



HAL
open science

Génération interactive de maillages hexaédriques structurés par blocs

Simon Calderan

► **To cite this version:**

Simon Calderan. Génération interactive de maillages hexaédriques structurés par blocs. Interface homme-machine [cs.HC]. Université Paris-Saclay, 2022. Français. NNT : 2022UPASG049 . tel-03859633

HAL Id: tel-03859633

<https://theses.hal.science/tel-03859633v1>

Submitted on 18 Nov 2022

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

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

Génération interactive de maillages
hexaédriques structurés par blocs
*Interactive generation of block-structured hexahedral
meshes*

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580, sciences et technologies de l'information et de la
communication (STIC)
Spécialité de doctorat : Informatique
Graduate School : Informatique et sciences du numérique, Référent :
Université d'Evry

Thèse préparée dans l'unité de recherche IBISC (Université Paris-Saclay, Univ
Evry), sous la direction de Guillaume HUTZLER, Maître de conférences et le
co-encadrement de Franck LEDOUX, Directeur de recherche

Thèse soutenue à Paris-Saclay, le 17 Mai 2022, par

Simon CALDERAN

Composition du jury

Jean-Christophe Janodet Professeur des universités, Université Évry-Val- d'Essonne	Président
Dominique Bechmann Professeure des universités, Université de Stras- bourg	Rapporteur
Jeanne Pellerin Cadre scientifique, TotalEnergies	Rapporteur
Agnès Arnould Maîtresse de conférences, Université de Poitiers	Examinatrice
Guillaume Hutzler Maître de conférences, Université Évry-Val- d'Essonne	Directeur de thèse

Titre : Génération interactive de maillages hexaédriques structurés par blocs.

Mots clés : Interaction Homme-Machine, Maillages hexahédriques, Simulation numérique.

Résumé : Les codes de simulation numérique reposant sur des méthodes de type éléments et volumes finis requièrent de discrétiser le domaine étudié – par exemple une pièce mécanique telle qu'un moteur, une aile d'avion, une turbine, etc. – à l'aide d'un maillage. En dimension 3, un maillage est un ensemble composé d'éléments volumiques simples, le plus souvent des tétraèdres ou des hexaèdres, qui partitionnent le domaine d'étude. Le choix de tétraèdres ou d'hexaèdres est principalement dicté par l'application (interaction fluide-structure, hydrodynamique, etc.). Si la génération automatique de maillages tétraédriques est un processus relativement maîtrisé aujourd'hui, générer des maillages hexaédriques est toujours un problème ouvert. Ceci est problématique pour les applications qui justement nécessitent impérativement des maillages hexaédriques puisque leur génération se fait de façon semi-automatique, ce qui peut prendre plusieurs semaines à plusieurs mois de temps ingénieur ! Alors que le temps consacré au processus de simulation numérique à proprement parler tend à diminuer du fait de la puissance des machines utilisées, le goulot d'étranglement est désormais dans la préparation des données, à savoir obtenir un modèle de CAO adapté au calcul, puis en générer un maillage. C'est dans ce contexte que s'inscrit la thèse proposée en suivant une approche hybride mêlant : 1. Le développement d'algorithmes (semi)-automatiques pour générer et modifier des maillages hexaédriques structurés par blocs ; 2. La mise en place d'un logiciel graphique interactif dédié à la manipulation de structures de blocs. Les

mécanismes d'interaction seront en outre utilisés pour guider les algorithmes dans leurs prises de décision, que ce soit à l'initialisation (critères à apposer sur des entités particulières de CAO) ou en cours d'algorithme (décision entre plusieurs options sur lesquelles l'algorithme ne peut se prononcer seul).

L'objectif de cette thèse n'est donc pas de fournir une solution automatique universelle, ce qui semble inatteignable actuellement, mais plutôt de réduire le temps ingénieur consacré à la génération du maillage en fournissant des outils plus adaptés. Dans cette optique, nous proposons de placer l'étude dans le prolongement de [LED10, KOW12, GAO15, GAO17], où est considéré le problème de simplification et d'enrichissement de maillages hexaédriques par insertion et suppression de couches de mailles. Dans tous ces travaux, les algorithmes proposés sont des algorithmes simples de type « glouton » où le maillage est modifié pas à pas pour converger vers une solution finale E_f : A chaque étape E_i , on fait l'hypothèse que la « meilleure » solution E_f sera obtenue en faisant le choix « optimal » pour E_i . Or en recherche opérationnelle, une telle approche est connue comme perfectible dès lors que le problème d'optimisation traité est non linéaire. L'idée est donc d'utiliser des approches usuelles en recherche opérationnelle et plus spécifiquement des systèmes multi-agents, couplées à des outils interactifs, pour permettre la génération de structures de blocs sur des CAO complexes.

Title : Interactive generation of block-structured hexahedral meshes.

Keywords : Computer-Human Interaction, Hexahedral meshes, Numeric simulation.

Abstract : The numerical simulation codes based on element and finite volume methods require discretizing the studied domain - for example a mechanical part such as a motor, an airplane wing, a turbine, etc. - using a mesh. In 3 dimensions, a mesh is a set composed of simple volumic elements, most often tetrahedrons or hexahedra, which partition the field of study. The choice of tetrahedrons or hexahedra is mainly dictated by the application (fluid-structure interaction, hydrodynamics, etc.). If the automatic generation of tetrahedral meshes is a relatively controlled process today, generating hexahedral meshes is still an open problem. This is problematic for applications that just imperatively require hexahedral meshes since their generation is done semi-automatically, which can take several weeks to several months of engineer time! While the time devoted to the digital simulation process itself tends to decrease due to the power of the machines used, the bottleneck is now in the preparation of the data, namely to obtain a CAD model adapted to the computation, then generate a mesh. It is in this context that we propose to develop a hybrid approach combining : 1. The development of (semi) -automatic algorithms for generating and modifying block-structured hexahedral meshes ; 2. The implementation of an interactive graphic software dedicated to the manipulation of block structures. The interaction me-

chanisms will also be used to guide the algorithms in their decision-making, whether at initialization (criteria to be affixed to particular CAD entities) or in the course of an algorithm (decision between several options on which the algorithm cannot pronounce itself).

The objective of this thesis is therefore not to provide a universal automatic solution, which seems unattainable at present, but rather to reduce the engineering time devoted to the generation of the mesh by providing more adapted tools. In this context, we propose to place the study as an extension of [LED10, KOW12, GAO15, GAO17], which considered the problem of simplification and enrichment of hexahedral meshes by insertion and removal of mesh layers. In all these works, the proposed algorithms are simple 'greedy' algorithms where the mesh is modified step by step to converge towards a final solution E_f : At each step E_i , one makes the hypothesis that the 'best' solution E_f will be obtained by making the 'optimal' choice for E_i . However, in operational research, such an approach is known to be perfectible since the problem of optimization treated is nonlinear. The idea is to use usual approaches in operational research and more specifically multi-agent systems, coupled to interactive tools, to allow the generation of block structures on complex CAOs.

Remerciements

Je tiens à remercier en premier lieu mes directeurs de thèse qui se sont engagés avec moi et qui m'ont aidés et soutenu pendant ces trois ans et demi. Merci à Guillaume d'avoir accepté de diriger cette thèse dont le sujet était éloigné de son domaine et d'avoir pourtant pris le temps de travailler et de se plonger dans ces problématiques avec nous. Et évidemment merci à Franck pour m'avoir d'abord pris en stage puis proposé de poursuivre en thèse sur un sujet qui m'intéressait et pour ces longues discussions passées à réfléchir aux solutions des problèmes que je rencontrais ou que je créais tout seul. Encore merci à eux deux parce que me connaissant j'admets volontiers que ça n'a pas du être évident tous les jours, surtout sur la fin, donc un respect à eux. Je souhaite également remercier les membres du jury qui ont accepté d'évaluer mes travaux. Merci à Dominique Bechmann et Jeanne Pellerin d'avoir rapporté ce manuscrit ainsi que Jean-Christophe Janodet et Agnès Arnould pour leur participation au jury de soutenance. Merci pour votre bienveillance ainsi que vos remarques pertinentes et constructives. Je voudrais maintenant remercier ceux que j'ai côtoyé tout au long de ma thèse. D'abord et avant tout Florian et Anton, ceux qui comme moi n'ont pas fait "une thèse en science mais en informatique". Sans eux mes années de thèse n'auraient pas été aussi cool et beaucoup de bugs n'auraient encore à ce jour pas été résolus. De la même façon merci à Eloïse et Alexiane avec qui j'ai également passé beaucoup de temps et qui rejoignent les deux premiers mais j'étais obligé de les mettre à part, je suis sûr que vous me comprendrez. Merci aussi à tout ceux de Teratec, doctorants et stagiaires avec qui j'ai passé de bons moments en plus de pouvoir échanger avec des personnes super fortes qui m'ont appris plein de choses et c'était toujours super intéressant...bon des fois moins que d'autres. Je n'oublie pas non plus François, mon camarade de maillage qui à la faculté à venir, donner une anecdote technique ou un problème de maillage, dire "C'est pas fou ça ?" et repartir. Aussi un grand merci à Nicolas pour sa bienveillance, son aide technique et pour toujours m'expliquer les choses, ça m'a permis d'apprendre énormément. Et bien évidemment je tiens à remercier Murielle qui m'a sauvé plusieurs fois pour l'administratif de la fac et qui, en toute honnêteté, à sûrement sauvé ma thèse. Enfin je remercie Hélène pour avoir la gentillesse de s'occuper de nous à Teratec même si on est des grands garçons et Brigitte pour toujours apporter un brin de bonne humeur dans la journée. Et pour finir je remercie ma famille et mes parents qui m'ont soutenu et aidé pendant toutes ces années. Un grand merci à vous tous.

Table des matières

1	Introduction	9
1.1	Maillage en dimension 2 et 3	9
1.2	De la difficulté du maillage par blocs	12
1.3	Blocs, champs d'orientation et interactivité	14
2	Notions et concepts	19
2.1	Notions et définitions pour le maillage hexaédrique	19
2.1.1	Notion de maillages et de cellules	19
2.1.2	Qualité géométrique d'un maillage hexaédrique	27
2.1.3	Classification d'un maillage sur un domaine géométrique	29
2.2	Comment générer un maillage hexaédrique ?	31
2.2.1	Approches par objet médian	35
2.2.2	Approches par polycubes	35
2.2.3	Approches par champs d'orientation	37
2.2.4	Approches interactives	38
2.3	Positionnement de nos travaux	40
3	Génération interactive de blocs par une approche duale	43
3.1	Champs d'orientation pour le maillage hexaédrique	46
3.1.1	Définition et représentation des champs d'orientations	46
3.1.2	Résolution du problème d'optimisation par relaxation Lagrangienne	49
3.1.3	Structure d'un champ d'orientations 3D	49
3.2	Construction d'une structure duale	52
3.2.1	Construction d'un feuillet dual par propagation	52
3.2.2	Filtres de contrôle	57
3.3	Validité de la structure duale	61
3.3.1	Définition des règles de validité	62
3.3.2	Algorithme de vérification des règles	66
3.4	Création de la structure primale de blocs	70
3.4.1	Extraction de la topologie	70
3.4.2	Ordonnement des sommets de blocs	74
4	Analyse et correction du champ d'orientations	81
4.1	Présentation du problème	81
4.2	Analyse d'un champ d'orientations	88
4.2.1	Détection des lignes singulières 3-5	88
4.2.2	Détection des zones de tremplin	89
4.3	Découpage du domaine géométrique	90
4.3.1	Partitionnement dirigé par le champ d'orientation pour la coupe géométrique	91

4.3.2	Reconstruction d'un modèle géométrique virtuel	93
4.4	Relaxation de contraintes	97
4.4.1	Présentation et intérêt de l'approche	97
4.4.2	Méthode proposée	99
5	Post-processing et résultats de notre approche	107
5.1	Génération du maillage hexaédrique final	108
5.1.1	Classification géométrique	108
5.1.2	Calcul de la discrétisation des arêtes	110
5.1.3	Créations des points du maillage	112
5.2	Résultats	113
5.2.1	Utilisation de l'outil	114
5.2.2	Résultats sur des modèles de types V	116
5.2.3	Résultats sur des modèles de types S et T	118
6	Conclusion et perspectives	123
6.1	Perspectives	124
6.1.1	Automatisation de la génération de la structure duale	124
6.1.2	Coupe d'un modèle géométrique à l'aide de l'information de champ d'orientation	124
6.1.3	Automatisation de la correction de champs d'orientation	125
6.1.4	Optimisation topologique de la structure de blocs	125

1 - Introduction

La simulation numérique de phénomènes physiques assistée par ordinateur a pris une ampleur incroyable depuis le début des années 2000 pour devenir aujourd'hui un outil incontournable dans de nombreux domaines technologiques et industriels tels que : la simulation de déformation de voitures pour éviter la destruction de prototype réels, la mise en place de nouveaux designs en aviation, la simulation de tsunami ou de tremblements de terre, etc. Simuler numériquement à l'aide d'un ordinateur un phénomène physique nécessite de passer d'une modélisation continue du problème à une modélisation discrète. D'un point de vue physique, il est nécessaire d'approcher les équations régissant les lois physiques concernées - l'écoulement d'un fluide, la résistance d'une structure, le déplacement d'une onde. D'un point de vue géométrique, il faut aussi mener un travail d'approximation pour représenter de manière discrète les domaines géométriques concernés - l'océan, une voiture, un pont, un avion. Les sciences de l'analyse numérique au sens large traitent de cette modélisation discrète et de sa validité par rapport au monde physique réel.

Nous nous intéressons dans ce travail à la discrétisation spatiale du domaine étudié. De façon générale, on discrétise des domaines géométriques de dimension 1, ou 1D, de dimension 2, ou 2D, et de dimension 3, ou 3D. La manière avec laquelle nous discrétisons un domaine aura un impact sur la résolution numérique que nous aurons du problème physique étudié. A titre d'exemple, considérons la discrétisation 1D qui reste relativement simple à mettre en œuvre et que nous avons tous utilisée pour approcher numériquement des intégrales simples durant notre scolarité (voir figure 1.1). Considérons que nous voulons calculer de manière approchée le résultat de l'intégrale $\int_a^b f(x)dx$ sur l'intervalle $[a, b]$. Plutôt que de calculer analytiquement une solution, nous allons découper l'intervalle en plusieurs morceaux et approcher la fonction f sur chacun des intervalles par des fonctions plus simples à évaluer. Pour diviser $[a, b]$ en intervalles, choisissons un entier $n > 0$ et posons

$$\forall i \in \llbracket 0, n \rrbracket, x_i = a + i \frac{b-a}{n}.$$

On a ainsi une subdivision uniforme de notre intervalle où chaque valeur x_i est l'abscisse d'un point de cette subdivision. Cette méthode très élémentaire permet de calculer de manière approchée l'aire sous la courbe f . Sur un intervalle $[x_i, x_{i+1}]$, on peut appliquer la méthode des rectangles à gauche, respectivement à droite, en utilisant la valeur en $f(x_i)$, respectivement $f(x_{i+1})$ sur tout l'intervalle. On obtient les approximations illustrées sur les figures 1.1-b et 1.1-c pour la fonction f dont l'aire calculée par intégration est représentée sur la figure 1.1-a. On constate aisément que l'aire approchée diffère de l'aire exacte et que le choix des fonctions simples sur chaque intervalle a un impact. Le passage à la méthode des trapèzes où l'on tient compte des valeurs aux deux extrémités de l'intervalle permet une meilleure approximation (voir figure 1.1-d). Les fonctions simples utilisées ont un impact important sur l'exactitude de la réponse fournie, tout comme la subdivision choisie. Sur la figure 1.1-e, le choix d'un nombre plus faible d'intervalles implique une moins bonne approximation de f , tandis que sur la figure 1.1-f, le choix d'une subdivision non uniforme de $[a, b]$ permettra de mieux approcher f pour un nombre fixé d'intervalles (et donc un même nombre de calculs).

1.1 . Maillage en dimension 2 et 3

Pour un problème simple (le calcul d'une aire), on peut constater que les choix de discrétisation effectués (nombre d'intervalles, taille de chaque intervalle, fonction sur chaque intervalle) ont une influence forte sur le résultat obtenu. Si l'on considère des problèmes physiques complexes sur des domaines géométriques 3D réels (pièces mécaniques par exemple), ces difficultés vont être démultipliées. En ce qui nous concerne, nous nous concentrons dans ce travail sur la discrétisation spatiale d'un domaine géométrique 3D, que nous

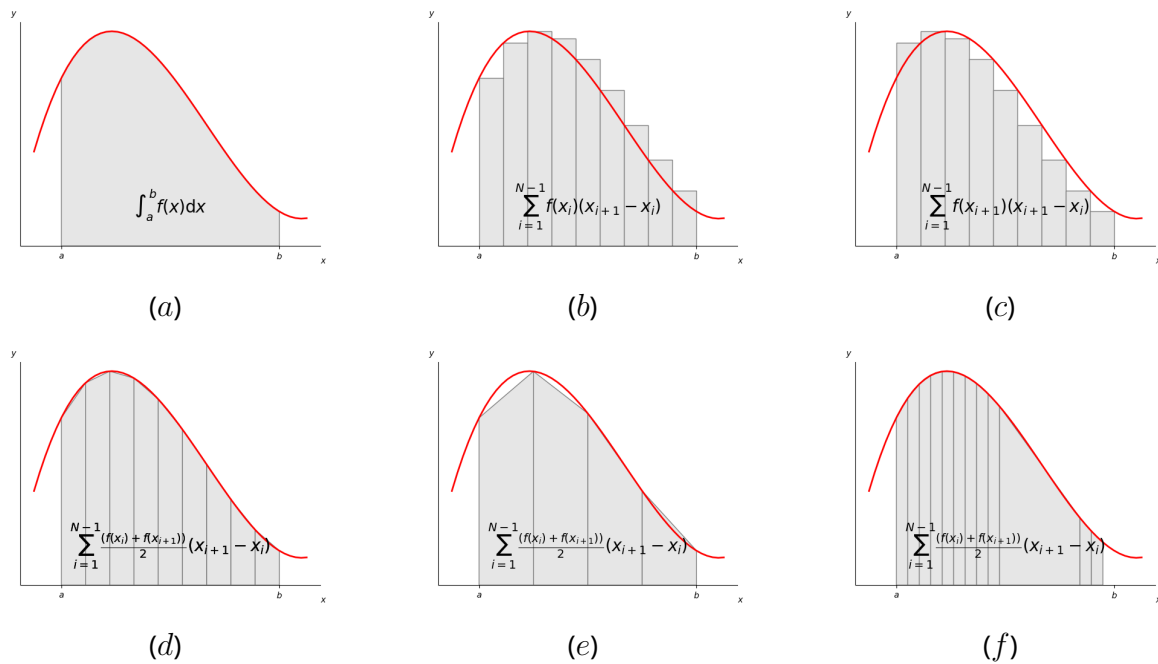


Figure 1.1 – Approximation du calcul de l'aire sous une courbe. En (a), l'aire est calculée analytiquement de manière exacte sous forme intégrale; de (b) à (d), l'aire est approchée en découpant l'intervalle $[a, b]$ en 9 intervalles $[x_i, x_{i+1}]$ sur lesquels on approche f par une fonction linéaire simple qui est $f(x_i)$ en (b), $f(x_{i+1})$ en (c) et $f(\frac{f(x_i) + f(x_{i+1})}{2})$ en (d). En (e), le nombre d'intervalles est diminué illustrant l'impact de la discrétisation sur le calcul de l'aire. En (f), une discrétisation non uniforme de l'intervalle $[a, b]$ permet de mieux approcher l'aire sous la courbe f .

noterons communément Ω . Comme pour notre intervalle 1D $[a, b]$, nous devons discrétiser Ω en un ensemble d'éléments simples. En 1D, l'intervalle $[a, b]$ était découpé en un ensemble de sous-intervalles, ou segments, qui *partitionnent* $[a, b]$. Par partitionner, on entend que l'union des sous-intervalles recouvre totalement $[a, b]$ et que pour tout couple (I_{-1}, I_{-2}) de sous-intervalles, I_{-1} et I_{-2} s'intersectent au plus en un point (le point x_i pour les intervalles $[x_{i-1}, x_i]$ et $[x_i, x_{i+1}]$ par exemple). En 3D, on va faire de même, à savoir partitionner Ω en un ensemble de "sous-intervalles", qui seront des cellules de dimension 3, les plus communes étant les tétraèdres et les hexaèdres. De manière générale, la discrétisation d'un domaine géométrique de dimension n par un ensemble d'éléments simples qui partitionne le domaine est appelé un **maillage**. La figure 1.2 montre des maillages 3D pour un même domaine discrétisé à l'aide d'hexaèdres (à gauche) et de tétraèdres (à droite). L'algorithme de maillage hexaédrique employé ici apporte une structuration dans le sens de la hauteur. Ce sera une propriété souvent recherchée par rapport aux maillages tétraédriques qui par nature sont moins structurés. Selon les méthodes numériques employées, un type de maillage sera privilégié par rapport à l'autre. Cependant une caractéristique "informatique" des maillages hexaédriques est que, pour une taille d'arête fixée, il faudra toujours environ 7 fois moins d'hexaèdres que de tétraèdres pour discrétiser un domaine Ω . Sur l'exemple de la figure 3, les arêtes de maillage étant de taille équivalentes, le maillage hexaédrique est composé de 1 155 hexaèdres et de 1 608 sommets quand le maillage tétraédrique contient 10 706 tétraèdres et 2 261 sommets. Ceci peut avoir une importance quand la simulation doit ensuite être exécutée sur des stations de travail "normales" et non pas sur des supercalculateurs dédiés au HPC. La taille des données est réduite pour un maillage hexaédrique et tous les calculs liés aux nombres de mailles s'exécutent plus rapidement.

En résumé, l'activité de maillage consiste à découper un domaine Ω en un ensemble fini d'éléments que sont les mailles. Mais contrairement au cas de la dimension 1, on rencontre de nombreuses difficultés et

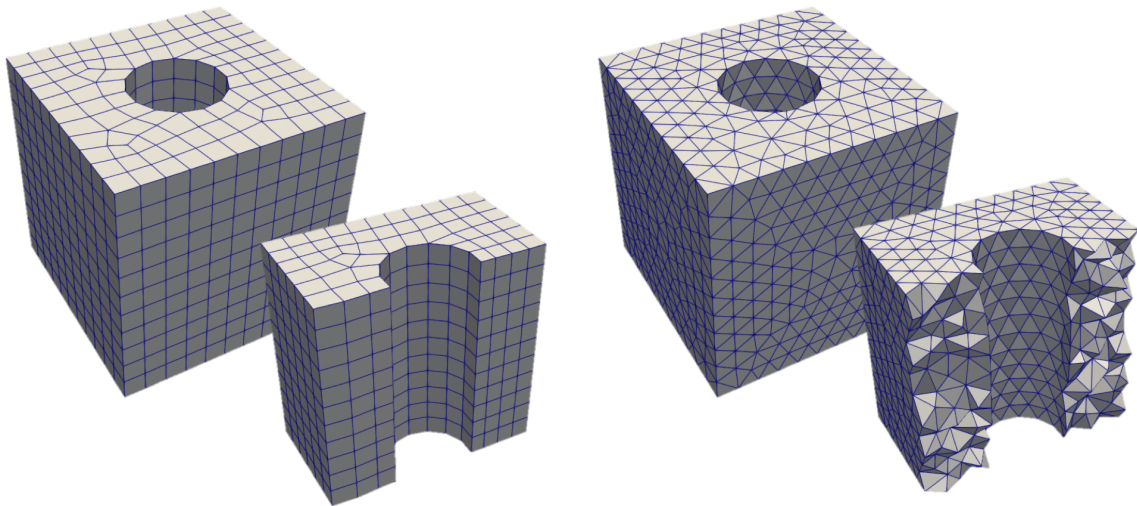


Figure 1.2 – Exemple d'un domaine géométrique discrétisé par un maillage hexaédrique à gauche et tétraédrique à droite. Pour chacun une vue pleine et une vue en coupe de l'objet sont proposées.

choix à effectuer et ce, même en se limitant aux éléments linéaires :

1. **Type de cellules** : comme vu précédemment, un domaine 3D peut être décomposé en éléments comme des tétraèdres ou des hexaèdres, mais aussi des pyramides, des prismes ou des polyèdres quelconques. Il en va de même en dimension 2 où le domaine peut être discrétisé par plusieurs types de polygones, des plus simples (les triangles) aux polygones quelconques en passant par les quadrangles. La question est alors de choisir quels éléments employer.
2. **Qualité géométrique des cellules** : une fois des éléments choisis, on voudra contrôler leur qualité. L'objectif de ces éléments étant de servir de support pour des calculs numériques, il faut éviter des aberrations numériques liées aux éléments eux-même. Par exemple, il est fréquent de calculer des aires ou des angles internes à un triangle, on préférera favoriser la création de triangles équilatéraux plutôt que de triangles très plats avec des angles aux sommets quasi nuls ou proches de la valeur π .
3. **Structuration** : selon le type de méthode numérique employée, il peut être préférable de disposer d'un maillage qui ressemble à une grille où soit en adéquation avec une propriété de la simulation (alignement sur une onde de propagation en acoustique ou sur l'interface entre deux matériaux en couplage fluide-structure). On privilégiera alors selon les cas des maillages hexaédriques structurés en un ensemble de grilles¹ [JFT99] ou l'utilisation de maillages tétraédriques anisotropes et adaptés [FG08].
4. **Contrôle de la taille** : comme en 1D, plus le maillage sera raffiné, c'est-à-dire composé de petites mailles, mieux on approchera numériquement le calcul à effectuer. Le problème est que cela peut induire un nombre de mailles bien trop important en 3D (des milliards de mailles dans certains cas). Il est alors intéressant de contrôler la taille des mailles pour placer de "petites" mailles dans les zones d'intérêt pour le calcul et des grandes mailles ailleurs.
5. **Discrétisation du bord** : le bord du domaine peut avoir une importance fondamentale selon les situations. On peut faire en sorte de disposer d'un maillage régulier et aligné avec le bord, ce qui est le cas de la figure 1.3-a, ou disposer d'un maillage irrégulier où les mailles de plus mauvaise qualité se retrouveront plutôt au bord (voir la figure 1.3-b).

1. On parlera alors de maillage hexaédrique structuré par blocs.

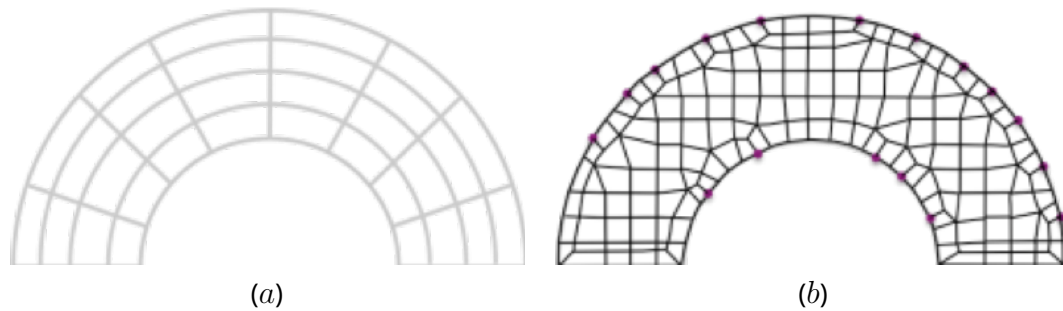


Figure 1.3 – En (a) le maillage obtenu est totalement structuré et des couches de mailles sont totalement alignées avec le bord; en (b), le maillage a été obtenu par intersection d'une grille qui a ensuite été modifiée en ajoutant une couche complète de mailles pour suivre le bord du domaine.

1.2 . De la difficulté pratique d'obtenir un maillage hexaédrique structuré par blocs

Comme on peut le voir, il existe de nombreuses façons de discrétiser un domaine 3D à l'aide d'un maillage. En pratique, le choix de la nature du maillage à utiliser dépend principalement des caractéristiques des codes de simulation qui utiliseront ce maillage comme discrétisation de Ω , des capacités de calcul disponibles (supercalculateur ou ordinateur individuel par exemple) et de la capacité à générer le maillage en question. En ce qui nous concerne, nous nous intéressons à la génération de maillages hexaédriques structurés par blocs, maillages qui sont utilisés pour des simulations de type hydrodynamique au CEA, mais aussi utilisés pour des problèmes de structure chez EDF par exemple (codes Aster [EDF20] et Saturne [EDF21a], Salome meca au CEA [EDF21b]). Un maillage hexaédrique structuré par blocs peut être vu comme un maillage à deux niveaux (voir figure 1.4) : le premier niveau, qualifié de **macro**, est composé de blocs hexaédriques grossiers (fig. 1.4-b) ; le second niveau, qualifié de **micro**, correspond au maillage final (fig. 1.4-c).

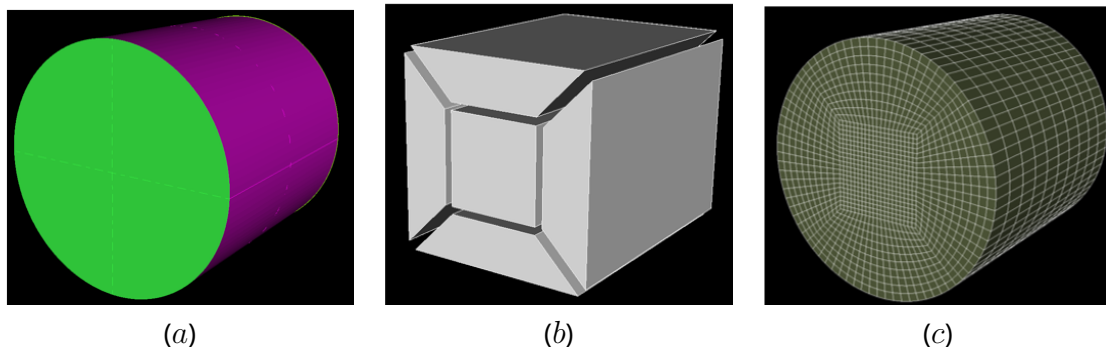


Figure 1.4 – Exemples d'entités dans une approche structurée par blocs. En (a), le domaine géométrique à discrétiser - ici un cylindre; en (b) une structure de blocs en O-grid; en (c), le maillage hexaédrique final où chaque bloc a été raffiné de manière uniforme par une grille.

La problématique principale de l'utilisation de maillages hexaédrique structurés par blocs est que leur génération n'est actuellement pas automatique ou alors pour des domaines simples et très spécifiques (obtenus par révolution ou extrusion de structures 2D). A titre d'exemple, considérons le processus illustré à la figure 1.5 où le logiciel **MGX** développé au CEA est utilisé pour générer la structure de blocs d'un objet 3D simple mais pour lequel aucune solution automatique ne donne aujourd'hui le résultat escompté. Partant du modèle illustré en 1.5-a, l'ingénieur va interactivement ajouter et modifier des blocs pour discrétiser l'objet. Le logiciel **MGX** fournira différentes aides à l'utilisateur mais ce dernier aura tout de même une succession importante d'étapes à réaliser. Sur les figures 1.5-b à 1.5-f, il va créer différents blocs en s'appuyant sur les courbes et surfaces du modèle initial, coller ces blocs entre eux et les découper pour bien "capturer" le

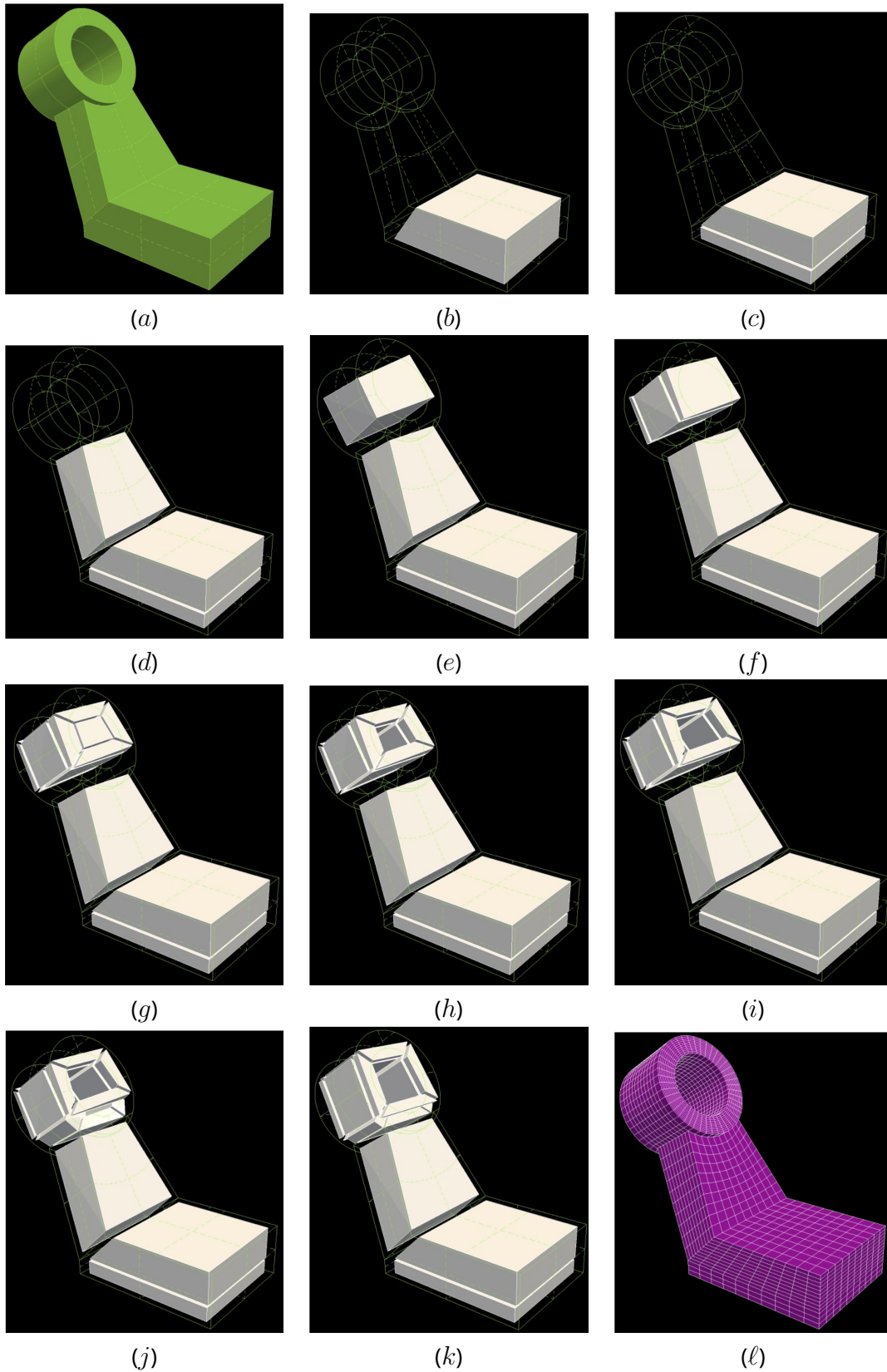


Figure 1.5 – Exemple de réalisation d'un maillage hexaédrique structuré par blocs à l'aide du logiciel **MGX** développé au CEA/DAM.

domaine géométrique. En 1.5-*g*, une structuration, dite en O-grid² est créée dans la partie supérieure du domaine pour ensuite supprimer des blocs en 1.5-*h* afin d'être en conformité avec le trou cylindrique du domaine géométrique. Une fois cette étape réalisée, il reste à recoller la structure de blocs correspondant au cylindre troué avec le reste des blocs (voir fig. 1.5-*j*) et à régulariser la position des sommets de blocs avant de générer le maillage final (voir fig. 1.5-*l*). Ce processus prend une quinzaine de minutes à un ingénieur expert ce qui est acceptable. Malheureusement, dès que la complexité de la pièce augmente, le temps nécessaire augmente énormément. La figure 1.6 illustre cela en considérant une étude menée en interne au CEA en 2018. Partant du domaine géométrique que nous avons discrétisé en blocs à la figure 1.5, le domaine est progressivement rendu plus complexe. Comme indiqué sur la figure, le nombre de blocs augmente de manière importante, tout comme le temps passé pour obtenir le maillage. En pratique, il est courant de devoir passer plusieurs jours à plusieurs semaines pour obtenir une structure de blocs adéquate et ce quelque soit le logiciel de maillage utilisé - **MGX**, Cubit[Lab], ICEM CFD-Hexa[ANS], etc.

GEOMETRIE						
TOPOLOGIE						
 Blocs 	29 blocs	59 blocs	62 blocs	92 blocs	132 blocs	174 blocs
Temps de création	15 mins	25 mins	30 mins	1 heure	1.5 heure	2 heures

Figure 1.6 – Temps consacré à la réalisation d'un maillage hexaédrique structuré par blocs avec le logiciel **MGX** pour une suite de modèles de complexité croissante.

1.3 . Structuration par blocs, champs d'orientation et interactivité

Si dans un contexte industriel, les ingénieurs utilisent des logiciels interactifs pour générer des maillages hexaédriques structurés par blocs, on peut se demander quelles sont les réponses académiques à ce problème. En fait, le constat est qu'aujourd'hui aucune approche ne permet de générer les structures de blocs souhaitées de façon automatique. Sans rentrer dans les détails (voir le chapitre 2 pour cela), on peut distinguer six familles d'approches pour générer des maillages hexaédriques : les approches par recombinaison, les méthodes frontales [SKOB06, SKO⁺10, TBM96, LW08], les approches basées sur l'intersection de grilles [Mar09, ZB04], les approches basées sur l'utilisation de l'objet médian [Qua14], les polycubes [THCM04, GSZ11] et les méthodes utilisant les champs d'orientation [RSL16, BRM⁺14, PBS20]. A notre connaissance, seules les approches par recombinaison et par intersection/immersion de grilles sont

2. On parle de O-grid, car les blocs existants sont entourés chacun de quatre blocs dans une direction donnée et ces quatre blocs forment un O autour du bloc initial.

capables de fournir automatiquement un maillage hexaédrique pour n'importe quel domaine Ω . Par contre, le maillage obtenu n'est pas du tout structuré par blocs et ne satisfait donc pas nos attentes. Parmi les autres familles de méthodes, celles utilisant un champ d'orientation sont les seules à produire directement la structure de blocs souhaitée sur des géométries simples (voir figure 1.7 [KLF16]). Une orientation 3D³ est un trièdre formé de trois directions 2 à 2 orthogonales. En définissant un champ d'orientation "lisse" sur l'ensemble du domaine Ω à discrétiser, l'idée est de s'appuyer sur ces orientations pour construire des hexaèdres.

Le caractère *lisse* du champ doit permettre d'extraire un maillage hexaédrique le plus structuré possible. En pratique, on peut évaluer le caractère lisse d'un champ d'orientation \mathcal{F} en un point $\mathbf{p} \in \Omega$ en calculant les rotations qui permettent de transporter l'orientation $\mathcal{F}_{\mathbf{p}}$ sur les orientations définies en tout point d'un voisinage de \mathbf{p} . Si les rotations sont très faibles, on est dans le cas idéal où l'on peut extraire une structure de grille hexaédrique. Dans le cas contraire, on n'a pas de structure du tout. A titre d'exemple, c'est ce que l'on peut voir sur les figures 1.7-*a* et *b*. Le champ est ici défini de manière discrète sur un maillage tétraédrique, les points considérés sont les sommets des tétraèdres et le voisinage d'un point \mathbf{p} est défini comme l'ensemble des sommets de la boule de \mathbf{p} , c'est-à-dire, tous les sommets partageant une arête avec le sommet \mathbf{p} . Sur la figure 1.7-*a*, les zones bleues foncées correspondent à des tétraèdres pour lesquels les rotations définies aux sommets sont quasiment identiques - rotation très faible. Au contraire, ceux en orange ont des orientations définies aux sommets fortement différentes. Aucune structure de grille ne peut en pratique être extraite de ces sommets. On dira de ces tétraèdres qu'ils sont **singuliers**. Pour l'exemple en question, tous les tétraèdres singuliers sont représentés en vert sur la figure 1.7-*b*. On en extrait la structure de blocs représentée de manière filaire sur la figure 1.7-*c* à l'aide de [KLF16]. On notera que les arêtes internes de la structure de blocs correspondent aux tétraèdres singuliers.

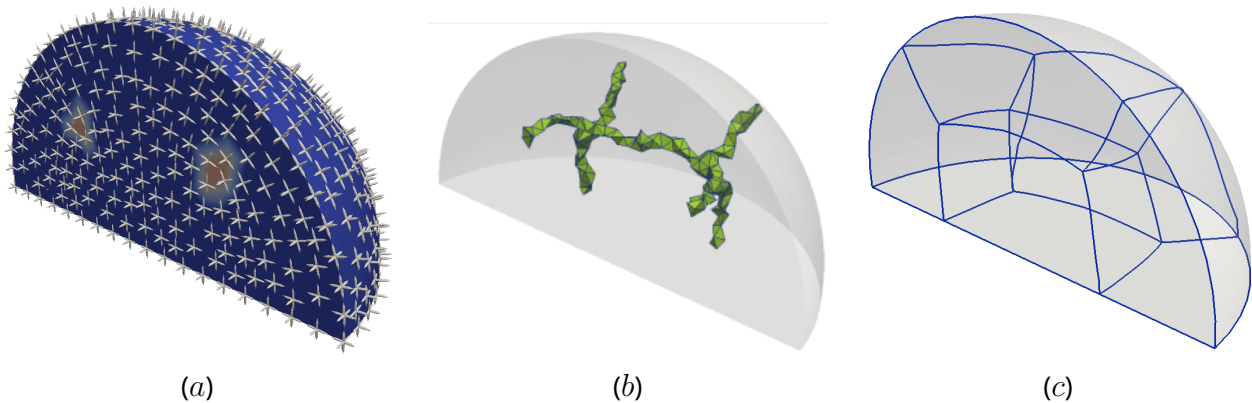


Figure 1.7 – Génération d'une structure de blocs à l'aide d'un champ d'orientation [KLF16]. Un champ d'orientation \mathcal{F} est calculé sur les sommets d'un maillage tétraédrique (a). D'un point de vue régularité, le champ dispose de singularités inscrites dans les tétraèdres verts (b). Ces derniers correspondent aux arêtes internes de la structure de blocs que l'on pourra extraire (c).

Le problème rencontré actuellement avec ces approches tient à plusieurs facteurs. Jusqu'à présent, les méthodes proposées génèrent un champ lisse dont la structure topologique, qui dans notre cas traverse les tétraèdres singuliers, n'est pas toujours celle d'un maillage hexaédrique structuré par blocs. Ce problème de fond se manifeste de deux façons :

1. Des lignes singulières, qualifiées 3 – 5 car connectant un sommet de valence 3 à un sommet de valence 5 sont fréquemment générées. Ces lignes ne sont pas compatibles avec une structure de

3. On parle de la notion de *frame* en anglais.

blocs hexaédriques. La ligne doit être entourée de 3 blocs à une extrémité et de 5 blocs, à l'autre extrémité. La transition de 3 vers 5 le long de la ligne est impossible sans l'ajout d'une ramification dans la structure de ligne singulière. Sur la figure 1.8, nous observons quatre lignes de ce type.

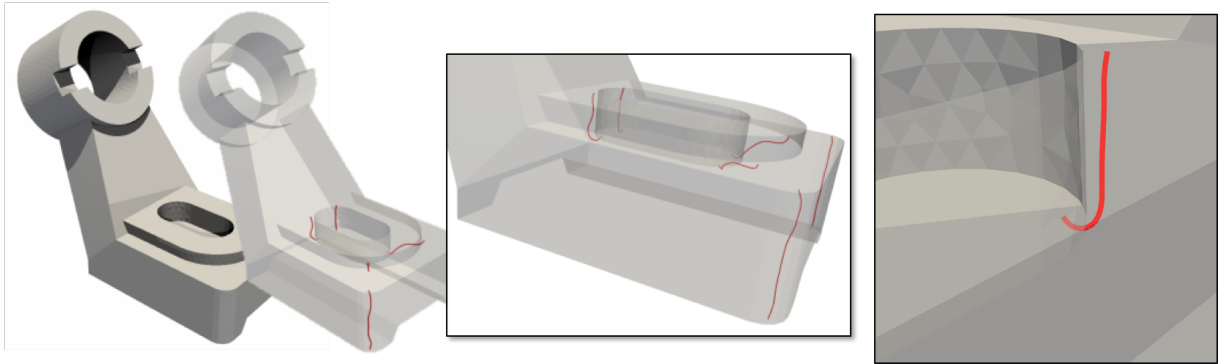


Figure 1.8 – Exemple d'un champs d'orientation incompatible avec la génération d'une structure de blocs. Les quatre lignes rouges autour de l'encoche cylindrique sur le domaine géométrique sont des lignes singulières 3 – 5.

2. On constate l'absence de lignes singulières dans des zones où l'on souhaiterait pourtant en avoir. On parle de *zones tremplins* et elles se manifestent par des portions de modèles qui ressemblent à un prisme à base trois. C'est par exemple le cas sur la figure 1.9 avec la partie prismatique posée sur un pavé. On peut voir sur les différentes incrustations que le champ, représenté par des croix est totalement régulier, aucune singularité n'y apparaît.

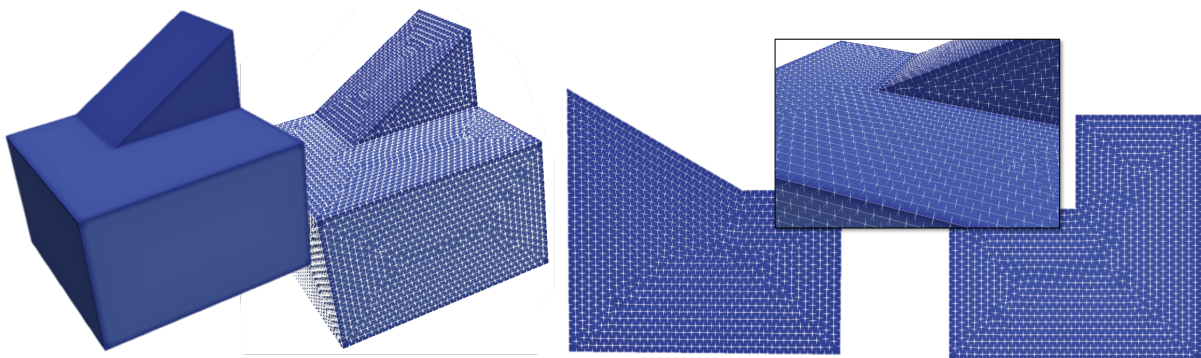


Figure 1.9 – Exemple de champs d'orientations où aucune singularité n'est détectée et qui pose problème pour l'extraction de blocs.

Une autre difficulté est d'ordre technique. La construction automatique d'une structure de blocs à partir du champ d'orientations est sujette à de nombreuses approximations numériques et l'implantation est difficile à rendre stable.

Enfin, une problématique plus générale au maillage pour la simulation numérique est que selon le phénomène physique à étudier et le code de simulation à utiliser, un domaine géométrique Ω peut être décomposé en blocs de plusieurs manières totalement différentes. Cette connaissance est actuellement détenue par le physicien qui réalise l'étude numérique. Proposer une solution totalement automatique dans ce contexte n'est donc pas une nécessité absolue. Par contre, réduire le temps de réalisation des maillages structurés par blocs serait une avancée importante.

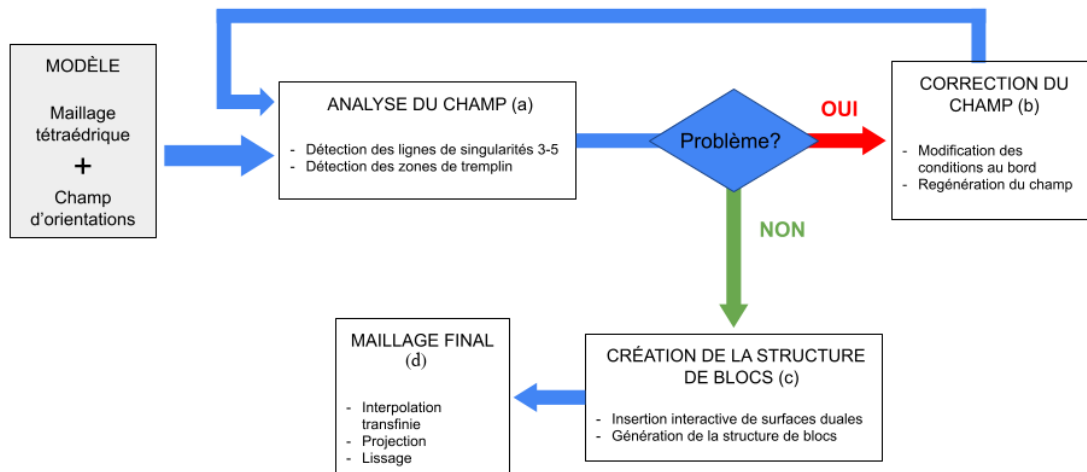


Figure 1.10 – Pipeline de notre méthode de génération de maillages hexaédriques structurés par blocs. L'étape en gris correspond aux données d'entrée de notre méthode.

Dans ce contexte, nous proposons d'étendre la plage d'utilisation des champs d'orientation pour le maillage hexaédrique structuré par blocs en suivant une approche interactive. A la lumière des limites précédemment citées de l'utilisation des champs d'orientation, nous classons les domaines géométriques Ω en trois catégories selon les caractéristiques du champ d'orientation \mathcal{F} que l'on peut générer :

1. **Type V.** La structure topologique de \mathcal{F} est compatible avec celle d'un maillage hexaédrique structuré par blocs ;
2. **Type S.** La structure topologique de \mathcal{F} contient des lignes 3 – 5 ;
3. **Type T.** Des zones de tremplin existent dans le domaine Ω .

Notre approche consiste à proposer un ensemble d'algorithmes à appliquer selon la classification précédente. Ces algorithmes sont regroupés au sein du pipeline de la figure 1.10. Partant de la donnée d'un domaine Ω et d'un champ d'orientation \mathcal{F}_Ω défini sur Ω , la première étape consiste à analyser \mathcal{F}_Ω afin de savoir quel est son type. S'il est de type **S** ou **T**, nous appliquons une étape de correction où l'utilisateur sélectionne interactivement les zones problématiques et la façon dont il souhaite les traiter. Le traitement de cette étape est l'objet du chapitre 4. Une fois le champ rendu valide, nous procédons à une création interactive de la structure de blocs en construisant itérativement le dual de cette structure. La procédure à suivre est décrite au chapitre 3. Le chapitre 5 complète les contributions de notre travail en montrant les résultats obtenus à l'aide de ce pipeline et de l'outil interactif développé durant cette thèse. Le chapitre 2 débutera ce document en introduisant les concepts de base nécessaires à la compréhension de ce travail. Enfin, le chapitre 6 conclura ce document et propose plusieurs perspectives à ce travail.

2 - Notions et concepts pour le maillage hexaédrique

De nombreux codes de simulation numérique nécessitent de disposer d'une discrétisation du domaine étudié qui soit en adéquation avec les schémas numériques employés. Nous nous intéressons dans ce travail au cas particulier et très contraignant des simulations nécessitant des maillages hexaédriques structurés par blocs. La problématique industrielle importante dans ce cas de figure est qu'il n'existe actuellement aucune approche automatique permettant de réaliser de tels maillages. Les ingénieurs y consacrent donc un temps important qui peut aller de plusieurs heures à plusieurs semaines selon les cas.

Dans ce chapitre, nous introduisons l'ensemble des concepts usuels pour le maillage hexaédrique structuré par blocs et essayons d'expliquer pourquoi il est difficile de fournir une solution automatique à la génération de tels maillages. Nous dresserons ensuite un rapide état de l'art des approches suivies aujourd'hui pour résoudre ce problème et positionnerons notre travail dans ce contexte.

En résumé

Dans ce chapitre, nous introduisons le contexte et les notations nécessaires à la lecture de l'ensemble de ce manuscrit. En particulier, nous nous intéressons aux caractéristiques topologiques des maillages hexaédriques et à la relation entre maillage et modèle géométrique de CAO dans le contexte de la discrétisation de pièces mécaniques. Nous positionnons enfin notre travail par rapport aux différents travaux de l'état de l'art en maillage hexaédrique, et plus particulièrement pour le problème du maillage structuré par blocs.

2.1 . Notions et définitions pour le maillage hexaédrique

De nombreuses définitions et notions ne dépendent pas de la dimension n , $n > 0$, de l'espace dans lequel l'objet que nous considérons est plongé. C'est pourquoi nous parlerons souvent d'"objets" de dimension n , même si notre travail se positionne en dimension 3 essentiellement, voire en dimension 2 pour certains exemples.

2.1.1 . Notion de maillages et de cellules

Pour manipuler des maillages hexaédriques, il est nécessaire de comprendre leur structure globale. Cette section introduit pour cela les concepts nécessaires en considérant deux représentations complémentaires : la représentation **primale** et la représentation **duale**.

Maillage primal

Étant donné un domaine géométrique Ω de dimension n , un maillage M de Ω partitionne Ω en un ensemble fini d'objets "simples" de dimension n , que nous nommerons des n -cellules. En dimension 2, les 2-cellules les plus usuellement utilisées sont les triangles et les quadrilatères, tandis qu'en dimension 3, ce seront les tétraèdres et les hexaèdres. Un tétraèdre est un volume bordé de 4 faces triangulaires, 6 arêtes et 4 sommets. Un hexaèdre est bordé de 6 faces quadrangulaires, 12 arêtes et 8 sommets. Le fait que M soit une partition de Ω signifie que Ω est géométriquement totalement recouvert par les n -cellules formant M et que deux n -cellules ne s'intersectent jamais (voir les figures 2.1-b et 2.1-c). En d'autres termes, tout point situé dans Ω appartient à une et une seule n -cellule de M .

D'un point de vue combinatoire, le bord d'une cellule de dimension n , ou n -cellule, est composé de i -cellules avec $0 \leq i \leq n$. De façon usuelle, nous appellerons les 0-cellules des *sommets*, les 1-cellules des

arêtes, les 2-cellules des faces et les 3-cellules des régions. Nous dirons qu'une i -cellule c_i est **incidente** à une n -cellule c_n si $i < n$ et c_i est une cellule qui participe à former le bord de c_n . Nous allons alors dire qu'une n -cellule c_j est **adjacente** à une n -cellule c_k s'il existe une i -cellule c_i incidente à la fois à c_j et c_k . Il est à noter que nous parlerons de **valence** d'un sommet pour indiquer le nombre de n -cellules auxquelles il est incident. Un sommet incident à p n -cellules sera alors de valence p .

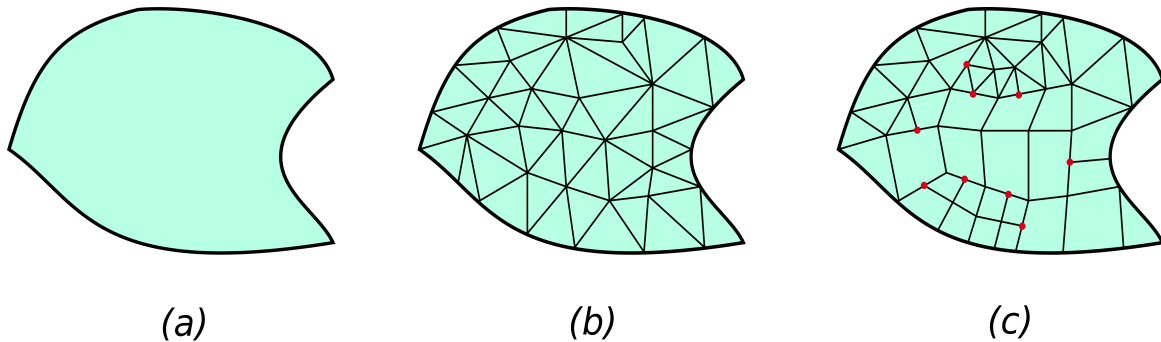


Figure 2.1 – Exemples 2D d'un domaine géométrique simple Ω (a), un maillage triangulaire conforme de Ω (b) et un maillage mixte non conforme où les sommets rouges sont non-conformes (c).

En reprenant la définition de [Mit05], nous appelons **nœud suspendu** un sommet qui se situe à l'intérieur d'une i -cellule, avec $i > 0$. Un maillage est alors dit **conforme** s'il ne possède aucun nœud suspendu. Sinon, il est non-conforme (voir figure 2.1 (c)). En ce qui nous concerne, nous nous intéressons aux maillage conformes purement hexaédriques. De tels maillages sont alors répertoriés comme étant structurés, non structurés ou structurés par blocs :

- Un maillage **structuré** correspond à une grille parfaite qui est déformée pour "épouser" le bord $\partial\Omega$ du domaine Ω . Tous les sommets internes du maillage sont incidents à exactement 4 quadrilatères en 2D et 8 hexaèdres en 3D. De même, à l'exception des sommets de coin, c'est-à-dire les 4 coins de la grille en 2D ou les 8 en 3D, tous les sommets positionnés sur $\partial\Omega$ sont incidents à exactement 2 quadrilatères en 2D et 4 hexaèdres en 3D. Cette régularité est utile pour faciliter l'écriture de schémas numériques que ce soit d'un point de vue informatique (voisinage constant et optimisation mémoire possible) que d'un point de vue mathématique (schémas de type différences finies simples à écrire, faible impact de la structuration du maillage, celui-ci étant similaire partout dans le domaine Ω). Le désavantage principal est qu'il n'est pas possible de discrétiser toute forme de domaine par un tel maillage, et dans le cas où l'on y arrive, on peut avoir des mailles de mauvaise qualité géométrique près de $\partial\Omega$.
- Un maillage **non structuré** pallie ce problème de qualité des mailles au détriment de la structure du maillage. Il offre donc le bénéfice de pouvoir discrétiser n'importe quel domaine Ω avec des mailles de qualités suffisantes pour un code de simulation. Par contre, ce dernier ne peut se reposer sur une structuration régulière du maillage, tout sommet interne pouvant avoir un nombre quelconque de mailles auxquelles il est incident.
- Un maillage **structuré par blocs** combine les qualités de maillages non-structurés et structurés. C'est en fait un ensemble de blocs quadrilatéraux en 2D et hexaédriques en 3D qui forment un macro-maillage non structuré. Chaque cellule de ce macro-maillage, i.e. un bloc, est alors subdivisée de manière structurée par une grille. De cette façon, il est possible de discrétiser tout type de domaine Ω par un ensemble de grilles régulières avec peu de sommets singuliers au sein du domaine.

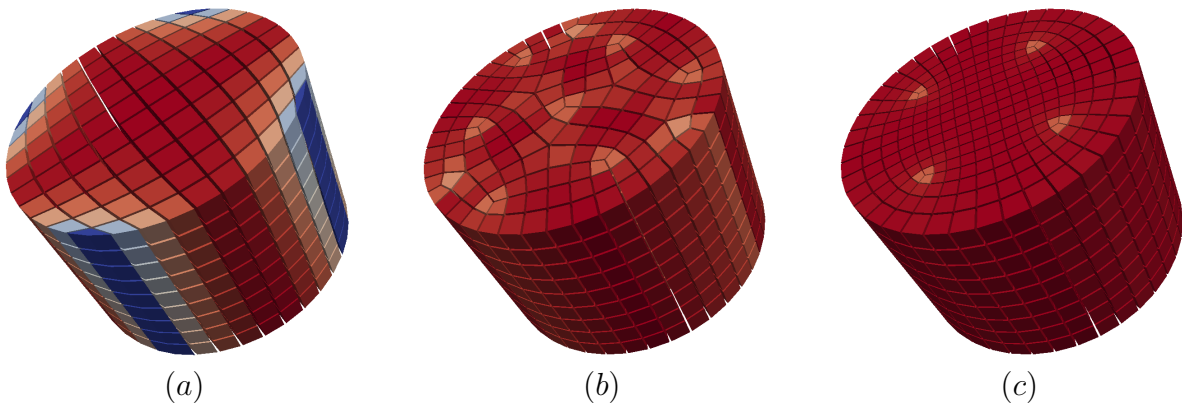


Figure 2.2 – Un même domaine discrétisé à l'aide d'un maillage hexaédrique structuré (a), non structuré (b) et structuré par blocs (c). La couleur des mailles est fonction de la qualité des éléments (qualité de type Scaled Jacobian [Knu01] en l'occurrence). Plus la maille est bleue, pire est sa qualité; plus elle est rouge, meilleure elle est. À taille de maille identique, on peut constater que la qualité individuelle des cellules est la plus mauvaise pour le maillage structuré le long de la frontière.

Structuration d'un maillage hexaédrique et notions associées

La génération de maillages tétraédriques en dimension 3 et triangulaires en dimension 2, bien qu'elle puisse encore faire l'objet d'améliorations sur de nombreux aspects comme la qualité des mailles, les contraintes sur la taille des mailles, l'adaptation des maillages pendant les simulations, les performances, . . . , peut être considérée comme réalisable. Des outils comme [Dis20, Si15, GR09] sont capables, sur une géométrie arbitraire, de produire un maillage tétraédrique et triangulaire de bonne qualité. La génération automatique fiable de maillages hexaédriques en dimension 3 pour une géométrie arbitraire reste quant à elle un problème ouvert. La principale raison à cette difficulté est que les maillages hexaédriques et quadrangulaires disposent d'une structuration topologique interne qui leur est spécifique. Cette structuration induit que les modifications topologiques réalisables sont de nature globale. Pour illustrer nos propos, prenons l'exemple de deux opérations simples qui sont l'insertion de point et la contraction d'arête dans un maillage 2D qui sont montrées aux figures 2.3 et 2.4. Pour chaque opération, un maillage triangulaire et un maillage quadrangulaire sont modifiés.

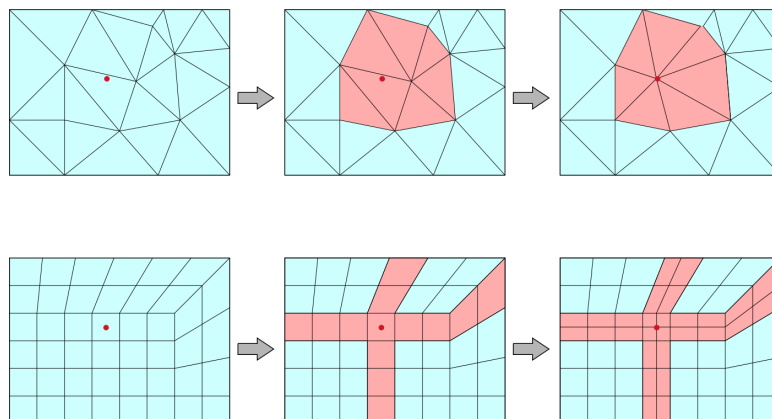


Figure 2.3 – Exemple d'insertion de point pour un maillage triangulaire (en haut) et un maillage quadrangulaire (en bas).

La Figure 2.3 nous montre l'opération d'insertion de point dans un maillage. Pour effectuer cette opéra-

tion dans un maillage triangulaire, une méthode classique consiste en premier lieu à détecter le triangle dans lequel est situé le point, puis à récupérer un ensemble connexe de triangles sous certaines conditions. Cet ensemble de triangles marqués en rouge dans la figure définit une *cavité* qui va ensuite être vidée en supprimant les triangles qu'elle contient. Enfin le maillage va être reconstruit en triangulant la cavité, c'est-à-dire qu'un nouveau triangle va être créé entre chaque arête du bord de la cavité et le point inséré. Dans le cas de maillage quadrangulaire, afin de conserver la conformité du maillage, la méthode d'insertion va différer. Une fois la détection du quadrangle contenant le point effectuée, nous traversons les lignes de quadrangles auxquelles il appartient. De nouvelles lignes de quadrangles contenant le nouveau point vont être insérées afin de conserver la conformité du maillage. De façon similaire la Figure 2.4 nous montre l'opération de contraction d'arête dans un maillage. Pour compresser une arête dans un maillage, les deux extrémités de l'arête vont se rapprocher jusqu'à ne former qu'un seul point unique. Nous pouvons voir que la compression d'arête a un impact local dans un maillage triangulaire. Dans le cas d'un maillage quadrangulaire, effectuer la même opération mène à créer deux triangles. Afin de conserver le maillage quadrangulaire, nous allons, à l'instar de l'insertion de point, propager la contraction d'arête le long de la ligne de quadrangles. Il est à noter que, dans le cas des maillages quadrangulaires, ces deux opérations ne sont pas restreintes au voisinage local et peuvent se propager dans l'ensemble du maillage.

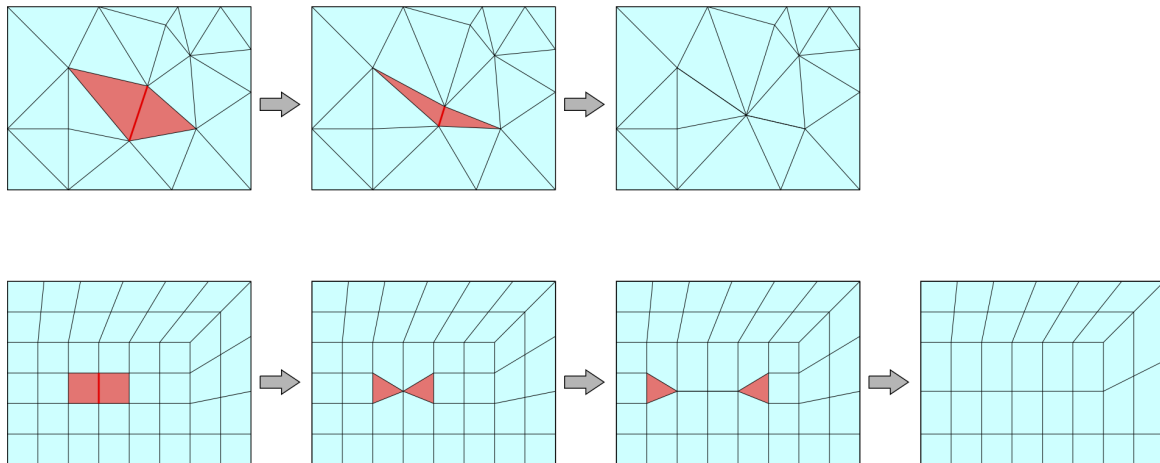


Figure 2.4 – Exemple de contraction d'arête pour un maillage triangulaire (en haut) et un maillage quadrangulaire (en bas).

Considérons maintenant la dimension 3 et les maillages hexaédriques. Pour comprendre la nature globale des modifications topologiques dans un maillage hexaédrique, considérons tout d'abord la structure d'un unique hexaèdre qui est composé de 6 faces, 12 arêtes et 8 sommets. Pour chaque face, il existe une face opposée dans l'hexaèdre, et pour chaque arête, il existe 3 arêtes "parallèles" (voir figure 2.5). Ceci partitionne les faces d'un hexaèdre en trois couples de faces et trois quadruplets d'arêtes. Cette notion d'arêtes parallèles et de faces opposées peut se propager dans tout le domaine donnant corps aux notions de **feuillet**, ou **couches** de mailles d'une part, et de **cordes**, ou **colonnes** de mailles d'autre part.

Considérons tout d'abord la notion de **feuillet**, ou **couche de mailles** à l'aide de l'exemple de la figure 2.6, où nous illustrons le processus de sélection d'un feuillet. En (a), l'arête orange du bord du maillage est sélectionnée. Elle est incidente à un unique hexaèdre dans lequel trois arêtes sont "topologiquement" parallèles à l'arête sélectionnée (colorées elles-aussi en orange sur la figure 2.6.b). Ces nouvelles arêtes appartiennent à d'autres hexaèdres dans lesquels elles ont aussi des arêtes qui leur sont parallèles. De proche en proche, on définit comme cela un ensemble d'arêtes parallèles au sein du maillage (voir figure 2.6.c) et

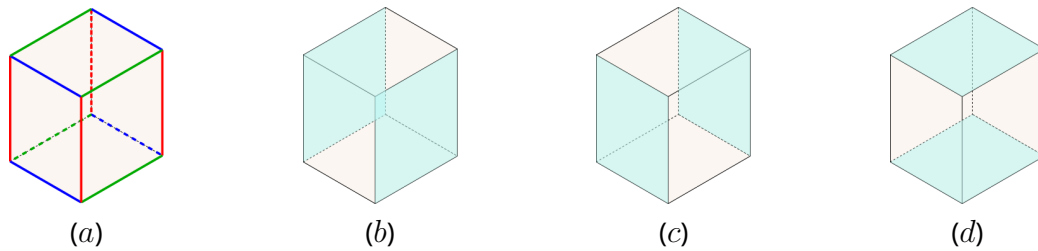


Figure 2.5 – Les arêtes d'un hexaèdre sont réparties en trois ensembles d'arêtes "parallèles" (rouge, vert et bleu en a); les faces sont elles réparties en trois groupes de paires de faces opposées (b, c et d).

un ensemble d'hexaèdres traversés (voir figure 2.6.d). Ces hexaèdres forment une couche de mailles et par construction un hexaèdre peut appartenir à trois couches de mailles au plus ¹.

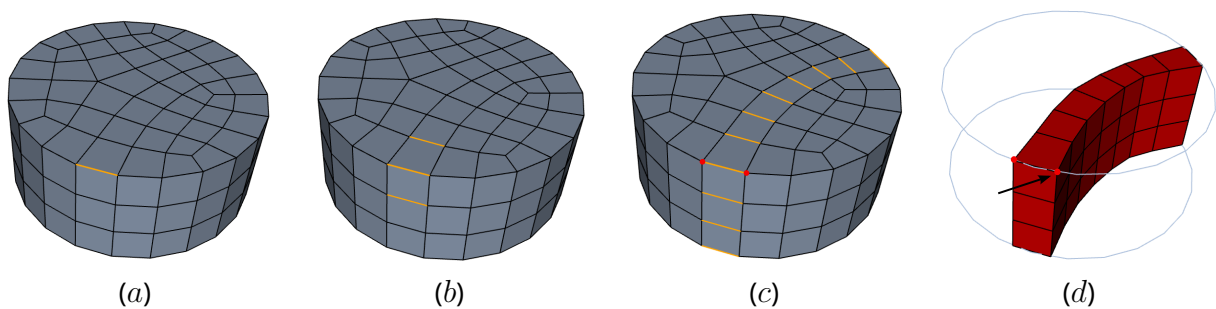


Figure 2.6 – Sélection usuelle d'un feuillet en considérant une arête sélectionnée (a) et la propagation de proche en proche dans le maillage en considérant les arêtes opposées dans les hexaèdres adjacents (b puis c) pour obtenir une couche complète d'hexaèdres (d).

Si l'on considère maintenant les faces opposées dans un hexaèdre, on peut appréhender la notion de **corde**, ou **colonne de mailles**. En sélectionnant une face dans un maillage hexaédrique on peut se propager de proche en proche au sein des hexaèdres voisins pour accéder à d'autres hexaèdres et ainsi de suite. Comme pour les feuillets, on parcourt ainsi une partie du maillage, comme on peut le voir en rouge sur la figure 2.7.a, b, c et d.

Maintenant que nous avons défini les notions de feuillets et de cordes comme éléments topologiques globaux aux maillages hexaédriques, nous allons pouvoir donner quelques exemples d'opérations de modification topologique. Les Figures 2.7 et 2.8 montrent une opération de contraction respectivement sur une corde et sur un feuillet.

Les Figures 2.9 et 2.10 illustrent quant à elles des opérations classiques d'insertion de mailles hexaédriques en utilisant les feuillets. La figure 2.9 illustre l'opération de découpage, ou "dicing" [MBT97]. Cette opération raffine le maillage en coupant les arêtes parallèles qui forment le feuillet, autrement dit nous coupons la couche de maille entièrement pour en créer une autre.

Dans le cas de la Figure 2.10, nous pouvons voir l'opération de "pillowing" [MT95], ou gonflement. Cette opération permet d'insérer une couche d'hexaèdres, qui forme un oreiller qui va entourer/envelopper un ensemble de mailles (en rouge). La création de l'oreiller se fait en trois étapes :

1. Identifier un ensemble de faces qui forment la surface qui sépare le maillage en deux sous-ensembles ;
2. Séparer les deux sous-ensembles d'hexaèdres le long de la surface. Cela consiste à dupliquer tous les sommets, arêtes et faces de la surface considérée ;

1. Au plus 3 couches de mailles, car une couche peut s'auto-intersecter et donc traverser un hexaèdre 2 ou 3 fois.

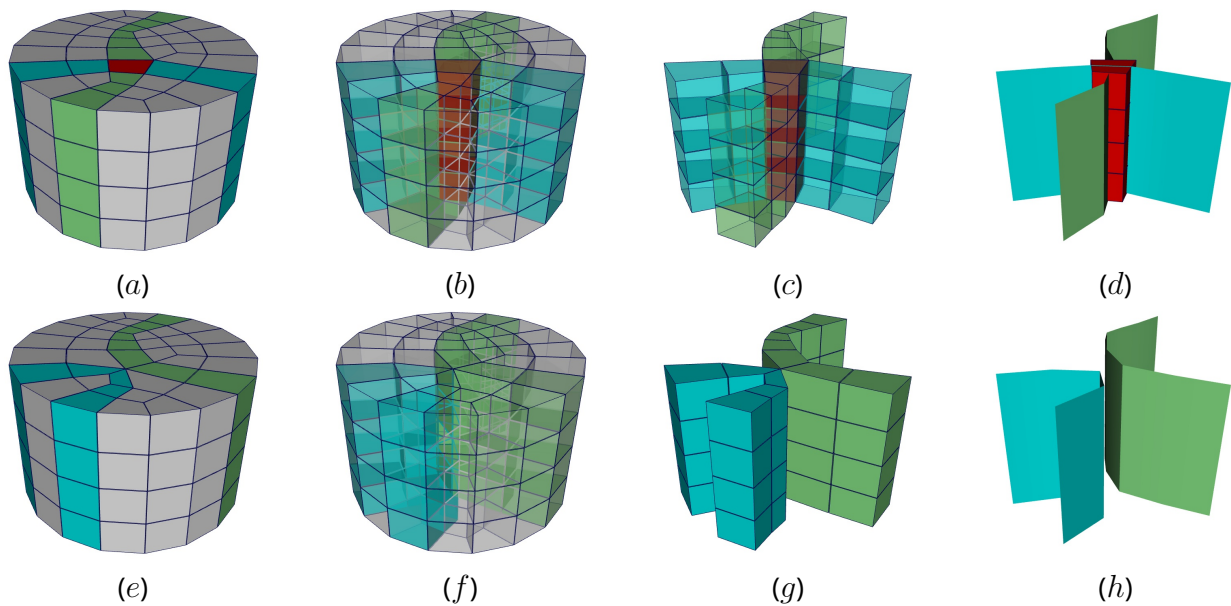


Figure 2.7 – Illustration de la notion de corde, ou couche de mailles en (a), (b), (c) et (d) et suppression de la corde en question, ce qui induit que les deux feuillets bleus et verts qui s'intersectaient auparavant ne font que se toucher le long d'une ligne d'arêtes ensuite (en e, f, g et h). Notons que les 3 premières colonnes de cette figure représentent tout ou partie du maillage sous forme primale tandis que la dernière colonne se base sur une vue duale (voir la notion de maillage dual page 26).

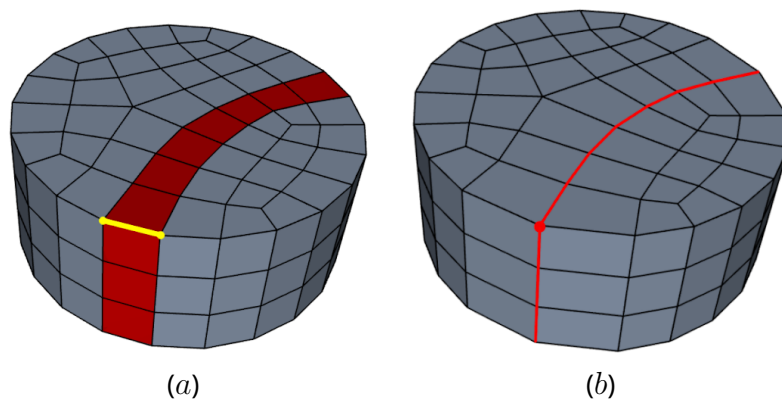


Figure 2.8 – Opération de contraction de l'arête jaune (a), toutes les arêtes du feuillet (en rouge) auquel appartient l'arête sont contractées (b).

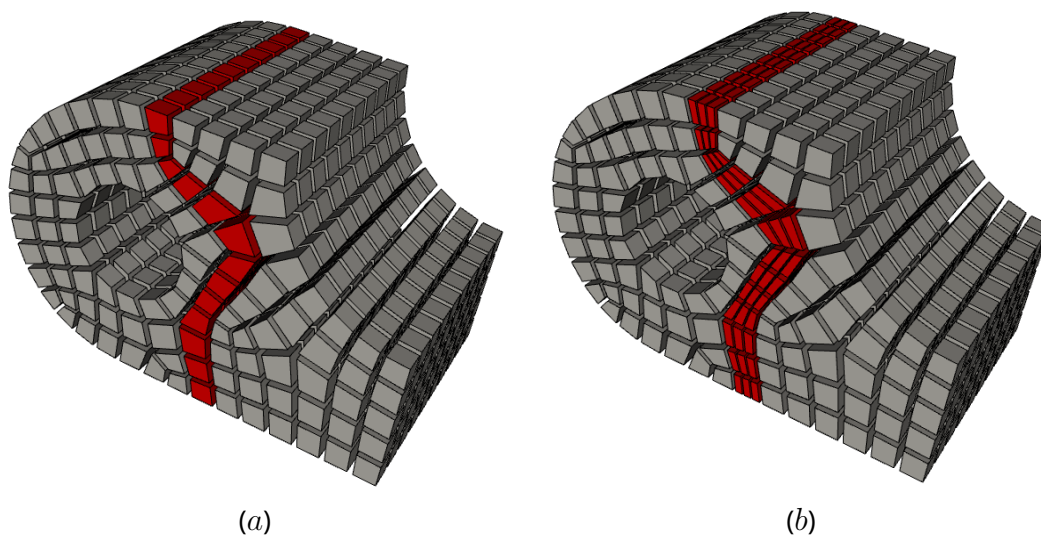


Figure 2.9 – Après la sélection d'une couche de mailles (a), on raffine le maillage en "multipliant" cette couche par découpe de l'ensemble des arêtes "parallèles" dans le feuillet (3 fois en b).

3. Connecter les deux sous-ensembles en insérant une couche d'hexaèdres. Les arêtes de la couche insérée relient les sommets qui ont été dupliqués lors de l'étape de séparation.

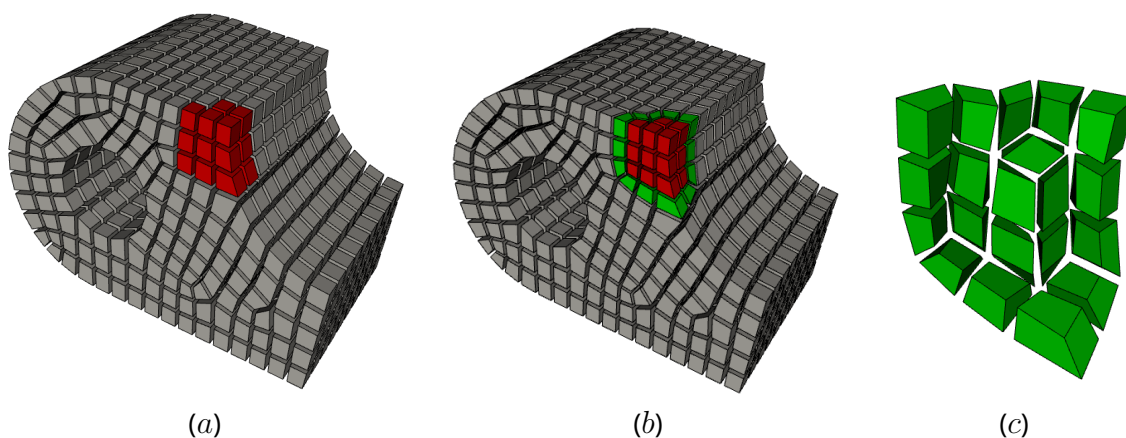


Figure 2.10 – Opération d'insertion de feuillets réguliers.

Toutes ces opérations permettent de modifier le maillage en conservant sa conformité. Il est à noter que certaines opérations, comme la contraction de corde peut modifier la topologie du maillage et ainsi faire apparaître des sommets singuliers.

Le sujet de nos travaux portant sur la génération de maillages hexaédriques structurés par blocs, nous allons donner quelques éléments sur la structuration par blocs. Nous appelons **structure de blocs** le macro-maillage hexaédrique qui définit cette structuration. Afin de clarifier le discours, nous appellerons **blocs** les 3-cellules de ce macro-maillage dans la suite du document. Cette structure de blocs étant un maillage hexaédrique, elle possède les mêmes caractéristiques que ceux-ci : on y retrouve les notions de feuillets et de cordes ainsi que les opérations de modification associées. La figure 2.11 illustre la relation entre la structure de blocs (b) et le maillage hexaédriques obtenu (c). Le maillage en (c) est obtenu après avoir subdivisé les blocs, en utilisant notamment l'opération de découpe : nous découpons des feuillets de blocs pour créer un

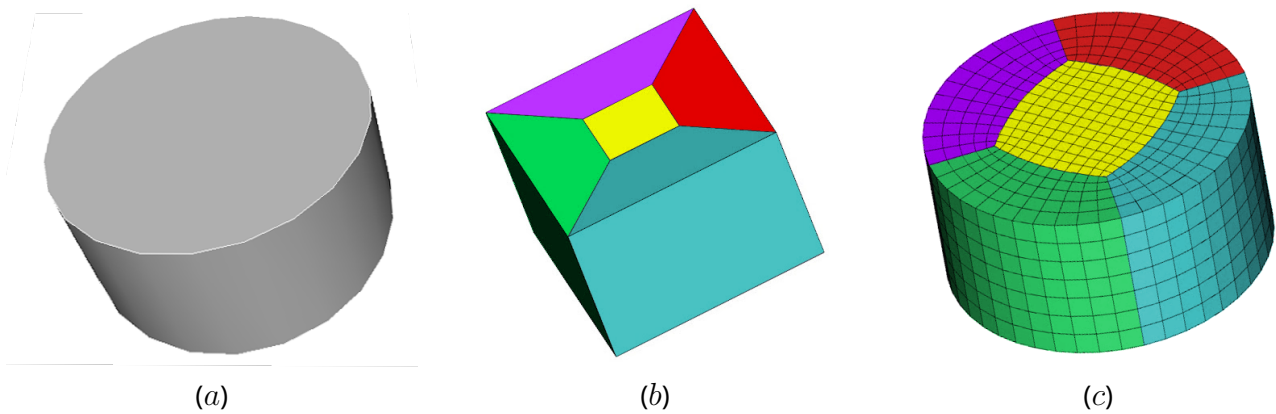


Figure 2.11 – Un exemple de modèle géométrique (a) à partir de laquelle une structure de blocs a été créée (a) et le maillage hexaédrique structuré obtenu en subdivisant les blocs (b).

ensemble de couche de mailles hexaédriques. Les hexaèdres du maillage (c) sont coloriés avec la couleur du bloc d'origine, cela permet de mettre en évidence que chaque subdivision de bloc produit une grille régulière et que les seuls sommets singuliers du maillage final sont issus des sommets singuliers de la structure de blocs (les sommets du bloc jaune en (b)). Des méthodes comme [LPP⁺20] proposent une décomposition de la géométrie en blocs en traçant des lignes de coupe sur la surface du domaine. Dans cette approche, les auteurs acceptent d'obtenir une structure de blocs comportant des éléments non hexaédriques. Chaque bloc est ensuite subdivisé avec une méthode de *midpoint subdivision* [PA97a], ou subdivision barycentrique, pour obtenir un maillage complètement hexaédrique.

Maillage dual pour le maillage hexaédrique

La notion de maillage dual est relative à la topologie, i.e. la structure du maillage et non à sa géométrie². Comme nous l'avons vu précédemment, la structure d'un maillage hexaédrique a une grande importance du fait de son caractère global. Cela va se traduire par une définition spécifique pour son maillage dual.

Notons \mathcal{M}^* le maillage dual d'un maillage M de dimension n . Alors, pour tout $k \in \llbracket 0, n \rrbracket$, pour toute k -cellule de M correspond une $(n-k)$ -cellule de \mathcal{M}^* . On a donc une "équivalence" directe entre les cellules des deux maillages. On retrouve cela aussi dans les relations d'incidence et d'adjacence : Si deux k -cellules de M partagent une $k-1$ -cellule c , alors les deux $(n-k)$ -cellules correspondantes de \mathcal{M}^* appartiendront au bord de la $(n-k+1)$ -cellule duale de la cellule c . Dans cette relation, M est qualifié de maillage **primal**. Un exemple de cette notion de dual est la relation qui relie la triangulation de Delaunay au diagramme de Voronoï où chacun est le dual de l'autre. Dans ce cas présent, un critère géométrique est aussi pris en compte. Par contre, d'un point de vue structurel, le dual d'un maillage triangulaire peut être construit comme suit (voir la figure 2.12) : pour chaque triangle du maillage primal, créer un point en son centre ; relier par une arêtes les points créés si leur triangles primaux partagent une arête ; pour chaque sommet du maillage primal, construire la face duale correspondante qui est bordée par le cycle d'arêtes duales connectant les sommets duaux des triangles primaux l'entourant.

Les structures duales de maillages quadrangulaires et hexaédriques auront, à l'instar de leurs maillages primaux, une structuration particulière. En l'occurrence, le dual d'un maillage quadrangulaire est un arrangement simple de courbes (voir figure 2.13(a) et (b)) et le dual d'un maillage hexaédrique est un arrangement simple de surfaces (voir figure 2.14)[MBBM97]. Un arrangement simple de courbes, respectivement de sur-

2. Pour des maillages linéaires, ou d'ordre 1, la géométrie du maillage est totalement induite par la position géométrique de ses sommets.

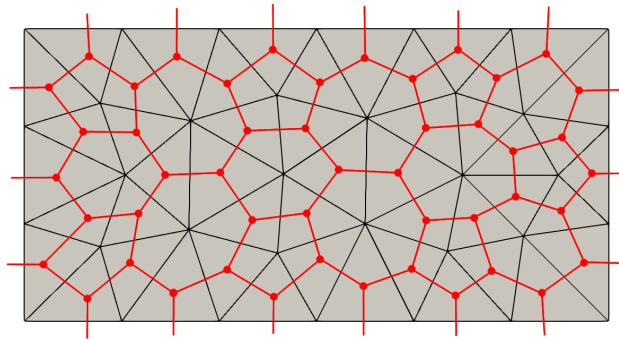


Figure 2.12 – Un exemple de maillage triangulaire en noir et de son maillage dual polyédrique en rouge.

faces, est un ensemble de courbes qui partitionnent l'espace 2D, respectivement 3D, E tel qu'un point de E peut être le lieu de l'intersection d'au plus deux courbes, respectivement trois surfaces.

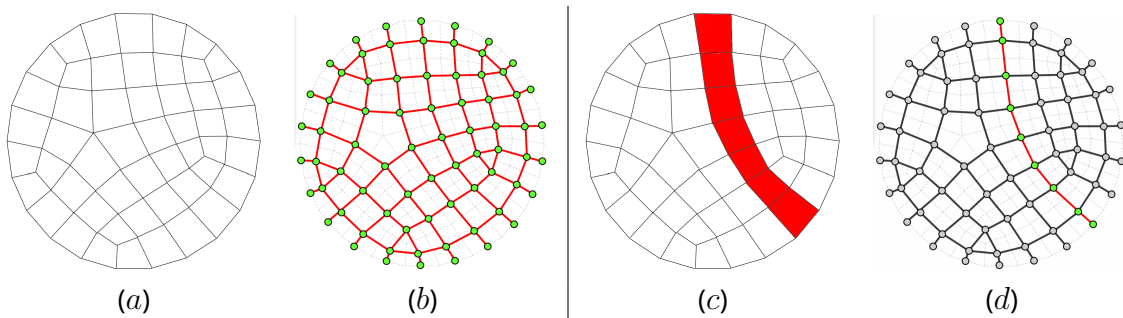


Figure 2.13 – Un maillage quadrangulaire (a) et son dual (b) et la relation entre un feuillet (c) et une courbe duale (d).

Il existe une relation directe entre le dual d'un maillage quadrangulaire ou hexaédrique et les notions de feuillets et de cordes. La figure 2.13(c, d) illustre cette relation sur un exemple 2D. Le feuillet du maillage quadrangulaire montré en rouge en (c) correspond à une courbe de la structure duale en rouge en (d). De la même façon, en 3D, un feuillet d'un maillage hexaédrique correspond à une surface de la structure duale, que nous appellerons dans la suite du document **surface duale**. Une corde dans un maillage hexaédrique correspond quant à elle à l'intersection de deux surfaces duales comme montré à la Figure 2.7(d). L'intersection de trois surfaces duales en un point du domaine correspond à un hexaèdre du maillage primal. Il est à noter qu'une surface duale peut s'auto-intersecter.

Nous pouvons utiliser la relation entre le dual et le primal pour facilement modifier le maillage hexaédrique. En effet, insérer une surface dans la structure duale est une opération bien définie qui équivaut à insérer une couche de mailles dans le maillage primal.

2.1.2 . Qualité géométrique d'un maillage hexaédrique

En simulation numérique, la qualité d'un maillage dépend principalement des critères du code de simulation qui l'utilise, voire même du phénomène physique que l'on cherche à modéliser. Ainsi pour un même code de simulation et un même domaine Ω , on peut faire des choix différents de maillage. S'il est donc complexe de vouloir définir une métrique de qualité universelle, on peut toutefois admettre que disposer de cellules de mauvaise qualité géométrique est problématique pour de nombreux schémas numériques dont les calculs sont justement basés sur la géométrie des cellules. Habituellement, on veut éviter les cellules avec

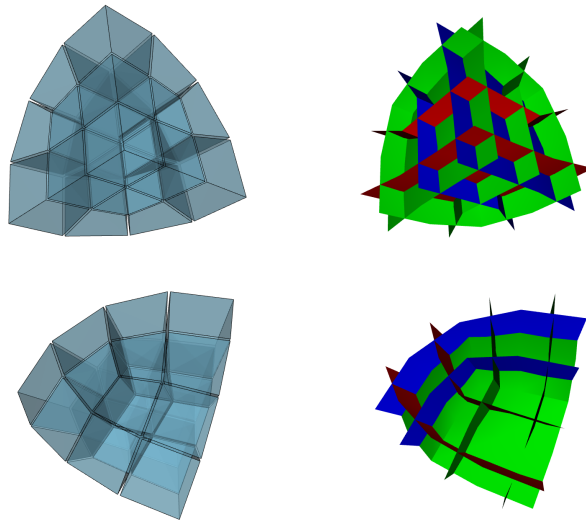


Figure 2.14 – Un maillage hexaédrique (gauche) et son dual (droite).

des volumes négatifs par exemple (cellules retournées qui s'auto-intersectent, cellules non convexes). C'est particulièrement le cas pour des schémas numériques de type "éléments finis" qui discrétisent le problème en éléments simples basés sur des fonctions de forme dépendant de la géométrie des cellules. Des cellules avec des angles très aigus ou obtus peuvent engendrer dans certains cas des problèmes de précision numérique. Afin de caractériser cette qualité géométrique, on peut se reposer sur le fait que chaque cellule, ici hexaédrique, peut être vue comme une déformation du cube unité, ou *élément de référence* (voir Figure 2.15). La géométrie de n'importe quel hexaèdre peut être transformée en l'élément de référence via une fonction de forme. Nous notons (x, y, z) (respectivement (x, y) en dimension 2) les coordonnées d'un point dans l'espace physique dans lequel vit notre hexaèdre et nous notons (ξ, η, ζ) (respectivement (ξ, η) en dimension 2) les coordonnées dans l'espace de référence (ou espace logique).

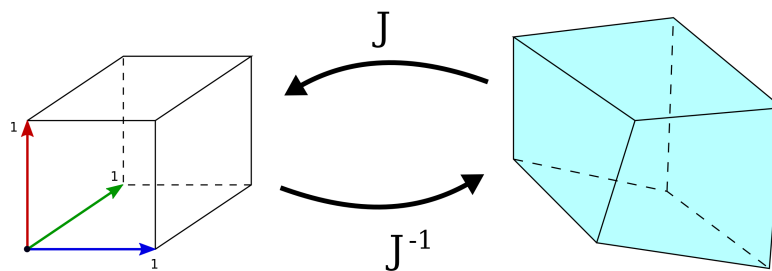


Figure 2.15 – Du cube de référence situé dans l'espace logique à un hexaèdre situé dans l'espace physique et inversement.

Une fonction de forme est bijective en chaque point de l'élément de référence et de son bord³ si la matrice jacobienne J de la transformation correspondant au changement de variable entre l'espace logique et l'espace physique n'est pas singulière. Cette condition peut être vérifiée en considérant que le déterminant de J n'est jamais nul en tout point du cube de référence. En d'autres termes, il est strictement positif. En dimension 2,

3. A chaque point de l'élément de référence correspond un et un seul point de l'hexaèdre situé dans l'espace physique.

la matrice jacobienne J_2 est définie comme suit :

$$\begin{pmatrix} \frac{\partial T}{\partial \xi} \\ \frac{\partial T}{\partial \eta} \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{pmatrix} \begin{pmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{pmatrix} = J \begin{pmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{pmatrix}. \quad (2.1)$$

De même, en dimension 3, on obtient la définition de J_3 :

$$\begin{pmatrix} \frac{\partial T}{\partial \xi} \\ \frac{\partial T}{\partial \eta} \\ \frac{\partial T}{\partial \zeta} \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{pmatrix} \begin{pmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{pmatrix} = J \begin{pmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{pmatrix}. \quad (2.2)$$

Quelle que soit la dimension (2 ou 3), la matrice jacobienne d'un simplexe T , peut être calculée en fonction des valeurs en ses sommets $\{s_i(x_i, y_i, z_i)\}_{i \in [0, n]}$, avec $n = 2$ ou 3 :

$$|J_2| = \begin{vmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{vmatrix} \quad \text{et} \quad |J_3| = \begin{vmatrix} x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\ x_2 - x_0 & y_2 - y_0 & z_2 - z_0 \\ x_3 - x_0 & y_3 - y_0 & z_3 - z_0 \end{vmatrix} \quad (2.3)$$

Dans le cas d'une cellule hexaédrique, une méthode usuelle pour calculer sa qualité géométrique est de considérer les huit tétraèdres issus de chacun de ses sommets et des trois arêtes incidentes. Cette approximation est usuelle même si elle ne garantit pas que l'on ait traité tous les points contenus dans l'hexaèdre (voir par exemple [Knu00, Knu01, Knu07] pour une discussion sur les cellules dont la qualité est définie par plusieurs jacobiens). Le déterminant le plus petit parmi les huit matrices issues de ses sommets définit la métrique *Jacobienne* de l'hexaèdre. Les longueurs de l'hexaèdre peuvent en outre être normalisées pour fournir un version *scaled* dont les valeurs sont comprises entre -1 et 1 (voir Figure 2.16) : une valeur négative indique que l'élément est non convexe ; une valeur de 1 indique que tous les angles internes entre deux faces valent 90° et généralement des valeurs acceptables pour une simulation sont comprises entre 0.2 et 1.0.

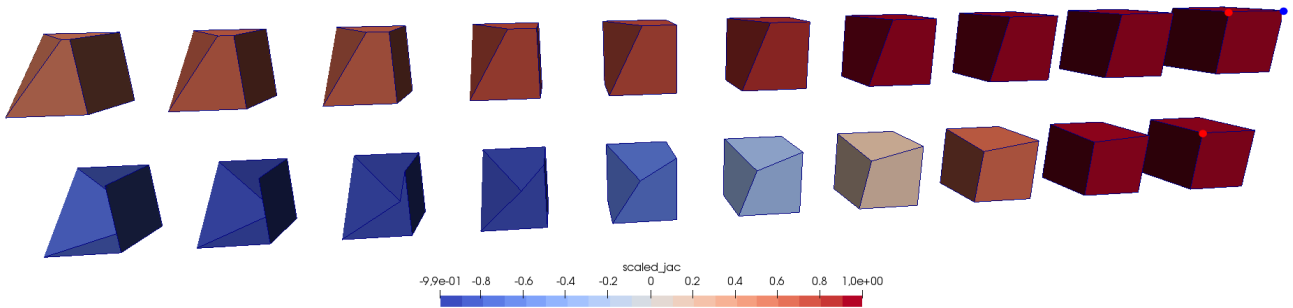


Figure 2.16 – Évolution de la valeur du *scaled jacobian* quand on déplace progressivement un sommet d'un cube parfait de la droite vers la gauche. Sur la ligne du haut, le sommet rouge se déplace vers le sommet bleu ; sur la ligne du bas, le sommet rouge se déplace vers le sommet opposé le long de la diagonale du cube.

2.1.3 . Classification d'un maillage sur un domaine géométrique

La définition d'un domaine géométrique Ω peut se faire de nombreuses manières. Il peut par exemple être défini de façon analytique par une fonction indiquant quels points de l'espace appartiennent à Ω . Il peut aussi être le résultat d'une succession d'opération Booléennes (union, intersection, différence) partant de primitives simples (en 3D, des sphères, pavés, prismes, cônes, cylindres, etc.), ou encore à partir d'un ensemble connecté de courbes et surfaces formant son bord. On parle alors de représentation *BRep* pour

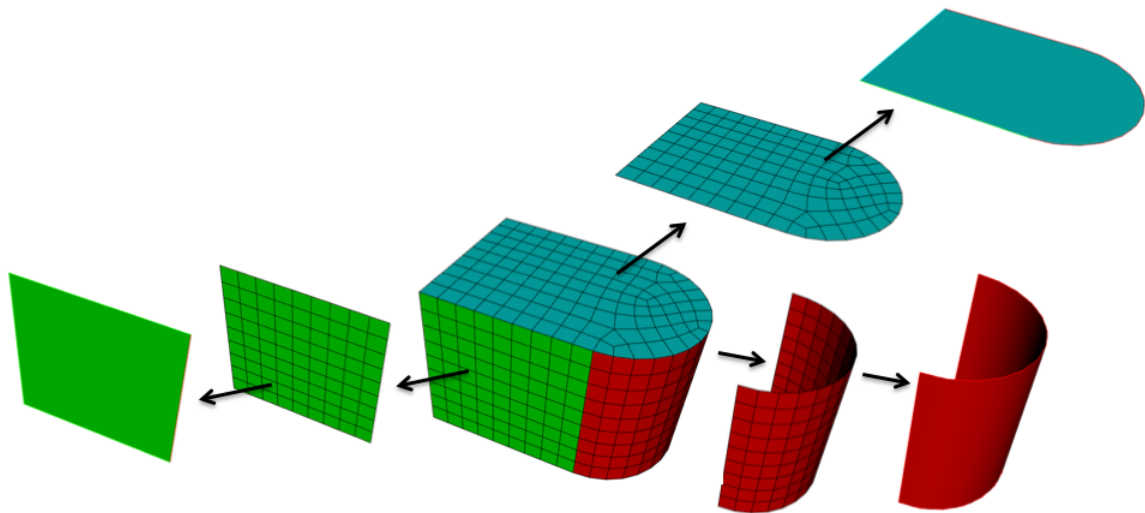


Figure 2.17 – Classification d'un maillage sur un modèle géométrique; les cellules classifiées sur une même entité géométrique sont coloriées de la même couleur.

"représentation par le bord". C'est cette dernière représentation, usuelle en CAO, que nous considérons principalement dans ce travail.

Dans le cas 3D d'un modèle géométrique de type BRep, ce modèle est composé de volumes, surfaces, courbes et sommets géométriques. Comme pour un maillage conforme, un modèle géométrique BRep partitionne l'espace mais avec plus de degrés de liberté, voire en pratique d'erreurs (surfaces qui s'intersectent, courbes de taille nulle, espaces non prévus entre courbes ou surfaces, etc.). Dans notre travail nous considérons que les modèles géométriques sont étanches (sans trous) et géométriquement valides (pas d'intersection entre surfaces, courbes, volumes). De manière informelle, un domaine géométrique de dimension 3 pourra être défini de manière constructive par un quadruplet (V, S, C, P) comme suit :

1. P est un ensemble non vide de points, ou sommets géométriques, de \mathbb{R}^3 ;
2. C est un ensemble de courbes de classe C^1 au moins, reliant des points de P ;
3. S est un ensemble de surfaces de classe C^1 au moins, chacune bordée par des courbes de C formant un ou plusieurs cycles⁴ ;
4. V est un ensemble non vide de volumes bordés par des surfaces de S^5 ;
5. Tout point de P est incident à au moins une courbe de C ;
6. Toute courbe de C est incidente à au moins une surface de S ;
7. Toute surface de S est incidente à au moins un volume de V .

En considérant une telle représentation de la géométrie, nous reprenons la notion de *classification géométrique* introduite dans [RS03], pour associer maillage et géométrie. Etant donné un domaine géométrique Ω représenté par un quadruplet $G_\Omega = (V, F, C, S)$ et un maillage M de Ω , la **classification géométrique** de M sur Ω est la donnée d'une surjection $\psi : M \rightarrow G_\Omega$ telle que pour toute cellule $c \in M$, $\dim(\psi(c)) \geq \dim(c)$. En d'autres termes, toute cellule de M est associée à une entité géométrique de G_Ω de dimension au moins égale. Puisque ψ est une surjection, plusieurs cellules de M peuvent être associées à la même entité géométrique de G_Ω . Ceci est nécessaire puisque une entité géométrique est discrétisée par plusieurs cellules de M .

4. Dans ce cas, un cycle représente le bord externe de la surface tandis que les autres modélisent des trous dans celle-ci.

5. Là-aussi, on peut avoir un ou plusieurs ensembles de surfaces pour modéliser l'existence de volumes à plusieurs bords.

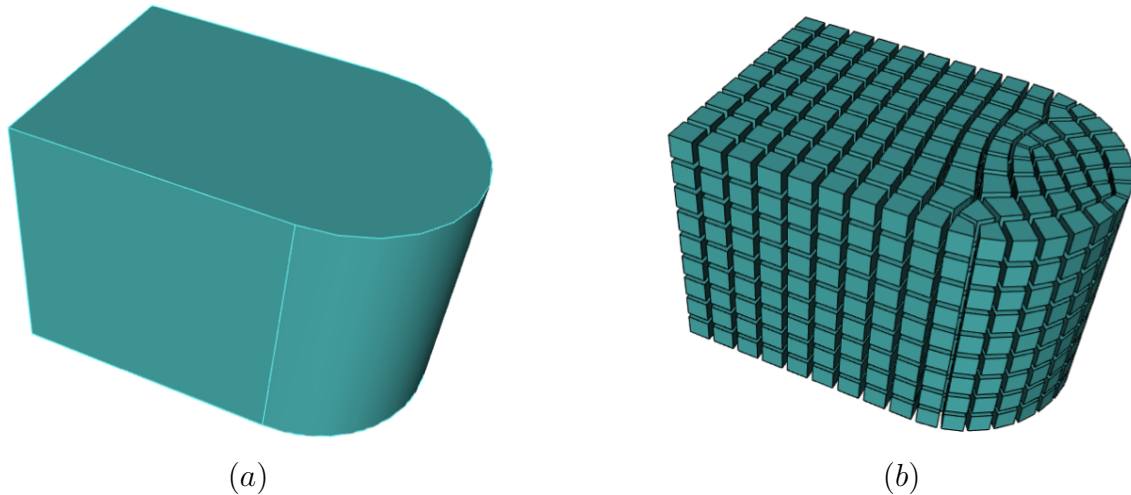


Figure 2.18 – Exemple d'un volume géométrique en (a) et d'un maillage hexaédrique le partitionnant en (b).

Considérons le volume et le maillage hexaédrique représenté sur la figure 2.18 : le maillage proposé en (b) partitionne bien le domaine géométrique représenté en (a). Sur la figure 2.17, est fournie la classification géométrique de trois surfaces de maillage. Celles-ci sont respectivement classifiées sur les surfaces géométriques S_1 , S_2 et S_3 . Inversement, on dira d'une surface géométrique, par exemple S_1 , qu'elle est *discrétisée* par la surface de maillage S_1^M . Nous avons maintenant tous les éléments pour définir la classification géométrique d'un maillage sur un objet géométrique. Étant donné le domaine géométrique $G_\Omega = (V, S, C, P)$ de dimension 3 et M un maillage le discrétisant. On dira que M est géométriquement classifié sur G_Ω si :

- Pour tout $s \in S$, il existe une unique surface s_s de maillage de M qui soit classifiée sur s ;
- Pour tout $c \in C$, il existe une unique ligne de maillage l_c de M qui soit classifiée sur c ;
- Pour tout $p \in P$, il existe un unique nœud n_p de M qui soit classifié sur p ;
- Si une courbe $c \in C$ est incidente à une surface $s \in S$, alors pour toute cellule (sommet ou arête) de l_c , il existe une face de s_s qui lui est incidente ;
- Si un point $p \in P$ est incident à une surface $s \in S$ alors il existe une face de s_s incidente à n_p ;
- Si un point $p \in P$ est incident à une courbe $c \in C$ alors il existe une arête de l_s incidente à n_p .

En respectant cette classification entre maillage et domaine géométrique, nous assurons la cohérence globale des associations géométriques des 0-, 1- et 2-cellules avec les entités géométriques de dimension inférieure à 3 strictement. Nous n'indiquons rien pour ce qui concerne les volumes géométriques et les 3-cellules du maillage. En pratique, la classification des 3-cellules géométriques est assurée par construction ou par coloriage en connaissance de la classification géométrique des cellules de dimension inférieure, qui sont classifiées sur le bord des volumes.

2.2 . Comment générer un maillage hexaédrique ?

Partant d'un domaine géométrique Ω , l'objectif est de générer un maillage hexaédrique permettant de répondre aux besoins d'un code de simulation. Ces besoins peuvent différer selon les schémas numériques employés, les problèmes physiques étudiés, et même le cas d'étude lui-même. Cependant, comme indiqué précédemment, des critères usuels sont recherchés :

- **Alignement des mailles au bord.** De manière générale, il est préférable de disposer de couches de mailles alignées le long de $\partial\Omega$. Cela peut aussi être le long de gradients physiques dans des simulations (alignement avec l'écoulement d'un fluide, un choc, etc.).

- **Qualité individuelle des mailles.** Pour exécuter un code de simulation numérique sur un maillage, il faut le plus souvent disposer de mailles de "bonne" qualité. Une unique maille de *scaled Jacobian* négatif peut empêcher l'exécution d'un code.
- **Régularité de la structure du maillage.** On souhaite disposer d'une structure de maillage la plus régulière possible (proche d'une grille presque partout).
- **Contrôle de la taille des éléments.** La taille des mailles doit pouvoir différer selon le lieu géométrique au sein du domaine Ω . On voudra en général de "petites" mailles dans les zones d'intérêt pour bien capturer la géométrie du phénomène physique étudié et des "grosses" mailles ailleurs pour minimiser l'empreinte mémoire globale et le temps d'exécution (généralement corrélé avec le nombre de mailles à parcourir et traiter).

L'ensemble de ces critères est *à priori* satisfait si l'on génère un maillage hexaédrique structuré par blocs, mais ceux-ci sont particulièrement difficiles à créer automatiquement. En pratique, ils sont généralement construits à l'aide d'outils semi-automatiques interactifs. Nous évoquons ces outils dans la suite de ce chapitre en considérant aussi les principales approches automatiques existantes. Pour un état de l'art plus complet, le lecteur intéressé pourra se référer à [SRRGRN14] par exemple.

Approches indirectes

Les approches indirectes reposent sur la génération d'un maillage tétraédrique où les tétraèdres seront modifiés topologiquement pour obtenir des hexaèdres. Deux approches sont généralement possibles. Tout d'abord, on pourra par exemple découper chaque tétraèdre en quatre hexaèdres (voir Fig. 2.19). Cette approche est robuste et automatique mais ne remplit aucun critère souhaité à l'exception de la possibilité de contrôler la taille des cellules.

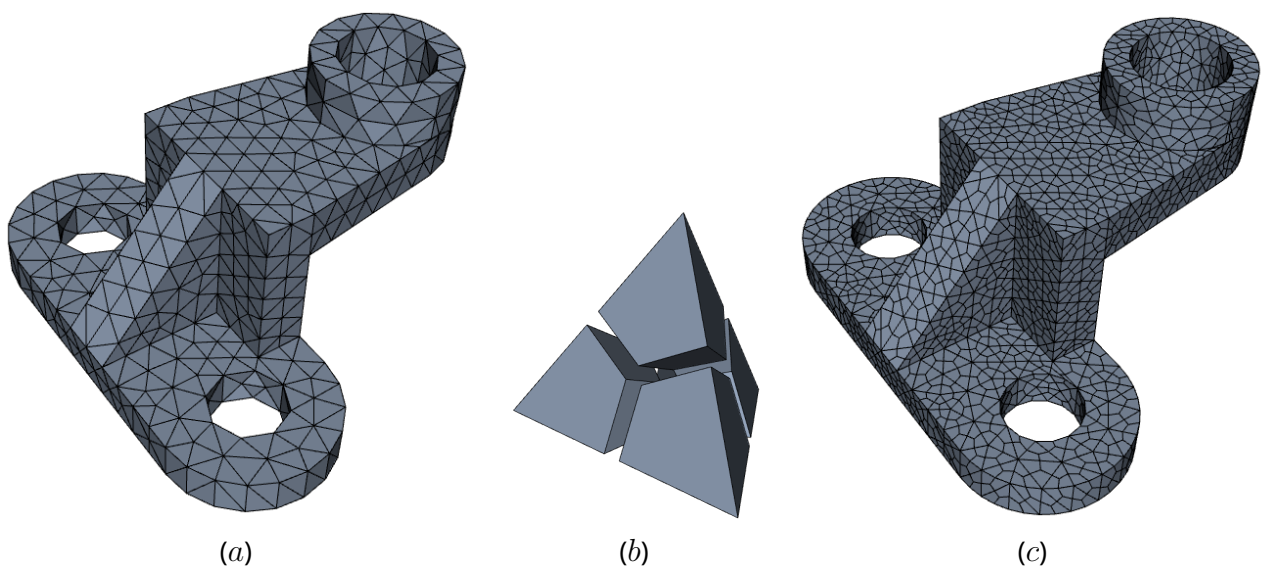


Figure 2.19 – Maillage par approche indirecte où chaque tétraèdre d'un maillage tétraédrique (a) est découpé en quatre hexaèdres suivant le motif illustré en (b) pour obtenir un maillage purement hexaédrique non structuré (c).

Le second type d'approche indirecte consiste à recombinaison entre eux des ensembles de tétraèdres pour former des hexaèdres. C'est notamment le cas des travaux de [PJVR18] (voir Fig. 2.20). Une fois le maillage tétraédrique \mathcal{T} généré, les auteurs identifient tous les hexaèdres valides qui peuvent être créés à partir des tétraèdres composant \mathcal{T} . Cette approche permet de générer des maillages de façon automatique. Cependant,

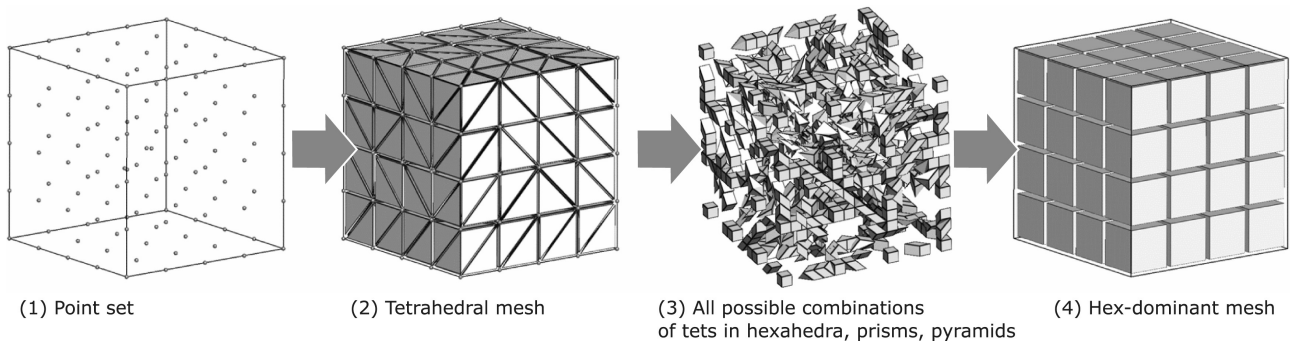


Figure 2.20 – Maillage par combinaison de tétraèdres [PJVR18]. En partant d'un ensemble de points en (1), un maillage tétraédrique va être généré (2). Ensuite en (3), les combinaisons de tétraèdres en hexaèdres et prismes sont identifiées dans ce maillage tétraédrique. Et pour finir le maillage final est généré en (4).

il n'y a aucune garantie d'obtenir des maillages composés entièrement d'hexaèdres. Ils sont considérés comme *hybrides*, c'est-à-dire qu'ils possèdent des hexaèdres mais également des tétraèdres et des prismes dans les zones où aucun hexaèdre valide n'a pu être identifié. On parle alors souvent de maillages **hex-dominants**.

Méthode	Alignement au bord	Qualité géométrique	Structuration	Contrôle de la taille
Indirecte	Non	Non, la découpe des tétraèdres induit des angles diédraux trop faibles ou des maillages hybrides	Non	Oui

Approche par intersection de grilles

Les approches par intersection de grilles [Sch96, Sch97, ZB04, Mar09, ISS09, HQZ13, GSP19, LPC21, PLC+21] immergent le domaine Ω au sein d'un maillage hexaédrique déjà existant, le plus souvent une grille parfaite G_Ω (par exemple une discrétisation de la boîte englobante de Ω). Le processus des différentes méthodes est alors le plus souvent le même (voir figure 2.21) :

1. De manière optionnelle, G_Ω est raffinée localement pour être bien adaptée à la capture des détails du bord de Ω ;
2. Les cellules de G_Ω incluses totalement ou partiellement dans Ω sont conservées. Cela produit le maillage M_Ω ;
3. Les sommets, arêtes et faces du bord de M_Ω sont projetés sur $\partial\Omega$.
4. Des couches d'hexaèdres sont insérées le long du bord de M_Ω pour en améliorer la qualité.

Au même titre que l'approche indirecte, l'utilisation d'une intersection de grilles offre l'avantage de la robustesse et en plus une structuration importante du maillage loin du bord de Ω . Le problème principal est justement relatif au bord et au manque de contrôle que l'on a sur la génération d'hexaèdres dans cette zone. En outre, l'utilisation de l'étape optionnelle de raffinement pour contrôler la taille des éléments où on le souhaite peut entraîner d'augmenter grandement le nombre de cellules du maillage (voir figure 2.22).

Méthode	Alignement au bord	Qualité géométrique	Structuration	Contrôle de la taille
Intersection de grille	Oui si insertion terminale de feuillets	Difficile à contrôler	Non	Oui si la grille est raffinée en conséquence

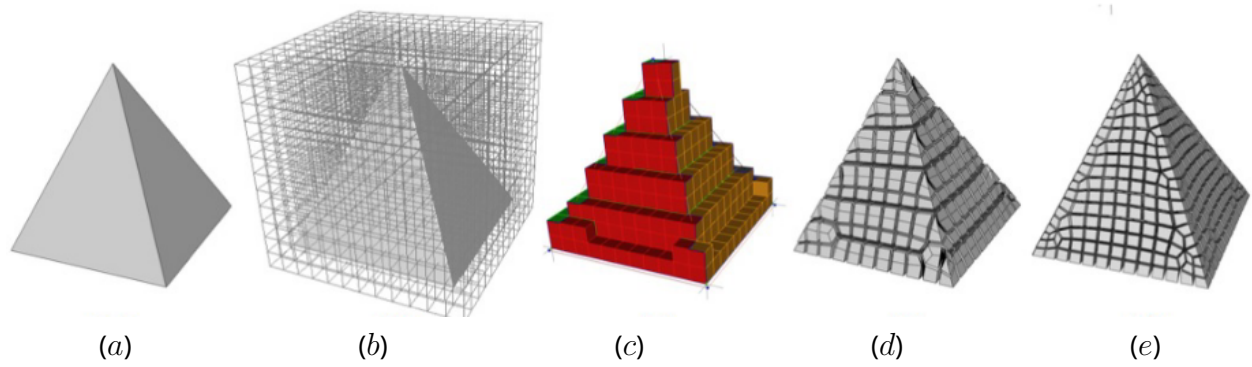


Figure 2.21 – Maillage par intersection de grille. La géométrie (a) est immergée dans une grille (b) pour ne conserver que les cellules de la grille intersectant la géométrie (c); Le maillage ainsi obtenu est déformé pour discrétiser la géométrie (d) avant d'être topologiquement modifié par insertion de couches pour améliorer la qualité des mailles au bord (e).

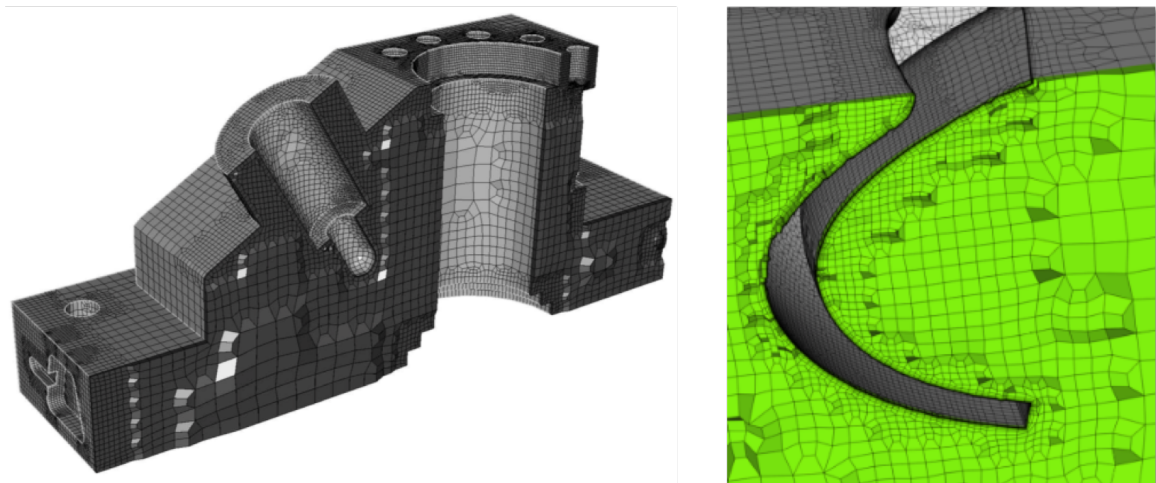


Figure 2.22 – Maillage par intersection de grille avec raffinement initial pour capturer les différents éléments géométriques [Mar09].

Les méthodes à base d'intersection de grilles sont les seules à être automatiques pour n'importe quel modèle géométrique. Elles sont donc aussi les seules à être proposées dans des logiciels professionnels [Dis20, Cor21, Lab] pour tout type de géométrie.

2.2.1 . Approches par objet médian

Comme leur nom l'indique, les approches par objet médian se basent sur l'utilisation de l'objet médian, ou axe médian, pour obtenir une structuration de l'objet découpable en blocs. L'objet médian d'un domaine Ω est défini par l'ensemble des points p de Ω tels qu'une sphère centrée en p touche au moins deux points distincts du bord $\partial\Omega$. L'objet médian obtenu est un "squelette" formé de surfaces dont les points sont équidistants de deux points du bord. Bien que l'utilisation de cet objet médian pour le maillage hexaédrique puisse différer selon les approches [PAS95, PA97b], on notera les points communs suivants :

1. L'objet médian va permettre d'obtenir une décomposition de Ω en sous-domaines ;
2. Ces sous-domaines peuvent ensuite être discrétisés en utilisant des méthodes simples (subdivision par point médian [PA97a] ou balayage [Qua14] par exemple) ;
3. Dans la plupart des cas, l'objet médian va être utilisé pour générer de nouveaux éléments topologiques afin de guider les algorithmes de génération de maillages hexaédriques (la génération de couloirs [Qua14] (voir Figure 2.23)).

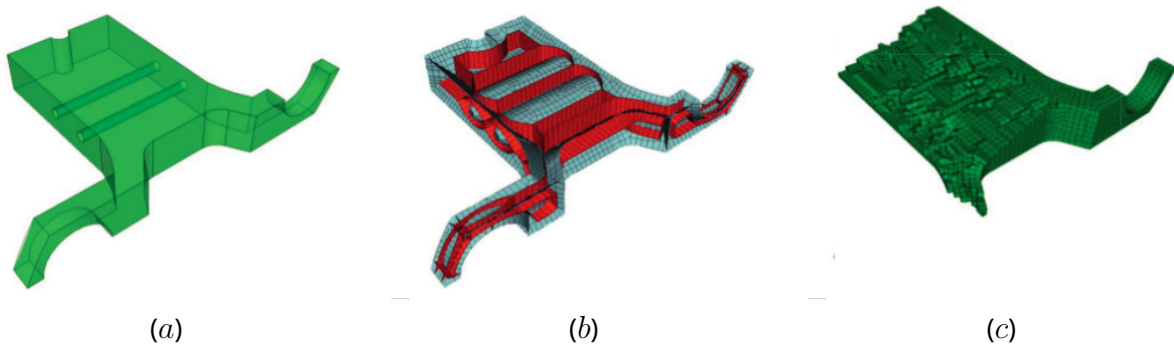


Figure 2.23 – Maillage par objet médian. A partir de la géométrie (a) un objet médian (en gris) va être généré, des couloirs (en rouge) vont être obtenus en projetant sur le bord de la géométrie les jonctions, c'est-à-dire les parties non-variétés de l'objet médian (c). L'objet médian et les couloirs sont maillés en 2D ce qui va permettre de guider une méthode de maillage par avancée de front pour générer un maillage hexaédrique dominant final (c).

Méthode	Alignement au bord	Qualité géométrique	Structuration	Contrôle de la taille
Axe médian	Oui	Difficile à contrôler	Pas toujours la structuration souhaitée	Oui

2.2.2 . Approches par polycubes

De façon imagée, les approches basées sur les polycubes consistent à déformer le domaine Ω dans un espace paramétrique $(\pm X, \pm Y, \pm Z)$ afin d'obtenir un polyèdre orthogonal, ou polycube, qui peut ensuite être discrétisé plus simplement avec une grille hexaédrique régulière. Ces méthodes reposent sur l'utilisation d'un maillage triangulaire T_Ω de Ω et sont dans la plupart du temps décomposées selon les étapes suivantes (voir Figure 2.24 [PRR+20]) :

1. La première étape est appelée **flagging** ou **assignation**. Les triangles formant le bord de T_Ω se voient assignés à un des axes ($\pm X, \pm Y, \pm Z$). Un ensemble de triangles formant une surface connexe et assignés à un même axe forment un *chart*, qui correspondra à une face polygonale du polycube généré. Pour un triangle t , l'axe choisi n'est pas nécessairement celui le plus proche de la normale sortante à t . C'est le fruit de plusieurs paramètres ayant pour but d'obtenir un polycube compact (peu de charts) et fidèle (proche géométriquement de la forme de Ω).
2. Une fois cette assignation effectuée, le maillage T_Ω est déformé (passage de la figure 2.24-*a* à 2.24-*b*). Cette déformation doit être effectuée en évitant de transformer la topologie du volume Ω . En outre, une étape dite de **quantization** est le plus souvent opérée pour que les charts soient sur des valeurs entières de (X, Y, Z) ;
3. Le polycube est alors discrétisé par une grille régulière en extrayant les points de coordonnées entières dans le domaine. On a des sommets parfaitement placés sur les coins, arêtes et faces au bord grâce à la quantization précédemment réalisée.
4. Le maillage obtenu est alors transporté sur le domaine initial via la déformation inverse et on obtient le maillage souhaité de Ω .

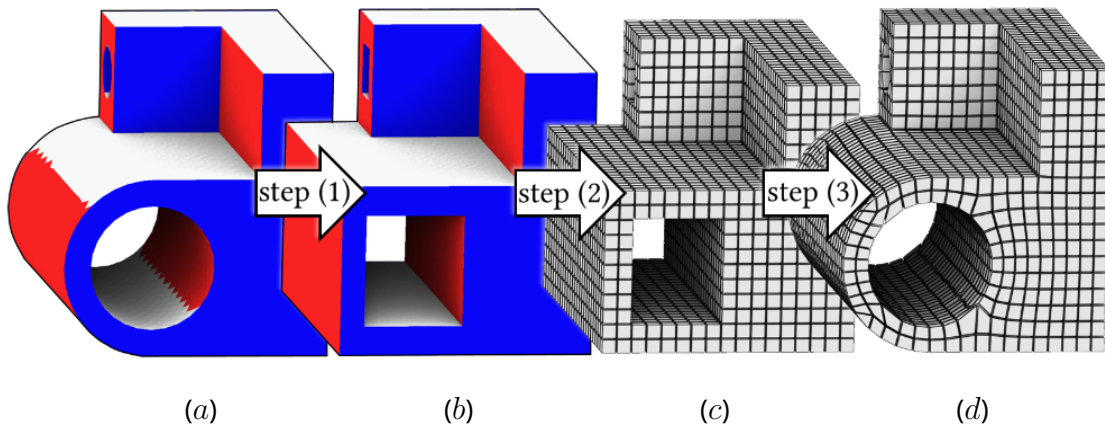


Figure 2.24 – Maillage par polycube. La géométrie (a) est déformée pour être alignée dans un espace paramétrique $\vec{u}, \vec{v}, \vec{w}$ (b). Le polycube obtenu est ensuite discrétisé par une grille régulière d'hexaèdres (c), la déformation inverse est appliquée à la grille pour générer le maillage final de la géométrie de base (d).

Les méthodes par polycube fonctionnent sur un grand nombre de domaines 3D mais fournissent intrinsèquement de mauvais résultats, ou pas de résultat du tout, dès que des zones de l'objet ont des angles inférieurs à 45 degrés. C'est par exemple le cas montré à la figure 2.25. On pourra remarquer que la même problématique est rencontrée avec les approches par champs d'orientation (voir introduction et la section suivante). Parmi les autres défauts importants, on notera la sensibilité du résultat par rapport à l'orientation choisie pour le polycube et l'absence de singularité interne. Par contre, il est possible d'optimiser un maillage de type polycube pour ajouter ensuite des couches de mailles supplémentaires et ainsi améliorer la qualité du résultat obtenu [CAS⁺19]. Notons qu'il existe de nombreux travaux traitant de la génération de polycube [THCM04, GSZ11, LVS⁺13, CLS16, HJS⁺14, FXBH16]. C'est aujourd'hui l'une des approches les plus prometteuses pour générer des maillages de qualité souhaitée, c'est-à-dire structurés en grilles et alignés au bord si l'on ajoute une étape d'insertion de couches de mailles au bord en post-traitement.

Méthode	Alignement au bord	Qualité géométrique	Structuration	Contrôle de la taille
Polycube	Non	Bonne qualité sauf sur le bord de la géométrie	Oui	Oui si la grille est raffinée en conséquence

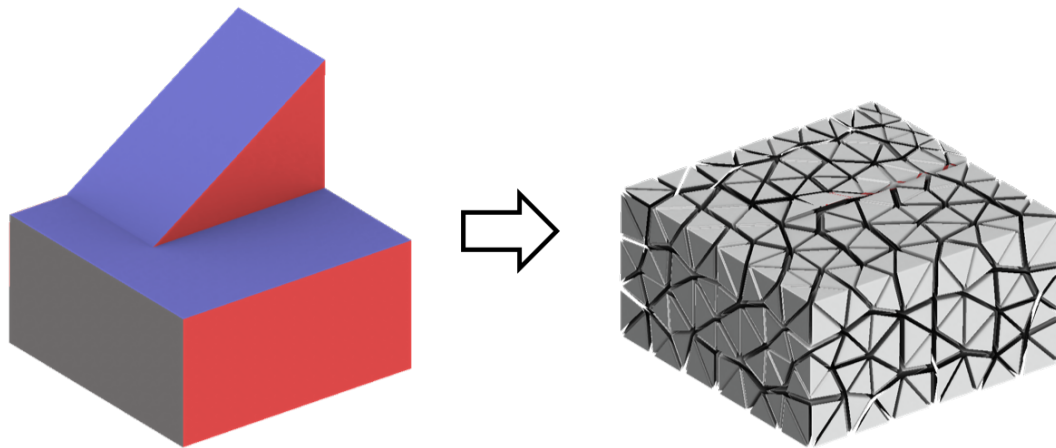


Figure 2.25 – Tentative de maillage par polycube d'une géométrie avec une zone de tremplin. Toute la surface de couleur bleue, sur le dessus de l'objet (à gauche) est assignée au même chart ce qui induit un écrasement de l'objet dans l'espace paramétrique (à droite).

2.2.3 . Approches par champs d'orientation

Les approches par champs d'orientation reposent sur la génération de trois champs vectoriels corrélés entre eux et qui définissent localement l'orientation d'un cube dans un domaine Ω [HTWB11, KLF16, NRP11, LBK16, PBS20, RSL16]. De manière imagée, une orientation est la donnée de trois directions deux à deux orthogonales. En définissant un champ d'orientation sur le domaine Ω tout entier, que nous notons \mathcal{F}_Ω , l'objectif est de disposer d'une information de directions pour créer un hexaèdre en chaque point de Ω et ce de manière cohérente et globale :

1. Les orientations définies en deux points proches du domaine doivent être quasiment alignées entre elles ; On dit que le champ doit être **localement lisse** ;
2. Les orientations définies sur le bord du domaine doivent être **alignées au bord** avec les vecteurs normaux au bord.

Les différents travaux basés sur l'utilisation de champ d'orientation ont tous pour objectif de générer un champs le plus lisse possible sous la contrainte de respecter les alignements au bord. Les approches diffèrent alors sur le type de schémas numériques, sur la fonctionnelle exacte à optimiser, sur la façon de respecter les contraintes de bord (reprojection ou relaxation lagrangienne), etc.

Une fois le champ créé, on va là aussi trouver des approches différentes. Si les travaux de [KLF16] tentent d'extraire la structure de blocs comme illustré à la figure 2.26, la plupart des autres travaux cherchent à générer un atlas de paramétrisations qui recouvrent globalement le domaine Ω pour extraire le maillage final directement à partir de cette paramétrisation. Dans les deux cas, il est aujourd'hui très difficile de garantir l'obtention d'un résultat valide. Le problème fondamental commun aux deux approches est que pour le moment, aucune méthode de génération de champ ne garantit l'obtention d'un champ compatible avec la structure topologique d'un maillage hexaédrique. Le second problème plus technique est d'assurer la robustesse des approches dans le voisinage des zones de singularités où le champ est mal défini et donc problématique à utiliser d'un point de vue numérique.

Méthode	Alignement au bord	Qualité géométrique	Structuration	Contrôle de la taille
Champs d'orientation	Oui	Bonne qualité	Structuré en bloc si champ d'orientations valide	Oui

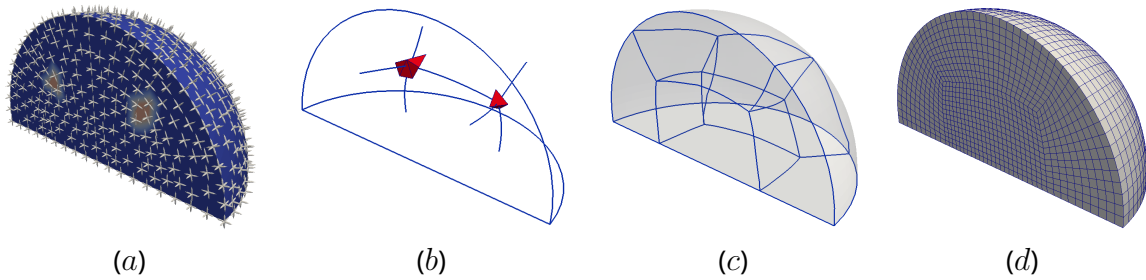


Figure 2.26 – Maillage par champ d'orientation. Un champ d'orientation \mathcal{F} est calculé sur les sommets d'un maillage tétraédrique qui discrétise la géométrie (a), le graphe de singularité est extrait de \mathcal{F} (b) ce qui permet de construire le squelette de la structure de blocs (c) et le maillage hexaédrique est généré en subdivisant la structure de blocs (d).

2.2.4 . Approches interactives

Puisque la génération automatique de maillages hexaédrique ne donne pas satisfaction, et ce spécialement dans le cas des maillages structurés par blocs, la plupart des maillages hexaédriques structurés par blocs utilisés en entreprise sont créés à l'aide de logiciels graphiques interactifs. Nous listons ci-après quelques outils commerciaux ou tout du moins non académiques utilisés en pratique avant de s'intéresser à des approches académiques liées à l'utilisation de champs d'orientation.

Logiciels industriels pour le maillage structuré par blocs

Les logiciels industriels permettant la génération de maillages structurés par blocs reposent principalement sur deux techniques :

1. La première technique consiste à **découper le domaine géométrique** en sous-domaines pour lesquels des algorithmes simples et connus de maillage hexaédrique peuvent s'appliquer. Cette approche nécessite de disposer d'opérations géométriques suffisamment robustes et n'est pas toujours adaptée selon la structure de blocs que l'on veut obtenir. Ainsi les opérations classiques utilisées consistent à couper le domaine à discrétiser par des plans, ou à prolonger des surfaces dans le volume, ce qui permet de facilement couper un objet de part en part en deux sous-parties. Il est par contre difficile de s'arrêter au milieu du domaine pour avoir des structures internes de blocs complexes. Dans cette catégorie, on dispose entre autres des logiciels Cubit [Lab] et Abaqus [SIM]. Avec le logiciel Cubit, la géométrie va être découpée jusqu'à ce que chaque sous-domaine puisse être maillé à l'aide d'une méthode de balayage (voir figure 2.27). Avec le logiciel Abaqus, le découpage est plus restrictif et nécessite d'obtenir des sous-domaines hexaédriques (voir figure 2.28).
2. La seconde technique consiste à **créer et manipuler une structure de blocs** sans jamais toucher à la géométrie. La structure en question est vue comme un macro-maillage dont les éléments sont classifiés sur le domaine géométrique qui, lui, reste inchangé (voir figure 2.29). Le logiciel ICEM-CFD Hexa [ANS] procède ainsi en considérant comme bloc d'origine la boîte englobante du domaine à discrétiser puis fournit de nombreuses opérations pour subdiviser ce premier bloc afin de discrétiser précisément le domaine Ω . Les blocs finalement obtenus sont hexaédriques et aisés à discrétiser à l'aide d'une grille. Dans le logiciel MGX développé au CEA, l'approche est similaire pour ce qui concerne la gestion des blocs et les opérations associées. Par contre l'approche ne se fait pas seulement par découpe mais aussi par assemblage et collage de blocs.

De manière générale, ces logiciels permettent d'avoir un contrôle important sur le maillage obtenu. Cependant, leur utilisation est relativement complexe et dédiée à un public averti d'ingénieurs experts.

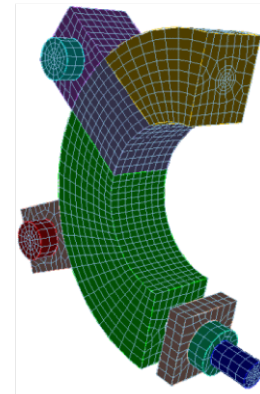
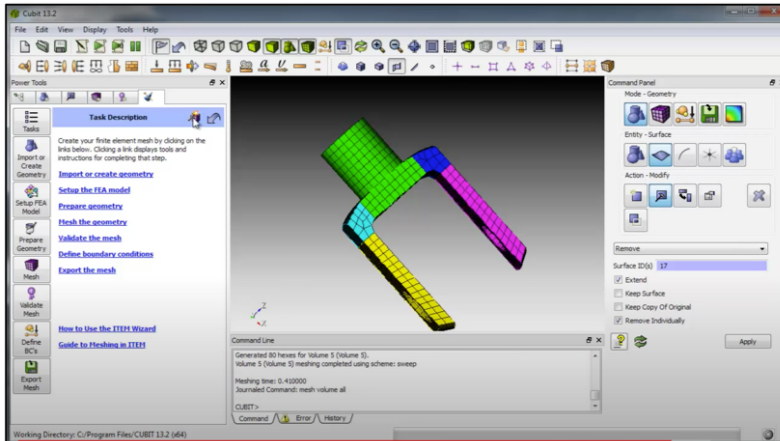


Figure 2.27 – Exemple du logiciel Cubit avec lequel un utilisateur découpe son modèle géométrique en morceaux pour lesquels un maillage par balayage est applicable.

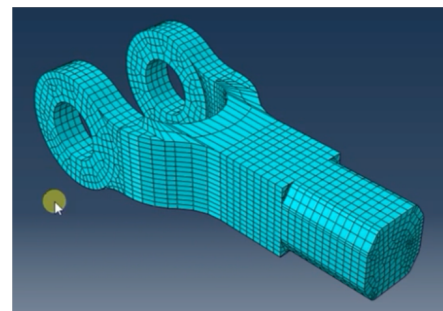
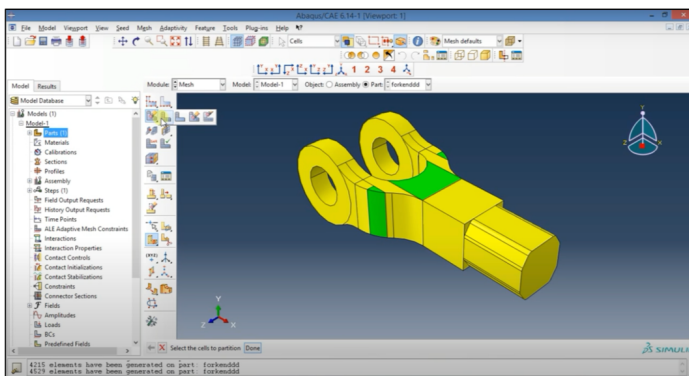


Figure 2.28 – Exemple du logiciel Abaqus avec lequel un utilisateur découpe son modèle géométrique en blocs simples qui seront ensuite discrétisés par des grilles.

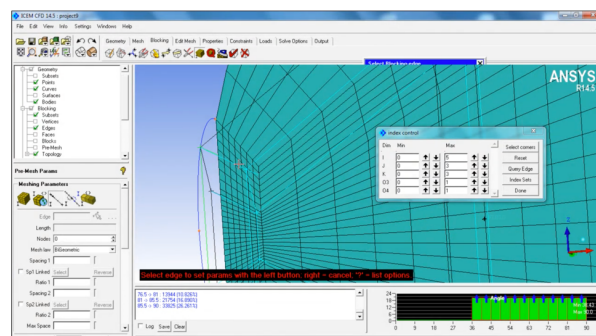
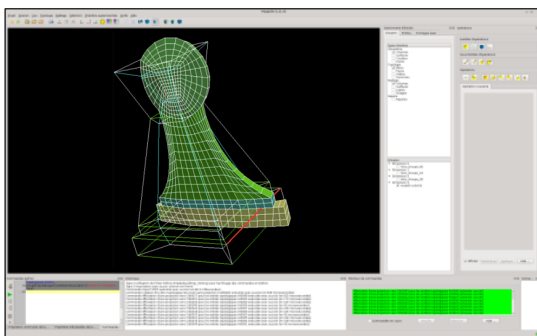


Figure 2.29 – Les logiciels **MGX** (à gauche) et **ICEM-CFD Hexa** (à droite) sont basés sur la construction d'une structure de blocs qui s'appuie sur un domaine géométrique sans le découper ou le modifier.

Générer un maillage de bonne qualité requière deux éléments importants : (1) du temps, qui peut aller de quelques heures à quelques semaines, et (2) un haut niveau d'expertise de l'utilisateur.

Approches semi-interactives

Les approches semi-interactives consistent à ajouter des interactions utilisateur à des méthodes de génération automatique de maillages hexaédriques afin de guider les algorithmes vers un résultat souhaité. Par exemple, les auteurs de [LZS⁺21] ajoutent un ensemble d'actions exécutables par l'utilisateur pour guider une approche de type *Polycube*. De leur côté, les travaux de [CBK12] se basent sur ce même type d'approche en utilisant les champs d'orientation (voir figure 2.30). Ils proposent la méthodologie suivante :

1. Un champ d'orientation surfacique est généré sur le bord du domaine (*a*) ;
2. De façon interactive, des courbes, qui correspondent à des cordes duales, sont tracées sur la surface en suivant le champ d'orientation (*b*) ;
3. A partir d'un nombre suffisant de courbes tracées, une structure de blocs quadrangulaires surfaciques va être obtenue à partir de cet ensemble de boucles duales (*d*) ;
4. Un maillage quadrangulaire surfacique est ensuite généré en exploitant la structure de blocs (*e*).

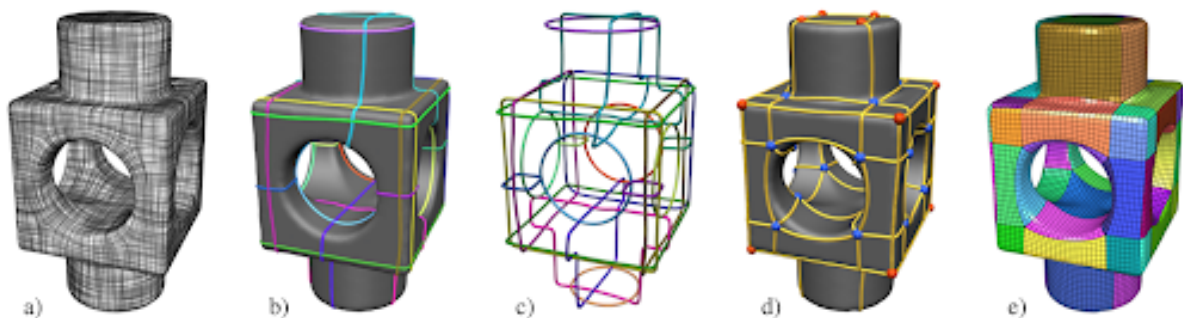


Figure 2.30 – Génération d'un maillage quadrangulaire surfacique à partir d'un champ d'orientation [CBK12]. Partant du champ représenté en (*a*) plusieurs courbes sont tracées pour partitionner la surface en (*b*) et (*c*). La structure duale est alors extraite en (*d*) pour obtenir finalement par raffinement le maillage final en (*e*).

Bien que l'approche proposée par [CBK12] se tourne vers la génération de maillages quadrangulaires surfaciques, elle reste un point de départ intéressant pour une méthode interactive de génération de maillages volumiques. D'ailleurs, les auteurs de [Tak19] proposent d'adapter cette méthode pour l'obtention d'un maillage volumique. De façon similaire, un champ d'orientation \mathcal{F}_Ω va être généré au bord du domaine Ω et ensuite l'utilisateur va créer une boucle duale au bord en suivant \mathcal{F}_Ω . De chaque boucle créée, une surface va être interpolée à l'intérieur du volume. Cette opération est répétée jusqu'à obtenir un ensemble de surfaces qui formeront la structure duale d'une structure de blocs hexaédriques.

2.3 . Positionnement de nos travaux

Nous récapitulons dans le tableau 2.1 les caractéristiques des différentes approches académiques pour le maillage hexaédrique. Au regard des propriétés des méthodes de maillage hexaédrique existantes, trois catégories sont potentiellement susceptibles d'être utilisées pour générer des maillages structurés par blocs : les méthodes basées sur l'objet médian, celle basées sur les polycubes et enfin celles utilisant des champs d'orientation. Dans ce travail, nous nous intéressons à l'extension de ces dernières d'un point de vue semi-automatique. Elles permettent d'obtenir des résultats qui correspondent aux critères recherchés (qualité et

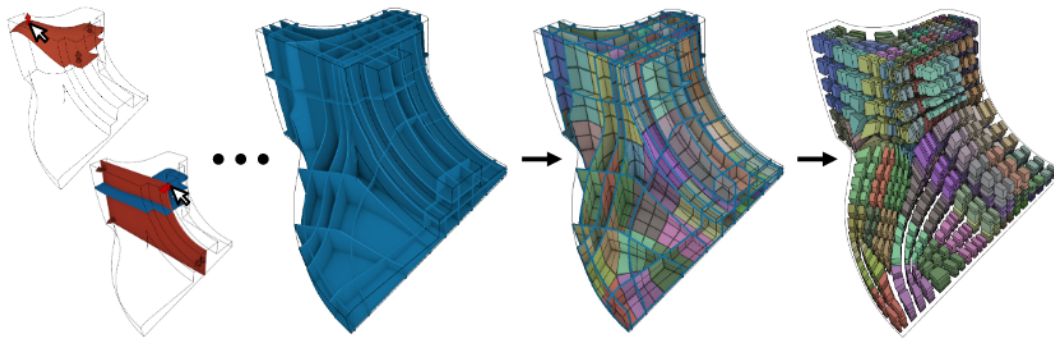


Figure 2.31 – Génération interactive d'une structure de blocs volumique par sélection de courbes duales sur le bord. L'hypothèse est faite que chaque courbe duale est le contour d'une surface duale qui est alors générée. L'ensemble de surfaces créé partitionne le domaine en blocs qui peuvent ensuite être discrétisés [Tak19].

alignement des mailles au bord, structure, etc...) mais ont le défaut de n'être réellement applicables que sur des domaines géométriques simples, la génération du champ d'orientations sur un domaine quelconque ne garantissant aucunement que le champ obtenu ait une structure compatible avec celle d'un maillage hexaédrique. C'est sur ce point que nous nous concentrons. Nous verrons en conclusion de ce manuscrit que différentes perspectives de nos travaux peuvent s'appliquer à l'extension des méthodes de type Polycube.

Méthode	Alignement au bord	Qualité géométrique	Structuration	Contrôle de la taille
Indirecte	Non	Non, la découpe de tétraèdres induit des angles diédraux trop faibles ou des maillages hybrides	Non	Oui
Intersection de grille	Oui si insertion terminale de feuillets	Difficile à contrôler	Non	Oui si la grille est raffinée en conséquence
Axe médian	Oui	Difficile à contrôler	Pas toujours la structuration souhaitée	Oui
Polycube	Non	Bonne qualité sauf sur le bord de la géométrie	Oui	Oui si la grille est raffinée en conséquence
Champs d'orientations	Oui	Bonne qualité	Structuré en bloc si champ d'orientation valide	Oui

Table 2.1 – Caractéristiques des principaux types d'algorithmes de maillage hexaédrique.

De façon plus précise, nos travaux se positionnent dans la continuité de [CBK12, Tak19] en considérant toutefois un champ d'orientation volumique 3D et non pas uniquement un champ d'orientation surfacique. Comme dit précédemment, les méthodes par champs d'orientation permettent d'obtenir des résultats qui correspondent aux critères recherchés mais ne sont applicables concrètement que sur des domaines géométriques simples. Cela est notamment le cas de deux configurations particulières : les **lignes singulières 3-5** et les **zones de tremplin** (ces configurations sont définies plus en détail à la section 4.1). La figure 2.32 illustre cela avec en (a) un modèle possédant deux *lignes singulières 3-5* en rouge et en (b) un modèle possédant

une *zone de tremplin* dont le bord est colorié en rouge. En considérant ces configurations, nous classons les domaines géométriques en trois catégories :

- Les domaines de **type V**, dits **valides**, ne possèdent aucune des deux configurations précédentes dans leur champ d'orientation ;
- Les domaines de **type S** (**S** pour **singularité**), possèdent au moins une *ligne singulière 3-5* dans leur champ d'orientations ;
- Les domaines de **type T** (**T** pour **tremplin**), possèdent au moins une *zone de tremplin* dans leur champ d'orientation.

Il est à noter que les deux dernières catégories ne sont pas exclusives. Pour un domaine Ω , un champ d'orientation \mathcal{F}_Ω peut contenir à la fois des *lignes singulières 3-5* ainsi que des *zones de tremplin*, nous dirons alors que le domaine Ω est de type **S** et **T**.

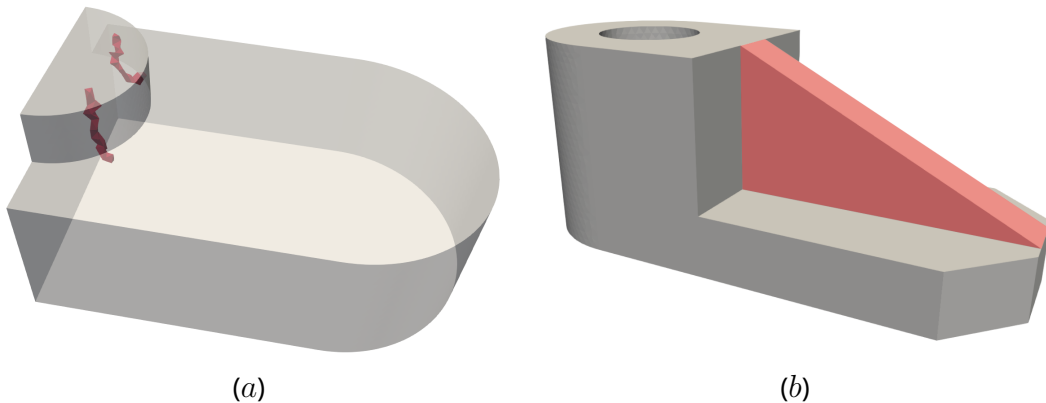


Figure 2.32 – Illustration des zones problématiques pour des champs d'orientation. En (a), deux lignes singulières 3-5 ne permettent pas de générer une structure de blocs ; en (b), la zone de tremplin rouge ne contient aucune singularité et devrait donc être discrétisée par un bloc hexaédrique unique, ce qui est problématique.

3 - Génération interactive de blocs par une approche duale

Comme mentionné au chapitre 2, la génération automatique de structures de blocs hexaédriques pour une géométrie quelconque est aujourd'hui un problème ouvert pour lequel aucune solution automatique n'existe. Dans ce travail, nous proposons de relaxer ce problème en considérant une approche interactive dans laquelle un ingénieur crée interactivement et incrémentalement une structure de blocs. Pour cela, nous nous appuyons sur deux choix majeurs :

1. **Nous construisons la représentation duale d'une structure de blocs**, à savoir un arrangement simple de surfaces [MBBM97], et non pas la structure de blocs directement. Nous considérons d'une part que le tracé interactif de surfaces dans un volume est une action aisée à appréhender interactivement et d'autre part que l'on peut établir un processus itératif durant lequel la structure de blocs peut toujours être validée à l'aide d'un système de règles.
2. **Nous nous appuyons sur des champs d'orientation pour disposer d'une information volumique afin de guider le traçage de surfaces en 3D**. Par analogie, ce champ d'orientation nous fournit une information similaire aux notions de courbures principales sur une surface 3D. Ceci démarque notre approche d'autres approches interactives [Tak19] ou automatiques [ZWG⁺18] qui considèrent uniquement l'information disponible sur le bord du domaine.

En considérant un domaine Ω de type V (voir la classification 2.3 proposée à la page 42) suffisamment discrétisé par un maillage tétraédrique T_Ω sur lequel vit un champ d'orientation \mathcal{F}_Ω défini aux sommets de T_Ω , notre objectif est de fournir un outil permettant de générer interactivement et itérativement le dual d'une structure de blocs hexaédriques. Une telle structure duale correspondant à un arrangement simple de surfaces, la méthode proposée consiste à ce que l'utilisateur insère au fur et à mesure un ensemble de surfaces qui viendront partitionner T_Ω , et donc par extension Ω , en un ensemble de zones correspondant aux cellules duales d'une structure hexaédrique. La figure 3.1 illustre cette création de surfaces sous une représentation dépendant de T_Ω . À partir du maillage proposé en (a), l'utilisateur crée interactivement des surfaces duales en (b) qui partitionnent T_Ω en régions duales en (c) qui nous permettent de générer une structure de blocs en (d).

Dans ce chapitre, nous exposons les différents ingrédients essentiels de ce pipeline. Tout d'abord, à la section 3.1.1, nous expliquons comment sont modélisés les champs d'orientation définis de manière discrète aux sommets de T_Ω et nous introduisons les concepts associés. Ensuite, nous abordons les trois étapes essentielles de notre méthode, à savoir le tracé de surface duale à la section 3.2, la validation d'un arrangement de surfaces à la section 3.3 et la transformation d'un arrangement de surfaces en une structure de blocs à la section 3.4. L'exemple de la figure 3.2 illustre les différentes étapes de notre approche. Nous partons d'une géométrie simple, ici un huitième de sphère discrétisé par un maillage tétraédrique T_Ω sur lequel nous avons calculé un champ d'orientation \mathcal{F}_Ω visible sur la figure 3.2-a. Ce sont les données d'entrée de

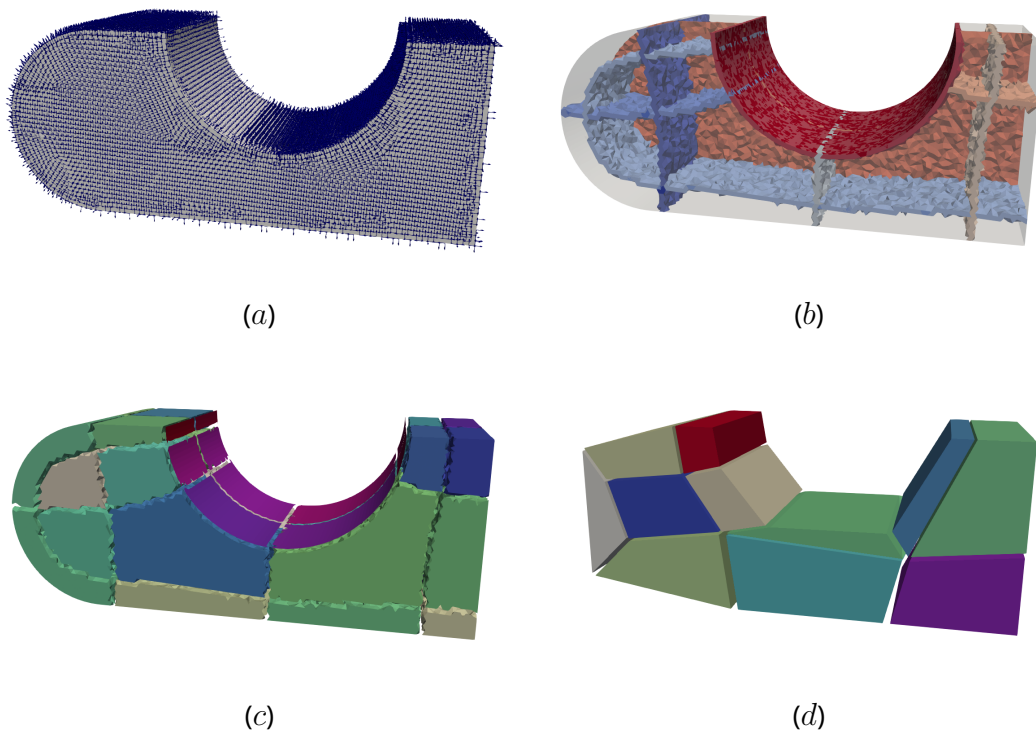


Figure 3.1 – Un exemple d'exécution de notre méthode de création de structure de blocs. En (a), le maillage tétraédrique avec un champ d'orientation assigné aux sommets. En (b), un ensemble de surfaces duales créées interactivement. En (c), les régions duales générées à partir de ces surfaces duales. Finalement en (d), la structure de blocs obtenue.

notre approche et nous partons du principe que leur génération est hors du cadre de notre travail. Néanmoins afin d'appréhender correctement la méthode de génération des champs d'orientations, nous indiquons à la Section 3.1 comment nous générons en pratique un tel champ en s'appuyant sur les travaux de [RSL16]. Nous allons ensuite entrer dans la phase de création interactive de la structure duale comme illustré sur la figure 3.2-*b*. Lors de cette phase, l'utilisateur va insérer, via des commandes interactives, des surfaces duales dans le volume qui suivent le champ d'orientation. Pour cela, l'utilisateur n'a besoin que de choisir un point de départ et une direction de coupe et notre algorithme crée automatiquement la coupe dans le volume. Cette phase est détaillée à la Section 3.2. Une fois les surfaces insérées nous passons à l'étape de vérification de la validité de la structure duale construite. Comme celle-ci est construite par l'utilisateur, nous avons besoin de vérifier qu'elle va permettre, a terme, de générer une structure de blocs. Nous avons donc défini un ensemble de règles que nous appliquons à la représentation discrète de notre structure duale soutenue par le maillage T_Ω , qui nous permet d'extraire des informations supplémentaires sur les régions duales, comme notamment leur classification géométrique. L'image de gauche de la figure 3.2-*c* nous montre la classification obtenue pour les régions duales. Le rond noir, le triangle bleu ciel et le carré rouge représentent respectivement une classification sur un sommet géométrique, une courbe et une surface géométrique. Puis, si la structure duale vérifie les règles données, nous allons utiliser les informations obtenues pour extraire la topologie de la structure de blocs (voir figure 3.2-*d*). En utilisant les relations qui lient structures primales et duales, nous allons d'abord associer les sommets de blocs par groupes de huit et ensuite retrouver le voisinage topologique de chaque sommet, afin de générer les blocs hexaédriques finaux.

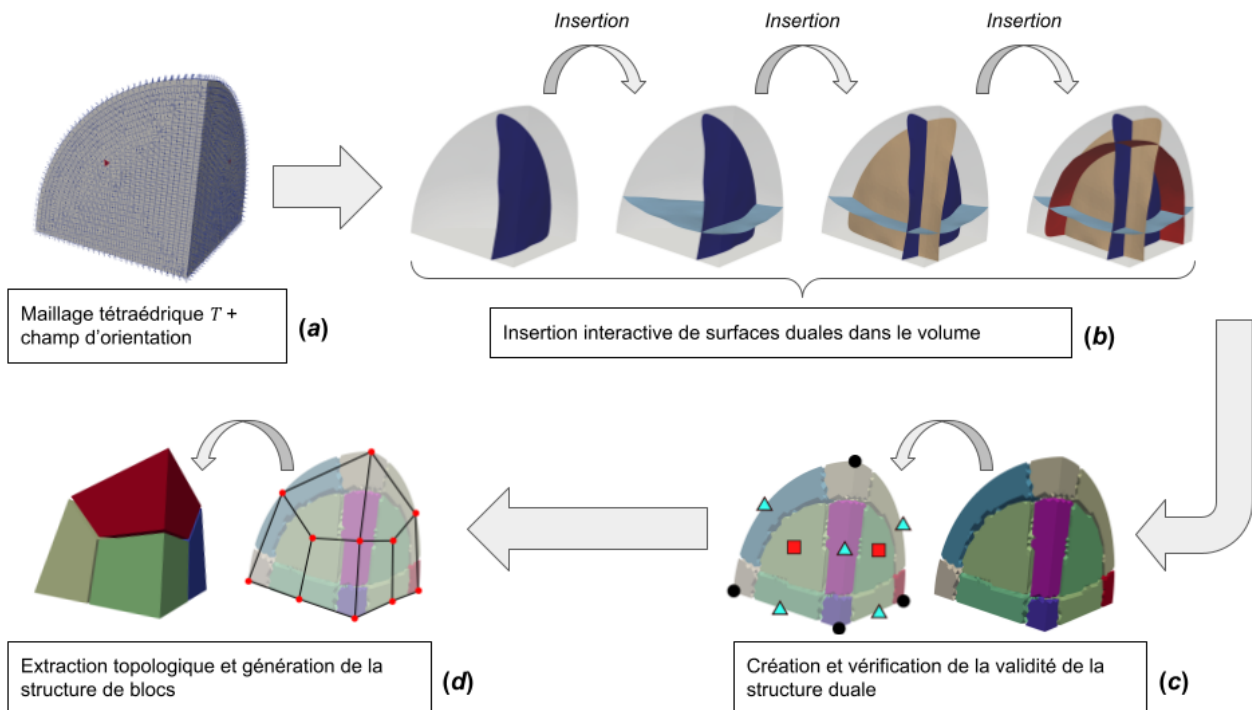


Figure 3.2 – Pipeline de notre approche interactive pour générer une structure de blocs pour des configurations de type V.

En résumé

Dans ce chapitre, nous proposons une méthode interactive de génération de blocs hexaédriques tirant bénéfice de la structure duale de tels maillages et de la relation entre maillages hexaédriques et champs d'orientation 3D. Partant d'un domaine Ω suffisamment discrétisé par un maillage tétraédrique T_Ω sur lequel vit un champ d'orientation \mathcal{F}_Ω défini aux sommets de T_Ω , notre approche repose sur :

- Le tracé de surfaces duales dans Ω suivant le champ dans le volume en suivant le champ d'orientation \mathcal{F}_Ω ;
- La définition d'un système de règles de validité de la structure duale générée ;
- La génération d'une structure valide de blocs hexaédriques pour Ω à partir du dual précédemment créé.

Cette approche est applicable pour les domaines géométriques de type V.

3.1 . Champs d'orientation pour le maillage hexaédrique

3.1.1 . Définition et représentation des champs d'orientations

Nous considérons dans ce travail l'utilisation des champs d'orientations définis de manière discrète aux sommets d'un maillage tétraédrique T_Ω qui discrétise le domaine Ω . En d'autres termes, une orientation est associée à chaque sommet du maillage tétraédrique. Nous introduisons ci-après une représentation fonctionnelle de ces orientations à l'aide d'harmoniques sphériques et indiquons comment ces champs sont calculés dans notre travail. En l'occurrence, le choix effectué correspond aux travaux de [RSL16] et nous reportons ici les éléments nécessaires par souci de complétude.

Représentation fonctionnelle des orientations

Une définition "intuitive" des orientations 3D est de les considérer comme des repères locaux, i.e. des bases directes orthonormées de \mathbb{R}^3 . Une orientation \mathcal{F} sera alors la donnée de trois vecteurs unitaires $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$, tels que $\mathbf{u} \cdot \mathbf{v} = 0$ et $\mathbf{w} = \mathbf{u} \wedge \mathbf{v}$.

Les orientations 3D sont naturellement connectées à la structure d'un hexaèdre, noté h , qui dispose lui aussi de trois directions principales $\{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3\}$, qui connectent ses faces opposées. Dans le cas d'un cube idéal d'arêtes de longueur 1, ces trois directions correspondront exactement à la définition d'orientations 3D. La seule distinction mais qui est très importante est que si l'on considère les vecteurs $\{\pm\mathbf{d}_1, \pm\mathbf{d}_2, \pm\mathbf{d}_3\}$, il existe en fait 24 orientations possibles pour représenter l'hexaèdre h (dû aux rotations à $\pi/2$ selon les 3 axes). Cet ensemble de rotations correspond au groupe de symétrie \mathcal{G} qui est justement invariant par les rotations de $\pi/2$ sur les trois axes. Étant donnée une orientation \mathcal{F} , on désigne l'ensemble des rotations équivalentes par la classe $[\mathcal{F}]$ et on note $[\mathcal{R}]$ la classe de référence définie à partir de l'orientation $\mathcal{R} = \{\mathbf{ox}, \mathbf{oy}, \mathbf{oz}\}$, avec $\mathbf{ox} = (1, 0, 0)$, $\mathbf{oy} = (0, 1, 0)$ et $\mathbf{oz} = (0, 0, 1)$.

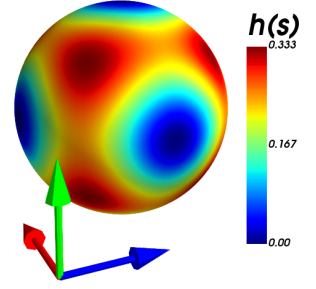
Malheureusement, une telle représentation est difficile à utiliser pour réaliser des calculs géométriques ou algébriques. Dans ce travail, nous adoptons la représentation fonctionnelle initialement introduite par [HTWB11] et reprise dans [RSL16]. Étant donnée une orientation $\mathcal{F} = \{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$, Huang et al. [HTWB11] introduisent la fonction $h : S^2 = \{s \in \mathbb{R}^3 \mid \|s\| = 1\} \rightarrow \mathbb{R}$ définie par

$$h(s) = s_x^2 s_y^2 + s_x^2 s_z^2 + s_y^2 s_z^2. \quad (3.1)$$

Il existe alors un unique vecteur $\mathbf{v}_r \in \{\pm\mathbf{ox}, \pm\mathbf{oy}, \pm\mathbf{oz}\}$ telle qu'une rotation de $\pi/2$ autour de \mathbf{v}_r transporte l'orientation de référence \mathcal{R} sur \mathcal{F} . Si on note \mathbf{M}_r la matrice de rotation correspondante, on peut montrer que :

$$h(\mathbf{M}_r s) = s_x^2 s_y^2 + s_x^2 s_z^2 + s_y^2 s_z^2 = h(s), \quad \forall s \in S^2.$$

La fonction h est différentiable d'ordre 2 et atteint des valeurs maximales pour les directions alignées avec les huit coins du cube unitaire centré à l'origine. Les valeurs minimales sont elles obtenues pour les directions qui pointent vers le centre des faces du cube (voir la figure incrustée à droite). Cette caractéristique de la fonction h s'étend à n'importe quelle orientation par changement de bases. Du coup, pour n'importe quelle orientation \mathcal{F} , on peut définir la fonction $h_{[\mathcal{F}]} = h(\mathbf{M}s)$ où \mathbf{M} est la matrice de rotation transportant \mathcal{R} sur \mathcal{F} . Une mesure de similitude entre deux orientations \mathcal{F}_1 et \mathcal{F}_2 peut alors être définie comme la norme L^2 entre les deux fonctions correspondantes :



$$d(\mathcal{F}_1, \mathcal{F}_2) = \int_{S^2} (h_{[\mathcal{F}_1]}(s) - h_{[\mathcal{F}_2]}(s))^2 ds \quad [\text{HTWB11}]. \quad (3.2)$$

Avec cette mesure, on a $d(\mathcal{F}_1, \mathcal{F}_2) = 0, \forall \mathcal{F}_1 \in [\mathcal{F}_2]$. D'un point de vue calculatoire, la fonction $h_{[\mathcal{F}]}$ peut être décomposée dans la base des harmoniques sphériques pour gagner en efficacité. Dans cette base, et sachant que l'on s'intéresse aux rotations uniquement, la fonction $h_{[\mathcal{R}]}$ peut être écrite comme suit :

$$h_{[\mathcal{R}]} = \sqrt{\frac{7}{12}} Y_4^0 + \sqrt{\frac{5}{12}} Y_4^4, \quad (3.3)$$

avec Y_ℓ^m l'harmonique sphérique réelle de degré ℓ et d'ordre m . Les harmoniques sphériques ont la propriété que la rotation d'harmoniques sphériques de degré ℓ produisent d'autres harmoniques sphériques de degré ℓ . Du coup, puisque que toute orientation \mathcal{F} peut être obtenue en appliquant des rotations de l'orientation de référence \mathcal{R} et que $h_{[\mathcal{R}]}$ est une combinaison linéaire d'harmoniques sphériques de degré 4, il en résulte que la représentation fonctionnelle de n'importe quelle orientation peut être décrite dans la base des harmoniques sphériques de degré 4 et donc comme combinaison linéaire des 9 harmoniques sphériques réelles de degré 4 $B = (Y_4^{-4}, Y_4^{-3}, \dots, Y_4^0, \dots, Y_4^4)$, c'est-à-dire

$$h_{[\mathbf{R}]} = B\tilde{\mathbf{r}}, \text{ avec } \tilde{\mathbf{r}} = (0, 0, 0, 0, \sqrt{\frac{7}{12}}, 0, 0, 0, \sqrt{\frac{5}{12}})^\top.$$

Chaque fonction $h_{[\mathcal{F}]}$ peut alors être représentée par

$$h_{[\mathcal{F}]} = B\tilde{\mathbf{f}} \text{ avec } \tilde{\mathbf{f}} = \mathbf{R}_B\tilde{\mathbf{r}},$$

où \mathbf{R}_B est une matrice de rotation 9×9 qui correspond à la matrice de rotation transformant \mathcal{R} en \mathcal{F} . Les vecteurs de coordonnées $\tilde{\mathbf{r}}$ et $\tilde{\mathbf{f}}$ sont appelés les **vecteurs de représentations** de l'orientation.

Génération d'un champ d'orientations 3D lisse

L'objectif de l'étape de génération d'un champ d'orientations pour un domaine géométrique Ω quelconque est de disposer d'un champ discret défini aux sommets de T_Ω , qui soit lisse et tel que les orientations définies sur le bord de Ω soient alignées localement avec les normales au bord. Le champ obtenu sera alors extrapolé linéairement dans chaque tétraèdre de T_Ω . Le caractère lisse du champ d'orientation peut être obtenu en minimisant l'équation (3.2) le long de chaque arête de T_Ω . Pour ne pas tenir compte uniquement du voisinage topologique, la mesure globale du caractère lisse prend en compte les longueurs des arêtes à l'aide de poids, dits cotangents, qui correspondent à des termes de la matrice de rigidité \mathbf{L} usuellement utilisée pour la méthode des éléments finis \mathbb{P}_1 (tétraédrique d'ordre 1). On minimise donc la quantité suivante :

$$E_s = \sum_{ij \in T_\Omega^1} \mathbf{L}_{ij} \int_{S^2} (h_i(s) - h_j(s))^2 ds \quad (3.4)$$

où \mathbf{L}_{ij} est le poids cotangent associé à l'arête ij . En, décomposant respectivement $h_i(s)$ et $h_j(s)$ dans la base B des harmoniques sphériques de degré 4, on obtient :

$$E_s = \sum_{ij \in T_\Omega^1} \mathbf{L}_{ij} \int_{S^2} (B\mathbf{a}_i - B\mathbf{a}_j)^2 ds \quad (3.5)$$

avec \mathbf{a}_i et \mathbf{a}_j les vecteurs de représentation des orientations associées aux sommets d'indice i et j . On peut alors écrire

$$E_s = \sum_{ij \in T_\Omega^1} \mathbf{L}_{ij} (\mathbf{a}_j - \mathbf{a}_i)^\top \left(\int_{S^2} B^2 d\alpha \right) (\mathbf{a}_j - \mathbf{a}_i). \quad (3.6)$$

Enfin, puisque la base de fonctions B est une base de la sphère unité S^2 , on peut simplifier l'expression pour obtenir :

$$E_s = \sum_{ij \in T_\Omega^1} \mathbf{L}_{ij} \|\mathbf{a}_j - \mathbf{a}_i\|^2. \quad (3.7)$$

Il est importante de noter que l'expression finale de E_s implique uniquement des vecteurs de dimension 9. Par conséquent, une solution à ce problème ne garantit pas que le vecteur $B\mathbf{a}_i$ vive sur S^2 . En d'autres termes, ce vecteur n'a aucune raison d'être unitaire. Cela justifie pourquoi les auteurs de [RSL16] introduisent la **contrainte de faisabilité** (f_{c_i}) suivante pour toute orientation \mathcal{F}_i :

$$\forall i, \exists \mathbf{R}_i \text{ une matrice de rotation } 9 \times 9 \text{ telle que } \mathbf{a}_i = \mathbf{R}_i \tilde{\mathbf{r}}. \quad (f_{c_i})$$

Il reste à contraindre la méthode génération à produire un champ dont les orientations seront alignées avec le bord géométrique de Ω . Pour cela, deux types de contraintes sont définies selon la position du sommet sur le bords :

1. Si le sommet $v_i \in \partial\Omega$ est localisé sur une courbe, alors on a une contrainte **forte** à respecter. Les trois directions de l'orientation associée \mathcal{F}_i sont totalement définies en fonction de la tangente à la courbe et des normales aux surfaces voisines.
2. Si le sommet $v_i \in \partial\Omega$ est localisé sur une surface, alors on a une contrainte **faible** à respecter. Une des trois directions de l'orientation associée \mathcal{F}_i doit être alignée avec la normale à ∂T_Ω au sommet v_i . Soit \mathbf{n}_i la normale en v_i et \mathbf{a}_i le vecteur de représentation de \mathcal{F}_i . En considérant que les matrices de rotation sont écrites sous la forme extrinsèque XYZ, Ray et al. [RSL16] distinguent deux cas :
 - Le vecteur \mathbf{n}_i est aligné avec l'axe z . La matrice de rotation transformant $\tilde{\mathbf{r}}$ en \mathbf{a}_i est une rotation d'axe z et d'angle θ . On a :

$$\tilde{\mathbf{f}} = \mathbf{R}_B^z(\theta) \tilde{\mathbf{r}} = \sqrt{\frac{7}{12}} \delta_9^5 + \mathbf{c}_v[0] \delta_9^9 + \mathbf{c}_v[1] \delta_9^1 \quad (3.8)$$

avec $\mathbf{c}_v = \left(\sqrt{\frac{5}{12}} \cos(4\theta), \sqrt{\frac{5}{12}} \sin(4\theta) \right)$ et δ_n^i le vecteur de dimension n dont la i^{me} composante vaut 1 et toutes les autres valent 0. Autrement dit, toute orientation associée à un point au bord dont la normale est alignée selon l'axe z doit vérifier :

$$\|\mathbf{c}_v\|^2 = 5/12. \quad (3.9)$$

- Si \mathbf{n}_i n'est pas aligné avec l'axe z , le vecteur \mathbf{n}_v est tourné pour être aligné avec z et se ramener à la première configuration. Cela nécessite de calculer une matrice de rotation 9D \mathbf{R}_v^B (voir [RSL16]) et on obtient la contrainte

$$\mathbf{a}_v = \sqrt{\frac{7}{12}} \mathbf{R}_v^B \delta_9^5 + \mathbf{c}[0] \mathbf{R}_v^B \delta_9^9 + \mathbf{c}[1] \mathbf{R}_v^B \delta_9^1. \quad (s_{c_i})$$

En résumé, chaque contrainte faible en un sommet du bord v_i introduit une variable intermédiaire c_i , 9 équations linéaires via l'équation (s_{c_i}), et la contrainte quadratique $\|\mathbf{c}_v\|^2 = 5/12$.

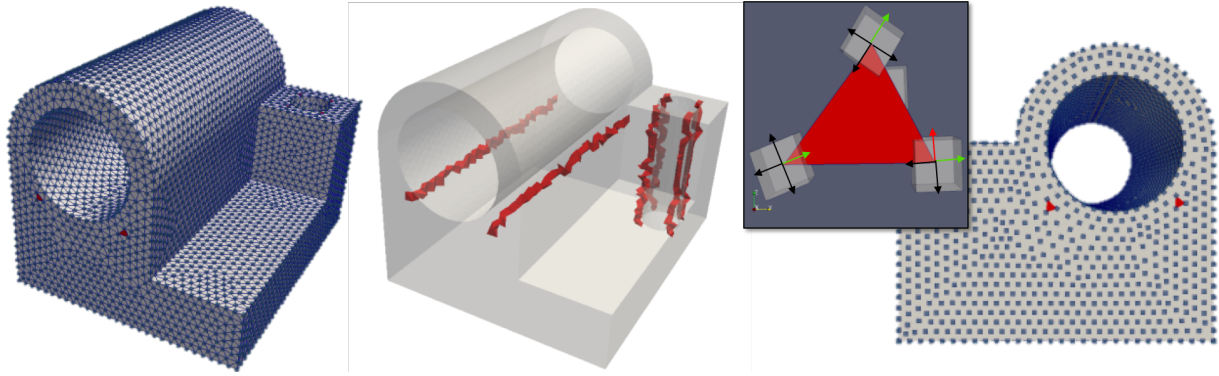


Figure 3.3 – Génération d'un champ d'orientations discrets. De gauche à droite : une vue 3D globale d'un domaine où chaque orientation est modélisé par un cube; des lignes de tétraèdres singuliers; un zoom sur un tétraèdre singulier traversé par une unique ligne singulière; enfin la vue 2D d'une des surfaces du domaine géométrique.

3.1.2 . Résolution du problème d'optimisation par relaxation Lagrangienne

Nous notons \mathbf{K} les indices des sommets du maillages localisés sur des courbes du bord, \mathbf{J} ceux des sommets localisés sur des surfaces du bord, et \mathbf{I} les autres. On peut alors traduire le problème de minimisation sous la forme :

$$\min \sum_{ij \in T_{\Omega}^1} \mathbf{L}_{ij} \|\mathbf{a}_j - \mathbf{a}_i\|^2 \quad (3.10)$$

sous les contraintes que $\forall i \in \mathbf{I} + \mathbf{J}, f_{c_i}$ est vérifiée et que $\forall j \in \mathbf{J}, sc_j$ est vérifié. Nous simplifions la résolution de ce problème en faisant plusieurs hypothèses. Tout d'abord, comme [RSL16], nous commençons par négliger la contrainte de faisabilité pour obtenir une solution relaxée à notre problème, puis nous projetons chaque orientation obtenue sur une solution faisable (cela revient à projeter les vecteurs de représentation obtenus sur la sphère unité 9D). La seconde hypothèse consiste à simplifier les contraintes faibles au bord : pour tout $j \in \mathbf{J}$ on a :

$$\exists \mathbf{c}_j \in R^2, \mathbf{a}_j = \sqrt{7/12} \mathbf{R}_v^B \delta_9^5 + \mathbf{c}_j[0] \mathbf{R}_v^B \delta_9^9 + \mathbf{c}_j[1] \mathbf{R}_v^B \delta_9^1. \quad (3.11)$$

Quand a_j sera projeté sur une solution faisable, la condition $\|\mathbf{c}_j\|^2 = 5/12$ sera automatiquement vérifiée. La nouvelle contrainte (3.11) peut être réécrite sous la forme $a_j - b = 0$, pour tout $j \in \mathbf{J}$. Nous appliquons alors le principe de relaxation Lagrangienne en introduisant la contrainte faible au bord dans la fonction à minimiser avec λ un terme de pénalité¹ :

$$\sum_{ij \in T_{\Omega}^1} \mathbf{L}_{ij} \|\mathbf{a}_j - \mathbf{a}_i\|^2 + \lambda \sum_{j \in \mathbf{J}} \|\mathbf{a}_j - \sqrt{7/12} \mathbf{R}_v^B \delta_9^5 - \mathbf{c}_j[0] \mathbf{R}_v^B \delta_9^9 - \mathbf{c}_j[1] \mathbf{R}_v^B \delta_9^1\|^2. \quad (3.12)$$

Un exemple de champ d'orientation obtenu en résolvant ce problème est illustré sur la figure 3.3. La plupart des tétraèdres le composant sont qualifiés de **réguliers** et quelques uns, en rouge sur la figure, sont qualifiés de **singuliers**.

3.1.3 . Structure d'un champ d'orientations 3D

Le champs d'orientation interpolé dans un tétraèdre régulier ne contient aucune singularité et on pourrait y "plaquer" une grille régulière d'hexaèdres alignés avec les orientations définies à ces quatre sommets. Au contraire, un tétraèdre singulier contient une ou plusieurs lignes singulières le traversant. On notera $SG(T_{\Omega})$ l'ensemble des tétraèdres singuliers d'un maillage tétraédrique T_{Ω} , et on l'appelle **graphe singulier**.

1. λ est fixé à 100 dans notre outil.

L'ensemble $SG(T_\Omega)$ contient des tétraèdres singuliers **stables dans une direction** et des tétraèdres singuliers **instables**. Les premiers sont traversés par une unique **ligne singulière** (les tétraèdres bleus sur la figure 3.4), tandis que les seconds contiennent un **point singulier volumique** du graphe, c'est-à-dire le point de rencontre de plusieurs lignes singulières (les tétraèdres rouges et blancs sur la figure 3.4). On dénotera par **point singulier surfacique** le point de rencontre entre une ligne singulière et la surface bordant Ω . Par abus de langage, si nous ne précisons pas, un point singulier sera un point singulier volumique.

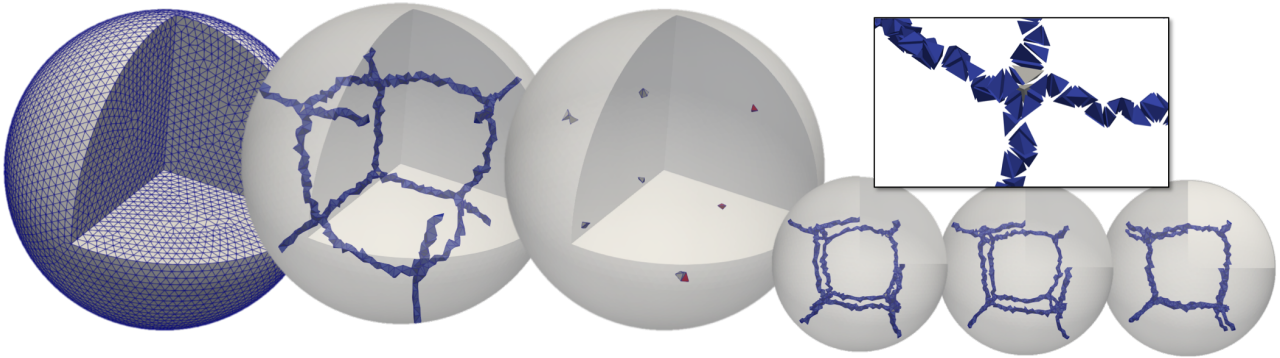


Figure 3.4 – *Graphe singulier d'un champ d'orientation discret. De la gauche vers la droite : le maillage tétraédrique initial; l'ensemble des tétraèdres intervenant dans le graphe ce qui nous donne la structure complète, les agglomérats de tétraèdre singuliers instables localisés aux points de rencontre des lignes singulières; un zoom sur l'un de ces points de rencontre et 3 vues en X, Y, Z, montrant les non-symétries du champ lié au calcul numérique effectué.*

Déterminer si un tétraèdre est régulier ou singulier se fait en pratique en inspectant les rotations minimales qui existent entre les orientations assignées aux sommets d'un tétraèdre de (v_i, v_j, v_k, v_ℓ) . Plus précisément, on inspecte chacune des faces du tétraèdre et on regarde les rotations associées. Considérons l'exemple de la figure 3.5 et plus précisément la face identifiée par les sommets d'indices 1, 2 et 3. Pour chaque orientation définie en ses sommets, nous commençons par identifier la direction la plus proche de la normale à la face (1,2,3). Une fois cette direction "éliminée", il nous reste des croix 2D en chacun des sommets 1, 2 et 3. Partant du sommet 1, on sélectionne le vecteur rouge. On passe ensuite au sommet 2 et on recherche parmi ses 4 vecteurs associés celui qui est aligné avec ce vecteur, en l'occurrence celui en vert. On fait de même entre les sommets 2 et 3 pour obtenir le vecteur vert au sommet 3 puis entre les sommets 3 et 1. On obtient alors le vecteur vert qui est différent du vecteur rouge d'où nous étions partis. On en déduit l'existence d'une singularité dans la face. Un tétraèdre singulier stable n'a que deux faces irrégulières tandis qu'un tétraèdre singulier instable peut en avoir trois ou quatre.

Notons qu'une fois une singularité détectée dans une face, on peut déterminer sa valence, qui vaudra nécessairement 3 ou 5 (voir figure 3.6-b et c). Pour cela, on recherche le long de chaque arête $[a, b]$ les points \mathbf{p}_i tels que le vecteur $s\mathbf{p}_i$, avec s la localisation de la singularité, soit colinéaire avec l'un des 4 vecteurs de la croix interpolée en \mathbf{p}_i à partir des valeurs en \mathbf{a} et \mathbf{b} .

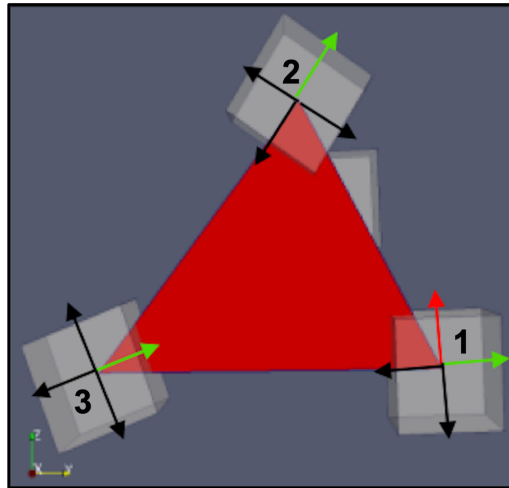


Figure 3.5 – Exemple d'un tétraèdre singulier dont la face d'indices 1, 2 et 3 est singulière.

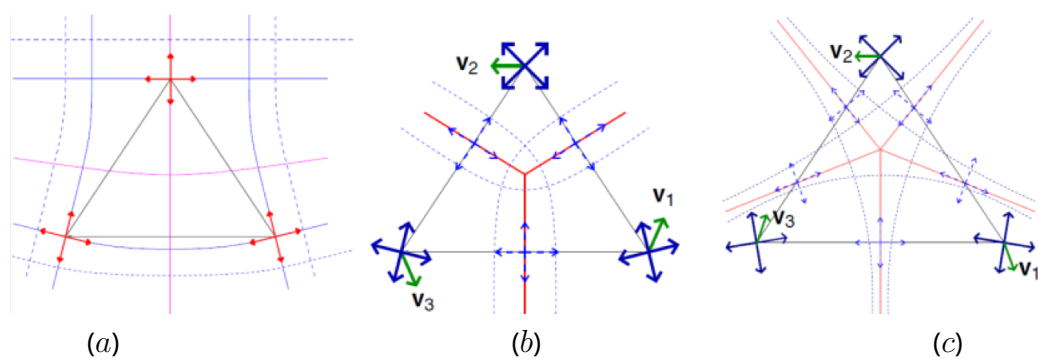


Figure 3.6 – Triangles réguliers et singuliers. En (a), les orientations aux trois sommets sont compatibles, le triangle est régulier. En (b) et (c), les triangles sont irréguliers et contiennent respectivement un point singulier de valence 3 et 5.

3.2 . Construction d'une structure duale

A partir d'un domaine Ω et de la donnée d'une discrétisation T_Ω suffisamment fine munie d'un champ d'orientation \mathcal{F}_Ω , nous souhaitons créer de façon interactive et itérative un ensemble de surfaces qui correspondront à la structure duale d'une structure de blocs hexaédriques. D'un point de vue topologique, cet ensemble doit vérifier la propriété de former un arrangement simple de surfaces [MBBM97], c'est-à-dire qu'au plus trois surfaces s'intersectent en un point de Ω . D'un point de vue géométrique, il est nécessaire que cet arrangement de surfaces partitionne Ω en un ensemble de zones duales d'une structure de blocs correctement classifiée sur Ω (voir section 2.3, page 42).

L'action principale de ce processus est la création d'une surface duale. Répétée un certain nombre de fois, nous obtiendrons un arrangement de surfaces qui partitionneront les tétraèdres de T_Ω en différentes zones. Il restera alors à valider que ce partitionnement correspond bien au dual d'une structure de blocs hexaédriques, ce qui sera l'objet de la section 3.3. Pour ce qui concerne la création d'une surface duale, étant donné le couple $(T_\Omega, \mathcal{F}_\Omega)$, nous proposons l'approche suivante :

1. L'utilisateur sélectionne un tétraèdre $t \in T_\Omega$. Puis nous interpolons linéairement au centre de t une orientation \mathcal{F}_t à partir des orientations définies au sommets de t . Il est à noter que cette interpolation n'est valide et donc acceptée que pour des tétraèdres réguliers. Au sein d'un tétraèdre singulier, le champ d'orientation est par définition mal défini et où nous ne cherchons pas à déterminer le résultat du procédé d'interpolation.
2. L'utilisateur a alors la possibilité de choisir l'une des trois directions de \mathcal{F}_t . Nous appelons d la direction retenue et elle définira la normale au centre de t de la surface duale que l'on veut générer au sein de T_Ω .
3. Avec la connaissance de (t, d) , nous allons construire un feuillet dual en partant de t . Ce procédé s'appuie sur deux concepts principaux : la propagation de la surface (voir la section 3.2.1) et l'utilisation de filtres de contrôle (voir la section 3.2.2). Notons que la création de la surface duale peut échouer si elle rencontre un tétraèdre singulier de $SG(T_\Omega)$ dans une configuration non-acceptable.
4. Si la propagation réussit et qu'elle satisfait les attentes de l'utilisateur, celui-ci peut répéter les étapes 1 à 3 jusqu'à l'obtention de l'ensemble de surfaces duales désirées.

Dans notre travail, nous appelons **feuillet dual** une représentation discrète d'une surface duale. Plus précisément un feuillet dual sera pour nous un ensemble connexe de tétraèdres de T_Ω qui partitionne T_Ω en deux sous-ensemble \mathbf{T}_1 et \mathbf{T}_2 tel qu'aucun tétraèdre de \mathbf{T}_1 ne possède une arête incidente à un tétraèdre de \mathbf{T}_2 .

3.2.1 . Construction d'un feuillet dual par propagation

Dans notre approche, nous proposons à l'utilisateur de créer un feuillet dual en sélectionnant interactive-ment les données d'entrée que sont un tétraèdre régulier, noté ici $t_0 \in T_\Omega$, et une direction \mathbf{d}_0 , sélectionnée parmi les trois directions de l'orientation \mathcal{F}_{t_0} interpolée en t_0 . Une fois ces données fournies, nous appliquons l'Algorithme 1. Cet algorithme extrait le feuillet en propageant une surface de coupe nommée \mathcal{S}_c dans T_Ω qui suit le champ d'orientation \mathcal{F} . La surface de coupe \mathcal{S}_c est définie localement à t_0 par le barycentre de t_0 , noté p_0 , et la direction \mathbf{d}_0 qui correspond à la normale à \mathcal{S}_c en p_0 . Partant de (p_0, \mathbf{d}_0) , on va propager \mathcal{S}_c en coupant successivement les tétraèdres rencontrés et en suivant au mieux le champ \mathcal{F} . Si on fait une analogie avec le parcours de structures de graphes, la coupe correspond à un parcours en largeur dans le maillage.

La coupe commence au sein du tétraèdre t_0 . Notons \mathcal{E}_{t_0} l'ensemble des triplets (e, \mathbf{n}, β) où e est une arête de t_0 intersectée par \mathcal{S}_c et \mathbf{n} un vecteur normal à \mathcal{S}_c au point $p \in e$, avec $p = (1 - \beta)e_0 + \beta e_1$, où (e_0, e_1) sont les extrémités de l'arête e et $\beta \in [0, 1]$. Les tétraèdres adjacents à une arête de $\mathcal{E}_{\mathcal{W}_i}$ sont les prochains tétraèdres à être coupés. A chaque étape i de la propagation de l'algorithme, ces tétraèdres

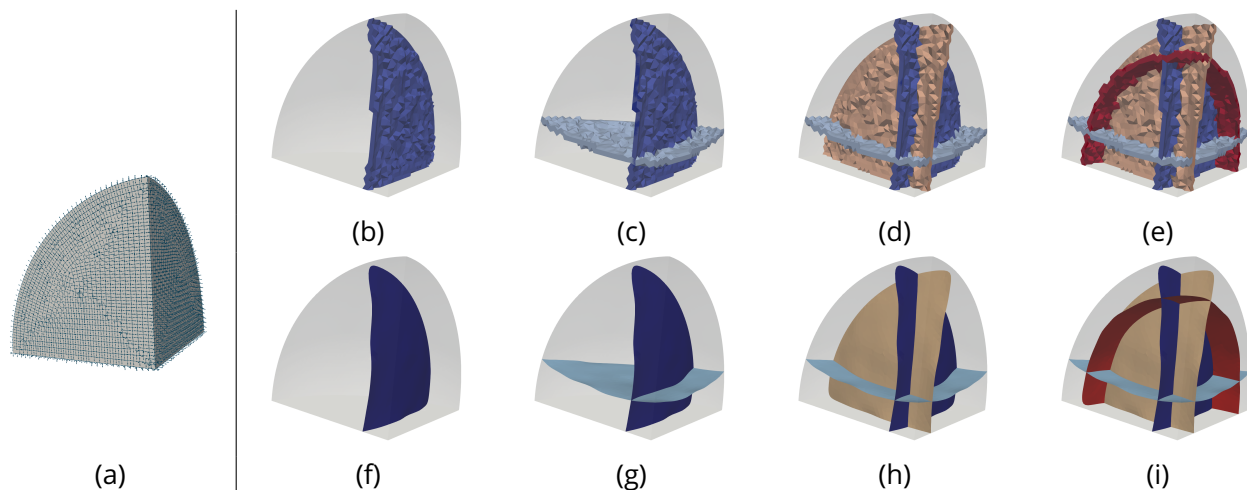


Figure 3.7 – Illustration de la création de la structure duale. En (a) les données d'entrées de notre algorithme à savoir un maillage T_Ω et le champ d'orientation discret au nœuds \mathcal{F} . En (b), (c) et (d), la structure duale après chaque itération de notre algorithme de création de feuillet duaux. Enfin en (e) la structure duale finale obtenue.

forment la vague \mathcal{W}_i . Ils vont être coupés selon les informations recueillies dans l'ensemble $\mathcal{E}_{\mathcal{W}_i}$. On répète ce procédé jusqu'à atteindre l'itération j où $\mathcal{W}_j = \emptyset$. La Figure 3.8 illustre les étapes de la création d'un feuillet avec en (a) le tétraèdre de départ t_0 , en (b) et (c) les différentes itérations de la propagation par vague et en (d) la surface de coupe ainsi que le feuillet dual extrait à partir de celle-ci.

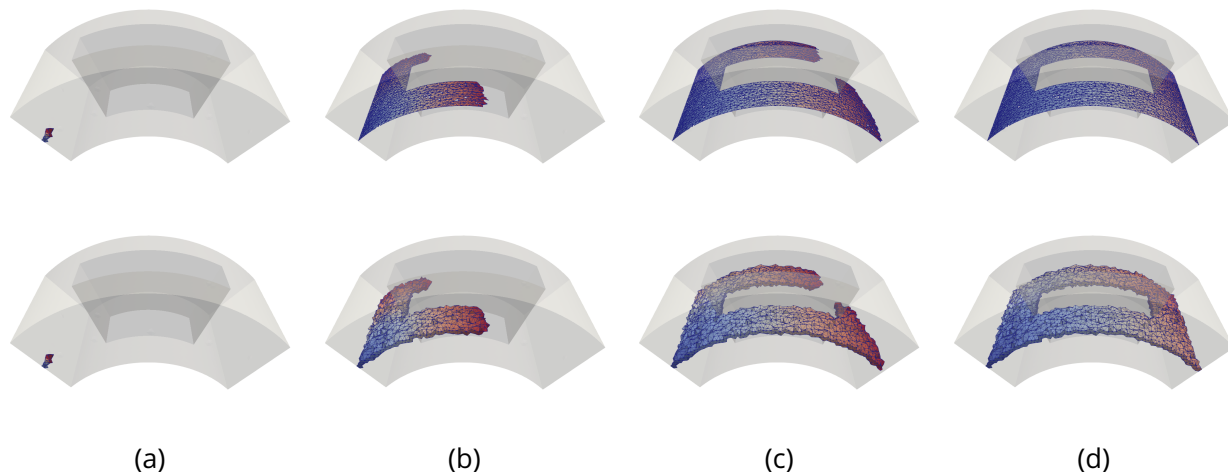


Figure 3.8 – Illustration de la création de feuillet dans un champ régulier : Sur la ligne du haut, une représentation surfacique de l'avancée du feuillet dual. En (a), on commence à partir d'un seul tétraèdre et ensuite on avance en (b) et (c) dans le domaine en suivant le champ d'orientation jusqu'à atteindre le bord du domaine dans toutes les directions de propagation en (d). La couleur d'un élément (un morceau de la surface ou un tétraèdre) est corrélée avec l'indice de la vague de propagation.

Initialisation de la propagation au sein du tétraèdre t_0

La première étape de l'Algorithme 1 consiste à créer le plan de coupe initial de S_0 dans t_0 . Pour ce faire, on définit l'opération $\text{closest_component}(\mathcal{F}_p, \mathbf{v})$, qui fournit le vecteur le plus proche de \mathbf{v} , dans

\mathcal{F}_p . Celui-ci est obtenu en calculant le produit scalaire entre \mathbf{u}_p , \mathbf{v}_p , \mathbf{w}_p et le vecteur \mathbf{v} , le vecteur le plus proche étant celui dont le produit scalaire avec \mathbf{v} est le plus proche de 1. Notons que tous les vecteurs sont bien unitaires dans nos calculs.

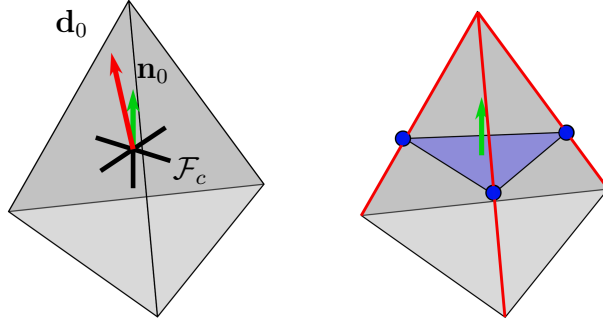
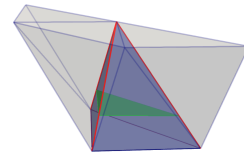


Figure 3.9 – Tétrahédre d'initialisation de la propagation, à gauche le vecteur \mathbf{n}_0 de \mathcal{F}_e est la direction la plus proche de \mathbf{d}_0 . A droite, le plan de coupe \mathcal{P}_0 et les arêtes intersectées en rouge.

On appelle p_0 le centre de masse des sommets de t_0 et \mathcal{F}_{c_0} l'orientation interpolée en p_0 à partir des orientations définies aux sommets de t_0 . On note $\mathbf{n}_0 = \text{closest_component}(\mathcal{F}_{c_0}, \mathbf{n}_0)$. Ensuite on crée le plan \mathcal{P}_0 qui passe par p_0 et qui a pour vecteur normal \mathbf{n}_0 (voir Figure 3.9). On a donc \mathcal{E}_{t_0} , l'ensemble des triplets (e, \mathbf{n}, β) , où e est une arête de t_0 intersectée par \mathcal{P}_0 au point p_{0_e} . Ici β représente les coordonnées barycentriques de p_{0_e} le long de l'arête e et \mathbf{n} est égal à $\text{closest_component}(\mathcal{F}\beta, \mathbf{n}_0)$, où $\mathcal{F}\beta$ est l'orientation interpolée à p_{0_e} à partir des orientations définies aux extrémités de e . L'interpolation des orientations est réalisée de façon linéaire en utilisant une représentation des orientations par des quaternions unitaires [KLF16].

Stratégie de propagation en largeur

Une fois l'algorithme initialisé, nous allons pouvoir créer le feuillet dual correspondant. Pour ce faire, nous propageons la surface de coupe \mathcal{S}_c au sein des tétraèdres de T_Ω en suivant le champ d'orientation \mathcal{F} . L'idée ici est de propager la surface par vagues successives de tétraèdres en utilisant une méthode de parcours en largeur en recalculant la normale à \mathcal{S}_c dans chaque tétraèdre rencontré. On débute le processus en récupérant les tétraèdres adjacents aux arêtes de \mathcal{E}_0 excepté t_0 . Ces tétraèdres forment la prochaine vague à traiter. On note $\mathcal{E}_{\mathcal{W}_0}$ toutes les arêtes qui ont été utilisées pour générer cette vague. Sur l'image de droite, le tétraèdre t_0 est représenté en foncé avec son plan de coupe en vert et les arêtes de \mathcal{E}_{t_0} sont colorées en rouge. Les tétraèdres de la première vague \mathcal{W}_0 générée à partir des arêtes de \mathcal{E}_{t_0} sont grisées dans cet exemple. Maintenant que nous avons défini la première vague \mathcal{W}_1 , on va propager \mathcal{S}_c , ce qui correspond aux lignes 5 à 20 de l'Algorithme 1. En premier lieu on définit $\mathcal{E}_{\mathcal{W}_{i+1}}$ l'ensemble des triplets d'intersection de \mathcal{S}_c dans les tétraèdres de \mathcal{W}_i . Par analogie, $\mathcal{E}_{\mathcal{W}_i}$ regroupe tous les triplets d'intersection qui ont permis d'obtenir la vague \mathcal{W}_i . Ensuite, on coupe chaque tétraèdre $t_k \in \mathcal{W}_i$, ce qui nous permet d'obtenir $\mathcal{E}_{\mathcal{W}_{i+1}}$, puis on récupère les tétraèdres non coupés adjacents aux arêtes de $\mathcal{E}_{\mathcal{W}_{i+1}}$ qui feront partie de la prochaine vague \mathcal{W}_{i+1} . Si aucun nouveau tétraèdre n'est découvert alors l'opération de création de feuillet est terminée.



La méthode de coupe utilisée durant la propagation est différente de celle utilisée pour t_0 . En effet pour t_0 , un plan local de coupe était créé au centre du tétraèdre avec pour vecteur normal la direction

Algorithme 1 : Création d'un feuillet dual

Data : (t_0, \mathbf{d}_0) **Result :** Tous les tétraèdres intersectés par la surface duale

```
1  $\mathcal{E}_{\{t_0\}} = \{(e, \mathbf{n}, \beta)\} \leftarrow \text{cutFirstTet}(t_0, \mathbf{d}_0);$ 
2  $\mathcal{W}_i \leftarrow \text{nexTets}(\mathcal{E}_{\{t_0\}}, \{t_0\});$ 
3  $\mathcal{E}_{\mathcal{W}_i} \leftarrow \mathcal{E}_{\{t_0\}};$ 
4  $\text{done} \leftarrow \text{false};$ 
5 while !done do
6    $\mathcal{E}_{\mathcal{W}_{i+1}} \leftarrow \emptyset;$ 
7   for  $t_k \in \mathcal{W}_i$  do
8      $\mathcal{E}_{\mathcal{W}_{i+1}} \leftarrow \mathcal{E}_{\mathcal{W}_{i+1}} \cup \text{cutTet}(t_k, \mathcal{E}_{\mathcal{W}_i});$ 
9   end
10   $\mathcal{E}_{\mathcal{W}_{i+1}} \leftarrow \text{applyControlFilters}(\mathcal{E}_{\mathcal{W}_{i+1}});$ 
11   $\mathcal{W}_{i+1} \leftarrow \text{nexTets}(\mathcal{E}_{\mathcal{W}_{i+1}}, \mathcal{W}_i);$ 
12   $\mathcal{E}_{\mathcal{W}_i} \leftarrow \mathcal{E}_{\mathcal{W}_{i+1}};$ 
13   $\mathcal{W}_i \leftarrow \mathcal{W}_{i+1};$ 
14  if  $\mathcal{W}_i = \emptyset$  then
15     $\text{done} \leftarrow \text{true};$ 
16  end
17  if reachSingularityWrong( $\mathcal{W}_i$ ) then
18     $\text{exit};$ 
19  end
20 end
```

\mathbf{n}_0 . Si on procède de manière similaire pour chaque tétraèdre t_k d'une vague \mathcal{W}_i , on obtiendra alors un ensemble de morceaux de surfaces non continues le long des arêtes partagées par des tétraèdres voisins. Si on regarde l'exemple de la Figure 3.10, créer un plan de coupe qui passe par le milieu des tétraèdre peut amener l'algorithme à sélectionner des tétraèdres supplémentaires et donc de propager \mathcal{S}_c dans une direction non voulue par l'utilisateur.

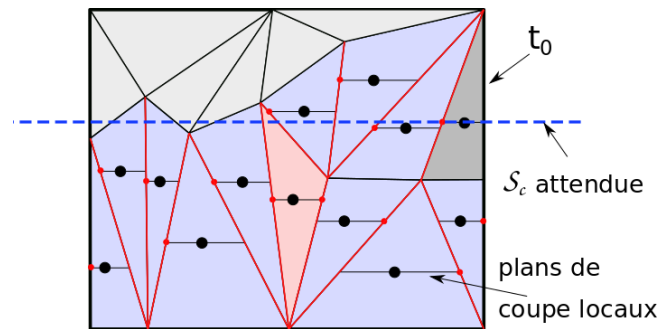
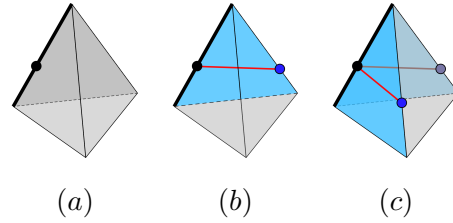


Figure 3.10 – Exemple 2D de propagation de coupe qui étant donnée les triangles intersectés peut s'éloigner fortement de la ligne attendue.

La propagation de la coupe se fait donc différemment et en l'occurrence, nous avons opté pour une propagation locale aux faces des tétraèdres formant une vague i . Cette méthode correspond à la fonction `cutTet` utilisée à la ligne 8 de l'Algorithme 1 et est détaillée dans l'Algorithme 2. Considérons un tétraèdre t_k à traiter. La propagation est faite le long des faces de t_k en partant des arêtes qui appartiennent à $\mathcal{E}_{\mathcal{W}_i}$.

Il est à noter qu'une face f d'un tétraèdre t_k peut avoir déjà été traitée pour le tétraèdre $t_{k'}$ qui partage f avec t_k , si $t_{k'}$ appartient à la vague actuelle ou à une vague précédente.

La propagation de la coupe dans un tétraèdre où une seule arête appartient à $\mathcal{E}_{\mathcal{W}_i}$ est illustrée sur l'image de droite avec en (a), l'état initial, en (b) la première face coupée et en (c) la deuxième face coupée. De façon générale, on commence l'algorithme de coupe dans t_k avec au moins une arête intersectée e sachant que $e \in \mathcal{E}_{\mathcal{W}_i}$ avec (e, \mathbf{n}, β) , on peut voir ces tétraèdres en gris transparent à la Figure 3.11 ainsi



que les points d'intersection en vert et les arêtes de $\mathcal{E}_{\mathcal{W}_i}$ en gras. Pour chaque face $f \in t_k$ incidente à e et non traitée, on calcule une approximation de la coupe dans le champ d'orientation en utilisant une méthode Runge-Kutta d'ordre 4. Cette étape correspond à la ligne 5 de l'Algorithme 2. Cela nous donne une nouvelle arête intersectée e' qu'on va ajouter à $\mathcal{E}_{\mathcal{W}_{i+1}}$ ainsi que les informations d'intersection (e', \mathbf{n}, β) . On peut voir à la Figure 3.11(b) les points d'intersection obtenus après l'étape de coupe des tétraèdres de la vague. La Figure 3.11(c) nous montre la surface, en vert, obtenue à la vague \mathcal{W}_i qui propage la surface de coupe \mathcal{S}_c .

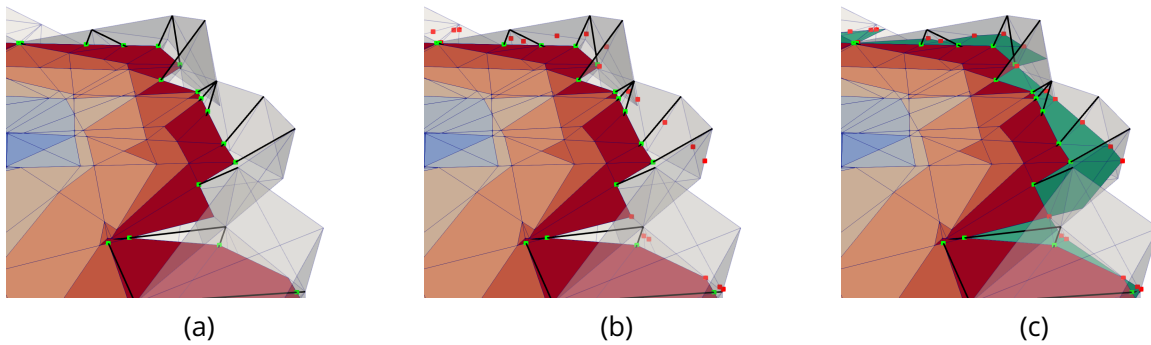


Figure 3.11 – Un exemple d'une itération de l'algorithme de propagation de la surface à la vague \mathcal{W}_i . On a en (a) les points d'intersections en vert ainsi que les arêtes de $\mathcal{E}_{\mathcal{W}_i}$ en gras. En (b), on voit en rouge les points d'intersection obtenus après la face de coupe des tétraèdres. Enfin, en (c), on constate l'évolution de la surface propagée.

Couper les faces indépendamment les unes des autres peut amener l'algorithme à des inconsistances dans \mathcal{S}_c à la fin d'une itération. En effet chaque face va produire un point d'intersection sur une nouvelle arête, arête qui peut appartenir à plusieurs faces à couper, et donc être coupée plusieurs fois. On ajoute donc des conditions pour gérer ces inconsistances. Notamment si deux arêtes d'une face f sont coupées alors que f n'a pas encore été traversée. C'est ici le cas à la Figure 3.12(a). La face représentée en gris clair possède deux arêtes déjà intersectées, une au point \mathbf{p}_x et l'autre au point \mathbf{p}_y . Dans cette situation propager \mathcal{S}_c par \mathbf{p}_x n'apporte pas de nouvelle arête intersectée et \mathbf{p}_y va propager \mathcal{S}_c dans la direction d'origine de \mathbf{p}_x . Pour éviter ce conflit on ajoute f à l'ensemble des faces traitées. De manière similaire, lors d'une itération de l'algorithme, une arête peut être intersectée par la coupe de deux faces différentes comme montré en (b). Pour éviter de dupliquer la surface de coupe on accepte un seul point d'intersection par arête.

Comme la propagation de la coupe est réalisée géométriquement en parcourant des faces du maillage T_Ω , il peut arriver que le point d'intersection de sortie sur une arête soit situé sur un sommet de T_Ω ². En pratique, nous évitons une telle configuration, en déplaçant légèrement de façon aléatoire les sommets intersectés de

2. ce qui équivaut à $\beta = 0$ ou 1 en coordonnées barycentrique le long de l'arête

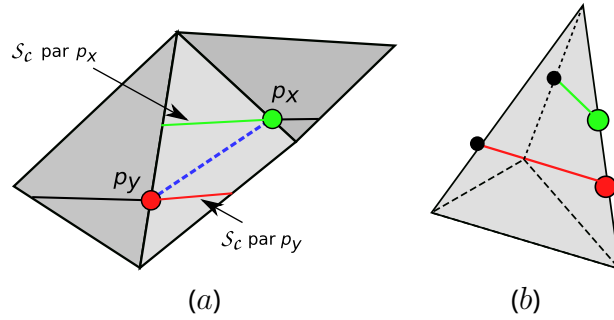


Figure 3.12 – Illustration de cas de propagation de coupe que nous voulons éviter. En (a), une face est intersectée deux fois le long d'arêtes différentes, une par p_x et une par p_y ce qui propagerait deux surfaces de coupe dans la face. En (b), une arête d'un tétraèdre est intersectée en deux points par des coupes qui proviennent de deux faces distinctes.

cette manière. Dans notre implémentation, nous conservons tous les sommets qui sont intersectés durant la vague \mathcal{W}_i dans l'ensemble \mathcal{N}_i et nous les déplaçons tous simultanément avec une perturbation aléatoire. Après cela, la coupe est recalculée dans les tétraèdres de \mathcal{W}_i qui sont adjacents à un sommet de \mathcal{N}_i . Ce procédé est répété jusqu'à obtenir $\mathcal{N}_i = \emptyset$. Dans nos différentes expérimentations, cet ensemble était toujours vide après un faible nombre d'itérations. On note aussi que dans notre implémentation, un sommet est considéré comme intersecté si le point d'intersection est situé à une distance inférieure ou égale à $l = \alpha \cdot \text{length}(e_{min})$ où e_{min} est la plus petite arête de T_ω et $\alpha = 0.1$.

Algorithme 2 : cutTet

Data : $(t_k, \mathcal{E}_{\mathcal{W}})$
Result : \mathcal{E}_{t_i}

```

1  $\mathcal{E}_{t_k} \leftarrow \emptyset;$ 
2 foreach edge  $e \in t_k$  do
3   if  $e \in \mathcal{E}_{\mathcal{W}}$  then
4     forall face  $f \in t_k \wedge e \in f$  do
5        $\mathcal{E}_{t_k} \leftarrow \mathcal{E}_{t_k} \cup \text{fieldApprox}(f);$ 
6     end
7   end
8 end
9 return  $\mathcal{E}_{t_i}$ 

```

L'algorithme va répéter ces étapes jusqu'à l'itération j où $\mathcal{W}_j = \emptyset$. Cela signifie que \mathcal{E}_j ne contient que des arêtes du bord du domaine et que tous les tétraèdres incidents à ces arêtes ont déjà été traités.

3.2.2 . Filtres de contrôle

Comme nous avons pu le décrire précédemment, notre procédé de création de feuillet dual consiste à propager une surface par vagues successives en effectuant des calculs d'intersection et en approximant une représentation de la surface de coupe localement à chaque tétraèdre. Chaque fois qu'un tétraèdre est traversé, des approximations numériques sont donc faites et de faibles erreurs géométriques peuvent apparaître. Alors qu'elles ont un faible impact localement, leur accumulation peut mener à des résultats inattendus. En particulier la surface de coupe peut se briser et se diviser (voir Figure 3.14). Afin d'éviter qu'une telle situation ne se produise, nous avons mis en place des filtres pour éliminer des éléments jugés indésirables de l'ensemble $\mathcal{E}_{\mathcal{W}_i}$ à la fin de chaque vague. Cette étape correspond à la ligne 10 de l'Algorithme 1.

Nous disposons concrètement de trois éléments de contrôle dans notre algorithme pour maîtriser la propagation de la surface. Le premier élément permet de contrôler si le champ d'orientation est stable dans la trajectoire du feuillet ; le second est un filtre basé sur la topologie de T_Ω ; enfin le dernier est un filtre qui prend en considération la géométrie du feuillet.

Traversée de lignes singulières

Durant la phase de propagation, il est possible de rencontrer des tétraèdres singuliers (ligne 17 de l'Algorithme 1), ce qui est problématique pour le calcul d'interpolation des orientations, celles-ci étant alors localement mal définies. Nous considérons en fait trois cas possibles lors de la rencontre de telles cellules :

1. Le tétraèdre singulier est instable, contient un point singulier volumique (voir page 50). On arrête alors la propagation du feuillet. Aucune direction n'est bien définie dans un tel tétraèdre et un feuillet ne peut le traverser.
2. Le tétraèdre singulier est stable et est traversé par une ligne singulière (voir page 50). A la différence d'un point singulier, une ligne singulière possède une direction stable dans laquelle le champ d'orientation peut être interpolé. Deux cas de figure sont alors possibles :
 - Soit la direction du vecteur normal local à \mathcal{S}_c est proche du vecteur tangent à la ligne singulière. On peut alors traverser le tétraèdre et la ligne singulière ;
 - Soit au contraire, la direction du vecteur normal local à \mathcal{S}_c est proche du vecteur tangent à la ligne singulière. On arrête alors le processus de propagation et le feuillet ne peut être créé.

La notion de proximité du vecteur normal local à \mathcal{S}_c avec le vecteur tangent de la ligne singulière - autrement dit sa direction stable - est définie de manière angulaire. En pratique, nous calculons le produit scalaire entre le vecteur de la direction stable et le vecteur normal local à \mathcal{S}_c obtenu lors de la vague précédente. Dans notre approche, on considère que ces directions sont alignées si l'angle déduit est inférieur à $\pi/4$.

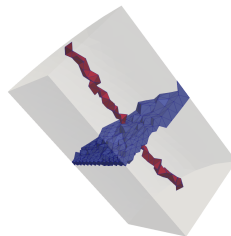


Figure 3.13 – Un exemple où le feuillet (en bleu) traverse une ligne singulière (en rouge) sans que celle-ci n'impacte la création du feuillet.

Filtre topologique

Durant la création du feuillet dual, on procède par vagues successives enrichissant ainsi progressivement l'ensemble des tétraèdres intersectés par la surface duale. Chacune de ces vagues est numérotée, et tous les tétraèdres qui appartiennent à une vague sont étiquetés avec le numéro en question. A titre indicatif, nous pouvons voir aux figures 3.8 et 3.11 les tétraèdres colorés selon l'indice de la vague.

Lors d'une vague i , le filtre topologique supprime les points d'intersection obtenus dans tous les tétraèdres de \mathcal{W}_i qui partagent une face ou une arête avec un tétraèdre qui appartient à la vague \mathcal{W}_j , avec $j \in [0, i - \lambda]$ et $\lambda \in \mathbb{N}^*$. Dans notre implémentation, nous fixons $\lambda = 2$. Ce filtre a un effet notamment lorsque la surface de coupe a besoin de se scinder et de se reconnecter. La Figure 3.14 illustre cela. Dans cet exemple la surface a du se scinder en deux le long du trou présent dans le domaine, les deux parties de la surface, en vert

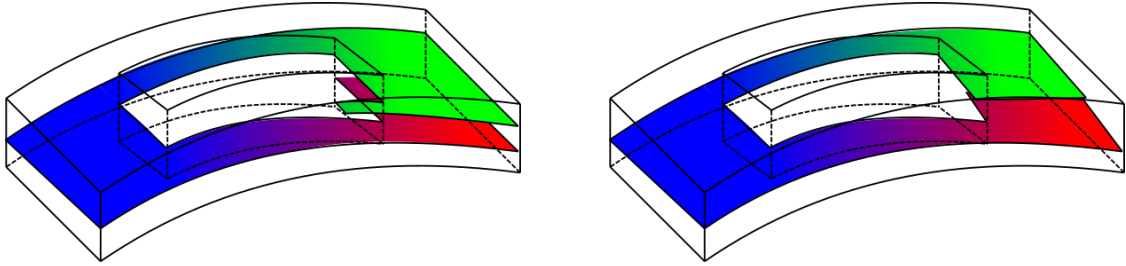


Figure 3.14 – *L'impact du filtre topologique sur un cas de séparation et de reconnection du feuillet dual. A gauche, le résultat sans le filtre, à droite l'application du filtre.*

et rouge, vont ensuite se reconnecter de l'autre côté. Cependant, à la suite d'approximations effectuées tout au long de la propagation, les deux parties de la surfaces ne se reconnectent pas, c'est ce qu'on voit sur l'image de gauche, où le filtre n'est pas appliqué. L'application du filtre nous permet d'obtenir le résultat à droite où la propagation de la surface est arrêtée. Les deux parties de la surface, bien que non reconnectées, intersectent des tétraèdres $\{t_i\}$ lors de l'itération i en contact avec des tétraèdres $\{t_j\}$ de vagues précédentes, les points d'intersection sont donc retirés de $\mathcal{E}_{\mathcal{W}_{i+1}}$. De cette manière, nous pouvons contenir une propagation qui se serait scindée en deux. Il est à noter que ce filtre est purement topologique et ne peut être appliqué que si les deux parties sont assez proche topologiquement pour être en contact.

Filtre géométrique

Le second filtre contrôle la propagation géométrique de chaque point dans $\mathcal{E}_{\mathcal{W}_{i+1}}$. Soit \mathbf{p}_i un tel point (voir Figure 3.15), on conserve pour ce point les deux points précédents utilisés pour le générer : le point \mathbf{p}_{i-1} qui appartient au même tétraèdre de \mathcal{W}_i que \mathbf{p}_i et le point \mathbf{p}_{i-2} qui appartient à un tétraèdre de la vague précédente \mathcal{W}_{i-1} . Nous allons ensuite vérifier la déviation entre la direction précédente \mathbf{d}_{i-1} et la direction actuelle \mathbf{d}_i . L'arête e est conservée dans $\mathcal{E}_{\mathcal{W}_{i+1}}$ si $\frac{\mathbf{d}_{i-1} \cdot \mathbf{d}_i}{\|\mathbf{d}_{i-1}\| \cdot \|\mathbf{d}_i\|} > \lambda_g$, avec $\lambda_g \in [0, 1]$. Le fait d'avoir $\lambda_g > 0$ empêche le feuillet de tourner de 90 degrés et de se retourner géométriquement dans le domaine. Ce filtre de contrôle est obligatoire près des lignes singulières où la forte courbure du champ d'orientation induit beaucoup plus de difficultés pour préserver la topologie de la surface duale. Nous pouvons voir à la Figure 3.16 un exemple d'application du filtre géométrique sur un feuillet qui passe entre des lignes singulières, augmenter la valeur de λ permet de réduire le nombre de points d'intersection et de réduire le nombre de tétraèdres dans le feuillet.

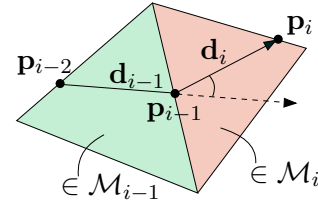


FIGURE 3.15 – Filtre géométrique basé sur les prédécesseurs.

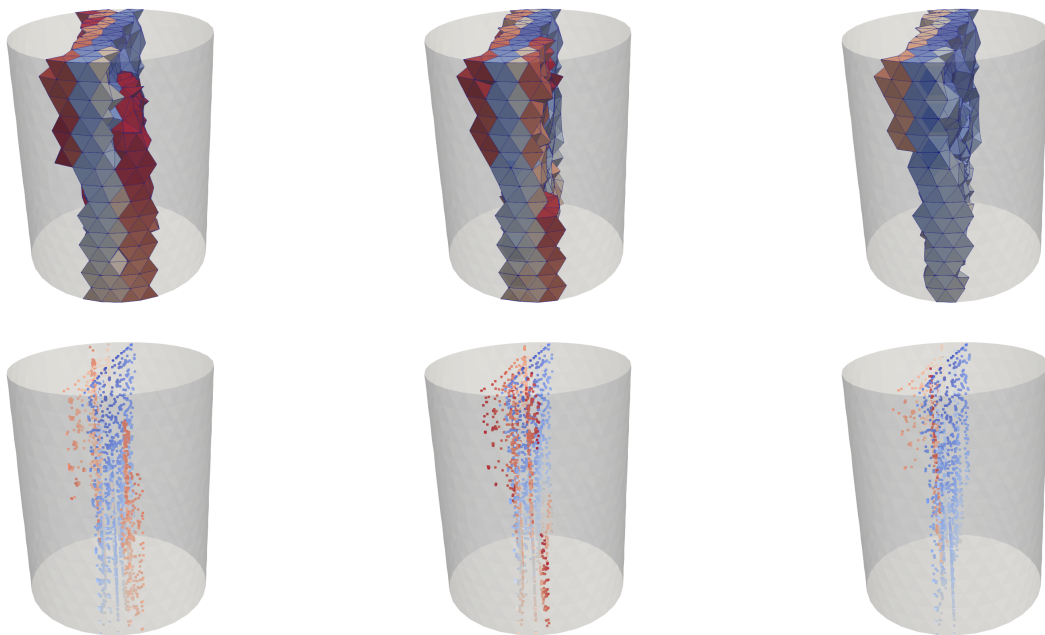


Figure 3.16 – L'impact du filtre géométrique sur la propagation du feuillet dual avec les tétraèdres du feuillet en haut et les points d'intersection créés. La première colonne montre le résultat obtenu sans appliquer le filtre. La seconde et troisième colonne correspondent respectivement à $\lambda = 0$ et $\lambda_g = \frac{\sqrt{2}}{2}$.

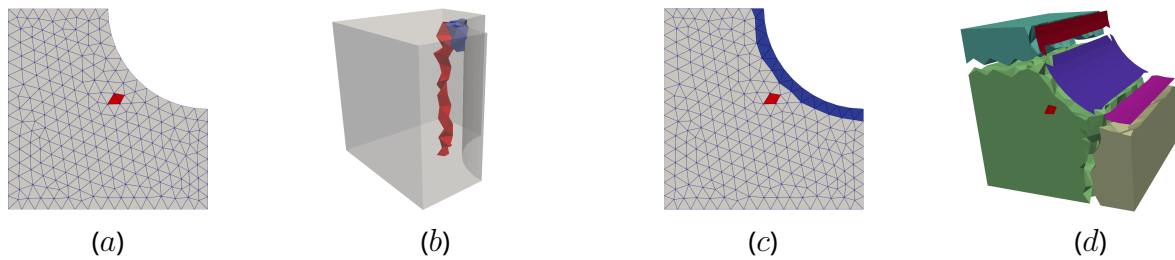


Figure 3.17 – Exemple où une ligne singulière est trop proche du bord (a) pour pouvoir tracer un feuillet dual (b). Nous utilisons notre méthode de feuillet au bord pour réussir à tracer un feuillet dans cette zone (c). En (d) nous pouvons voir la couche de tétraèdres plats se comportant comme le reste des tétraèdres.

Les feuillets au bord

La forme du domaine géométrique Ω ainsi que la discrétisation du maillage T_Ω peuvent grandement influencer le champ d'orientation notamment sur l'emplacement des lignes singulières. C'est le cas de l'exemple de la Figure 3.17 (a) où l'on peut voir que la ligne singulière en rouge est située près du bord du domaine. Cette configuration a un impact sur notre méthode de création de feuillets duaux car nous voulons pouvoir créer un feuillet dual entre la ligne singulière en question et la surface géométrique proche. Si le feuillet dual que l'on cherche à construire est trop proche du bord il va se coller à celui-ci et s'il entre en contact avec la ligne singulière cela peut arrêter sa création (b). Il existe selon [KLSO12] trois niveaux de feuillets fondamentaux dans un maillage hexaédriques, ces feuillets permettent d'obtenir des maillages bien structurés près du bord, ce qui est une caractéristique que nous recherchons. Les feuillets fondamentaux de niveau 1 correspondent à une couche de mailles alignée le long d'une surface géométrique. Selon la correspondance entre une couche de mailles hexaédriques et une surface duale nous pouvons alors dire qu'il existe une surface duale alignée le long d'une surface. Partant de ce constat, nous introduisons la création d'un feuillet dual le long d'une surface géométrique en ne tenant pas compte du champ d'orientations. Pour ce faire nous sélectionnons tous les tétraèdres qui possèdent un sommet classifié sur la surface géométrique ou sur l'une des courbes géométriques de son bord (c). Une fois cette opération terminée nous créons artificiellement une couche de mailles pour simuler le fait que le feuillet dual se situe à l'intérieur du volume. Pour chaque face au bord des tétraèdres du feuillet nous créons un tétraèdre plat au bord de la géométrie (d). Cette méthode nous permet de gérer les cas où les lignes singulières sont trop proches du bord.

3.3 . Validité de la structure duale

A la section précédente, nous avons vu comment créer une surface de coupe dans T_Ω en s'appuyant sur le champ d'orientations \mathcal{F} . En pratique, notre approche consiste à insérer de telles surfaces itérativement jusqu'à obtenir une **décomposition valide** de T_Ω , c'est-à-dire un partitionnement des tétraèdres de T_Ω non intersectés par une surface duale en un ensemble de zones qui forment le dual d'une structure de blocs hexaédriques.

La problématique à résoudre est donc de définir des règles de validité permettant d'analyser la décomposition de T_Ω réalisée par l'utilisateur. Si l'application interactive et itérative de l'algorithme décrit dans la Section 3.2 permet de partitionner interactivement les tétraèdres de T_Ω , le découpage obtenu à un instant donné peut ne pas correspondre à une structure de blocs qui "remplisse" la géométrie de Ω tout en étant en accord avec la topologie du champ d'orientation \mathcal{F} . Des critères doivent donc être définis. Pour illustrer nos propos, on peut considérer l'exemple de la Figure 3.18 où seule la dernière découpe permet de générer un maillage quadrangulaire respectant la géométrie du bord du domaine et le champ d'orientation.

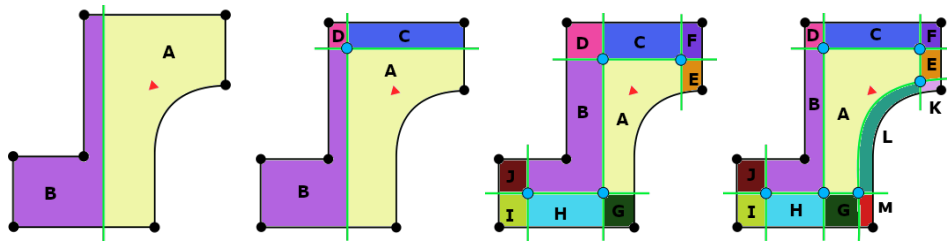


Figure 3.18 – Exemple 2D d’insertions successives de cordes duales (équivalentes aux feuillets duaux 3D) afin de générer un découpage valide du domaine en régions duales. Le découpage illustré sur la figure la plus à droite valide l’ensemble des règles nécessaires à la génération d’une structure de blocs.

Afin de produire une structure duale valide, les auteurs [CBK12, ZWG⁺18] proposent de contraindre la création de la structure duale à séparer les points singuliers du champs et les éléments géométriques surfaciques caractéristiques que sont les sommets et les courbes. Ils se reposent sur une approche surfacique sans prendre en compte les éléments volumiques. Pour notre part, nous nous inspirerons des contraintes de constructions de la structure duale de [Tak19] pour définir dans la partie 3.3.1 un ensemble de règles volumiques de validité de notre structure duale. Ces règles sont applicables à n’importe quel moment du procédé de création de la structure duale. Techniquement, nous nous appuyons sur la décomposition induite de T_Ω par la structure duale pour développer un algorithme qui permet de vérifier les règles de validité. Cet algorithme sera détaillé à la section 3.3.2.

3.3.1 . Définition des règles de validité

Notre approche repose sur deux éléments importants de la structure duale : les **nœuds duaux**, autrement dit les 0-cellules, et les **régions duales**, c’est-à-dire les n -cellules, où n est la dimension de la structure duale. Pour expliquer notre approche, nous allons considérer l’exemple 2D de la Figure 3.18. En 2D nous ne manipulons pas des surfaces duales mais des cordes duales, et les régions duales sont remplacées par des patchs 2D. En outre, les nœuds duaux sont l’intersection de deux cordes duales. On rappelle enfin que la correspondance entre le dual et le primal est la suivante :

- Un nœud dual correspond à un bloc primal ;
- Un patch 2D correspond à un sommet de bloc primal.

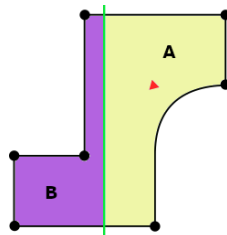


Figure 3.19 – Structure duale non valide, elle ne possède aucun nœud dual.

Considérons l’exemple de la Figure 3.19, où une unique corde a été insérée. Cette corde unique forme une structure duale dépourvue de nœud dual. D’après la définition de la correspondance entre le dual et le primal, il est nécessaire de disposer d’au moins un nœud dual pour générer une structure de blocs non vide. On en déduit alors la première règle de validation pour une structure duale \mathcal{SB} :

1. Soit N l’ensemble des 0-cellules de \mathcal{SB} , on a nécessairement $N \neq \emptyset$.

Autrement dit, la structure duale \mathcal{SB} doit posséder au moins un nœud dual.

Capture de la géométrie

Nous voulons que la structure de blocs corresponde à la géométrie de Ω . On dira qu'elle **capture** Ω . On définit pour cela la correspondance entre la structure de blocs et un modèle géométrique Ω de la façon suivante : Soient S, C, Sf et V , respectivement l'ensemble des sommets, courbes, surfaces et volumes géométriques de Ω et N, E, F et R respectivement l'ensemble des nœuds, arêtes, faces et régions de la structure de blocs (voir la figure 3.20 pour un exemple 2D). Nous dirons que SB capture Ω si et seulement si :

- Pour tout $s \in S$, il existe un nœud $n \in N$ qui soit classifié sur s ;
- Pour tout $c \in C$, il existe un ensemble non nul $E^c \subseteq E$ d'arêtes qui soient classifiées sur c ;
- Pour tout $f \in Sf$, il existe un ensemble non nul $F^f \subseteq F$ de faces qui soient classifiées sur f ;
- Pour tout $v \in V$, il existe un ensemble non nul $R^v \subseteq R$ de régions qui soient classifiées sur v ;
- Pour tout $n \in N$, n est classifié sur une seule entité géométrique de Ω .

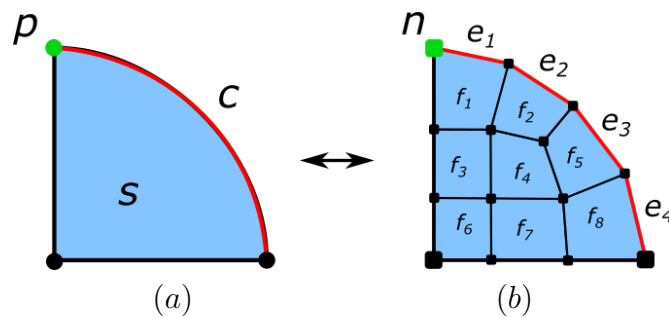


Figure 3.20 – Illustration de la correspondance entre la géométrie et la structure de blocs.

D'après les notions de correspondance entre une structure duale et une structure primale, une région duale en 3D, respectivement un patch dual en 2D, correspond à un nœud primal. Ces notions sont purement topologiques et la discrétisation du domaine Ω requiert de tenir compte de considérations géométriques aussi. Ainsi pour que la structure de blocs obtenue discrétise correctement Ω , il faut en particulier qu'à tout sommet $s \in \partial\Omega$ soit classifié un nœud de la structure de blocs SB .

Pour s'en assurer, nous considérons la relation qui lie les régions duales en 3D, respectivement les patches duaux en 2D, avec le domaine géométrique Ω et en particulier avec les éléments - sommets, courbes, surfaces - qui composent son bord $\partial\Omega$. Nous appelons cette relation la **caractérisation géométrique duale** et nous la notons \mathcal{C}_D . En dimension n , cette relation associe à toute n -cellule c_i de la structure duale une entité géométrique participant à la définition de Ω . Cette entité pourra donc être un sommet, une courbe, une surface ou un volume. Pour toute n -cellule c_i de la structure duale générée, la relation $\mathcal{C}_D(c_i)$ désigne l'unique élément géométrique de plus petite dimension qui soit adjacent à la cellule c_i . Si cet élément n'est pas unique, la cellule duale ne peut être caractérisée géométriquement. Considérons à titre d'exemple, la figure 3.21. En (a), la région duale \mathbf{r} représentée par un cube bleu est incident à 3 surfaces, trois courbes (en trait noir) et un sommet (en rouge). L'entité géométrique de plus petite dimension étant un sommet unique, nous avons alors $d(\mathcal{C}_D(\mathbf{r})) = s$ où s est le sommet en question. De façon similaire, en (b), la région duale \mathbf{r} est incidente une courbe et deux surfaces, elle est donc caractérisée sur cette courbe. En (c), la région duale \mathbf{r} est caractérisée sur une surface. Enfin, si une région duale n'est incidente à aucun sommet, courbe ou surface formant $\partial\Omega$, elle est caractérisée dans le volume géométrique la contenant.

La caractérisation géométrique duale des régions duales nous permet de définir la classification des nœuds de la structure de blocs ainsi que leur position. Autrement dit :

- Si une région duale est caractérisée sur un sommet géométrique s , alors le nœud de bloc correspondant sera situé sur s ;

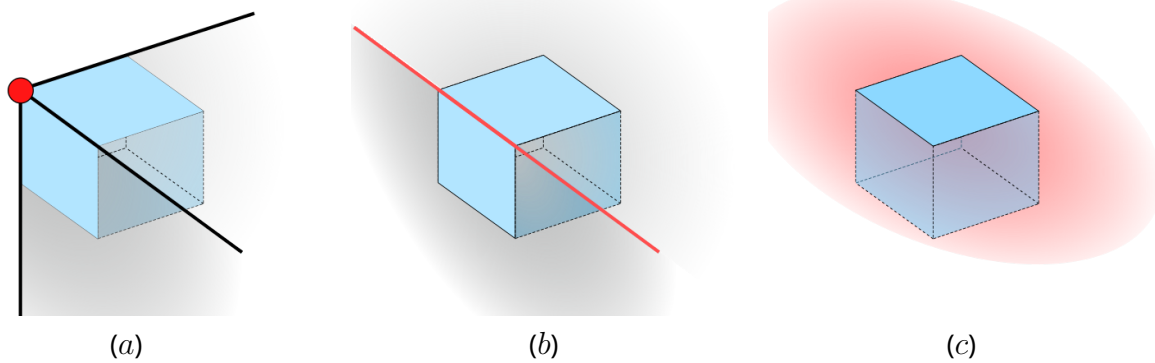


Figure 3.21 – Illustration des différents cas de caractérisation géométrique duale d'une région duale, avec dans chaque cas l'entité géométrique sur laquelle elle est caractérisée en rouge. La région duale est caractérisée au sommet (a), à la courbe (b) et à la surface (c).

- Si la région duale est caractérisée sur une courbe géométrique, respectivement une surface géométrique, alors le nœud de bloc sera positionnée librement sur la courbe, respectivement la surface, en question ;
- Enfin pour les régions duales caractérisées sur un volume, deux cas de figure peuvent se présenter :
 1. soit la région contient un point singulier du champ, et alors le nœud correspondant sera positionné sur cette singularité ;
 2. soit la région ne contient aucun point singulier, et alors le nœud correspondant est libre de se déplacer dans ce volume.

Reprenons maintenant notre exemple de la Figure 3.19 et insérons une nouvelle corde duale comme illustré sur la Figure 3.22. On obtient alors un nœud dual et quatre régions duales nommées A , B , C et D . Cette structure duale vérifie la règle 1 et la structure de blocs correspondante contient un bloc primal. Maintenant si nous caractérisons géométriquement nos régions duales, on constate que les régions duales A , B , C et D sont toutes caractérisées sur au moins un sommet géométrique. En particulier, les régions duales A et B sont associées à plusieurs sommets géométriques. Il est donc impossible de positionner le nœud de bloc correspondant de manière unique³. On définit alors la règles suivante :

2. Pour tout sommet géométrique s de Ω , il existe une et une seule région duale r qui soit caractérisée géométriquement sur s .

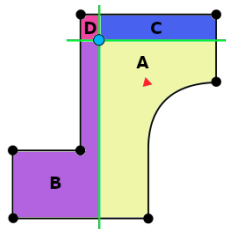


Figure 3.22 – Structure duale non valide. les régions duales A et B sont géométriquement caractérisées par respectivement deux et trois sommets géométriques.

En considérant cette nouvelle règle, la Figure 3.23 nous montre en (a) une structure duale qui vérifie à la fois les règles 1 et 2. Pour cela, trois nouvelles cordes ont été ajoutées à la structure duale pour créer

³. Si on fait un choix parmi les sommets géométriques alors le maillage obtenu ne discrétisera pas le domaine Ω correctement.

les nouvelles régions duales E, F, G, H, I et J . Nous disposons désormais d'une et une seule région duale pour caractériser chaque sommet géométrique. On peut voir en (b) la structure de blocs correspondante.

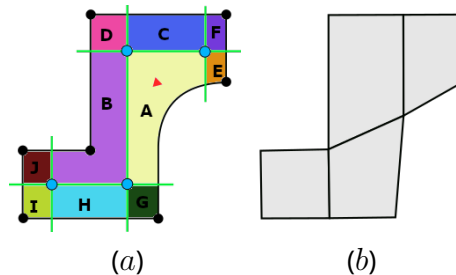


Figure 3.23 – Une structure duale valide (a) qui correspond à une structure de blocs (b).

Correspondance avec le champ d'orientation.

Nous avons proposé jusqu'à présent des règles qui permettent de vérifier la correspondance avec la géométrie mais ces règles n'assurent par contre pas que la topologie de la structure duale corresponde à celle du champ d'orientation. En théorie, le champ d'orientation nous fournit une information de direction à suivre pour la structure de blocs et le maillage hexaédrique à l'intérieur du domaine Ω . Dans les zones où le champ est régulier un bloc unique permettrait de fournir la structure de grille correspondante. Par contre, des singularités du champ indiquent naturellement des sommets où plusieurs blocs se rencontrent. L'idée ici est de placer les nœuds et les arêtes de blocs sur le graphe de singularité, c'est-à-dire qu'un sommet de bloc sera placé sur un point singulier et qu'un chemin d'arêtes de bloc sera créé le long des lignes singulières. Les auteurs de [CBK12, ZWG⁺18] sont confrontés à ce problème d'isoler les points singuliers sur le bord du domaine. Pour le résoudre interactivement, ils connectent artificiellement les points singuliers par des lignes, nommées *indicateurs de séparation*, qui devront être coupés par une corde duale par la suite. Pour notre part, nous poursuivons l'utilisation des régions duales pour caractériser la correspondance avec les singularités du champ. En l'occurrence comme pour les sommets géométriques, une région duale ne peut contenir qu'un seul point singulier (en 2D comme en 3D). Tout comme [ZWG⁺18], nous considérons les sommets géométriques comme des points qui doivent être séparés des points singuliers puisqu'ils correspondent chacun à un nœud de bloc.

3. Pour tout point singulier p surfacique ou volumique, il existe une et une seule région duale r contenant p .

Un cas conflictuel apparaît alors suite à cette règle dans les régions duales qui contiennent un point singulier et une courbe ou une surface géométrique. Nous avons vu que si une région duale est classifiée sur une courbe ou une surface alors le nœud de bloc correspondant est situé sur l'entité géométrique en question. Si nous prenons l'exemple 2D de la Figure 3.24, on a en (a) le domaine géométrique Ω avec un champ d'orientation contenant une singularité représentée par un triangle rouge. En (b), nous voyons une région duale en rouge qui est caractérisée géométriquement sur une courbe géométrique tout en contenant la singularité du champ. Si nous respectons la topologie du champ et que l'on positionne le nœud de bloc correspondant à la région duale rouge sur la singularité du champ, nous ne discrétisons pas correctement Ω . Si par contre, on se positionne sur la courbe, nous obtenons un unique bloc dont on peut extraire une grille telle que celle illustrée en (c). Bien que valide, ce maillage possède des mailles avec une forte distorsion en plus de ne pas respecter l'alignement au bord, ce que nous voulons éviter. Nous allons alors définir la règle suivante : une région duale au bord ne peut contenir une singularité du champ d'orientation.

En suivant cette règle, on obtient à la Figure 3.24 (d) une structure duale où la région conflictuelle est maintenant séparée en deux et avec une région duale centrale qui correspond à la singularité du champ et qui n'est caractérisée sur aucune entité géométrique du bord de Ω . Enfin, en (e) nous obtenons le maillage correspondant à la structure duale, nous voyons que cette règle contraint l'alignement du maillage avec le bord de Ω . On obtient donc la règle suivante à respecter.

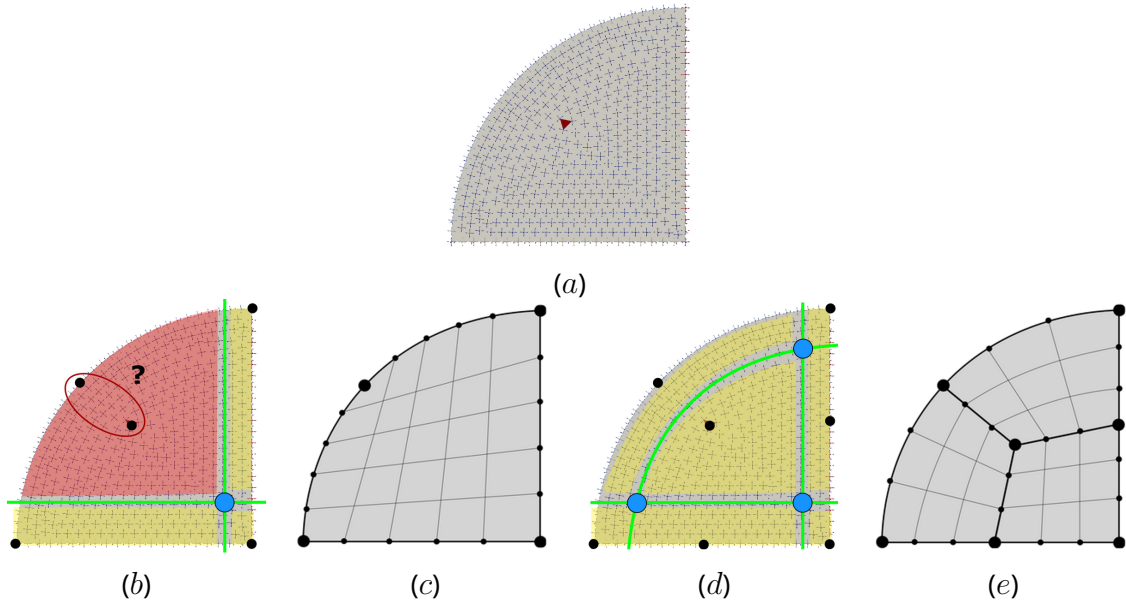


Figure 3.24 – Exemple de domaine géométrique muni de son champ d'orientation en (a). Le triangle rouge contient un point singulier du champ. En (b) et (d) deux structures duales différentes avec en (c) et (e) un maillage quadrangulaire correspondant pour chaque structure duale.

4. Pour toute région duale \mathbf{r} , si $\dim(\mathcal{C}(\mathbf{r})) < 3$ alors \mathbf{r} ne contient pas de point singulier volumique.

Si nous appliquons maintenant cette nouvelle règle à notre exemple nous allons insérer une nouvelle corde duale entre la singularité contenue dans la région duale A et le bord. Nous obtenons trois nouvelles régions duales K , L et M comme illustré à la Figure 3.25.

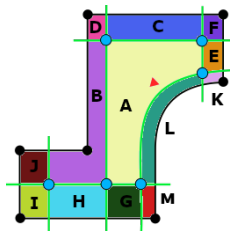


Figure 3.25 – La structure duale vérifie désormais les quatre règles exprimées.

3.3.2 . Algorithme de vérification des règles

Maintenant que nous avons défini les règles de validité que doit vérifier la structure duale, nous allons détailler notre algorithme de validation. Nous allons pour ce faire travailler sur le maillage T_Ω qui discrétise le domaine géométrique 3D Ω et qui supporte notre structure duale. On rappelle que dans notre implémentation, nous définissons un feuillet dual comme étant un ensemble de tétraèdres partitionnant T_Ω en deux sous-ensembles distincts de tétraèdres. Une fois nos feuillet duaux extraits, nous obtenons donc plusieurs

sous-ensembles de tétraèdres. Chacun de ces sous-ensembles est bordé par des faces du bord de T_Ω ou des faces adjacentes à des tétraèdres contenus dans les feuilletts duaux. On va alors vérifier que la structure duale valide les quatre règles précédemment établies.

La règle 1 nous impose d'avoir au moins un nœud dual dans la structure duale. Un nœud dual est l'intersection de trois surfaces duales et par correspondance dans notre implémentation un nœud dual est l'intersection de trois feuilletts duaux. En pratique, durant l'algorithme de création de feuilletts, nous sauvegardons pour chaque tétraèdre t trois couples $\{(\mathbf{d}_i, i), (\mathbf{d}_j, j), (\mathbf{d}_k, k)\}$, qui indiquent chacun pour une direction d_ℓ , avec $\ell = i, j$ ou k , l'indice i, j ou k du feuillet correspondant. Chacune des directions $\mathbf{d}_i, \mathbf{d}_j$ et \mathbf{d}_k correspond à une des directions de l'orientation interpolée en t . Le feuillet dual traversant t associée à la direction \mathbf{d} est tel que le vecteur normal au feuillet soit aligné avec \mathbf{d} localement à t . Donc si nous disposons d'un tétraèdre qui, pour chaque direction, est associée à un feuillet dual alors le tétraèdre est traversé 3 fois et peut donc être considéré comme contribuant à former un nœud dual. La Figure 3.26 nous montre en (a) un tétraèdre t_k intersecté par trois feuilletts dans les trois directions du champ. Comme les feuilletts duaux sont des ensembles de tétraèdre intersectés par une surface de coupe, on a vu dans la Section 3.2 que la discrétisation du maillage a un impact sur l'épaisseur des feuilletts duaux. Ce phénomène influe également sur les nœuds duaux, en effet les feuilletts peuvent s'intersecter en un ensemble de tétraèdres. Ces tétraèdres auront alors tous la même combinaison de feuilletts associés aux directions du champ. On considère cet ensemble comme représentant un nœud dual unique. C'est le cas sur la Figure 3.26(b) où t_k est un ensemble de plusieurs tétraèdres.

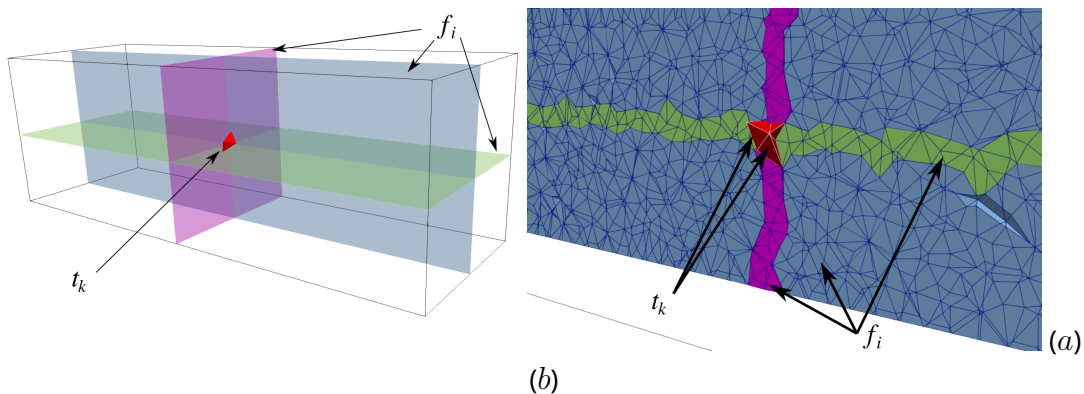


Figure 3.26 – Deux représentation des feuilletts duaux f_i et du nœud dual obtenu. En (a) les feuilletts f_i sont représentés par leur surface de coupe et le nœud dual obtenu est le tétraèdre t_k . En (b), une coupe du maillage avec les tétraèdres appartenant aux feuilletts f_i et le nœud dual obtenu représenté par un ensemble de tétraèdres t_k intersectés par les feuilletts f_i dû à la qualité des mailles.

Afin de vérifier que notre structure duale valide les règles établies à la Section 3.3.1, nous avons également besoin d'informations sur les régions duales et plus particulièrement sur leurs bords. En ce qui nous concerne, une région duale est un volume bordé par un ensemble de surfaces duales et/ou le bord du domaine Ω . Dans notre approche, une région duale est décrite par le triplet $\mathbf{r} = \{i, \mathbf{c}, \sigma\}$ avec i l'indice de \mathbf{r} , \mathbf{c} l'ensemble des entités géométriques de dimension minimale adjacentes à \mathbf{r} et σ l'ensemble des feuilletts duaux qui forment le bord de \mathbf{r} . Notons que $\mathbf{r}.\mathbf{c}$ est un ensemble et non pas une valeur unique, comme préconisé par la règle 3, car en cours de construction de la structure duale, cette règle peut ne pas être vérifiée. Dans le maillage T_Ω , \mathbf{r} est un ensemble connexe de tétraèdres. Pour obtenir cet ensemble, nous appliquons un algorithme de coloriage sur les tétraèdres de T_Ω . Nous appelons "coloriage" le fait de donner une valeur à un ensemble de cellules du maillage. Décrivons maintenant l'algorithme de coloriage que nous appliquons. Nous partons d'un maillage dont chaque tétraèdre est soit traversé par au moins un feuillet dual (en bleu foncé sur la Figure 3.27(a)) soit traversé par aucun. Dans ce second cas, nous qualifierons le tétraèdre comme étant *libre*

(en blanc sur la figure). Le but de la méthode est d'associer à chaque composante connexe de tétraèdres libres une couleur. Nous notons $color$ la fonction qui à un tétraèdre associe la couleur de la région à laquelle il appartient. Nous initialisons $color$ comme suit. Pour tout tétraèdre $t \in T_\Omega$, $color(t) = 0$ si t est libre et $color(t) = -1$ sinon. Nous allons ensuite pouvoir commencer à colorier le maillage T_Ω pour créer les régions duales. Pour cela, nous appliquons successivement les algorithmes 3 et 4.

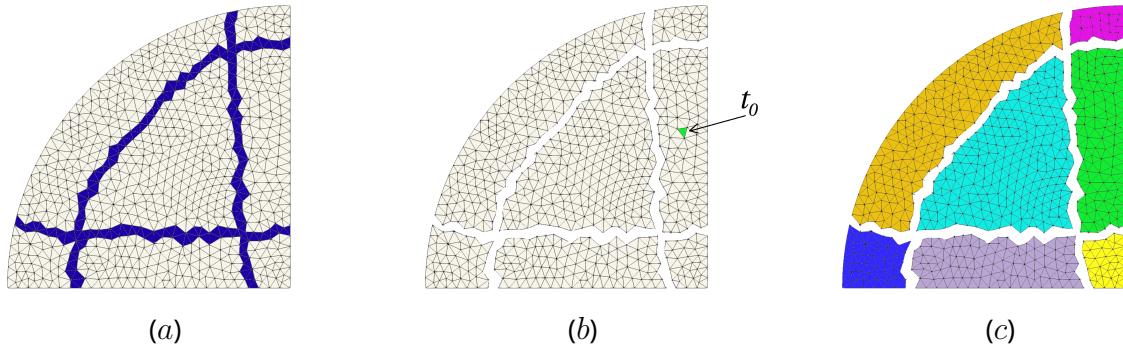


Figure 3.27 – Une illustration 2D de l'algorithme de coloriage du maillage. En (a) les tétraèdres traversés par au moins un feuillet dual sont coloriés avec la valeur -1 (couleur bleue); Le maillage avec seulement les tétraèdres libres et t_0 le point de départ de la coloration en (b); Et en (c) le maillage obtenu suite à l'application complète de l'algorithme de coloriage, plus aucun tétraèdre n'est libre.

Algorithme 3 : dualRegionsCreation

```

1 dual_regions_set  $\leftarrow \emptyset$ ;
2  $r_{ID} \leftarrow 1$ ;
3 foreach tet  $t_0 \in T_\Omega$  do
4   if  $color(t_0) = 0$  then
5      $\mathbf{r} \leftarrow$  new dual region;
6      $\mathbf{r}.i \leftarrow r_{ID}$ ;
7     colorDual( $\mathbf{r}, t_0$ );
8     if  $isGoodRegion(\mathbf{r})$  then
9       insert  $\mathbf{r}$  into  $dual\_regions\_set$ ;
10       $r_{ID} ++$ ;
11    end
12  end
13 end

```

Nous allons commencer par chercher un tétraèdre libre t_0 , c'est-à-dire tel que $color(t_0) = 0$ (en vert à la Figure 3.27(b)). Ce tétraèdre va être le point de départ de notre algorithme de création d'une région duale. Partant de t_0 , nous pouvons créer une nouvelle région duale \mathbf{r} d'indice r_{ID} . C'est cet indice qui encode la "couleur" de la région duale qui sera propagée dans le maillage (lignes 4 et 5 de l'algorithme 3).

Nous entrons ensuite dans la phase de propagation de la couleur dans un ensemble de tétraèdres libres t (Algorithme 4). Si t est libre donc $color_t = 0$ alors il est colorié avec l'indice de \mathbf{r} (ligne 4), si $color(t) = -1$ cela signifie que t est un tétraèdre traversé par un feuillet dual qui forme le bord de \mathbf{r} (ligne 12). Lors de la propagation pour obtenir la région duale \mathbf{r} nous allons également parcourir la classification géométriques des nœuds des tétraèdres (lignes 5 à 7) afin de mettre à jour l'ensemble $\mathbf{r}.c$. Une fois la phase de coloriage du maillage pour \mathbf{r} achevée, si l'ensemble $\mathbf{r}.c$ contient au plus un sommet géométrique, la caractérisation

Algorithme 4 : *colorDual*

```
Data : ( $\mathbf{r}, t_0$ )
1 insert  $t_0$  into next_tets;
2 foreach tet  $t \in next\_tets$  do
3   if  $color(t) = 0$  then
4      $color(t) \leftarrow \mathbf{r}.i$ ;
5     foreach node  $n \in t$  do
6       insert  $\mathcal{C}(n)$  into  $\mathbf{r}.c$ ;
7     end
8     isSingularPoint( $t$ );
9     insert getNeighbors( $t$ ) into next_tets;
10  end
11  else if  $color(t) = -1$  then
12    insert getDualSheets( $t$ ) into  $\mathbf{r}.\sigma$ ;
13  end
14 end
```

géométrique de \mathbf{r} est correcte, sinon la règle 2 n'est pas respectée. Nous aurions alors au moins deux sommets de tétraèdres de la région duale \mathbf{r} qui seraient classifiés sur des sommets géométriques différents. Nous conservons également les points singuliers du champ pendant le parcours des tétraèdres. Lors de la propagation, nous testons si le tétraèdre courant est singulier (ligne 8). Cette étape va nous permettre de vérifier la règle 3. Si plusieurs points singuliers sont trouvés pendant le parcours pour colorier une région duale alors celle-ci ne valide pas la règle 3. Il est cependant à noter qu'un tétraèdre singulier ne représente pas un point singulier du graphe de singularité. En effet les lignes singulières sont représentées par un ensemble de tétraèdres singuliers dans T_Ω , nous considérerons donc seulement les tétraèdres singuliers possédant une face au bord de T_Ω - ceux aux extrémités des lignes singulières - ainsi que ceux considérés comme instables⁴ à l'intérieur de T_Ω - formant un point de rencontre entre lignes de singulières. Si une région duale ne contient au plus qu'un de ces tétraèdres alors elle vérifie la règle 3. Nous pouvons voir un tel exemple à la Figure 3.28, la région duale \mathbf{r} en jaune est traversée par une ligne singulière en bleu, et elle ne contient qu'un seul tétraèdre singulier t_s qui est une extrémité de ligne singulière en rouge. La région duale \mathbf{r} vérifie ainsi la règle 3.

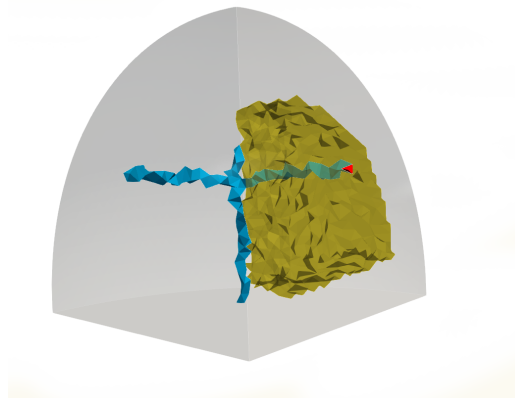


Figure 3.28 – Un exemple de région duale valide en jaune possédant qu'un seul point singulier en rouge.

4. Voir page 50.

Une fois l'ensemble des tétraèdres libres parcourus et coloriés, nous disposons des régions duales de notre structure. Pour toutes les régions, nous connaissons leurs caractérisations géométriques et les points singuliers du champ qu'elles contiennent. Nous allons finalement tester pour les régions duales caractérisées par une entité au bord de Ω (donc de dimension inférieure ou égale à 2) si elle contiennent un point singulier volumique.

3.4 . Création de la structure primale de blocs

Nous avons vu à la Section 3.3 comment vérifier la validité d'une structure duale. Une fois cette étape effectuée, nous voulons générer la structure primale de blocs correspondante. C'est l'objet de cette section où nous allons nous appuyer sur la caractérisation géométrique des régions duales et sur leur relation d'adjacence pour construire les blocs correspondants. Partant d'une structure duale telle que celle de la figure 3.29-*a*, nous construisons la structure de blocs présentée en (*b*). Pour ce faire notre méthode se divise en deux étapes importantes :

1. Nous commençons par extraire la topologie de la structure de blocs, c'est-à-dire que l'on va identifier comment regrouper les régions duales par groupe de 8 pour obtenir les sommets de chaque bloc primal.
2. Ensuite, on ordonne ces sommets afin d'obtenir un bloc hexaédrique valide.

Enfin, la géométrie des blocs sera induite par la caractérisation géométrique des régions duales.

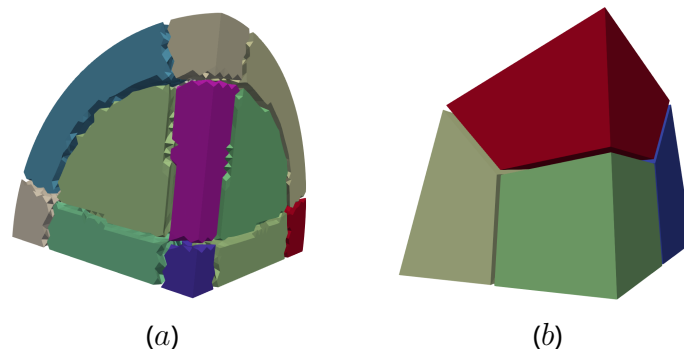


Figure 3.29 – Illustration des étapes de la génération de structure de blocs. En (*a*) les régions duales de la structure duale, en (*b*) les blocs générés grâce à (*a*).

3.4.1 . Extraction de la topologie

Le but de cette étape est de passer d'un nuage de points dont nous avons les coordonnées grâce aux régions duales caractérisées géométriquement sur des entités géométrique composant Ω et des points singuliers de \mathcal{F} . La problématique est que nous voulons créer cette structure de blocs en réduisant l'utilisation de calculs géométriques, qui peuvent amener des erreurs numériques ou des cas de géométries complexes. Pour cela, notre approche consiste à reconstruire un maximum d'informations topologiques à partir des régions et de feuilles définis sur T_Ω . En particulier, nous cherchons à extraire les liens d'adjacences entre régions de la structure duale.

Pour ce faire, nous nous basons sur les travaux de [Tak19], où les auteurs proposent également d'utiliser l'adjacence des régions duales pour recréer la topologie de la structure de blocs. Leur méthode consiste à assigner un code à chaque région et nœud dual. Ce code représente la position des cellules duales dans le domaine par rapport aux feuillet duaux. Étant donné un feuillet dual f , trois symboles sont utilisés pour caractériser la position d'une cellule duale par rapport à f : les symboles $+$ et $-$ permettent d'indiquer la

position relative d'une cellule par rapport à f ; le symbole $\mathbf{0}$ est utilisé pour indiquer si une entité est située sur le feuillet. Par définition, seuls les nœuds duaux peuvent être situés sur des feuillets duaux. Une cellule (région ou nœud) est donc encodée par un vecteur prenant ces valeurs dans $\{+, -, \mathbf{0}\}^n$, où n est le nombre de feuillets duaux. La Figure 3.30⁵ montre un exemple de cet encodage avec un découpage par quatre feuillets d'un domaine simple. On y voit les codes des régions duales en noir et ceux des nœuds duaux en vert. En utilisant cet encodage, on peut rapidement définir la position relative d'un bloc par rapport à un autre et induire leurs relations d'adjacence.

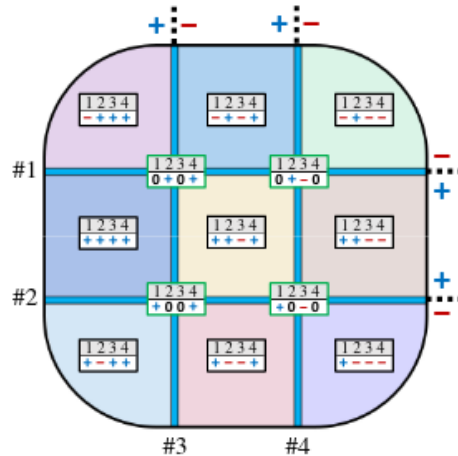


Figure 3.30 – Illustration de la méthode d'extraction de la topologie de la structure duale proposée par [Tak19].

Un des cas problématiques dans la méthode de [Tak19] est l'incapacité à gérer les feuillets qui s'auto-intersectent, comme mentionné dans leurs travaux à la Figure 3.31. Leur méthode est globale et implique que chaque feuillet doit partitionner le domaine en seulement deux sous-domaines. Si ce n'est pas le cas, comme le feuillet rouge de la Figure 3.31, les côtés $+$ et $-$ nécessaires pour encoder les cellules sont mal définis. En pratique, les auteurs de [Tak19] ne rencontrent pas cette configuration : ils interdisent les cordes auto-intersectantes en surface et ensuite créent des feuillets dans le volume par la "simple" définition d'une surface partitionnant en deux le domaine et dont le bord serait une corde créée en surface.

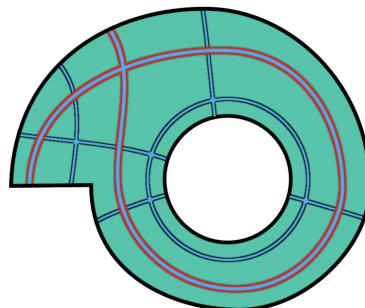


Figure 3.31 – Cas problématique énoncé dans les travaux [Tak19] dû à l'auto-intersection du feuillet dual en rose.

5. Image issue de l'article [Tak19]

Notre méthode

En prenant en compte les limitations de leur approche et le fait que nous travaillons avec un champ d'orientation volumique duquel on peut extraire des feuillets auto-intersectants, nous proposons d'adopter une solution plus locale. Nous conservons le principe d'encodage des cellules duales par rapport aux feuillets duaux, cependant le code obtenu ne sera plus défini de manière géométrique absolue (position d'un côté ou de l'autre d'un feuillet) mais comme la liste des feuillets adjacents à cette cellule. Considérons l'exemple de structure duale 2D de la Figure 3.32(a). En (b), nous avons l'encodage obtenu en appliquant l'approche proposée par [Tak19], tandis qu'en (c) est représenté l'encodage obtenu avec notre approche. Comme indiqué précédemment, notre encodage d'une cellule repose sur la connaissance de feuillets incidents à cette cellule. Pour une région, nous aurons donc la liste ordonnée par ordre croissant des indices des feuillets duaux. Pour un nœud, ce sera la liste des feuillets traversant le traversant. Notons qu'avec cette définition, la taille du code de chaque cellule peut varier.

De manière similaire aux travaux de [Tak19], nous regroupons les régions duales qui "encerclent un nœud dual". Chaque ensemble contiendra les huit régions duales correspondant aux huit sommets du bloc primal correspondant au nœud dual encerclé. Dans notre exemple 2D, un bloc primal possède quatre sommets, nous devons donc regrouper les quatre régions duales correspondantes. Nous utilisons le code du nœud dual à traiter, celui-ci possédant la liste des feuillets le traversant, toute région duale qui contient l'ensemble de ces feuillets dans son code est alors incidente au nœud dual.

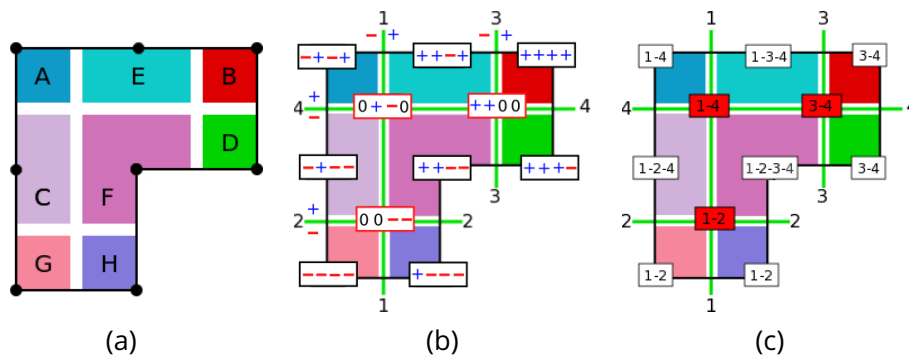


Figure 3.32 – Comparaison entre la méthode de codage proposée par [Tak19] (b) et notre méthode (c).

Reprenons maintenant notre exemple 2D avec la Figure 3.33. Supposons que nous souhaitons traiter le nœud dual n désigné par un rond bleu en (a). Ce nœud n a pour code [1-4] comme montré en rouge en (b), il est donc traversé par les feuillets 1 et 4. On va alors tester le code de chaque région r duale, si celui-ci contient les feuillets 1 et 4 alors la région r est sélectionnée. En utilisant cette méthode on obtient alors les régions A , E , F et C . Grâce à l'extraction de la topologie de la structure duale, nous savons que les sommets de blocs correspondants à ces régions duales forment un bloc primal.

Cas particuliers des multi-intersections de feuillets

Un cas particulier de la méthode de [Tak19] concerne les intersections multiples entre mêmes feuillets. Le problème lié aux intersections multiple est que plusieurs nœuds duaux vont avoir le même code. On peut voir à la Figure 3.34 un exemple de cas d'intersection multiple entre les feuillets 1 et 2. On voit également que notre méthode, en (c), de manière similaire à celle de [Tak19], en (a), ne permet pas de différencier les nœuds duaux. Ils sont encodés avec le même code. Les auteurs proposent alors de rajouter un feuillet qui sépare les deux nœuds, ce qui leur permet d'obtenir un code différent pour chacun des nœuds. Alors que cette approche fonctionne pour [Tak19], comme on peut le voir à la Figure 3.34(b), elle ne permet pas de

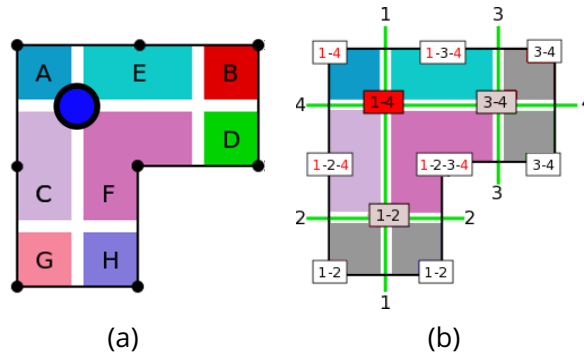


Figure 3.33 – En (a) le nœud dual en bleu dont nous voulons retrouver les régions duales adjacentes. En (b) les quatre régions duales associées au code du nœud dual sont coloriées, les autres grisées.

traiter ce cas particulier dans notre approche. En effet, les surfaces qu'ils créent dans le domaine ne sont pas contraintes dans le volume, autrement dit, si on prend la surface de coupe S on peut avoir S_1 et S_2 qui coupent un domaine Ω et tel que $S_1 \in S$, $S_2 \in S$ et $S_1 \cap S_2 = \emptyset$. Le feuillet 3 de la Figure 3.34(b) possède deux surfaces de coupe internes, ce qui permet de partitionner le domaine en deux. Les deux nœuds duaux étant chacun d'un côté différent du feuillet 3, on peut alors obtenir un code différent. On ne peut cependant pas utiliser ce traitement des intersections multiples puisque notre méthode de création des feuillets se base sur l'utilisation du champ d'orientation à l'intérieur du domaine. Nous ne pouvons donc pas partitionner le domaine en deux de manière similaire à [Tak19]. De plus même en créant les feuillets duaux 3 et 4 à la Figure 3.34, nous pouvons constater que les codes des nœuds problématiques ne changent pas puisque les feuillets 3 et 4 ne sont pas incidents avec les nœuds problématiques.

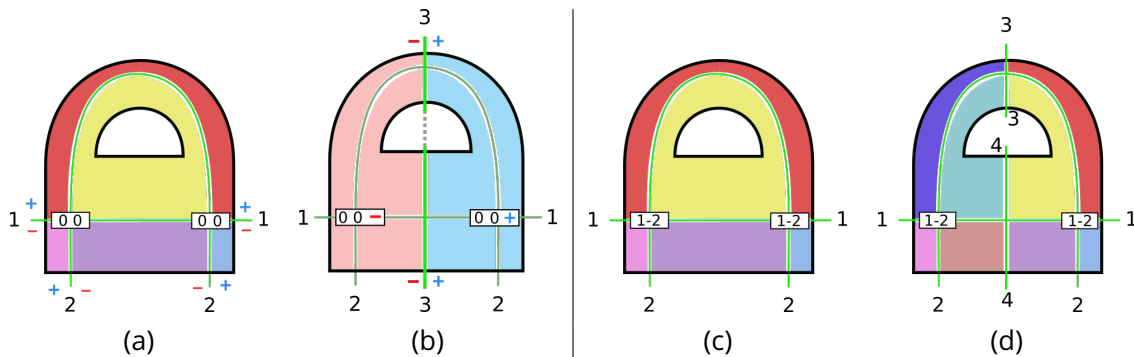


Figure 3.34 – Comparaison entre la méthode de [Tak19] à gauche et la nôtre à droite sur un cas de feuillet dual avec une intersection multiple. En (a) et (c), ces cas sont traités de façon normale, les deux nœuds duaux ont le même code. En (b) et (d) des feuillets duaux sont rajoutés pour gérer ce cas particulier.

Cependant il est à noter que même s'il est impossible de différencier les nœuds par leur code, l'étape de sélection des régions duales nous permet d'obtenir celles qui sont adjacentes aux nœuds duaux problématiques. La Figure 3.35 nous montre un exemple de régions duales sélectionnées malgré l'intersection multiple. Nous connaissons les nœuds problématiques N ainsi que l'ensemble \mathbf{R} des régions adjacentes à ceux-ci, on a donc besoin de développer une méthode qui nous permet, pour chaque nœud $n_i \in N$ de générer un sous-ensemble \mathbf{R}_i de \mathbf{R} tel que chaque région duale de \mathbf{R}_i soit incidente à n_i . Notre solution consiste à utiliser la position des cellules duales dans le domaine pour retrouver dans \mathbf{R} les régions duales les plus proches d'un nœud n_i . Pour ce faire nous nous servons de la représentation explicite de notre structure duale sur le maillage T_Ω . Partant d'un tétraèdre t qui est traversé par les trois feuillets f_1, f_2 et f_3 qui s'intersectent pour former le nœud dual n_i , nous nous propageons en largeur dans tous les tétraèdres voisins également traversés par les feuillets duaux. De cette façon nous allons être bloqués localement au

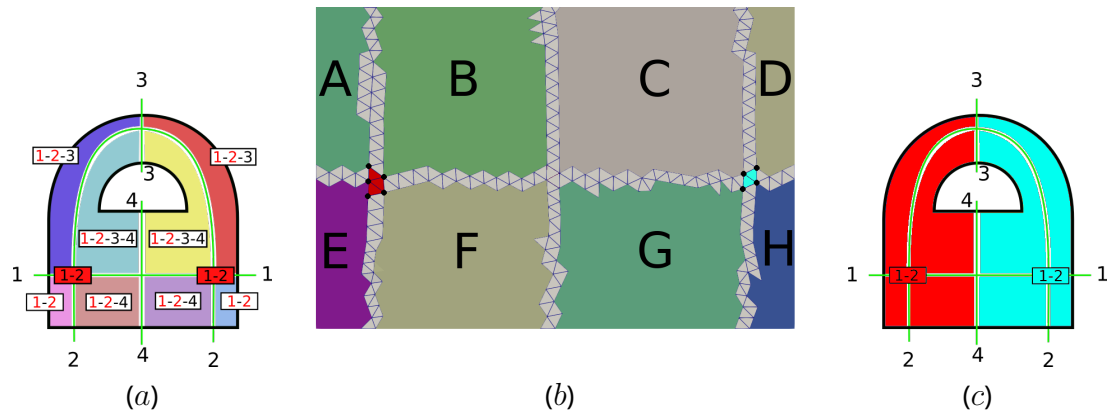


Figure 3.35 – Notre codage des régions duales sur un cas de multi-intersection (a). En (b) chaque nœud dual va être traité indépendamment, ainsi le nœud dual de gauche (en rouge) est en contact avec quatre régions duales, tout comme le nœud dual de droite (en bleu clair). Cela nous permet d'associer les régions duales à chaque nœud en (c).

nœud dual. Cette étape peut être vue à la Figure 3.35(b) où nous avons parcouru les tétraèdres en rouge à gauche. Si un sommet de l'un des tétraèdres parcouru est incident à un tétraèdre appartenant à une région duale de \mathbf{R} alors nous l'ajoutons dans \mathbf{R}_i . Dans notre exemple en (b) les sommets des tétraèdres en rouge sont en contact avec les régions duales A, B, E et F. Cette opération est répétée tant que des tétraèdres traversés par f_1, f_2 et f_3 n'ont pas été parcourus. Dans pareil cas, cela signifie qu'un autre nœud dual n'a pas été traité. De cette façon, nous obtenons les tétraèdres en bleu ciel à droite en (b) dont les sommets sont en contact avec les régions duales C, D, G et H. Cette opération nous permet d'obtenir un ensemble de régions duales pour chaque nœud dual en (c).

Cas particulier des feuillet auto-intersectants

De façon similaire aux intersections multiples, les feuillet auto-intersectants amène un cas particulier lors de la génération des codes des nœuds duaux. Les travaux de [Tak19] montrent que ce cas est limitant pour leur approche dû à leur utilisation d'une représentation implicite des feuillet duaux ainsi qu'à leur méthode d'extraction topologique pour la création des blocs. Notre système de codage ainsi que notre extraction des régions duales incidentes nous permet de gérer les cas d'auto-intersection. Nous illustrons nos propos avec la Figure 3.36. La génération des codes des cellules duales nous permet d'obtenir le résultat en (a). Considérons le nœud rouge où le feuillet 1 s'auto-intersecte et extrayons les régions duales qui lui sont incidente. Le fait que ce nœud ne soit incident qu'à un seul feuillet fait que son code vaut [1], indice de feuillet partagé par toutes les régions duales de notre structure. En utilisant, comme pour les multiples intersections, la représentation explicite sur T_Ω d'un nœud dual, on arrive à sélectionner les régions duales incidentes à ce nœud. Le résultat de la sélection est illustré en (b) avec les quatre régions colorées.

3.4.2 . Ordonnancement des sommets de blocs

Nous avons réduit notre ensemble total de sommets à un ensemble de groupes de 8 sommets correspondants aux sommets des blocs primaux. Cependant, ces ensembles ne sont pas munis d'une relation d'ordre. Il est nécessaire de reconstruire une numérotation locale des huit sommets en question pour ne pas générer un hexaèdre dégénérés (croisement entre les arêtes, torsion du volume). Nous proposons pour cela l'approche suivante pour chaque groupe de huit sommets :

1. Nous commençons par reconstruire le voisinage topologique de chaque sommet pour définir les coins du bloc, c'est-à-dire huit quadruplets (s, s_i, s_j, s_k) de sommet formant le coin de l'hexaèdre à créer au sommet s ;

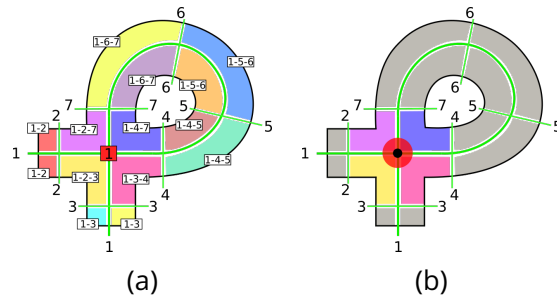


Figure 3.36 – En (a), notre codage des régions duales appliqué à un cas de feuillet auto-intersectant (le feuillet 1), nous permet d'obtenir en (b) le nœud dual en rouge associé avec ses quatre régions duales.

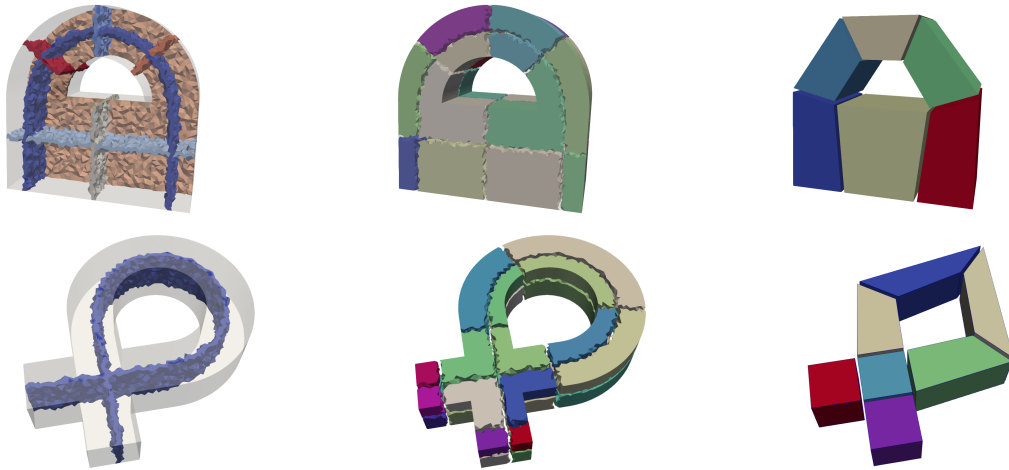


Figure 3.37 – Notre méthode de création de feuillets duaux (à gauche), de génération de régions duales (au centre) et de structure de blocs (à droite) pour un cas d'intersection de feuillet dual multiple sur la première ligne et un cas d'auto-intersection sur la seconde ligne.

2. Une fois muni de ces huit quadruplets, le bloc hexaédrique souhaité pourra être reconstruit sur des critères purement combinatoires impliquant les huit coins.

Définition du voisinage en chaque coin de bloc

La première étape de notre approche consiste à reconstruire le voisinage topologique de chaque sommet du bloc, c'est-à-dire pour nous une information de **coin** associée au sommet. Un coin est défini par son sommet d'origine n ainsi que trois sommets voisins n_x , n_y et n_z comme illustré à la Figure 3.38.

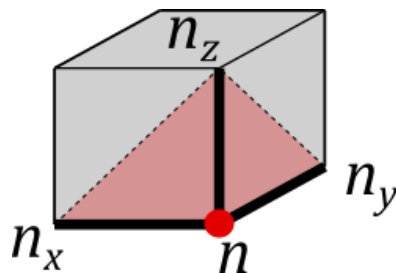


Figure 3.38 – Le coin du sommet de bloc n qui regroupe n_x , n_y et n_z .

La notion de voisinage étant topologique, nous préférons éviter les méthodes géométriques qui peuvent amener des erreurs numériques. Un choix analogue est effectué dans [CBK12, ZWG⁺18]. On a défini que deux

sommets de blocs sont considérés comme voisin si une arête de bloc les relie, or si on se réfère aux relations des cellules duales vues à la Section 2.1.1 on sait que si deux régions duales \mathbf{r}_1 et \mathbf{r}_2 sont adjacentes alors il existe une arête du maillage primal qui relie les sommets primaux n_1 et n_2 , qui correspondent respectivement à \mathbf{r}_1 et \mathbf{r}_2 . Nous proposons alors l'Algorithme 5 pour créer les coins de bloc. Cet algorithme prend en entrée B , un ensemble de huit sommets répertoriés pour créer un bloc primal et $f_{0,1,2}$ les trois feuillets duaux qui forment le nœud dual correspondant au bloc en question. Pour chaque sommet n dans B nous allons alors créer le coin $corner_n$. Pour cela, nous parcourons les tétraèdres t du maillage T_Ω marqués comme faisant parti de la région duale \mathbf{r}_n , correspondant au nœud n , et testons si t est en contact avec l'un des feuillets de $f_{0,1,2}$, c'est-à-dire qu'une des faces de t est adjacente à un tétraèdre t_f de $f_{0,1,2}$ (ligne 6 de l'Algorithme 5). Si t est en contact avec l'un des feuillets, alors nous allons récupérer la région duale adjacente. La Figure 3.39 montre une illustration de cette étape. En (a), nous pouvons voir les faces en rouge des tétraèdres t au bord de la région duale **A**. Pour récupérer la région duale voisine nous parcourons les sommets des tétraèdres t_f et si l'un est incident à un tétraèdre marqué avec une autre région duale alors nous insérons l'indice de cette région dans l'ensemble $corner_n$. Cette étape est exposée en (b) où des sommets des tétraèdres t_f sont en contact avec les régions duales **B** et **D** tandis qu'aucun sommet n'est en contact avec la région duale **C**. Sur cet exemple, nous savons alors que le sommet correspondant à la région duale **A** est voisin des sommets correspondant aux régions duales **B** et **D**. Cet algorithme nous permet d'obtenir les huit coins topologiques du bloc à créer.

Algorithme 5 : *Création d'un coin pour un sommet donnée.*

```

Data :  $(B, f_{0,1,2})$ 
Result :  $corners_B$ 
1  $corners_B \leftarrow \emptyset;$ 
2 foreach node  $n \in B$  do
3    $\mathbf{r}_n \leftarrow \text{getDualRegion}(n);$ 
4    $corner_n \leftarrow \emptyset;$ 
5   foreach tet  $t \in \mathbf{r}_n$  do
6     if  $\text{isAnInterface}(t, f_{0,1,2})$  then
7        $\text{insert getNeighborRegion}(t)$  into  $corner_n;$ 
8     end
9   end
10   $\text{insert } corner_n$  into  $corner_B$ 
11 end
12 return  $corner_B;$ 

```

Ordonnancement combinatoire des coins

Maintenant que nous avons défini les informations des huit coins d'un bloc primal, nous allons pouvoir ordonner localement les sommets pour générer celui-ci. Il existe plusieurs méthodes de combinaison de sommets pour créer des hexaèdres. Cependant ces méthodes prennent en entrée un maillage tétraédrique et utilisent des combinaisons de tétraèdres pour ordonner les sommets et former des hexaèdres ([PJR17]). Notre approche diffère totalement. L'étape précédente nous a fourni les informations de voisinage topologique nécessaires à la construction des blocs hexaédriques et nous pouvons voir les informations de *coin* non pas comme un tétraèdre mais comme un ensemble de trois triangles, un sur chaque face de bloc comme le montrent les triangles t_0, t_1 et t_2 sur la Figure 3.40.

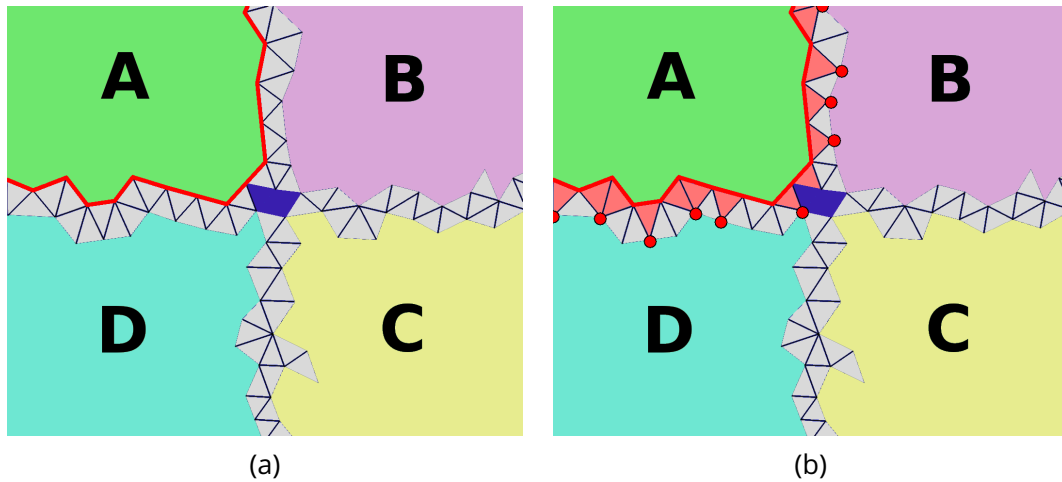


Figure 3.39 – Ici nous cherchons les régions duales voisines de la région duale **A**. En (a) nous atteignons le bord (en rouge) de la région duale en contact avec les feuillettes duales et en (b) nous traversons ces feuillettes duales pour entrer en contact avec les régions duales voisines.

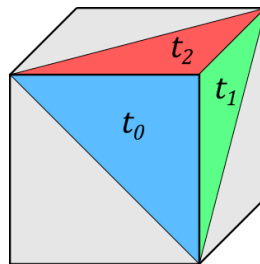


Figure 3.40 – Un coin de sommet de bloc peut aussi être vu comme un ensemble de trois triangles au bord du bloc.

Or si nous regardons la Figure 3.41 le nombre de combinaisons possible de triangles pour représenter la face ABCD n'est que de deux (montrés à gauche). En supposant que nous appliquons notre approche de *coins* aux sommets A,B,C et D nous obtenons le résultat en (a). Nous choisissons arbitrairement un sommet, ici A, nous avons ses deux voisins B et D, nous obtenons le triangle ABD, pour obtenir le deuxième triangle de la face il nous suffit de trouver le voisin commun à B et D qui n'est pas A. Nous avons donc en (b) le deuxième triangle BCD. Pour tout découpage en deux triangles ABD et BDC nous obtenons la face ordonnée ABCD.

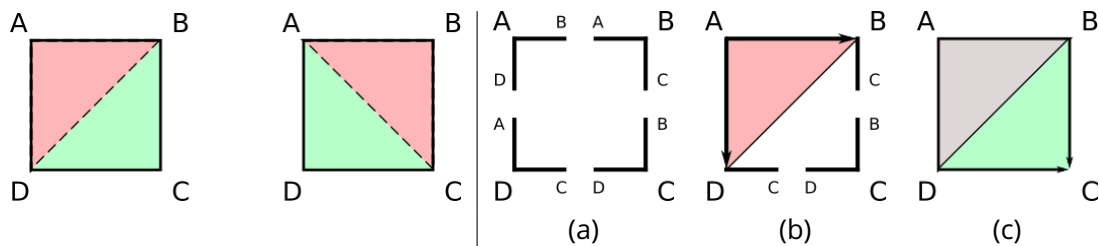


Figure 3.41 – A gauche, les deux combinaisons de triangles sur une face quadrangulaire. A droite le déroulé de la méthode d'ordonnement des sommets d'une face quadrangulaire.

Nous proposons alors d'appliquer cette méthode pour retrouver l'ordre des sommets en suivant l'exemple de la Figure 3.41 :

1. Notre groupe de sommets A,B,C,D,E,F,G et H à ordonner ;
2. Les *coins* sont calculés pour chaque sommet en utilisant la méthode illustré à la figure 3.39 ;
3. Nous choisissons, de manière arbitraire, un sommet, ici A, ainsi que deux voisins, ici C et E ;
4. Nous appliquons la méthode combinatoire vue précédemment pour ordonner quatre sommets en une face et ainsi obtenir la face AEFC ;
5. Nous construisons la face opposée, la différence ici est que nous pouvons directement ordonner les sommets. En effet, les voisins des sommets A, E, F et C non sélectionnés vont former la face opposée et nous les ordonnons en suivant l'ordre de la face AEFC. Ceci nous permet d'obtenir la face DHBG ;
6. Nous pouvons alors créer le bloc hexaédrique AEFCDHBG.

Exemple génération de structure de blocs par insertion de feuillets duaux

L'approche proposée dans ce chapitre nous permet de générer une structure de blocs valide pour des modèles de type V. Plusieurs résultats ont été montrés dans l'ensemble du chapitre, dont celui que nous présentons ci-après à la figure 3.43. Nous pouvons voir les trois grandes étapes de notre approche avec en (a) l'ensemble des feuillets duaux tracés de la section 3.2, en (b) les régions duales générées et validées par l'ensemble des règles de la section 3.3 et enfin, en (c), les blocs générés par notre méthode présentée à la section 3.4. Une liste plus importante de résultats obtenus en utilisant notre méthode génération de structure de blocs hexaédriques sera proposée au chapitre 5.

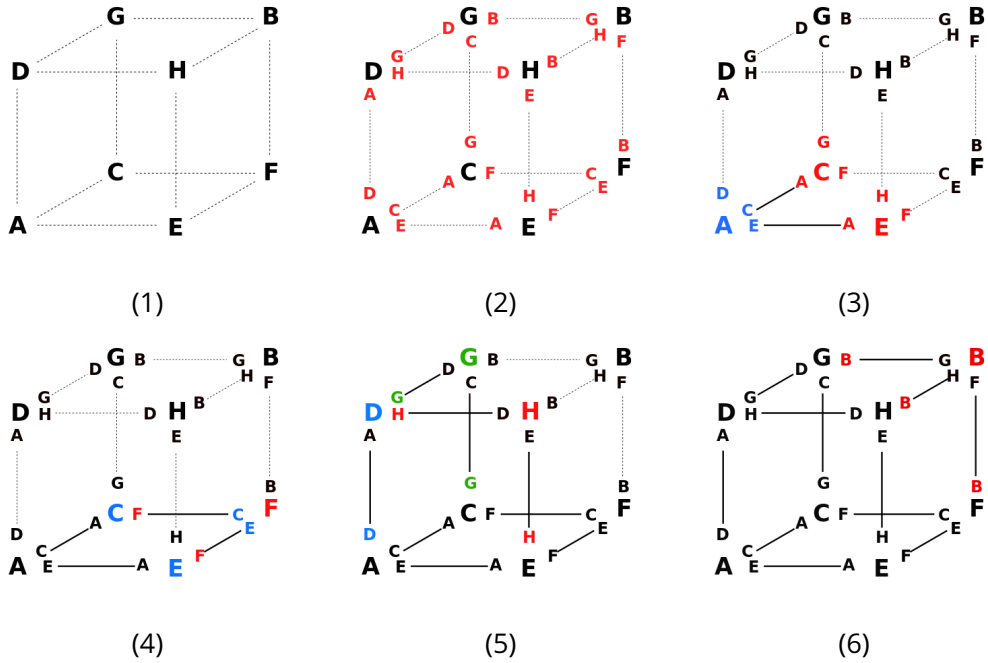


Figure 3.42 – Exemple d'exécution de la méthode d'ordonnancement des sommets de blocs sur un exemple 3D.

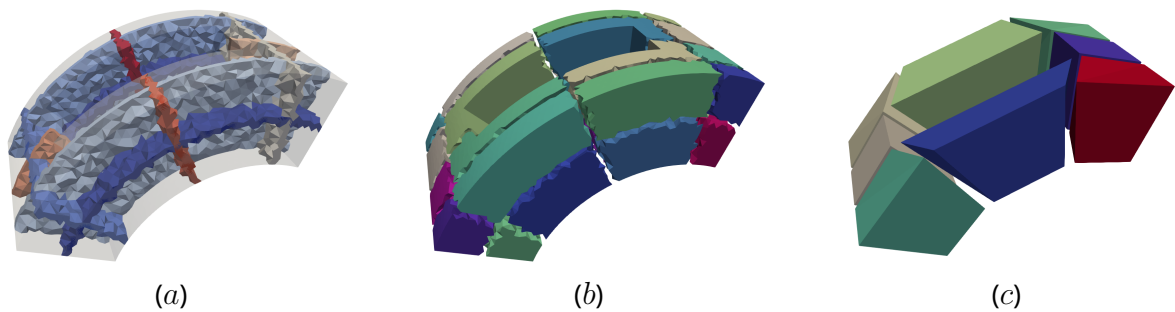


Figure 3.43 – Résultat de notre méthode de génération de structure de blocs. En (a) les feuilletts duaux, en (b), les régions duales et en (c) les blocs hexaédriques.

4 - Analyse et correction du champ d'orientations

Les champs d'orientations aujourd'hui utilisés pour la génération de maillages hexaédriques ne correspondent que trop rarement à la structure topologique souhaitée. En particulier, nous nous retrouvons rapidement confrontés à la présence des configurations de type S et T mentionnées précédemment dans ce manuscrit, à savoir des lignes singulières de type 3-5 ou des zones de tremplins (voir la figure 4.1 pour un exemple de modèle contenant les deux configurations).

Afin de pouvoir appliquer l'approche proposée au chapitre 3 pour n'importe quel domaine géométrique, nous proposons dans ce chapitre de corriger le champ d'orientations pour qu'il remplisse les critères souhaités. Cette correction est une nouvelle fois guidée par l'utilisateur. Après avoir précisé la problématique à traiter à la section 4.1, nous indiquons à la section 4.2 comment identifier algorithmiquement les lignes singulières 3-5 et les zones de tremplin avant de proposer deux approches pour les supprimer aux sections 4.3 et 4.4.

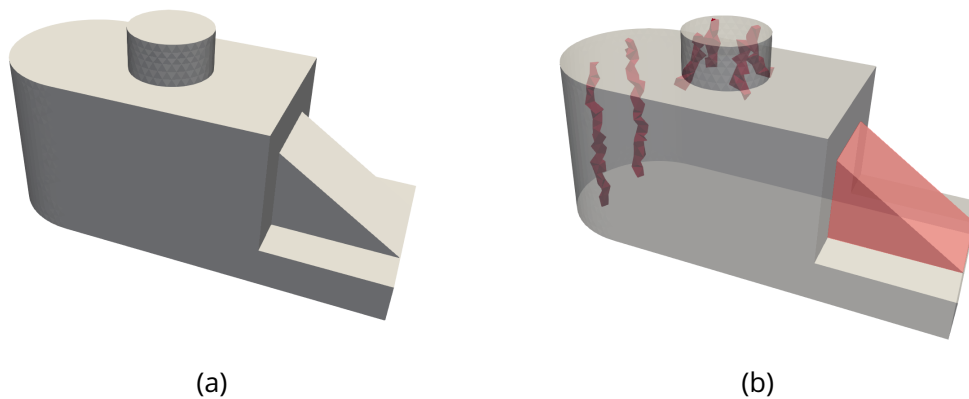


Figure 4.1 – Un exemple de modèle où le champ contient quatre lignes singulières 3-5 - dans le cylindre collé en partie supérieure - ainsi qu'une zone de tremplin marquée en rouge sur le modèle. Ce modèle nous servira d'exemple pour illustrer notre méthode dans ce chapitre.

En résumé

La méthode proposée au chapitre 3 requiert de disposer d'un champ d'orientations dont la structure topologique est en adéquation avec celle d'un maillage hexaédrique. Ceci n'étant fréquemment pas le cas, nous proposons dans ce chapitre deux approches, une nouvelle fois basées sur l'intervention humaine, pour modifier le champ d'orientations en conséquence. Nous constatons que nous sommes alors capables d'éliminer des configurations problématiques de lignes 3-5 et de zones de tremplin. Pour cela, nous nous sommes appuyés sur la découpe du maillage tétraédrique initial et sur des générations successives de champs d'orientations en relâchant des contraintes au bord.

En résumé, les contributions principales de ce chapitre sont de fournir :

- un algorithme d'analyse de modèles 3D et de détection des cas problématiques ;
- une méthode de correction locale du champ d'orientations.

4.1 . Présentation du problème

L'un des avantages importants des champ d'orientations pour le maillage hexaédrique est que les méthodes de génération de ces champs prennent en compte l'alignement avec le bord de la géométrie, comme

indiqué à la Section 2.2. Cependant ces méthodes de génération ne garantissent pas d'obtenir un champ dont la topologie corresponde à celle d'un maillage hexaédrique semi-structuré. De tels cas sont montrés à la Figure 4.2 où des géométries ont été discrétisées par des maillages mixtes mêlant hexaèdres et tétraèdres. Nous pouvons voir que certaines zones du domaine ont été maillées avec des tétraèdres rouges, ce sont en l'occurrence les zones où le champ d'orientation ne correspond pas à une grille régulière.

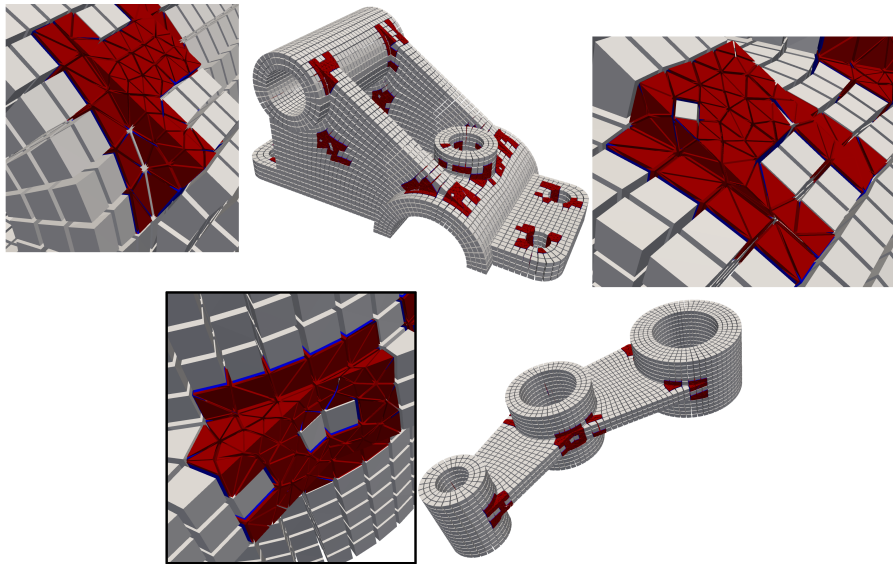


Figure 4.2 – Deux exemples de maillages hexa-dominants avec en rouge les tétraèdres générés dans les zones où le champ d'orientation ne permet pas de créer des hexaèdres.

S'il n'est pas rare dans un modèle qu'une zone du champ d'orientation ne corresponde pas à une grille régulière (Figure 4.2), de telles zones n'apparaissent toutefois pas de façon aléatoire, il existe des formes géométriques à l'origine de ces zones. En étudiant ces formes nous pouvons voir qu'elles impactent le champ d'orientation de deux manières différentes en créant (1) les *lignes singulières 3-5* et (2) les *zones de trempin*.

Lignes singulières 3-5

Intuitivement les lignes singulières 3-5 sont des lignes singulières dont une extrémité est une singularité de valence 3 et l'autre extrémité est une singularité de valence 5 comme le montre la Figure 4.4 où s est une ligne singulière qui a pour extrémité p_3 , une singularité 3 et p_5 une singularité 5. Ces lignes singulières apparaissent dans une configuration particulière mettant en œuvre trois surfaces géométriques x , x_1 et x_2 , telles que :

- x soit adjacente respectivement à x_1 par la courbe c_1 et à x_2 par la courbe c_2 ;
- Sur la surface x_1 , c_1 est une courbe concave, ce qui crée une singularité de valence 5 ;
- Et sur la surface x_2 , c_2 est une courbe convexe, ce qui crée une singularité de valence 3.

De telles configurations géométriques sont montrées à la Figure 4.3.

Les lignes singulières 3-5 posent un problème pour les méthodes de génération de maillages hexaédriques car elles ne correspondent pas à un ensemble conforme de grilles régulières le long de la ligne singulière. Pour illustrer cela, considérons l'exemple de la Figure 4.4. En (a) nous voyons la ligne singulière 3-5 S dans le volume, le point singulier p_5 est une singularité de valence 5. Autrement dit, dans le maillage hexaédrique, ce point devrait être voisin à cinq hexaèdres. De la même manière, le point singulier p_3 est une singularité de valence 3, et ce point devrait être voisin de trois hexaèdres. Lors de la génération d'un maillage hexaédrique (ou dans notre cas d'une structure de blocs) nous voulons que chaque ligne singulière du champ soit placée sur un chemin d'arête du maillage. Si nous supposons qu'il existe une arête S sur ce chemin tel que l'arête

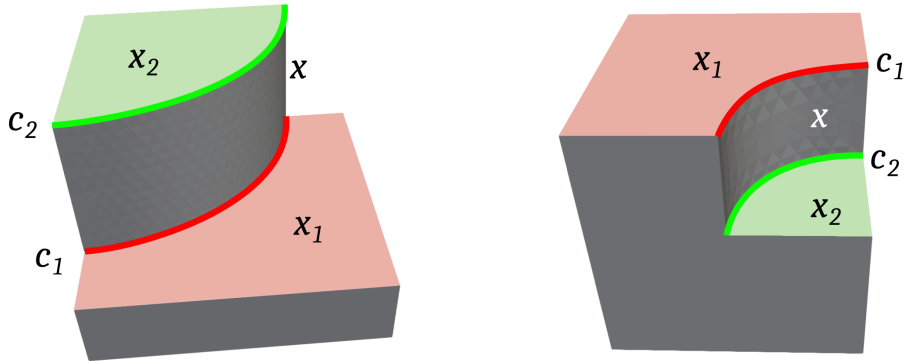


Figure 4.3 – Deux exemples de configurations géométriques menant à l'apparition de lignes singulières 3-5 le long de la surface x dans le champ d'orientation, une singularité de valence 5 va se former sur la surface x_1 et une singularité de valence 3 va se former sur la surface x_2 .

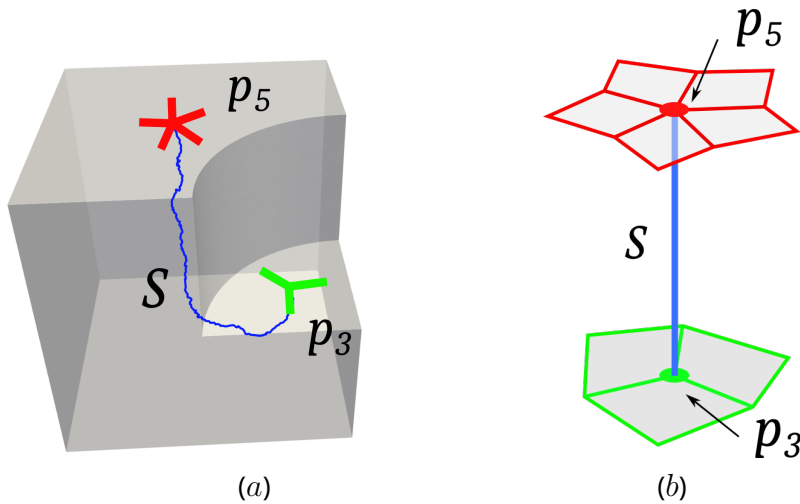


Figure 4.4 – Exemple de ligne singulières 3-5 s avec comme extrémités le point singulier p_3 , une singularité de valence 3 et le point singulier p_5 , une singularité de valence 5. En (a) la ligne singulière dans le modèle et en (b) une représentation topologique de s .

avant S soit adjacente à 5 hexaèdres et celle après S ne soit adjacente qu'à 3 hexaèdres (en b), alors il doit exister une configuration d'hexaèdres permettant cette transition autour de S . Dans [LZC+18], les auteurs listent l'ensemble des configurations acceptables autour d'une arête pour un maillage hexaédrique et aucune ne correspond à cela. Et si l'on étend la transition du passage de 5 vers 3 à plusieurs arêtes, ce n'est pas le cas non plus.

Pour le problème simple académique illustré sur la figure 4.5- a , on a une unique ligne de singularités 3-5, 3 étant la valence de la singularité sur la surface intermédiaire et 5 celle de la singularité sur la surface du dessus. Ce que nous souhaitons en pratique obtenir est le résultat de la figure 4.5- b , où les lignes singulières sont dissociées en deux lignes traversant le domaine complet du haut vers le bas. Le champ d'orientation aurait alors la structure permettant de générer un ensemble de grilles régulières.

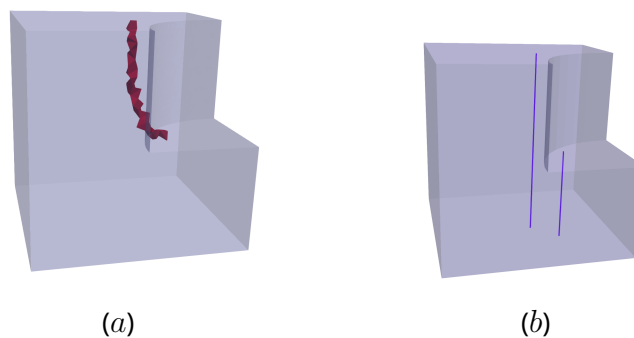


Figure 4.5 – Un cas de graphe de singularité avec une ligne singulière 3-5 entre les points A et B (a) et en (b) la ligne singulière 3-5 est séparée en deux pour former une ligne singulière de valence 3 et une ligne singulière de valence 5.

Les zones de tremplin

De façon usuelle nous appelons *tremplin*, (ou *ski jump* en anglais), des configurations géométriques ressemblant à des rampes de saut comme illustré à la Figure 4.6. A la Figure 4.6(c) nous pouvons voir un domaine géométrique possédant des tremplins. Les zones en rouges sont des **zones de tremplin**, ce sont des zones du domaine formant un prisme à base trois et pour lesquels les champs d'orientations sont totalement réguliers et "voient" le prisme à base 3 comme étant un pavé droit. Si l'on considère le bord du domaine géométrique, les zones de tremplin apparaissent notamment le long de faces triangulaires où lorsqu'un angle d'une surface géométrique est inférieur à 45° .

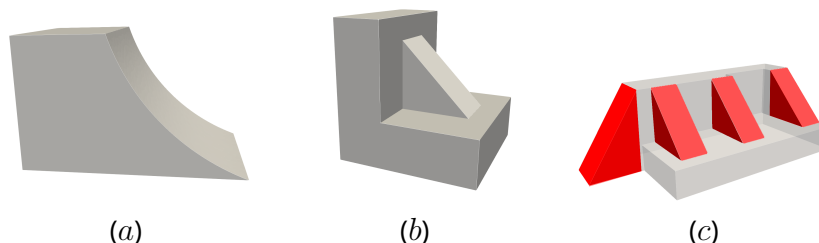


Figure 4.6 – En (a) et (b) deux configurations géométriques provoquant l'apparition de zones de tremplin. En (c) un modèle possédant quatre zones de tremplins en rouge.

La Figure 4.7 va nous permettre d'illustrer en 2D pourquoi le champ d'orientation dans une zone de tremplin ne représente pas une grille régulière conforme sans pour autant qu'une singularité du champ n'apparaisse. Tout d'abord en (a) nous pouvons voir le champ d'orientation obtenu dans un rectangle,

chaque direction du champ est représentée par une couleur, une en rouge et l'autre en bleu. Le champ est aligné au bord du domaine et est par conséquent régulier dans un rectangle. Intéressons-nous à la direction du champ en bleu, celle-ci est orthogonale à la courbe entre les points a et b et donc alignée avec les deux bords orthogonaux à cette courbe. Maintenant si nous déplaçons le point a vers b jusqu'à obtenir un point unique c pour obtenir la géométrie montrée en (b) nous pouvons voir que la direction du champ en bleu a suivi le déplacement du point a et continue à être alignée avec le bord sans qu'aucune singularité ne soit apparue dans le champ. Cela équivaut à écraser un côté de la grille, celle-ci reste régulière mais ne permet plus d'extraire une grille de quadrilatères, des triangles apparaissant nécessairement au voisinage du point c .

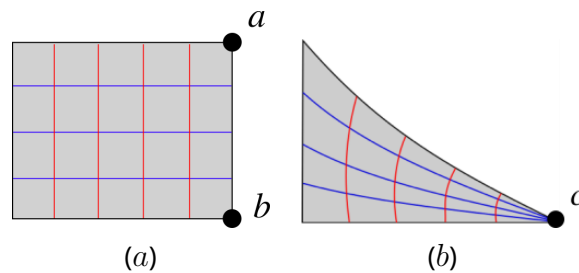


Figure 4.7 – Exemple 2D d'une configuration de type Tremplin. La structure du champ d'orientation est modélisée par deux champs de lignes, de couleur rouge et bleue, qui sont localement orthogonales entre elles et orthogonales au bord géométrique. En (a) , pour un rectangle on obtient une structure de grille, tandis qu'en (b) , la régularité des deux champs induit que la grille rester unique mais est dégénérée au point c .

L'élément problématique des zones de tremplin est la régularité du champ, il représente ici une grille régulière cependant nous ne pouvons pas construire d'hexaèdres conformes proche du point c , ceux-ci auraient alors une face écrasée. Une illustration d'un cas de champ d'orientation compatible avec un structure de blocs pour une zone de tremplin est montrée à la Figure 4.8. En partant d'un champ d'orientation régulier d'une zone de tremplin en (a) , si une singularité est placée au milieu de la surface géométrique triangulaire cela permettrait de séparer le champ d'orientation en un ensemble de grilles régulières conformes.

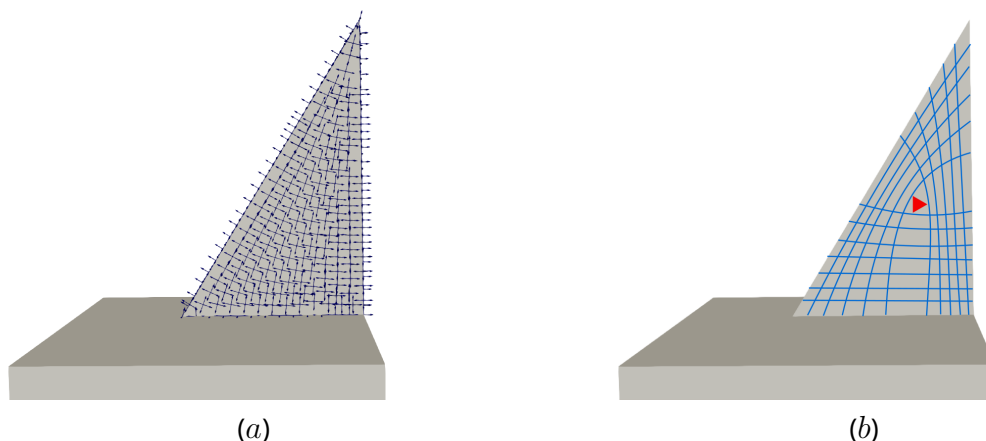


Figure 4.8 – Exemple de champ d'orientation 3D pour un cas élémentaire de zone de tremplin. En (a) le champ généré est bien régulier et valide par rapport aux critères d'alignement au bord et d'orthogonalité locale. On ne peut par contre malheureusement pas en extraire une structure de grille purement hexaédrique. En (b) , l'insertion d'une singularité de valence 3 au centre de la face triangulaire permettrait d'obtenir le résultat souhaité.

Les zones de tremplin sont problématiques pour découper le domaine par des surfaces duales pour deux raisons majeures. La première est fondamentale et liée à ce que l'on vient d'indiquer précédemment. Si

l'on suppose que l'on insère correctement les feuillets, l'un des blocs générés ultérieurement ne sera pas hexaédrique mais prismatique. La seconde, plus technique, est que le tracé de surfaces duales orientées vers la courbe où le champ se "resserre" est vite numériquement compliqué et les résultats ne sont pas ceux escomptés (voir figure 4.9).

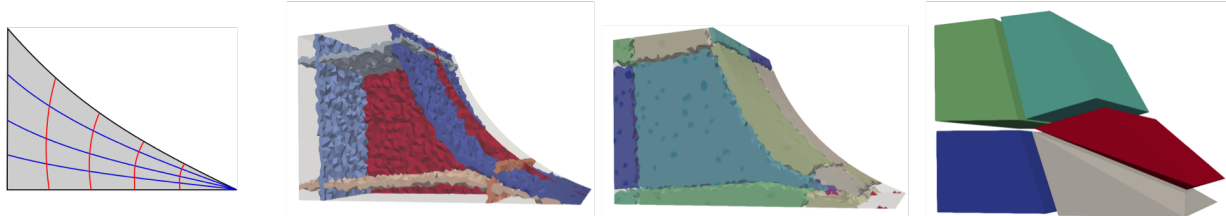


Figure 4.9 – Exemple de tracés de surfaces duales à l'aide de l'approche du chapitre 3 dans une zone de tremplin. De gauche à droite, la vue idéalisée 2D illustrant la problématique du champ d'orientation régulier dans une telle zone; le tracé de surfaces duales; l'extraction de zones duales; en enfin les blocs primaires extraits. La structure de blocs que l'on arrive à obtenir en relaxant des contraintes de validité échoue à capturer la pointe de la zone de tremplin.

Quelles solutions à ces problèmes ?

Les configurations de type S et T précédemment illustrées sont un problème pour notre approche mais aussi pour toute approche souhaitant générer des structures de blocs à partir de champs d'orientation (ou d'ailleurs de la géométrie seule pour d'autres approches). Le découpage du domaine en suivant les caractéristiques du champ d'orientation n'est possible que si la structure du champ est celle d'un maillage hexaédrique. Les configurations de types S et T sont des exemples de situation où cette correspondance n'existe pas.

Puisque nous adoptons une approche interactive, nous pouvons effectuer des choix nous permettant de gérer l'inadéquation du champ avec une structure de blocs. Notre première idée a été de relâcher certaines règles sur les contraintes de validité de la structure duale. Cela nous a permis de gérer localement des configurations de type S mais aucunement celles de type T. Par exemple, sur la figure 4.10, nous générons une structure de blocs valides pour une ligne 3-5 en modifiant nos règles pour séparer les extrémités de la ligne 3-5 dans deux zones (colorées en vert et fuchsia sur la figure), mais accepter que ces zones contiennent une ligne singulière tout en étant adjacentes au bord du domaine. Cela nous permet de générer la structure de blocs présentée sur la droite de la figure.

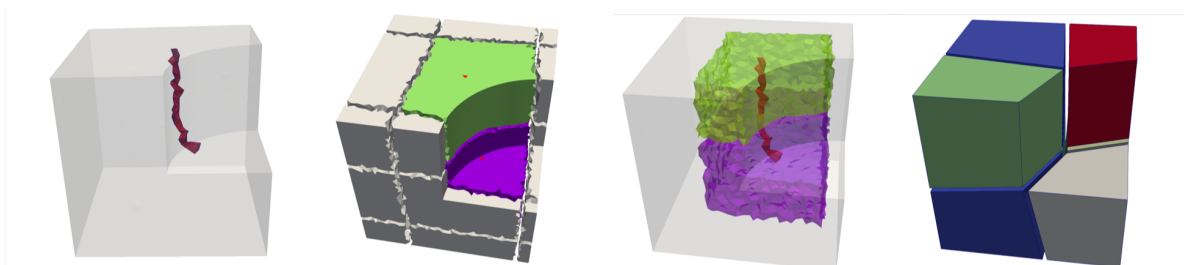


Figure 4.10 – Exemple de tracés de surfaces duales à l'aide de l'approche du chapitre 3 dans une zone de tremplin.

Partant de ce constat, nous avons considéré plusieurs approches pour infléchir le résultat obtenu par l'algorithme de génération de champ. Tout d'abord, nous avons considéré le **découpage du modèle géométrique**. Si on regarde du côté des outils commerciaux pour gérer le cas de modèles complexes, l'approche

est de découper le modèle en sous-parties de façon récursive jusqu'à ce que chaque sous-partie puisse être maillée par une méthode de génération de maillage hexaédrique élémentaire. Dans notre cas, l'idée est de découper le domaine géométrique pour isoler les zones problématiques et les traiter séparément. Prenons l'exemple d'un domaine Ω avec un champ d'orientation \mathbf{F} non valide tel que celui de la figure 4.11, pour lequel nous avons une configuration de type T. En utilisant des opérations interactives on va sélectionner une partie de la géométrie (ici en rouge), et la dissocier du reste du domaine pour obtenir deux sous-parties Ω_0 et Ω_1 . On pourra ensuite générer un champ d'orientation valide sur la partie grise et appliquer un motif pour la zone prismatique. Le recollement final restera alors à être effectué. Le point problématique de cette approche réside dans la découpe interactive du modèle. Si sur ce cas élémentaire, on voit directement quoi faire, ce n'est pas toujours simple. Ensuite, se pose la question de comment faire ce tracé et comment le propager dans le volume pour avoir un coupe qui convienne. C'est l'approche que nous explorons à la section 4.3.

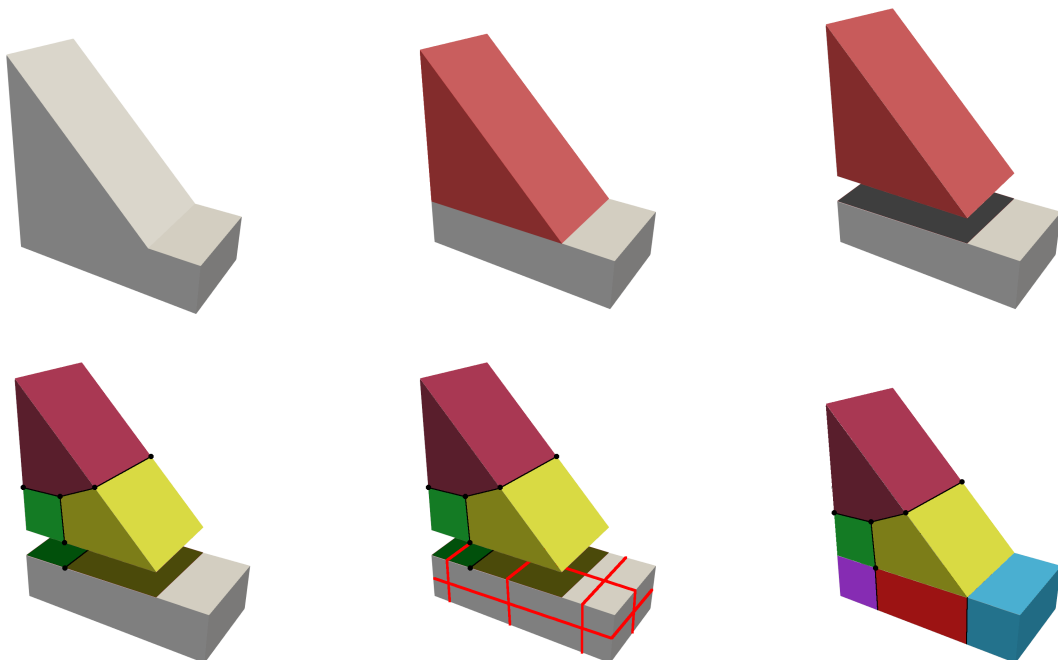


Figure 4.11 – Exemple de découpe interactive que l'on pourrait mener pour dissocier un domaine en deux sous-parties, chacune discrétisée à l'aide d'un algorithme différent : à base de champ d'orientation pour la partie basse et par application de type midpoint subdivision pour la partie haute.

Une seconde option est de modifier la structure du champ d'orientation pour le rendre compatible avec la structure de blocs voulus et cela sans découper la géométrie. Ceci peut être effectué par **ajout de contraintes sur le champs**. C'est par exemple le cas de [ZWG⁺18], où les auteurs proposent une méthode de correction de lignes singulières 3-5 en recalculant un champ de croix au bord du domaine grâce aux informations fournies par le champ d'orientation volumique. Les méthodes mises en oeuvre nécessitent des connaissances mathématiques hors du spectre de notre travail. En considérant la génération du champ d'orientation comme un algorithme de type *boite noire* tel que décrit à la section 3.1, nous allons **relâcher des contraintes sur le champ à générer**. Nous savons que les configurations de type S et T sont le résultat de la contrainte d'alignement au bord. Notre idée est de relâcher localement les contraintes d'alignement au bord lors de la résolution du système de génération de champ d'orientation. Les éléments problématiques apparaissent car le système tente de minimiser le non alignement du champ d'orientation interne avec celui au bord du domaine. Intuitivement relâcher les contraintes d'alignement au bord près des éléments problématiques

laisserait la liberté au champ de se lisser et de s'aligner en une grille régulière. Ce sera le propos de la section 4.4.

4.2 . Analyse d'un champ d'orientations

Nous avons montré que les modèles de type S et T sont problématiques pour notre méthode définie au chapitre 3 car les champs d'orientations générés sur ces types de modèles possèdent des lignes singulières 3-5 ou des zones de tremplin. Afin de déterminer le type d'un modèle nous devons analyser le champ d'orientation et détecter ces deux types de configuration. Nous indiquons dans cette section le travail technique réalisé en pratique pour les détecter.

4.2.1 . Détection des lignes singulières 3-5

La détection des lignes singulières 3-5 repose sur l'analyse locale du champ d'orientation discret \mathcal{F} calculé aux sommets du maillage tétraédrique T_Ω . La détection des lignes commence par qualifier individuellement tous les tétraèdres de T_Ω comme étant *réguliers* ou *singuliers* pour le champ \mathcal{F} . Pour un tétraèdre $t \in T_\Omega$, on évalue pour chacun de ces 4 faces, si celle-ci est traversée par une ligne singulière du champ comme expliqué à la section 3.1.3, page 49 (voir la figure 3.5 en particulier). Si c'est le cas, on considère la projection du champ d'orientation dans le plan défini par la face considérée et on applique la même stratégie que celle définie dans [Kow13].

Pour calculer le degré d'une singularité \mathbf{S} localisée dans un triangle $t(s_1, s_2, s_3)$, on commence par calculer la position exacte de \mathbf{S} . Puis, on parcourt le bord du triangle t , à savoir ses trois arêtes. Pour chaque point \mathbf{P} situé sur une arête $[s_i, s_j]$ du triangle t , on regarde à quel moment le vecteur \mathbf{SP} est aligné avec l'une des direction de l'orientation définie au point \mathbf{P} , orientation interpolée à partir des valeurs aux sommets s_i et s_j (voir figure 4.12). Ce problème peut être résolu numériquement le long de chaque arête pour obtenir 0, 1, ou 2 points d'intersection. En pratique on en aura 3 ou 5 pour les champs d'orientations considérés ici.

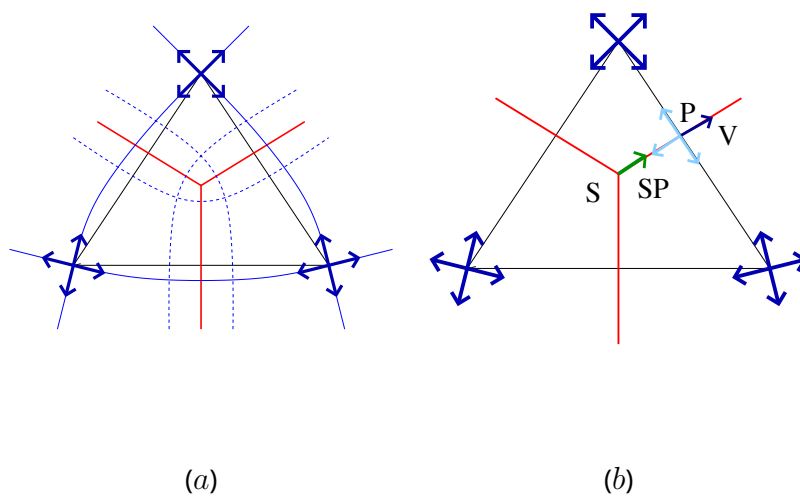


Figure 4.12 – Exemple 2D d'un triangle contenant une singularité de valence 3. En (a), une représentation stylisée des trajectoires suivies par des lignes de champ en traversant un tel triangle. En (b), le degré est calculé en cherchant le long de chaque arête les points \mathbf{P} tel que les vecteurs \mathbf{SP} et \mathbf{V} soient colinéaires.

Remarque : Les différents calculs effectués le sont dans le plan tangent au triangle $t(s_1, s_2, s_3)$ considéré. De ce fait, une étape préalable consiste à aligner les orientations définies aux sommets s_1 , s_2 , et s_3 avec le vecteur tangent \mathbf{n}_t au triangle t . Pour cela, une matrice de rotation minimale transportant le vecteur unitaire de l'orientation $\mathcal{F}(s_i)$ le plus proche de \mathbf{n}_t sur \mathbf{n}_t est calculée et appliquée.

Une fois l'ensemble des tétraèdres désignés comme réguliers ou singuliers, on récupère les ensembles connexes de tétraèdres singuliers et on analyse le type des tétraèdres. Un tétraèdre peut être traversé par une unique ligne singulière, on parle de tétraèdre stable ou 2-singulier - car deux de ses faces sont singulières - ou le lieu de rencontre de plusieurs lignes singulières, et le tétraèdre sera alors qualifié d'instable ou de 3 ou 4-singulier. Si un ensemble connexe de tétraèdres singuliers, noté \mathcal{S}_T , ne contient que des tétraèdres 2-singuliers alors il forme une ligne singulière connectant deux points au bord du domaine ou une boucle interne au domaine (cas d'un tore par exemple). Si par contre, l'ensemble \mathcal{S}_T contient des tétraèdres 3 ou 4-singuliers, alors on le décompose en branches connectant des singularités au bord avec des singularités internes du domaine (comme pour la sphère). Dans le cas où \mathcal{S}_T forme une ligne de tétraèdres 2-singuliers connectant deux points du bord, on analyse le degré des singularités dans les triangles au bord. Ils peuvent être tous les deux de degré 3 ou 5 ou alors de degrés différents ce qui nous permet de classer la ligne singulière en question comme étant de type 3-5 et détecter une configuration de type S.

4.2.2 . Détection des zones de tremplin

La détection des zones de tremplin se fait aux sommets géométriques des surfaces prises individuellement en analysant le champ d'orientation sur la surface au voisinage de ses points (voir les zones rouges sur la figure 4.13). Nous voulons détecter dans cette zone si le champ d'orientation va converger vers le sommet géométrique concerné. Pour ce faire nous allons comparer les directions des orientations sur les courbes géométriques bordant la surface considéré et adjacent au point en question. Considérons le cas de la figure 4.13 où le point p est un coin de la surface bleutée et c_1 et c_2 sont les deux courbes géométriques incidentes à p bordant la surface en question.

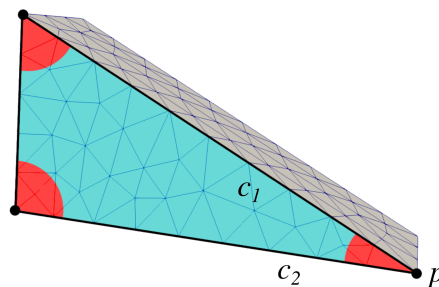


Figure 4.13 – Un exemple de modèle sur lequel nous détectons une zone de tremplin. La surface bleutée en premier plan va être analysée et plus précisément le champ d'orientation au voisinage des sommets géométriques représentés par les zones en rouges.

Pour analyser le champ d'orientation au voisinage de p sur c_1 et c_2 nous utilisons les nœuds du maillage T_Ω , classifiés sur le modèle géométrique. Soit n_1 et n_2 respectivement les nœuds classifiés sur c_1 et c_2 , tels que n_1 et n_2 sont adjacents à p . La Figure 4.14 montre deux telles configurations. Notons \mathcal{F}_1 et \mathcal{F}_2 les orientations définies localement aux nœuds n_1 et n_2 . Nous allons maintenant récupérer la direction des orientations \mathcal{F}_1 et \mathcal{F}_2 qui sont localement tangentes avec les courbes géométriques. Autrement dit, pour \mathcal{F}_1 on cherche le vecteur de direction \mathbf{d}_1 parmi le trois directions de \mathcal{F}_1 dont le produit scalaire avec le

vecteur $\mathbf{n}_1\mathbf{p}$ est supérieur à ϵ . La même opération est effectuée pour \mathcal{F}_2 . Les vecteurs $\mathbf{n}_1\mathbf{p}$ et $\mathbf{n}_2\mathbf{p}$ sont les vecteurs normaux aux courbes c_1 et c_2 respectivement et vivant localement dans le plan tangent à la surface considérée (on utilise pour cela les triangles au bord incident aux nœuds n_1 et n_2). Sur les figures 4.14-a et 4.14-c, on a en vert la direction \mathbf{d}_1 tangente à c_1 et en rouge la direction \mathbf{d}_2 tangente à c_2 .

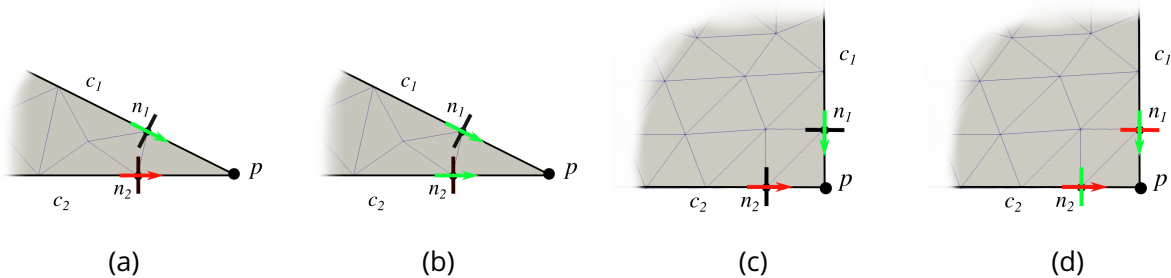


Figure 4.14 – Deux zones proches d'un sommet géométriques analysées avec notre méthode. En (a) et en (c) sont montrées les directions du champ d'orientation tangentes aux courbes géométriques. En (b) et en (d) ces directions sont comparées pour détecter le comportement du champ d'orientation au voisinage du sommet géométrique p . Dans le premier cas, en (b), les directions du champ tangentes aux courbes sont considérées alignées, ce qui induit une zone de tremplin tandis que dans le deuxième cas, en (d), les directions ne sont pas alignées.

Maintenant que nous avons les directions du champ d'orientation tangentes pour chaque courbe géométrique on va pouvoir "comparer" ces directions. Pour cette étape nous sélectionnons arbitrairement la direction \mathbf{d}_1 et recherchons la direction \mathbf{d} de \mathcal{F}_2 la plus alignée avec \mathbf{d}_1 . Si $\mathbf{d} = \mathbf{d}_2$ alors cela signifie que les directions du champ d'orientation entre c_1 , c_2 et p pointent vers p . Si nous reprenons l'exemple de la Figure 4.14, nous voyons en (b) et (d) l'étape de comparaison du champ d'orientation local à n_1 et n_2 pour deux cas différents, le premier est un cas où p est considéré comme un point *non valide* qui induit la détection d'une zone de tremplin, tandis que dans le second cas, p est un point valide.

4.3 . Découpage du domaine géométrique

Comme expliqué à la section 4.1, la génération du champ d'orientation sur de nombreux modèles fait apparaître des éléments problématiques qui perturbent localement le tracé de surfaces duales. À partir d'un domaine Ω et d'une discrétisation T_Ω possédant un champ d'orientation \mathcal{F}_Ω , nous allons déterminer de quel type est Ω en appliquant les méthodes d'analyse du champ d'orientations de la Section 4.2. Si Ω est de type V alors nous pouvons directement utiliser notre méthode interactive de création de structure duale (Chapitre 3), sinon nous souhaitons découper la géométrie en sous-domaines afin d'obtenir un ensemble de sous-domaines de type V et les traiter séparément.

L'objectif de l'étape présentée ici est de pouvoir découper le domaine Ω en limitant le nombre d'actions réalisées par l'utilisateur. Pour un domaine Ω telle que celui illustré sur la figure 4.15-a, la seule action à effectuer par l'utilisateur est la sélection du bord de la "zone" à découper (en rouge sur la figure 4.15-b). Une fois cette opération terminée, le processus de coupe est automatique et effectué sur le maillage T_Ω . Celui-ci est découpé et chaque partie est traitée pour reconstruire un bord géométrique cohérent (voir la figure 4.15-c pour la partie non prismatique). Une structure duale peut être construite dans cette partie, et un motif peut être appliqué dans la seconde partie pour ensuite envisager de recoller les deux parties ensemble. L'approche que nous proposons se déroule de la façon suivante :

1. L'utilisateur sélectionne une face triangulaire $f \in T_\Omega$ au bord du domaine. Puis nous utilisons la classification de f pour obtenir la surface géométrique à laquelle f appartient. Cette opération est

répétée jusqu'à obtenir un ensemble de surfaces géométriques qui seront tout ou une partie du bord de la zone à découper ;

2. Un algorithme de *MinCut-MaxFlow* [BK04] est appliqué pour créer une coupe dans le volume en respectant le bord sélectionné par l'utilisateur ;
3. Comme le volume est coupé, nous reconstruisons alors le bord du domaine géométrique le long de la coupe, cela équivaut à re-calculer la classification des cellules sur la coupe.

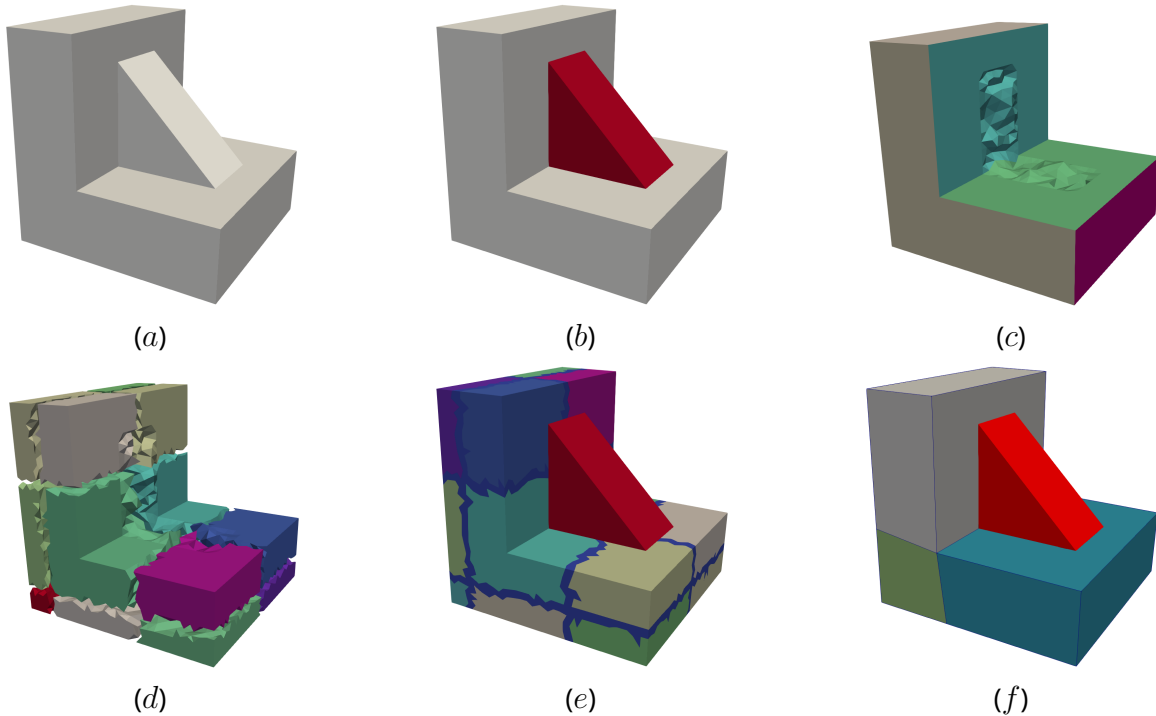


Figure 4.15 – Exemple de traitement d'un domaine de type T (a). Le bord de la zone à découper est sélectionné (en rouge) (b), pour obtenir une nouvelle géométrie qui représente un domaine de type V (c). Une structure duale est construite sur la nouvelle géométrie (d, e). La structure de blocs générée devrait alors être recollée avec celle du prisme rouge (f).

4.3.1 . Partitionnement dirigé par le champ d'orientation pour la coupe géométrique

Dans notre approche nous proposons à l'utilisateur de sélectionner des surfaces géométriques au bord pour guider un algorithme de MinCut-MaxFlow afin de trouver une coupe topologique dans le volume en partitionnant le maillage T_Ω . Cette méthode permet d'éviter de faire une coupe géométrique qui reposeraient sur le noyau géométrique d'un moteur de CAO.

Initialisation du problème de partitionnement

Une fois que la zone à retirer est sélectionnée, nous pouvons déclencher l'algorithme de coupe volumique. Nous voulons trouver dans le maillage T_Ω un ensemble connexe de faces qui sépare T_Ω en deux sous-ensembles tel que les triangles de la zone sélectionnée appartiennent à un sous-ensemble, et les autres triangles du bord à l'autre sous-ensemble. En sachant cela, nous pouvons transposer notre problème de coupe volumique en un problème de coupe dans un graphe. En l'occurrence, nous construisons le graphe $\mathcal{G} = (V, E)$, avec :

- V l'ensemble des sommets v_i du graphe tel que chaque sommet v_i correspond au tétraèdre $t_i \in T_\Omega$;
- E l'ensemble des arêtes e_i du graphe telle que chaque arête

$$e_i = (v_x, v_y) | (v_x, v_y) \in V \times V \wedge x \neq y \wedge \exists f_i \text{ tel que } f_i \text{ est la face commune à } t_x \text{ et } t_y;$$

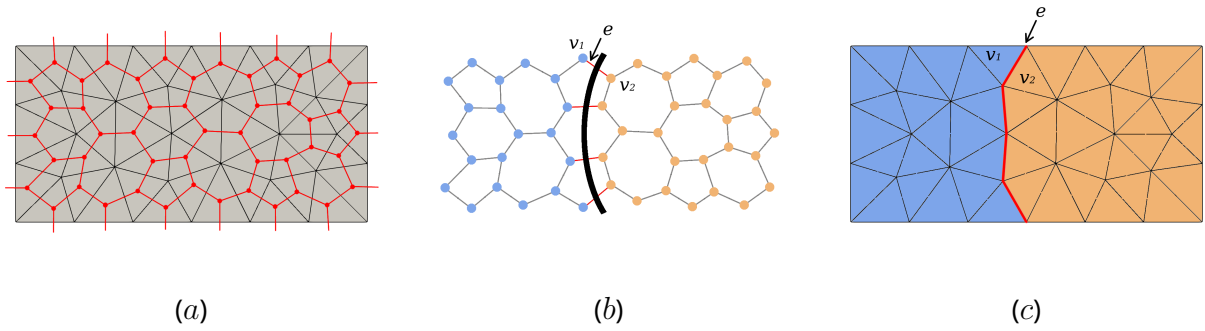


Figure 4.16 – En (a), un maillage triangulaire et son graphe dual (en rouge) sont représentée; En (b), le partitionnement en 2 parties des sommets du graphe où les arêtes reliant des sommets des deux parties sont coupés par la courbe noire; En (c), le partitionnement équivalent dans le maillage triangulaire. La courbe qui coupée des arêtes du graphe en (b) est désormais un chemin d'arêtes dans le maillage.

Trouver une coupe dans \mathcal{G} revient à trouver une coupe volumique dans le maillage. Nous reportons sur le graphe \mathcal{G} la contrainte de séparation des triangles au bord entre les deux parties à calculer. Pour cela, nous marquons d'une part tous les tétraèdres qui ont une face en contact avec une face non sélectionnée à 1, et d'autre part tous les tétraèdres qui ont une face en contact avec les surfaces géométriques sélectionnées à 2. Les tétraèdres restants, internes au maillage, sont eux marqués à la valeur 0. Ce marquage des tétraèdres est reporté sur les sommets correspondants de \mathcal{G} .

De plus nous souhaitons que la coupe suive le champ d'orientation \mathcal{F}_Ω . Nous affectons donc à chaque face f_i du maillage triangulaire T_Ω une valeur \mathbf{a}_i qui correspond à son alignement avec le champ d'orientation. Pour cela, nous interpolons le champ d'orientation au centre de chaque face pour obtenir l'orientation $\mathcal{F}(f_i)$ puis pour chacune des directions $\mathbf{d} \in \mathcal{F}(f_i)$ nous calculons le produit scalaire $\mathbf{d} \cdot \mathbf{n}_i$ où \mathbf{n}_i est le vecteur normale unitaire à la face f_i . La valeur d'alignement de la face f_i est alors définie par :

$$\mathbf{a}_i = \max(|\mathbf{d}_1 \cdot \mathbf{n}_i|, |\mathbf{d}_2 \cdot \mathbf{n}_i|, |\mathbf{d}_3 \cdot \mathbf{n}_i|) \quad (4.1)$$

où $\mathbf{d}_1, \mathbf{d}_2$ et \mathbf{d}_3 sont les trois directions de l'orientation $\mathcal{F}(f_i)$. On obtient ainsi pour chaque face $f_i \in T_\Omega$ une valeur a_i comprise entre 0 et 1 avec 1 comme valeur d'alignement maximale.

Résolution du problème de partitionnement

Nous appliquons alors l'algorithme de Mincut sur le graphe \mathcal{G} pour trouver la coupe minimale qui partitionne les sommets de \mathcal{G} en deux sous-ensembles. Étant donnés les sommets à répartir dans deux parties (ceux marqués 1 et 2 précédemment), l'algorithme va répartir les autres sommets (marqués 0) entre les parties 1 et 2 de telle façon que la coupe entre les parties 1 et 2 soit de poids minimal. Pour obtenir une coupe de poids minimal qui soit en accord avec le champs d'orientation, nous pondérons les arêtes du graphe avec les valeurs d'alignement des faces correspondantes. Nous choisissons pour chaque arête e_i correspondant à une face $f_i \in T_\Omega$, le poids $\omega_i = 2 - \mathbf{a}_i$. Ainsi, une arête de poids faible correspond bien à une face dont la normale est alignée avec le champ. Nous choisissons la valeur numérique 2 pour qu'une arête correspondant à une face parfaitement alignée avec le champ ait une valeur de 1 et non de 0. Si nous définissons une valeur minimale de poids d'arête du graphe à 0, ce qui correspond aux faces parfaitement alignées avec le champ d'orientation, nous pourrions obtenir une coupe avec un poids total de 0 cependant l'ensemble des faces sélectionnées pourrait être trop important et non-cohérente de façon géométrique avec la coupe souhaitée.

4.3.2 . Reconstruction d'un modèle géométrique virtuel

Pour pouvoir appliquer l'algorithme présenté au chapitre 3, nous avons besoin que le maillage T_Ω soit classifié sur un domaine géométrique. Considérons le maillage tétraédrique du domaine Ω , auquel nous appliquons l'algorithme de coupe en sélectionnant la zone au bord en rouge montré à la figure 4.17-*a*. Nous obtenons à la figure 4.17-*b* le maillage du domaine Ω' qui correspond à l'ensemble des tétraèdres de la partie initiale grise vue en (*a*). Toutes les faces triangulaires du maillage situées sur le bord sont classifiées sur une surface géométrique de Ω , ici représenté par des couleurs différentes. Nous pouvons voir un ensemble de faces en bleu foncé, ces faces se situaient initialement à l'intérieur de Ω et se retrouvent, suite à l'opération de coupe, sur le bord de Ω' , nous donnons, pour cet ensemble de faces, le terme d'**interface de coupe**. Les faces étant maintenant sur le bord du domaine Ω' , il est nécessaire de recalculer une classification pour ces faces afin de reconstruire le bord géométrique en introduisant le moins de nouvelles entités géométriques dans le domaine pour obtenir la géométrie la plus simple possible pour notre méthode de création de feuillet duaux du Chapitre 3. Pour cela, nous procédons en plusieurs étapes :

1. Nous allons reconstruire les courbes géométrique en connectant les sommets géométriques du bord de la coupe. Pour effectuer cette étape nous allons utiliser un algorithme de plus court chemin sur les arêtes du maillage T_Ω .
2. Une fois que toutes les courbes sont construites, nous allons reconstruire les surfaces géométriques en analysant les entités géométriques voisines avec le bord de la coupe.

Un élément important de cette opération est la **bordure**, c'est-à-dire l'ensemble des courbes géométriques qui séparent la surface du domaine initial Ω et l'interface de coupe. C'est en utilisant les informations topologiques et géométriques fournies par la bordure que nous pourrons reconstruire le bord géométrique de Ω' .

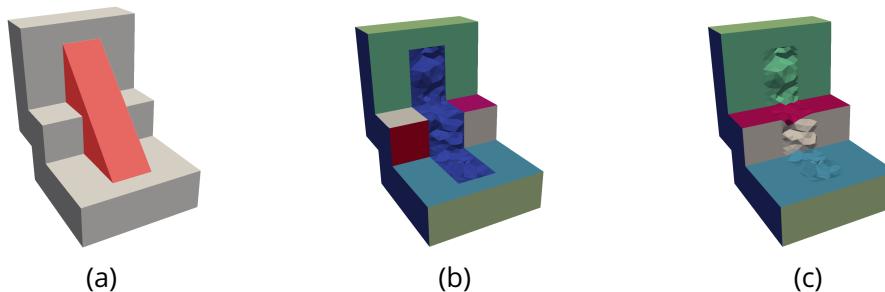


Figure 4.17 – Les différentes étapes de la reconstruction de surface. En (*a*) la zone à découper en rouge. En (*b*) le domaine découpé avec les surfaces géométriques de différentes couleurs et en bleu l'interface de la coupe. Enfin en (*c*) la surface géométrique reconstruite.

Reconstruction des courbes géométriques

Dans le cas simple, l'interface de coupe ne correspond qu'à une seule surface géométrique (figure 4.18(*a*)). Cependant dans la plupart des cas l'interface de coupe correspond à un ensemble de surfaces (figure 4.18(*b*)). Ces surfaces étant séparées par des courbes géométriques, nous allons commencer par reconstruire les courbes géométriques pour nous faciliter par la suite la reconstruction des surfaces. Pour ce faire, nous allons reconnector ensemble des sommets géométriques opposés de la bordure. Nous devons d'abord définir les sommets que nous nommerons *libres*, c'est-à-dire les sommets qui doivent être reconnectés. De manière usuelle, les courbes géométriques à reconstruire sur l'interface de coupe prolongent des courbes déjà existantes du domaine.

Pour déterminer quels sommets sont libres nous calculons l'angle entre les courbes géométriques auxquelles le sommet est incident. Nous appelons *courbes sortantes*, les courbes géométriques d'un sommet qui

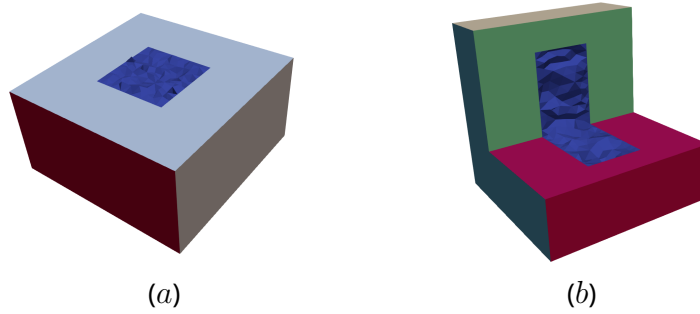


Figure 4.18 – Deux exemples de coupes géométriques. En (a) la reconstruction du bord est triviale, le bord de l'interface de coupe n'est en contact qu'avec une seule surface géométrique. En (b) le bord de l'interface de coupe est en contact avec deux surfaces géométriques séparées par une courbe qu'il faut alors prolonger sur l'interface de coupe.

ne font pas parties de la bordure. Prenons le sommet géométrique s , pour toutes les courbes sortantes c_s de s nous allons calculer l'angle entre c_s et chaque courbe de la bordure c_b dont s est incident. La réalisation de ce calcul s'effectue à l'aide des arêtes du maillage T_Ω . Nous avons n_s le noeud de T_Ω situé sur le sommet géométrique s , ainsi que e_{c_s} et e_{c_b} les arêtes incidentes à n_s et situées respectivement sur les courbes c_s et c_b . Nous calculons alors le produit scalaire $a_{c_s, c_b} = \mathbf{e}_{c_s} \cdot \mathbf{e}_{c_b}$. Cette opération est répétée sur toutes les combinaisons de courbes sortantes et de courbe de la bordure dont s est incident. Si pour une combinaison c_i , une courbe sortante, et c_j , une courbe de la bordure, nous avons $a_{c_i, c_j} < -\frac{\sqrt{3}}{2}$, alors s n'est pas un sommet libre.

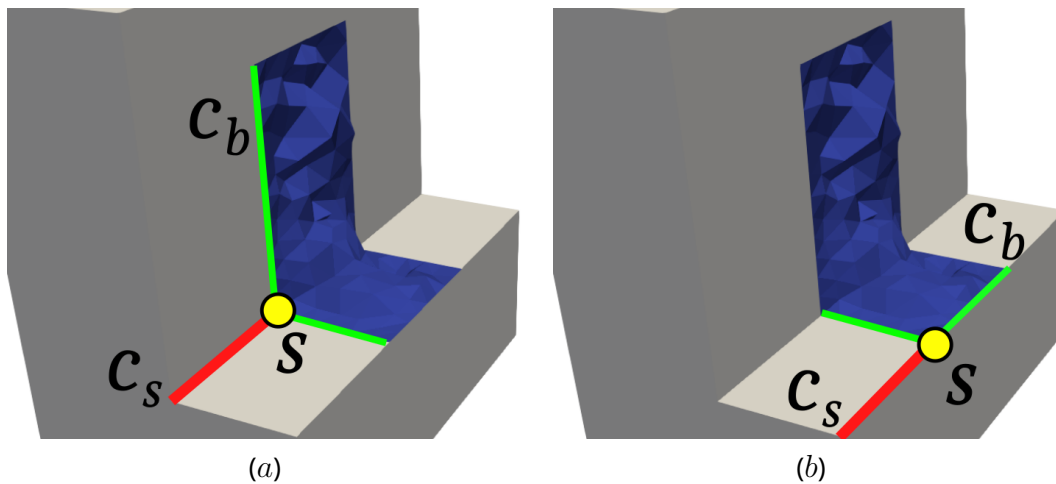


Figure 4.19 – Un exemple d'interface de coupe. Nous testons dans deux cas différents si le sommet s est libre, c'est-à-dire que nous calculons le produit scalaire entre la courbe sortante (en rouge) c_s et chacune des courbes de la bordure (en vert) c_b . En (a), nous pouvons voir que pour c_s forme un angle droit avec chacune des c_b nous avons alors $a_{c_s, c_b} = 0$, dans ce cas là le sommet s est libre. En (b), une des courbes c_b est alignée avec c_s donc nous avons un $a_{c_s, c_b} = -1$, le sommet s n'est donc pas libre.

Nous avons maintenant un ensemble de sommets géométriques à reconnecter entre-eux. Pour cela nous allons utiliser le maillage tétraédrique T_Ω , en sachant qu'une courbe géométrique du domaine qui a pour extrémités les sommets géométriques s_0 et s_1 , correspond à un chemin d'arêtes entre les noeuds n_{s_0} et n_{s_1} dans T_Ω . Nous allons calculer, pour chaque sommet libre s_i , le plus court chemin d'arêtes du maillage entre le noeuds n_{s_i} et tous les noeuds n_{s_k} où s_k est un noeud libre différent de s_i . La figure 4.20 (a) montre les faces triangulaires du maillage T_Ω qui forment l'interface de coupe, le sommet s_i , en bleu, que nous

voulons reconnecter avec un des sommets s_k , en vert. Pour ce faire nous allons utiliser un algorithme de Dijkstra [Dij59] qui permet de calculer, sur un graphe, le plus court chemin entre un sommet source et un sommet d'arrivée, ce qui correspond à notre problématique. Nous considérons l'interface de coupe comme un graphe $G' = (V', E')$ avec :

- V' est l'ensemble des sommets v_i du graphe où v_i correspond au nœud n_i du maillage tel que n_i est incident à une face appartenant à la coupe ;
- E est l'ensemble des arêtes e_i du graphe où $e_i = v_x, v_y$ tel que $(v_x, v_y) \in V$, $x \neq y$ et il existe une arête du maillage entre les nœuds n_x et n_y .

Afin de réduire la taille du graphe nous ne prenons pas en compte les nœuds du maillage ainsi que les arêtes du maillages qui composent les courbes géométriques de la bordure. De plus lors du traitement d'un sommet s_i , nous retirons des sommets d'arrivée de l'algorithme de Dijkstra les sommets libres voisins à s_i , c'est-à-dire tous les sommets s_j tel qu'il existe une courbe géométrique qui à pour extrémités les sommets s_i et s_j (c'est pour cette raison que les nœuds libres voisins à s_i sont en rouge à la figure 4.20).

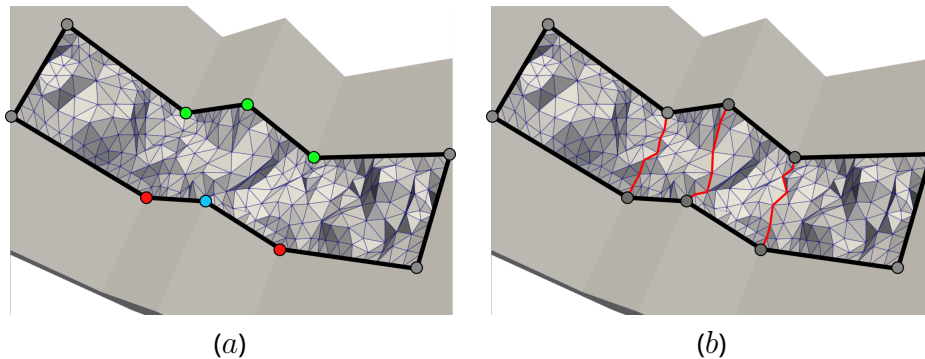


Figure 4.20 – En (a), nous allons reconnecter le sommet libre en bleu avec l'un des sommets libres verts en calculant le plus court chemin composé d'arêtes du maillage en sachant que les sommets libres en rouge ne sont pas pris en compte puisque déjà voisins au sommet à reconnecter. En (b) nous avons en rouge le plus court chemin d'arête entre chaque sommet à reconnecter ce qui crée les nouvelles courbes géométriques.

Afin de créer un bord géométrique dans l'interface de coupe qui soit le plus cohérent avec le bord du domaine déjà existant, nous allons guider l'algorithme de Dijkstra en pondérant les arêtes de notre graphe pour que le chemin d'arête optimal entre deux sommets suive le champ d'orientation. Pour ce faire, nous calculons l'alignement entre le champ d'orientation et les arêtes du maillage, le poids de l'arête étant égal au plus grand produit scalaire entre les trois directions du champ local à l'arête et le vecteur de direction de l'arête noté e . Nous obtenons alors pour l'arête e la pondération $\omega(e)$ suivante :

$$\omega(e) = \max(|\mathbf{d}_1 \cdot \mathbf{e}|, |\mathbf{d}_2 \cdot \mathbf{e}|, |\mathbf{d}_3 \cdot \mathbf{e}|)$$

où \mathbf{d}_1 , \mathbf{d}_2 et \mathbf{d}_3 sont les trois directions du champ d'orientation local à l'arête e . Cette opération nous permet de construire les courbes géométriques sur l'interface de coupe comme illustré à la figure 4.20 (b).

Reconstruction des surfaces géométriques

Nous allons maintenant reconstruire les surfaces géométriques sur l'interface de coupe, autrement dit nous allons changer la classification des nœuds du maillage. Une méthode simple serait de créer de nouvelles surfaces géométriques sur l'interface de coupe cependant nous rappelons que nous voulons obtenir la géométrie la plus simple possible, c'est-à-dire en minimisant le nombre d'entités géométriques nécessaires pour définir le domaine Ω' . Si nous prenons l'exemple de la figure 4.21, où nous coupons la zone en rouge de la géométrie en (a). Nous présentons deux cas de reconstruction du bord géométrique. En (b) une nouvelle

surface a été créée ce qui permet d'obtenir un bord géométrique valide avec deux surfaces ainsi que quatre sommets géométriques. Or intuitivement, nous voyons que l'interface de coupe et le bord existant ne forment qu'une seule surface géométrique comme le montre le cas (c). Pour obtenir le résultat montré en (c) nous allons une nouvelle fois analyser la bordure de la coupe, cette fois ci nous allons extraire des informations géométriques et topologiques des courbes de la bordure.

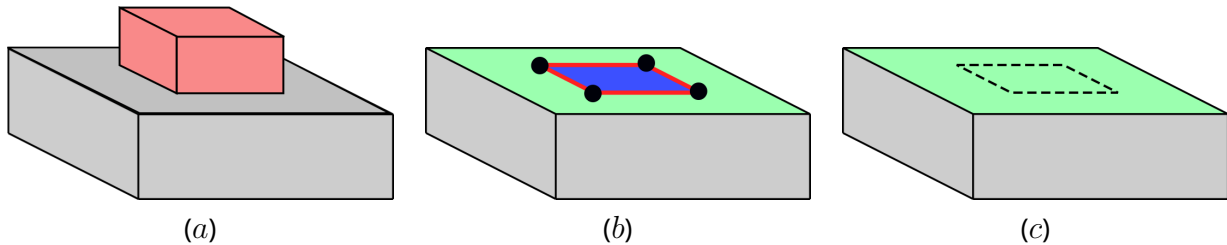


Figure 4.21 – Une illustration d'une coupe géométrique. En (a), la zone rouge va être découpée. En (b), nous reconstruisons une surface sur l'interface de coupe, ce qui fournit un bord valide mais non optimal. En (c), nous reconnectons l'interface de coupe avec le reste du bord de la géométrie pour créer un bord cohérent avec la géométrie obtenue.

Nous allons parcourir les courbes géométriques de la bordure. Pour chaque courbe c nous allons relever deux informations, d'abord la surface géométrique s à laquelle c est incidente puis nous allons définir si l'angle de la courbe c entre s et l'interface de coupe est soit concave, convexe ou "plat". Pour calculer l'angle d'incidence de c nous utilisons les triangles du maillage T_Ω . La courbe c est définie sur T_Ω comme un ensemble d'arêtes E_c . Pour chaque arête e de E_c , nous allons calculer l'angle entre deux faces triangulaires f_s et f_{int} auxquelles e est incidente, f_s est située sur la surface s et f_{int} est située sur l'interface de coupe. Nous calculons alors l'angle a entre \mathbf{n}_s et \mathbf{n}_{int} , respectivement les normales sortantes au domaine de f_s et f_{int} . Nous définissons alors l'arête e comme :

- Concave, si $a \geq 220 \text{ deg}$;
- Convexe, si $60 \text{ deg} \leq a \leq 120 \text{ deg}$;
- Plate, si $a \leq 45 \text{ deg}$;

Une fois cette opération réalisée sur toutes les arêtes e de E_c nous allons définir *conc*, *conv* et *plat* respectivement le nombre d'arêtes concaves, convexes et plates et la catégorie de l'angle d'incidence de c sera défini par $\max(\text{conc}, \text{conv}, \text{plat})$.

Cette étape va nous permettre de déterminer avec quelle surface géométrique du bord de Ω' nous pouvons associer l'interface de coupe. Si une courbe de la bordure de coupe possède un angle d'incidence plat entre l'interface de coupe et la surface s cela signifie que nous avons combiné ces deux éléments en une seule surface. Si plusieurs surfaces peuvent être combinées avec l'interface de coupe alors elles seront toutes combinées en une seule. Prenons l'exemple de la figure 4.22, en (a) nous pouvons voir deux surfaces géométriques verte et rouge séparées par l'interface de coupe, chaque surface possède un angle d'incidence plat avec l'interface de coupe elles vont donc être combinées ensemble. Pour ce faire tous les nœuds du maillage T_Ω situés sur l'interface de coupe et la surface rouge vont être classifiés sur la surface verte ainsi nous obtenons une surface unique montrée en (b).

La reconstruction du bord géométrique de Ω' nous permet de construire une structure duale dans le domaine en utilisant la méthode d'insertion de feuillets du Chapitre 3. Cependant cette méthode de génération de structure de blocs par coupe géométrique, bien que prometteuse puisqu'elle permet d'isoler les éléments problématiques des modèles, possède de nombreuses limitations dont nous discutons plus en détail au chapitre 6. En outre, elle nécessite de nombreux développements informatiques supplémentaires pour fournir des interactions 3D supplémentaires.

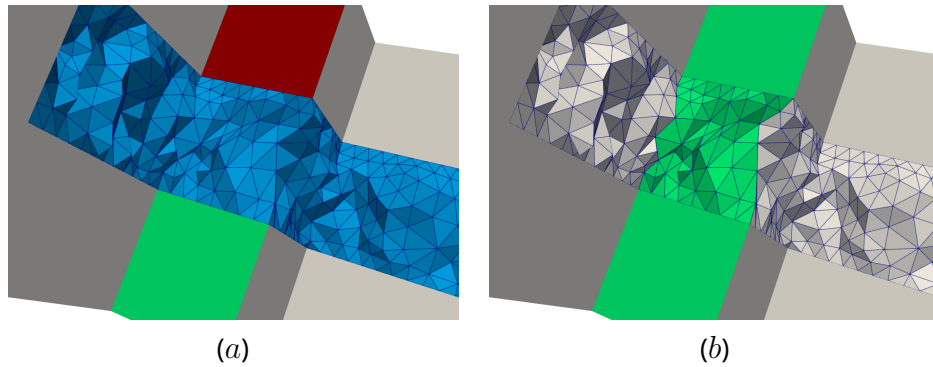


Figure 4.22 – En (a), nous avons deux surfaces, une verte et une rouge, du bord qui étaient séparées par la forme géométrique que nous venons de découper, représentée par l'interface de coupe en bleu. Maintenant que nous avons enlevé la forme géométrique ces deux surfaces se retrouvent en contact par l'interface de coupe, afin de réduire le nombre d'entités géométriques et de garder un bord du domaine simple nous fusionnons ces deux surfaces.

4.4 . Relaxation de contraintes

Dans cette section, nous allons présenter notre méthode de correction de champs d'orientations par relâchement de contraintes d'alignement au bord. En premier lieu nous allons expliquer l'intérêt de l'approche du relâchement de contraintes d'alignement dans nos travaux à la section 4.4.1, puis nous présenterons notre méthode de correction de champs d'orientation à la section 4.4.2.

4.4.1 . Présentation et intérêt de l'approche

Un autre axe de recherche pour corriger les problèmes du champ d'orientation se base sur la modification du champ lui-même. Nous savons que les problèmes du champ sont liés aux contraintes d'alignement comme indiqué à la Section 4.1 et que nous connaissons le champ d'orientation optimal local que nous souhaiterions obtenir.

Une approche envisagée consiste à ajouter des contraintes lors de la génération du champ pour guider le système vers la solution désirée. Une méthode de correction des lignes singulières 3-5 par ajout de singularités au bord est proposée par [ZWG⁺18]. Les auteurs proposent d'utiliser les extrémités des lignes 3-5 pour ajouter de nouvelles singularités de valence 3 et 5 sur le bord du domaine puis de générer à nouveau un champ d'orientations surfacique contraint par ces nouvelles singularités. Cette approche leur permet d'obtenir un champ d'orientation surfacique correspondant à un champ optimal pour une singularité 3-5 comme illustré à la Figure 4.5 (b). Cependant nous souhaitons générer des blocs en utilisant notre méthode donnée au Chapitre 3 et celle-ci requiert un champ d'orientation volumique du domaine et nous n'avons pas de garantie que générer un champ d'orientation volumique à partir du champ corrigé par la méthode de [ZWG⁺18] fournisse un résultat optimal pour notre approche. De plus leur méthode ne permet pas de gérer le cas des zones de tremplins et par conséquent les configurations de type T.

En outre, contrôler le champ d'orientation en ajoutant des contraintes supplémentaires lors de la génération est une opération compliquée. Ces méthodes requièrent en général un travail numérique important afin de fournir le bon ensemble de contraintes permettant d'obtenir le champ d'orientation souhaité et cet ensemble de contraintes peut varier en fonction du modèle donné. Nous pouvons prendre l'exemple de l'approche proposée par [VSL16] dont le but est d'identifier la courbe géométrique qui engendre l'apparition de la singularité 3-5 pour ensuite l'extruder à l'aide de *streamlines* dans le volume. Autrement dit, de nouvelles surfaces sont virtuellement ajoutées dans le volume et un nouveau champ d'orientations est calculé. Notons que, comme indiqué dans [MCR19], l'extrusion de la courbe dans le volume peut être perturbée par les

singularités du champ d'orientation.

Afin d'expliquer nos choix, rappelons que notre travail se place dans un cadre utilisateur. Notre but est d'ajouter des éléments interactifs à des méthodes de génération par champ d'orientation afin d'augmenter leur robustesse tout en obtenant des résultats sur des cas limitant de ces approches. Nous voulons fournir un outil permettant à un utilisateur de générer un maillage structuré par blocs qui respecte au mieux les critères de qualités énoncés à la Section 2.1.2 tout en réduisant le temps ingénieur consacré à la création du maillage. En partant du postulat que rajouter des contraintes supplémentaires lors de la génération du champ d'orientation ou la modification de la géométrie sont des problèmes complexes, nous proposons une approche différente de correction de champ d'orientation. Au lieu d'ajouter des contraintes, notre approche consiste à réduire les contraintes d'alignement au bord déjà existantes qui mènent à l'apparition des lignes singulières 3-5 et des zones de tremplin. Nous avons montré à la Section 4.1 que c'est la contrainte d'alignement avec le bord de la géométrie qui provoque l'apparition de zones non valides dans le champ d'orientation, autrement dit qui ne correspondent pas à un ensemble de grilles régulières. En relaxant cette contrainte d'alignement avec le bord de la géométrie lors de la génération, le champ d'orientation va naturellement s'aligner sur une grille régulière. De cette façon les éléments non valides du champ d'orientation ne vont plus apparaître. La contre-partie de cette relaxation de contraintes au bord est que le maillage final ne sera plus aligné avec le bord du domaine géométrique. Nous ne pouvons donc pas nous permettre de relaxer les contraintes d'alignement au bord sur l'ensemble du domaine, le maillage obtenu serait alors équivalent à un résultat de méthode par *polycubes* où toutes les singularités sont projetées sur le bord du domaine. La Figure 4.23 nous montre une structure de blocs obtenue en utilisant notre méthode de génération du Chapitre 3 sur un champ d'orientation non contraint sur l'intégralité du bord du domaine géométrique. Nous pouvons voir que des blocs ont été générés dans les zones problématiques du champ d'orientation non corrigé mais que nous avons perdu les deux lignes singulières valides - celles les plus à gauche du domaine.

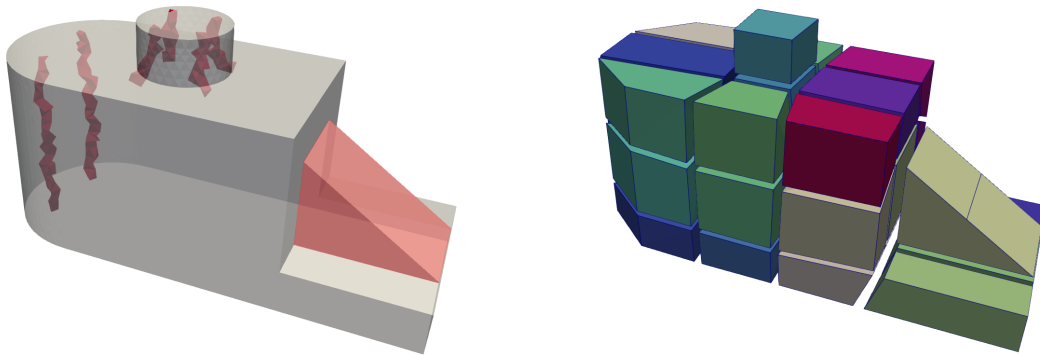


Figure 4.23 – Un exemple de structure de blocs obtenue en relaxant toutes les contraintes d'alignement au bord, ici les sommets de blocs sont projetés sur le bord du domaine géométrique.

Les travaux de [MCR19] proposent une méthode automatique de correction de lignes singulières 3-5 se rapprochant de notre idée de relaxation de contraintes. Leur méthode, illustrée à la Figure 4.24, peut se résumer comme suit :

1. Les **courbes caractéristiques** d'une ligne singulière 3-5 sont identifiées. Ces courbes sont les courbes géométriques qui provoquent l'apparition de la singularité dans le champ d'orientation ;
2. Chaque point singulier de la ligne singulière est projeté sur la courbe caractéristique la plus proche puis une nouvelle ligne est calculée entre chaque extrémités projetée (les lignes vertes sur la figure 4.24-c) ;
3. Un champ d'orientation va être de nouveau généré en adaptant les conditions au bord du domaine pour respecter les nouvelles lignes singulières projetées au bord sur la figure 4.24-d.

Cette méthode permet de corriger les lignes singulières 3-5 tout en conservant les lignes singulières valides. Par contre, elle ne permet toujours pas de prendre en compte les configurations de type T (zones de tremplin) fréquemment rencontrées dans les modèles de CAO.

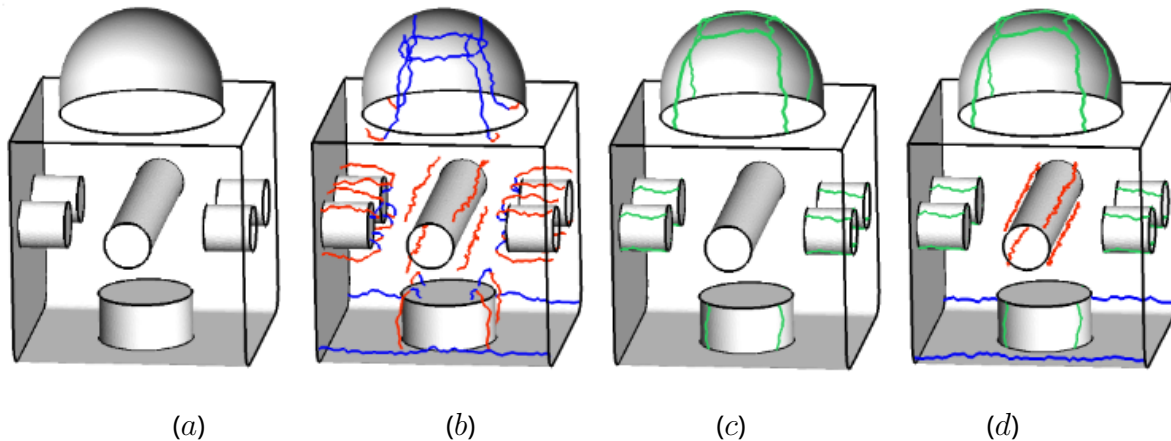


Figure 4.24 – Les étapes de la méthode de correction de lignes 3-5 de [MCR19]. Un champ d'orientation est généré sur le domaine géométrique (a). Le graphe de singularité obtenu est montré en (b), les lignes singulières 3-3 sont coloriées en bleu, les lignes 5-5 sont en rouge et les lignes 3-5 partagent les deux couleurs. Les lignes singulières 3-5 sont plaquées sur le bord du domaine pour obtenir les lignes vertes en (c). Finalement un champ d'orientation est de nouveau généré en (d) (images issue de [MCR19]).

Notre idée est alors d'utiliser des opérations interactives pour sélectionner des zones au bord proches des éléments problématiques et de relaxer les contraintes seulement dans ces zones. De cette manière, d'un point de vue utilisateur, nous rendons consistant les méthodes de correction pour les lignes singulières 3-5 (type S) ainsi que pour les zones de tremplin (type T) en plus de permettre de maximiser l'alignement du champ d'orientation avec le bord de la géométrie. Après traitement, nous obtenons des modèles de type V sur lesquels s'applique la méthode de génération de structure de blocs décrite au chapitre 3.

4.4.2 . Méthode proposée

Pour rappel dans nos travaux le champ d'orientation associé à un domaine Ω est discrétisé et assigné aux nœuds du maillage tétraédrique T_Ω qui discrétise Ω . Notre but est de relâcher certaines contraintes d'alignement des orientations portées par les nœuds de T_Ω au bord de la géométrie lors de la résolution du problème de minimisation d'énergie pour la génération du champ d'orientation. Pour cela, nous procédons en deux temps :

1. Tout d'abord, nous sélectionnons les nœuds au bord dont nous voulons relâcher la contrainte ;
2. Ensuite, nous générons à nouveau le champ globalement au domaine Ω .

Ce processus est répété interactivement jusqu'à obtenir un modèle de type V. Afin d'illustrer notre propos, considérons l'exemple de la figure 4.25. Dans la colonne (a), sont illustrées les représentations visuelles que nous exposons aux utilisateurs pour repérer les configurations de type S et T : Sur la ligne du haut, une ligne singulière 3-5 est exhibée à l'utilisateur ; sur celle du bas des points géométriques *non valides* d'une zone de tremplin sont repérés par des sphères rouges . Sur la colonne (b), l'utilisateur sélectionne des surfaces géométrique proches des zones problématique que nous considérons comme source des problèmes du champ lors de la génération. La génération d'un nouveau champ avec une relaxation des contraintes sur les surfaces sélectionnées en (b) amènera dans les situations présentées sur la colonne (c) où aucune configuration problématique n'est alors détectée.

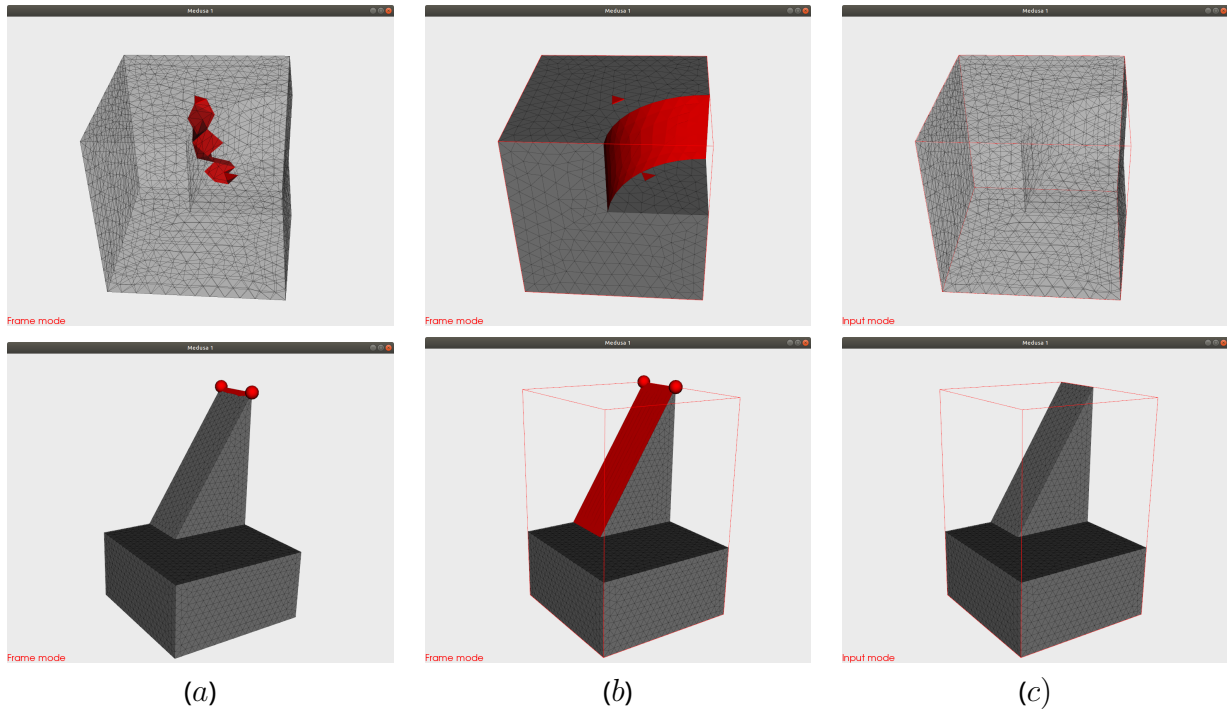


Figure 4.25 – Mise en image de notre méthode de relaxation de contraintes d’alignement avec sur la ligne du haut le traitement du modèle notch et sur la ligne du bas celui d’un modèle tremplin. En (a) le modèle ainsi que les problèmes de champ d’orientation avant notre méthode de correction, en (b) la phase de sélection des surfaces où la contrainte d’alignement sera relâchée et en (c) le modèle ainsi que les problèmes de champ d’orientation après notre méthode de correction.

Sélection de la zone au bord

Durant cette étape, l’utilisateur va sélectionner de façon interactive des nœuds du bord du domaine. L’objectif est de sélectionner une zone au bord proche de l’élément problématique que nous voulons supprimer et suffisamment grande pour que la relaxation de contraintes ait un impact concret lors de la re-génération du champ d’orientation. Même si nous utilisons une méthode interactive pour la sélection nous voulons réduire au maximum le nombre d’actions à effectuer par l’utilisateur, le but étant de pouvoir sélectionner rapidement et efficacement un grand nombre de nœuds de T_Ω .

Sélection de surfaces au bord. Pour sélectionner rapidement des zones du bord, nous nous reposons sur la classification géométrique du maillage T_Ω sur le domaine géométrique Ω . Plus précisément, nous avons décidé de sélectionner des surfaces géométriques complètes de Ω . Pour effectuer cette opération, l’utilisateur sélectionne interactivement un tétraèdre t ayant l’une de ses faces sur le bord du domaine. C’est la seule action interactive que doit effectuer l’utilisateur. Une fois le tétraèdre t choisi l’algorithme de sélection de la surface géométrique va se déclencher. Il procède comme suit. Tout d’abord, nous récupérons la face f de t classifiée au bord de Ω , c’est-à-dire que $\mathcal{C}(f) = (s, 2)$ où s est la surface géométrique sur laquelle f est classifiée. Dans le cas, où le tétraèdre sélectionné dispose de plusieurs faces au bord classifiées sur des surfaces différentes, nous abandonnons l’opération. Ayant la connaissance de s , nous récupérons ensuite toutes les faces f_i de T_Ω telles que $\mathcal{C}(f_i) = (s, 2)$. Soit F cet ensemble de faces, et N_F , l’ensemble des nœuds T_Ω tel que $n_i \in N_F$ si et seulement si n_i est incident à au moins une face de F . Il est à noter que N_F contient les nœuds incidents aux faces classifiées sur s ce qui inclue les nœuds classifiés sur les courbes géométriques qui forment le bord de s . Ce point est important pour obtenir les résultats souhaités avec l’algorithme de génération de champ d’orientations que nous utilisons. Pour illustrer notre propos,

considérons la Figure 4.26 qui montre les modèles *notch* (à gauche) et *tremplin* (à droite) après l'étape de relaxation de contraintes en n'ayant pas sélectionné les nœuds de T_Ω sur les courbes géométriques. Nous pouvons très clairement voir en (a) que la ligne singulière 3-5 montrée à la Figure 4.25 (a) a disparu. Cependant de nouvelles lignes singulières 3-5 se sont formées proche des courbes géométriques. De façon similaire en (b), on voit qu'après l'étape de relaxation notre méthode d'analyse du champ détecte toujours des sommets géométriques non valides, c'est-à-dire que le problème de la zone de tremplin n'est pas corrigé si l'alignement du champ avec les courbes géométrique n'est pas relâché.

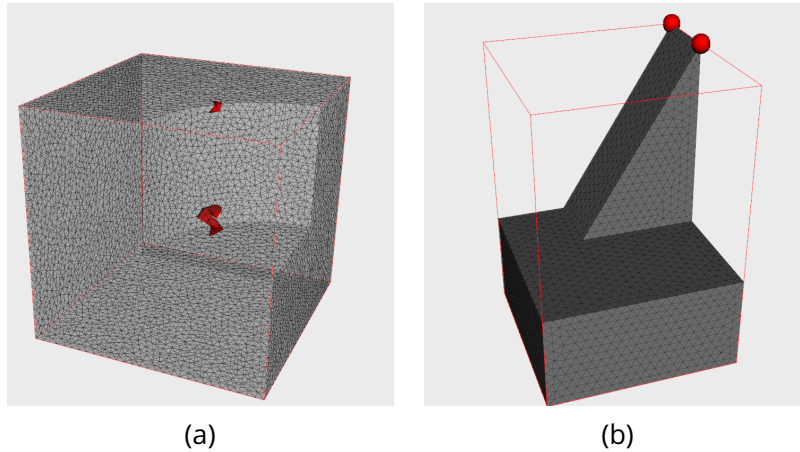


Figure 4.26 – Les modèles notch et tremplin analysés par notre outil après avoir effectué la même opération qu'à la Figure 4.25 mais cette fois-ci en ne relâchant pas les contraintes des nœuds sur les courbes géométriques du domaine Ω .

Sélection volumique du voisinage d'une ligne singulière 3-5. La sélection de surfaces géométriques est une méthode robuste et simple pour obtenir un ensemble de nœuds de T_Ω pour l'étape de relaxation de contraintes. Cependant dans certains cas, sélectionner une surface complète crée une zone au bord trop grande. Cette situation se produit notamment avec les lignes singulières 3-5. A titre d'exemple, considérons le modèle présenté sur la Figure 4.27. En (a), le champ d'orientation du domaine possède deux lignes singulières dont une 3-5. En (b) la surface proche de cette ligne singulière 3-5 est sélectionnée pour correction du champ d'orientation. Enfin, la figure (c) montre que le champ généré après relaxation des contraintes de la surface sélectionnée en rouge ne possède plus la ligne singulière 3-5 mais il ne possède plus non plus la deuxième ligne du singularité du champ qui elle était valide.

Nous proposons alors une deuxième méthode de sélection de zone au bord. Cette méthode a pour but d'obtenir un ensemble T_n de tétraèdres autour d'une ligne singulière sélectionnée par l'utilisateur. Notons s_0 , l'ensemble des tétraèdres de T_Ω contenant une telle ligne. L'ensemble T_n est construit itérativement et interactivement par l'utilisateur en débutant de l'ensemble $T_0 = s_0$ à l'aide de la relation

$$T_{i+1} = T_i + \{t \in T_\Omega - T_i / \exists t' \in T_i, t \cap t' \in T_\Omega^0\}.$$

En d'autres termes, l'ensemble T_{i+1} est égal à l'ensemble T_i augmenté de l'ensemble des tétraèdres qui partagent un nœud avec un tétraèdre de T_i . La Figure 4.28 (b) montre l'ensemble T_1 autour de la ligne singulière 3-5 sélectionnée. L'utilisateur va ensuite augmenter, de façon itérative et interactive, créer les ensembles T_2, T_3, \dots jusqu'à obtenir une zone au bord suffisamment large comme montré en (c). Cette méthode permet de relâcher les contraintes au bord dans une zone plus locale à la ligne singulière comme illustré en (d) où la ligne 3-5 est supprimée tout en conservant l'alignement au bord du champ d'orientation proche de la ligne singulière 3-3 qui elle est valide.

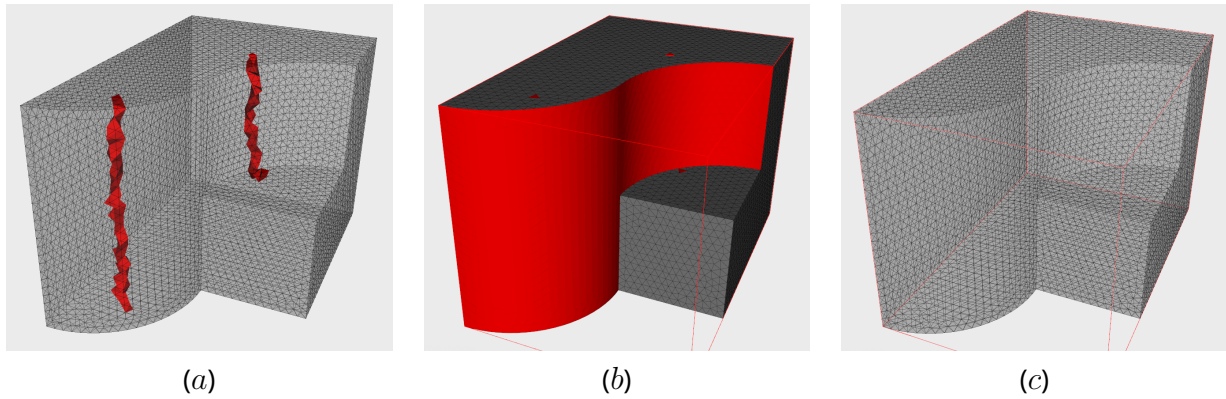


Figure 4.27 – Un exemple de domaine géométrique pour lequel le champ d'orientation possède une ligne singulière 3-3 et une ligne singulière 3-5 (a). Nous sélectionnons la surface géométrique pour la relaxation de contraintes en (b) et en (c), après re-génération du champ d'orientation, les deux lignes singulières ont disparu.

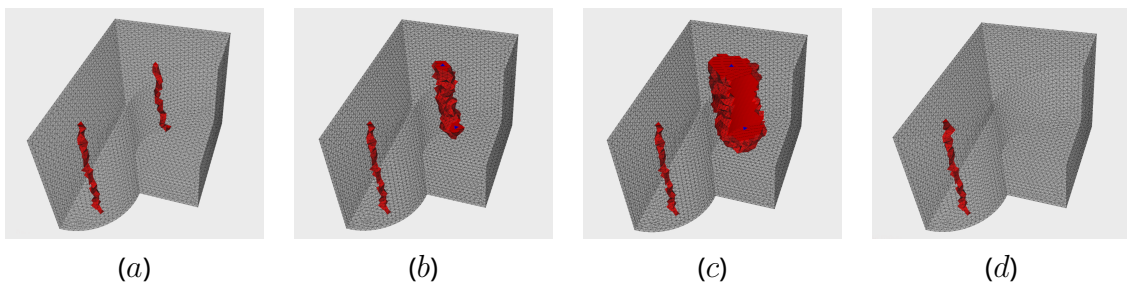


Figure 4.28 – Un exemple d'utilisation de la méthode de correction de champ d'orientation en sélectionnant une ligne singulière. En (a), le modèle avec les lignes singulières du champ. En (b), la ligne singulière 3-5 est sélectionnée. En (c), l'ensemble des tétraèdres sélectionnés par l'utilisateur. Enfin en (d), le champ obtenu après l'étape de re-génération du champ d'orientation avec relâchement de contraintes.

Relaxation du champ d'orientation

Lors de la première génération du champ d'orientation les contraintes d'alignement ont été initialisées sur les nœuds de T_Ω en fonction de leur classification géométrique. Pour rappel, nous résolvons le problème 3.12, présenté page 49 et qui a pour objectif de minimiser la fonction suivante

$$\sum_{ij \in T_\Omega^1} \mathbf{L}_{ij} \|\mathbf{a}_j - \mathbf{a}_i\|^2 + \lambda \sum_{j \in \mathbf{J}} \|\mathbf{a}_j - \sqrt{7/12} \mathbf{R}_v^B \delta_9^5 - \mathbf{c}_j[0] \mathbf{R}_v^B \delta_9^9 - \mathbf{c}_j[1] \mathbf{R}_v^B \delta_9^1\|^2 \quad (4.2)$$

où l'ensemble \mathbf{J} contient les nœuds du bord du domaine positionnés sur des surfaces de Ω . Le premier terme contrôle le caractère régulier du champ d'orientation généré tandis que le second terme qui implique les nœuds de \mathbf{J} contrôle l'alignement au bord. Après sélection de sommets au bord à ne plus considérer comme contraint, nous les supprimons simplement de l'ensemble \mathbf{J} et nous générons à nouveau un champ d'orientations. Ces nœuds, devenus *libres*, ne sont donc plus pris en compte dans le système de contraintes et auront le même comportement que les nœuds internes. La Figure 4.29 nous montre l'impact de cette opération sur le champ d'orientation des modèles *notch* en haut et *tremplin* en bas. En (a) et en (c) les champs d'orientations sont générés avec les contraintes d'alignement au bord, nous pouvons notamment voir en (a) la singularité du champ en rouge sur la surface du domaine formée par l'alignement du champ au bord. Nous pouvons également voir en (c) le champ d'orientation s'aligne en direction du sommet géométrique tout en restant régulier. Nous appliquons notre méthode de relaxation de contraintes sur ces modèles ce qui nous permet d'obtenir les champs d'orientation montrés en (b) et (d). Dans les deux cas nous pouvons voir que les orientations, en rouge, des surfaces sélectionnées pour la relaxation de contraintes ne sont plus alignées avec le bord du domaine, ce qui a pour effet en (b) de faire disparaître la singularité de valence 3 ainsi que la singularité de valence 5 et en (d) de ne plus aligner le champ en direction du sommet géométrique.

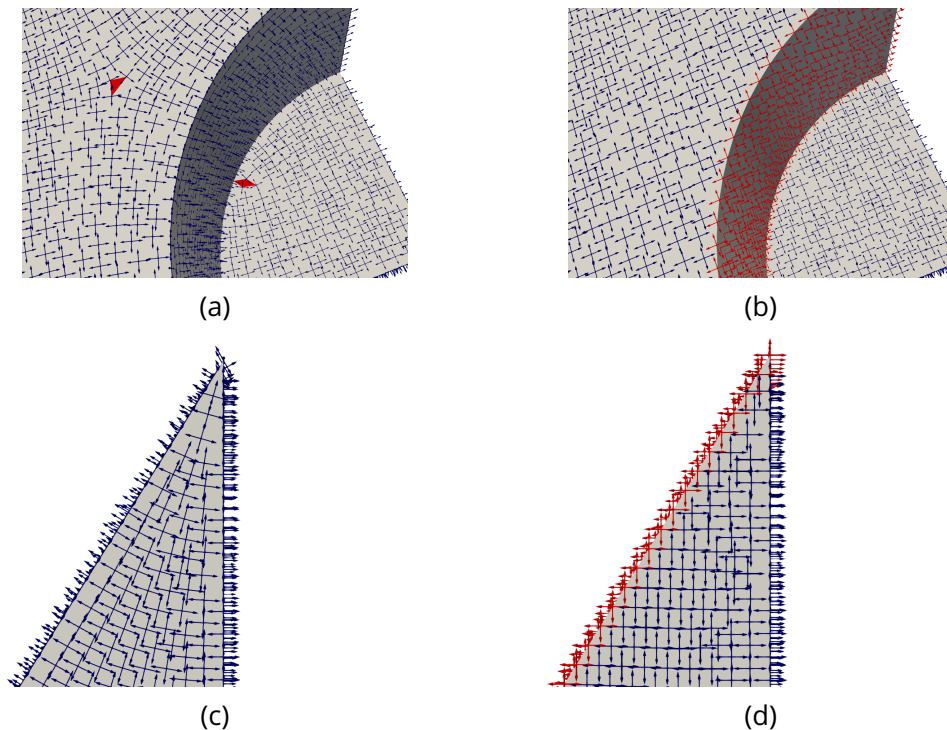


Figure 4.29 – Le champ d'orientation avant et après notre correction, respectivement sur le modèle notch en (a) et (b) et sur le modèles tremplin en (c) et (d). Dans les deux cas, les orientations en rouge sont celles qui ont été sélectionnées pour la relaxation de contraintes.

Exemples de relaxation de contraintes sur des configurations de type S et T

L'approche proposée dans cette section nous permet de traiter les configurations de type S et T en corrigeant les champs d'orientation en quelques interactions pour obtenir un champ valide pour notre approche de tracé interactif de surfaces duales. Des résultats obtenus avec notre approches sont montrés à la Figure 4.30 où nous avons pu générer un maillage hexaédrique sur les modèles *notch* et *tremplin*.

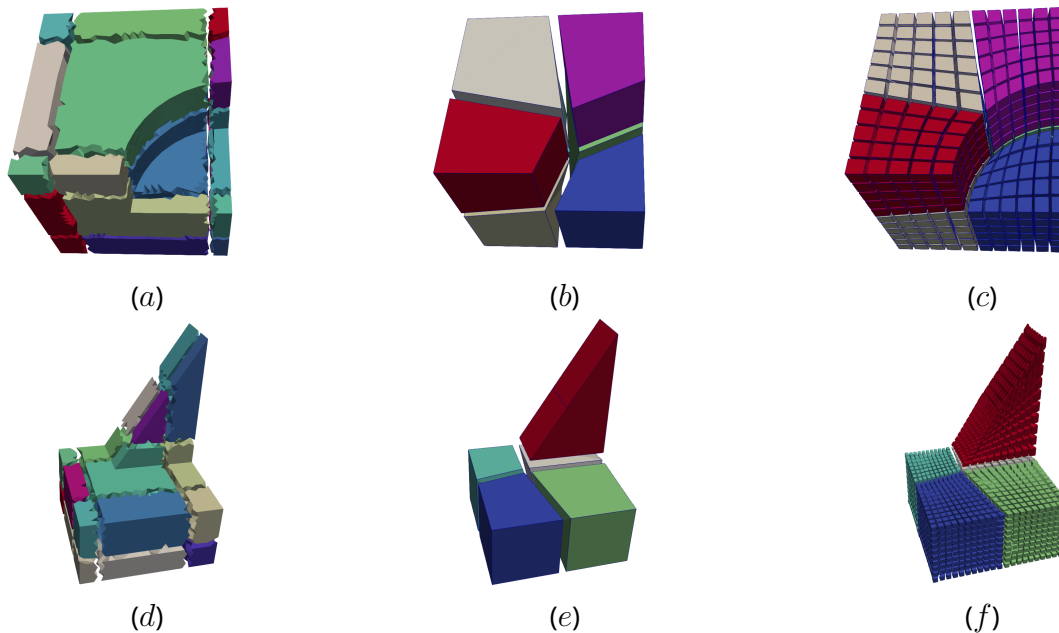


Figure 4.30 – Les régions duales, structures de blocs et maillage hexaédriques obtenus sur les modèles du notch en haut, et du tremplin en bas, dont les champs d'orientations ont été corrigés.

La Figure 4.31 illustre que la sélection de deux surfaces nous permet de rendre possible le découpage en blocs d'un modèle disposant de lignes singulières 3-5 et d'une zone de tremplin. La structure de blocs finale est montrée à la figure 4.32. Une liste plus importante de résultats obtenus en utilisant notre méthode de correction de champ d'orientation sera proposée au chapitre 5.

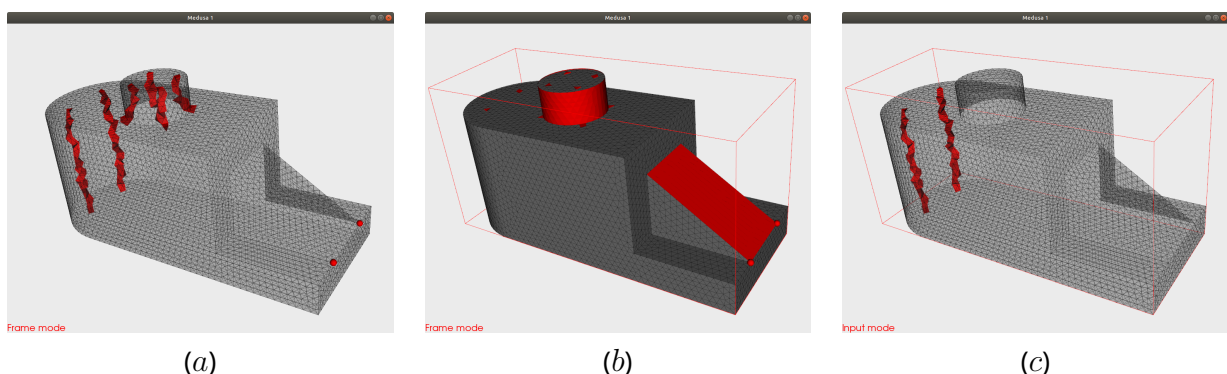


Figure 4.31 – Mise en image de notre méthode de correction de champ d'orientation sur notre modèle d'exemple. En (a) le champ d'orientation analysé où sont affichés les lignes singulières ainsi que les points géométriques non valides. En (b) deux surfaces sont sélectionnées interactivement avec notre outil. Enfin en (c) le champ d'orientation après notre méthode de correction.

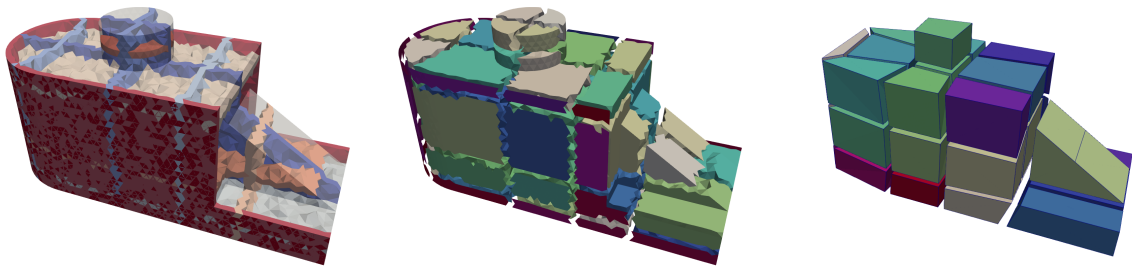


Figure 4.32 – Structure de blocs finalement obtenue après l'étape de correction illustrée à la figure 4.31. On peut noter la présence d'un bloc ressemblant à un prisme à base 3 caractéristique de la relaxation de contraintes dans des zones de tremplin.

5 - Post-processing et résultats de notre approche

Nous avons proposé dans le Chapitre 3 une méthode de génération de structure de blocs en traçant des feuillettes duaux qui suivent le champ d'orientation dans un domaine Ω . Cette méthode nous a permis de traiter les configurations de type V. Cependant des limitations à cette approche sont apparues sur les modèles de type S et T où les éléments problématiques des champs d'orientations nous empêchaient de tracer des feuillettes dans certaines zones du domaine. Pour remédier à cela, nous avons développé une méthode de correction de champ d'orientation proposée au Chapitre 4 afin de pouvoir gérer ces cas limitant. Dans ce chapitre, nous complétons la présentation de notre travail en présentant différents exemples d'application de notre méthode. Celle-ci générant une structure de blocs classifiés sur la géométrie, nous commençons ce chapitre en indiquant comment le maillage final est en pratique généré (voir section 5.1). Par la suite, nous présentons l'utilisation de l'outil développé durant cette thèse en considérant les trois types de configurations rencontrées.

L'exemple de la figure 5.1 montre notre méthode finale de génération de structure de blocs. Pour rappel nous partons d'un domaine géométrique Ω discrétisé par un maillage tétraédrique T_Ω sur lequel nous avons calculé un champ d'orientation \mathcal{F}_Ω montré à la figure 5.1-a. Nous commençons par analyser le champ d'orientation afin de détecter les configurations de type S et T, ce qui correspond à la figure 5.1-b. Si de telles configurations existent dans le champ d'orientation nous pouvons corriger le champ, d'abord en sélectionnant de manière interactive un ensemble de zones au bord du domaine à la figure 5.1-c. Puis nous générons le champ d'orientation une nouvelle fois en relaxant la contrainte d'alignement au bord sur les zones sélectionnées, ce qui nous permet d'obtenir un champ corrigé comme montré à la figure 5.1-d. Enfin nous pouvons reprendre notre méthode de génération de structure de blocs présentée au chapitre 3, résumé à la figure 5.1-e.

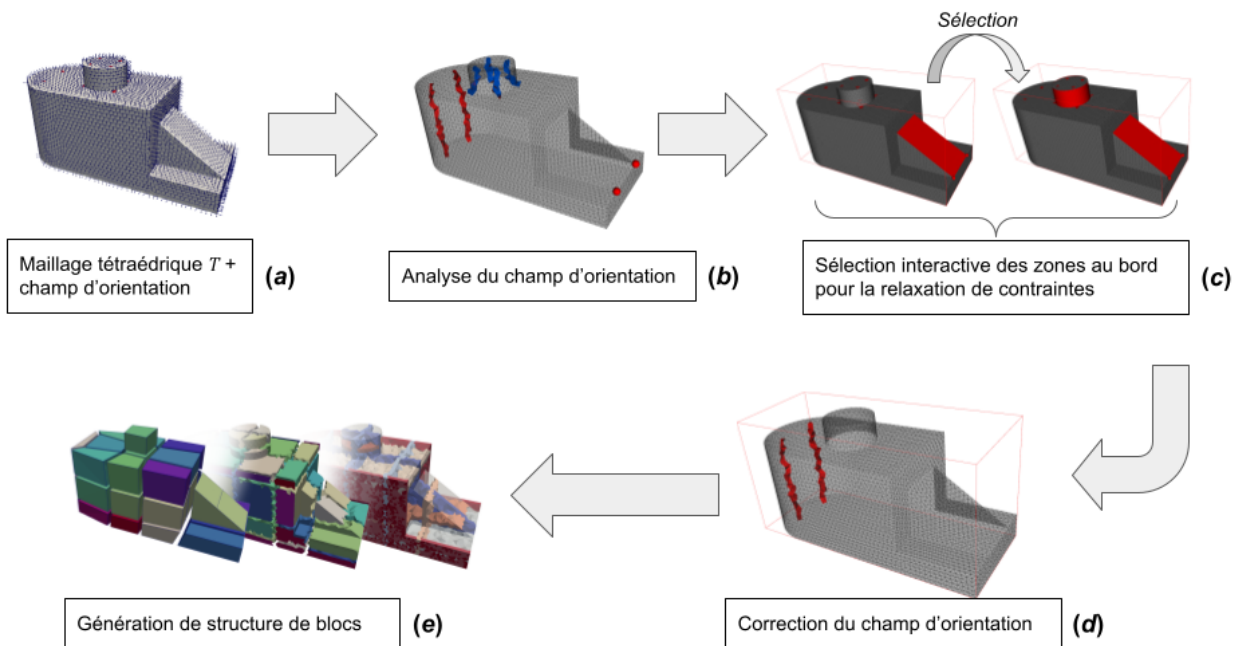


Figure 5.1 – Pipeline final de notre approche de génération de structure de blocs. Nous avons ajouté les étapes de correction de champs d'orientation pour gérer les configurations de type S et T.

En résumé

Ce chapitre présente dans un premier temps l'étape finale de notre pipeline de génération de maillage. Étant donné une structure de blocs obtenue à l'aide des algorithmes proposés au chapitres 3 et 4, comment générons-nous le maillage hexaédrique structuré par blocs. La seconde partie du chapitre est dédiée à la présentation de l'utilisation de l'outil développé durant cette thèse. Différents cas de modèles géométriques, caractéristiques des configurations de type V, S et T sont discrétisés par une structure de blocs hexaédriques à l'aide de notre approche.

5.1 . Génération du maillage hexaédrique final

Notre objectif initial était de générer un maillage hexaédrique qui discrétise le domaine géométrique Ω tout en étant conforme et structuré par blocs.

Dans les chapitres précédents, nous avons montré comment nous pouvions interactivement générer des structures de blocs. Il reste désormais à discrétiser chaque bloc de manière conforme entre eux et en respectant la géométrie du bord du domaine Ω . En effet, la structure de blocs est un maillage hexaédrique grossier qui ne discrétise pas précisément le domaine Ω . De plus nous devons permettre à l'utilisateur de contrôler la taille des mailles de façon locale en conservant la conformité du maillage final. Pour ce faire, nous proposons l'approche suivante :

1. Tout d'abord, nous allons classifier les arêtes et les faces de la structure de blocs. Cette classification va nous permettre ensuite de faciliter le positionnement des nœuds du maillage hexaédrique final, dont la méthode est expliquée à la Section 5.1.1.
2. Nous allons ensuite assigner une valeur de discrétisation aux arêtes de blocs en nous basant sur la méthode de discrétisation utilisé dans l'algorithme de submapping de [RGSR09]. Nous proposons à la Section 5.1.2 une modification du programme linéaire donné par [RGSR09] pour qu'il (1) corresponde à notre structure de blocs et (2) puisse être guidé par l'utilisateur.
3. Une fois la discrétisation des arêtes de blocs calculée, nous générons le maillage hexaédrique final de façon automatique en utilisant une méthode d'interpolation transfinie.

5.1.1 . Classification géométrique

La classification géométrique représente une étape importante pour la génération de maillage lorsque l'on discrétise des modèles de type CAO où nous avons connaissance de la représentation par les bords du modèle. Savoir qu'un nœud de maillage vit sur une courbe ou une surface de CAO, nous permet potentiellement de le déplacer de manière optimale en considérant les caractéristiques géométriques intrinsèques des entités (normales, courbures). Comme présenté à la Section 2.1.2, nous voulons que le maillage final soit le mieux aligné possible le long du bord du domaine Ω . La classification géométrique nous permet d'utiliser des opérations comme par exemple la projection de points sur des entités géométriques. Cette classification géométrique du maillage peut aussi avoir une importance lors de l'utilisation de ces maillages en simulation numérique : si par exemple il est nécessaire de définir une condition aux limites ou une condition de bord sur une portion du bord, il est aisé via la classification géométrique de reporter ces considérations sur les cellules du maillage classifiées sur l'entité géométrique en question.

Dans la suite de cette section nous allons définir une méthode simple permettant de classifier les arêtes ainsi que les faces de la structure de blocs (les sommets eux ont déjà été classifiés lors de la création des blocs). Une fois les cellules de la structure de blocs classifiées nous pourrions propager cette classification aux sommets du maillage final qui discrétisent ces cellules.

Classification géométrique des arêtes de blocs

Pour classifier les arêtes ainsi que les faces des blocs nous allons utiliser les informations de classification contenues dans les régions duales correspondantes aux sommets de blocs qui définissent le bord de la cellule, définie à la Section 3.3.1. Nous cherchons toujours à classifier une cellule sur l'entité géométrique en contact avec la plus petite dimension. Prenons un exemple en 2D de classification d'arêtes de bloc illustré à la Figure 5.2. Soit e_1 l'arête de bloc connectant les nœuds n_A et n_B et \mathbf{A} et \mathbf{B} les régions duales correspondantes aux nœuds n_A et n_B . Nous allons chercher l'entité de plus petite dimension capturée à la fois par \mathbf{A} et \mathbf{B} . La classification de l'arête e_1 est alors définie par $\mathcal{C}(e_1) = \min \mathcal{C}(\mathbf{A}) \cap \mathcal{C}(\mathbf{r}_{n_B})$. Sur notre exemple en (b) nous voyons que \mathbf{A} capture les courbes c_1, c_2 et à la surface s et que \mathbf{B} capture les courbes c_1, c_6 et à la surface s également, ici l'entité géométrique commune de plus petite dimension est la courbe c_1 donc nous avons la classification $\mathcal{C}(e_1) = c_1$. Prenons maintenant le cas de l'arête e_2 dont les sommets sont n_C et n_D , de manière analogue la région duale \mathbf{D} capture c_2 et c_3 ainsi que la surface s et \mathbf{C} capture la courbe c_6 et la surface s , ici l'entité géométrique commune de plus petite dimension est la surface s donc $\mathcal{C}(e_2) = s$.

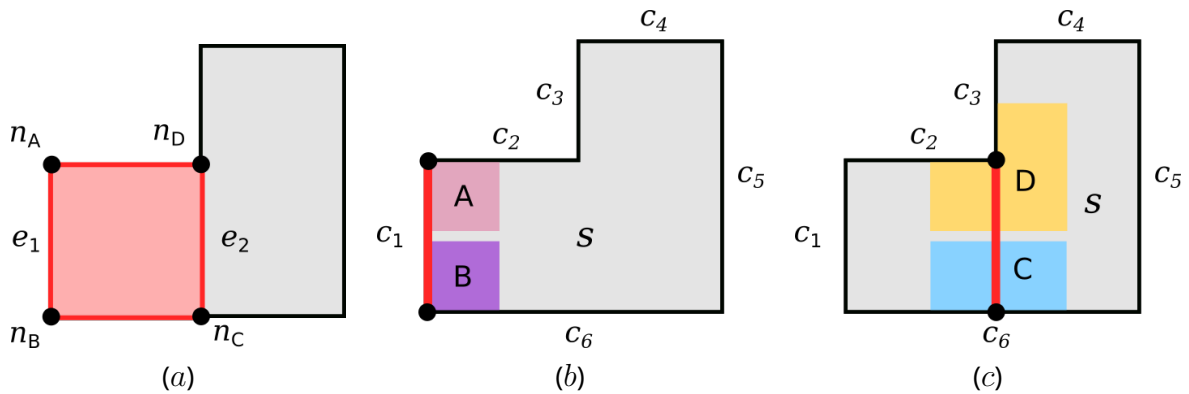


Figure 5.2 – Illustration 2D de la méthode de classification des arêtes des blocs. Les arêtes e_1 et e_2 du bloc rouge en (a) vont être classifiées. En (b) l'arête e_1 est traitée, les informations de classification géométrique des régions duales \mathbf{A} et \mathbf{B} des sommets de e_1 nous permettent de déterminer que l'arête est classifiée sur la courbe géométrique c_1 . De façon similaire en (c), en utilisant les informations de classification des régions duales \mathbf{C} et \mathbf{D} , nous classifions l'arête e_2 sur la surface géométrique s .

Classification géométrique des faces de blocs

Nous allons ensuite appliquer une méthode similaire pour déterminer la classification des faces des blocs cette fois-ci en prenant en compte les quatre sommets de la face. Pour une face f nous avons alors l'équation suivante : $\mathcal{C}(f) = \min \mathcal{C}(\mathbf{r}_{n_0}) \cap \mathcal{C}(\mathbf{r}_{n_1}) \cap \mathcal{C}(\mathbf{r}_{n_2}) \cap \mathcal{C}(\mathbf{r}_{n_3})$. Avec \mathbf{r}_{n_i} la région duale correspondante au nœud n_i . Cependant cette méthode ne permet pas dans tous les cas d'obtenir une classification correcte pour une face de bloc. Deux exemples de configuration particulières sont montrées à la Figure 5.3. Sur cette figure, dans les deux cas, la face de bloc en rouge qui se situe à l'intérieur du volume va être classifiée sur la surface géométrique bleue, cette surface est l'entité géométrique de plus petite dimension commune aux quatre sommets de la face de bloc rouge. Afin de corriger ce problème nous allons en plus utiliser des informations sur les arêtes qui forment le bord de la face, dans le cas (a) les arêtes de la face sont classifiées sur des courbes géométriques concaves et dans le cas (b) deux arêtes de la face en rouge sont classifiées à l'intérieur du volume il est donc impossible que la face soit située sur une surface géométrique.

5.1.2 . Calcul de la discrétisation des arêtes

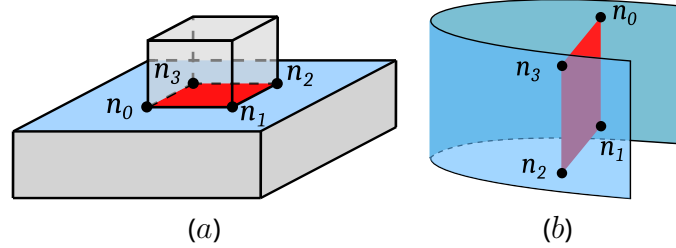


Figure 5.3 – Illustration de deux cas où nous avons besoin de raffiner les informations nécessaires pour obtenir la classification de la face de bloc rouge.

A ce stade, nous avons un ensemble de blocs hexaédriques que nous voulons raffiner pour obtenir le maillage final en respectant deux critères : (1) le maillage obtenu doit être conforme ; (2) l'utilisateur doit pouvoir contrôler la taille des éléments si besoin. Une méthode simple pour construire une discrétisation d'un bloc hexaédrique est la méthode dite de "mapping" [CO82]. L'idée du mapping 3D est d'insérer un maillage structuré $i \times j \times k$, c'est-à-dire que toutes les arêtes opposées du bloc vont avoir le même nombre de subdivision. La Figure 5.4 montre les groupes d'arêtes coloriés en bleu, rouge et vert et subdivisés respectivement avec $i = 4$, $j = 2$ et $k = 5$.

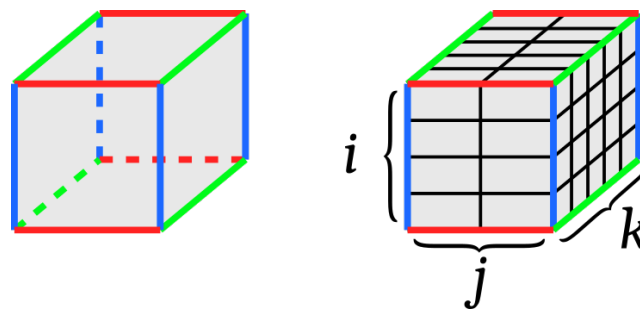


Figure 5.4 – Une illustration de maillage par approche de "mapping". A gauche, chaque arête du bloc est coloriée en fonction de groupe d'arêtes opposées (en bleu, rouge et vert). A droite, un nombre de subdivision est associé à chaque groupe d'arêtes opposées, nous obtenons donc une grille régulière dans le bloc.

Appliquer l'algorithme de mapping sur les blocs indépendamment peut mener à la non conformité du maillage final si une arête de bloc est découpée différemment pour des blocs qui lui sont incidents. Nous rappelons que pour conserver la conformité dans un maillage hexaédrique lors de la subdivision d'un hexaèdre h , nous devons propager cette opération dans les trois couches¹ de maillage dont h fait partie. Nous avons défini à la Section 2.1.1 que pour chaque arête e d'un maillage hexaédrique, nous pouvons déterminer un feuillet primal qui traverse toutes les arêtes "opposées" à e dans le maillage. Nous notons désormais \mathbf{E}_f l'ensemble des arêtes opposées qui forment le feuillet primal f . Pour tous feuillets primaux f_i et f_j , nous avons $\mathbf{E}_{f_i} \cap \mathbf{E}_{f_j} = \emptyset$, c'est-à-dire qu'une arête du maillage n'appartient qu'à un seul ensemble d'arêtes opposées. En regardant la Figure 5.5, nous voyons en (a) deux feuillets primaux représentés par leur groupe d'arêtes E_V en vert et E_R en rouge ainsi que le bloc b traversé par les deux feuillets primaux. Nous allons "mapper" le bloc b avec différentes valeurs. Tout d'abord en (b), nous subdivisons les arêtes vertes en 3 et les arêtes rouges en 2 ; puis en (c), nous inversons les valeurs de subdivision des deux groupes d'arêtes de b . Ces valeurs de subdivision sont propagées dans chaque groupe d'arêtes respectif E_V et E_R . Dans les deux cas nous pouvons voir que le bloc b est correctement discrétisé : nous obtenons bien une grille régulière conforme avec les grilles créées dans les blocs voisins.

1. Une couche par direction de l'hexaèdre, c'est-à-dire par quadruplet d'arêtes opposées dans l'hexaèdre.

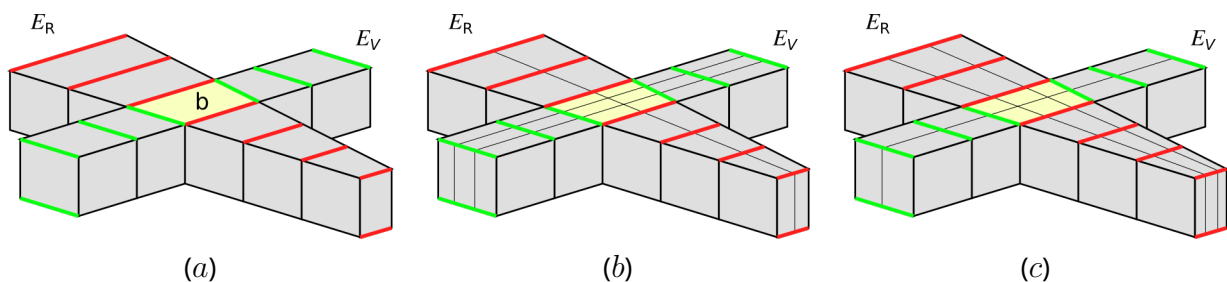


Figure 5.5 – Une illustration de la relation entre le "mapping" d'un bloc b et de deux ensembles d'arêtes opposées E_V en vert et E_R en rouge dont b fait parti (a). Nous avons en (b) un exemple de discrétisation avec une valeur de 3 pour les arêtes de E_V et de 2 E_R . En (c) E_V est discrétisée avec une valeur de 2 et E_R une valeur de 3.

Autrement dit, pour conserver la conformité du maillage lors de l'étape de subdivision, nous devons nous assurer que toutes les arêtes d'un feuillet primal sont subdivisées avec le même nombre d'arêtes.

Pour s'en assurer, une solution simple serait d'assigner le même nombre de subdivisions aux arêtes de tous les blocs de notre structure. Pour chaque bloc nous aurons alors une grille régulière $i \times i \times i$. Cette méthode ne permet pas d'avoir un contrôle local sur la taille des mailles et engendre généralement la création de mailles soit trop petites, soit trop grandes dans certaines zones du domaine. Afin de pouvoir contrôler la taille des mailles, nous avons repris la méthode d'affectation d'intervalles proposée par [RGS09] et l'avons adaptée à notre contexte. Les auteurs de [RGS09] proposent de calculer à l'aide d'un programme linéaire (que nous rappelons ci-après) le nombre de subdivisions à effectuer sur chaque arête d'un polygone en 2D et d'un polyèdre en 3D en fonction d'une taille de maille cible. Leur approche est définie pour une méthode de submapping ([WWB05]) qui permet de discrétiser ces formes géométriques. Étant donné un domaine Ω représenté par son bord, l'approche considérée dans [RGS09] consiste à parcourir de façon "ordonnée" le bord de Ω pour associer à chacune des surfaces le composant une direction principale permettant de "transformer" Ω en un polycube² (voir figure 5.6). Chaque face du polycube généré est alors colorée avec les directions $\pm I$, $\pm J$ ou $\pm K$, ce qui leur permet de définir des ensembles de surfaces et donc de courbes opposées. Ils calculent le nombre d'intervalles de subdivision des courbes afin que chaque ensemble de courbes opposées (par exemple étiquetés $+I$ et $-I$) dans le modèle possède un nombre égal d'arête et permette ainsi l'application de l'algorithme de submapping. Ce qui nous intéresse dans cette approche est la capacité de pouvoir assurer l'obtention d'un maillage conforme global tout en considérant une taille de maille cible.

Pour contrôler la taille des éléments du maillage final, nous simplifions actuellement le problème en considérant que nous désirons obtenir des mailles dont les arêtes sont toutes les plus proches d'une taille $elem_size$ définie par l'utilisateur. À partir de cette taille cible, nous dérivons une valeur cible N_e pour chaque arête e de la structure de bloc. Nous posons $N_e = length(e)/elem_size$, où $length(e)$ est la longueur de l'arête e . Dans notre implémentation, nous définissons comme valeur par défaut pour $elem_size$ la valeur $\frac{1}{10}d(BB_0, BB_1)$ où BB_0 et BB_1 sont les deux points d'extrémité de la boîte englobante de notre domaine Ω . Il est à noter que notre approche des feuilletts primaux nous permettrait de définir une taille d'éléments différente pour un ensemble d'arêtes.

Notre objectif est de discrétiser avec le même nombre de subdivision les arêtes "opposées" dans tous les feuilletts primaux. En considérant n_e la discrétisation que l'on souhaite avoir d'une arête e , nous avons à

2. Les auteurs de [RGS09] ne se placent pas du tout dans le cadre des approches de type polycube. Ils considèrent uniquement un parcours du bord sans se soucier de la déformation de l'intérieur du domaine.

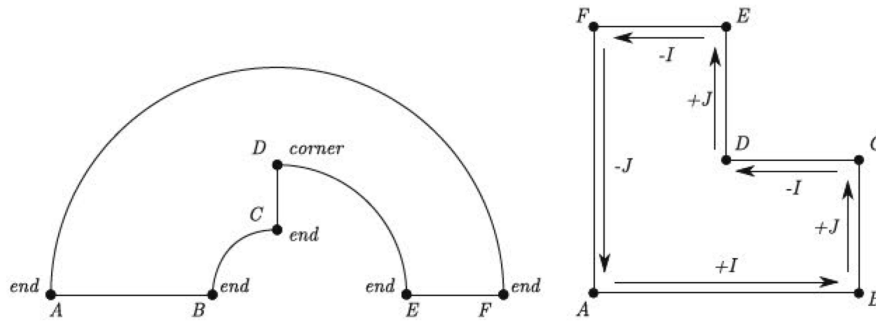


Figure 5.6 – Assignment $\{\pm I, \pm J\}$ des courbes formant le bord d'un domaine 2D en (a) pour obtenir la représentation en (b) qui servira de support au calcul des contraintes de discrétisation d'arêtes afin d'obtenir le maillage final dans [RGS09].

résoudre le problème suivant :

$$\begin{cases} \min \sum_e \omega_e |n_e - N_e|, \\ \text{contraint par :} \\ n_i = n_j, \text{ pour tout couple d'arêtes } (e_i, e_j) \in E_f, \\ n_e \in \mathbb{N}, \text{ et } n_e > 0 \text{ pour toute arête } e. \end{cases} \quad (5.1)$$

L'objectif de ce programme linéaire est de minimiser une somme pondérée des différences entre n_e la discrétisation calculée et N_e la discrétisation cible pour chaque arête e . Le poids $\omega_e = 1/\text{length}(e)$ permet de pénaliser un grand nombre de subdivisions d'arête de petite taille. Les contraintes du problèmes sont liées à la nature entière de la subdivision ($n_e \in \mathbb{N}^+$) et au fait que les arêtes opposées dans un feuillet primal doivent toutes être discrétisées de façon identique ($n_e = n_i$, avec les arêtes e et i incluses dans l'ensemble d'arêtes E_f du feuillet primal f).

D'un point de vue implémentation, nous utilisons le solveur GLPK [And18] pour résoudre notre système. Cela induit de linéariser le problème en ne considérant pas la minimisation de termes impliquant des valeurs absolues. Pour cela, nous utilisons une approche classique s'appuyant sur l'introduction de variables intermédiaire, notées ici D_e et d_e . Ces variables représentent respectivement la différence positive et négative de $|n_e - N_e|$. Dès lors, le problème 5.1 peut être réécrit :

$$\begin{cases} \min \sum_e \omega_e (D_e + d_e), \\ \text{contraint par :} \\ n_i = n_j, \text{ pour tout couple d'arêtes } (e_i, e_j) \in E_f, \\ n_e \in \mathbb{N}, \text{ et } n_e > 0 \text{ pour toute arête } e, \\ D_e \in \mathbb{N}, \text{ et } D_e \geq 0 \text{ pour toute arête } e, \\ d_e \in \mathbb{N}, \text{ et } d_e \geq 0 \text{ pour toute arête } e, \\ D_e \geq n_e - N_e, \text{ pour toute arête } e, \\ d_e \leq N_e - n_e, \text{ pour toute arête } e. \end{cases} \quad (5.2)$$

5.1.3 . Créations des points du maillage

Maintenant que nous avons calculé le nombre de subdivisions pour chaque arête de la structure de blocs nous allons pouvoir créer le maillage final. Pour ce faire nous utilisons une méthode par interpolation transfinie, aussi appelé **TFI** de l'anglais *TransFinite Interpolation* ([GT82]). Cette opération nous permet de créer l'ensemble des nœuds du maillage hexaédrique final. Lorsque nous discrétisons un bloc avec les valeurs

i , j et k nous insérons également l'indice des nœuds du maillages créés dans une matrice (i, j, k) , cette matrice nous permet par la suite de trouver les combinaisons de 8 nœuds qui forment des hexaèdres.

La particularité de notre approche est que nous partons de blocs hexaédriques linéaires, la discrétisation de ces blocs va créer un maillage très rectiligne, de plus il est possible que certains nœuds créés se situent hors du domaine à mailler Ω . Un tel exemple est montré à la figure 5.7, nous discrétisons le bloc (a) créé à partir de la géométrie (d), lorsque nous discrétisons les arêtes du blocs certains nœuds se situent à l'extérieur de la géométrie comme nous pouvons le voir en (b). Continuer l'opération de maillage par TFI de cette façon nous permet d'obtenir un maillage hexaédrique mais celui-ci ne correspond pas à la géométrie (c). Pour éviter ces erreurs, lors de la discrétisation d'une entité de la structure de blocs, nous allons propager la classification géométrique de l'entité aux nœuds du maillages qui sont créés. Suite à cela nous allons projeter les nœuds créés sur la géométrie comme montré en (e), ici nous montrons cette opération seulement sur les nœuds des arêtes de blocs pour un gain de clarté mais elle est effectuée également sur les nœuds classifiés sur des surfaces du domaine. De cette manière nous obtenons un maillage qui correspond à la géométrie (f).

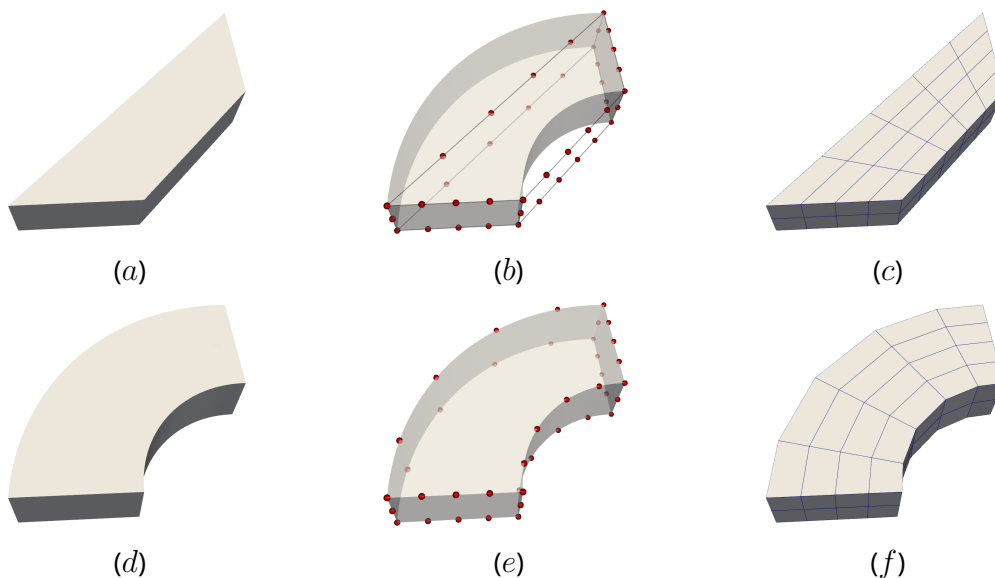


Figure 5.7 – Un exemple de discrétisation de bloc non projeté en haut et projeté sur la géométrie en bas.

5.2 . Résultats

Dans cette section nous allons présenter les résultats de notre méthode de génération de structures de blocs. L'ensemble des travaux présentés dans le début de ce chapitre et dans les chapitres précédents a été implémenté en C++ au sein de la bibliothèque de génération de maillage open-source GMDS du CEA.

Nous avons également conçu et développé un outil qui nous permet de visualiser et interagir avec les maillages que nous souhaitons traiter. Cet outil à été implémenté en C++ en utilisant la bibliothèque VTK [SML06]. Nous allons présenter un exemple d'exécution de notre méthode de création de génération de maillage structuré par blocs à l'aide de notre outil à la Section 5.2.1. Nous présenterons ensuite des résultats obtenus sur les différents types de configurations en commençant par les configurations de type V à la Section 5.2.2 puis sur les configurations de type S et T à la Section 5.2.3.

5.2.1 . Utilisation de l'outil

Dans cette partie de la section nous allons dérouler une exécution de notre méthode à l'aide de notre outil, chaque étape est montrée à la figure 5.8. Pour commencer nous chargeons le fichier VTK définissant le maillage tétraédrique de notre modèle dans notre outil, cela nous permet de visualiser le maillage (*a*). Ensuite nous allons entamer la phase de correction du champ d'orientation présentée au Chapitre 4. Nous allons afficher le graphe de singularités (*b*), cette étape nous permet de mettre en évidence les problèmes du champ d'orientation du modèle. Dans le cas présent, nous pouvons voir que le champ d'orientation possède 4 lignes singulières 3-5, en bleu, au niveau de la jonction entre la partie cylindrique avec le reste du modèle ainsi que deux zones de tremplin caractérisées par les sommets géométriques marqués à l'aide d'une sphère rouge. L'utilisateur peut alors sélectionner un ensemble de surfaces du modèle pour la relaxation de contrainte au bord. Ici nous décidons de sélectionner 3 surface pour corriger le champ d'orientation (*c*). D'abord, la surface du cylindre qui est orthogonale au reste du modèle, en effet c'est l'alignement du champ d'orientation avec la courbure de cette surface qui va faire apparaître des singularités de valence 3 au dessus du cylindre et des singularités de valence 5 à la base de celui-ci. Puis, pour les zones de tremplin, nous sélectionnons les surfaces orthogonales aux surfaces triangulaires. En relaxant l'alignement du champ avec ces surfaces, nous permettons au champ de ne plus s'aplatir vers les sommets géométriques. Nous générons une nouvelle fois le champ d'orientation dans le modèle (*d*), celui-ci est alors corrigé des éléments problématiques. Nous pouvons alors réaliser l'étape d'insertion de feuillets duaux présentée au Chapitre 3. Nous commençons par sélectionner un tétraèdre du bord (*e*), les deux directions de coupe vont apparaître. Pour rappel, ces directions sont les normales au plan de coupe dans le tétraèdre sélectionné. Nous choisissons de tracer le feuillet en partant de la direction normale de coupe rouge, ce qui nous permet d'obtenir le résultat en (*f*). Nous pouvons changer l'opacité du maillage, cette opération nous permet de plus facilement visualiser les feuillets duaux insérés (*g*). L'opération d'insertion de feuillets duaux est donc répétée 12 fois pour créer la structure duale (*h*). A la suite de ça, nous allons générer les régions duales en vérifiant que celles-ci valident les règles définies à la Section 3.3.1. Si les règles sont validées alors nous visualisons les régions duales dans notre outil (*i*), puis nous générons la structure de blocs correspondantes (*j*). Enfin nous générons le maillage hexaédrique final (*k*) de la méthode présentée à la Section 5.1. Ces 3 dernières étapes sont automatisées, cependant nous attendons une validation utilisateur pour passer à l'étape suivante, ceci afin de pouvoir visualiser chaque structure.

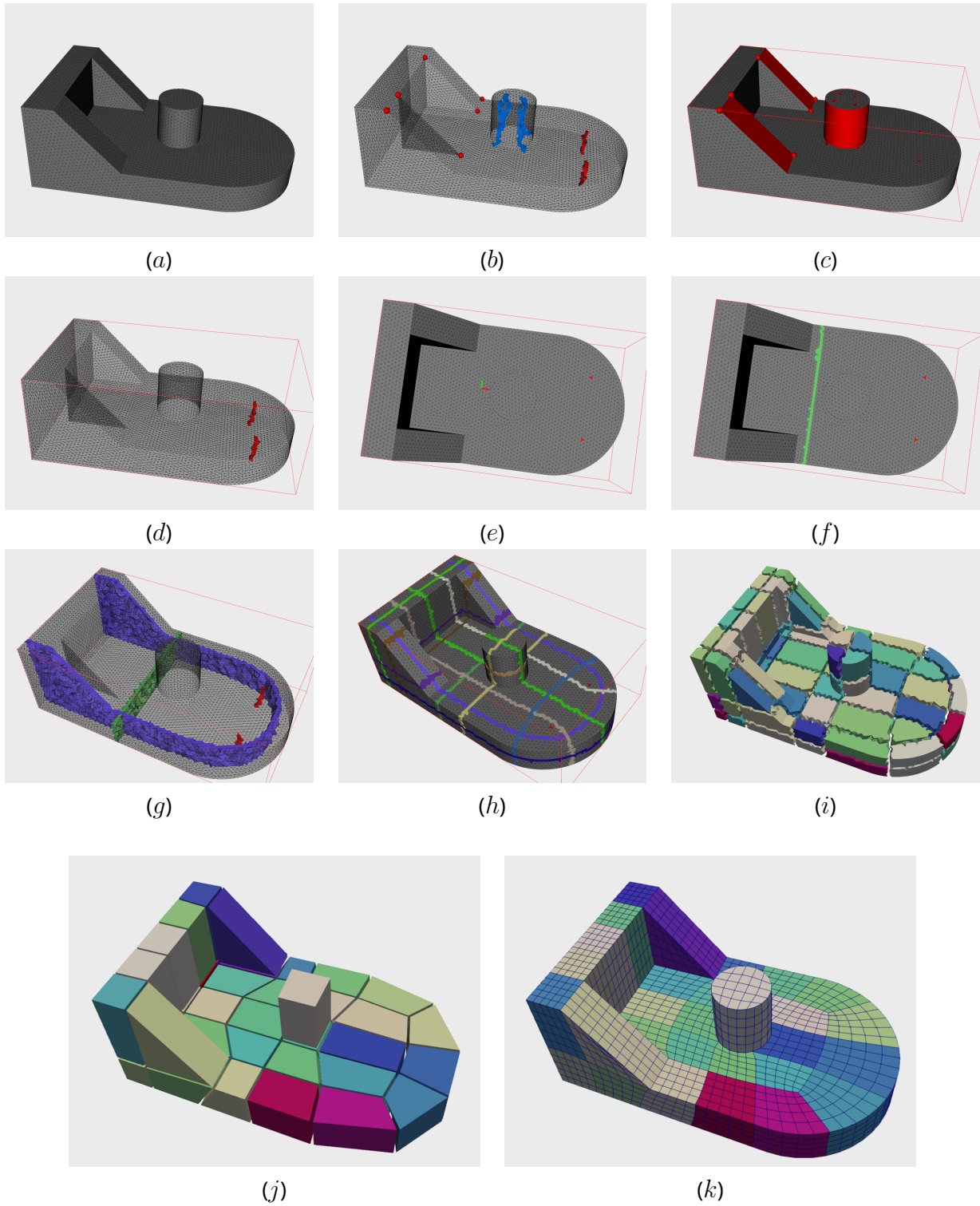


Figure 5.8 – Un exemple d'exécution de notre méthode sur l'outil que nous avons développé.

5.2.2 . Résultats sur des modèles de types V

Nous allons maintenant présenter plusieurs résultats de notre méthode de création de structure de blocs du Chapitre 3 sur des modèles de type V. Pour rappel, le champ d'orientation de ces modèles ne possède aucun élément problématique. Nous pouvons voir à gauche aux figures 5.9, 5.10 et 5.11 que nous pouvons tracer les feuillets duaux dans l'ensemble des domaines, nous n'avons alors qu'à vérifier grâce aux régions duales, au centre sur les figures, que nous avons construit le bon nombre de feuillets duaux. Le cas échéant, comme le champ d'orientation permet de pouvoir insérer des feuillets duaux partout dans le domaine nous n'avons qu'à ajouter des feuillets jusqu'à ce que la structure de blocs valide les règles. Nous pouvons alors générer la structure de blocs correspondante à droite sur les figures.

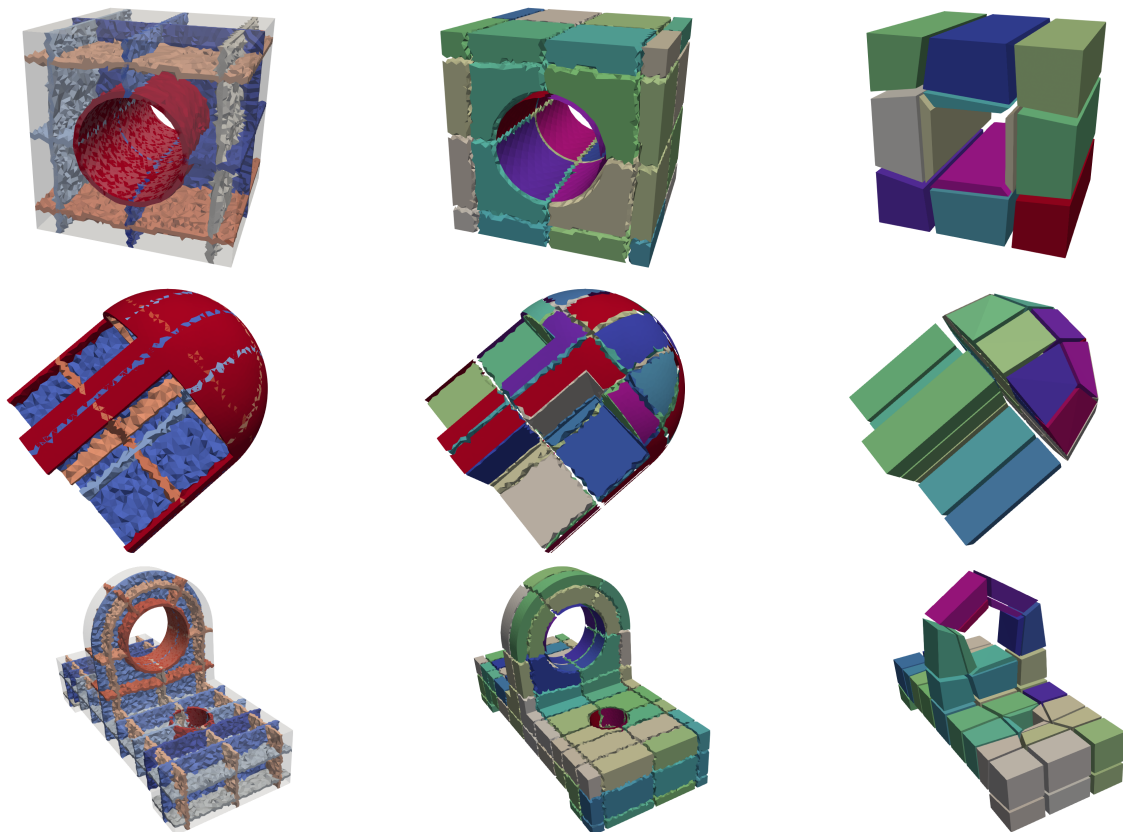


Figure 5.9 – Résultats pour les modèles B10, B58 et S38 avec, les feuillets duaux à gauche, les régions duales au centre et la structure de blocs à droite.

Cette méthode permet de facilement créer une structure de blocs sur n'importe quelle géométrie, si le champ d'orientation est valide alors nous pouvons insérer des feuillets dans le domaine. C'est le cas de l'exemple de la figure 5.12, où notre domaine géométrique a la forme d'un tétraèdre. Le champ d'orientation montré en (a) est valide, nous pouvons effectuer 4 opérations simples d'insertion de feuillet pour obtenir la géométrie maillée en (b).

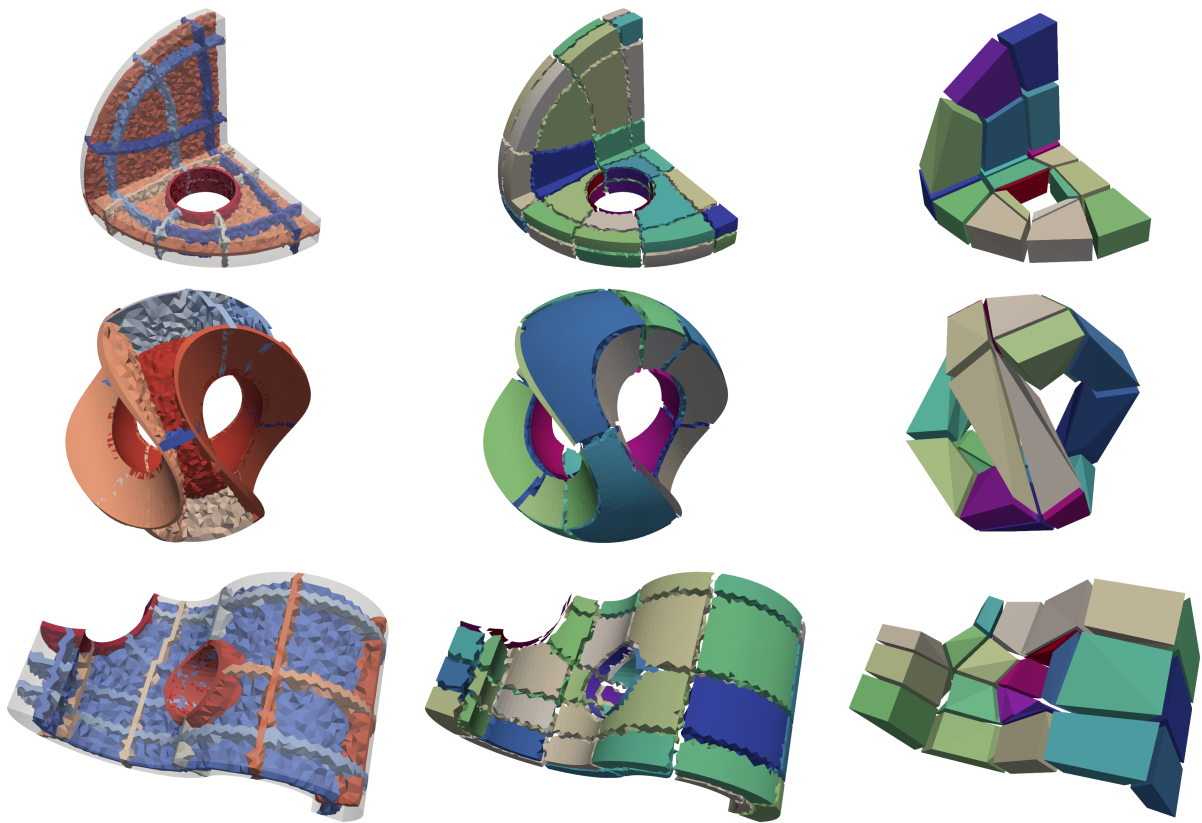


Figure 5.10 – Résultats pour les modèles S34, sculpture-A et B1 avec, les feuilletts duaux à gauche, les régions duales au centre et la structure de blocs à droite.

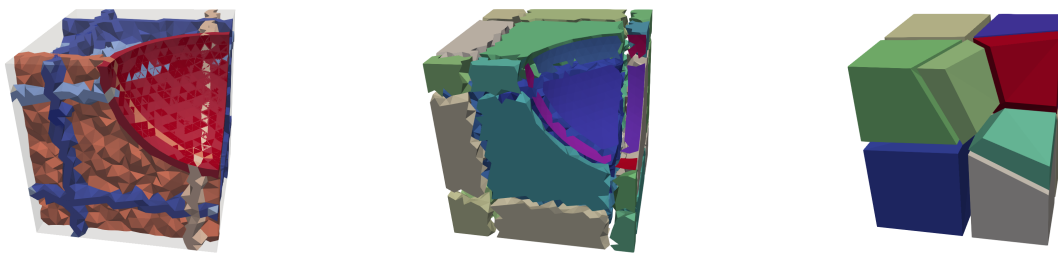


Figure 5.11 – Résultats pour le modèle B8 avec, les feuilletts duaux à gauche, les régions duales au centre et la structure de blocs à droite.

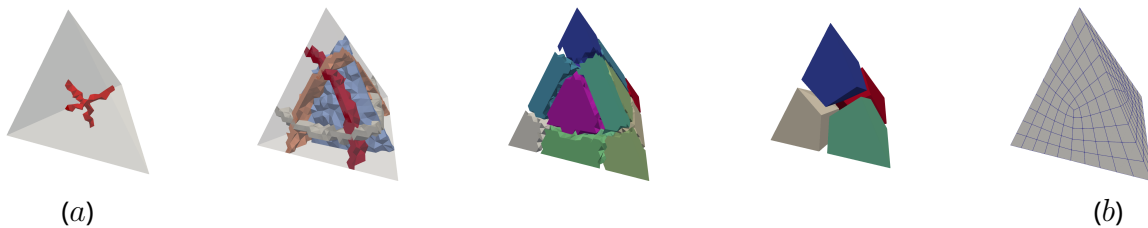


Figure 5.12 – Exemple d'exécution sur une forme tétraédrique, en (a) le champ d'orientation valide nous permet de générer le maillage en (b).

5.2.3 . Résultats sur des modèles de types S et T

Dans cette partie nous allons présenter les résultats de notre méthode sur des modèles contenant des configurations de type S et T où les champs d'orientation ne nous permettent pas de tracer des feuilletts duaux dans l'ensemble du domaine. Ici nous montrons nos résultats aux figures 5.14, 5.15 et 5.16. Ces résultats sont obtenus après l'étape de correction de champ d'orientation présentée au Chapitre 4. Les résultats obtenus par notre méthode sont proches de résultats d'approche par polycubes comme le montre l'exemple de la figure 5.13, où nous supprimons les 4 lignes singulières 3-5, ce qui nous permet d'appliquer notre méthode de génération de structure de blocs pour obtenir un maillage hexaédrique. Cependant nos résultats sont de meilleure qualité, selon les critères exposés à la Section 2.1.2, quand le champ d'orientation d'un modèle possède des lignes singulières valides. En effet, nous relaxons la contrainte d'alignement au bord localement aux éléments problématiques du champ, ce qui nous permet de conserver au maximum un résultat typé champ d'orientation sur le reste du modèle.

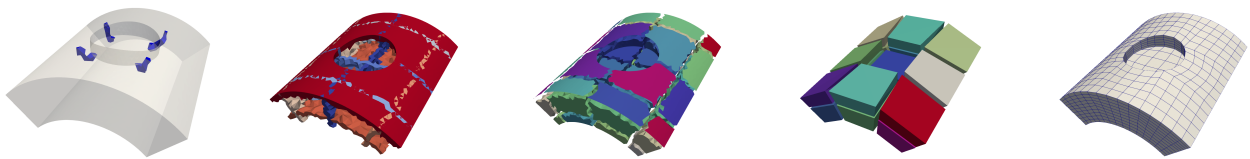


Figure 5.13 – Exemple d'un cas où notre méthode fourni un résultat équivalent à une approche polycube puisque toutes les lignes singulières du champ sont des 3-5, après correction le champ est régulier dans tout le domaine.

De plus il n'existe pas actuellement de méthode robuste sur les champs d'orientation qui permette de fournir de façon automatique des résultats sur des modèles disposant de configurations de types S et T. Certaines méthode automatique se basent sur la même approche que nous, c'est-à-dire qu'elles acceptent de perdre l'alignement au bord au profit d'une robustesse sur l'obtention de maillage hexaédrique. C'est notamment le cas de [ZWG⁺20] qui utilisent la méthode de correction de lignes singulières proposée par [RCR19]. Néanmoins le cas des zones de tremplin reste un problème non géré par les approches automatiques actuelles. Ceci est dû à la complexité d'une définition formelle et générale du problème qui permette d'appliquer un algorithme de correction sur n'importe quel cas de zone de tremplin. L'interactivité ainsi que l'analyse du champ d'orientation nous permet, dans notre cas, de fournir suffisamment d'éléments à l'utilisateur pour gérer les situations rencontrées.



Figure 5.14 – Résultats pour les modèles M6, S24 et B79 avec, les feuilletaux à gauche, les régions duales au centre et la structure de blocs à droite.

Nous avons indiqué plusieurs fois dans le document que la géométrie influe grandement sur la qualité du champ d'orientation généré. Pour illustrer nos propos nous avons créé quatre formes triangulaires en faisant varier les angles des triangles, ces géométries sont montrées à la figure 5.17. Le cas (a) est un exemple de zone de tremplin basique, aucune ligne singulière du champ n'apparaît au centre du triangle, nous pouvons néanmoins voir une singularité au niveau de la pointe du triangle. En (b) le champ d'orientation à un comportement particulier, nous retrouvons des singularités aux pointes du triangle où l'angle est de 45° , mais dans ce cas une ligne singulière est apparue cependant elle est très proche d'une pointe du triangle. Le champ d'orientation de la géométrie (b) est toujours dirigé vers les sommets géométriques, même avec la ligne singulière nous sommes toujours dans le cas d'une zone de tremplin. Le champ d'orientation de la géométrie (c) ne possède qu'une ligne singulière au centre du triangle qui ne gêne en aucun cas le tracer de feuilletaux. Nous pouvons également voir à la figure 5.18 les résultats de notre méthode de génération de maillage sur les cas de la figure 5.17(b), en ayant corrigé le champ d'orientation, ce qui nous permet d'obtenir un maillage typé polycube et (c) où nous obtenons les 3 blocs autour de la singularité et un maillage de bonne qualité.

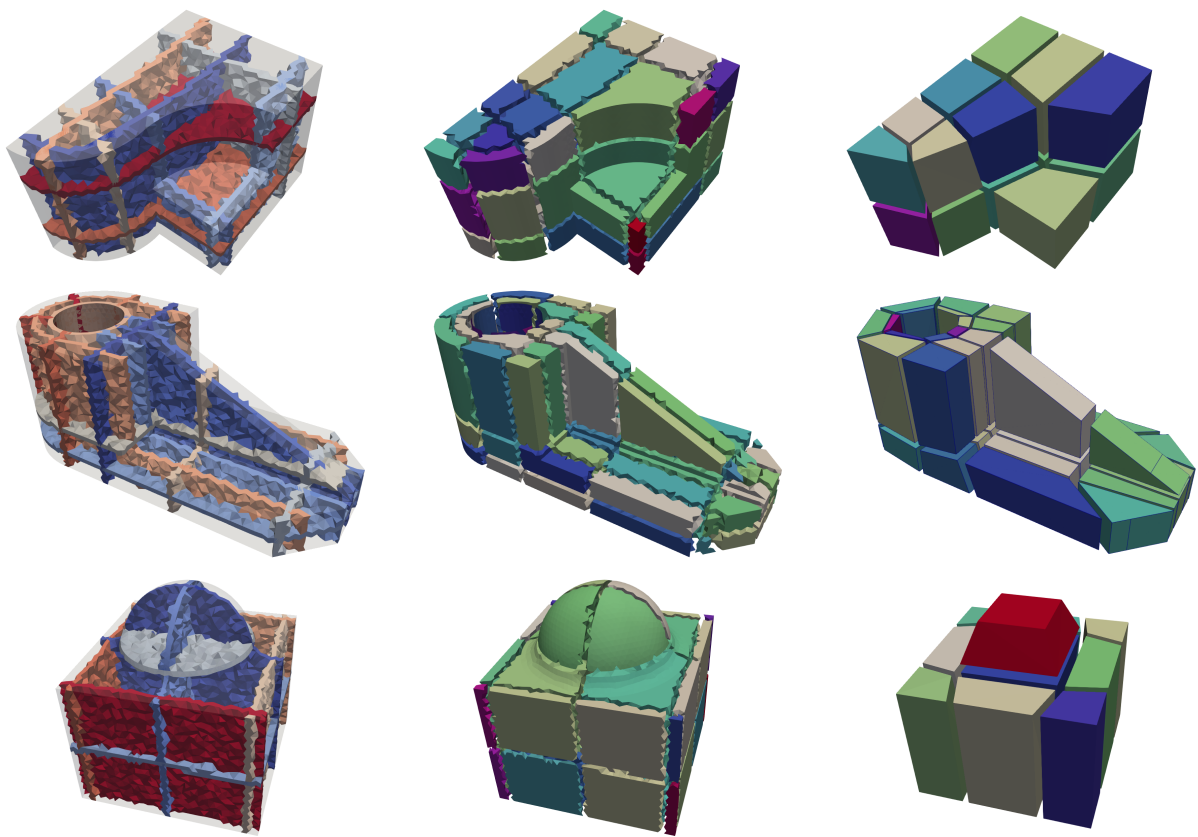


Figure 5.15 – Résultats pour les modèles notch-2, S33 et B25 avec, les feuillets duaux à gauche, les régions duales au centre et la structure de blocs à droite.

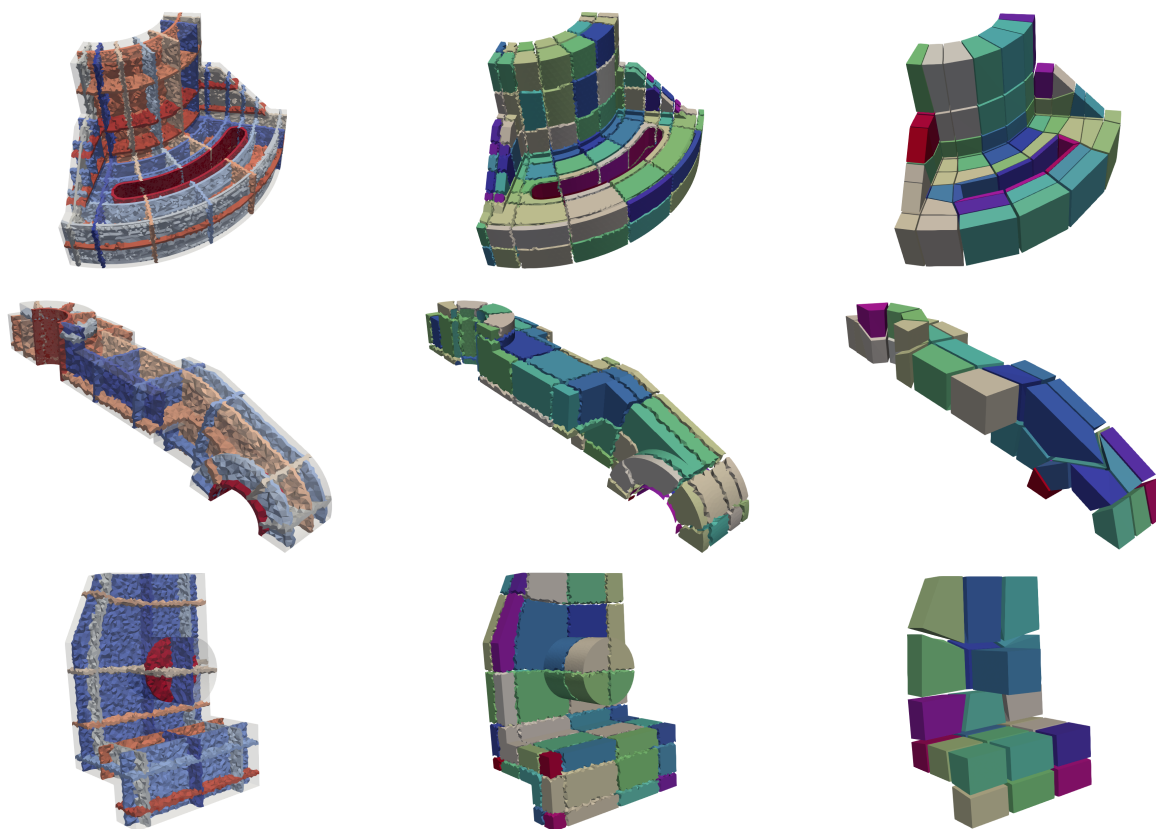


Figure 5.16 – Résultats pour les modèles S10, S27 et S42 avec, les feuilletés duaux à gauche, les régions duales au centre et la structure de blocs à droite.

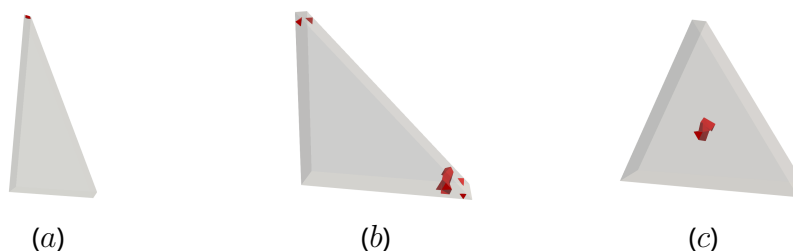


Figure 5.17 – Quatre cas de formes triangulaires avec leur graphe de singularité en rouge. Le champ d'orientation des cas a et b n'est pas valide, il doit être corrigé. Le cas c quant à lui possède une seule singularité au centre ce qui nous permet directement d'appliquer notre méthode d'insertion de feuilletés duaux.

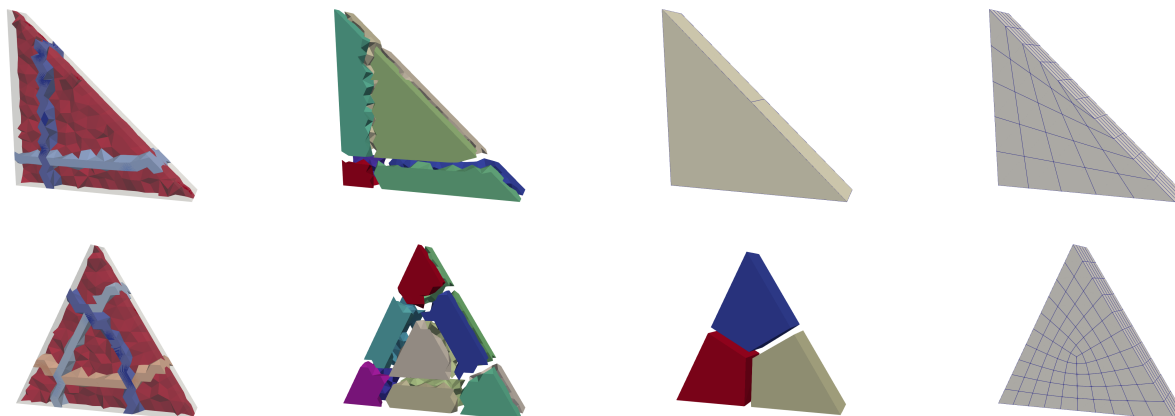


Figure 5.18 – Application de notre méthode de génération de maillage sur les cas (b), en haut, et (c), en bas, de la figure 5.17.

6 - Conclusion et perspectives

L'objectif de ce travail de thèse était d'étudier et de proposer de nouvelles solutions interactives pour la génération de maillages hexaédriques structurés par blocs. Ces maillages requis dans de nombreux domaines de la simulation numérique sont en pratique créés aujourd'hui de façon interactive par des ingénieurs experts à l'aide de logiciels graphiques. Cette activité, très consommatrice en temps ingénieur, peut nécessiter plusieurs semaines de travail pour des géométries complexes. Cette durée excessive devient aujourd'hui un écueil majeur pour les études amonts où des modèles sont conçus numériquement par modifications successives, nécessitant de réaliser un maillage à chaque itération.

Dans ce contexte, nous avons proposé de nous appuyer sur l'utilisation d'un champs d'orientations 3D, "objet" géométrique qui fournit une information continue de directions au sein du volume à discrétiser pour guider la création interactive de structures de blocs. Parmi les grandes familles d'algorithmes de maillage hexaédrique aujourd'hui étudiés, les méthodes à base de champs d'orientations permettent de générer des maillages hexaédriques structurés par blocs pour de nombreux domaines géométriques simples. Par contre, elles échouent dès lors que les modèles rencontrés sont plus réalistes. En se basant sur la connaissance d'un champ d'orientation défini sur le domaine Ω à discrétiser, nous pouvons piloter un outil interactif à l'aide d'une information volumique supplémentaire.

Au chapitre 3, nous avons justement introduit une approche interactive pour générer des structures de blocs à partir de champs d'orientations. Cette étape nécessite de disposer d'un maillage tétraédrique T_Ω suffisamment raffiné du domaine Ω et d'un champ d'orientation défini aux sommets de T_Ω . Dans ce chapitre, nous partons du postulat que le champ d'orientation dispose d'une structure compatible avec celle d'une structure de blocs hexaédriques. En pratique, on évalue la validité du champ en détectant les configurations de type S et T comme indiqué dans la première partie du chapitre 4. Dans ces conditions, notre approche permet de manière interactive de construire le dual d'une structure de blocs par insertion successive de surfaces duales. Ces surfaces partitionne T_Ω en un ensemble de régions duales discrètes formées de tétraèdres, régions desquelles nous extrayons une structure de blocs. Cette extraction est contrôlée par un ensemble de règles de validité qui nous assurent que la structure de blocs générée finalement est correctement classifiée et discrétise bien le domaine géométrique Ω .

Puisque les champs d'orientations générés par les méthodes usuelles de l'état de l'art telles que celle que nous utilisons [RSL16] n'ont que rarement une topologie compatible avec celle d'un maillage hexaédrique structuré par blocs, l'approche proposée au chapitre 3 n'est utilisable directement que pour peu de domaines géométriques. C'est pourquoi le chapitre 4 introduit à la fois une méthode pour détecter les configurations problématiques pour l'algorithme de génération de champ, et une approche simple pour les éliminer en relaxant des contraintes d'alignement au bord en préambule de la génération des champs d'orientations. D'un point de vue interactif, la relaxation de contraintes est "facile" à appréhender humainement et nécessitent que peu d'interactions 3D nouvelles à définir.

Une seconde approche étudiée au chapitre 4 a été de découper le domaine géométrique en sous-domaine pour isoler les zones problématiques. En pratique, les différents essais que nous avons pu effectuer pour effectuer des découpes ont montré que :

1. La sélection des zones à découper peut demander l'implémentation de nombreux interacteurs 3D supplémentaires pour isoler les zones. On aura par exemple besoin dans de nombreux cas d'isoler des portions de surfaces, et non pas des surfaces complètes, en traçant des lignes dans le champ surfacique. Ceci nécessite de nombreux développements informatiques supplémentaires.

2. Une fois les découpes effectuées et les blocs générés dans chaque sous-domaine, il reste encore à les recoller entre eux de manière conforme. Si d'un point de vue théorique, nous avons identifié que le recollement est possible selon les propriétés du champs sur la surface de contact ¹, il reste un travail important d'implémentation à réaliser.

Dans les deux cas de figure, la quantité de développement à réaliser nous a poussé à envisager cette option comme une perspective de travail à court terme de ces travaux de thèse.

Enfin, en permettant de nettoyer les champs d'orientations des configurations de type S et T , nous avons pu générer en quelques interactions des structures de blocs pour des modèles non couverts aujourd'hui par les méthodes automatiques de maillage hexaédriques basées sur les champs d'orientations. En particulier, nous avons montré que les contributions cumulées des chapitres 3 et 4 permettent de générer des maillages structurés par blocs pour divers modèles tels que ceux présentés au chapitre 5. En comparaison avec l'utilisation d'outils interactifs tels que Magix3D, développé au CEA, le gain de temps pour l'utilisateur est important pour les modèles testés.

6.1 . Perspectives

6.1.1 . Automatisation de la génération de la structure duale

En présence d'un modèle de type V , l'approche interactive introduite au chapitre 3 permet de créer les structures de blocs souhaitées. Si notre objectif initial était d'explorer l'utilisation des champs d'orientations dans un contexte interactif, cette étape peut *a priori* être totalement automatisée. En s'appuyant sur les règles de validité que nous avons définies, il est possible d'insérer itérativement l'ensemble des feuillettes duales nécessaires jusqu'à obtention du résultat souhaité.

C'est un travail que nous allons mener à court terme car il permettra de réduire la partie interactive de notre travail à la seule correction des configurations de type S et T . En outre, cela évitera aussi à l'utilisateur de devoir comprendre et visualiser les champs d'orientations 3D, ce qui n'est pas nécessairement aisée.

6.1.2 . Coupe d'un modèle géométrique à l'aide de l'information de champ d'orientation

Au chapitre 4, nous avons étudié la possibilité de découper un domaine géométrique discrétisé à l'aide d'un maillage tétraédrique disposant d'un champ d'orientation discret. Notre objectif était de scinder un modèle disposant de configurations de type S et T en plusieurs sous-modèles plus simples de type V . En pratique, comme indiqué précédemment, nous nous sommes heurtés à plusieurs limitations. En premier lieu, la sélection du bord de la coupe n'est pas optimale pour plusieurs raisons :

- Elle est très dépendante du maillage tétraédrique. Il arrive que des tétraèdres possèdent une face sur une surface sélectionnée et une autre face sur une surface non sélectionnée ce qui pose problème par la suite pour la reconstruction du bord. Il faudrait alors modifier les éléments que nous sélectionnons et par conséquent modifier également le modèle du graphe utilisé pour l'algorithme de coupe ;
- La méthode de sélection ne permet d'obtenir un bord de coupe viable que lorsque celui-ci c'est bien défini sur la géométrie. On revient ici sur le problème qu'il doit être possible de sélectionner des portions de surfaces et non des surfaces entières. Ceci nécessite de nombreux développements supplémentaires aussi bien au niveau interactif qu'au niveau de la représentation informatique des surfaces et de la classification géométrique entre blocs et entités de CAO. En pratique, il faudrait modifier de façon intrusive ces notions au sein de la bibliothèque GMDS et du logiciel MGX pour obtenir les résultats souhaités. De telles interventions étaient hors du cadre possible de la thèse mais mériteront d'être étudiées à l'avenir.

1. Le champ doit être totalement régulier dans cette surface

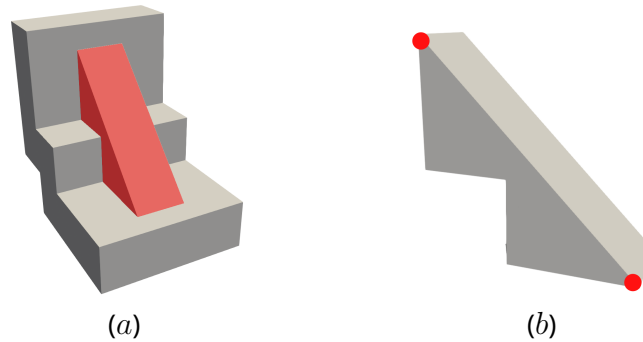


Figure 6.1 – Nous découpons la zone rouge, en (a) du reste de la géométrie. La forme obtenue en (b) reste difficilement à mailler, elle possède deux zones de tremplin.

Ensuite, notre méthode de reconstruction du bord permet de reconnecter des sommets géométriques entre eux afin de recréer des courbes et par la suite des surfaces sur la coupe. Notre approche nous demande d'avoir un nombre pair de sommets à reconnecter. Dans le cas contraire une opération d'insertion de sommet géométrique est nécessaire, de plus cela peut amener à obtenir un sommet qui doit se reconnecter à plusieurs autres sommets. Cette opération pourrait être effectuée à l'aide d'actions interactives, cependant nous aimerions minimiser le nombre d'interaction ainsi que le temps ingénieur et maximiser l'automatisation.

Enfin, comme précisé ci-avant, le but de la coupe géométrique était d'obtenir un ensemble de modèles de type V, malheureusement nous n'avons aucune garantie, même en effectuant plusieurs coupes, que tous les sous-modèles obtenus après les coupes seront de type V. La figure 6.1 nous montre un tel cas. Nous découpons la géométrie au niveau de la zone sélectionnée (en rouge) en (a). Cependant la forme obtenue n'est pas une configuration de type V, elle possède deux zones de tremplin et ne peut donc pas être utilisée pour notre méthode d'insertion de feuillets duaux.

6.1.3 . Automatisation de la correction de champs d'orientation

La correction automatique de lignes singulières 3-5 est déjà proposée par [RCR19]. Il est donc tout à fait possible d'automatiser cette étape dans notre contexte aussi.

Par contre, nous avons présenté une méthode d'analyse de champ d'orientation et de détection de sommet caractéristiques de zones de tremplin qui elle semble plus difficile à automatiser en suivant une approche classique. En effet, selon les cas de figure, la sélection des zones de tremplin ne se fait pas toujours de la même façon. On peut par contre estimer qu'il serait possible de cataloguer les différentes configurations rencontrées et les solutions apportées interactivement par l'utilisateur pour lui proposer ensuite dans des situations similaires. On peut envisager une solution totalement automatique, ou bien semi-assisté où des sélections possibles seraient proposées à l'utilisateur.

6.1.4 . Optimisation topologique de la structure de blocs

La relaxation des contraintes de bord sur les champs d'orientations nous permet d'éliminer les configurations de type S et T mais implique que la structure de blocs obtenue ne respecte plus les contraintes d'alignement au bord partout le long du bord du domaine. Il apparaît alors nécessaire d'introduire des couches supplémentaires de blocs hexaédriques au bord du domaine pour pouvoir récupérer cet alignement. Ceci peut se réaliser simplement de manière globale en post-traitement comme dans bon nombre de méthodes de type Polycube. Malheureusement, l'amélioration de qualité n'est pas globale et si on ajoute bien des blocs aux endroits souhaités, on en introduit aussi dans des zones qui étaient auparavant correctes et pour lesquelles la nouvelle configuration est moins optimale. Les auteurs de [CAS+19] permettent une insertion "sélective" localisée. En résolvant un problème global en programmation linéaire en nombres entiers sous contraintes, ils sélectionnent une nappe de faces du maillage le long de laquelle une couche de mailles sera insérée. Cette

approche mériterait d'être appliquée dans notre cas où le fait de travailler sur la structure de blocs réduit grandement le nombre d'inconnues du problème à résoudre.

Bibliographie

- [And18] Andrew Makhorin. Glpk, 2018.
- [ANS] ANSYS. *ICEM CFD brochure*. <https://www.ansys.com/-/media/Ansys/corporate/resourcelibrary/brochure/ansys-icem-cfd-brochure.pdf>.
- [BK04] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26 :1124–1137, September 2004.
- [BRM⁺14] Tristan Carrier Baudouin, Jean-François Remacle, Emilie Marchandise, François Henrotte, and Christophe Geuzaine. A frontal approach to hex-dominant mesh generation. *Advanced Modeling and Simulation in Engineering Sciences*, 1(1) :1–30, 2014.
- [CAS⁺19] G. Cherchi, P. Alliez, R. Scateni, M. Lyon, and D. Bommès. Selective padding for polycube-based hexahedral meshing. *Computer Graphics Forum*, 38(1) :580–591, 2019.
- [CBK12] M. Campen, D. Bommès, and L. Kobbelt. Dual loops meshing. *ACM Transactions on Graphics (TOG)*, 31 :1 – 11, 2012.
- [CLS16] Gianmarco Cherchi, Marco Livesu, and Riccardo Scateni. Polycube simplification for coarse layouts of surfaces and volumes. *Computer Graphics Forum*, 35(5) :11–20, 2016.
- [CO82] W A Cook and W R Oakes. Mapping method for generating three-dimensional meshes : past and present. 1 1982.
- [Cor21] CoreForm. CoreForm, 2021.
- [Dij59] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische matematik*, 1(1) :269–271, 1959.
- [Dis20] Distene SAS. MeshGems, 2020.
- [EDF20] EDF. Code_aster, 2020.
- [EDF21a] EDF. Code_saturne, 2021.
- [EDF21b] EDF. Salome, 2021.
- [FG08] Pascal Frey and Paul George. *Mesh Generation : Application to Finite Elements : Second Edition*. Iste, 2008.
- [FXBH16] Xianzhong Fang, Weiwei Xu, Hujun Bao, and Jin Huang. All-hex meshing using closed-form induced polycube. *ACM Trans. Graph.*, 35(4) :124 :1–124 :9, 2016.
- [GR09] Christophe Geuzaine and Jean-François Remacle. Gmsh : A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79 :1309 – 1331, 2009.
- [GSP19] Xifeng Gao, Hanxiao Shen, and Daniele Panozzo. Feature preserving octree-based hexahedral meshing. *Computer Graphics Forum*, 38(5) :135–149, 2019.
- [GSZ11] James Gregson, Alla Sheffer, and Eugene Zhang. All-hex mesh generation via volumetric polycube deformation. *Computer Graphics Forum*, 30(5) :1407–1416, 2011.
- [GT82] William J. Gordon and Linda C. Thiel. Transfinite mappings and their application to grid generation. *Applied Mathematics and Computation*, 10-11 :171–233, 1982.
- [HJS⁺14] Jin Huang, Tengfei Jiang, Zeyun Shi, Yiyong Tong, Hujun Bao, and Mathieu Desbrun. ℓ_1 -based construction of polycube maps from complex shapes. *ACM Trans. Graph.*, 33(3) :25 :1–25 :11, 2014.

- [HQZ13] Kangkang Hu, Jin Qian, and Yongjie Jessica Zhang. Adaptive all-hexahedral mesh generation based on a hybrid octree and bubble packing. In *Proceedings of International Meshing Roundtable*, 2013.
- [HTWB11] Jin Huang, Yiyong Tong, Hongyu Wei, and Hujun Bao. Boundary aligned smooth 3d cross-frame field. *ACM Trans. Graph.*, 30(6) :1–8, 2011.
- [ISS09] Yasushi Ito, Alan M Shih, and Bharat K Soni. Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates. *International Journal for Numerical Methods in Engineering*, 77(13) :1809–1833, 2009.
- [JFT99] Nigel P. Weatherill Joe F. Thompson, Bharat K. Soni, editor. CRC Press, 1999.
- [KLF16] Nicolas Kowalski, Franck Ledoux, and Pascal Frey. Smoothness driven frame field generation for hexahedral meshing. *Computer-Aided Design*, 72 :65–77, 2016.
- [KLSO12] N. Kowalski, Franck Ledoux, Matthew Staten, and Steven Owen. Fun sheet matching : Towards automatic block decomposition for hexahedral meshes. *Engineering With Computers - EWC*, 28 :1–13, 07 2012.
- [Knu00] 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. *International Journal for Numerical Methods in Engineering*, 48(8) :1165–1185, 2000.
- [Knu01] Patrick M. Knupp. Algebraic mesh quality metrics. *SIAM Journal on Scientific Computing*, 23(1) :193–218, 2001.
- [Knu07] Patrick Knupp. Remarks on mesh quality. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2007.
- [Kow13] Nicolas Kowalski. *Domain partitioning using frame fields : applications to quadrilateral and hexahedral meshing*. PhD thesis, Université Pierre et Marie Curie, Paris, 2013.
- [Lab] Sandia National Laboratories. *CUBIT*. <https://cubit.sandia.gov/>.
- [LBK16] Max Lyon, David Bommès, and Leif Kobbelt. Hexex : robust hexahedral mesh extraction. *ACM Trans. Graph.*, 35(4) :123, 2016.
- [LPC21] Marco Livesu, Luca Pitzalis, and Gianmarco Cherchi. Optimal dual schemes for adaptive grid based hexmeshing. *ACM Trans. Graph.*, 2021.
- [LPP⁺20] Marco Livesu, Nico Pietroni, Enrico Puppo, Alla Sheffer, and Paolo Cignoni. Loopycuts : Practical feature-preserving block decomposition for strongly hex-dominant meshing. *ACM Trans. Graph.*, 39(4), 2020.
- [LVS⁺13] Marco Livesu, Nicholas Vining, Alla Sheffer, James Gregson, and Riccardo Scateni. Polycut : Monotone graph-cuts for polycube base-complex construction. *ACM Trans. Graph.*, 32(6) :171 :1–171 :12, 2013.
- [LW08] Franck Ledoux and Jean-Christophe Weill. An extension of the reliable whisker weaving algorithm. In *Proceedings of International Meshing Roundtable*, pages 215–232, 2008.
- [LZC⁺18] Heng Liu, Paul Zhang, Edward Chien, Justin Solomon, and David Bommès. Singularity-constrained octahedral fields for hexahedral meshing. *ACM Trans. Graph.*, 37(4), jul 2018.
- [LZS⁺21] Lingxiao Li, Paul Zhang, Dmitriy Smirnov, S Mazdak Abulnaga, and Justin Solomon. Interactive all-hex meshing via cuboid decomposition. *arXiv preprint arXiv :2109.06279*, 2021.
- [Mar09] Loïc Maréchal. Advances in octree-based all-hexahedral mesh generation : Handling sharp features. In *Proceedings of International Meshing Roundtable*, pages 65–84, 2009.

- [MBBM97] Peter Murdoch, Steven Benzley, Ted Blacker, and Scott A. Mitchell. The spatial twist continuum : A connectivity based method for representing all-hexahedral finite element meshes. *Finite Elements in Analysis and Design*, 28(2) :137 – 149, 1997.
- [MBT97] D J Melander, S E Benzley, and T J Tautges. Generation of multi-million element meshes for solid model-based geometries : The dicer algorithm. 6 1997.
- [MCR19] Reberol Maxence, Alexandre Chemin, and Jean-François Remacle. Multiple approaches to frame field correction for cad models. 12 2019.
- [Mit05] William Mitchell. Hamiltonian paths through two- and three-dimensional grids. *Journal of Research of the National Institute of Standards and Technology*, 110 :127, 03 2005.
- [MT95] Scott A. Mitchell and Timothy J. Tautges. Pillowing doublets : Refining a mesh to ensure that faces share at most one edge. In *Proceedings of International Meshing Roundtable*, 1995.
- [NRP11] M. Nieser, U. Reitebuch, and K. Polthier. Cubecover–parameterization of 3d volumes. *Computer Graphics Forum*, 30(5) :1397–1406, 2011.
- [PA97a] M. A. Price and C. G. Armstrong. Hexahedral mesh generation by medial surface subdivision : part ii. solids with flat and concave edges. 40(1) :111–136, jan 1997.
- [PA97b] M.A. Price and C.G. Armstrong. Hexahedral mesh generation by medial surface subdivision : Part ii. solids with flat and concave edges. *International Journal for Numerical Methods in Engineering*, 40(1) :111–136, 1997.
- [PAS95] M.A. Price, C.G. Armstrong, and M.A. Sabin. Hexahedral mesh generation by medial surface subdivision : Part i. solids with convex edges. *International Journal for Numerical Methods in Engineering*, 38(19) :3335–3359, 1995.
- [PBS20] David Palmer, David Bommès, and Justin Solomon. Algebraic representations for volumetric frame fields. *ACM Trans. Graph.*, 39(2), 2020.
- [PJR17] Jeanne Pellerin, Amaury Johnen, and Jean-François Remacle. Combining tetrahedra into hexahedra : a vertex based strategy. 05 2017.
- [PJVR18] Jeanne Pellerin, Amaury Johnen, Kilian Verhetsel, and Jean-François Remacle. Identifying combinations of tetrahedra into hexahedra : A vertex based strategy. *Computer-Aided Design*, 105 :1–10, 2018.
- [PLC⁺21] Luca Pitzalis, Marco Livesu, Gianmarco Cherchi, Enrico Gobbetti, and Riccardo Scateni. Generalized adaptive refinement for grid-based hexahedral meshing. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 40(6), 2021.
- [PRR⁺20] François Protais, Maxence Reberol, Nicolas Ray, Etienne Corman, Franck Ledoux, and Dmitry Sokolov. Robust Quantization for Polycube Maps. working paper or preprint, December 2020.
- [Qua14] William Roshan Quadros. Laytracks3d : A new approach to meshing general solids using medial axis transform. *Procedia Engineering*, 82 :72 – 87, 2014.
- [RCR19] Maxence Reberol, Alexandre Chemin, and Jean-François Remacle. Multiple approaches to frame field correction for cad models. *arXiv preprint arXiv :1912.01248*, 2019.
- [RGSR09] Eloi Ruiz Gironès and Josep Sarrate Ramos. *Automatic generation of quadrilateral structured meshes using linear programming and transfinite interpolation*. Apr 2009.
- [RS03] J.-F. Remacle and M.S. Shephard. An algorithm oriented mesh database. *International Journal for Numerical Methods in Engineering*, 58(2), 2003.
- [RSL16] Nicolas Ray, Dmitry Sokolov, and Bruno Lévy. Practical 3d frame field generation. *ACM Trans. Graph.*, 35(6) :233, 2016.

- [Sch96] R. Schneiders. A grid-based algorithm for the generation of hexahedral element meshes. *Engineering with Computers*, 12(3) :168–177, 1996.
- [Sch97] R. Schneiders. An algorithm for the generation of hexahedral element meshes based on a octree technique. In *Proceedings of International Meshing Roundtable*, pages 183–194, 1997.
- [Si15] Hang Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software*, 41(2) :1–36, 2015.
- [SIM] SIMULIA. *Abaqus, Non-Linear Finite Element Analysis*. <https://www.simuleon.com/simulia-abacus/>.
- [SKO⁺10] Matthew L. Staten, Robert A. Kerr, Steven J. Owen, Ted D. Blacker, Marco Stupazzini, and Kenji Shimada. Unconstrained plastering - hexahedral mesh generation via advancing-front geometry decomposition. *International Journal for Numerical Methods in Engineering*, 81 :135–171, 2010.
- [SKOB06] Matthew L. Staten, Robert A. Kerr, Steven J. Owen, and Ted D. Blacker. Unconstrained paving and plastering : Progress update. In *Proceedings of International Meshing Roundtable*, 2006.
- [SML06] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualisation Toolkit*. Kitware, 4 edition, 2006.
- [SRRGRN14] Josep Sarrate Ramos, Eloi Ruiz-Gironés, and Francisco Javier Roca Navarro. Unstructured and semi-structured hexahedral mesh generation methods. *Computational Technology Reviews*, 10 :35–64, 2014.
- [Tak19] K. Takayama. Dual Sheet Meshing : An Interactive Approach to Robust Hexahedralization. *Computer Graphics Forum*, 2019.
- [TBM96] Timothy J Tautges, Ted Blacker, and Scott A Mitchell. The whisker weaving algorithm : a connectivity-based method for constructing all-hexahedral finite element meshes. *International Journal for Numerical Methods in Engineering*, 39(19) :3327–3349, 1996.
- [THCM04] Marco Tarini, Kai Hormann, Paolo Cignoni, and Claudio Montani. Polycube-maps. *ACM Trans. Graph.*, 23(3), 2004.
- [VSL16] Ryan Viertel, M. Staten, and F. Ledoux. Analysis of non-meshable automatically generated frame fields. 2016.
- [WWBB05] M. C. Whiteley, David R. White, Steven E. Benzley, and Ted D. Blacker. Two and three-quarter dimensional meshing facilitators. *Engineering with Computers*, 12 :144–154, 2005.
- [ZB04] Yongjie Zhang and Chandrajit L. Bajaj. Adaptive and quality quadrilateral/hexahedral meshing from volumetric imaging data. In *Proceedings of International Meshing Roundtable*, pages 365–376, 2004.
- [ZWG⁺18] Z. Zheng, R. Wang, S. Gao, Y. Liao, and M. Ding. Dual surface based approach to block decomposition of solid models. *Proceedings of the 26th International Meshing Roundtable*, 2018.
- [ZWG⁺20] Zhihao Zheng, Rui Wang, Shuming Gao, Yizhou Liao, and Mao Ding. Automatic block decomposition based on dual surfaces. *Comput. Aided Des.*, 127 :102883, 2020.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France