids using medial axis transform ». In : *Computer-Aided Design* 72 (2016), p. 102-117.

[Ray+06] Nicolas Ray, Wan Chiu Li, Bruno Lévy, Alla Sheffer et Pierre Alliez. « Periodic global parameterization ». In : *Transactions on Graphics (TOG)* 25.4 (oct. 2006), p. 1460-1485. ISSN : 0730-0301. DOI : 10.1145/1183287.1183297.

[Ray+08] Nicolas Ray, Bruno Vallet, Wan Chiu Li et Bruno Lévy. « N-symmetry direction field design ». In : *Transactions on Graphics (TOG)* 27.2 (avr. 2008), p. 1-13. ISSN : 0730-0301. DOI : 10.1145/1356682.1356683.

[Ray+17] Nicolas Ray, Dmitry Sokolov, Maxence Reberol, Franck Ledoux et Bruno Lévy. « Hexahedral Meshing : Mind the Gap ! » 2017.

[Reb17] Maxence Reberol. *FFES : Fast Finite Element Sampling*. https://github.com/mxncr/FFES. 2017.

[Rem+05] Jean-Francois Remacle, Xiangrong Li, Mark S. Shephard et Joseph E. Flaherty. « Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods ». In : *Int. J. Numer. Meth. Engng* 62.7 (2005), p. 899-923. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/nme.1196.

[Rem+06] Jean-François Remacle, Nicolas Chevaugeon, Émilie Marchandise et Christophe Geuzaine. « Efficient visualization of high-order finite elements ». In : *Int. J. Numer. Meth. Engng* 69.4 (2006), p. 750-771. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/nme.1787.

[Rem+13] J.-F. Remacle, F. Henrotte, T. Carrier-Baudouin, E. Béchet, E. Marchandise, C. Geuzaine et T. Mouton. « A frontal Delaunay quad mesh generator using the Linf norm ». In : *Int. J. Numer. Meth. Engng* 94.5 (fév. 2013), p. 494-512. ISSN : 0029-5981. DOI : 10.1002/nme.4458.

[Riv08] Béatrice Rivière. *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations*. Society for Industrial et Applied Mathematics, jan. 2008. ISBN : 9780898716566, 9780898717440. DOI : 10.1137/1.9780898717440.

[RL16] Maxence Reberol et Bruno Lévy. « Low-order continuous finite element spaces on hybrid non-conforming hexahedral-tetrahedral meshes ». 2016.

[RL17] Maxence Reberol et Bruno Lévy. « Computing the distance between two finite element solutions defined on different 3D meshes on a GPU ». In : *SIAM Journal on Scientific Computing (accepted)* (2017).

[Roa02] Patrick J. Roache. « Code Verification by the Method of Manufactured Solutions ». In : *J. Fluids Eng.* 124.1 (2002), p. 4. ISSN : 0098-2202. DOI : 10.1115/1.1436090.

[RRS09] Eloi Ruiz-Gironés, Xevi Roca et Josep Sarrate. « A New Procedure to Compute Imprints in Multi-sweeping Algorithms ». In : *Proceedings of the 18th International Meshing Roundtable*. Springer Berlin Heidelberg, 2009, p. 281-299. ISBN : 9783642043185, 9783642043192. DOI : 10.1007/978-3-642-04319-2_17.

[RS06] A. Ramos et J.A. Simões. « Tetrahedral versus hexahedral finite elements in numerical modelling of the proximal femur ». In : *Medical Engineering & Physics* 28.9 (nov. 2006), p. 916-924. ISSN : 1350-4533. DOI : 10.1016/j.medengphy.2005.12.006.

[RS15] Nicolas Ray et Dmitry Sokolov. « On smooth 3d frame field design ». In : *arXiv preprint arXiv :1507.03351* (2015).

[RSL16] Nicolas Ray, Dmitry Sokolov et Bruno Lévy. « Practical 3D frame field generation ». In : *Transactions on Graphics (TOG)* 35.6 (nov. 2016), p. 1-9. ISSN : 0730-0301. DOI : 10.1145/2980179.2982408.

[Saa03] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial et Applied Mathematics, jan. 2003. ISBN : 9780898715347, 9780898718003. DOI : 10.1137/1.9780898718003.

[Sah+08] Onkar Sahni, Kenneth E. Jansen, Mark S. Shephard, Charles A. Taylor et Mark W. Beall. « Adaptive boundary layer meshing for viscous flow simulations ». In : *Engineering with Computers* 24.3 (avr. 2008), p. 267-285. ISSN : 0177-0667, 1435-5663. DOI : 10.1007/s00366-008-0095-0.

[Sah+16] Onkar Sahni, Aleksandr Ovcharenko, Kedar C. Chitale, Kenneth E. Jansen et Mark S. Shephard. « Parallel anisotropic mesh adaptation with boundary layers for automated viscous flow simulations ». In : *Engineering with Computers* 33.4 (avr. 2016), p. 767-795. ISSN : 0177-0667, 1435-5663. DOI : 10.1007/s00366-016-0437-2.

[Sch96] R. Schneiders. « A grid-based algorithm for the generation of hexahedral element meshes ». In : *Engineering with Computers* 12.3-4 (sept. 1996), p. 168-177. ISSN : 0177-0667, 1435-5663. DOI : 10.1007/bf01198732.

[She+94] Jonathan Richard Shewchuk et al. *An introduction to the conjugate gradient method without the agonizing pain*. 1994.

[She+99] A. Sheffer, M. Etzion, A. Rappoport et M. Bercovier. « Hexahedral Mesh Generation using the Embedded Voronoi Graph ». In : *Engineering with Computers* 15.3 (sept. 1999), p. 248-262. ISSN : 0177-0667, 1435-5663. DOI : 10.1007/s003660050020.

[She02] Jonathan Shewchuk. « What is a good linear finite element ? interpolation, conditioning, anisotropy, and quality measures (preprint) ». In : *University of California at Berkeley* 73 (2002).

[SHM14] Ruben Sevilla, Oubay Hassan et Kenneth Morgan. « The use of hybrid meshes to improve the efficiency of a discontinuous Galerkin method for the solution of Maxwell's equations ». In : *Computers & Structures* 137 (juin 2014), p. 2-13. ISSN : 0045-7949. DOI : 10.1016/j.compstruc.2013.01.014.

[Sho+04] Nissan Shoykhet, Elena S. Di Martino, David A. Vorp et Kenji Shimada. « Assessment of Hex-Dominant Mesh Efficacy for Nonlinear Finite Element Method Structural Analyses ». In : *Advances in Bioengineering*. ASME, 2004. ISBN : 0791847039. DOI : 10.1115/imece2004-61773.

[Si15a] Hang Si. « TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator ». In : *ACM Trans. Math. Softw.* 41.2 (fév. 2015), p. 1-36. ISSN : 0098-3500. DOI : 10.1145/2629697.

[Si15b] Hang Si. « TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator ». In : *ACM Trans. Math. Softw.* 41.2 (fév. 2015), p. 1-36. ISSN : 0098-3500. DOI : 10.1145/2629697.

[SJ07] Jason F Shepherd et Chris R Johnson. *Topologic and geometric constraint-based hexahedral mesh generation*. T. 68. 03. 2007.

[SK00] Kambiz Salari et Patrick Knupp. *Code Verification by the Method of Manufactured Solutions*. Rapp. tech. Office of Scientific et Technical Information (OSTI), juin 2000. DOI : 10.2172/759450.

[SLS04] Y. Su, K.H. Lee et A. Senthil Kumar. « Automatic hexahedral mesh generation for multi-domain composite models using a hybrid projective grid-based method ». In : *Computer-Aided Design* 36.3 (mar. 2004), p. 203-215. ISSN : 0010-4485. DOI : 10.1016/s0010-4485(03)00079-4.

[Sok+16] Dmitry Sokolov, Nicolas Ray, Lionel Untereiner et Bruno Lévy. « Hexahedral-Dominant Meshing ». In : *Transactions on Graphics (TOG)* 35.5 (juin 2016), p. 1-23. ISSN : 0730-0301. DOI : 10.1145/2930662.

[SRR14] J. Sarrate, E. Ruiz-Gironés et X. Roca. « Unstructured and Semi-Structured Hexahedral Mesh Generation Methods ». In : *Comp. Tech. Rev.* 10 (sept. 2014), p. 35-64. ISSN : 2044-8430. DOI : 10.4203/ctr.10.2.

[SS86] Youcef Saad et Martin H. Schultz. « GMRES : A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems ». In : *SIAM J. Sci. and Stat. Comput.* 7.3 (juil. 1986), p. 856-869. ISSN : 0196-5204, 2168-3417. DOI : 10.1137/0907058.

[SSW96] Robert Schneiders, Roland Schindler et Frank Weiler. « Octree-based generation of hexahedral element meshes ». In : *IN PROCEEDINGS OF THE 5TH INTERNATIONAL MESHING ROUNDTABLE*. Citeseer. 1996.

[Sta+09] Matthew L. Staten, Robert A. Kerr, Steven J. Owen, Ted D. Blacker, Marco Stupazzini et Kenji Shimada. « Unconstrained plastering-Hexahedral mesh generation via advancing-front geometry decomposition ». In : *Int. J. Numer. Meth. Engng* (2009), n/a-n/a. ISSN : 0029-5981, 1097-0207. DOI : 10.1002/nme.2679.

[Stu64] John Stuelpnagel. « On the Parametrization of the Three-Dimensional Rotation Group ». In : *SIAM Rev.* 6.4 (oct. 1964), p. 422-430. ISSN : 0036-1445, 1095-7200. DOI : 10.1137/1006093.

[SVB17] Justin Solomon, Amir Vaxman et David Bommes. « Boundary Element Octahedral Fields in Volumes ». In : *Transactions on Graphics (TOG)* 36.3 (mai 2017), p. 1-16. ISSN : 0730-0301. DOI : 10.1145/3065254.

[SWP13] S. R. Slattery, P. P. H. Wilson et R. P Pawlowski. « The Data Transfer Kit : A geometric rendezvous-based tool for multiphysics data transfer ». In : *Proceedings of the 2013 International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. 2013.

[TAB99] Stefan Turek, Mike Altieri et Christian Becker. *Proposal for Sparse Banded Blas techniques*. IWR, 1999.

[Tar+04] Marco Tarini, Kai Hormann, Paolo Cignoni et Claudio Montani. « PolyCube-Maps ». In : *ACM SIGGRAPH 2004 Papers on - SIGGRAPH '04*. T. 23. 3. ACM. ACM Press, 2004, p. 853-860. doi : 10.1145/1186562.1015810.

[TBK06] Stefan Turek, Chr Becker et Susanne Kilian. « Hardware-oriented numerics and concepts for PDE software ». In : *Future Generation Computer Systems* 22.1 (2006), p. 217-238.

[TBM96] Timothy J Tautges, Ted Blacker et Scott A Mitchell. « The whisker weaving algorithm : A connectivity-based method for constructing all-hexahedral finite element meshes ». In : *International Journal for Numerical Methods in Engineering* 39.19 (1996), p. 3327-3349.

[TEC11] Srinivas C. Tadepalli, Ahmet Erdemir et Peter R. Cavanagh. « Comparison of hexahedral and tetrahedral elements in finite element analysis of the foot and footwear ». In : *Journal of Biomechanics* 44.12 (août 2011), p. 2337-2343. issn : 0021-9290. doi : 10.1016/j.jbiomech.2011.05.006.

[TPG99] G.M. Treece, R.W. Prager et A.H. Gee. « Regularised marching tetrahedra : Improved iso-surface extraction ». In : *Computers & Graphics* 23.4 (août 1999), p. 583-598. issn : 0097-8493. doi : 10.1016/s0097-8493(99)00076-x.

[Tur+10] Stefan Turek, Dominik Göddeke, S Buijssen et Hilmar Wobker. « Hardware-oriented multigrid finite element solvers on GPU-accelerated clusters ». In : *Scientific Computing with Multicore and Accelerators* (2010), p. 113-130.

[Tur+56] MJ Turner, RJ Clough, HC Martin et LJ Topp. « Stiffness and Deflection Analysis of Complex Structures ». In : *Journal of the Aeronautical Sciences* 23.9 (sept. 1956), p. 805-823. issn : 1936-9956. doi : 10.2514/8.3664.

[ÜFE10] Markus Üffinger, Steffen Frey et Thomas Ertl. « Interactive High-Quality Visualization of Higher-Order Finite Elements ». In : *Computer Graphics Forum*. T. 29. 2. Wiley Online Library. 2010, p. 337-346. doi : 10.1111/j.1467-8659.2009.01603.x.

[Vax+16] Amir Vaxman, Marcel Campen, Olga Diamanti, David Bommes, Klaus Hildebrandt, Mirela Ben-Chen et Daniele Panozzo. « Directional field synthesis, design, and processing ». In : *SIGGRAPH ASIA 2016 Courses on - SA '16*. T. 35. 2. Wiley Online Library. ACM Press, 2016, p. 545-572. isbn : 9781450345385. doi : 10.1145/2988458.2988478.

[VSL16] Ryan Viertel, Matthew L Staten et Franck Ledoux. *Analysis of Non-Meshable Automatically Generated Frame fields.* Rapp. tech. Sandia National Laboratories (SNL-NM), Albuquerque, NM (United States), 2016.

[Whi+95] David R White, Lai Mingwu, Steven E Benzley et Gregory D Sjaardema. « Automated hexahedral mesh generation by virtual decomposition ». In : *Proceedings of the 4th International Meshing Roundtable, Sandia National Laboratories, Albuquerque, USA*. 1995, p. 165-176.

[Wie97] Christian Wieners. « Conforming discretizations on tetrahedrons, pyramids, prisms and hexahedrons ». In : *Preprint, University of Stuttgart* (1997).

[WK97] David R White et Paul Kinney. « Redesign of the paving algorithm : Robustness enhancements through element by element meshing ». In : *6th International Meshing Roundtable*. T. 10. 1997, p. 830.

[WNR04] Erke Wang, Thomas Nelson et Rainer Rauch. « Back to elements-tetrahedra vs. hexahedra ». In : 2004.

[WO01] P. Wesseling et C.W. Oosterlee. « Geometric multigrid with applications to computational fluid dynamics ». In : *Journal of Computational and Applied Mathematics* 128.1-2 (mar. 2001), p. 311-334. issn : 0377-0427. doi : 10.1016/s0377-0427(00)00517-3.

[WR16] Johan Winges et Thomas Rylander. « Higher-order brick-tetrahedron hybrid method for Maxwell's equations in time domain ». In : *Journal of Computational Physics* 321 (sept. 2016), p. 698-707. issn : 0021-9991. doi : 10.1016/j.jcp.2016.05.063.

[YS03] Soji Yamakawa et Kenji Shimada. « Fully-automated hex-dominant mesh generation with directionality control via packing rectangular solid cells ». In : *Int. J. Numer. Meth. Engng.* 57.15 (2003), p. 2099-2129. issn : 0029-5981, 1097-0207. doi : 10.1002/nme.754.

[ZB06] Yongjie Zhang et Chandrajit Bajaj. « Adaptive and quality quadrilateral/hexahedral meshing from volumetric data ». In : *Computer Methods in Applied Mechanics and Engineering* 195.9-12 (fév. 2006), p. 942-960. issn : 0045-7825. doi : 10.1016/j.cma.2005.02.016.

[ZC15] Suqin Zeng et Elaine Cohen. « Hybrid volume completion with higher-order Bézier elements ». In : *Computer Aided Geometric Design* 35-36 (mai 2015), p. 180-191. issn : 0167-8396. doi : 10.1016/j.cagd.2015.03.008.

[ZCY16] Lei Zhao, Geng Chen et Wenhua Yu. « An Efficient Algorithm for SAR Evaluation from Anatomically Realistic Human Head Model Using DGTD with Hybrid Meshes. » In : *Applied Computational Electromagnetics Society Journal* 31.6 (2016).

[Zga+96] F.-X. Zgainski, J.-L. Coulomb, Y. Marechal, F. Claeyssen et X. Brunotte. « A new family of finite elements : The pyramidal elements ». In : *IEEE Trans. Magn.* 32.3 (mai 1996), p. 1393-1396. issn : 0018-9464. doi : 10.1109/20.497507.

[Zie+13] Olgierd Cecil Zienkiewicz, Robert Leroy Taylor, Robert Leroy Taylor et JZ Zhu. *The Finite Element Method : Its Basis and Fundamentals*. Elsevier, 2013. ISBN : 9781856176330. DOI : 10.1016/c2009-0-24909-9.

Résumé

Cette thèse s'intéresse à la génération, à l'utilisation et à l'évaluation des maillages hex-dominants, composés d'hexaèdres et de tétraèdres, dans la cadre de la simulation numérique par la méthode des éléments finis. Les éléments finis hexaédriques sont souvent préférés aux éléments tétraédriques car ils offrent un meilleur ratio entre précision et temps de calcul dans un certain nombre de situations. Cependant, si la génération automatique de maillages tétraédriques est aujourd'hui un domaine bien maîtrisé, ce n'est pas le cas de la génération de maillages hexaédriques alignés avec le bord, qui reste un problème largement ouvert. En l'absence de progrès significatifs, les approches actuelles se contentent de maillages hex-dominants afin de tirer parti des performances supérieures des hexaèdres et de la flexibilité géométrique des tétraèdres, qui rend possible le maillage automatique.

Dans une première partie, nous développons des algorithmes robustes pour la génération de maillages hex-dominants à partir de champs de directions, notamment pour l'isolement et le remplissage des régions difficiles à mailler (singularités et autres dégénérescences). Dans la seconde partie, nous essayons de déterminer dans quelles situations et dans quelle mesure les maillages hexaédriques, et hex-dominants générés précédemment, sont plus intéressants que les maillages tétraédriques. Ceci implique spécifiquement d'étudier plusieurs manières d'effectuer des simulations par éléments finis avec les maillages hybrides, dont une approche où nous utilisons des contraintes de continuité pour maillages non-conformes. Pour mesurer l'influence du maillage sur l'approximation des solutions, nous proposons une nouvelle méthode d'échantillonnage pour calculer très efficacement des distances globales entre solutions éléments finis définies sur des domaines compliqués.

Mots-clés: simulation numérique, maillage, maillage hex-dominant, méthode des éléments finis, comparaison de maillages

Abstract

This thesis focuses on generation, usage and evaluation of hex-dominant meshes, which are made of hexaehedra and tetrahedra, in the context of the finite element method. Hexahedron finite elements are often preferred to tetrahedron elements because they offer a better compromise between accuracy and computation time in certain situations. However, if tetrahedral meshing is a well mastered subject, it is not the case of hexahedral meshing. Generating hexahedral meshes with elements aligned to the borders is still an open and difficult problem. Meanwhile, current automated approaches can use hex-dominant meshes in order to take advantage of both hexahedron accuracy and geometrical flexibility of tetrahedra.

In the first part, we develop robust algorithms for the generation of hex-dominant meshes with elements aligned with the borders. Specifically, we propose a method to extract and fill the areas where hexahedral meshing is difficult (singularities and degeneracies). In the second part, we try to identify and to quantify the advantages of hexahedral and hex-dominant meshes over tetrehedral ones. This requires to study various ways to apply the finite element method on hybrid meshes, including one in which we propose to use continuity constraints on hexahedral-tetrahedral non-conforming meshes. To measure the impact of meshes on the finite element accuracy, we develop a new sampling method which allows to compute efficiently global distances between finite element solutions defined on complicated 3D domains.

Keywords: numerical simulation, meshing, hex-dominant meshing, finite element method, mesh comparison