Génération automatisée de maillages hexaédriques par balayage généralisé

Dmitry Sokolov, Nicolas Ray, Franck Ledoux
dmitry.sokolov@univ-lorraine.fr, nicolas.ray@inria.fr, franck.ledoux@cea.fr

Nancy, LORIA, Pixel team
`https://pixel.inria.fr`

1 Contexte

En informatique, on représente souvent les objets géométriques par des maillages. En particulier, les volumes sont en général représentés par des maillages composés de tétraèdres, car leur génération et leur manipulation sont assez bien maitrisées. Néanmoins, certaines simulations numériques préfèrent travailler avec des hexaèdres (petits cubes déformés), ce qui pose le problème de convertir des maillages tetraédriques en maillage hexaédriques. Ce problème est extrêmement complexe du fait des contraintes combinatoires fortes auxquelles sont soumis les maillages hexaédraux (contrairement aux maillages tetraédraux).

Notre défi est de construire des solutions de maillage hexaédrique automatique, de qualité comparable aux modèles faits à la main, mais avec un gain de temps considérable grâce à l'automatisation. La génération de maillage est une étape importante dans le processus d'analyse technique, souvent négligée pour privilégier le développement des solveurs. L'analyse des activités montre qu'elle représente maintenant un goulot d'étranglement pour le développement et l'adoption de la simulation numérique dans l'industrie. Le projet met donc l'accent sur l'interaction entre la recherche et les partenaires industriels qui s'engagent à bêta-test et valider les résultats. Nous prévoyons confronter les résultats de la recherche aux besoins de l'industrie, et orienter les axes de recherche vers des maillages qui se sont avérés utiles pour l'industrie.

Malgré tous les échecs passés dans le développement de méthodes de maillage hexaédrique automatisé, à comparer avec la maturité et l'efficacité des approches de maillage tétraédrique, les attentes et les besoins du monde de la simulation numérique pour des méthodes hexaédriques efficaces sont encore extrêmement élevés.

Une idée nouvelle et originale pour générer des maillages hexaédriques a été proposée récemment [SRUL16, RSL16]. Elle provient de l'observation que des mailles hexaédriques de bonne qualité ressemblent presque partout à une grille déformée. L'idée de paramétrisation globale est de construire une transformation f du domaine vers un espace paramétrique, où le domaine déformé peut être maillé par une grille régulière. La transformation inverse f^{-1} appliquée à cette grille produit le maillage hexaédrique du domaine, aligné avec le bord de l'objet. La force de cette approche est que la transformation peut admettre certaines discontinuités. La figure 1 nous montre un exemple : nous partons d'un maillage tétraédrique (à gauche) et nous voulons le déformer de manière à ce que son bord soit aligné avec la grille unitaire. Pour permettre une arête singuliere (valence 3) dans le resultat (à droite), le maillage d'entrée est découpé le long des faces marquées en rouge. La grille entière unitaire induit alors le maillage hexaédrique de topologie désirée.

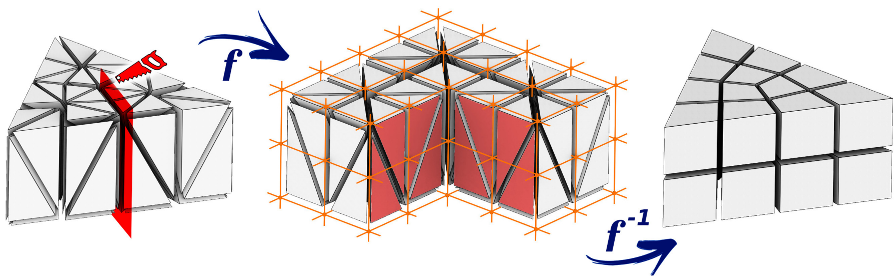

Figure 1 – Maillage hexaédrique par paramétrisation globale. **À gauche :** le maillage tétraédrique d'entrée. Pour permettre une arête singuliere au centre, le maillage est coupé le long du plan rouge. **Au milieu :** le maillage dans l'espace paramétrique ; notez que le bord est aligné avec les axes du système des coordonnées. **À droite :** le maillage final.

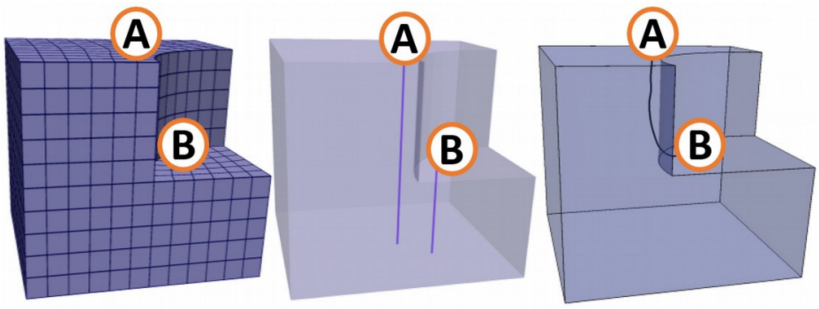

Figure 2 – Le graphe de singularités d'un champ de direction est valide lorsqu'il suit une direction de balayage possible (milieu), mais peut être invalide sinon (droite)

2 Difficulité et piste de recherche

Les paramétrisations globales sont construites en deux étapes : l'étape *Frame Field (FF)* définit l'orientation de la grille à chaque point du domaine, et l'étape *Cube Covering (CC)* génère une paramétrisation globale qui alignera le maillage final avec le champ d'orientation *(FF)* défini dans la première étape. Quand les deux étapes réussiront, le résultat est un maillage completement hexaédrique très régulier. Malheureusement, l'étape *FF* peut générer un champ d'orientation avec une structure topologique (son graphe de singularités) qui n'est pas valide pour la deuxième étape, et l'étape *CC* génère souvent des mappings invalides avec (localement) des Jacobiens négatifs. Notre objectif est d'améliorer la qualité et la robustesse de ces deux étapes.

Le point bloquant actuellement est la génération du champs de directions *FF*. Actuellement, les *Frame field* sont générés par une heurisitque qui minimise la courbure du champ ; ce choix est raisonnable pour minimiser la déformation de la grille. Cependant, il peut produire des graphs de singularités à partir desquels il est impossible de dériver une paramétrisation valide (Figure 2). En fait, les graphes de singularités valides sont extrêmement contraints : une courbe de singularité valide suit obligatoirement une des directions du champ, et les deux autres directions définissent un champ de directions 2D avec la singularité qui se trouve sur la courbe. Cette contrainte peut être imposée a posteriori par des optimisations combinatoires [LLX$^+$12], mais cette stratégie n'est que locale. Nous voulons étudier une alternative qui intégrera les contraintes d'intégrabilité directement lors de la génération du champ.

Les singularités d'un *Frame Field* 2D sont simples à caractériser : un point est singulier si l'intégrale de la rotation du champ le long du bord d'un voisinage (infiniment petit) de ce point n'est pas nulle. Pour le remaillage quadrangulaire (petits carrés déformés), cela implique la présence d'un sommet irrégulier i.e. de valence différente de 4. Cette caractérisation ne s'étend malheureusement pas au cas volumique car l'intégration des rotations $3D$ est mal définie. Cependant, nous savons que les seules courbes de singularité qui correspondent à un remaillage hexaèdre sont des singularités 2D « extrudées » selon la troisième direction : elles peuvent devenir des arêtes partagées par un nombre d'hexaèdres différent de 4.

Conceptuellement, l'algorithme de génération de champs de directions propage à l'intérieur du domaine les contraintes définies sur la frontière. Ces contraintes imposent qu'un axe du champ de direction soit égal à la normale de la surface du domaine. Le graphe de singularités provient d'endroits où l'algorithme doit faire face à des contraintes contradictoires, mais la plupart d'entre elles n'ont que deux des trois axes contraints : la dernière direction reste bien définie.

Notre idée est de définir ce champ vectoriel « stable » dans le domaine, et de contraindre les deux autres directions pour éviter de tourner lorsqu'il est advecté par cette direction. Cette approche imitera l'opération de balayage effectuée dans la modélisation des maillages hexaédriques. Pour atteindre notre objectif, nous devons définir un champ de direction stable à l'intérieur du domaine, et lisser le champ de directions pour qu'il ne tourne pas lorsqu'il se déplace dans la direction stable.

Notre algorithme actuel d'optimisation des champs repose sur des harmoniques sphériques pour fusionner les contraintes à l'intérieur du domaine, et il sera plus facile de dériver un champ vectoriel localement stable à partir de cette représentation. Le défi sera de manipuler des objets complexes où cette direction stable n'est pas représentée par le même axe partout dans le domaine. Le simple fait de fixer la direction stable et de minimiser la courbure du champ ne garantit pas que les singularités s'aligneront sur la direction stable ; nous devons contraindre le champ à avoir une rotation minimale lorsque nous nous déplaçons dans la direction stable. Nous essaierons de l'appliquer de deux façons : soit dans le solveur par un terme d'anisotropie forte ou des multiplicateurs de Lagrange, ou bien en manipulant un nouvel ensemble de variables qui satisfont naturellement la contrainte. Ce travail s'inscrit dans la continuité de [RSL16], qui est une référence qui présente les algorithmes que nous utilisons actuellement pour générer les *frame fields* 2D et 3D.

Compétences espérées : la principale qualité attendue est l'envie d'apprendre et de travailler en équipe. Une bonne intuition géométrique en $3D$ sera un plus. Être à l'aise en programmation sera nécessaire pour pouvoir tester différentes idées. Notre base de code est en C++, mais les difficultés seront principalement d'ordre algorithmiques (ni syntaxiques, ni architecturales).

Références

[LLX+12] Yufei Li, Yang Liu, Weiwei Xu, Wenping Wang, and Baining Guo. All-hex meshing using singularity-restricted field. *ACM Trans. Graph.*, 31(6) :177 :1–177 :11, November 2012.

[RSL16] Nicolas Ray, Dmitry Sokolov, and Bruno Levy. Practical frame-field generation. *Transactions on Graphics (SIGGRAPH Asia conf. proc)*, 35(6) :233 :1–233 :9, November 2016.

[SRUL16] Dmitry Sokolov, Nicolas Ray, Lionel Untereiner, and Bruno Lévy. Hexahedral-dominant meshing. *ACM Transactions on Graphics*, 36(4), June 2016.