Projet de Fin d'Étude

14 septembre 2009

Mémoire de Master Informatique de l'Université de Bordeaux 1
Rapport de Projet de Fin d'Études de l'ENSEIRB

Equilibrage de charge et distribution de données pour des simulations parallèles irrégulières couplées : application à la propagation multi-échelle de fissures dans les matériaux.

Maîtres de stage :
Olivier COULAUD
Aurélien ESNARD
Jean ROMAN

Encadrant pédagogique :
François PELLEGRINI

Période de stage :

du 09 février 2009 au 30 septembre 2009

Mohamed Amine EL AFRIT
www.melafrit.com
postmaster@melafrit.com

Remerciements

Avant toute chose, je tiens à remercier toutes les personnes qui, d'une façon ou d'une autre, ont fait que ce stage au sein de l'INRIA ait pu se réaliser.

Je tiens à remercier *Jean Roman* et *Olivier Coulaud* de m'avoir accueilli dans leur équipe et de me donner l'occasion de travailler au sein d'une équipe de recherche dans l'une des plus grandes institut de recherche françaises.

Un grand merci pour *Aurélien Esnard* qui m'a aidé pendant la phase de modélisation et de développement et qui n'a pas hésité pour répondre à mes questions et à me donner des conseils et m'apporter des corrections sur ce rapport.

Merci à *François Pellegrini*, mon encadrant pédagogique, de m'avoir suivi et de m'avoir donner des conseils.

Merci à *Nicolas Richart* de m'avoir donner de son temps pour m'aider à résoudre quelques problèmes techniques et je lui souhaite une bonne chance pour la suite de sa thèse.

Merci encore à *Olivier* et à *Guillaume Anciaux* pour leur soutien qui m'a permi de comprendre plusieurs aspects du domaine du couplage et de la simulation multiéchelle avant de rentrer dans le sujet.

Je remercie également tout les membres de l'équipe qui m'ont accueilli chaleureusement pendant ces 8 mois de stage et qui on créé une ambiance très agréable.

Résumé

On souhaite étudier certains phénomènes dans les matériaux à travers des simulations multi-physiques. Il existe deux grandes familles de modélisation des materiaux, selon la propriété que l'on désire observer et le degré de précision voulu. On choisit parmi les méthodes de simulation en fonction de l'échelle à laquelle on veut observer le matériau. Il arrive que l'on a besoin de faire une simulation dans deux échelles différentes, car chaque type de modélisation possède des avantages et des inconvénients et on souhaite bénéficier en même temps des avantages de chaque modèle.
Une bibliothèque a été réalisée dans le cadre du travail de thèse de *Guillaume Anciaux* [8] pour coupler des codes en parallèles dédiés à la dynamique moléculaire et à l'élasticité dynamique. Un découpage des données est réalisé afin de les distribuer sur les processeurs disponibles. Le découpage des données de chacun des codes considérés séparément sont réalisés pour obtenir le meilleur équilibrage de charge, mais une fois couplées, ces décompositions de domaines induisent couramment un mauvais équilibrage global et cela notamment pour les processeurs ayant en charge la zone de couplage. L'une des perspectives de la thèse [8] était de trouver un moyen pour l'équilibrage de la charge. Cela est en partie l'objet de ce travail.
Dans le cadre de ce projet, on va proposer des solutions génériques basées sur des modélisations du problème à base d'hypergraphes pour mieux équilibrer la charge de calcul sur les processeurs.

Mots clefs : , parallélisme, couplage de code, partitionnement de graphe et d'hypergraphe, redistribution de données

Table des matières

Remerciements ii

Table des figures **1**

I Présentation de l'entreprise **3**

Chapitre 1 Présentation de l'entreprise **5**

1.1 L'*INRIA* en bref 5
1.2 Quelques chiffres 6
1.3 INRIA Bordeaux - Sud Ouest 6

II Contexte et existant **9**

Chapitre 2 Contexte et description de l'existant **11**

2.1 Présentation du projet et motivations 11
2.2 Description de l'existant : LibMultiScale 12
2.2.1 La dynamique moléculaire 13
2.2.2 La mécanique des milieux continus 14
2.2.3 Le couplage 16

Chapitre 3 Cadre général : rappel sur les graphes et le problème de partitionnement **21**

3.1 Rappel de la théorie des graphes et des hypergraphes 21
3.1.1 Les graphes 21
3.1.2 Les Hypergraphes 22
3.1.3 Avantage des hypergraphes 23
3.2 Le problème de partitionnement de graphes et d'hypergraphes 24
3.2.1 Explication du problème 24
3.2.2 Outils de partitionnement 25
3.2.3 Exemple de partitionnement 25

III Modelisation et réalisations : module de partitionnement et d'équilibrage de charge **27**

Chapitre 4 Problèmatique et modélisation **29**

4.1 Description de la problèmatique 30
4.2 Cas de test de référence 31
4.3 Dépendances des calculs 33
4.3.1 Formules préliminaires pour le calcul des coûts 33

4.3.2 Dépendence de calcul entre les éléments . . . 36
4.4 Modélisation avec un hypergraphe . . . 39
4.4.1 Modélisation du maillage avec un hypergraphe . . . 39
4.4.2 Modélisation du couplage . . . 41
4.5 Calcul des coûts . . . 42
4.5.1 Coût de calcul des éléments du maillage . . . 42
4.5.2 Coût des hyper-arêtes de l'hypergraphe . . . 43
4.5.3 Etude d'un exemple . . . 44
4.6 Différentes stratégies de partitionnement étudiées . . . 45
4.7 Optimisation . . . 47

Chapitre 5 Description de notre module d'équilibrage de charge **49**

5.1 Description de l'environnement d'exécution du module . . . 49
5.2 Structures de données utilisées . . . 51
5.2.1 Structure **maillage** . . . 51
5.2.2 Structure **graphe** . . . 52
5.2.3 Structure **hypergraphe** . . . 52
5.3 Algorithme principale du module d'équilibrage de charge . . . 53
5.4 Description des autres sous modules . . . 54
5.4.1 Chargement du maillage en mémoire . . . 54
5.4.2 Construction du graphe à partir du maillage . . . 54
5.4.3 Construction de l'hypergraphe à partir du graphe . . . 54
5.4.4 Sauvegarde du maillage . . . 54
5.4.5 Calcul des statistiques et génération d'un fichier de diagnostique . . . 55

IV Résultats et Perspectives **57**

Chapitre 6 Résultats **59**

6.1 Cas d'une géometrie simple . . . 61
6.1.1 Description du maillage . . . 61
6.1.2 Partitionnement classique du maillage en le modélisant par un graphe . . . 61
6.1.3 Partitionnement du maillage en le modélisant par un hypergraphe classique . . . 63
6.1.4 Partitionnement du maillage en le modélisant par un hypergraphe avec des hyper-arêtes de couplage . . . 66
6.1.5 Partitionnement du maillage en le modélisant par un hypergraphe avec technique des points fixes . . . 68
6.1.6 Partitionnement du maillage en le modélisant par un hypergraphe avec la technique de multi-contraintes . . . 70
6.2 Cas d'une géometrie plus complexe . . . 71
6.2.1 Description du maillage . . . 71

6.2.2 Partitionnement classique du maillage en le modélisant par un graphe 71
6.2.3 Partitionnement du maillage en le modélisant par un hypergraphe classique 74
6.2.4 Partitionnement du maillage en le modélisant par un hypergraphe avec des hyper-arêtes de couplage 77
6.2.5 Partitionnement du maillage en le modélisant par un hypergraphe avec technique des points fixes 80
6.2.6 Partitionnement du maillage en le modélisant par un hypergraphe avec la technique de multi-contraintes 83
6.3 Mise en évidence des intérêts et limitations des hyper-arêtes de couplage 83
6.4 Conclusion 84

Chapitre 7 Conclusion générale et perspectives **87**

Glossaire **89**

V Annexes **91**

Annexe A Utilisation de GMSH **93**

A.1 Utilisation de *GMSH* 93
A.2 Exemples 93

Annexe B Préparation d'un cas de test pour LibMultiScale **97**

B.1 Configuration pour *Lammps* 98
B.2 Configuration pour *SimulPack* 98

Annexe C Exemple de description d'une géometrie simple avec Gmsh **99**

Annexe D Exemple de description d'une géometrie complexe avec Gmsh **101**

D.1 Exemple d'une zone de recouvrement rectangulaire arrondie 101
D.2 Exemple d'une zone de recouvrement rectangulaire avec demi-cercles 104

Annexe E Exemple de fichier de diagnostique généré par notre module **107**

Annexe F Comportement de notre module d'équilibrage de charge en fonction des coûts des arêtes/hyper-arêtes et des poids des sommets du graphe/hypergraphe **111**

F.1 Exemple de scripts pour lancer un diagnostique 111
F.1.1 Enregistrement des mesures 111
F.1.2 Fichier généré 112
F.1.3 Script pour tracer les courbes avec Gnuplot 112
F.2 Résultat des diagnostiques 113

F.2.1 Cas d'une modélisation avec un graphe 113
F.2.2 Cas d'une modélisation avec un hypergraphe 114

Bibliographie **122**

Table des figures

2.1 Déscription du materiau à deux échelles avec une zone de recouvrement [8]. 13
2.2 Description lagrangienne du mouvement d'un milieu continu [8]. 14
2.3 Déformation du cristal pour l'approximation de Cauchy-Born. 16
2.4 Diagramme en tâches du déroulement d'une simulation couplant deux modèles. 17
2.5 Graphe de tâches d'un pas de temps avec les complexités associées [8]. 18

3.1 Exemple de représentation d'un graphe. 22
3.2 Exemple de représentation d'un hypergraphe [4]. 23
3.3 Modélisation de la communication par un graphe ou un hypergraphe [17, 20, 5, 19]. 23
3.4 Communication d'un élément sur la frontière de la coupe avec ses voisins [17]. 24
3.5 partitionnement multi-niveaux [12]. 25
3.6 Exemple de partitionnement d'un graphe dans le cas de maillages [17, 20, 5]. 26

4.1 Décomposition par macro-boîte du domaine utilisé par la plupart des codes de dynamique moléculaire [8]. 30
4.2 Décomposition de domaine par bipartitionnement récursif d'un maillage non structuré. [8] . 30
4.3 Un cas de propagation de fissure [8]. 31
4.4 Distribution de la charge dans les cas de 16/8 et 20/8 processeurs sont affectés à la dynamique moléculaire(en bas)/mécanique continue(en haut). [8] 32
4.5 Comparaison entre les charges de calcul d'un processeur A ayant en charge une partie de la zone de recouvrement Ω^R (partie hachurée en haut à gauche de la figure (a)) et d'un processeur B ayant en charge une partie de la zone normale Ω^N (partie hachurée en bas à droite de la figure (a)). 33
4.6 Dépendence d'un élément avec ses voisins. 37
4.7 Dépendence de calcul dans la tâche d'integration pour les éléments finis. 38
4.8 Déformation du crystal pour l'approximation de cauchy-born [8]. 39
4.9 Modélisation en hypergraphe. [17] 40
4.10 Influence de l'augmentation de la connectivité sur les résultats de partitionnement. 40
4.11 Construction des hyper-arêtes de couplage. 41
4.12 Exemple de calcul de coût. 45
4.13 Les différentes stratégies de partitionnement. 46
4.14 Partitionnement à la main pour garantir un résultat optimal dans le cas où le nombre de processeurs de DM qui participent au couplage correspond au nombre de processeurs de la mécanique continu. 47

5.1 Description de l'environnement d'exécution de notre module. 50
5.2 Les tableau *eptr* et *eind* qui décrivent les hyper-arêtes de l'hypergraphe [1]. 53
5.3 Construction de la liste des cellules par noeud (structure duale). 55

6.1 Étiquettes pour distinguer les différentes stratégies. 60

6.2 Résultat de partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe en utilisant la technique de multi-contraintes (PHg_MC). 70
6.3 Partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe en utilisant la technique de multi-contraintes (PHg_MC). 83
6.4 Mise en évidence du rôle des hyper-arêtes de couplage pour un meilleur partitionnement : Partitionnement d'un maillage de géometrie simple avec 2 processeurs DM et 3 processeurs EF. 84
6.5 Limitations des hyper-arêtes de couplage : partitionnement d'un maillage ayant une zone de recouvrement carrée (cas d'un coût uniforme avec la technique des points fixes). 85

A.1 Maillage obtenu avec gmsh. [16] 94

B.1 Dépendance des fichiers pour construire un cas de test (argon 2D) pour LibMultiscale. . . . 97

C.1 Maillage simple généré avec gmsh 99

D.1 Maillage simple généré avec gmsh 101
D.2 Paramètres de configuration de la zone de couplage pour une géometrie complexe. 102
D.3 Maillage simple généré avec gmsh 104

Liste des tableaux

4.1 Outils pour le calcul des coûts. 36

4.2 Coût de calcul des éléments (sommets de l'hypergraphe). 42

4.3 Coût des hyper-arêtes. 44

6.1 Diagnostique du partitionnement classique du maillage d'une géometrie simple en le modélisant par un graphe. 61

6.2 Résultats de partitionnement classique du maillage d'une géometrie simple en le modélisant par un graphe. 63

6.3 Diagnostique du partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe. 63

6.4 Résultats de partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe. 65

6.5 Diagnostique du partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe et en rajoutant des hyper-arêtes des couplage. 66

6.6 Résultats de partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe et en rajoutant des hyper-arêtes des couplage. 67

6.7 Diagnostique du partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe en utilisant la technique des points fixes. 68

6.8 Résultats de partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe en utilisant la technique des points fixes. 69

6.9 Diagnostique du partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe en utilisant la technique de multi-contraintes (PHg_MC). 70

6.10 Diagnostique du partitionnement classique du maillage d'une géometrie complexe en le modélisant par un graphe. 71

6.11 Partitionnement classique du maillage d'une géometrie complexe en le modélisant par un graphe. 73

6.12 Diagnostique du partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe. 74

6.13 Partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe. 76

6.14 Diagnostique du partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe et en rajoutant des hyper-arêtes des couplage. 77

6.15 Partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe et en rajoutant des hyper-arêtes des couplage. 79

6.16 Diagnostique du partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe en utilisant la technique des points fixes. 80

6.17 Partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe en utilisant la technique des points fixes. 82

6.18 Diagnostique du partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe en utilisant la technique de multi-contraintes (PHg_MC). 83

F.1 Influence de la variation du poids des sommets et du cout des arêtes de la zone de recouvrement du graphe sur l'équilibrage de charge et les communications. 114

F.2 Influence de la variation du poids des sommets et du cout des hyper-arêtes normales et de couplage de la zone de recouvrement de l'hyper-graphe sur l'équilibrage de charge et les communications. 116

F.3 Influence de la variation du poids des sommets et du cout des hyper-arêtes normales et de couplage de la zone de recouvrement de l'hyper-graphe sur l'équilibrage de charge et les communications (avec technique des points fixes). 119

Première partie

Présentation de l'entreprise

1

Présentation de l'entreprise

Ces informations sont tirées du site officiel de l'INRIA (http://www.inria.fr)

1.1 L'INRIA en bref

Créé en 1967 à Rocquencourt près de Paris, l'*INRIA* (Institut national de recherche en informatique et en automatique) est un établissement public à caractère scientifique et technologique (EPST) placé sous la double tutelle du ministère de la recherche et du ministère de l'économie, des finances et de l'industrie.

Les principales missions de l'*INRIA* sont décrites par le décret du 2 août 1985 portant sur l'organisation et le fonctionnement de l'institut.

L'*INRIA* a l'ambition d'être au plan mondial, un institut de recherche au coeur de la société de l'information.

L'*INRIA*, institut national de recherche en informatique et en automatique, placé sous la double tutelle des ministères de la recherche et de l'industrie, a pour vocation d'entreprendre des recherches fondamentales et appliquées dans les domaines des sciences et technologies de l'information et de la communication (STIC). L'institut assure également un fort transfert de technologie en accordant une grande attention à la formation par la recherche, à la diffusion de l'information scientifique et technique, au développement, à l'expertise et à la participation à des programmes internationaux.

Jouant un rôle fédérateur au sein de la communauté scientifique de son domaine et au contact des acteurs industriels, l'*INRIA* est un acteur majeur dans le développement des STIC en France. L'*INRIA* accueille 3 800 personnes réparties dans ses 8 centres de recherche situés à Rocquencourt, Rennes, Sophia Antipolis, Grenoble, Nancy, Bordeaux, Lille et Saclay. 2 800 d'entre elles sont des scientifiques de l'*INRIA* et d'organismes partenaires (CNRS, universités, grandes écoles) qui travaillent dans plus de 150 équipes-projets de recherche communes. Un grand nombre de chercheurs de l'*INRIA* sont également enseignants, et leurs étudiants (environ 1 000) préparent leur thèse dans le cadre des équipes-projets de recherche de l'*INRIA*.

L'*INRIA* développe de nombreux partenariats avec le monde industriel et favorise le transfert technologique et la création d'entreprises dans le domaine des STIC. Plus de 90 entreprises ont été créées grâce au soutien de sa filiale *INRIA*-Transfert, spécialisée dans l'accompagnement, l'évaluation, la qualification et le financement des jeunes entreprises innovantes de haute technologie informatique. L'*INRIA* est actif au sein d'instances de normalisation comme l'*IETF*, l'*ISO* ou le *W3C* dont il a été le pilote européen de 1995 à 2002.

Enfin, l'institut entretient d'importantes relations internationales : en Europe, l'*INRIA* est membre du consortium *ERCIM*, qui regroupe des instituts de recherche de 19 pays européens. L'*INRIA* participe à environ 120 actions dans le cadre du 6e *PCRD* et 40 actions dans le cadre du 7e *PCRD*, essentiellement dans le domaine des *STIC*. à l'international, l'institut collabore avec de nombreuses institutions scientifiques et universitaires (laboratoires de recherche conjoints tels que *LIAMA*, équipes de recherche associées, programmes de formation et de stages, etc.).

Le budget de l'*INRIA* s'élève à 186 millions d'euros, dont 20 % proviennent de ses propres contrats de

recherche et produits de valorisation.

La stratégie de l'institut repose sur la combinaison étroite de l'excellence scientifique et du transfert technologique. L'objectif essentiel de l'*INRIA* pour les années 2008-2012 est de réaliser des percées scientifiques et technologiques dans sept domaines prioritaires :

- Modélisation, simulation et optimisation de systèmes dynamiques complexes,
- Programmation : sécurité et fiabilité des systèmes informatiques,
- Communication, information et calcul ubiquitaires,
- Interaction avec des mondes réels ou virtuels,
- Ingénierie numérique,
- Sciences numériques,
- Médecine numérique.

1.2 Quelques chiffres

- Ressources budgétaires (janvier 2008),
  - budget total : 186 M Euros HT,
  - ressources propres : 1/5.

- Ressources humaines (janvier 2008),
  - 3 800 personnes, dont 2 100 rémunérées par l'*INRIA*,
  - 2 800 scientifiques, dont 1 300 chercheurs et enseignants-chercheurs, 1 000 doctorants, et 500 post doctorants et contractuels.

- Activités scientifiques (janvier 2008),
  - 150 équipes-projets de recherche,
  - 4 000 publications scientifiques,
  - 24 conférences internationales organisées ou co-organisées par l'*INRIA* ayant mobilisé 2 500 participants dont 1 700 étrangers,
  - 14 300 heures d'enseignement.

- Relations industrielles (janvier 2008).
  - 790 contrats de recherche actifs,
  - 230 brevets actifs,
  - 80 logiciels déposés à l'Agence pour la Protection des Programmes,
  - 89 sociétés de technologie issues de l'*INRIA*, depuis Ilog, aujourd'hui cotée au Nasdaq, jusqu'aux 9 dernières créées en 2007.

1.3 INRIA Bordeaux - Sud Ouest

La création en janvier 2008 du centre de recherche INRIA Bordeaux - Sud Ouest est basée sur les équipes-projets et services créés à Bordeaux et à Pau dans le cadre de l'unité de recherche Futurs, avec le soutien fort du Conseil Régional d'Aquitaine.

Un centre de recherche ouvert et dynamique Les équipes de recherche y ont été construites en s'appuyant sur des partenariats forts avec les universités et grandes écoles des sites concernés et en collaboration

avec leurs laboratoires réputés.

Grâce au dynamisme de ces collaborations, à l'apport de personnels ayant effectué une mobilité depuis d'autres sites de l'INRIA et en s'appuyant sur une politique de recrutement de chercheurs et de personnels de soutien à la recherche de haute qualité, le centre de recherche emploie directement, début 2008, plus de 120 personnes. Il compte 13 équipes-projet de recherche, et 2 équipes dont l'évaluation est en cours et qui devraient devenir prochainement équipes-projets. Le centre rassemble globalement plus de 250 personnes.

Dans le cadre de son plan stratégique, l'INRIA entend définir son développement à Bordeaux - Sud Ouest en liaison forte avec le pole de compétitivité Aerospace Valley. Des partenaires industriels et académiques seront également étroitement associés. On peut citer par exemple Total, Safran/Turbomeca, Thales, Rhodia, le CEA (en particulier dans le cadre du programme de recherche autour du Laser Mégajoule), France Telecom, EDF, Airbus, SNCF, etc. directement, mais aussi au travers du pôle Aerospace Valley.

Les capacités de développement du centre et les partenariats engagés permettent d'envisager une croissance d'au moins 50

Ce site web vous permettra de mieux connaître les activités de recherche scientifique et de développement de l'INRIA Bordeaux - Sud Ouest ainsi que celles touchant au transfert technologique, l'autre objectif stratégique de l'INRIA. Etre à l'écoute de l'industrie, c'est favoriser une dynamique fructueuse entre questions scientifiques et résolution de problèmes du monde socio-économique.

Deuxième partie

Contexte et existant

2

Contexte et description de l'existant

Sommaire

2.1 Présentation du projet et motivations **11**
2.2 Description de l'existant : LibMultiScale **12**
2.2.1 La dynamique moléculaire 13
Formulation du problème 13
Intégration en temps : schéma d'intégration 13
2.2.2 La mécanique des milieux continus 14
Formulation du problème 14
Intégration en temps : schéma d'intégration 15
2.2.3 Le couplage 16
Système d'équations couplées 16
Description algorithmique du coupleur 16

Résumé : Dans ce chapitre nous allons décrire le code *LibMultiScale* qui permet de coupler deux codes simulant la propagation d'une fissure dans un matériau à deux échelles différentes : un code pour la dynamique moléculaire à l'échelle microscopique, au voisinage de la fissure, et un code pour l'élasticité dynamique (mécanique des milieux continus) à une échelle macroscopique. Nous rappellerons également les algorithmes d'intégration en temps utilisés pour ces deux codes ainsi que la tâche d'intégration rajoutée par le coupleur qui fait la correction des vitesses pour le couplage des deux codes.

2.1 Présentation du projet et motivations

La simulation du comportement physico-chimique des matériaux, comme le suivi de la propagation de fissures ou encore l'étude de propriétés macroscopiques de matériaux hybrides, nécessite de modéliser les phénomènes à des échelles d'espace et de temps très différentes. Dans ces simulations, on couple une échelle macroscopique à une échelle microscopique, celle-ci peut être une échelle atomique voire électronique dans certains couplages. Ce couplage " **micro-macro** " est indispensable pour obtenir une bonne description de la matière. La modélisation par une échelle plus grossière (macroscopique) permet de modéliser essentiellement des échantillons de tailles plus importantes. Ces nouvelles simulations numériques sont développées en couplant différents codes parallèles représentant chacun différents modèles physiques.

Si l'on considère la propagation de fissures par exemple (cas des lentilles de focalisation du Laser Méga-Joule), la méthode multi-échelle qui a été développée suppose que deux modèles soient couplés finement : un modèle atomique (en dynamique moléculaire) et un modèle continu (approximation par éléments finis). La transmission de l'information entre ces deux modèles se fait par une zone de recouvrement où les deux modèles vont cohabiter.

Dans le cadre d'un couplage en parallèle, un découpage des données est réalisé afin de les distribuer sur les processeurs disponibles. Ce découpage a une très grande influence sur la performance du coupleur. Les découpages des données de chacun des codes considérés séparément sont réalisés pour obtenir un meilleur

équilibrage de charge respectif, mais une fois couplés, ces décompositions de domaines induisent couramment un mauvais équilibrage global et cela notamment pour les processeurs ayant en charge la zone du couplage. En effet, **il faut en fait maximiser le nombre de processeurs en charge du couplage pour diminuer le temps de calcul de celui-ci, tout en minimisant le nombre de domaines connectés entre les modèles pour diminuer les communications entre les codes parallèles sous-jacents.**[1] Un enjeu important dans ces simulations multi-physiques mais aussi dans toute simulation impliquant un couplage de codes complexes (météorologie, climatologie, ...) est de conserver le caractère **extensible** de l'exécution parallèle globale des codes couplés.

L'objectif de ce projet est d'améliorer le parallélisme du coupleur développé dans la thèse de *G.Anciaux* [8] dans le cadre de la propagation de fissures et à l'aide de la bibliothèque **LibMultiScale** [7], en étudiant différentes stratégies d'équilibrage de charge.

D'un point de vue logiciel, nous utiliserons le logiciel de partitionnement de graphes **METIS** [2] ou d'hypergraphes **HMETIS** [1] et **PATOH** [20], pour analyser, implémenter et tester les différentes stratégies.

Le plan de ce rapport est le suivant : dans la partie II on va décrire la bibliothèque *LibMultiScale* qui a été développée avant ce travail [8]. Ensuite on présentera dans le chapitre 3 le cadre général en se basant sur la théorie des graphes et sur les articles de référence que nous avons trouvé concernant l'équilibrage de charge à l'aide des outils de partitionnement. On décrira le problème de partitionnement dans son cadre général et particulier et nous verrons les outils qui existent et comment ils permettent de résoudre notre problème d'équilibrage de charge. On décrira par la suite notre contribution en terme de modélisation et réalisations dans la partie III. Finalement nous présentons et analysons les résultats que nous avons obtenus dans le chapitre 6 et pour terminer, nous concluons et donnons des perspectives dans le chapitre 7.

2.2 Description de l'existant : LibMultiScale

On souhaite étudier certains phénomènes dans les materiaux comme le phénomène de propagation de fissure. Il existe deux grandes familles de modélisation des materiaux selon la propriété que l'on désire étudier et le degré de précision voulu. La principale distinction entre ces deux types de modélisation porte principalement sur l'échelle à laquelle on modélise le materiau :

- Échelle microscopique pour décrire la matière de manière très précise. On parle de la *Dynamique moléculaire*.
- Échelle macroscopique pour le traitement d'échantillons de taille réelle et là il s'agit de *la mécanique des milieux continus*

Sachant que chaque type de modélisation possède des avantages et des inconvénients et qu'on souhaite profiter en même temps des avantages de chaque modèle, nous devons **coupler** les deux modèles.

Il y aura alors une zone qui décrira le materiau à l'échelle macroscopique et une zone qui décrira à une échelle beaucoup plus fine (microscopique), comme dans la figure 2.1. On trouve entre les deux zones une zone où les deux modèles sont valides, c'est la *zone de recouvrement* qu'on appelle aussi *bridging zone*.

[1] On appelle *sous-jacents*, les codes dédiés à la dynamique moléculaire et la mécanique des milieux continus

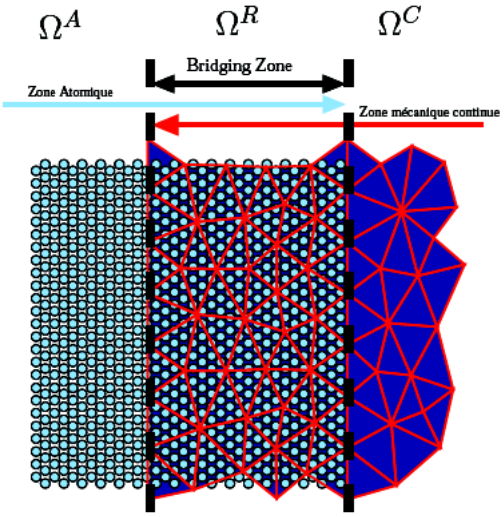


Fig. 2.1: Déscription du materiau à deux échelles avec une zone de recouvrement [8].

Dans cette section nous présentons brièvement les différents modèles et algorithmes de simulation pour chacun des modèles de description des matériaux (microscopique et macroscopique). La majorité des informations de cette section sont issues de la thèse de G.Anciaux [8].

2.2.1 La dynamique moléculaire

La dynamique moléculaire[2] consiste à étudier la trajectoire d'une molécule en appliquant les lois de la mécanique classique, et a pour but de simuler le mouvement des atomes au cours du temps, à partir de l'énergie potentielle calculée par la mécanique moléculaire.

Formulation du problème

Le mouvement d'un système de m atomes est régi par l'équation de Newton classique 2.1 :

$$M\frac{d^2X_i(t)}{dt^2} = f_i^{int} + f_i^{ext} \tag{2.1}$$

$$avec \quad f_i^{int} = \frac{\partial W(r_i)}{\partial d} = -(\nabla_d W) \tag{2.2}$$

où M est la matrice des masses, $X_i(t)$ est la position à l'instant t, f_i^{int} et f_i^{ext} son respectivement les forces interieurs et exterieurs appliquées sur l'atome i, W est l'énergie potentielle interne, qui est une fonction donnée. L'évolution d'un atome i en formulation Lagrangienne est donné par équation :

$$m_i\ddot{d}_i = -f_i^{int} + f_i^{ext} \tag{2.3}$$

où d_i est le déplacement de l'atome i à l'instant t définit par : $X_i(t) = X_0^i + d_i$, X_0 est le vecteur des positions atomiques initiales.

Intégration en temps : schéma d'intégration

La discrétisation en temps de 2.2.1 est réalisée par le schéma de *Velocity Verlet* [6]. La tâche d'intégration en temps, T_S^A, pour le modèle atomique est :

[2] cf. Glossaire

$$T_S^A \begin{cases} \dot{d}_i^{n+\frac{1}{2}} & = & \dot{d}_i^n + \frac{\triangle t}{2m_i} f_i^n \\ d_i^{n+1} & = & d_i^n + \triangle t \dot{d}_i^{n+\frac{1}{2}} \\ calcul\ de\ f_i^{n+1} & & \\ \dot{d}_i^{n+1} & = & \dot{d}_i^{n+\frac{1}{2}} + \frac{\triangle t}{2m_i} f_i^{n+1} \end{cases}$$

Selon cet algorithme d'integration, on va calculer les déplacements d_i d'un atome i à chaque pas de temps n et les vitesses $\dot{d}_i$ à chaque demi pas de temps $n + \frac{1}{2}$.
Le calcul de la force fait intervenir en général plusieurs potentiels qui sont dûs aux interactions intermoléculaires et intramoléculaires[3]. Ici pour l'évaluation de la force, on se limite au potentiel intermoléculaire de Lennard-Jones[4] qui est donné par :

$$V_{lj}(r) = 4\epsilon[(\frac{r_0}{r})^{12} - (\frac{r_0}{r})^6] \tag{2.4}$$

où r est la distance entre les deux noyaux atomiques et r_0 et ϵ sont des constantes. Les valeurs typiques pour l'argon sont : $r_0 = 3,35A$ et $\epsilon = 0,01eV$ [4].

2.2.2 La mécanique des milieux continus

Formulation du problème

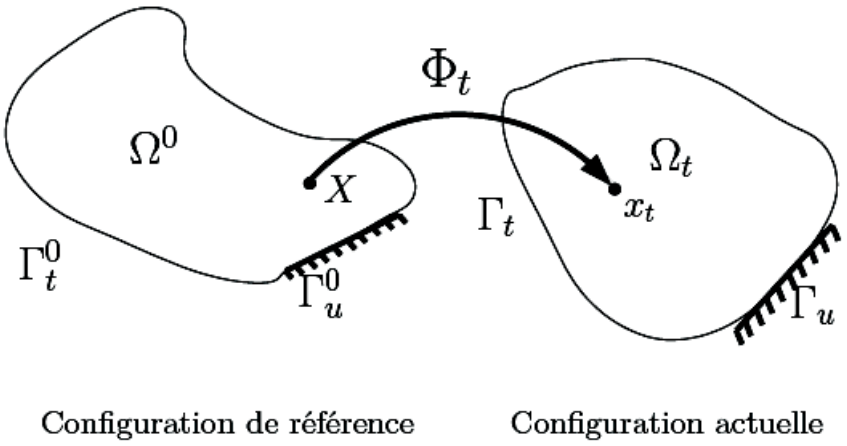


Fig. 2.2: Description lagrangienne du mouvement d'un milieu continu [8].

Dans le cadre de la mécanique des milieux continus[5] , on souhaite modéliser un solide $\mathcal{S}$ dont on se donne le contour Γ (comme dans la figure 2.2). Le solide est alors défini par la matière contenue dans l'espace délimité par ce contour. La matière est supposée avoir des propriétés continues dans cet espace. Chaque volume infinitésimal du solide est déplacé dans le temps par les contraintes propres au matériau étudié. On note Ω l'ensemble des points du solide $\mathcal{S}$.

On se donne les équations de la conservation des moments et de la masse.

$$\begin{cases} \rho\frac{\partial^2 u}{\partial t^2} - div\sigma & = & f & dans\ \Omega \\ \sigma.n & = & t & sur\ \Gamma_t \\ u & = & 0 & sur\ \Gamma_u \end{cases}$$

[3] Il s'agit des potentiel angulaire, de liaison et de torsion pour les interaction intramoléculaires, et des potentiels de *Buckingham* et de *Lennard-Jones* pour les interactions intermoléculaires
[4] cf. Glossaire
[5] cf. Glossaire

où Γ_t et Γ_u sont respectivement la partie du contour de Ω sur lequel s'exerce les efforts et la partie du contour fixe de Ω, u est le champ de déplacement lagrangien, ρ représente la masse volumique, σ est le tenseur des contraintes de Cauchy, f est la densité volumique des efforts externes et t est la densité surfacique des efforts externes. On se propose de résoudre ces équations. On multiplie l'équation de conservation du système précédent par une fonction test w et on intègre sur tout le domaine Ω :

$$\int \rho \frac{\partial^2 u}{\partial t^2}.wd\Omega - \int div\sigma.wd\Omega = \int f.wd\Omega \tag{2.5}$$

On discrétise le domaine Ω_0 par une triangulation $\mathcal{M}$ constituée par un ensemble $\mathcal{N}$ de noeuds et un ensemble $\mathcal{E}$ d'éléments qui interconnectent les précédents noeuds [8, 25, 30, 32, 33]. Le principe des éléments finis[6] [33, 32, 29, 31] utilise le maillage pour approcher de manière discrète tout champ f de l'espace. On utilise des fonctions d'interpolations associées à chaque noeud. Généralement on utilise des fonctions polynomiales comme fonctions d'interpolation. Pour des polynômes de degré 1, on parle d'éléments finis P_1. Ces fonctions forment une base de l'espace vectoriel $\mathcal{F}^{P_1}$ qui décrit un ensemble de fonctions linéaires à l'intérieur de chaque élément. Toute fonction de l'espace sera alors approximée par une fonction dans $\mathcal{F}$ par :

$$f(x) = \sum_I \varphi_I(x) f_I \tag{2.6}$$

Où f_I est la valeur du champ interpolé sur le noeud I et φ_I la fonction de base associée au noeud I. La formulation faible (Lagrangienne) des équations de la mécanique des milieux continus [8] nous amène à un système newtonien qu'il faut discrétiser en temps. Comme il faut coupler ce modèle avec celui de la dynamique moléculaire, on souhaite avoir le même formalisme et pour cela on définit un Hamiltonien particulier (cf. [8]).

Intégration en temps : schéma d'intégration

On utilise le même schéma de discrétisation en temps que pour la dynamique moléculaire :

$$T_S^C \begin{cases} \dot{u}_I^{n+\frac{1}{2}} & = & \dot{u}_I^n + \frac{\triangle t}{2M_I} F_I^n \\ u_I^{n+1} & = & u_I^n + \triangle t \dot{u}_I^{n+\frac{1}{2}} \\ calcul\ de\ F_I^{n+1} & & \\ \dot{u}_I^{n+1} & = & \dot{u}_I^{n+\frac{1}{2}} + \frac{\triangle t}{2M_I} F_I^{n+1} \end{cases}$$

La force F_I doit etre calculée sur chacun des noeuds du maillage des éléments finis pour mettre à jour la vitesse de déplacement. Pour chaque élément on évalue la force sur ses 3 noeuds. Chaque élément contribue d'une partie pour le calcul de chacune des forces de ses noeuds. Pour déterminer la contribution d'un élement K dans le calcul de la force F_I sur un de ses noeud I il faut évaluer le tenseur des contraintes[7] σ_K de l'élément K calculé au centre de gravité du triangle (élément) K. Chaque élément dépendra des éléments voisins qui l'entourent (voir figure 4.6). La force F aura donc la forme suivante :

$$F_i = \sum_K F_{iK} \tag{2.7}$$

Où i est l'un des noeuds de K et F_{iK} est la contribution de l'élement K pour le calcul de la force F_i sur le noeud i.

Le *tenseur de contrainte* d'un élement quelconque (exemple l'élement ABC dans la figure 4.6) est calculé, avec les conditions de l'approximation de *Cauchy-Born* [8], sur une boule de rayon R_{cut} dont le centre correspond au centre de gravité de l'élement en question après avoir subit la déformation $F = I + \bigtriangledown u$ (voir la figure 2.3). La complexité de cette opération est proportionnelle au nombre d'atomes présent dans la boule de rayon R_{cut}.

[6] cf. Glossaire
[7] cf. Glossaire

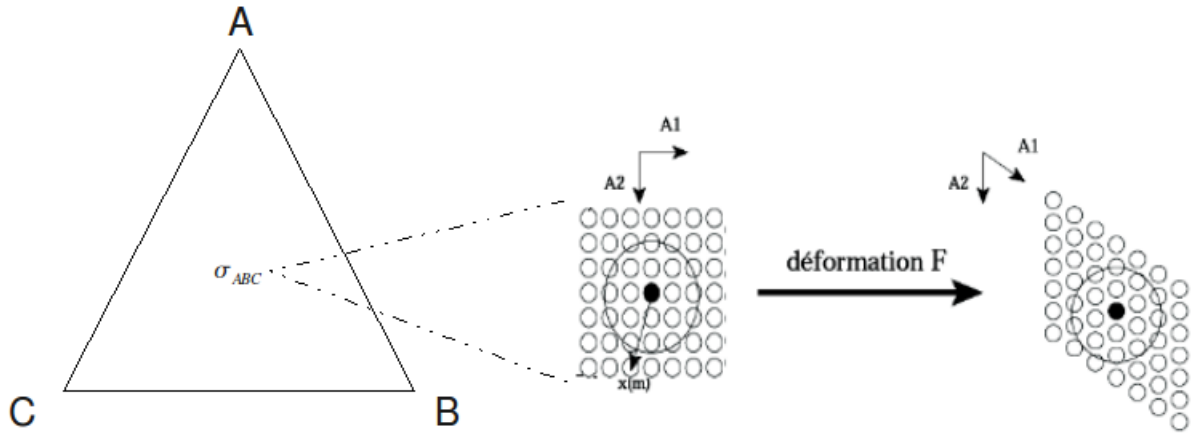


Fig. 2.3: Déformation du cristal pour l'approximation de Cauchy-Born.

2.2.3 Le couplage

Le couplage des deux modèles est réalisé en imposant des contraintes pour que les déplacements des atomes soient identiques entre les deux modèles.

Système d'équations couplées

Les contraintes appliquées dans la zone de recouvrement servent à recoller ensemble les deux modèles. Les équations de mouvement sont donc modifiées par des multiplicateurs de Lagrange. Le systéme d'équations final décrit dans [27] est :

$$\bar{M}_I \ddot{u}_I = -f_I^R + \sum_k \lambda_k \frac{\partial g_k}{\partial u_I} \tag{2.8}$$

$$\bar{m}_i \ddot{d}_i = f_i^R + \sum_k \lambda_k \frac{\partial g_k}{\partial d_i} \tag{2.9}$$

Avec $\bar{M}_I = \alpha(X_I)M_I$ et $\bar{m}_i = (1-\alpha(X_i))m_i$ sachant que M_I est la masse condensée sur le noeud I, m_i la masse de l'atome i et α est un facteur entre 0 et 1 qui est définie dans la zone de recouvrement par :

$$\alpha(X) = \frac{l(X)}{R} \tag{2.10}$$

Où $l(X)$ est la distance du bord de la zone mécanique continue et R étant la largeur de la zone de recouvrement. Ce système d'équation permet de décrire les corrections qu'il faut apporter au vitesse pendant la phase de couplage.

Description algorithmique du coupleur

Identification des tâches La figure 2.4 décrit le diagramme de tâches de la simulation.

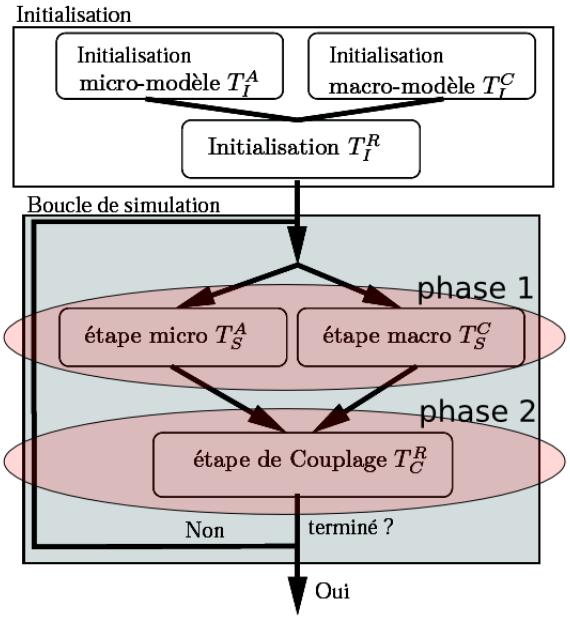


Fig. 2.4: Diagramme en tâches du déroulement d'une simulation couplant deux modèles.

On ne s'intéresse, pour l'instant, qu'aux tâches de la boucle principale d'intégration et aux tâches effectuées lors d'un pas de temps de simulation [8]. Sur le diagramme des tâches 2.4 on peut distinguer deux phases principales dans la boucle de simulation. Pendant la première phase les deux codes de la dynamique moléculaire et de la mécanique des milieux continus font leur calcul séparément. Cette première phase regroupe alors les deux étapes : *étape micro* T_S^A et *étape macro* T_S^C sur le diagramme 2.4. Le couplage se fait pendant la deuxième phase. C'est la phase de correction des vitesses calculés lors de la phase précédente. C'est *l'étape de couplage* T_C^R.

Sur le diagramme 2.5, on peut observer un détail que nous n'avons pas évoqué jusqu'à présent et qui est la synchronisation des codes à chaque pas de temps. En effet, les deux codes doivent se trouver dans une situation équivalente pour pouvoir être couplés. Ceci s'explique en analysant les deux étapes de calcul pendant la phase 1 et l'étape de calcul pendant la phase 2.
En effet, la valeur de la vitesse $\ddot{u}_I^{n+1}$, qui est calculée à la fin de l'étape de calcul dédiée aux éléments finis pendant la phase 1 de la boucle de simulation (selon la tâche d'integration T_S^C expliqué à la sous section 2.2.2), est nécessaire pour le calcul du second membre lors de la 2ème phase (à la sous étape (1b) de l'étape de couplage (selon la tâche d'integration T_C^R de cette section)).
Dans cette configuration des algorithmes d'integration, on doit attendre que tous les processeurs finissent leur calcul pour pouvoir passer d'une étape de calcul à une autre. C'est aussi un point que l'on étudiera pour voir la possibilité de désynchroniser les tâches.

[8] on ne s'intéresse pas aux tâches d'initialisation

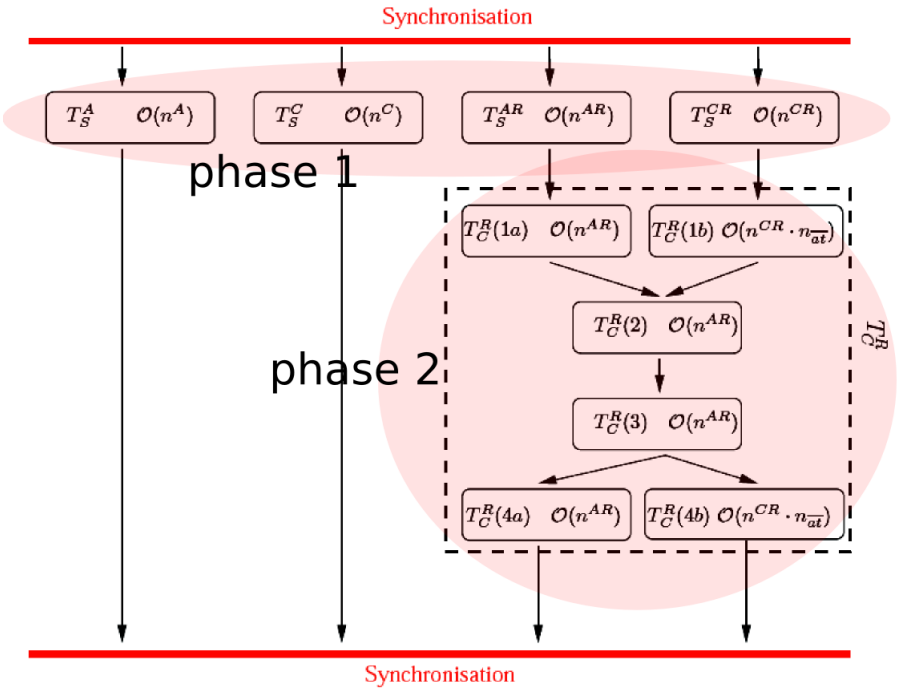


Fig. 2.5: Graphe de tâches d'un pas de temps avec les complexités associées [8].

On note n^A le nombre d'atomes non présents dans la zone de couplage, n^{AR} le nombre d'atomes qui sont contenus dans la zone de recouvrement, n^C le nombre de noeuds non présents dans le recouvrement et n^{CR} le nombre de noeuds présents dans le recouvrement. En suivant la même logique de notation, on distingue entre T_S^A, T_S^C et T_S^{AR}, T_S^{CR} où les tâches assignées d'un R supplémentaire désigne l'intégration par le schéma de *Verlet* des inconnues contenues dans la zone de recouvrement.

Tâche d'intégration Dans la phase 2 (figure 2.4) de couplage, notée T_C^R, on corrige les vitesses. Le terme de couplage est apporté par la contrainte de Lagrange introduite [8] de la manière suivante : pour chaque atome i présent dans la zone de recouvrement, la contrainte g_i est :

$$g_i = U_i - d_i \tag{2.11}$$

$$= \sum_j \varphi_j(X_i) u_j - d_i \tag{2.12}$$

Cette phase de couplage fait intervenir les deux codes via le calcul des multiplicateurs de Lagrange λ qui maintiennent la cohérence globale. On assure ainsi que la contrainte reste nulle.

Le schéma de correction T_C^R est :

$$T_C^R \begin{cases} (1a) & rhs_i^A = -\dot{d}_\alpha^{n+1} \\ (1b) & rhs_i^C = \sum_J \varphi_J(X_\alpha) \dot{u}_J^{n+1} \\ (2) & rhs_i = rhs_i^A + rhs_i^C \\ (3) & \lambda = A^{-1} rhs \\ (4a) & \dot{d}_\alpha^{n+1} \leftarrow \dot{d}_\alpha^{n+1} - \frac{\triangle t}{2 m_\alpha \alpha_\alpha} \lambda \\ (4b) & \dot{u}_I^{n+1} \leftarrow \dot{u}_I^{n+1} + \frac{\triangle t}{2 m_I \alpha_I} \sum_i \varphi_I(X_\alpha) \lambda_\alpha \end{cases} \qquad \begin{array}{l} A_{\alpha\beta} = \sum_J \frac{\varphi_J(X_\alpha)\varphi_J(X_\beta)}{M_J \alpha(X_I)} - \frac{\delta_{\beta\alpha}}{m_i(1-\alpha(X_\alpha))} \\ rhs_i = \sum_J \varphi_J(X_\alpha) \dot{u}_J - \dot{d}_\alpha \end{array}$$

Le schéma d'integration de la mécanique des milieux continus dans l'état actuel nécessite une synchronisation à chaque pas de temps (voir figure 2.5) pour pouvoir poursuivre les calcul car la mise à jour de

la vitesse lors de la dernière étape de la tâche d'integration dépend du calcul de la force F (voir schéma d'integration du domaine continu). On ne peut pas passer à l'étape de couplage avant la mise à jour de la vitesse car celle ci sera corrigée pendant cette étape. D'après les dépendances que nous avons exprimées dans la figure 4.7 on constate qu'il est possible de fusionner la dernière étape de mise à jour de la vitesse, lors de la tâche d'integration, avec l'étape de calcul de la force F car pendant ce calcul on va parcourir tout les éléments et l'idée est qu'on peut profiter de ce parcours pour mettre à jour au fure et à mesure la contribution de l'élément visité sur la valeur finale de F (formule 2.6). On reprendra cet aspect dans les perspectives.

3

Cadre général : rappel sur les graphes et le problème de partitionnement

Sommaire

3.1 Rappel de la théorie des graphes et des hypergraphes 21

3.1.1 Les graphes 21

Pésentation générale 21

Définition 22

3.1.2 Les Hypergraphes 22

Pésentation générale 22

Définition [14, 20, 4] 22

3.1.3 Avantage des hypergraphes 23

3.2 Le problème de partitionnement de graphes et d'hypergraphes 24

3.2.1 Explication du problème 24

3.2.2 Outils de partitionnement 25

3.2.3 Exemple de partitionnement 25

Résumé : Dans ce chapitre nous allons rappeler quelques notions sur les graphes et les hypergraphes et nous discuterons des avantages que peuvent apporter les hypergraphes par rapport aux graphes classiques en terme de modélisation. Nous allons introduire également le problème de partitionnement de graphes et d'hypergraphes et les outils qui sont à notre disposition pour le faire.

3.1 Rappel de la théorie des graphes et des hypergraphes

3.1.1 Les graphes

Pésentation générale

La théorie des graphes s'est développée dans diverses disciplines telles que la chimie, la biologie, les sciences sociales. Depuis le début du 20eme siècle, elle constitue une branche à part entière des mathématiques et de l'informatique.

De manière générale, un graphe permet de représenter la structure, les connexions d'un ensemble complexe en exprimant les relations entre ses éléments : réseau de communication, réseaux routiers, interaction de diverses espèces animales, circuits électriques, etc...

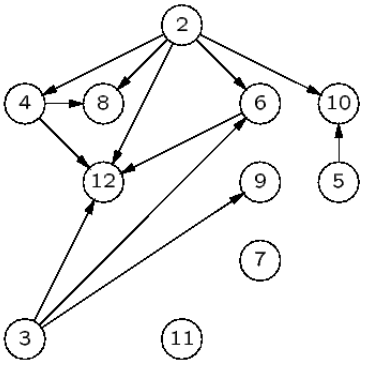


Fig. 3.1: Exemple de représentation d'un graphe.

Définition

Par définition, *un graphe est un ensemble de points, dont certaines paires sont directement reliées par un lien*. Ces liens peuvent être orientés, c'est-à-dire qu'un lien entre deux points u et v relie soit u vers v, soit v vers u : dans ce cas, le graphe est dit *orienté*. Sinon, les liens sont symétriques, et le graphe est *non-orienté*.

Plus formellement, un graphe simple G est un couple formé de deux ensembles : un ensemble $X = \{x_1; x_2; ...; x_n\}$ dont les éléments sont appelés sommets, et un ensemble $A = \{a_1; a_2; ...; a_m\}$, partie de l'ensemble $\mathcal{P}_2(X)$ des parties à deux éléments de X, dont les éléments sont appelés arêtes. On notera $G = (X; A)$. Lorsque $a = \{x; y\} \in A$, on dit que a est l'arête de G d'extrémités x et y. Les sommets x et y sont dits adjacents dans G.

3.1.2 Les Hypergraphes

Pésentation générale

Les hypergraphes[1] sont des objets mathématiques généralisant la notion de graphe dans le sens où les arêtes ne relient plus seulement un ou deux sommets, mais un nombre quelconque de sommets. De nombreux théorèmes de la théorie des graphes se généralisent naturellement aux hypergraphes. Les hypergraphes ont des applications dans tous les domaines où on utilise la théorie des graphes : résolution de problèmes en satisfaisant des contraintes, traitement d'images, optimisation d'architecture réseaux, modélisation, circuit électronique, etc...

Les hypergraphes sont récemment utilisés en calcul scientifique pour améliorer les distributions de calcul dans le domaine du parallĺisme par exemple pour mieux modéliser les communications [26, 17, 1, 20, 5, 21].

Définition [14, 20, 4]

Soient $V = \{v_1, v_2, \ldots, v_n\}$ un ensemble de sommets, $E = \{E_1, E_2, \ldots, E_m\}$ une famille de parties de V, avec $(m, n) \in \mathbb{N}_*^2$ une paire d'entiers non nuls.

Un hypergraphe H est un couple (V, E) tel que :
- $\forall i \in [1, m] \quad E_i \neq \varnothing$
- $\bigcup_{i=1}^{m} E_i \subseteq V$.

À l'instar des graphes, on dit que :
- Les éléments de V sont les sommets de H,

[1] cf. Glossaire

– Les éléments de E sont les arêtes de H, également appelées hyper-arêtes.

On peut décrire un hypergraphe par une matrice $\mathcal{A}_{n,m}$ à coefficients 0 ou 1 telle que :

$$\forall a_{i,j} \in \mathcal{A}, \quad a_{i,j} = \begin{cases} 1 & si\ v_i \in E_j \\ 0 & sinon \end{cases}$$

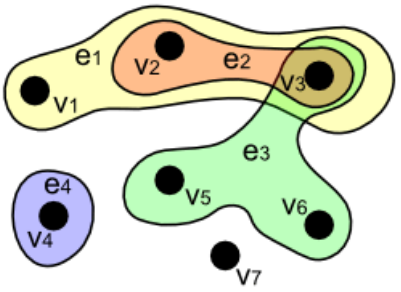
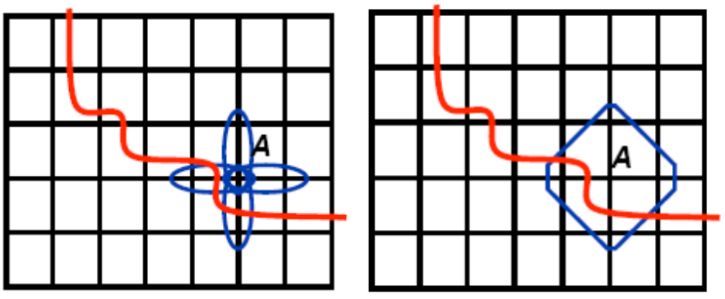


Fig. 3.2: Exemple de représentation d'un hypergraphe [4].

Dans la figure 3.2 on représente un exemple d'hypergraphe où :
$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$,
$E = \{e_1, e_2, e_3, e_4\} = \{\{v_1, v_2, v_3\}, \{v_2, v_3\}, \{v_3, v_5, v_6\}, \{v_4\}\}$.

3.1.3 Avantage des hypergraphes

Plusieurs traveaux mettent en évidence les avantages des hypergraphes [17, 20, 5, 19]. On represente sur la figure 3.3-(a) une modélisation d'une grille en graphe dont les points sont les sommets du graphe et sur la figure (b) une modélisation en hypergraphe. Imaginons que le partitionneur va faire une coupe selon la ligne rouge pour obtenir 2 partitions. Dans le cas du graphe, on constate que le partitionneur a coupé deux arêtes liant l'élément A à l'autre partie. Ce qui signifie qu'il faut deux communications pour cet élément avec l'autre partition pour achever sa part de calcul, alors qu'en réalité il ne faut qu'une seule communication. C'est ce qui est estimé avec plus de précision en modélisant avec un hypergraphe sur la figure 3.3-(b). En effet on ne coupe qu'une seule hyper-arête.



(a) La modélisation par graphe ne donne qu'une approximation sur le volume de communication.

(b) La modélisation par hypergraphe donne une mesure exacte du volume de communication.

Fig. 3.3: Modélisation de la communication par un graphe ou un hypergraphe [17, 20, 5, 19].

Si on essaye de regarder de plus près, on constate que sur l'exemple de la figure 3.4 l'élément e doit communiquer avec deux voisins qui ne sont pas dans sa partition. Si on modélise avec un graphe, alors on doit compter deux communications pour l'élément e entre le processeur rouge et le processeur bleu car le partitionneur a coupé 2 arêtes. Par contre, en modélisant avec un hypergraphe l'hyper-arête liée à cet élément sera coupée une seule fois et donc on compte une seule communication entre les deux processeurs. Ce dernier cas est plus réaliste car effectivement il y aura une seule communication entre les deux processeurs afin de faire le calcul pour l'élément e.

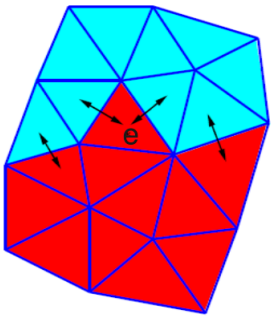


Fig. 3.4: Communication d'un élément sur la frontière de la coupe avec ses voisins [17].

3.2 Le problème de partitionnement de graphes et d'hypergraphes

3.2.1 Explication du problème

Le problème de partitionnement de graphe consiste à diviser un graphe en morceaux, tels que les différentes parties ont à peu près la même taille et avec peu de liens entre les morceaux.

Prenons l'exemple d'un graphe $G(V, E)$, où V désigne l'ensemble des sommets et E l'ensemble des arêtes. Le problème de partitionnement de graphe non pondéré est le suivant :

Compte tenu de G et un entier $k > 1$, partitionner V en k parties (sous-ensemble) V_1, V_2, ... V_k tels que les parties sont disjointes et ont la même taille tout en minimisant le nombre d'arêtes coupées dans les différentes parties. En pratique, un petit déséquilibre ε dans la taille d'une partie est généralement admis, et le critère d'équilibre s'écrit $MAX_i|V_i| \leq \frac{(1+\epsilon)|V|}{k}$.

Dans le cas général d'un graphe pondéré, les sommets ou/et des arêtes peuvent être pondérés. Le problème de partitionnement de graphe consiste alors à diviser G en k parties disjointes, telles que **les parties ont approximativement le même poids** et **la taille de la coupe est minimisée**[2]. Lorsque $k = 2$, le problème est également appelé *bissection de graphe*.

On peut étendre le problème au partitionnement d'hypergraphes. L'avantage d'un hypergraphe est de pouvoir connecter plus de deux sommets par une hyper-arête.

Le problème de partitionnement de graphe et d'hypergraphe est un problème NP-complet[3] [15], Par conséquent, la recherche d'une solution optimale nécessite une exploration combinatoire de l'espace des solutions. De nombreuses heuristiques ont été proposées afin de trouver des solutions quasi-optimales en un temps polynomial. Dans la plupart de ces méthodes, le problème de partitionnement est considéré dans sa globalité (méthodes directes) et l'espace des solutions explorées augmente rapidement par rapport à la taille du problème, ce qui rend ces méthodes coûteuses. Afin de réduire la taille des problèmes considérés, les algorithmes multi-niveaux ("*multi-level*") utilisent une approche *divide & conquer* représentée sur la figure 3.5.

[2] *La taille d'une coupe* est la somme des poids des arêtes qu'elle contient, tandis que *le poids d'une partie* est la somme des poids des sommets dans la partie.

[3] cf. Glossaire

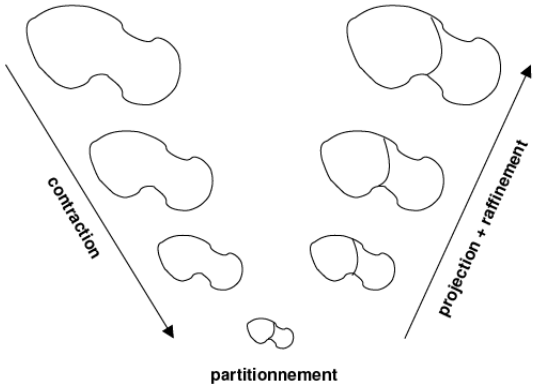


Fig. 3.5: partitionnement multi-niveaux [12].

Dans cette approche, on construit une famille de graphes de plus en plus petits (par contraction), puis on calcule un placement valide sur le plus petit graphe et on projette le résultat de graphe en graphe jusqu'au graphe initial. À la remontée du placement, il est possible d'appliquer un algorithme d'optimisation locale afin d'affiner le placement calculé (raffinement). Actuellement, les meilleurs algorithmes de placement connus sont des algorithmes multi-niveaux [20, 5, 21], qui utilisent un algorithme de décomposition pour calculer le placement sur le plus petit graphe contracté.

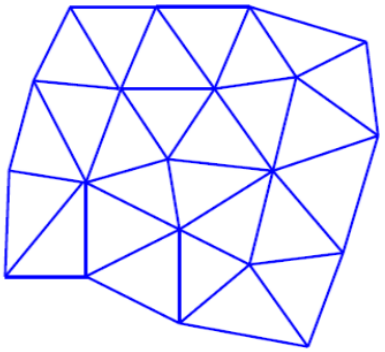
3.2.2 Outils de partitionnement

PATOH [20] est l'outil réputé le plus performant pour le partitionnement d'hypergraphe. Il permet également de partitionner des graphes. Malgré son manuel d'utilisation incomplet [28] on a pu l'utiliser assez facilement. D'après l'article [20], *PATOH* utilise la méthode de la bisection récursive. *SCOTCH* [23] est déjà utilisé actuellement dans le client *AMEL* de *LibMultiScale* [8] pour faire le partitionnement du graphe généré à partir du maillage. *METIS* [2] est aussi un outil de partitionnement performant pour le partitionnement des graphes. Il existe aussi dans la même famille *HMETIS* qui est un partitionneur d'hypergraphe.

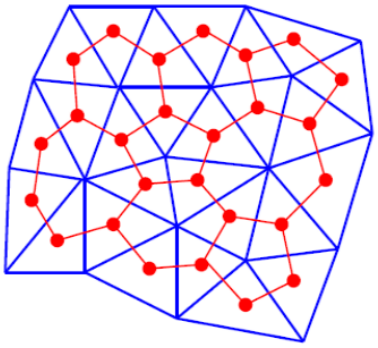
3.2.3 Exemple de partitionnement

Une application importante de partitionnement de graphe ou d'hypergraphe est l'équilibrage de charge pour le calcul parallèle, où chaque sommet correspond à une tâche de calcul élémentaire et les arêtes correspondent à des dépendances dans les calculs. Il existe à l'heure actuelle plusieurs outils pour le partitionnement de graphe et d'hypergraphe (voir la section 3.2.2).

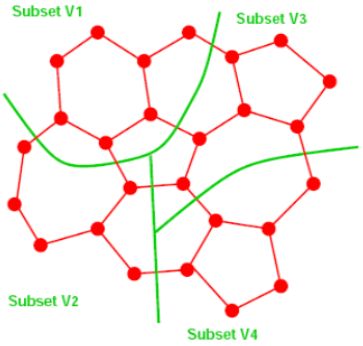
Sur la figure 3.6 on a un exemple de partitionnement d'un maillage modélisé par un graphe. On se donne un bout de maillage (le maillage est une discrétisation qui sert de support au calcul), et on modélise les éléments (e.g. triangles) par les sommets du graphe qu'on se propose de partitionner.



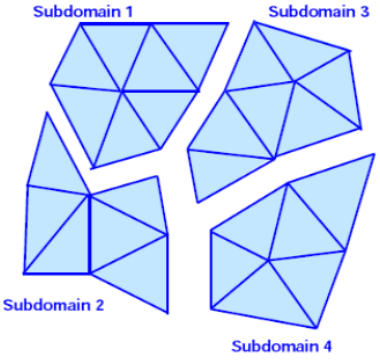
(a) Maillage initial.



(b) Modélisation des triangles en sommets de graphe.



(c) Partitionnement du graphe.



(d) reconstruction du maillage partitionné en sous domaines.

Fig. 3.6: Exemple de partitionnement d'un graphe dans le cas de maillages [17, 20, 5].

Troisième partie

Modelisation et réalisations : module de partitionnement et d'équilibrage de charge

4

Problèmatique et modélisation

Sommaire

4.1 Description de la problèmatique **30**
4.2 Cas de test de référence **31**
4.3 Dépendances des calculs **33**
4.3.1 Formules préliminaires pour le calcul des coûts 33
4.3.2 Dépendence de calcul entre les éléments 36
Coût de calcul de la force F_I 37
4.4 Modélisation avec un hypergraphe **39**
4.4.1 Modélisation du maillage avec un hypergraphe 39
4.4.2 Modélisation du couplage 41
4.5 Calcul des coûts **42**
4.5.1 Coût de calcul des éléments du maillage 42
Dans la zone normale 43
Dans la zone de recouvrement 43
4.5.2 Coût des hyper-arêtes de l'hypergraphe 43
Hyper-arête normale 44
Hyper-arête de couplage 44
4.5.3 Etude d'un exemple 44
4.6 Différentes stratégies de partitionnement étudiées **45**
4.7 Optimisation **47**

Résumé : Dans ce chapitre nous posons tout d'abord la problématique en illustrant avec un exemple de réference qui nous servira pour évaluer les résultats que nous présenterons dans les chapitres suivants. Ensuite, nous discuterons de la modélisation du problème et des calculs de coûts que nous avons effectué pour pondérer les sommets et les arêtes/ hyper-arêtes du graphe/hypergraphe. Enfin, Nous décrirons les différentes stratégies que nous avons utilisées pour partitionner le graphe ou l'hypergraphe.

Le materiau sera décrit avec deux modèles en même temps pour bénéficier des avantages de chacun : la dynamique moléculaire, dans la zone où l'on souhaite avoir plus de précision et la mécanique des milieux continus pour le reste. Entre ces deux zones, il y a une zone d'intersection dans laquelle coexiste les deux modèles, c'est la zone de recouvrement (ou bridging zone) comme c'est indiqé sur la figure 2.1. L'enjeux majeur pour avoir un code extensible réside dans l'équilibrage des charges des deux codes lorsqu'on les exécute sur plusieurs processeurs. D'après les résultats des expériences effectuées dans [8], on constate que le coût de calcul du couplage est loin d'être insignifiant et qu'il doit être pris en compte dans le partitionement, c'est l'objet de ce travail.

On va tout d'abord voir la manière avec laquelle on découpe les domaines puis la répartition de la charge entre les diffŕents processeurs.

4.1 Description de la problèmatique

Décomposition de domaine du modèle de dynamique moléculaire

Il s'agit d'une décomposition en boîtes uniformes du domaine de la dynamique moléculaire. Chaque boîte est associée à un et un seul processeur. Le cas présenté sur la figure 4.1 est celui d'une initiation de fissure pour un domaine 2D, réparti sur 20 processeurs.

Fig. 4.1: Décomposition par macro-boîte du domaine utilisé par la plupart des codes de dynamique moléculaire [8].

On constate que les 10 processeurs ayant en charge les 10 régions rectangulaires, en bas de la figure 4.1, vont participer au calcul de la phase de couplage (phase 2) d'un pas de temps d'une simulation. Ce sont les processeurs qui se chargent de la zone Ω^R

Décomposition de domaine du modèle de mécanique des milieux continus utilisant les élèments finis

le domaine est entièrement discrétisé par un maillage. Une technique couramment utilisée pour équilibrer la charge est celle basée sur un bipartitionement récursif de l'espace.

La figure 4.2 présente le découpage d'un domaine rectangulaire discrétisé par un maillage non structuré (à base de triangles) et produit par *METIS* [2]. Le partitionneur de graphe *Scotch* [23] a aussi été utilisé dans *LibMultiScale*. Le partitionnement qui existait déjà est basé sur une modélisation des éléments finis par un graphe. Sur la figure la zone qui correspond à Ω^C correspond à la zone continue et Ω^R correspond à la zone de recouvrement.

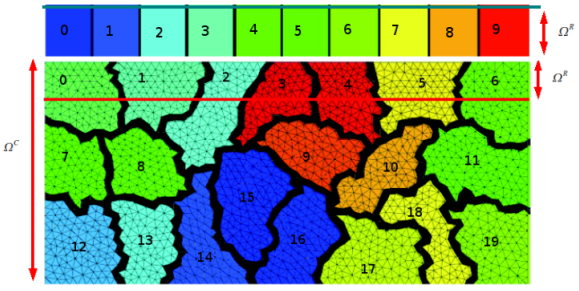


Fig. 4.2: Décomposition de domaine par bipartitionnement récursif d'un maillage non structuré. [8]

La décomposition algorithmique en boîtes de l'espace de simulation de la dynamique moléculaire ne peut pas être remise en cause car elle est imposée par le code lui-même. De plus, chaque boîte doit être affectée à un et un seul processeur de DM[1]. On ne peut donc essentiellement jouer que sur la nature des partitionnements du maillage d'éléments finis du modèle de la mécanique des milieux continus. C'est

[1] On utilise DM, afin de simplifier la notation, pour désigner le domaine de la dynamique moléculaire

pourquoi nous allons essentiellement nous intéresser à la décomposition du domaine de la mécanique des milieux continus.

On remarque sur la figure 4.2 d'une part, du point de vue de l'équilibrage de la charge entre les processeurs de la mécanique des éléments finis, que le partitionnement n'est pas très régulier, bien que les parties semblent être équilibrées. En effet, certains processeurs ne participent pas au calcul pendant la phase 2 de couplage (voir la section 2.2.3 et la figure 2.4) car ils ne sont pas présents dans la zone de recouvrement Ω^R. Ces processeurs seront donc au repos pendant la phase de correction des vitesses[2], alors que les processeurs qui sont en charge de la zone de recouvrement seront occupés pendant toute la simulation et durant les deux phases de calcul.

D'autre part, on remarque du point de vue des communications, qu'il y a aussi un déséquilibre en terme de nombre de communications entre les processeurs de la zone moléculaire et les processeurs de la zone continue. En effet certains processeurs de DM communiquent avec un seul processeur de EF[3], d'autres avec 2 ou 3 etc... (cf. figure 4.4).

Ces deux remarques constituent les deux principales contraintes de notre problème d'équilibrage de la charge. Dans le cas idéal, on souhaiterait d'une part que tous les processeurs affectés aux *EF* participent au calcul du couplage pour aller plus vite et pour équilibrer la charge de calcul entre les processeurs et entre les 2 phases de calcul et d'autre part avoir un minimum de communication avec les processeurs de la dynamique moléculaire.

4.2 Cas de test de référence

Les cas de test et les résultats que nous présentons dans cette section proviennent de la thèse [8]. On décrit ici la propagation d'une fissure dans un cristal d'argon en dimension 2 avec des conditions limites (voir figure 4.3). Nous avons une zone atomique rectangulaire de 300 Å par 600 Å constituée de plus de 135 000 atomes d'argon modélisés comme un cristal parfait.

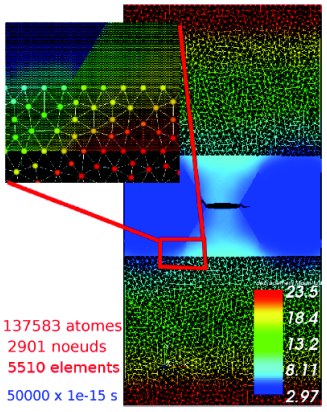


Fig. 4.3: Un cas de propagation de fissure [8].

[2]La phase de correction des vitesses est la phase de couplage, reprendre le chapitre 2.2 pour plus de détails.
[3]On utilise *EF* pour désigner le domaine continue décrit par les éléments finis.

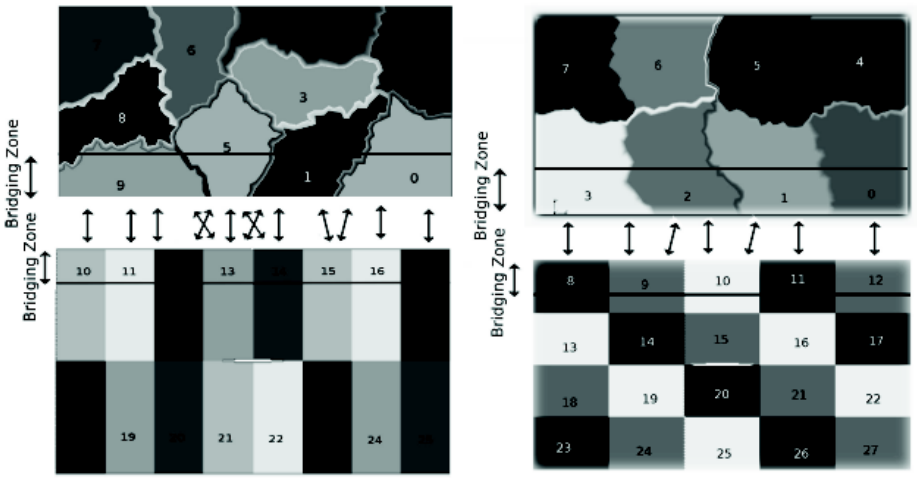


Fig. 4.4: Distribution de la charge dans les cas de 16/8 et 20/8 processeurs sont affectés à la dynamique moléculaire(en bas)/mécanique continue(en haut). [8]

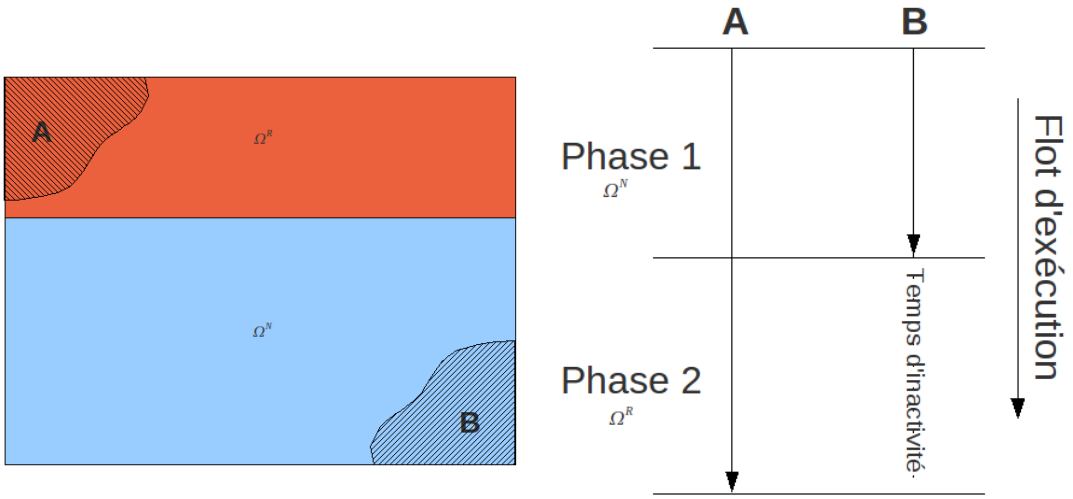
En analysant les resultats du point de vue de l'équilibrage de la charge et des communications entre les deux modèles (figure 4.4), on constate que les processeurs n'ont pas la même charge de calcul pendant la première phase et la seconde phase[4] d'un pas de simulation. Par exemple le processeur 9 qui fait les calculs pour la mécanique des milieux continus (sur la figure 4.4 à gauche) doit communiquer avec 4 processeurs en face qui sont les processeurs 10, 11, 12 et 13 de la dynamique moléculaire. De plus, on constate que ce processeur a beaucoup de charge de calcul pendant la phase de couplage par rapport à la charge des processeurs 5 et 1. Il va ralentir les autres processeurs du milieu continu qui n'ont pas beaucoup de charges pendant la phase de couplage comme le 5 et le 1 ou le 8 qui a encore beaucoup moins de charge de calcul pour cette phase. Il y a un déséquilibre en terme de communication entre les processeurs de EF[5] et MD[6]. En effet, le processeur 9 des EF doit cummuniquer avec 4 processeurs de DM (indiqués précédement), alors que le processeur 5 de EF communique avec seulement le processeur 13 de MD. De plus, les processeurs de EF qui n'ont pas de charge pendant la phase de couplage, ne communiquent avec aucun processeur de MD. La répartition obtenue à droite sur la figure 4.4 est mieux équilibrée en terme de communication mais n'est pas bien équilibrée en terme de la répartition de la charge globale. En effet les processeurs 4, 5, 6 et 7 qui n'ont pas de charge dans la zone de recouvrement, seront au repos pendant la phase de couplage. La répartition serait plus efficace si ces processeurs participaient aussi à la phase de couplage.

La figure 4.5 represente un cas de partitionnement dans lequel le processeur A effectue ses calculs pendant la phase 1 et la phase 2 de la simulation car il est en charge d'une partie (hachurée en haut à gauche) de la zone de recouvrement Ω^R alors que le processeur B ne fait des calcul que pendant la phase 1 de la simulation car il est en charge d'une partie (hachurée en bas à droite) de la zone normale Ω^N (hors de la zone de couplage). Ce dernier sera au repos pendant la phase 2 de couplage. Il est clair ici que le processeur A a beaucoup plus de charge que le processeur B bien que, géométriquement ces deux processeurs ont deux régions de même surface qui leur a été affectée initialement par le partitionneur.
On peut corriger ce déséquilibre en affectant deux fois plus de charges pour le processeur B mais cela ne résoud pas le problème car à cause de la barrière de synchronisation, le processeur A aura un temps d'inactivité à la fin de la première phase.

[4] Les processeur qui participent à la phase de couplage sont les processeurs présent dans la zone de recouvrement (bridging zone sur la figure 4.4)

[5] On utilisera *EF* pour désigner le domaine de la mécanique des milieux continus (abréviation de **F**inite **E**lement)

[6] On utilisera *MD* pour désigner le domaine de la dynamique moléculaire (abréviation de **M**olecular **D**ynamic)



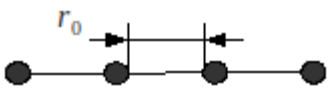
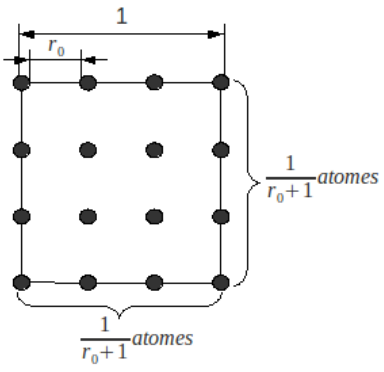
(a) Parties affectées au processeurs A et B.

(b) Charge de calcul pendant les deux phases pour les processeurs A et B.

Fig. 4.5: Comparaison entre les charges de calcul d'un processeur A ayant en charge une partie de la zone de recouvrement Ω^R (partie hachurée en haut à gauche de la figure (a)) et d'un processeur B ayant en charge une partie de la zone normale Ω^N (partie hachurée en bas à droite de la figure (a)).

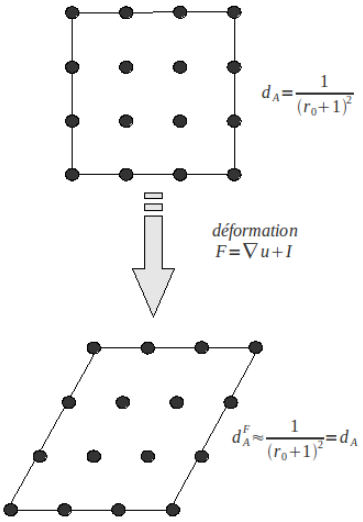
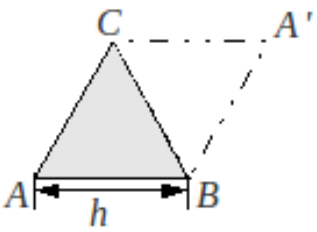
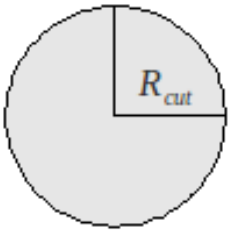
4.3 Dépendances des calculs

4.3.1 Formules préliminaires pour le calcul des coûts

Notation	Description	Schéma	Formule
r_0	Distance inter-atomique pour l'Argon.		$r_0 \simeq 1.224521$ Å (4.1)
d_A	Densité atomique (nombre d'atomes par unité de surface)		$d_A = \frac{1}{(r_0+1)^2}$ (4.2)

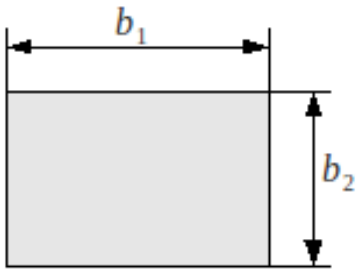
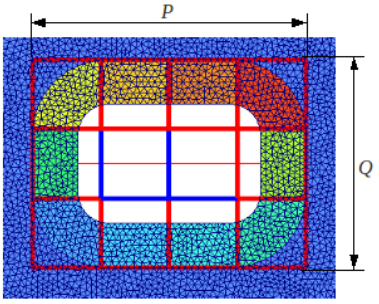
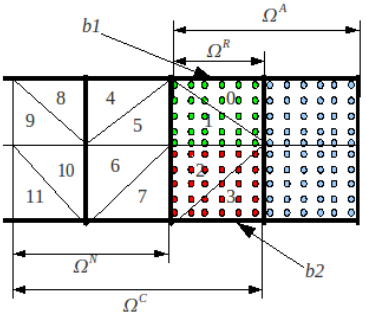
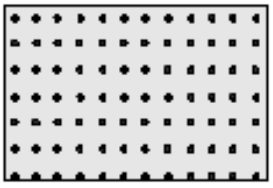
... suite à la page suivante ...

... suite de la page précédente ...

Notation	Description	Schéma	Formule
d_A^F	Densité atomique après déformation (nombre d'atomes par unité de surface déformé avec l'approximation de Cauchy-Born). Dans le cas de faibles déformations, on fera l'approximation $d_A^F \simeq d_A$. i.e $F = Id + \varepsilon B$ où B est une perturbation.		$d_A^F = d_A = \frac{1}{(r_0+1)^2}$ (4.3)
$\mathcal{A}_T$	Aire d'un triangle ABC. Pour simplifier on calcule l'aire dans un plan (i.e en supposant que les ordonnées selon z sont constantes). On pourra considérer que le triangle est équilatéral de coté h avec h une donnée du problème.		$\mathcal{A}_T = \frac{1}{2}.\|\|\overrightarrow{AB} \wedge \overrightarrow{AC}\|\|$ (4.4) $\mathcal{A}_T \simeq \frac{h^2}{2}$ (4.5)
$\mathcal{A}_{D_{Rcut}}$	Aire d'un disque de rayon R_{cut}, où $Rcut$ est le rayon de coupure, qui est donné.		$\mathcal{A}_{D_{Rcut}} = \pi.R_{cut}^2$ (4.6)

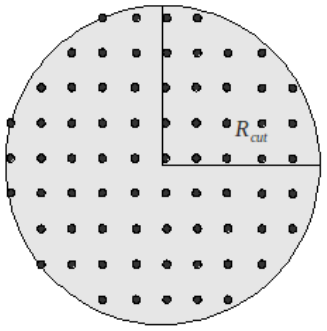
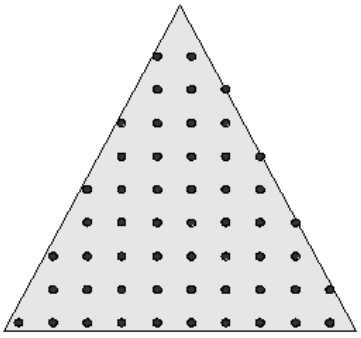
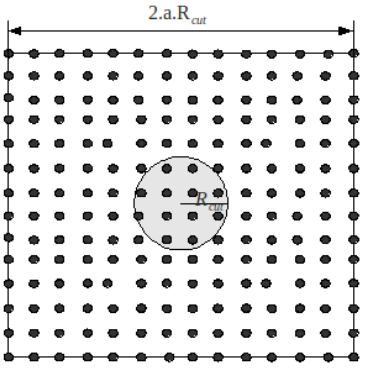
... suite à la page suivante ...

... suite de la page précédente ...

Notation	Description	Schéma	Formule
$\mathcal{A}_{Bote}$	Aire d'une boîte de dimension b_1 et b_2		$\mathcal{A}_{bote} = b_1.b_2$ (4.7)
P et Q	Nombre de processeurs affectés à la dynamique moléculaire respectivement selon x et y dans la zone de recouvrement.		
N_e^{BZ}	Nombre d'éléments dans la zone de recouvrement avec $\mathcal{A}_{BZ}$ la surface de la zone de recouvrement.		$N_e^{BZ} \simeq \frac{\mathcal{A}_{BZ}}{\mathcal{A}_T}$ (4.8)
N_{bote}^A	Nombre d'atomes dans une boîte de dimension b_1 et b_2.		$N_{bote}^A = \mathcal{A}_{bote}.d_A$ (4.9)

... suite à la page suivante ...

... suite de la page précédente ...

Notation	Description	Schéma	Formule
$N^A_{D_{Rcut}}$	Nombre d'atomes dans un disque de rayon R_{cut}.		$N^A_{D_{Rcut}} = \mathcal{A}_{D_{Rcut}}.d_A$ (4.10)
N^A_T	Nombre d'atomes dans un triangle ABC.		$N^A_T = \mathcal{A}_T.d_A$ (4.11) $N^A_T = \frac{1}{2}\left(\frac{h}{r_0+1}\right)^2$ (4.12)
$N^A_{Crystal}$	Nombre d'atomes dans un crystal de dimension $2aRcut$, où $Rcut$ est le rayon de coupure, qui est donné et a est une variable donnée du problème (dans le cas de l'Argon on prendra $a = 4$)		$N^A_{Crystal} = 4.a^2.Rcut^2.d_A$ (4.13) $N^A_{Crystal} \simeq 16 N^A_{D_{Rcut}}$ (4.14)

Tab. 4.1: Outils pour le calcul des coûts.

4.3.2 Dépendence de calcul entre les éléments

Comme on l'a expliqué dans la section 2.2.2, la force sur un noeud I, F_I, doit etre calculée sur chacun des noeuds du maillage des éléments finis pour mettre à jour la vitesse de déplacement par la suite. Pour

chaque élément, on évalue la force sur ses 3 noeuds. Pour comprendre comment évaluer cette force, on considère un élément du maillage et ses voisins qui participent au calcul. Sur la figure 4.6, on représente les voisins d'un éléments (ABC).

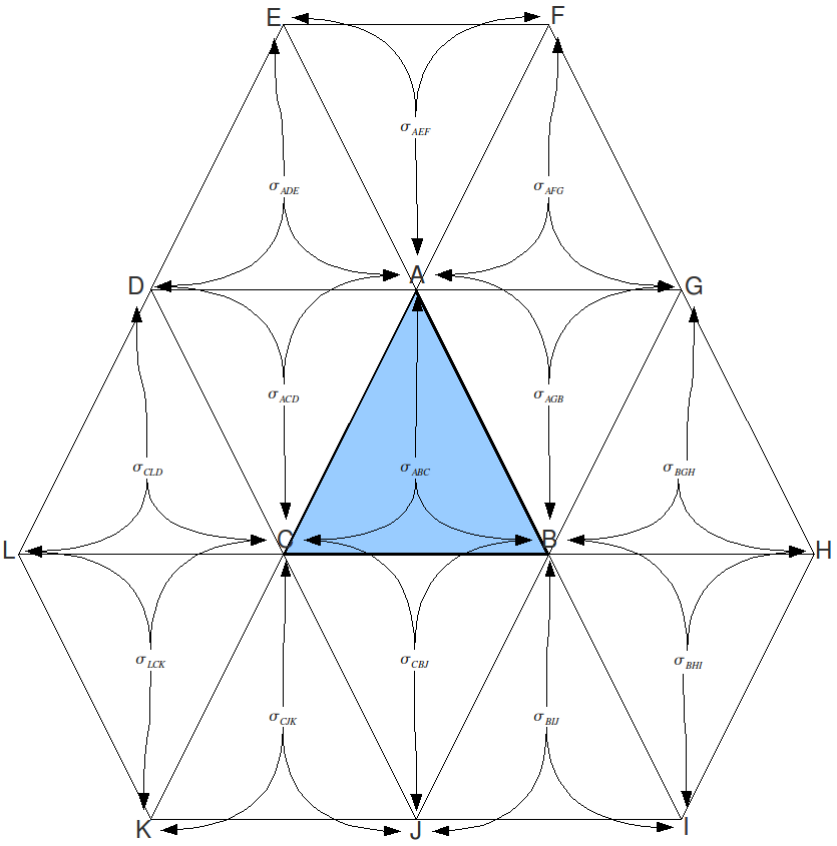


Fig. 4.6: Dépendence d'un élément avec ses voisins.

En se referant à la tâche d'integration de la mécanique des milieux continues T_S^C (cf. section 2.2.2, page 15), et au schéma de dépendence d'un élément avec ses voisins (figure 4.6), on a tracé sur la figure 4.7 les dépendances de calcul pour un élément ABC du maillage.

Coût de calcul de la force F_I

Notation On note $\mathcal{C}_{op}$ le coût d'une opération élémentaire.

Reprenons l'algorithme d'integration en temps de la partie macroscopique T_S^C (cf. section 2.2.2, page 15).

$$T_S^C \begin{cases} \left.\begin{array}{lcl} \dot{u}_I^{n+\frac{1}{2}} & = & \dot{u}_I^n + \frac{\triangle t}{2M_I} F_I^n \\ u_I^{n+1} & = & u_I^n + \triangle t \dot{u}_I^{n+\frac{1}{2}} \end{array}\right\} * & \text{boucle sur les sommets} \\ \left.\begin{array}{l} calcul\ de\ F_I^{n+1} \end{array}\right\} ** & \text{boucle sur les triangles} \\ \left.\begin{array}{lcl} \dot{u}_I^{n+1} & = & \dot{u}_I^{n+\frac{1}{2}} + \frac{\triangle t}{2M_I} F_I^{n+1} \end{array}\right\} *** & \text{boucle sur les sommets} \end{cases}$$

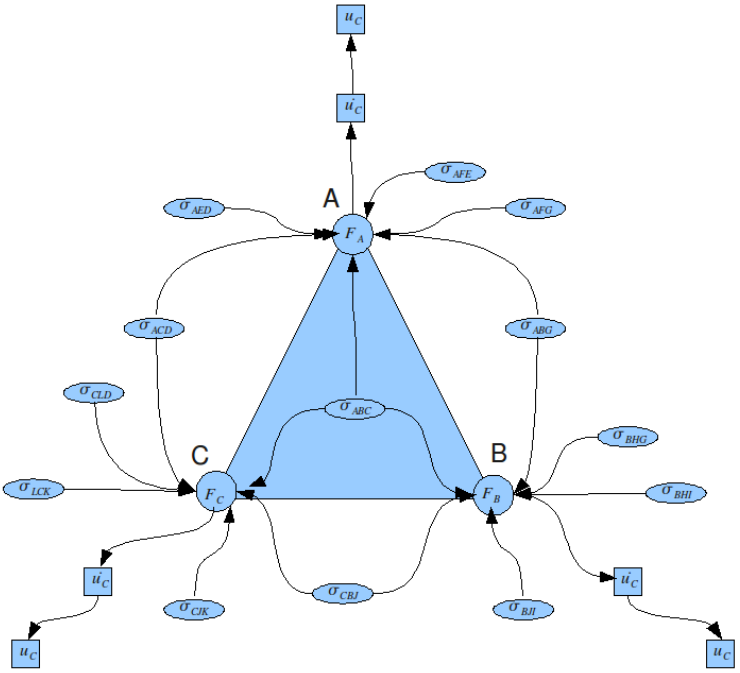


Fig. 4.7: Dépendence de calcul dans la tâche d'integration pour les éléments finis.

La boucle ** sur les triangles contient les 3 étapes décrites par l'algorithme 1, où *"déformation-crystal"* permet de déformer le crystal comme dans la figure 4.8, *"calcul-potentiel"* permet de calculer le potentiel de *piola* donné par la formule 4.15 [8, 11] qui relie l'énergie interieure au tenseur de *Piola-Kirchoff*. L'étape *"pondération-sommets"* permet d'accumuler la valeur de la force calculée sur les sommets de l'élément du maillage considéré. Le tenseur de Piola est calculé par :

$$P_{iJ} = \sum_m w'^e(r^m).\frac{r_i^m r_J^{0m}}{r^m} \tag{4.15}$$

où on somme sur tout les atomes qui sont dans un disque de rayon R_{Cut}, r^m la distance entre l'origine et le m^{eme} atome et w est l'énergie potentielle interne qui est donnée par l'équation 4.16 (on ne rentre pas dans les détails de ces équations [8], on ne s'interesse ici qu'à la complexité de leur calcul) :

$$\begin{aligned} W &= \sum_i \sum_{j>i} \phi(\|x_i - x_j\|) \\ avec \quad \phi(r) &= 4\epsilon[(\frac{c}{r})^{12} - (\frac{c}{r})^6] \end{aligned} \tag{4.16}$$

Algorithme 1 Algorithme de calcul de la force F

Pour tout les éléments du maillage **Faire**
  déformation-crystal()
  calcul-potentiel()
  pondération-sommets()
FinPour

On note $\mathcal{C}_F$ la complexité de cet algorithme qui est la somme des complexités C_1, C_2 et C_3 respectivement

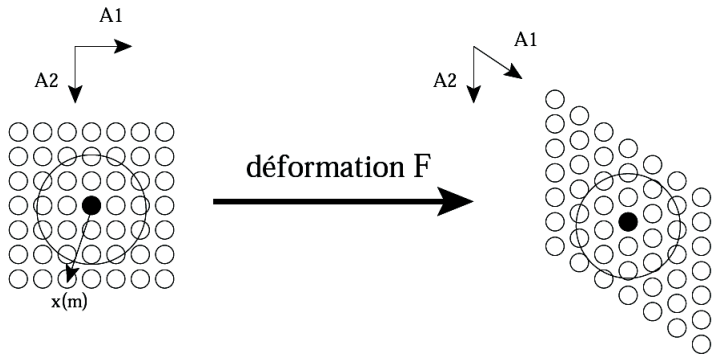


Fig. 4.8: Déformation du crystal pour l'approximation de cauchy-born [8].

de chaque étape. Cette complexité est donnée par 4.17.

$$
\begin{aligned}
C_F &= \overbrace{\alpha.N^A_{Crystal}C_{op}}^{C_1} + \overbrace{(\beta_1 N^A_{D_{Rcut}}).(\beta_2 N^A_{D_{Rcut}})C_{op}}^{C_2} + \overbrace{\delta C_{op}}^{C_3} \\
&= [d^2 N^A_{Crystal}C_{op} + 26 \times 8(N^A_{D_{Rcut}})^2 + 3(d+1)]C_{op} \\
&\simeq [208(N^A_{D_{Rcut}})^2 + 16d^2 N^A_{D_{Rcut}} + 3(d+1)]C_{op}
\end{aligned}
\tag{4.17}
$$

où C_{op} est le coût d'une opération élémentaire [7] et d est la dimension du problème.

4.4 Modélisation avec un hypergraphe

4.4.1 Modélisation du maillage avec un hypergraphe

Comme nous l'avons expliqué dans la section 3.1.3, la modélisation par un hypergraphe a l'avantage de pouvoir estimer les coûts de manière plus précise que dans la cas d'une modélisation par un graphe. Un autre avantage des hypergraphes est le fait de pouvoir regrouper plusieurs éléments ayant une propriété commune dans une seule hyper-arête. Cela nous permettra de modéliser le couplage (cf. section 4.4.2).

Les vrais voisins d'un élément (cf. figure 4.6) sont tous les triangles qui l'entourent et non seulement les triangles qui partagent un côté avec lui (dans le cas 2D). En effet, le calcul sur chaque élément du maillage dépend des 12 voisins qui l'entourent (voir figure 4.6). Mais on ne va considérer que les voisins qui ont plus de contribution que les autres i.e dans le cas 2D par exemple, ce sont les triangles ayant un coté en commun. Dans la figure 4.6, les voisins du triangle ABC qui seront considérés sont les triangles AGB, BJC et CDA.

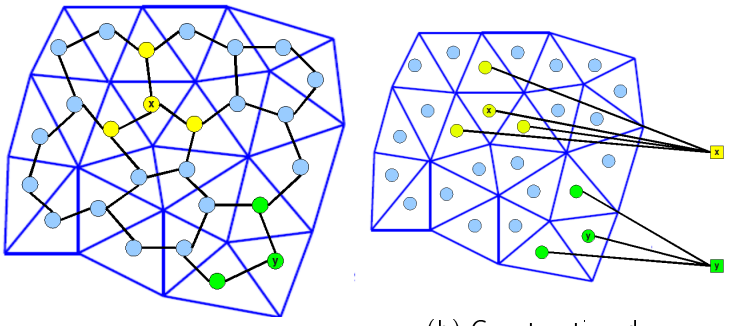
On décrit notre hypergraphe de la manière suivante : chaque élément (triangle) [8] sera regroupé avec ses premiers voisins (3 voisins dans le cas 2D) [9] dans une seule hyper-arête. Cette modélisation est représentée sur la figure 4.9. Pour chaque élément, on construit une hyper-arête. Exemple pour l'élément x on construit l'hyper-arête jaune x qui regroupe les sommets voisin de x.

Le choix de ne considérer que les premiers voisins pour chaque éléments est justifié par le fait que l'augmentation de la "connectivité" n'a, à priori, pas d'influence sur le résultat du partitionneur. En effet, si

[7] Pour simplifier, on suppose que les opérations d'addition, de soustraction, de multiplication et de division ont le même coût unitaire C_{op}.

[8] On confondra les termes *élément du maillage* et *triangle* qui correspondent à un sommet du graphe généré à partir du maillage.

[9] Dans une première approche, on dit que deux triangles du maillage sont voisins s'ils partagent un côté.



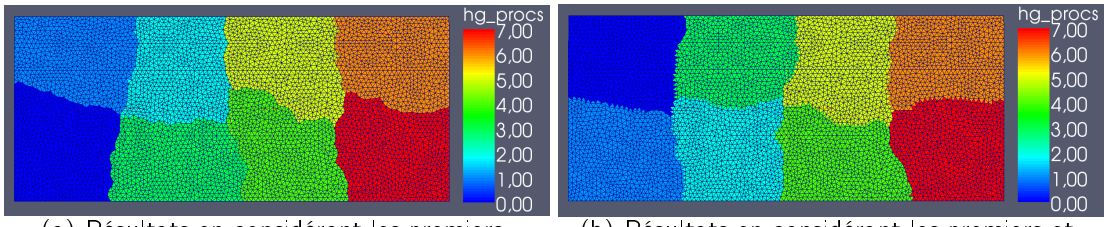
(a) Regroupement des éléments voisins pour x et y.

(b) Construction des hyper-arêtes associées aux triangle x et y.

Fig. 4.9: Modélisation en hypergraphe. [17]

on considère plus de voisins (i.e plus d'éléments par hyper-arête), cela ne fait qu'augmenter la "connectivité" sur l'hypergraphe du point de vue du partitionneur.

Nous avons fait un test pour vérifier notre simplification. Sur la figure 4.10, on représente les résultats du test de variation de connectivité, où la figure (a) représente le résultat de partitionnement en ne considérant que 4 éléments par hyper-arête (i.e les voisins à une distance 1) et la figure (b) représente le résultat de partitionnement en considérant les voisins qui participent au calcul de chaque élément jusqu'à une disatnce 2. La seule différence apparente est au niveau du coût de la coupe effectuée par le partitionneur (edgecut) : i.e dans le premier cas cette valeur est 445, dans le second cas cette valeur est de 959 ce qui est normal car dans le second cas, les hyper-arête se chevauchent davantage. Les deux résultats sont équivalents d'après les pourcentages de charge affecté à chaque processeur qui sont du même ordre de grandeur, ce qui justifie notre simplification. Pour être plus rigoureux, il faudrait étudier le comportement du partitionneur en faisant varier le nombre de sommets par hyper-arête, en commençant par 1 sommet par hyper-arête (cas d'un graphe) et rajouter les voisins de proche en proche jusqu'à ce que chaque hyper-arête contienne tous les sommets (cas limite).



(a) Résultats en considérant les premiers voisins.

- nb parts = 8
- nb vertices = 9542
- nb hyperedges = 9542
- edgecut = 445
- total weights = 9542
- % part weights = [12 13 13 13 13 12 13 12]

(b) Résultats en considérant les premiers et seconds voisins.

- nb parts = 8
- nb vertices = 9542
- nb hyperedges = 9542
- edgecut = 959
- total weights = 9542
- % part weights = [13 13 13 13 12 12 12 12]

Fig. 4.10: Influence de l'augmentation de la connectivité sur les résultats de partitionnement.

4.4.2 Modélisation du couplage

L'un des avantages de l'utilisation d'un hypergraphe est de pouvoir estimer avec précision les coûts de communication (voir la section 3.1.3). Maintenant on va utiliser un autre avantage des hyper-arêtes qui consiste à regrouper plusieurs sommets dans un seul ensemble. On pourra ensuite jouer sur le coût des hyper-arêtes ou des sommets pour contraindre le partitionneur de ne pas couper certaines hyper-arêtes car le principe du partitionneur est de minimiser les coûts des coupes. Naturellement si on met des coûts convenables on va guider le partitionneur, en quelques sorte, pour bien répartir la charge. Maintenant on va utiliser cette technique pour minimiser les communications dans la zone de recouvrement. En effet, on va bénéficier de la caractéristique principale des hypergraphes qui est le fait de regrouper plusieurs sommets dans une seule hyper-arête. On mettra un poids plus important[10] sur ces hyper-arêtes spéciales pour que le partitionneur essaye de ne pas les couper. On appellera ces hyper-arêtes spéciales : *hyper-arête de couplage*.

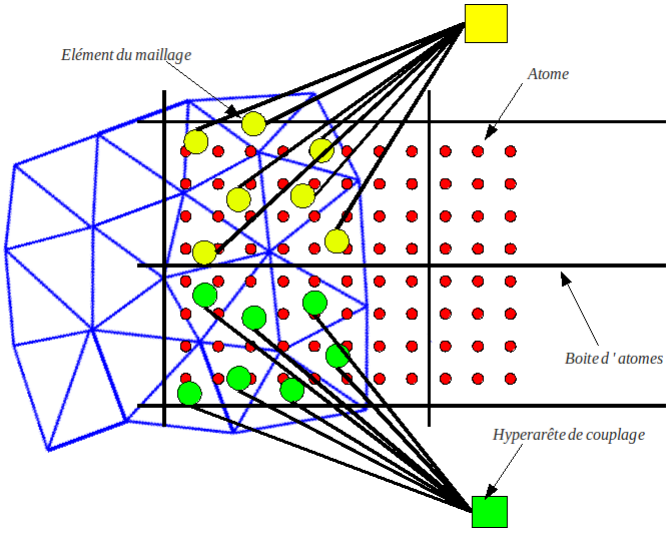


Fig. 4.11: Construction des hyper-arêtes de couplage.

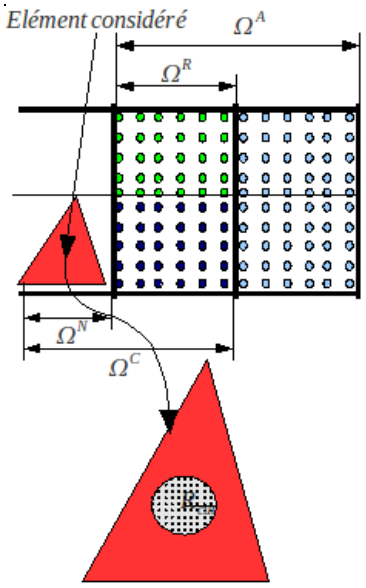
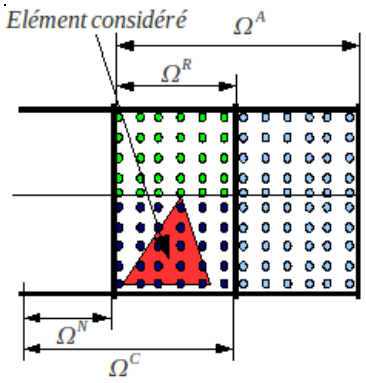
Sur la figure 4.11 nous représentons la zone de recouvrement qui fait intervenir en même temps des éléments du maillage de la zone continue (les triangles) et des boîtes d'atomes de la zone de la dynamique moléculaire. Pour construire les hyper-arêtes de couplage, on va regrouper les éléments du maillage qui appartiennent à une même boîte d'atomes. Ensuite on va leur donner un coût plus grand par rapport aux hyper-arêtes normales. Cela va contraindre le partitionneur à essayer au mieux de ne pas couper l'ensemble des éléments qui communiquent avec une même boîte d'atomes. On verra dans la partie résultat que cela va nous permettre d'optimiser les communications entre les processeurs de la dynamique moléculaire et les processeur de la mécanique des milieux continus.

[10] On fera varier les poids des sommets et des hyper-arêtes pour trouver les meilleurs valeurs (cf. Annexe F).

4.5 Calcul des coûts

On a vu dans le section 4.4 que les éléments du maillage sont représentés par les sommets d'un hypergraphe dont les hyper-arêtes sont constituées par les éléments voisins.

4.5.1 Coût de calcul des éléments du maillage

Notation	Description	Schéma	Formule
C_e	Coût de calcul pour un élément fini dans la zone normale (représenté par un sommet de l'hypergraphe).		cf. équation 4.18
C_e^{BZ}	Coût de calcul pour un élément fini dans la zone de recouvrement (représenté par un sommet de l'hypergraphe).		cf. équation 4.19

Tab. 4.2: Coût de calcul des éléments (sommets de l'hypergraphe).

Les poids qu'on affectera aux sommets de l'hypergraphe modélise le coût de calcul nécessaire pour un élément fini. Il y a deux types de sommets dans l'hypergraphe : les sommets qui sont dans la zone de recouvrement (ou bridging zone) et les sommets hors de cette zone (appelée *zone normale* dans la suite du document). Les coûts de calcul sont évalués en fonction du nombre d'atomes.

Dans la zone normale

C'est le coût du calcul de la force au centre de gravité d'un triangle (voir la formule 4.17) dans les conditions de l'approximation de Cauchy-Born. Cela correspond au nombre d'atomes dans un disque de rayon R_{cut} (Cf. tableau 4.1).

$$\begin{aligned} C_e = C_F &= \left[208\left(N^A_{D_{Rcut}}\right)^2 + 16d^2 N^A_{D_{Rcut}} + 3(d+1)\right] C_{op} \\ &= \left[208\pi^2\left(\frac{R_{cut}}{r_0+1}\right)^4 + \pi\left(\frac{4dR_{cut}}{r_0+1}\right)^2 + 3(d+1)\right] C_{op} \end{aligned} \tag{4.18}$$

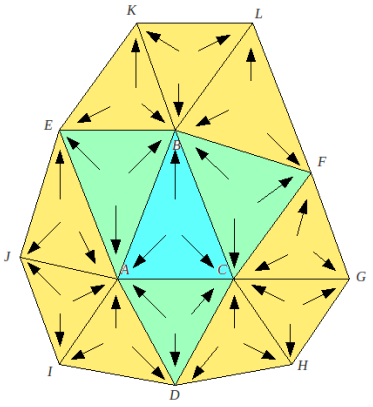
Dans la zone de recouvrement

En plus de ce coût, on ajoute le coût de couplage qui est le nombre d'atomes par éléments (cf. tableau 4.1). On supposera que les triangles sont isométriques dans la zone de recouvrement.

$$\begin{aligned} C_e^{BZ} &= C_e + C_{coup} = C_e + 3N^A_T.C_{op} \\ &= \left[208\left(N^A_{D_{Rcut}}\right)^2 + 16d^2 N^A_{D_{Rcut}} + 3N^A_T + 3(d+1)\right] C_{op} \\ &= \left[208\pi^2\left(\frac{R_{cut}}{r_0+1}\right)^4 + \pi\left(\frac{4dR_{cut}}{r_0+1}\right)^2 + \frac{3}{2}\left(\frac{h}{r_0+1}\right)^2 + 3(d+1)\right] C_{op} \end{aligned} \tag{4.19}$$

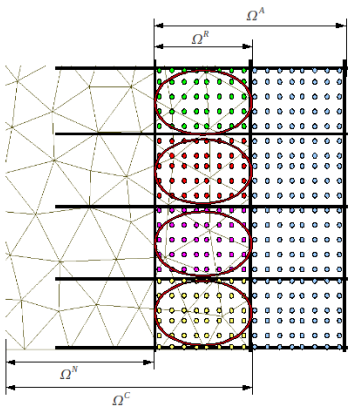
4.5.2 Coût des hyper-arêtes de l'hypergraphe

Le coût affecté aux hyper-arêtes modélise les communications, et s'exprime en volume de donnéesélémentaires à échanger entre les deux parties d'une hyper-arête coupée

Notation	Description	Schéma	Formule
C_h	Coût pour une hyper-arête normale (dans la zone normale ou dans la zone de recouvrement).		cf. équation 4.20

... suite à la page suivante ...

... suite de la page précédente ...

Notation	Description	Schéma	Formule
C_h^{Coup}	Coût pour une hyper-arête de couplage (qui est donc dans la zone de recouvrement).		cf. équation 4.21

Tab. 4.3: Coût des hyper-arêtes.

Hyper-arête normale

Après l'approximation que nous avons fait pour la modélisation des hyper-arêtes normales, on considère que chacune contient 4 éléments ayant un côté en commun. Ce coût est proportionnelle à la dimension du problème.

$$C_h \sim d \tag{4.20}$$

Hyper-arête de couplage

Une hyper-arête de couplage regroupe les éléments (triangles dans le cas 2D) de la zone de couplage qui sont dans une même boîte d'atomes (voir figure 4.11). L'objectif principal de ces hyper-arêtes est de guider le partitionneur pour essayer de ne pas couper les éléments qui appartiennent à une même boîte d'atomes. Nous allons faire varier le coût sur ces hyper-arêtes spéciales pour trouver la meilleure valeur γ qui nous permet d'obtenir les meilleurs résultats (voir les mesures de l'annexe F).

$$C_h^{Coup} \sim \gamma \tag{4.21}$$

4.5.3 Etude d'un exemple

Sur l'exemple de la figure 4.12, on represente un maillage simple dans le cas 2D contenant 12 éléments (triangles) dont 4 sont dans la zone de recouvrement Ω^R et 4 boîtes d'atomes dont deux, *b1* et *b2* sont dans la zone de recouvrement.

L'hypergraphe généré à partir de ce maillage contient 12 sommets $S_{i_{i\in 0..11}}$ et 12 hyper-arêtes normales $H_{i_{i\in 0..11}}$ (avec i l'indice associé à chaque élément) et 2 hyper-arêtes de couplage HC. Plus précisément, on a :

- sommets dans la zone de recouvrement : S_0, S_1, S_2, S_3,

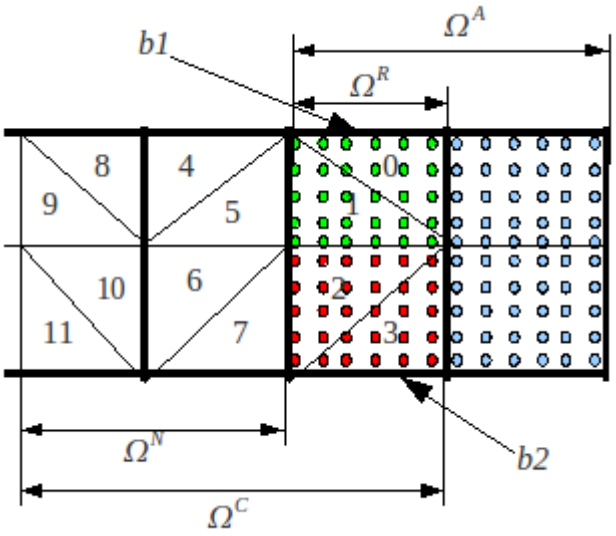


Fig. 4.12: Exemple de calcul de coût.

- sommet dans la zone normale : S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, S_{11},
- hyper-arêtes : H_0 : $\{S_0; S_1\}$, H_1 : $\{S_1; S_0; S_2; S_5\}$, H_2 : $\{S_2; S_1; S_3; S_7\}$, H_3 : $\{S_3; S_2\}$, H_4 : $\{S_4; S_5; S_8\}$, H_5 : $\{S_5; S_1; S_4; S_6\}$, H_6 : $\{S_6; S_5; S_7; S_10\}$, H_7 : $\{S_7; S_2; S_6\}$, H_8 : $\{S_8; S_4; S_9\}$, H_9 : $\{S_9; S_8; S_{10}\}$, H_10 : $\{S_{10}; S_6; S_9; S_{11}\}$, H_11 : $\{S_{11}; S_{10}\}$,
- hyper-arêtes : HC_1 : $\{S_0; S_1\}$, HC_2 : $\{S_2; S_3\}$.

Dans notre cas on prendra les valeurs suivantes : $d = 2$ (en dimension 2), $r_0 = 1.224521$ Å, $R_{cut} = 2,5r_0$ et $h = 5r_0$.

En appliquant les formules 4.18 ; 4.19 et 4.20 avec les valeurs précédentes, on obtient les coûts suivants :

$$C_e \simeq 179347\ C_{op}$$

$$C_e^{BZ} \simeq (179347 + 33)\ C_{op} = 179380\ C_{op}$$

$$C_h \simeq 2$$

On constate que $C_e \sim C_e^{BZ}$ car la valeur du coup de couplage qu'on rajoute à C_e est faible devant C_e. Cela ne sera pas détecté par le partitionneur. On verra dans le chapitre 6 que nous mettrons des coûts artificielles[11] pour avoir de meilleurs résultats.

Nota Bene : On ne prend pas en compte des coûts de communications extérieures dans le calcul de C_e^{BZ}.

4.6 Différentes stratégies de partitionnement étudiées

La figure 4.13 représente les différentes stratégies que nous avons testées pour partitionner le maillage. Nous avons commencé tout d'abord par un *partitionnement classique du maillage* en le modélisant par un graphe tel que les sommets du graphe sont les éléments finis du maillage et il y a une arête entre deux sommets s'ils représentent deux triangles ayant un côté en commun (comme l'exemple de la figure 3.6-b). Pour ce partitionnement, nous avons procédé de deux façons : *avec coût uniforme*[12] pour les sommets et les

[11] Pour déterminer expériementalement de bonnes valeurs des coûts avec les différentes stratégies, nous les avons fait varier et nous avons tracé les courbes dans l'annexe F.

[12] C'est de cette manière que le partitionnement a été réalisé dans *LibMultiScale* au moment de la reprise de l'existant.

arêtes dans la zone de couplage et la zone normale (Ω^N) [13] et *avec coût supplémentaire de couplage* pour les sommets ou bien pour les arêtes ou bien pour les deux.

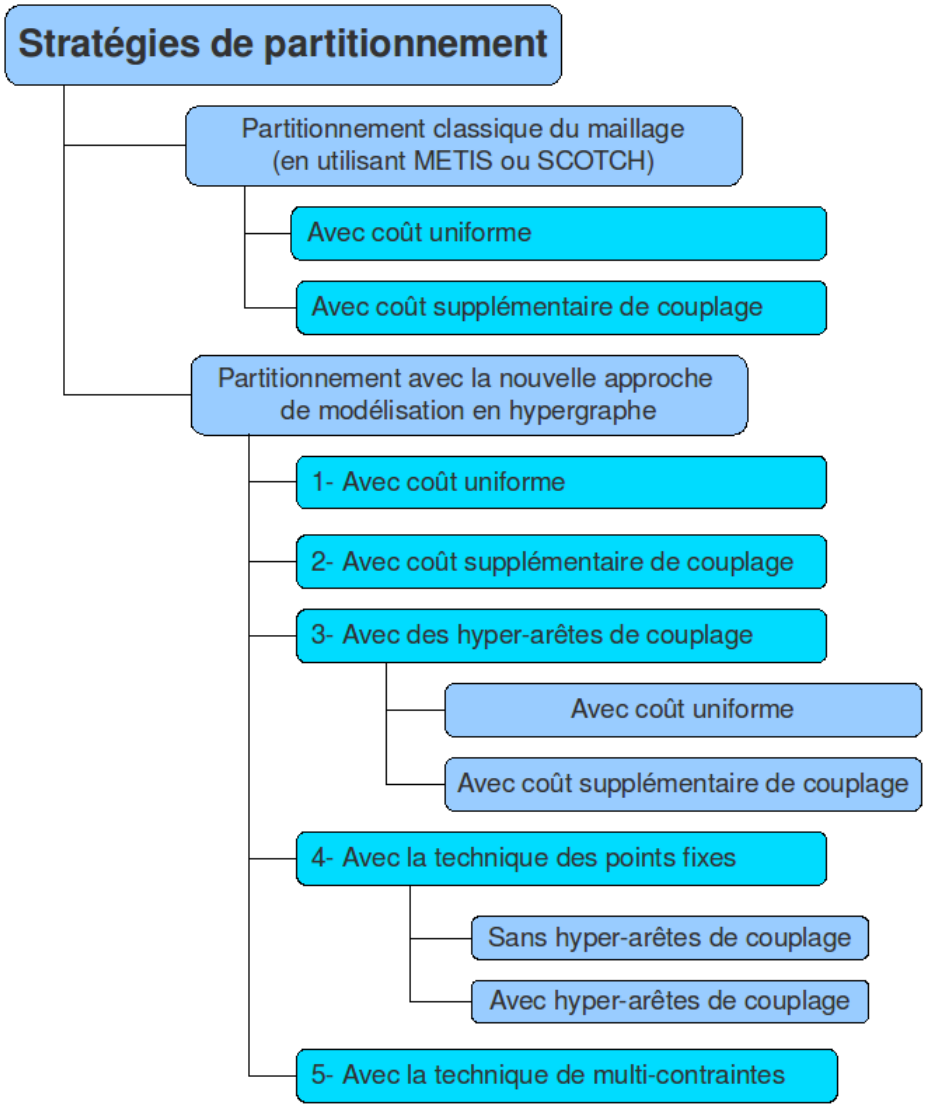


Fig. 4.13: Les différentes stratégies de partitionnement.

Ensuite nous avons partitionné le maillage avec notre *nouvelle approche de modélisation en hypergraphe*. Pour ce partitionnement nous avons procédé de 5 manières différentes.

La première stratégie consiste à partitionner l'hypergraphe *en mettant un coût uniforme* pour les sommets et les hyper-arêtes dans la zone de couplage et la zone normale (Ω^N). Cela va nous permettre de voir l'influence de la modélisation avec un hypergraphe[14] en la comparant avec l'ancienne modélisation en graphe.

La deuxième stratégie consiste à *rajouter un coût supplémentaire de couplage* pour les sommets, ou bien

[13] On appelle la zone normale (Ω^N) la zone qui n'est pas la zone de roucouvrement, i.e. c'est la zone de la mécanique continue Ω^C privée de la zone de recouvrement Ω^R ($\Omega^N = \Omega^C - \Omega^R$).

[14] La connectivitée augmente car selon notre modélisation on va regroupper les éléments voisins dans une même hyper-arête et comme dans la figure 4.9 (b) et non pas simplement en regrouppant deux éléments avec une simple arête comme dans le cas classique avec un graphe

pour les hyper-arêtes ou bien pour les deux, qui sont dans la zone de recouvrement par rapport aux autres de la zone normale.

La troisième stratégie consiste à *rajouter des hyper-arêtes de couplage* en mettant un coût uniforme ou bien avec un coût spplémentaire pour le couplage. On a regroupé, comme dans la figure 4.11 les éléments du maillage de la zone de recouvrement, qui sont dans une même boîte, dans une hyper-arête. Avec cette approche nous avons contraint le partitionneur à essayer de ne pas couper une hyper-arête de couplage car celle-ci aura un coût plus grand que les hyper-arêtes normales[15].

La quatrième stratégie consiste à procéder par *la technique des points fixes*. Cette technique consiste à partitionner l'hypergraphe en deux temps : dans un premier temps, on va partitionner la partie qui représente la zone de recouvrement entre tous les processeurs de la mécanique des milieux continus que nous avons en disposition. Ensuite, dans un deuxième temps, partitionne le reste de l'hypergraphe, avec les mêmes processeurs, en tenant compte du premier partititionnement (c'est à dire en fixant les partitions déjà obtenues de la première étape du partitionnement). Avec cette technique, on est sûr que tous les processeurs auront une charge équilibrée pendant les deux phases de la simulation. Le problème avec cette stratégie est qu'on risque d'avoir des parties qui sont affectées au même processeur mais qui ne sont pas contigûe sur le maillage.

Enfin la cinquième stratégie utilise la technique de *partitionnement multi-contraintes* qui consiste à prendre en compte deux coûts par sommets, celui de la phase 1 et de la phase 2, sans rajouter des hyper-arêtes de couplage cette fois.

4.7 Optimisation

Dans le cas particulier où le nombre de processeurs de la dynamique moléculaire en charge de la zone de recouvrement (i.e. qui participent au couplage) est égal au nombre de processeurs de la mécanique des milieux continus, on va procéder en utilisant le partitionnement par la technique des points fixes en initiant manuellement la première phase pour garantir qu'on obtient le meilleur partitionnement dans cette zone. Le meilleur partitionnement dans ce cas consiste à affecter chaque élément de la zone de couplage à la boite de DM qui le contient comme dans la figure 4.14. Les éléments qui sont géometriquement en intersection avec une boîte seront affectées à un seul processeur de la zone continue.

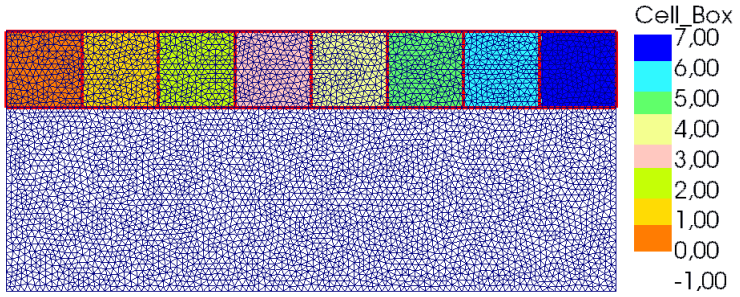


Fig. 4.14: Partitionnement à la main pour garantir un résultat optimal dans le cas où le nombre de processeurs de DM qui participent au couplage correspond au nombre de processeurs de la mécanique continu.

[15] Une hyper-arête normale est une hyper-arête qui regroupe les éléments voisins comme dans la figure 4.9.

5

Description de notre module d'équilibrage de charge

Sommaire

5.1 Description de l'environnement d'exécution du module **49**
5.2 Structures de données utilisées **51**
5.2.1 Structure ***maillage*** 51
5.2.2 Structure ***graphe*** 52
5.2.3 Structure ***hypergraphe*** 52
5.3 Algorithme principale du module d'équilibrage de charge **53**
5.4 Description des autres sous modules **54**
5.4.1 Chargement du maillage en mémoire 54
5.4.2 Construction du graphe à partir du maillage 54
5.4.3 Construction de l'hypergraphe à partir du graphe 54
5.4.4 Sauvegarde du maillage 54
5.4.5 Calcul des statistiques et génération d'un fichier de diagnostique 55

Résumé : Dans ce chapitre nous allons introduire le module que nous avons développé en décrivant schématiquement et algorithmiquement son environnement d'exécution et les entrées/sorties des principales fonctions qui le constituent. Tout le code produit est disponible en ligne sur la geforge de l'*INRIA* à l'adresse : `http://gforge.inria.fr/projects/mpicpl/`.

5.1 Description de l'environnement d'exécution du module

Dans le chapitre 4, nous avons modélisé le maillage de la zone continue Ω^C (figure 2.1) sous forme d'hypergraphe afin d'utiliser les partitionneurs décrits au chapitre 3.

La figure 5.1 montre l'environnement de notre module d'équilibrage de charge. Tout d'abord pour construire le cas test[1], on commence par construire le maillage en utilisant l'outil de génération de maillage gmsh[2] [16] (Annexe A). À la sortie de *GMSH*, on obtient un fichier au format msh qui décrit l'ensemble de la zone continue Ω^C qui inclu également la zone de recouvrement Ω^R (figure 2.1). Le fichier de maillage est lu par notre module qui va détecter la zone de recouvrement et la zone normale, construire l'hypergraphe et rajouter les coûts de calcul sur chaque élément (triangle) et hyper-arête. Ensuite le module partitionne en utilisant les outils décrits dans la section 3.2.2 et finalement génère un fichier de sortie au format msh ou bien au format VTK[3] pour visualiser le résultat de partitionnement.

[1] Se reférer à l'annexe B et à la documentation de *LibMultiScale* [7] pour construire un cas de test.
[2] cf. Glossaire
[3] cf. Glossaire

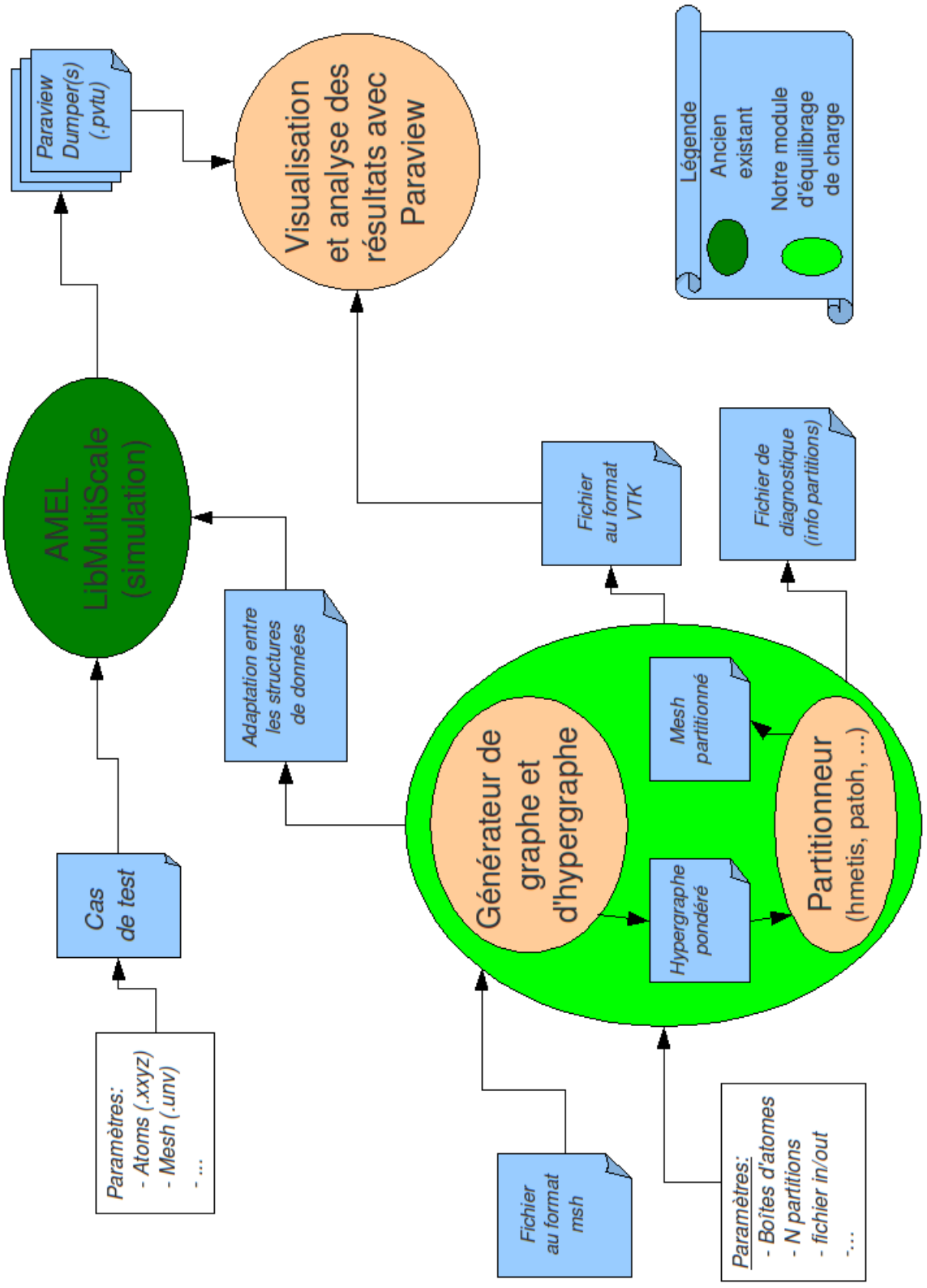


Fig. 5.1: Description de l'environnement d'exécution de notre module.

5.2 Structures de données utilisées

LibMultiScale est codé principalement en *C++*, *Simulpack* qui simule la mécanique des milieux continus et *lammps* qui simule la dynamique moléculaire sont codés en language C. Notre Module d'équilibrage de charge sera codé en langage *C*.

5.2.1 Structure maillage

Généralement les éléments d'un maillage dans le cas 2D ont une géometrie triangulaire, mais nous avons choisi une structure générique pour décrire le maillage quelque soit la forme géometrique des éléments. Cette structure contient les champs suivants :

```
typedef struct Mesh_ {
  int elm_size;
  int nb_nodes;
  int nb_cells;
  double * nodes;
  int * cells;
  int nb_node_vars;
  int nb_cell_vars;
  Variable * node_vars;
  Variable * cell_vars;
} Mesh;
```

elm_size	:	taille d'un élément (nombre de noeuds décrivant un élément)
nb_nodes	:	nombre des noeuds du maillage
nb_cells	:	nombre des éléments du maillage
nodes	:	tableau des coordonnées des noeuds (de taille *nb_nodes x dimension*)
cells	:	tableau des numéros de noeuds formant chaque élément (de taille *nb_cells* $\times$ *elm_size*)
nb_node_vars	:	nombre de variables associées aux noeuds
nb_cell_vars	:	nombre de variables associées aux éléments
node_vars	:	tableau décrivant les variables des noeuds (de taille *nb_node_vars*)
cell_vars	:	tableau décrivant les variables de éléments (de taille *nb_cell_vars*)

Cette structure contient une sous-structure *Variable* qui va évoluer au cours de l'exécution de notre programme afin de décrire les variables associées aux noeuds ou aux éléments du maillage. Nous avons décrit la sous-structure *Variable* de la manière suivante :

```
typedef struct Variable_ {
  char id[255];
  int nb_tuples;
  int nb_components;
  double * data;
} Variable;
```

id	:	identifiant de la variable de taille maximale = 255
nb_tuples	:	nombre de n-uplets
nb_components	:	nombre de composantes par n-uplet
data	:	tableau contenant la description de taille *nb_tuples* $\times$ *nb_components*

5.2.2 Structure graphe

Il existe plusieurs manière pour représenter un graphe [22]. Les deux présentations usuelles de graphes en informatique sont en utilisant un *tableau de listes* ou une *matrice d'adjacence*. Une troisième manière existe (par matrice d'incidence) mais présente généralement peu d'intérêts. Nous avons choisi de représenter notre graphe sous forme de *tableau de listes* car l'espace mémoire occupé par cette représentation est linéaire en fonction du nombre des sommets. En effet, on a intérêt à utiliser une telle représentation car les sommets du graphe modélisent les éléments du maillage et dans les cas réels de modélisation d'un matériau, le maillage contient un grand nombre d'éléments. De plus, cela nous facilitera la tâche pour connaitre rapidement les sommets voisins d'un sommet donné pour pouvoir construire une hyper-arête par la suite quand on construira l'hypergraphe qui modélise le maillage. La structure graphe est une structure intermédiaire qui nous aide pour construire rapidement l'hypergraphe décrite à la section suivante.

Cette structure contient les champs suivants :

```
typedef struct Graph_ {
  int nvtxs;
  int nedges;
  List * adjacency;
} Graph;
```

nvtxs	:	nombre de sommets
nedges	:	nombre d'arrêtes
adjacency	:	liste des sommets voisins (de taille *nvtxs*)

Cette structure contient une sous-structure *List* qui va contenir, comme son nom l'indique, la liste des sommets voisins à un sommet.

```
typedef struct List_ {
  int nb;
  int max;
  int * array;
} List;
```

nb	:	nombre d'éléments
max	:	nombre maximum d'éléments
array	:	tableau d'éléments (de taille *max*)

Pour des raisons d'optimisation et d'utilisation efficace des partitionneurs, nous avons également implanté la structure *graphe CSR* basée sur des listes d'adjacence contigües en mémoire.

5.2.3 Structure hypergraphe

Pour représenter l'hypergraphe, nous avons choisi une structure standard utilisée par la plupart des partitionneurs d'hypergraphe comme *HMETIS* [1] et *PATOH* [20].

Cette structure contient les champs suivants :

```
typedef struct Hypergraph_ {
  int nvtxs;
  int nhedges;
  int * eptr;
  int * eind;
  int * vwgts;
  int * hewgts;
} Hypergraph;
```

nvtxs	:	nombre de sommets
nhedges	:	nombre d'hyper-arêtes
eptr	:	tableau des indirection sur *eind* (de taille *nhedges+1*)
eind	:	tableau de tout les hyperarêtes (comme un ensemble de sommets)
vwgts	:	tableau des poids des sommets (de taille *nvtxs*)
hewgts	:	tableau des poids des hyper-arêtes (de taille *nhedges*)

Un exemple d'hypergraphe représenté avec cette structure est sur la figure 5.2.

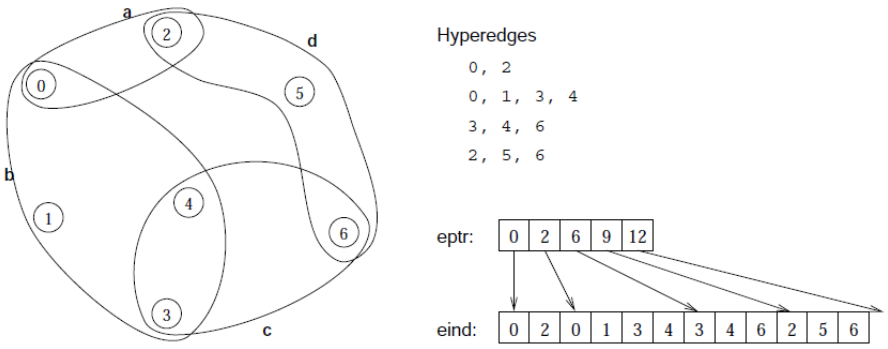


Fig. 5.2: Les tableau *eptr* et *eind* qui décrivent les hyper-arêtes de l'hypergraphe [1].

5.3 Algorithme principale du module d'équilibrage de charge

La fonction principale qui se trouve dans le fichier *libmultiscale-gen.c* est décrite par l'algorithme 2. Cette fonction, en s'appuyant sur les différents sous-modules, réalise le partitionnement du maillage d'éléments finis(représenté par un hypergraphe) en *nbparts* parties. On suppose que l'on dispose d'une grille $P \times Q$ boîtes qui cadrille la zone de recouvrement. En outre, il produit des sorties sous forme de diagnostiques et de fichiers de maillage.

Algorithme 2 Algorithme principale du module équilibrage de charge

Entrées: maillage.msh : fichier ; nbparts, P, Q :entier

Sortie: maillagePartitionné.vtk, maillagePartitionné.msh, diagnostique.txt : fichiers

```
init(structMeshFE, structMeshMD, structGraph, structHyperGraph)
structMeshFE ← chargerMaillage(maillage.msh, structMeshFE)
structMeshFE ← detecterZoneRecouvrement(structMeshFE)
structMeshMD ← constructionBoitesAtomes(structMeshFE, P, Q)
structMeshFE ← ajouterVariablesDivers(structMeshFE)
structMeshMD ← ajouterVariablesDivers(structMeshMD)
structGraph ← générerGraph(structMeshFE)
structHyperGraph ← générerHypergraph(structGraph)
partitions ← partitionnerHypergraph(structHyperGraph, nbparts)
structMeshFE ← ajouterVariablesPartitions(structMeshFE, partitions)
diagnostique.txt ← calculDiagnostique(structMeshFE, structMeshMD, P, Q, nbparts)
maillagePartitionné.vtk ← sauvgarderVTK(structMeshFE)
maillagePartitionné.msh ← sauvgarderMSH(structMeshFE)
```

5.4 Description des autres sous modules

5.4.1 Chargement du maillage en mémoire

Comme son nom l'indique, ce module va permettre de charger en mémoire le maillage qui est décrit dans un fichier d'entrée au format *msh*. On va faire une boucle de lecture de ce fichier ligne par ligne et on remplira au fure et à mesure la structure *Mesh*. Pour la lecture du fichier, on se base sur le standard du format *msh* décrit dans [10].

Le maillage est décrit dans un fichier au format *msh* à l'entrée de notre module. Selon les normes de ce format [10] (cf. annexe A), un maillage est décrit de la manière suivante : tout d'abord on décrit tout les noeuds du maillage avec leur coordonnées dans une première section. Ensuite, dans une seconde section, on décrit les cellules (triangles dans notre cas) du maillage en indiquant les noeuds qui le forment.

5.4.2 Construction du graphe à partir du maillage

Pour construire le graphe intermédiaire représentant chaque élément du maillage avec ses voisins, on utilise l'algorithme 3 de la fonction *mesh2Graph*. Ce graphe sera utilisé par la suite dans la fonction *generateHypergraph* pour construire l'hypergraphe. Le rôle de la fonction *créerListeCellulesParNoeud* utilisée dans l'algorithme 3 est de construire (un "dual" de la description du maillage) la liste des éléments (triangles) qui sont attachés à chaque noeud comme c'est expliqué dans l'exemple de la figure 5.3. *cellulesParNoeud[i]* est la liste (type *List*) des indices des cellules partageant ce noeud. Cette structure est ensuite utilisée pour construire le graphe représentant le maillage.

Algorithme 3 Algorithme de la fonction de construction de graphe à partir du maillage

Entrées: mesh : Mesh
Sortie: graph : Graph

```
cellulesParNoeud ← créerListeCellulesParNoeud(mesh)
Pour tout les cellules i du maillage Faire
  Pour toute arrête e = (n1 , n2) de la cellule i Faire
    Si ∃ j tel que j ∈ cellulesParNoeud[n1] ∩ cellulesParNoeud[n2] Alors
      graph->ListVoisins[i] ← graph->ListVoisins[i] ∪ j
    FinSi
  FinPour
FinPour
```

5.4.3 Construction de l'hypergraphe à partir du graphe

En se basant sur le graphe obtenu par la fonction *mesh2Graph*, décrit précédement, on construit un hypergraphe utilisant le principe expliqué dans la figure 5.2. La fonction permettant de construire l'hypergraphe est la fonction *generateHypergraph* décrite par l'algorithme 4.

5.4.4 Sauvegarde du maillage

Après le partitionnement du maillage, il faut le sauvegarder dans un format convenable pour la visualisation des résultats. Ce module permet justement de générer un fichier décrivant le même maillage chargé au préalable par le module de chargement, mais en lui rajoutant les informations nécessaires pour décrire la répartition des données entre les processeurs. Nous utiliserons pour se faire le format de fichier *VTK* et l'outil *ParaView* pour afficher ce fichier. (cf. chapitre 6 pour les résultats produits).

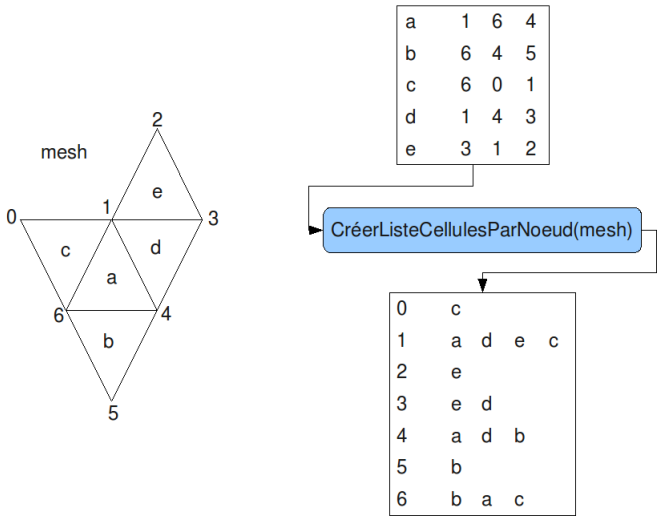


Fig. 5.3: Construction de la liste des cellules par noeud (structure duale).

5.4.5 Calcul des statistiques et génération d'un fichier de diagnostique

Afin de pouvoir analyser les résultats précisément, nous avons développé une fonction qui se charge de faire les diagnostiques et l'analyse du partitionnement obtenu. Ce fichier contient une section par stratégie. Chaque section contient des coefficients de qualité que nous avons construit pour évaluer la qualité du partitionnement les détails pour chaque processeur. Un exemple de fichier de diagnostique est fourni en annexe E. Ce fichier pourra être amélioré en fonction de ce qu'on veut mesurer.

Algorithme 4 Algorithme de la fonction de construction de l'hypergraphe à partir du graphe

Entrées: g : Graph ; mesh : maillage ; coûtHeCouplage, coûtHeNormale, poidsNormal, poidsCouplage : entier
Sortie: hg : Hypergraph

```
Pour tous les sommets v de g Faire
  hg ← ajouterHyperarêteNormale(v)
FinPour
Pour tous les boites d'atomes en relation avec la zone de couplage de mesh Faire
  hg ← ajouterHyperarêteCouplage(mesh)
FinPour
Pour tous les hyperarêtes nHe normale Faire
  nHe->coût ← coûtHeNormale
FinPour
Pour tous les hyperarêtes cHe de couplage Faire
  nHe->coût ← coûtHeCouplage
FinPour
Pour tous les sommet v de hg Faire
  Si v se trouve dans la zone de couplage Alors
    v->poids ← poidsCouplage
  Sinon
    v->poids ← poidsNormal
  FinSi
FinPour
```

Quatrième partie

Résultats et Perspectives

6

Résultats

Sommaire

6.1 Cas d'une géometrie simple **61**
6.1.1 Description du maillage 61
6.1.2 Partitionnement classique du maillage en le modélisant par un graphe 61
6.1.3 Partitionnement du maillage en le modélisant par un hypergraphe classique . . 63
6.1.4 Partitionnement du maillage en le modélisant par un hypergraphe avec des hyper-arêtes de couplage 66
6.1.5 Partitionnement du maillage en le modélisant par un hypergraphe avec technique des points fixes 68
6.1.6 Partitionnement du maillage en le modélisant par un hypergraphe avec la technique de multi-contraintes 70
6.2 Cas d'une géometrie plus complexe **71**
6.2.1 Description du maillage 71
6.2.2 Partitionnement classique du maillage en le modélisant par un graphe 71
6.2.3 Partitionnement du maillage en le modélisant par un hypergraphe classique . . 74
6.2.4 Partitionnement du maillage en le modélisant par un hypergraphe avec des hyper-arêtes de couplage 77
6.2.5 Partitionnement du maillage en le modélisant par un hypergraphe avec technique des points fixes 80
6.2.6 Partitionnement du maillage en le modélisant par un hypergraphe avec la technique de multi-contraintes 83
6.3 Mise en évidence des intérêts et limitations des hyper-arêtes de couplage **83**
6.4 Conclusion **84**

Résumé : Dans ce chapitre, nous allons décrire les résultats que nous avons obtenu en se basant sur les différentes stratégies que nous avons considérées pour le partitionnement du maillage et que nous avons présentées dans la section 4.6. Nous allons faire une comparaison entre le cas test de référence (voir section 4.2, page 32) et les résultats que nous avons obtenus.

Pour simplifier les notations, on affecte une étiquette pour chaque stratégie suivie selon les abréviations bleues de la figure 6.1.

On présentera les résultats des mesures dans des tableaux ayant une entête comme celle ci :

Stratégie	figure	coût Ha			poids S		Equilibrage du poids		Communication	
		HaC	N	B	N	B	ph1	ph2	FE-FE	FE-MD

où *"Stratégie"* désigne la stratégie suivie identifiée par l'une des étiquettes de la figure 6.1, *"figure"* est une référence vers la figure illustrant le résultat obtenu. *"coût Ha"* et *"poids s"* désignent respectivement les coûts des hyper-arêtes et les poids des sommets que nous avons choisi pour chaque expérience ; *"HaC"* désigne le coût sur une hyper-arête de couplage[1] ; *"N"* désigne le coût/poids de l'hyper-arête/arête/sommet dans la zone *normale* du maillage, c'est-à-dire la zone qui se situe en dehors de la zone de recouvrement

[1] Comme c'est expliqué sur la figure 4.11

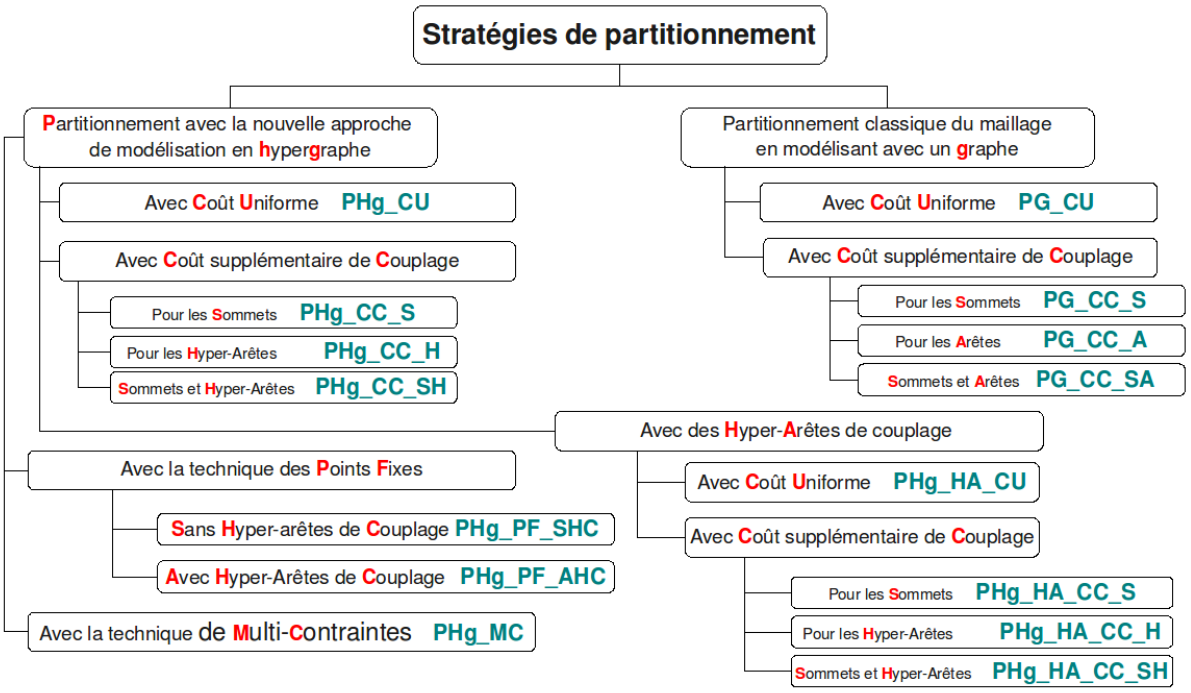


Fig. 6.1: Étiquettes pour distinguer les différentes stratégies.

"B" désigne le coût/poids de l'hyper-arête/arête/sommet dans la zone de recouvrement Ω^R. *"Équilibrage du poids"* et *"Communication"* désignent les résultats de mesure que nous avons calculé, où *"ph1"* et *"ph2"* désignent respectivement la 1ere phase et la 2eme phase de calcul. *"FE-FE"* désigne les résultats de mesure des communications entre les processeurs de la mécanique des milieux continus et *"FE-MD"* désigne les résultats de mesure des communications entre les processeurs de la mécanique des milieux continus d'une part et les processeurs de la dynamique moléculaire d'autre part.

Pour les mesures de l'équilibrage du poids et des communications, on aura 8 valeurs à récupérer pour chaque mesure comme c'est expliqué dans le tableau suivant :

Equilibrage du poids		Communication	
ph1	ph2	FE-FE	FE-MD
m1 , σ1	m2 , σ2	min , max	min , max

où *"m1"* et *"m2"* sont respectivement les moyennes (en % du poids total) des tailles des parties de la phase 1 et de la phase 2 affectées aux processeurs de FE et σ1 et σ2 sont, respectivement, leurs écart-types. Pour les communications FE-FE, *"min"* et *"max"* désignent respectivement les nombres min et max de communications entre les processeurs de FE. Pour les communications FE-MD, *"min"* et *"max"* désignent respectivement les nombres min et max de communications entre un processeur de FE et de MD.

Nous allons commencer par analyser les résultats obtenus dans le cas d'une géometrie simple dans la section 6.1. Ensuite nous allons étudier dans la section 6.2 le cas d'une géometrie 2D plus complexe.

Pour chaque famille de stratégie, nous allons faire varier les poids des sommets et les coûts des arêtes

ou hyper-arêtes afin de voir le comportement de notre module d'équilibrage de charge et pour déterminer les meilleurs valeurs qui permettent d'obtenir les meilleurs résultats. Les résultats des diagnostiques pour les variations des poids sont en annexe F.

6.1 Cas d'une géometrie simple

Nous étudions le partitionnement dans le cas d'une géometrie simple en prenant autant de processeurs que de boîtes d'atomes, soit 8 ici.

6.1.1 Description du maillage

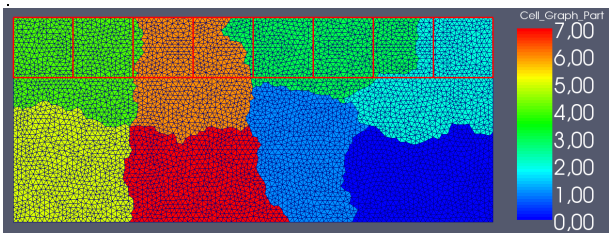
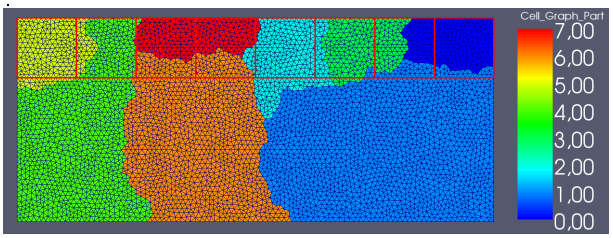
Nous avons généré, à l'aide de *GMSH* [10], le maillage de la figure C.1 avec la description d'un fichier au format *geo* [2] qui est en annexe C.

6.1.2 Partitionnement classique du maillage en le modélisant par un graphe

Stratégie	figure (tableau)	coût A		poids S		Equilibrage du poids		Communication	
		N	B	N	B	ph1	ph2	FE-FE	FE-MD
PG_CU	6.2 (a)	1	1	1	1	12,5 ; 0	12,5 ; 13,5	3 ; 6	0 ; 4
PG_CC_S	6.2 (b)	1	1	1	10	12.5 ; 0,01	12,5 ; 3,8	2 ; 5	2 ; 4
PG_CC_A	6.2 (c)	1	10	1	1	12,5 ; 0,00	12,5 ; 12,7	3 ; 6	0 ; 3
PG_CC_SA	6.2 (d)	1	10	1	10	12.5 ; 0,02	12,5 ; 4,8	3 ; 8	0 ; 3

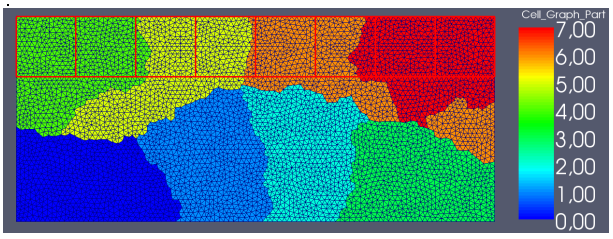
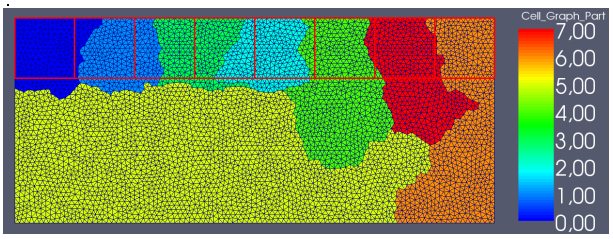
Tab. 6.1: Diagnostique du partitionnement classique du maillage d'une géometrie simple en le modélisant par un graphe.

[2] Le format geo est le format d'entré pour *GMSH* (voir annexe A)

Ref 6.2	Résultat	Commentaires
(a)	 **Avec un coût uniforme sur la zone normale et la zone de recouvrement (PG_CU).** % répart ph1 : [12 13 13 13 12 13 13 13] % répart ph2 : [0 0 17 36 26 0 21 0] com FE-FE : [3 6 4 4 3 3 5 4] com FE-MD : [0 0 2 4 3 0 2 0] En mettant le même coût sur les sommets et arêtes du graphe de la zone de recouvrement et de la zone normale, on constate que bien que les parties semblent être équilibrées d'un point de vue de la charge totale (en moyenne $12,5$ avec un écart-type faible, 0, dans ce cas), celle-ci ne sont pas équilibrés entre les deux phases de calcul. En effet, l'écart-type calculé entre les charges affectées aux processeurs pendant la phase 2 et de $13,5$. L'écart-type est exprimé en % du poids total.	Les processeurs qui n'ont pas de charge dans la zone de recouvrement auront un temps d'inactivité pendant la phase 2 de couplage. Le partitionneur ne tient pas en compte des deux phases si on ne met pas de coût supplémentaire sur les élément de la zone de recouvrement. **Cette configuration correspond aux coûts réels de LibMultiScale**. En effet, comme on a vu dans la section 4.5, les coûts de calculs réels ne permettent pas de mettre en évidence le surcoût de couplage et on donc si on se base sur ces coûts réels on doit mettre un coût uniforme sur tous les sommets et les arêtes.
(b)	 **Avec un poids supplémentaire de couplage pour les sommets (PG_CC_S).** % répart ph1 : [13 12 12 12 12 13 12 13] % répart ph2 : [15 5 15 15 10 15 9 15] com FE-FE : [3 5 5 4 4 2 5 4] com FE-MD : [2 3 3 2 2 2 3 4] En mettant un poids supplémentaire sur les sommets de la zone de recouvrement (ici on a mis 10 fois le coût normal), on constate que tous les processeurs participent à la 2eme phase et que l'écart-type pour cette phase a diminué par rapport à la stratégie PG_CU (3,8 dans ce cas).	Ça reste quand même déséquilibré entre les deux phases. En effet le processeur 1 a très peu de charge pendant la 2eme phase et aura un temps d'inactivité considérable par rapport aux autres processeurs, tandis que le processeur 7, qui est en charge d'une région totalement incluse dans la zone de recouvrement, ralentira les autres processeurs ayant moins de charge de couplage (comme les processeurs 1 et 4).

... suite à la page suivante ...

... suite de la page précédente ...

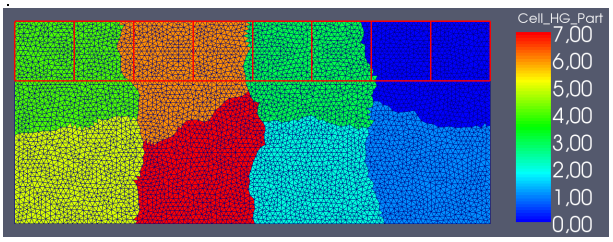
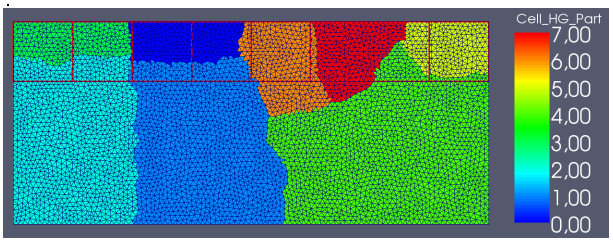
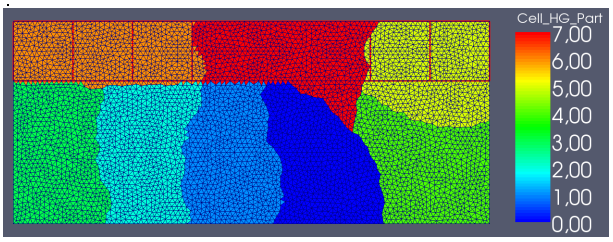
Ref 6.2	Résultat	Commentaires
(c)	 **Avec un coût supplémentaire de couplage pour les arêtes (PG_CC_A).** % répart ph1 : [12 13 13 13 13 13 12 13] % répart ph2 : [0 0 0 0 27 22 22 29] com FE-FE : [4 4 5 4 3 6 5 3] com FE-MD : [0 0 0 0 3 2 3 3]	En mettant un coût supplémentaire sur les arêtes de la zone de recouvrement (ici on a mis 10 fois le coût normal), les résultats ne s'améliorent pas car on constate que l'écart-type pour la moyenne de la charge de calcul pendant la 2eme phase augmente (ici $12,7$) ce qui signifie que certains processeurs ont très peu (ou bien n'ont pas de tout) de charge pendant cette phase. On peut le constater d'après la figure (c) où les processeurs 0, 1, 2 et 3 n'ont pas de charge sur la zone de couplage.
(d)	 **Avec un coût supplémentaire de couplage pour les sommets et les arêtes(PG_CC_SA).** % répart ph1 : [12 13 13 12 13 12 12 13] % répart ph2 : [15 15 15 15 13 0 12 14] com FE-FE : [3 4 4 4 4 8 3 4] com FE-MD : [2 2 2 2 3 0 2 3]	Si on met maintenant un coût supplémentaire sur les sommets et les arêtes en même temps, on constate que les résultats ne s'améliorent pas vraiment. Bien que l'ecart-type ait diminué ça reste énorme (ici $4,8$). En effet, on constate que le processeur 6 n'a quasiment pas de charge sur la zone de couplage (ph2). Le partitionneur n'a donc équilibré que la charge totale.

Tab. 6.2: Résultats de partitionnement classique du maillage d'une géometrie simple en le modélisant par un graphe.

6.1.3 Partitionnement du maillage en le modélisant par un hypergraphe classique

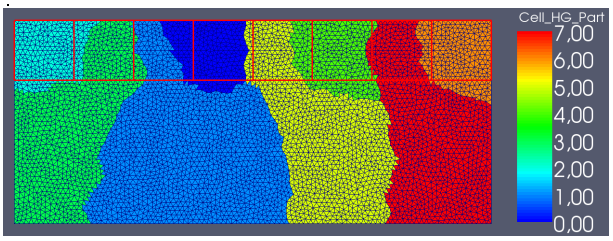
Stratégie	figure	coût Ha			poids S		Equilibrage du poids		Communication	
		cHe	N	B	N	B	ph1	ph2	FE-FE	FE-MD
PHg_CU	6.4 (a)	0	1	1	1	1	12.5 ; 0.02	12.5 ; 12	2 ; 5	0 ; 4
PHg_CC_S	6.4 (b)	0	1	1	1	10	12.5 ; 0.01	12.5 ; 3.6	2 ; 4	2 ; 4
PHg_CC_H	6.4 (c)	0	1	10	1	1	12.5 ; 0.04	12.5 ; 16	2 ; 6	0 ; 4
PHg_CC_SH	6.4 (d)	0	1	10	1	10	12.5 ; 0.02	12.5 ; 2.8	1 ; 4	2 ; 3

Tab. 6.3: Diagnostique du partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe.

Ref 6.4	Résultat	Commentaires
(a)	 **Avec un coût uniforme sur la zone normale et la zone de recouvrement (PHg_CU).** % répart ph1 : [13 13 13 13 12 12 12 12] % répart ph2 : [27 0 0 25 23 0 26 0] com FE-FE : [2 3 3 5 2 3 4 4] com FE-MD : [3 0 0 4 2 0 3 0]	**Cette configuration correspond aux coûts réels de LibMultiScale** car on met des coûts uniformes sur les sommets et hyper-arêtes de l'hypergraphe. On obtient un résultat semblable à PG_CU. En effet on constate ici que les processeurs 1, 2, 5 et 7 n'ont pas de charge sur la zone de recouvrement et auront donc un temps d'inactivité pendant la phase de couplage. Ces processeurs n'ont naturellement pas de communications avec les processeurs de la dynamique moléculaire, c'est pour cela la valeur minimale de communication FE-MD est nulle.
(b)	 **Avec un poids supplémentaire de couplage pour les sommets (PHg_CC_S).** % répart ph1 : [13 12 12 13 12 13 12 13] % répart ph2 : [15 9 9 15 6 15 15 15] com FE-FE : [4 4 3 2 4 2 4 3] com FE-MD : [3 4 2 2 2 2 3 2]	En mettant un poids supplémentaire sur les sommets de l'hypergraphe, on constate que tous les processeurs participent au calcul de couplage car la valeur minimale de communication FE-MD n'est pas nulle (ici 2). La répartition reste déséquilibrée pendant la 2eme phase car l'écart type à la moyenne est de $3,6$.
(c)	 **Avec un coût supplémentaire de couplage pour les hyper-arêtes (PHg_CC_H).** % répart ph1 : [12 12 12 13 12 13 12 12] % répart ph2 : [0 0 0 0 0 26 39 35] com FE-FE : [3 3 4 2 3 2 3 6] com FE-MD : [0 0 0 0 0 3 4 4]	En mettant un coût supplémentaire sur les hyper-arêtes de l'hypergraphe, on constate qu'il y a plus de déséquilibre entre les parties (ici l'écart type est de 16 pour la phase 2) et que les processeurs 0, 1, 2, 3 et 4 ne participent pas au couplage, ce qui explique la valeur minimale de communication FE-MD qui est nulle ici.

... suite à la page suivante ...

... suite de la page précédente ...

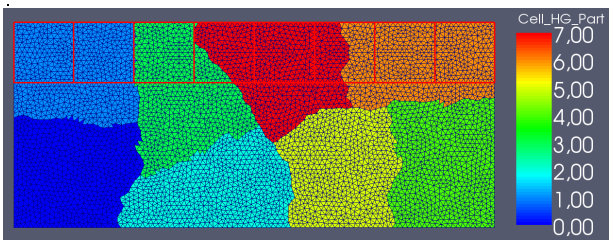
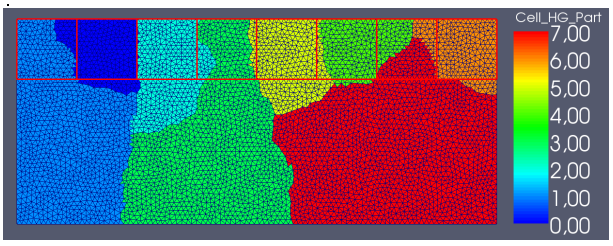
Ref 6.4	Résultat	Commentaires
(d)	.  **Avec un coût supplémentaire de couplage pour les sommets et les hyper-arêtes(PHg_CC_SH).** % répart ph1 : [12 12 12 12 13 13 13 13] % répart ph2 : [15 7 15 11 15 10 15 11] com FE-FE : [2 3 1 2 2 4 1 3] com FE-MD : [2 2 2 2 3 2 2 2]	En rajoutant un coût supplémentaire sur les sommets et hyper-arêtes de la zone de couplage, on constate que tous les processeurs participent au couplage mais la répartition n'est pas bien équilibrée pour la phase 2 (écart-type de $2,8$). En effet le processeur 1, par exemple, n'est en charge que de 7% de la zone de couplage alors que le processeur 2 a 15% de charge pendant la seconde phase.

Tab. 6.4: Résultats de partitionnement du maillage d'une géometrie simple en le modélisant par un hyper-graphe.

6.1.4 Partitionnement du maillage en le modélisant par un hypergraphe avec des hyper-arêtes de couplage

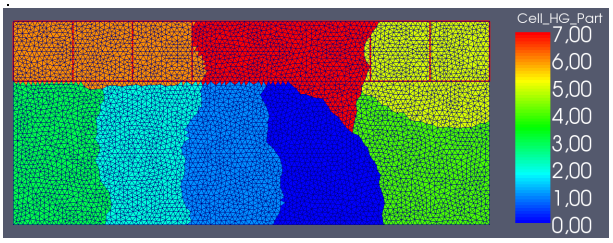
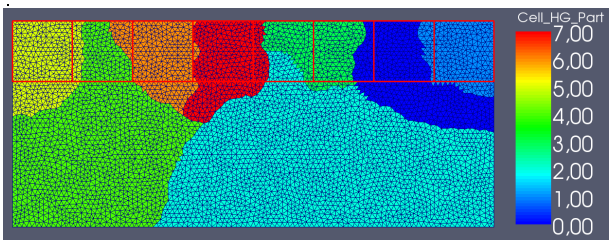
Stratégie	figure	coût He			poids v		Equilibrage du poids		Communication	
		cHe	N	B	N	B	ph1	ph2	FE-FE	FE-MD
PHg_HA_CU	6.6 (a)	50	1	1	1	1	12.5 ; 0.03	12.5 ; 13	2 ; 4	0 ; 3
PHg_HA_CC_S	6.6 (b)	50	1	1	1	10	12.5 ; 0.09	12.5 ; 3.3	2 ; 4	1 ; 3
PHg_HA_CC_H	6.6 (c)	50	1	10	1	1	12.5 ; 0.05	12.5 ; 16	2 ; 5	0 ; 4
PHg_HA_CC_SH	6.6 (d)	50	1	10	1	10	12.5 ; 0.15	12.5 ; 4.2	1 ; 4	1 ; 3

Tab. 6.5: Diagnostique du partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe et en rajoutant des hyper-arêtes des couplage.

Ref 6.6	Résultat	Commentaires
(a)	 **Avec un coût uniforme sur la zone normale et la zone de recouvrement (PHg_HA_CU).** % répart ph1 : [12 13 13 12 13 13 12 13] % répart ph2 : [0 25 0 15 0 0 32 28] com FE-FE : [3 2 4 4 2 4 3 4] com FE-MD : [0 2 0 2 0 0 3 3]	**Cette configuration correspond aux coûts réels de LibMultiScale** car on met des coûts uniformes sur les sommets et hyper-arêtes de l'hypergraphe. En rajoutant des hyper-arêtes de couplage, on constate que le partitionneur a tendance à ne pas vouloir les couper. En effet, la coupe entre les processeurs 1 et 3 a suivi la forme de la boite dans la zone de recouvrement. Par contre la répartition n'est pas équilibrée car le processeur 3 n'a que 15% de charge pendant la phase 2 alors que les processeurs 1, 6 et 7 ont plus que 25%. Le processeur 6 qui a 32% de charge pendant le couplage va ralentir les autres. On remarque aussi que les processeurs 0, 2, 4 et 5 n'ont pas de charge pendant cette phase.
(b)	 **Avec un coût supplémentaire de couplage pour les sommets (PHg_HA_CC_S).** % répart ph1 : [12 12 12 13 13 12 13 12] % répart ph2 : [15 10 14 9 15 15 15 6] com FE-FE : [2 3 3 4 3 3 2 4] com FE-MD : [2 1 2 1 2 3 2 1]	En rajoutant un coût supplémentaire sur les sommets avec des hyper-arêtes de couplage, on n'obtient pas un bon partitionnement. En effet, la charge n'est pas bien répartie pendant la phase de couplage car on obtient un écart type de $3,3$. Cependant il s'agit du meilleur résultat pour cette série.

... suite à la page suivante ...

... suite de la page précédente ...

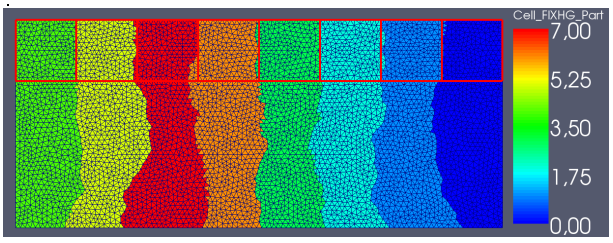
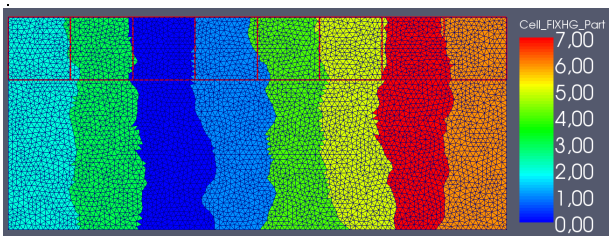
Ref 6.6	Résultat	Commentaires
(c)	 **Avec un coût supplémentaire de couplage pour les hyper-arêtes normales (PHg_HA_CC_H).** % répart ph1 : [12 12 12 12 13 13 13 13] % répart ph2 : [0 0 0 0 0 27 39 34] com FE-FE : [3 4 3 2 3 2 4 5] com FE-MD : [0 0 0 0 0 3 4 3]	En rajoutant un coût supplémentaire sur les hyper-arêtes avec des hyper-arêtes de couplage, on obtient un mauvais partitionnement. En effet, il y aura 5 processeurs sur 8 qui sont inactifs pendant la phase de couplage. La charge n'est pas bien répartie pendant la phase de couplage car on obtient un énorme écart type de 16.
(d)	 **Avec un coût supplémentaire de couplage pour les sommets et les hyper-arêtes normales (PHg_HA_CC_SH).** % répart ph1 : [12 12 13 13 13 12 12 13] % répart ph2 : [14 15 3 15 9 15 15 15] com FE-FE : [3 1 4 3 4 1 2 4] com FE-MD : [2 2 1 2 1 2 3 2]	En rajoutant un coût supplémentaire sur les sommets et les hyper-arêtes (10 fois plus) avec des hyper-arêtes de couplage (50 fois plus), la charge n'est pas bien bien répartie sur la phase 2 et il y a des processeurs qui ont beacoup plus de charges que d'autres sur cette phase et qui vont les ralentir pendant cette phase. Ce résultat est néanmoins comparable au cas (b), le meilleur résultat pour cette série.

Tab. 6.6: Résultats de partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe et en rajoutant des hyper-arêtes des couplage.

6.1.5 Partitionnement du maillage en le modélisant par un hypergraphe avec technique des points fixes

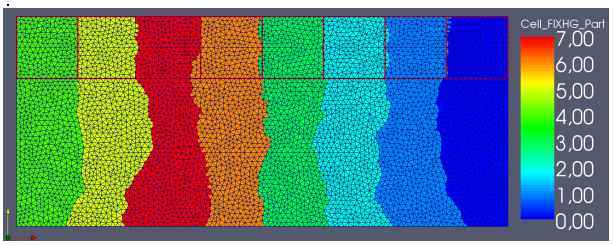
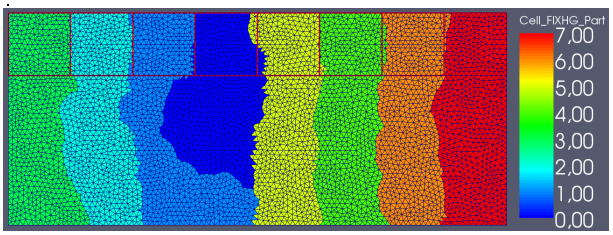
Stratégie	figure	coût He			poids v		Equilibrage du poids		Communication	
		cHe	N	B	N	B	ph1	ph2	FE-FE	FE-MD
PHg_PF_SHC	6.8 (a)	0	1	1	1	1	12.5 ; 0.04	12.5 ; 0.14	1 ; 2	1 ; 2
PHg_PF_SHC	6.8 (b)	0	1	10	1	10	12.5 ; 0.12	12.5 ; 0.02	1 ; 2	2 ; 3
PHg_PF_AHC	6.8 (c)	50	1	1	1	1	12.5 ; 0.04	12.5 ; 0.14	1 ; 2	1 ; 2
PHg_PF_AHC	6.8 (d)	50	1	10	1	10	12.5 ; 0.08	12.5 ; 0.06	1 ; 3	1 ; 3

Tab. 6.7: Diagnostique du partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe en utilisant la technique des points fixes.

Ref 6.8	Résultat	Commentaires
(a)	 **Avec un coût supplémentaire de couplage pour les sommets et sans hyper-arêtes de couplages (PHg_PF_SHC).** % répart ph1 : [13 13 12 13 12 12 13 13] % répart ph2 : [13 13 13 13 12 12 12 12] com FE-FE : [1 2 2 2 1 2 2 2] com FE-MD : [1 2 2 2 1 1 2 2]	**Cette configuration correspond aux coûts réels de LibMultiScale** car on met des coûts uniformes sur les sommets et hyper-arêtes de l'hypergraphe. On obtient un bon résultat ici. En effet, les parties sont bien équilibrée et l'écart type est faible pour les deux phases ($0,04$ pour la phase 1 et $0,14$ pour la phase 2). Du point de vue de la communication aussi, on obtient un bon résultat. En effet, chaque processeur de FE communique avec 1 à 2 processeur de FE et avec 1 à 2 processeurs de MD aussi.
(b)	 **Avec un coût supplémentaire de couplage pour les sommets et les hyper-arêtes et sans hyper-arêtes de couplages (PHg_PF_SHC).** % répart ph1 : [13 13 13 13 12 12 12 12] % répart ph2 : [13 12 13 12 12 13 13 12] com FE-FE : [2 2 1 2 2 2 1 2] com FE-MD : [2 3 2 3 3 3 2 2]	En rajoutant des coûts supplémentaires sur les sommets et hyper-arêtes de la zone de recouvrement, On obtient un bon résultat aussi. En effet, les parties sont bien équilibrées et l'écart-type est faible pour les deux phases avec une légère amélioration d'équilibrage pour la phase 2 puisque l'écart-type diminue ($0,02$ pour la phase 2) mais ça reste comparable au résultat (a). Du point de vue de la communication aussi, on obtient un bon résultat. En effet, chaque processeur de FE communique avec 1 à 2 processeurs de FE (comme le cas précédent) et avec 2 à 3 processeurs de MD (avec une légère dégénéréscence par rapport au cas précédent).

... suite à la page suivante ...

... suite de la page précédente ...

Ref 6.8	Résultat	Commentaires
(c)	 **Avec un coût uniforme pour les sommets et les hyper-arêtes et avec hyper-arêtes de couplages (PHg_PF_AHC).** % répart ph1 : [13 13 12 13 12 12 13 13] % répart ph2 : [13 13 13 13 12 12 12 12] com FE-FE : [1 2 2 2 1 2 2 2] com FE-MD : [1 2 2 2 1 1 2 2]	**Cette configuration correspond aux coûts réels de LibMultiScale** car on met des coûts uniformes sur les sommets et hyper-arêtes de l'hypergraphe. On obtient un bon résultat ici comme dans le cas (a). On constate donc que les hyper-arêtes de couplage n'ont pas un rôle significatif ici bien qu'elles permettent de guider le partitionneur dans certains cas.
(d)	 **Avec un coût supplémentaire de couplage pour les sommets et les hyper-arêtes et avec hyper-arêtes de couplages (PHg_PF_AHC).** % répart ph1 : [12 13 13 12 12 12 12 13] % répart ph2 : [12 13 12 13 13 12 13 12] com FE-FE : [2 3 2 1 2 3 2 1] com FE-MD : [3 2 1 2 2 3 2 1]	Les parties sont équilibrées aussi en rajoutant des coûts supplémentaires sur les sommets et hyper-arêtes de la zone de recouvrement, l'écart type est faible pour les deux phases avec une légère amélioration pour l'équilibrage de la phase 1 puisque l'écart type diminue ($0,12$ pour la phase 1 sans hyper-arête de couplage et $0,08$ avec hyper-arêtes de couplage) mais ça reste comparable au résultat (b). Du point de vue de la communication, chaque processeur de FE communique avec 1 à 3 processeur de FE et avec 1 à 3 processeurs de MD aussi.

Tab. 6.8: Résultats de partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe en utilisant la technique des points fixes.

6.1.6 Partitionnement du maillage en le modélisant par un hypergraphe avec la technique de multi-contraintes

Stratégie	figure	Equilibrage du poids		Communication	
		ph1	ph2	FE-FE	FE-MD
PHg_MC	6.2	12.5 ; 0.22	12.5 ; 0.27	4 ; 6	2 ; 6

Tab. 6.9: Diagnostique du partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe en utilisant la technique de multi-contraintes (PHg_MC).

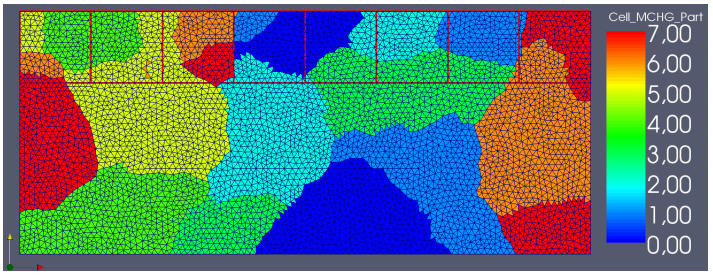


Fig. 6.2: Résultat de partitionnement du maillage d'une géometrie simple en le modélisant par un hypergraphe en utilisant la technique de multi-contraintes (PHg_MC).

% répart ph1 : [12 12 12 12 13 12 13 13]
% répart ph2 : [12 12 12 13 12 12 13 13]
com FE-FE : [4 5 6 5 5 4 5 6]
com FE-MD : [2 4 4 3 2 3 6 3]

Avec cette stratégie, on impose au partitionneur d'équilibrer la charge pour les deux phases, ce qui est bien garantit de ce point de vue là. En effet, dans l'exemple de la figure 6.2 chaque processeur a en moyenne $12,5\%$ de charge de calcul pendant la 1ere phase avec un écart type de $0,22$ et $12,5\%$ de charge pendant la 2eme phase avec un écart type de $0,27$. Mais on ne peut pas garantir que les parties affectées au même processeur soient connexes, ce qui augmente les communications de manière très remarquable. En effet, chaque processeur de FE communique avec 4 à 6 processeurs de FE et avec 2 à 6 processeurs de MD.

6.2 Cas d'une géometrie plus complexe

On a étudié dans la section 6.1 les résultats que nous avons obtenu dans le cas d'une géometrie simple. On va voir dans cette section ce que nous avons obtenu dans le cas d'une géometrie plus compliquée. Nous partitionnons le maillage en 8 et utilisons une grille de 3×3 boîtes d'atomes telles que 8 boîtes soient présentes dans la zone de couplage (avec un nombre de triangle variables).

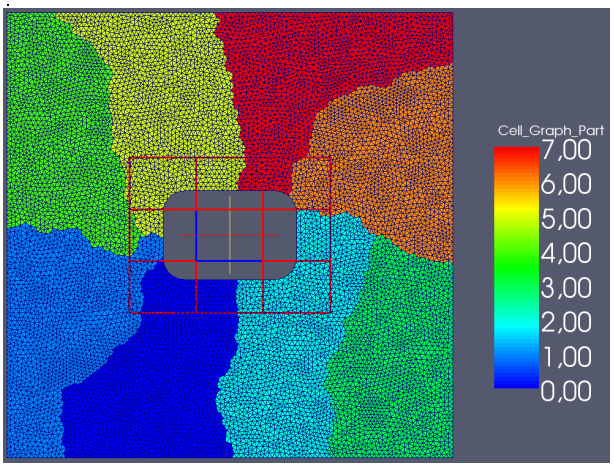
6.2.1 Description du maillage

Nous avons généré, à l'aide de *GMSH* [10], le maillage de la figure D.3 avec la description d'un fichier au formart *geo* qu'on a mis en annexe D.

6.2.2 Partitionnement classique du maillage en le modélisant par un graphe

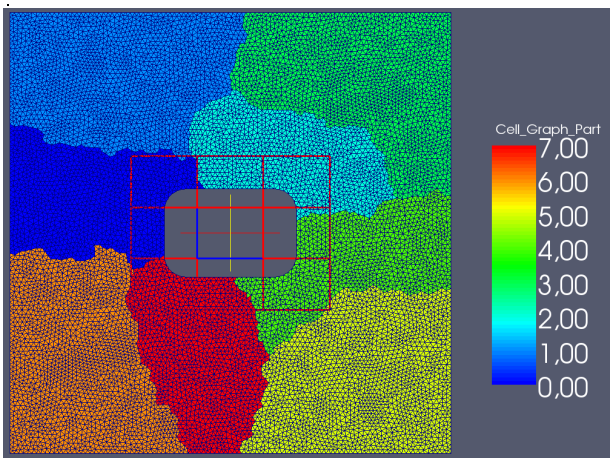
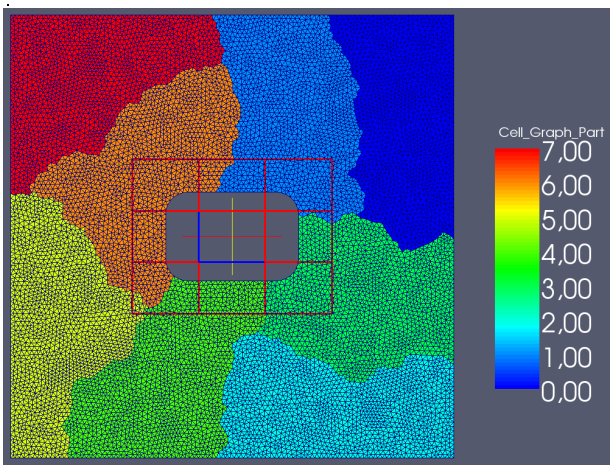
Stratégie	figure (tableau)	coût A		poids S		Equilibrage du poids		Communication	
		N	B	N	B	ph1	ph2	FE-FE	FE-MD
PG_CU	6.11 (a)	1	1	1	1	12.5 ; 0	12.5 ; 10,6	3 ; 4	0 ; 4
PG_CC_S	6.11 (b)	1	1	1	4	12.5 ; 0	12.5 ; 13,5	3 ; 5	0 ; 4
PG_CC_A	6.11 (c)	1	4	1	1	12.5 ; 0	12.5 ; 14,1	3 ; 5	0 ; 4
PG_CC_SA	6.11 (d)	1	4	1	4	12.5 ; 0	12.5 ; 13,4	3 ; 5	0 ; 3

Tab. 6.10: Diagnostique du partitionnement classique du maillage d'une géometrie complexe en le modélisant par un graphe.

Ref 6.11	Résultat	Commentaires
(a)	 **Avec un coût uniforme sur la zone normale et la zone de recouvrement (PG_CU).** % répart ph1 : [12,5 12,5 12,5 12,5 12,5 12,5 12,5 12,5] % répart ph2 : [19 7 28 0 2 26 8 10] com FE-FE : [3 4 4 3 3 4 4 3] com FE-MD : [3 2 4 0 1 3 2 2]	**Cette configuration correspond aux coûts réels de LibMultiScale**. Il y a un déséquilibre entre les parties pendant la 2eme phase. En effet les processeurs 2 et 5, qui ont respectivement 28% et 26% de la charge de cette phase, ralentirons les autres.

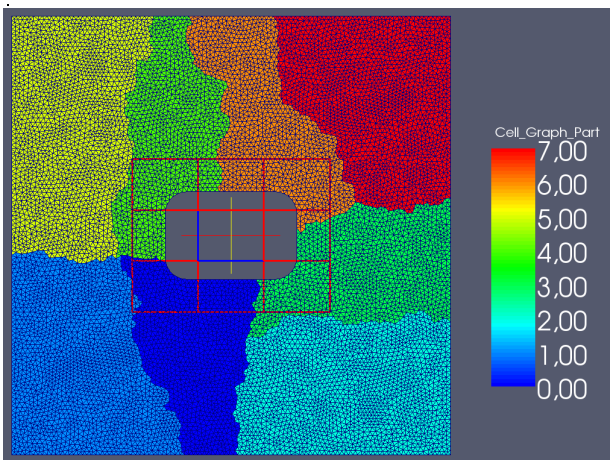
... suite à la page suivante ...

... suite de la page précédente ...

Ref 6.11	Résultat	Commentaires
(b)	 **Avec un coût supplémentaire de couplage pour les sommets (PG_CC_S).** % répart ph1 : [12,5 12,5 12,5 12,5 12,5 12,5 12,5 12,5] % répart ph2 : [24 0 28 0 26 0 0 22] com FE-FE : [5 4 5 4 5 3 3 5] com FE-MD : [4 1 4 0 3 0 0 3]	Le surcoût qu'on rajoute aux sommets de la zone de recouvrement n'a pas amélioré les résultats ici. On constate qu'il y a un déséquilibre plus important entre les parties pendant la 2eme phase. En effet les processeurs 0, 2, 4 et 7, qui partagent la charge de cette phase, seront occupé tout le temps alors que les autres processeurs auront un temps d'inactivité remarquable.
(c)	 **Avec un coût supplémentaire de couplage pour les arêtes (PG_CC_A).** % répart ph1 : [12,5 12,5 12,5 12,5 12,5 12,5 12,5 12,5] % répart ph2 : [0 23 0 22 20 0 35 0] com FE-FE : [3 5 3 5 5 4 5 4] com FE-MD : [0 3 0 3 3 0 4 0]	Avec un coût supplémentaire seulement sur les hyper-arêtes, on obtient le même résultat que dans (b).

... suite à la page suivante ...

... suite de la page précédente ...

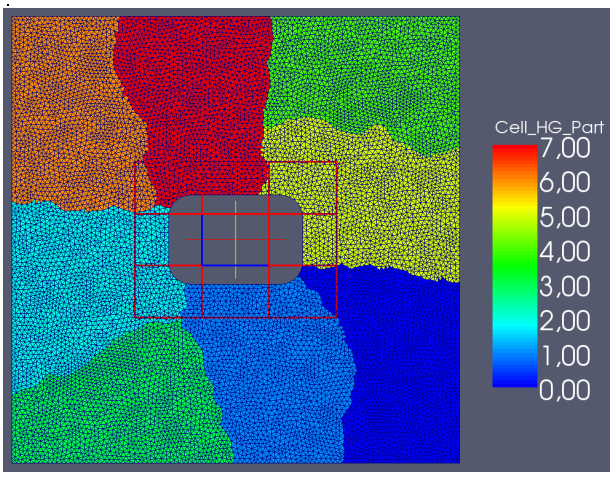
Ref 6.11	Résultat	Commentaires
(d)	 **Avec un coût supplémentaire de couplage pour les sommets et les arêtes(PG_CC_SA).** % répart ph1 : [12,5 12,5 12,5 12,5 12,5 12,5 12,5 12,5] % répart ph2 : [26 0 0 22 27 0 25 0] com FE-FE : [5 4 3 5 5 3 4 3] com FE-MD : [3 0 0 3 3 0 3 0]	Avec un coût supplémentaire sur les sommets et hyper-arêtes, les résultats ne s'améliorent pas et on a toujours un déséquilibre entre les parties.

Tab. 6.11: Partitionnement classique du maillage d'une géometrie complexe en le modélisant par un graphe.

6.2.3 Partitionnement du maillage en le modélisant par un hypergraphe classique

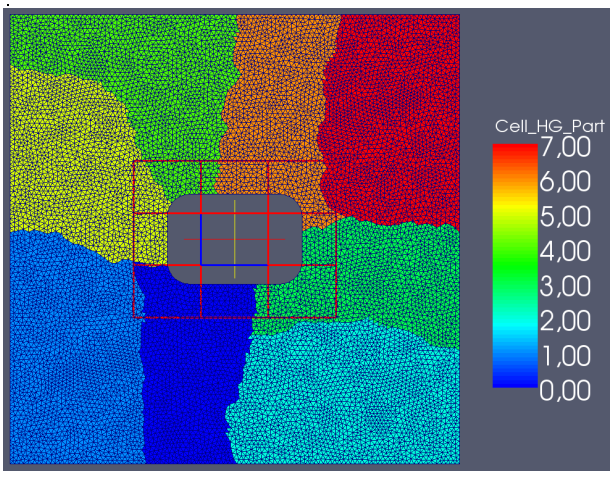
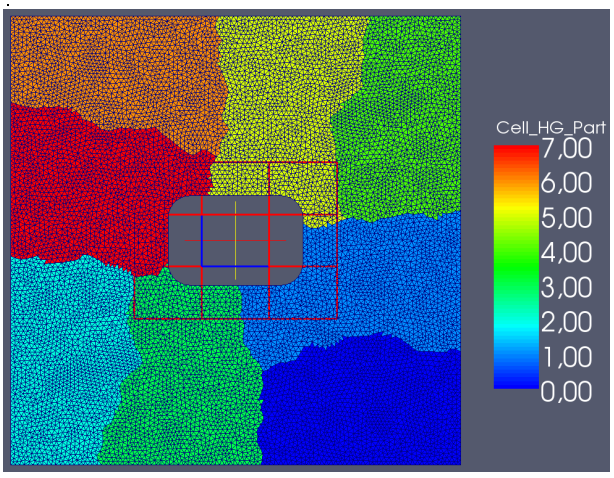
Stratégie	figure	coût Ha			poids S		Equilibrage du poids		Communication	
		cHe	N	B	N	B	ph1	ph2	FE-FE	FE-MD
PHg_CU	6.13 (a)	0	1	1	1	1	12.5 ; 0	12.5 ; 11	2 ; 4	0 ; 4
PHg_CC_S	6.13 (b)	0	1	1	1	4	12.5 ; 0.04	12.5 ; 10	2 ; 4	0 ; 3
PHg_CC_H	6.13 (c)	0	1	4	1	1	12.5 ; 0	12.5 ; 13	2 ; 4	0 ; 4
PHg_CC_SH	6.13 (d)	0	1	4	1	4	12.5 ; 0	12.5 ; 13	2 ; 4	0 ; 4

Tab. 6.12: Diagnostique du partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe.

Ref 6.13	Résultat	Commentaires
(a)	 **Avec un coût uniforme sur la zone normale et la zone de recouvrement (PHg_CU).** % répart ph1 : [12,5 12,5 12,5 12,5 12,5 12,5 12,5 12,5] % répart ph2 : [4 26 21 0 0 24 2 23] com FE-FE : [2 3 4 2 2 3 2 4] com FE-MD : [1 3 3 0 0 4 1 2]	**Cette configuration correspond aux coûts réels de LibMultiScale**. La modélisation avec un hypergraphe n'a pas amélioré les résultats dans cette configuration par rapport à 6.11 (a).

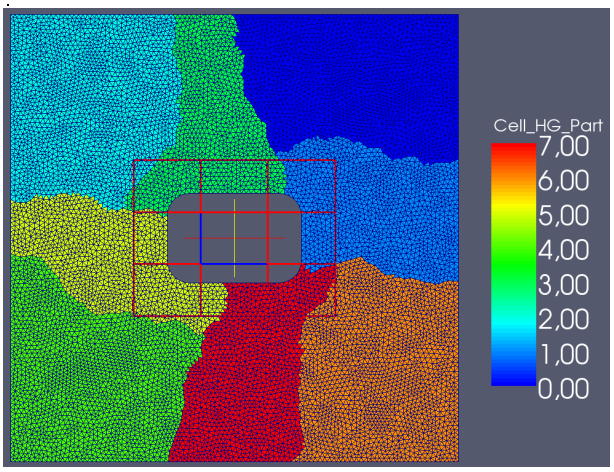
... suite à la page suivante ...

... suite de la page précédente ...

Ref 6.13	Résultat	Commentaires
(b)	 **Avec un coût supplémentaire de couplage pour les sommets (PHg_CC_S).** % répart ph1 : [13 13 13 13 12 12 12 12] % répart ph2 : [25 0 0 23 12 15 24 1] com FE-FE : [4 2 2 4 2 3 3 2] com FE-MD : [3 0 0 3 2 3 3 1]	On constate qu'il y a un déséquilibre entre les parties pendant la 2eme phase même en rajoutant des coûts supplémentaires sur les sommets de la zone de recouvrement.
(c)	 **Avec un coût supplémentaire de couplage pour les hyper-arêtes (PHg_CC_H).** % répart ph1 : [12,5 12,5 12,5 12,5 12,5 12,5 12,5 12,5] % répart ph2 : [0 27 0 20 0 27 0 26] com FE-FE : [2 4 2 4 2 4 2 4] com FE-MD : [0 3 0 3 0 3 0 4]	On constate qu'il y a un déséquilibre entre les parties pendant la 2eme phase même en rajoutant des coûts supplémentaires sur les hyper-arêtes de la zone de recouvrement.

... suite à la page suivante ...

... suite de la page précédente ...

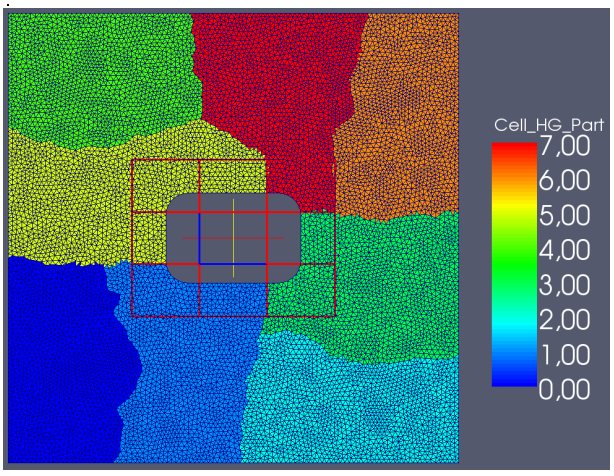
Ref 6.13	Résultat	Commentaires
(d)	 **Avec un coût supplémentaire de couplage pour les sommets et les hyper-arêtes(PHg_CC_SH).** % répart ph1 : [12 12 12 13 12 13 12 12] % répart ph2 : [0 21 0 29 0 28 0 21] com FE-FE : [2 4 2 4 2 4 2 4] com FE-MD : [0 3 0 4 0 4 0 2]	Avec un coût supplémentaire sur les sommets et hyper-arêtes de l'hypergraphe, les résultats ne s'améliorent pas et on a toujours un déséquilibre entre les parties.

Tab. 6.13: Partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe.

6.2.4 Partitionnement du maillage en le modélisant par un hypergraphe avec des hyper-arêtes de couplage

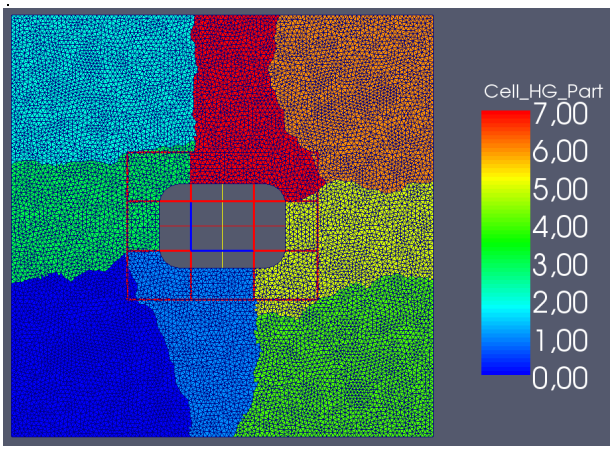
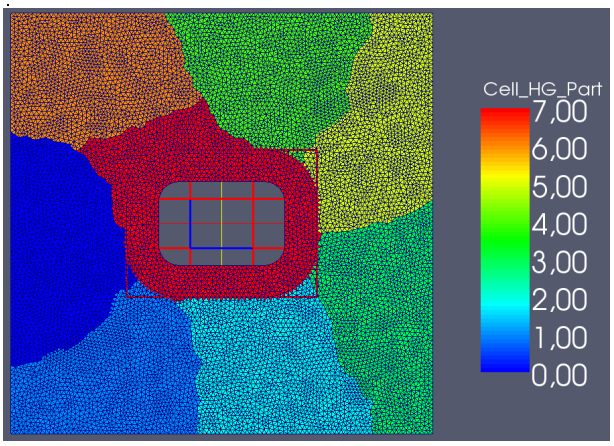
Stratégie	figure	coût He			poids v		Equilibrage du poids		Communication	
		cHe	N	B	N	B	ph1	ph2	FE-FE	FE-MD
PHg_HA_CU	6.15 (a)	50	1	1	1	1	12.5 ; 0.07	12.5 ; 14	2 ; 4	0 ; 3
PHg_HA_CC_S	6.15 (b)	50	1	1	1	4	12.5 ; 0.03	12.5 ; 12	2 ; 4	0 ; 3
PHg_HA_CC_H	6.15 (c)	50	1	4	1	1	12.5 ; 0.09	12.5 ; 33	3 ; 7	0 ; 8
PHg_HA_CC_SH	6.15 (d)	50	1	4	1	4	12.5 ; 0.16	12.5 ; 18	3 ; 5	0 ; 5

Tab. 6.14: Diagnostique du partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe et en rajoutant des hyper-arêtes des couplage.

Ref 6.15	Résultat	Commentaires
(a)	 **Avec un coût uniforme sur la zone normale et la zone de recouvrement (PHg_HA_CU).** % répart ph1 : [13 12 13 13 13 13 12 12] % répart ph2 : [0 27 0 24 0 36 0 13] com FE-FE : [2 4 2 4 2 4 2 4] com FE-MD : [0 2 0 3 0 3 0 1]	**Cette configuration correspond aux coûts réels de LibMultiScale**. On remarque que le partitionneur ne coupe pas au milieu d'une boite comme avant mais il essaye faire des coupes en suivant les limites des boites (i.e les limites qui sont décrites par les hyper-arêtes de couplage). Les communications FE-MD sont alors améliorées. Mais les partitions sont toujours déséquilibrées d'après les mesures.

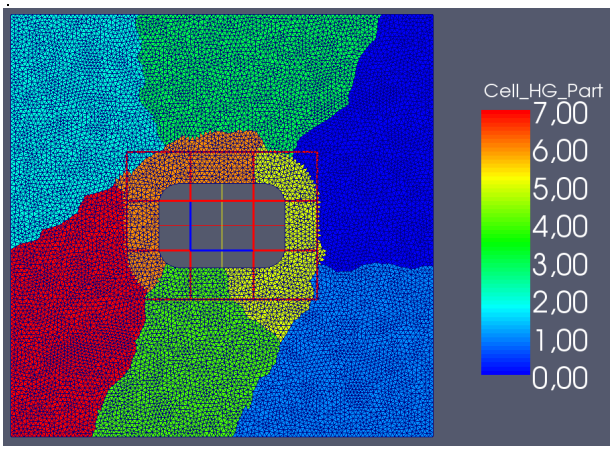
... suite à la page suivante ...

... suite de la page précédente ...

Ref 6.15	Résultat	Commentaires
(b)	 **Avec un coût supplémentaire de couplage pour les sommets (PHg_HA_CC_S).** % répart ph1 : [12 13 12 13 13 12 12 12] % répart ph2 : [0 27 0 23 0 24 0 26] com FE-FE : [2 4 2 4 2 4 2 4] com FE-MD : [0 2 0 2 0 3 0 2]	Les résultats ne s'améliorent pas en rajoutant un surcoût sur les sommets de la zone de recouvrement et on a toujours un déséquilibre malgré que les hyper-arêtes de couplages ont permi d'améliorer les coupes au niveau de la zone de recouvrement.
(c)	 **Avec un coût supplémentaire de couplage pour les hyper-arêtes normales (PHg_HA_CC_H).** % répart ph1 : [12 12 12 12 12 12 13 13] % répart ph2 : [0 0 0 0 0 0 0 100] com FE-FE : [3 3 3 3 3 3 3 7] com FE-MD : [0 0 0 0 0 0 0 8]	Avec un surcoût uniquement sur les hyper-arêtes normales de la zone de recouvrement, la répartition est très déséquilibrée. Sur la figure (c), on represente le pire des résultats que nous avons obtenu dans cette configuration, où toute la zone de recouvrement a été affecté à un seul processeur.

... suite à la page suivante ...

... suite de la page précédente ...

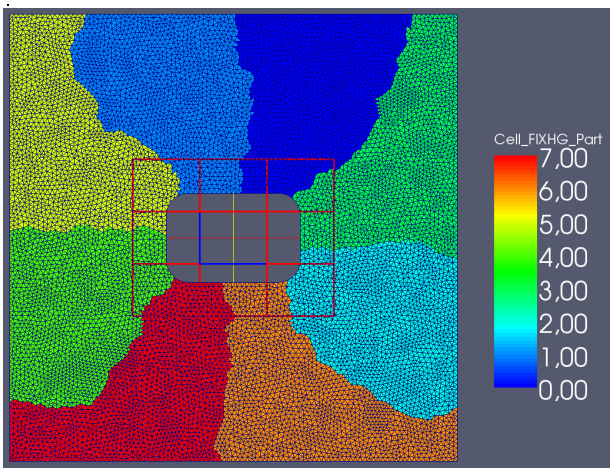
Ref 6.15	Résultat	Commentaires
(d)	 **Avec un coût supplémentaire de couplage pour les sommets et les hyper-arêtes normales (PHg_HA_CC_SH).** % répart ph1 : [12 13 12 12 13 12 13 12] % répart ph2 : [0 0 0 0 14 42 43 0] com FE-FE : [3 3 3 4 4 5 5 3] com FE-MD : [0 0 0 0 2 4 5 0]	Avec un coût supplémentaire sur les sommets et hyper-arêtes de l'hypergraphe, les résultats ne s'améliorent pas et on a toujours un déséquilibre entre les parties.

Tab. 6.15: Partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe et en rajoutant des hyper-arêtes des couplage.

6.2.5 Partitionnement du maillage en le modélisant par un hypergraphe avec technique des points fixes

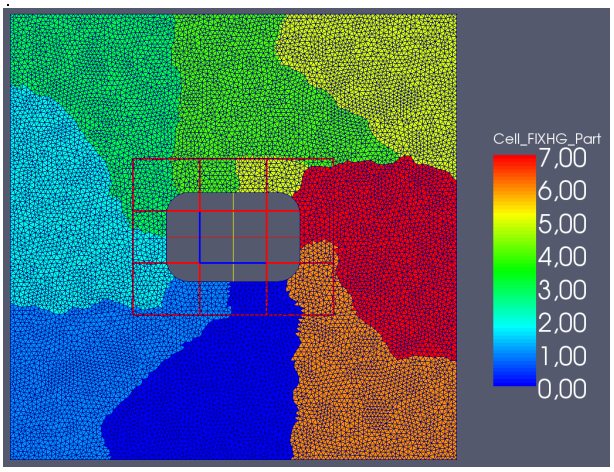
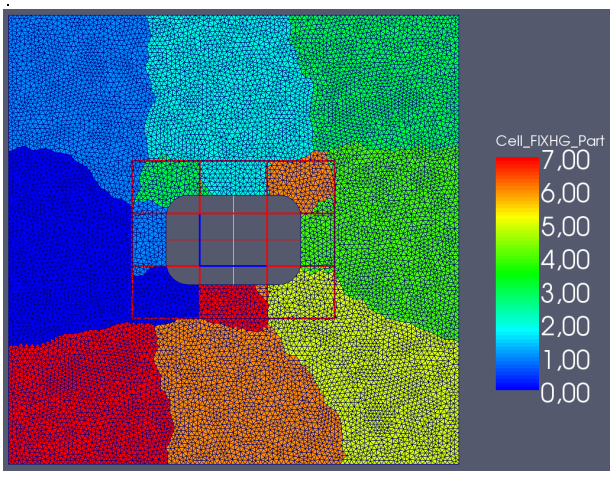
Stratégie	figure	coût He			poids v		Equilibrage du poids		Communication	
		cHe	N	B	N	B	ph1	ph2	FE-FE	FE-MD
PHg_PF_SHC	6.17 (a)	0	1	1	1	1	12.5 ; 0	12.5 ; 0.20	2 ; 2	2 ; 2
PHg_PF_SHC	6.17 (b)	0	1	4	1	4	12.5 ; 0.2	12.5 ; 0.06	2 ; 3	2 ; 2
PHg_PF_AHC	6.17 (c)	50	1	1	1	1	12.5 ; 0.1	12.5 ; 0.12	3 ; 6	1 ; 3
PHg_PF_AHC	6.17 (d)	50	1	4	1	4	12.5 ; 0.2	12.5 ; 0.02	2 ; 2	1 ; 3

Tab. 6.16: Diagnostique du partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe en utilisant la technique des points fixes.

Ref 6.15	Résultat	Commentaires
(a)	 **Avec un coût supplémentaire de couplage pour les sommets et sans hyper-arêtes de couplages (PHg_PF_SHC).** % répart ph1 : [13 13 13 13 13 13 13 13] % répart ph2 : [13 13 12 12 13 12 13 13] com FE-FE : [2 2 2 2 2 2 2 2] com FE-MD : [2 2 2 2 2 2 2 2]	**Cette configuration correspond aux coûts réels de LibMultiScale**. On obtient ici un bon équilibrage de la charge pendant la 1ere phase et la 2eme phase. En effet, chaque processeur est en charge de $12,5\%$ de la charge de chaque phase avec des écart-type 0 et $0,2$ respectivement pour les phases 1 et 2. D'un point de vue des communications on constate aussi que les résultats s'améliorent. En effet chaque processeur de FE communique avec 2 processeurs au maximum de MD.

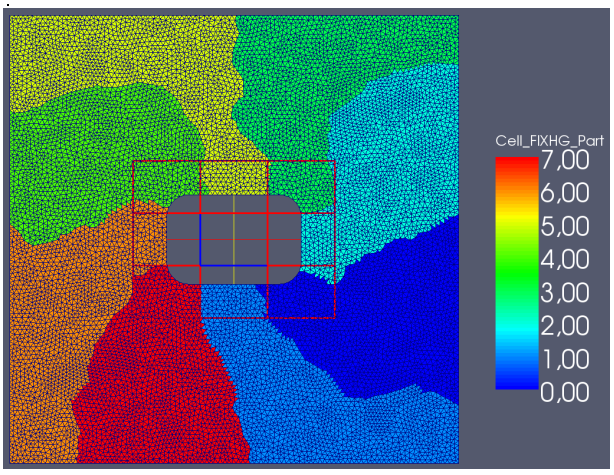
... suite à la page suivante ...

... suite de la page précédente ...

Ref 6.15	Résultat	Commentaires
(b)	 **Avec un coût supplémentaire de couplage pour les sommets et les hyper-arêtes et sans hyper-arêtes de couplages (PHg_PF_SHC).** % répart ph1 : [12 12 13 13 12 12 13 13] % répart ph2 : [13 13 13 12 12 12 13 13] com FE-FE : [2 2 2 2 3 2 2 3] com FE-MD : [2 2 2 2 2 2 2 2]	En rajoutant un coût supplémentaire sur les sommets de l'hypergraphe, la répartition reste bien équilibrée du point du vue charge de calcul pendant les deux phases. Les communications sont aussi acceptables mais on constate que certaines parties affectées à un même processeur ne sont pas connexes ce qui coûte des communications supplémentaires entre les processeurs de FE.
(c)	 **Avec un coût uniforme pour les sommets et les hyper-arêtes et avec hyper-arêtes de couplages (PHg_PF_AHC).** % répart ph1 : [12 12 13 13 12 13 13 13] % répart ph2 : [13 13 13 12 13 12 12 13] com FE-FE : [3 3 3 4 3 3 6 3] com FE-MD : [1 3 2 2 3 1 2 1]	**Cette configuration correspond aux coûts réels de LibMultiScale**. En rajoutant des hyper-arêtes de couplage par rapport à la configuration (a) la répartition reste équilibrée du point de vue équilibrage de la charge pour les deux phases avec des écart-types de $0,1$ et $0,12$ respectivement pour les deux phases. Mais, comme dans le résultat de (b), on constate que certaines parties affectées à un même processeur ne sont pas connexes.

... suite à la page suivante ...

... suite de la page précédente ...

Ref 6.15	Résultat	Commentaires
(d)	 **Avec un coût supplémentaire de couplage pour les sommets et les hyper-arêtes et avec hyper-arêtes de couplages (PHg_PF_AHC).** % répart ph1 : [13 13 13 13 12 12 13 13] % répart ph2 : [13 13 13 12 13 12 13 13] com FE-FE : [2 2 2 2 2 2 2 2] com FE-MD : [2 1 3 1 2 2 3 1]	On obtient ici un bon équilibrage de la charge pendant la 1ere phase et la 2eme phase. En effet, chaque processeur est en charge de $12,5\%$ de la charge de chaque phase avec des écart-types $0,2$ et $0,02$ respectivement pour les phase 1 et 2. Du point de vue des communications, on obtient également de bon résultats. En effet chaque processeur de FE communique avec 3 processeurs au maximum de MD.

Tab. 6.17: Partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe en utilisant la technique des points fixes.

6.2.6 Partitionnement du maillage en le modélisant par un hypergraphe avec la technique de multi-contraintes

Stratégie	figure	Equilibrage du poids		Communication	
		ph1	ph2	FE-FE	FE-MD
PHg_MC	6.3	12.5 ; 0.22	12.5 ; 0.53	2 ; 5	2 ; 4

Tab. 6.18: Diagnostique du partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe en utilisant la technique de multi-contraintes (PHg_MC).

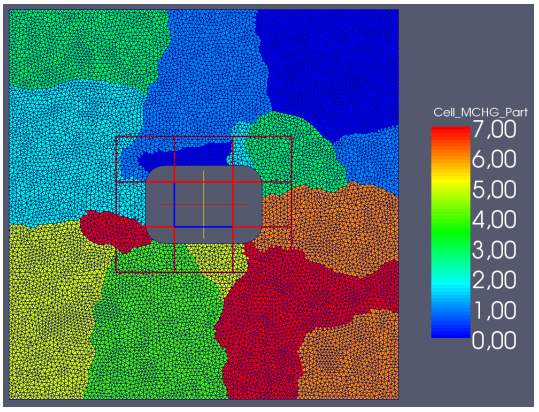


Fig. 6.3: Partitionnement du maillage d'une géometrie complexe en le modélisant par un hypergraphe en utilisant la technique de multi-contraintes (PHg_MC).

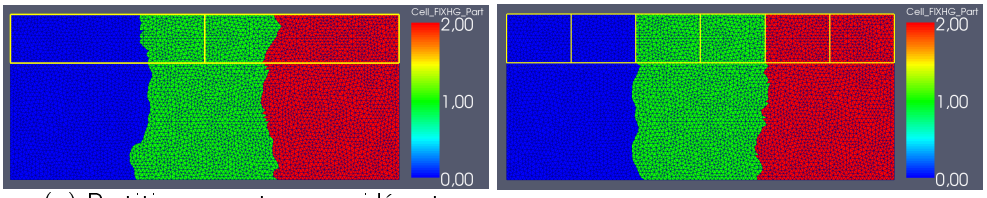
% répart ph1 : [13 12 12 13 13 12 13 12]
% répart ph2 : [13 13 13 13 13 11 12 12]
com FE-FE : [3 4 5 4 2 3 3 4]
com FE-MD : [2 2 4 2 2 2 2 3]

Avec cette stratégie le partitionneur équilibre la charge pour les deux phases. En effet, dans l'exemple de la figure 6.3 chaque processeur a en moyenne $12,5\%$ de charge de calcul pendant la 1ere phase avec un écart-type de $0,22$ et $12,5\%$ de charge pendant la 2eme phase avec un écart-type de $0,53$. Mais on ne peut pas garantir que les parties affectées au même processeur soient connexes, ce qui augmente les communications. En effet, chaque processeur de FE communique avec 2 à 5 processeurs de FE et avec 2 à 4 processeurs de MD.

6.3 Mise en évidence des intérêts et limitations des hyper-arêtes de couplage

La figure 6.4 représente les résultats de partitionnement d'un maillage rectangulaire sur 3 processeurs pour les EF et 2 processeurs pour la DM. En rajoutant des hyper-arêtes modélisant des boîtes d'atomes virtuelles (comme si on subdivise chaque boîte en sous-boîtes) on constate que celles-ci ont permis de guider le partitionneur pour que les coupes suivent les limitations des nouvelles hyper-arêtes. Dans le premier cas (figure 6.4-(a)) les résultats de partitionnement sont assez satisfaisants mais si le nombre de processeurs était plus grand on pourrait mieux mettre en évidence l'importance du lissage des partitions que permet de garantir ces hyper-arêtes spéciales.

La figure 6.5 représente les résultats de partitionnement d'un maillage ayant une zone de recouvrement carrée. On constate que le partitionneur n'a pas suivie les hyper-arêtes de couplage dans ce cas. On aurait voulu partitionner la zone de recouvrement en 4 parties et que chaque partie corresponde géométriquement



(a) Partitionnement en considérant uniquement les hyper-arêtes de couplage principales : une hyper-arête par boîte d'atomes.

(b) Partitionnement avec des hyper-arêtes de couplage secondaires : 2 hyper-arêtes de couplage par boîte d'atomes.

Fig. 6.4: Mise en évidence du rôle des hyper-arêtes de couplage pour un meilleur partitionnement : Partitionnement d'un maillage de géometrie simple avec 2 processeurs DM et 3 processeurs EF.

à une boîte d'atomes, mais ici le partitionneur a fait une coupe horizontale qui sépare chaque hyper-arête de couplage en 2. Les hyper-arêtes secondaire de couplage (figure 6.5-(d)) ont permis de lisser les résultats au niveau des communications, mais cela n'améliore en rien l'équilibrage de la charge dans la phase de couplage (phase 2).

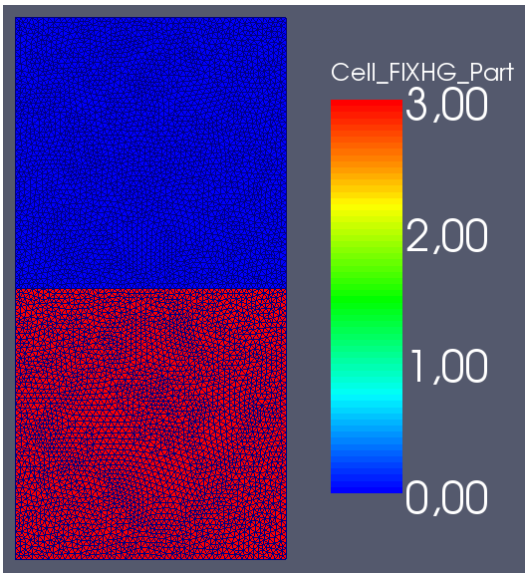
6.4 Conclusion

D'après les comparaisons précédentes, on constate qu'on obtient les meilleurs résultats avec la technique des points fixes. En effet, le fait de partitionner séparément la région de recouvrement et la région normale garantit l'équilibrage de charge entre tous les processeurs et pendant les deux phases de la simulation. Par contre en partitionnant tout le maillage d'un seul coup et en jouant sur les poids de la zone normale et de la zone de recouvrement, il n'est pas possible de contraindre le partitionneur à équilibrer chaque phase de calcul. En effet, il on trouve un déséquilibre surtout pour la 2eme phase. Avec la technique multi-contraintes, on a pu garantir un équilibrage de charge pendant les deux phases : tous les processeurs ont en moyenne la même charge sur les deux phases avec un écart-type faible. Mais avec cette dernière technique, on n'a aucun contrôle sur les communications et on constate que souvent les parties affectées au même processeur ne sont pas connexes ce qui augmente considérablement les communications entre les processeurs, aussi bien pour les communications FE-FE que les communications FE-MD.

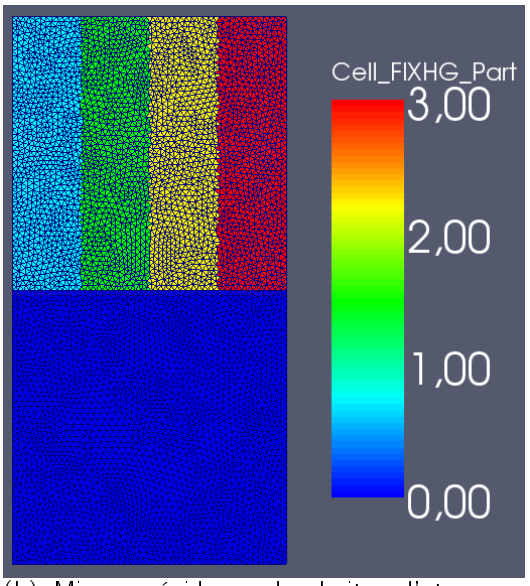
On constate aussi que les hyper-arêtes de couplage permettent dans certains cas de guider le partitionneurs pour avoir un meilleur partionnement, Mais leur utilisation n'est pas indispensable. En effet, on a vu qu'en utilisant la technique des points fixes, on obtient le même résultat avec ou sans hyper-arêtes de couplage.

La stratégie de partitionnement avec les points fixes garantit de meilleurs résultats dans le cas d'une géometrie simple. Par contre si la géometrie est plus complexe, cette stratégie ne donne de bon résultats que dans des configurations particulières : lorsqu'on ne rajoute pas de surcoût sur la zone de recouvrement (cas de *LibMultiScale*) et sans hyper-arêtes de couplage ou bien en rajoutant un coût supplémentaire pour les sommets et hyper-arêtes de la zone de recouvrement et avec hyper-arêtes de couplage.

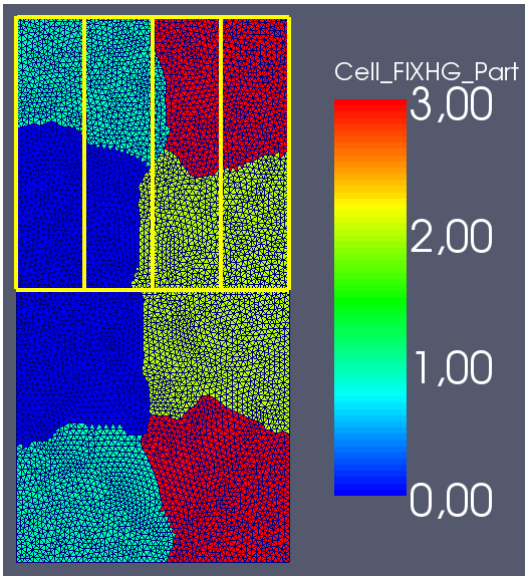
Au final, nous sommes conscients qu'il s'agit de résultats préliminaires, et qu'il faudrait étudier des cas plus complexes où le nombre de processeursde FE et le nombre de boîtes d'atomes dans la zone de recouvrement sont différents.



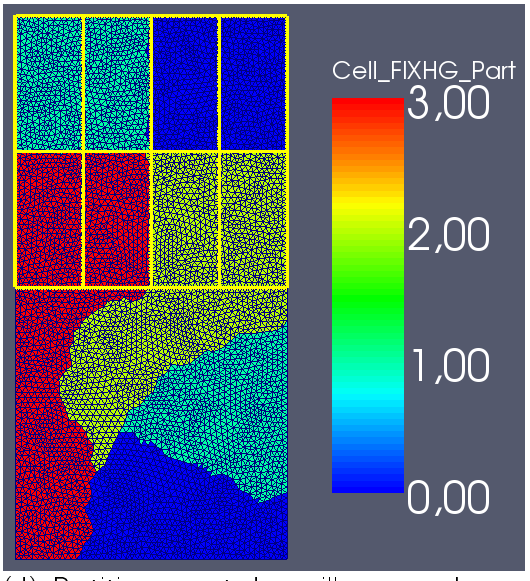
(a) distinction de la zone de recouvrement et de la zone normale : la zone de recouvrement est la zone Bleue.



(b) Mise en évidence des boites d'atomes : chaque élément de la zone de recouvrement du maillage porte la couleur de la boite qui lui contient.



(c) Partitionnement du maillage avec hyper-arêtes de couplage verticales modélisant les boites d'atomes.



(d) Partitionnement du maillage avec hyper-arêtes de couplage principales modélisant les boites d'atomes et avec d'autres hyper-arêtes secondaires de couplage pour guider le partitionneur.

Fig. 6.5: Limitations des hyper-arêtes de couplage : partitionnement d'un maillage ayant une zone de recouvrement carrée (cas d'un coût uniforme avec la technique des points fixes).

7

Conclusion générale et perspectives

En guise de conclusion, nous souhaitons donner un ensemble de perspectives pour ce travail. On a vu dans le chapitre 4 que nous avons modélisé les dépendances de calculs entre les éléments en regroupant dans une seule hyper-arête les éléments concernés avec une simplification qui consiste à considérer les premiers voisins de chaque élément. Nous avons testé la validité de cette simplification mais pour être plus rigoureux, il faudrait faire une étude plus détaillée pour justifier ce choix (cf. section 4.4.1).

Nous avons obtenu les meilleurs résultats de partitionnement en utilisant les hyper-arêtes de couplage et la technique des points fixes. Avec les hyper-arêtes de couplage, nous pouvons optimiser le nombre de communications et avec la technique des points fixes nous pouvons garantir un bon équilibrage de la charge pendant les deux phases. Mais nous ne pouvons pas assurer que les parties affectées à un même processeur soient contiguës sur le maillage.

Nous avons fait des calculs des coûts en se basant sur les complexités et les dépendances de calcul et nous avons constaté que le coût de couplage, bien qu'il ne soit pas négligeable, reste dominé par les autres coûts de calcul. Il s'avère plus pertinent de mettre des coût artificiels pour mieux guider le partitionneur. Pour déterminer les valeurs de ces surcoûts qu'on devrait mettre, nous avons fait une étude exhaustive en faisant varier les coûts et voir leur influence sur les résultats de partitionnement (Annexe F).

La principale nouveauté dans ce travail est la modélisation avec des hypergraphes et l'introduction de la notion d'hyper-arête de couplage. On a vu que pour les stratégies de partitionnement sans points fixes, l'utilisation des graphes et des hypergraphes sont équivalentes. Pour finir la comparaison entre la modélisation avec un graphe et la modélisation avec un hypergraphe dans notre cas, il faudrait voir les résultats de partitionnement du maillage en le modélisant par un graphe et en utilisant la stratégie des points fixes. Nous n'avons pas encore implanté la fonction qui utilise cette stratégie dans le cas d'un graphe, mais dans le cas d'un hypergraphe on a vu que c'était la meilleure stratégie de partitionnement pour garantir un équilibrage de la charge pendant la 1ere et la 2eme phase.

D'après nos calculs on a constaté que les coûts théoriques de couplage sont négligeables devant les autres coûts, ce qui ne correspond pas à la réalité. En effet, d'après les mesures [8], le temps passé pendant la phase de couplage n'est pas négligeable. Cette différence est due au fait que nous n'avons pas pris en compte les communications extérieures (entre les processeurs de la dynamique moléculaire et des processeurs de la mécanique des milieux continus) dans les calculs. Du point de vue de la modélisation en graphe ou hypergraphe, ces communications induisent un surcoût en temps dans la phase de couplage, qui s'apparente à un poids supplémentaire qu'il faudrait ajouter à C_e^{BZ} (formule 4.19). Ces coûts dépendent principalement des architectures utilisés et on peut les estimer difficilement. Par ailleurs, il reste à intégrer notre module dans *LibMultiScale* afin de finir la validation de ce travail.

Le schéma d'integration de la mécanique des milieux continus dans l'état actuel nécessite une synchronisation à chaque pas de temps pour pouvoir poursuivre les calculs car la mise à jour de la vitesse lors de la dernière étape de la tâche d'integration dépend du calcul de la force F (voir shéma d'integration du domaine continu page 15). On ne peut donc pas passer à la phase de couplage avant la mise à jour de la vitesse car celle-ci sera corrigée pendant cette étape. D'après les dépendances que nous avons exprimé dans le chapitre 4, nous avons constaté qu'il est possible de fusionner la dernière étape de mise à jour de la vitesse, lors de la tâche d'integration, avec l'étape de calcul de la force F, car pendant ce calcul on va parcourir tous les

éléments. L'idée est qu'on peut bénéficier de ce parcours et mettre à jour au fure et à mesure la contribution de l'élément visité sur la valeur finale de F et cela afin de réduire la synchronisation entre les deux phases.

Nous avons utilisé les outils *PATOH* et *METIS* pour le partitionnement. Nous avons en disposition également de *SCOTCH* et *ZOLTAN* que nous n'avons pas encore testés. Pour analyser les résultats, nous avons développé un module qui se charge de faire le diagnostique du résultat et de calculer les statistiques de répartition de la charge et des communications entre les processeurs pour évaluer la qualité du partitionnement dans toutes les stratégies que nous avons testé. On pourra aussi chercher à évaluer avec plus de précision la qualité des communications entre les processeurs des éléments finis et les processeurs de la dynamique moléculaire (prenant en compte des volumes de communications).

Les cas test que nous avons considérés, ainsi que la conception de notre module de répartition de la charge ne prennent en compte qu'une géometrie en deux dimensions. Mais notre structure de donnée est générique et nous avons préparé le module pour une future évolution vers des cas en 3D.

Tout le code produit est disponible en ligne sur la geforge de l'*INRIA*[1].

[1] Le code source est à l'adresse : http://gforge.inria.fr/projects/mpicpl/.

Glossaire

Lennard-Jones

Le potentiel de *Lennard-Jones*, qui est plus précisément une énergie potentielle, est souvent utilisé lors de simulations numériques de type dynamique moléculaire pour décrire les interactions entre deux atomes au sein d'un gaz monoatomique de type gaz rare.

VTK

Visualization ToolKit est une bibliothèque de classes C++ Open Source pour la visualisation de données 2D/3D et le traitement d'image.

éléments finis

La méthode des éléments finis est utilisée en analyse numérique pour résoudre des équations aux dérivées partielles. Celles ci peuvent representer le comportement dynamique de certains ph énomènes physique

borderline

borderline est un cluster x86 64 bits qui contient 10 noeuds interconnectés par un réseau infiniband, quadro-opteron (dual core) à 2.6 GHz avec 32 GB. Ce cluster est à Bordeaux et fait partie du réseau *Grid'5000*

dynamique moléculaire

Elle consiste à étudier la trajectoire d'une molécule en appliquant les lois de la mécanique classique, et a pour but de simuler le mouvement des atomes au cours du temps, à partir de l'énergie potentielle calculée par la mécanique moléculaire.

gmsh

GMSH est un outil de modelage, maillage et post-traitement diffusé sous licence GNU GPL (exécutables, sources et documentation disponibles sur ce site `http://www.geuz.org/gmsh`).

hypergraphes

Les hypergraphes sont des objets mathématiques généralisant la notion de graphes dans le sens où les arêtes ne relient plus seulement un ou deux sommets, mais un nombre quelconque de sommets (compris entre un et le nombre de sommets de l'hypergraphe).

mécanique des milieux continus

est le domaine de la mécanique qui s'intéresse à la déformation des solides et à l'écoulement des fluides, nous nous intéresserons ici essentiellement à la déformation des solides.

problème NP-complet

On dit qu'un problème est *NP* (**Non déterministe Polynomial**) quand on ne connaît pas un algorithme de résolution polynomiale en la taille des données, cependant qu'il est possible à l'aide d'un algorithme polynomiale en la taille des données de vérifier si une réponse proposée est effectivement une solution. Intuitivement, les problèmes *NP-complet* sont les problèmes qui peuvent être résolus en énumérant l'ensemble des solutions possibles et en les testant à l'aide d'un algorithme polynomial.

tenseur des contraintes

Le tenseur des contraintes est une représentation utilisée en mécanique des milieux continus pour

représenter l'état de contrainte, c'est-à-dire les forces surfaciques (parfois appelées efforts) mises en jeu entre les portions déformées du milieu. Le terme a été introduit par *Cauchy* vers 1822. [4]

Cinquième partie

Annexes

A

Utilisation de GMSH

A.1 Utilisation de GMSH

GMSH [16] est un outil de maillage libre ou réglé qui intègre un modeleur et des fonctionnalités de posttraitement. Le modeleur, simple d'emploi, permet de préparer le maillage de structures par constructions géométriques (création de lignes, surfaces, translations/rotations, extrusions...). Le modèle de la structure peut être construit interactivement (à la souris) ou bien par un fichier de commandes. La documentation de *GMSH* est disponible [10]. La description de la géométrie de la structure est contenue dans un fichier .geo, le maillage produit est enregistré par défaut avec l'extension .msh ;

Ramarque Dans *GMSH*, on visualise les entités constituant un physical par le menu Tools/Visibility et en sélectionnant le numéro.

Attention Il ne faut pas créer des physicals regroupant d'autres physicals, car cela conduirait à produire des mailles doubles.
Le numéro des surfaces physiques[1] doit être inférieur à 1 000 000.

A.2 Exemples

On crée une géométrie très simple : un carré de côté 1. Ici, on crée les points, les lignes, la surface ; on aurait aussi pu la faire par translation/extrusion...

On crée un physical qui contient la surface carrée de numéro 107.

Dans le fichier .geo on décrit le contour, les points et lignes interessantes et aussi la surface à mailler. le fichier .geo va contenir le code suivant pour notre exemple.

[1] Le mot clé pour désigner les surface physique dans *gmsh* est : *Physical Surface*.

```
L=1;
d=0.45;
Point(1) = {0, 0, 0, d};
Point(2) = {L, 0, 0, d};
Point(3) = {L, L, 0, d};
Point(4) = {0, L, 0, d};
Line (1) = {1, 2};
Line (2) = {2, 3};
Line (3) = {3, 4};
Line (4) = {4, 1};
Line Loop (106) = {2, 3, 4, 1};
Plane Surface (6) = {106};
Physical Surface(107) = {6};
```

Le maillage créé avec l'algorithme 2D est sur la figure B.1 :

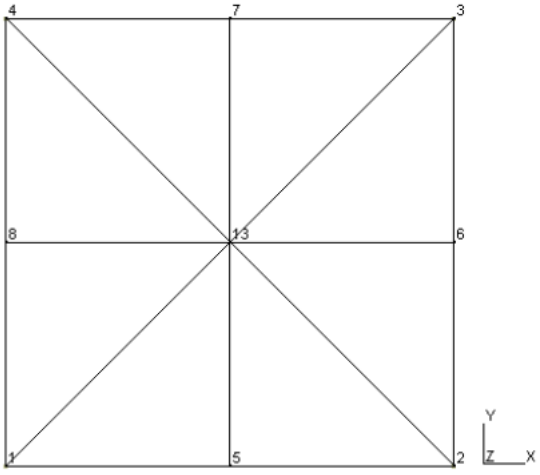


Fig. A.1: Maillage obtenu avec gmsh. [16]

Remarque En ligne de commande, on peut obtenir le fichier msh en tapant la commande :
$> gmsh -o out.msh -format msh -2 in.geo
Le -2 pour dir 2D
et on visualise avec $> gmsh out.msh

Le fichier du maillage *GMSH* contient la description des noeuds et des mailles :

```
$NOD
9
1 0 0 0
2 1 0 0
3 1 1 0
4 0 1 0
5 0.4999999999999931 0 0
6 1 0.4999999999999931 0
7 0.500000000000007 1 0
8 0 0.500000000000007 0
13 0.5000000000000001 0.5 0
$ENDNOD
$ELM
8
1 2 107 6 3 13 1 5
2 2 107 6 3 8 1 13
3 2 107 6 3 5 2 13
4 2 107 6 3 13 2 6
5 2 107 6 3 6 3 13
6 2 107 6 3 13 3 7
7 2 107 6 3 7 4 13
8 2 107 6 3 13 4 8
$ENDELM
```

Remarque La syntaxe de ce fichier est décrite dans la section "File formats/Gmsh mesh file format" du manuel de *GMSH*.

B

Préparation d'un cas de test pour LibMultiScale

On se propose de construire un cas de test *argon2D* pour le client *AMEL*[1] de *LibMultiScale*.
Ici on se basera sur l'exemple de géometrie décrite dans le manuel de *LibMultiScale* [7] pour construire notre exemple.
Les étapes sont détaillé dans le manuel d'utilisation de *LibMultiScale*. Par contre, la configuration du cas de test qui a été construite dans le manuel repose principalement sur l'utilisation de *stamps* pour faire la simulation de la dynamique moléculaire et de *libmesh* pour la mécanique des milieux continus. Le problème est que ces deux codes sous-jacent ne sont pas disponible pour nous et qu'on ne peut simuler qu'avec *lammps* pour la dynamique moléculaire et *simulpack* pour la mécanique des milieux continus. Donc on va décrire dans cet annexe les modifications et les descriptions necessaires pour un configuration avec *lammps* et *simulpack*.
Sur la figure B.1 on décrit les dépendances des fichiers de configuration du cas de test. Le fichier principale qu'on passe au client *AMEL* est le fichier

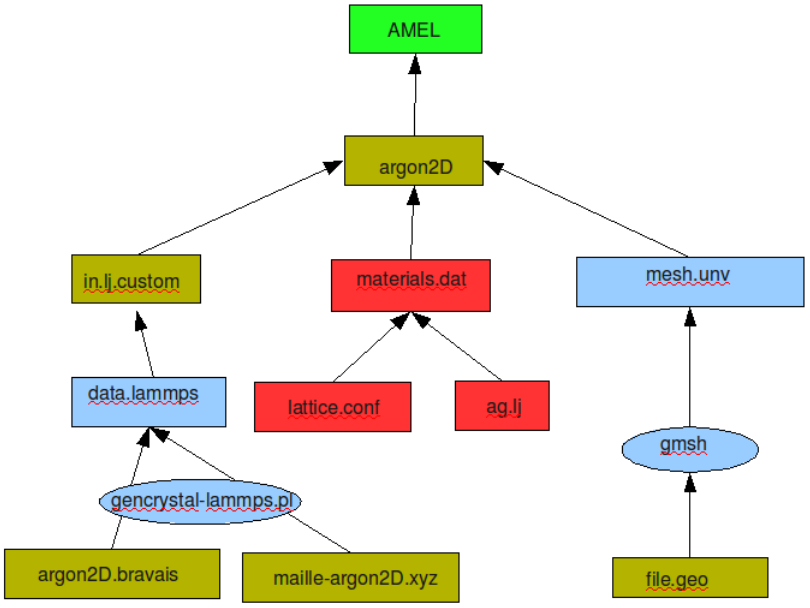


Fig. B.1: Dépendance des fichiers pour construire un cas de test (argon 2D) pour LibMultiscale.

pour lancer la simulation sur borderline[2] [3] [13] par exemple sur 4 processeurs pendant 10 pas de calcul, on utilise la commande suivante :
mpirun -np 4 AMEL argon2D 10

Les sections supplémentaire pour décrire la configuration de *lammps* et *simulpack* sont :

[1] AMEL : Atomistic Multi-domain coupling with External finite elements for LibMultiScale
[2] cf. Glossaire

B.1 Configuration pour Lammps

```
#Lammps section
Section Lammps RealUnits
DOMAIN_NAME md
LAMMPS_FILE config-lammps/in.lj.custom
#definition des geometries utiles pour la geometrie totale
#values used to define the half circles and central square
LET ANGLE1 = pi/2
LET ANGLE2 = 3 * ANGLE1
LET rayon = h/2 + bridgedSize
LET delta = r0/6
GEOMETRY 2 BALL 2 -l/2 0 0 rayon-delta ANGLE1 ANGLE2
GEOMETRY 3 BALL 2 l/2 0 0 rayon-delta ANGLE2 ANGLE1
GEOMETRY 4 CUBE 2 -l/2 l/2 -rayon+delta rayon-delta 0 0
GEOMETRY 5 UNION 2 3
GEOMETRY 6 UNION 5 4
#Useful definition for the definition of the atomistic boundary
#from a shrinked version of the atomistic domain
LET rcut = r0 * 2.6
GEOMETRY 7 SHRINK 5 rcut 0 0
GEOMETRY 8 SHRINK 4 0 rcut 0
GEOMETRY 9 UNION 7 8
GEOMETRY 10 GROW 5 r0 0 0
GEOMETRY 11 GROW 4 0 r0 0
GEOMETRY 12 UNION 10 11
GEOMETRY 13 SUB 12 9
DOMAIN_GEOMETRY 6
DOMAIN_BOUNDARY_GEOMETRY 13
endSection
```

B.2 Configuration pour SimulPack

```
#SimulPack section
Section SimulPack RealUnits
MESH_FILE config-libmesh/mymesh2.unv L/2 L/2 L/2
DOMAIN_NAME mesh
GEOMETRY 17 CUBE 2 -L/2 L/2 -L/2 L/2 0 0
GEOMETRY 18 SHRINK 5 bridgedSize 0 0
GEOMETRY 19 SHRINK 4 0 bridgedSize 0
GEOMETRY 20 UNION 18 19
GEOMETRY 21 SUB 17 20
PBC 0 0 0
TIMESTEP 1e-15
ELEMENT_TYPE TRIANGLE1
DOMAIN_GEOMETRY 21
MATERIAL_FILE materials-lj-argon2D.dat
endSection
```

C

Exemple de description d'une géometrie simple avec Gmsh

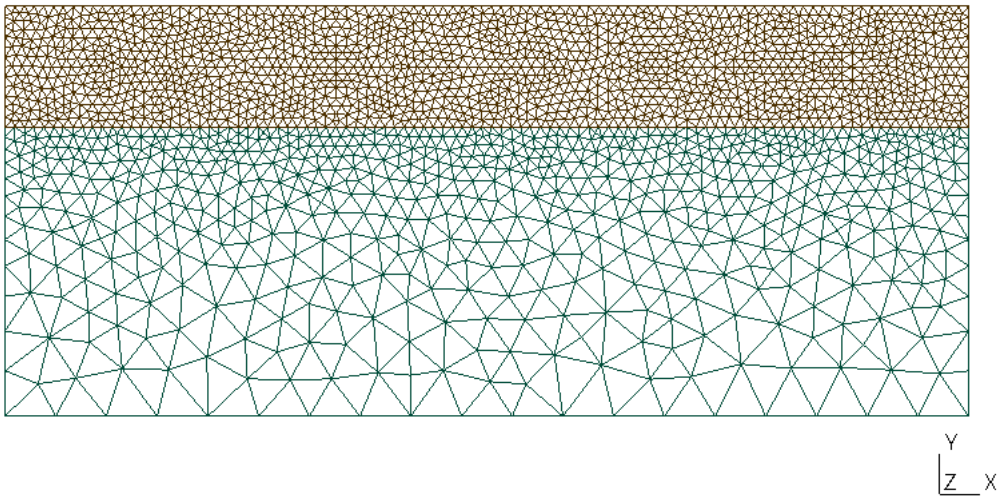


Fig. C.1: Maillage simple généré avec gmsh

À l'aide des variables *lc1* et *lc2* on peut paramétrer la taille des triangles (dans nos cas test du chapitre 6, nous avons pris les valeurs $lc1 = 0.02$ et $lc2 = 0.02$). Les autres variables (l_x, l_x, r_out_x, r_out_y, r_in_x, r_in_y) permettent de définir le contour et de limiter la zone de recouvrement Ω_R et la zone normale $\Omega_C - \Omega_R$ (voir la figure 2.1 page 13).

lc1 : parametrer la taille des triangle de la zone de recouvrement Ω_R
lc2 : parametrer la taille des triangle de la zone de continue $\Omega_C - \Omega_R$

```
lc1 = 0.1;
lc2 = 0.02;

l_x = 1;
l_y = 1;

r_out_x = 1;
r_out_y = 0.4;

r_in_x = 1;
r_in_y = 0.15;
```

```
Point(1)  = {-r_in_x , -r_in_y , 0 , lc2};
Point(2) = { r_in_x , -r_in_y , 0 , lc2};
Point(3) = {-r_out_x , -r_out_y , 0 , lc2};
Point(4) = { r_out_x , -r_out_y , 0 , lc2};
Point(5) = {-l_x , -l_y , 0 , lc1};
Point(6) = { l_x , -l_y , 0 , lc1};

Line(1) = {1,2};
Line(2) = {2,4};
Line(3) = {3,4};
Line(4) = {3,1};
Line(5) = {4,6};
Line(6) = {6,5};
Line(7) = {5,3};

Line Loop(160) = {-3,4,1,2};
Plane Surface(161) = {160};
Physical Surface(162) = {161};

Line Loop(170) = {6,7,3,5};
Plane Surface(171) = {170};
Physical Surface(172) = {171};
```

D

Exemple de description d'une géometrie complexe avec Gmsh

D.1 Exemple d'une zone de recouvrement rectangulaire arrondie

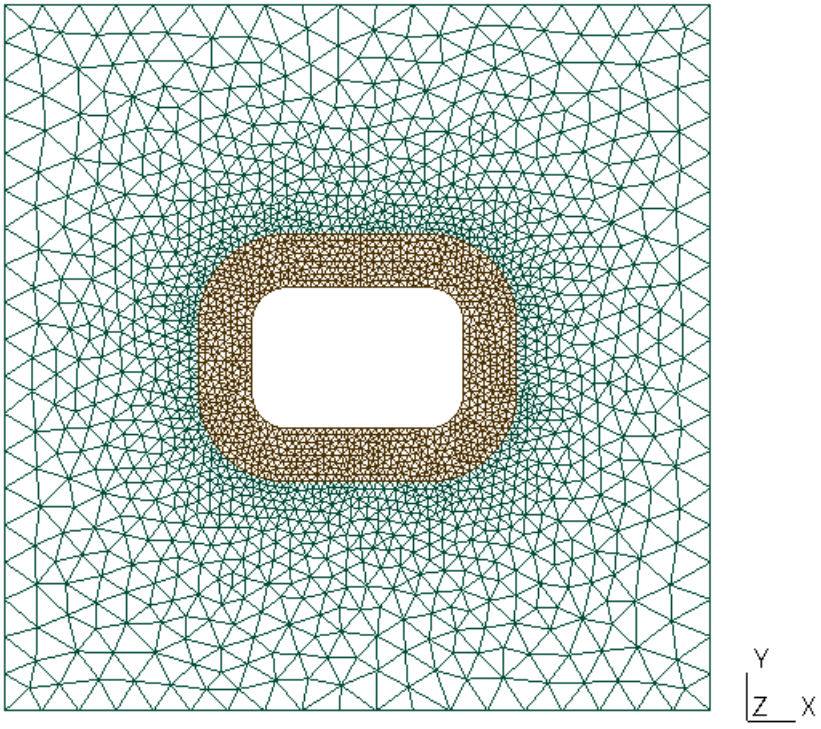


Fig. D.1: Maillage simple généré avec gmsh

À l'aide des variables *lc1* et *lc2* on peut paramétrer la taille des triangles (dans nos cas test du chapitre 6, nous avons pris les valeurs $lc1 = 0.02$ et $lc2 = 0.02$). Les variables (l_x, l_x, r_out_x, r_out_y, r_in_x, r_in_y) permettent de définir le contour et de limiter la zone de recouvrement Ω_R et la zone normale $\Omega_C - \Omega_R$ (voir la figure 2.1 page 13). Les autres variables (c_out_x, c_out_y, c_in_x, c_in_y permettent de définir les centre des cercles). Les différents paramètres de configuration qu'on vien de citer sont shématisés sur la figure D.2.

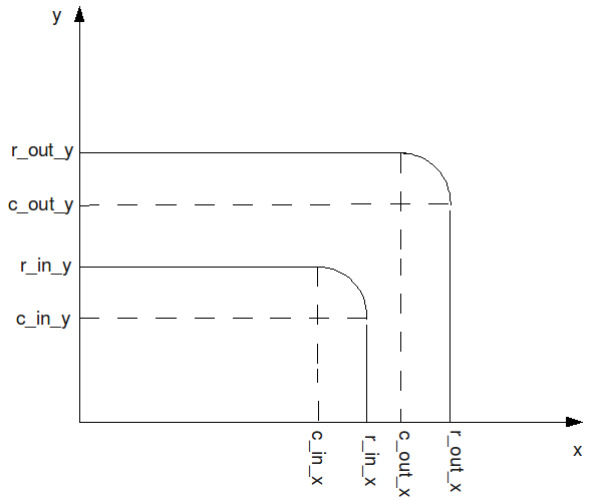


Fig. D.2: Paramètres de configuration de la zone de couplage pour une géometrie complexe.

lc1	:	taille des triangle de la zone de recouvrement Ω_R
lc2	:	taille des triangle de la zone de continue $\Omega_C - \Omega_R$
l_x	:	abscisse du contour exterieur du maillahe
l_y	:	ordonnée du contour exterieur du maillahe
r_out_x	:	abscisse de la limite exterieure de la zone de couplge
r_out_y	:	ordonnée de la limite exterieure de la zone de couplge
r_in_x	:	abscisse de la limite interieure de la zone de couplge
r_in_y	:	ordonnée de la limite interieure de la zone de couplge
c_out_x	:	abscisse du centre de l'arc exterieur de la zone de couplge
c_out_y	:	ordonnée du centre de l'arc exterieur de la zone de couplge
c_in_x	:	abscisse du centre de l'arc interieur de la zone de couplge
c_in_y	:	ordonnée du centre de l'arc interieur de la zone de couplge

```
lc1 = 0.1;
lc2 = 0.02;

l_x = 1;
l_y = 1;

r_out_x = 0.45;
r_out_y = 0.35;

r_in_x = 0.3;
r_in_y = 0.2;

c_out_x = 0.2;
c_out_y = 0.1;

c_in_x = 0.2;
c_in_y = 0.1;

Point(1) = {-l_x , -l_y , 0 , lc1};
Point(2) = {-l_x ,  l_y , 0 , lc1};
Point(3) = { l_x , -l_y , 0 , lc1};
Point(4) = { l_x ,  l_y , 0 , lc1};
```

```
Point(5) = {-c_out_x , -c_out_y , 0 , lc2};
Point(6) = {-c_out_x ,  c_out_y , 0 , lc2};
Point(7) = { c_out_x , -c_out_y , 0 , lc2};
Point(8) = { c_out_x ,  c_out_y , 0 , lc2};
Point(9)  = {-c_in_x , -c_in_y , 0 , lc2};
Point(10) = {-c_in_x ,  c_in_y , 0 , lc2};
Point(11) = { c_in_x , -c_in_y , 0 , lc2};
Point(12) = { c_in_x ,  c_in_y , 0 , lc2};
Point(13)  = {-c_out_x , -r_out_y , 0 , lc2};
Point(14)  = {-c_out_x ,  r_out_y , 0 , lc2};
Point(15)  = { c_out_x , -r_out_y , 0 , lc2};
Point(16)  = { c_out_x ,  r_out_y , 0 , lc2};
Point(17)  = {-r_out_x , -c_out_y , 0 , lc2};
Point(18)  = {-r_out_x ,  c_out_y , 0 , lc2};
Point(19)  = { r_out_x , -c_out_y , 0 , lc2};
Point(20)  = { r_out_x ,  c_out_y , 0 , lc2};
Point(21)  = {-c_in_x , -r_in_y , 0 , lc2};
Point(22)  = {-c_in_x ,  r_in_y , 0 , lc2};
Point(23)  = { c_in_x , -r_in_y , 0 , lc2};
Point(24)  = { c_in_x ,  r_in_y , 0 , lc2};
Point(25)  = {-r_in_x , -c_in_y , 0 , lc2};
Point(26)  = {-r_in_x ,  c_in_y , 0 , lc2};
Point(27)  = { r_in_x , -c_in_y , 0 , lc2};
Point(28)  = { r_in_x ,  c_in_y , 0 , lc2};

Line(1) = {3,1};
Line(2) = {1,2};
Line(3) = {2,4};
Line(4) = {4,3};
Line(5) = {15,13};
Circle(6) = {13,5,17};
Line(7) = {17,18};
Circle(8) = {18,6,14};
Line(9)  = {14,16};
Circle(10) = {16,8,20};
Line(11) = {20,19};
Circle(12) = {19,7,15};
Line(13) = {23,21};
Circle(14) = {21,9,25};
Line(15) = {25,26};
Circle(16) = {26,10,22};
Line(17) = {22,24};
Circle(18) = {24,12,28};
Line(19) = {28,27};
Circle(20) = {27,11,23};

Line Loop(21) = {5,6,7,8,9,10,11,12};
Line Loop(22) = {13,14,15,16,17,18,19,20};
Plane Surface(161) = {21,22};
Physical Surface(162) = {161};

Line Loop(24) = {1,2,3,4};
Plane Surface(171) = {24,21};
Physical Surface(172) = {171};
```

D.2 Exemple d'une zone de recouvrement rectangulaire avec demi-cercles

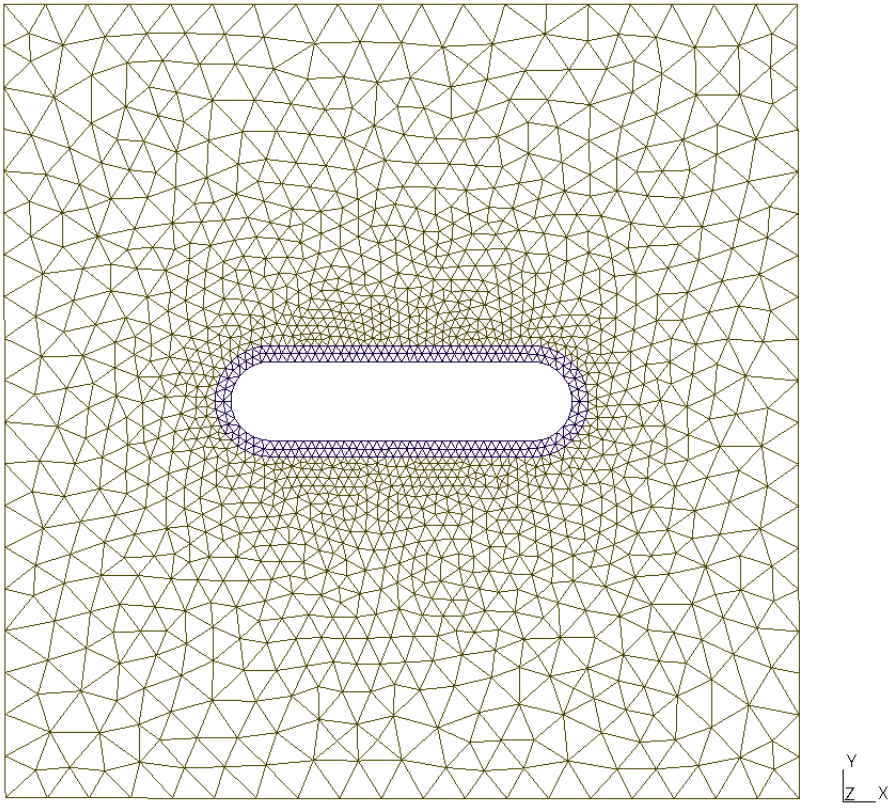


Fig. D.3: Maillage simple généré avec gmsh

La construction de ce maillage est décrite dans la documentation de LibMultiScale [7].

```
lc2 = 0.1;
lc1 = 0.02;

l_x = 1;
l_y = 1;

c_x = 0.33;
c_y = 0;

r_in_x = 0.43;
r_in_y = 0.1;

r_out_x = 0.47;
r_out_y = 0.14;

Point(1) = {-l_x , -l_y, 0  , lc2};
Point(2) = {-l_x ,  l_y , 0 , lc2};
Point(3) = { l_x , -l_y , 0 , lc2};
Point(4) = { l_x ,  l_y , 0 , lc2};
```

```
Point(5) = {-c_x , -r_in_y , 0 , lc1};
Point(6) = {-c_x ,  r_in_y , 0 , lc1};
Point(7) = { c_x , -r_in_y , 0 , lc1};
Point(8) = { c_x ,  r_in_y , 0 , lc1};

Point(9) =  {-c_x , -r_out_y , 0 , lc1};
Point(10) = {-c_x ,  r_out_y , 0 , lc1};
Point(11) = { c_x , -r_out_y , 0 , lc1};
Point(12) = { c_x ,  r_out_y , 0 , lc1};

Point(13) = {-r_in_x  , c_y , 0 , lc1};
Point(14) = {-r_out_x , c_y , 0 , lc1};
Point(15) = { r_in_x  , c_y , 0 , lc1};
Point(16) = { r_out_x , c_y , 0 , lc1};

Point(17) = {-c_x , c_y , 0 , lc1};
Point(18) = { c_x , c_y , 0 , lc1};

Circle(1) = {8,18,15};
Circle(2) = {15,18,7};
Circle(5) = {10,17,14};
Circle(6) = {6,17,13};
Circle(7) = {13,17,5};
Circle(8) = {9,17,14};
Circle(9) = {11,18,16};
Circle(10) = {12,18,16};

Line(11) = {6,8};
Line(12) = {5,7};
Line(13) = {9,11};
Line(14) = {10,12};

Line Loop(15) = {10,-9,-13,8,-5,14};
Line Loop(16) = {11,1,2,-12,-7,-6};

Plane Surface(17) = {15,16};
Physical Surface(162) = {17};

Line(18) = {4,3};
Line(19) = {1,2};
Line(20) = {2,4};
Line(21) = {3,1};

Line Loop(22) = {18,21,19,20};

Plane Surface(23) = {22,15};
Physical Surface(172) = {23};
```

E

Exemple de fichier de diagnostique généré par notre module

Le fichier suivant contien les résultats des diagnostiques effectué sur un partitionnement d'un maillage décrit par une géometrie simple (voir annexe C). Le partitionnement a été effectué avec *PATOH*.

```
DIAGNOSTIC FILE : ../resultats/prts8_P8_Q1_nV10_nHe10_bV10_bHe10_cHe100_diagnostic.txt

              ***********************************************
              ***********DIAGNOSTICS AND STATISTICS**********
              ***********************************************


CSR GRAPH DIAGNOSTICS
- nb parts = 8
- P = 8
- Q = 1
- normal vertex wght = 10
- normal edge cost = 10
- bridging vertex wght = 10
- bridging edge cost = 10
- nb vertices = 9542
- nb edges = 14173
- edgecut = 2520
- total weights = 95420
- total weights for step 1 (without bridging) = 66580
- total weights for step 2 = 28840
- total part weights in % = [13 13 12 13 13 12 13 13 ]
- average total weight balancing = 12.50 ; sd = 0.00
- step 1 part weights in % (without bridging) = [18 18 10 3 9 18 7 18 ]
- average step 1 weight balancing (without bridging) = 12.50 ; sd = 5.73
- step 2 part weights in % = [0 0 18 35 22 0 26 0 ]
- average step 2 weight balancing = 12.50 ; sd = 13.23
- nbre of FE procs communicating with each FE proc = [3 6 4 4 6 4 3 4 ]
- nbre min of FE procs in communication with 1 FE proc = 3
- nbre max of FE procs in communication with 1 FE proc = 6
- nbre of MD procs communicating with each FE proc = [0 0 2 4 3 0 3 0 ]
- nbre min of MD procs in communication with 1 FE proc = 0
- nbre max of MD procs in communication with 1 FE proc = 4

MESH DIAGNOSTICS
```

```
- nb parts = 8
- P = 8
- Q = 1
- element size = 3
- nb nodes = 4912
- nb cells = 9542
- nb node variables = 2
- nb cell variables = 4
- edgecut = 278
- total weights = 9542
- total part weights in % = [ 12 12 12 12 12 13 13 13 ]
- average total weight balancing = 12.50 ; sd = 0.28
- total weights for step 1 (without bridging) = 6658
- step 1 part weights in % (without bridging) = [ 17 8 17 3 16 18 18 2 ]
- average step 1 weight balancing (without bridging) = 12.50 ; sd = 6.75
- total weights for step 2 = 2884
- step 2 part weights in % = [ 0 21 1 35 4 0 0 39 ]
- average step 2 weight balancing = 12.50 ; sd = 15.57
- nbre of FE procs communicating with each FE proc = [ 2 3 4 4 4 3 2 4 ]
- nbre min of FE procs in communication with 1 FE proc = 2
- nbre max of FE procs in communication with 1 FE proc = 4
- nbre of MD procs communicating with each FE proc = [ 0 2 2 4 2 0 2 4 ]
- nbre min of MD procs in communication with 1 FE proc = 0
- nbre max of MD procs in communication with 1 FE proc = 4

HYPERGRAPH DIAGNOSTICS
- nb parts = 8
- P = 8
- Q = 1
- normal vertex wght = 10
- normal hedge cost = 10
- bridging vertex wght = 10
- bridging hedge cost = 10
- nb vertices = 9542
- nb hedges = 9550
- edgecut = 4780
- total weights = 95420
- total part weights in % = [13 12 13 13 13 13 12 13 ]
- average total weight balancing = 12.50 ; sd = 0.00
- total weights for step 1 (without bridging) = 66580
- step 1 part weights in % (without bridging) = [18 5 8 18 9 18 18 7 ]
- average step 1 weight balancing (without bridging) = 12.50 ; sd = 5.50
- total weights for step 2 = 28840
- step 2 part weights in % = [0 30 24 0 21 0 0 25 ]
- average step 2 weight balancing = 12.50 ; sd = 12.69
- nbre of FE procs communicating with each FE proc = [2 3 4 4 4 4 3 2 ]
- nbre min of FE procs in communication with 1 FE proc = 2
- nbre max of FE procs in communication with 1 FE proc = 4
- nbre of MD procs communicating with each FE proc = [0 3 3 0 3 0 0 2 ]
- nbre min of MD procs in communication with 1 FE proc = 0
- nbre max of MD procs in communication with 1 FE proc = 3

HYPERGRAPH WITH FIX CELLS DIAGNOSTICS
- nb parts = 8
```

```
- P = 8
- Q = 1
- normal vertex wght = 10
- normal edge cost = 10
- bridging vertex wght = 10
- bridging edge cost = 10
- nb vertices (brg hg) = 2884
- nb hyperedges (brg hg) = 2892
- nb vertices = 9542
- nb hyperedges = 9550
- unbalanced factor = 5
- fix unbalanced factor = 5
- brg edgecut = 2520
- fix edgecut = 6810
- total weights = 95420
- total part weights in % = [12 12 13 13 13 13 13 12 ]
- average total weight balancing = 12.50 ; sd = 0.06
- total weights for step 1 (without bridging) = 66580
- step 1 part weights in % (without bridging) = [12 12 13 13 12 12 12 13 ]
- average step 1 weight balancing (without bridging) = 12.50 ; sd = 0.13
- total weights for step 2 = 28840
- step 2 part weights in % = [12 12 12 12 13 13 13 12 ]
- average step 2 weight balancing = 12.50 ; sd = 0.22
- nbre of FE procs communicating with each FE proc = [2 2 1 2 2 2 2 1 ]
- nbre min of FE procs in communication with 1 FE proc = 1
- nbre max of FE procs in communication with 1 FE proc = 2
- nbre of MD procs communicating with each FE proc = [2 2 1 2 3 2 2 1 ]
- nbre min of MD procs in communication with 1 FE proc = 1
- nbre max of MD procs in communication with 1 FE proc = 3

MULTICONST HYPERGRAPH DIAGNOSTICS
- nb parts = 8
- P = 8
- Q = 1
- normal vertex wght = 10
- normal hedge cost = 10
- bridging vertex wght = 10
- bridging hedge cost = 10
- nb vertices = 9542
- nb hedges = 9542
- edgecut = 768
- total weights = 4771
- total part weights in % = [12 12 13 13 12 12 13 13 ]
- average total weight balancing = 12.50 ; sd = 0.24
- total weights for step 1 (without bridging) = 3329
- step 1 part weights in % (without bridging) = [13 12 13 13 12 12 13 13 ]
- average step 1 weight balancing (without bridging) = 12.50 ; sd = 0.27
- total weights for step 2 = 1442
- step 2 part weights in % = [12 12 13 12 13 12 13 13 ]
- average step 2 weight balancing = 12.50 ; sd = 0.30
- nbre of FE procs communicating with each FE proc = [4 5 2 4 6 4 4 7 ]
- nbre min of FE procs in communication with 1 FE proc = 2
- nbre max of FE procs in communication with 1 FE proc = 7
- nbre of MD procs communicating with each FE proc = [3 3 2 4 3 3 5 3 ]
```

```
- nbre min of MD procs in communication with 1 FE proc = 2
- nbre max of MD procs in communication with 1 FE proc = 5
```

F

Comportement de notre module d'équilibrage de charge en fonction des coûts des arêtes/hyper-arêtes et des poids des sommets du graphe/hypergraphe

F.1 Exemple de scripts pour lancer un diagnostique

F.1.1 Enregistrement des mesures

```
# $1 is the path to data directory that contains mesh file without the '/' at the end
# $2 is the path to desired result directory without the '/' at the end
if [ x$1 == "x" ]; then
echo "arg 1 require path to data directory containing msh file (without '/' at the end)"
exit 1
fi
DATADIR=$1
if [ ${DATADIR:$(expr ${#DATADIR} - 1):1} == "/" ]; then
echo "without '/' at the end !"
exit 1
fi
if [ x$2 == "x" ]; then
echo "arg 2 require path to result directory (without '/' at the end)"
exit 1
fi
RESULTDIR=$2
if [ ${RESULTDIR:$(expr ${#RESULTDIR} - 1):1} == "/" ]; then
echo "without '/' at the end !"
exit 1
fi
rm -rf *.dat
for i in $(seq 1 50) do
 ../../src/libmultiscale-gen $DATADIR/mesh_simple.msh 8 8 1 1 1 $i 1 0 $RESULTDIR 1 1 vWgts_HG
done
#bridging edges cost var
for i in $(seq 1 50) do
 ../../src/libmultiscale-gen $DATADIR/mesh_simple.msh 8 8 1 1 1 1 $i 0 $RESULTDIR 1 1 heCosts_HG
done
```

F.1.2 Fichier généré

Colonne	Description
1	Poids des sommets de la zone de non-recouvrement
2	Coût des arêtes/Hyper-Arêtes de la zone de non-recouvrement
3	Poids des sommets de la zone de recouvrement
4	Coût des arêtes/Hyper-Arêtes de la zone de recouvrement
5	Coût des arêtes/Hyper-Arêtes de couplage
6	Équilibrage du poids totale
7	Écart type pour l'équilibrage du poids totale
8	Équilibrage du poids pour la phase 1
9	Écart type pour l'équilibrage du poids pour la phase 1
10	Équilibrage du poids pour la phase 2
11	Écart type pour l'équilibrage du poids pour la phase 2
12	Min communication FE-FE
13	Max communication FE-FE
14	Min communication FE-MD
15	Max communication FE-MD

```
1 1 1 1 0 12.50 0.00 12.50 5.47 12.50 12.61 2 4 0 3
1 1 2 1 0 12.50 0.05 12.50 10.37 12.50 12.02 2 5 0 4
1 1 3 1 0 12.50 0.10 12.50 11.98 12.50 9.21 2 5 1 4
1 1 4 1 0 12.50 0.04 12.50 12.52 12.50 7.19 2 4 2 3
1 1 5 1 0 12.50 0.03 12.50 12.48 12.50 5.77 2 4 2 4
```

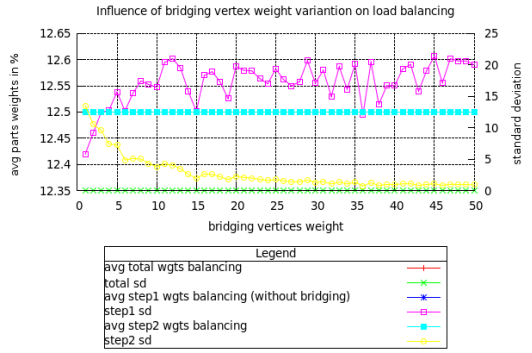
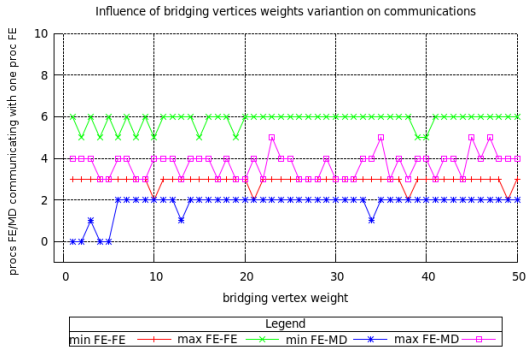
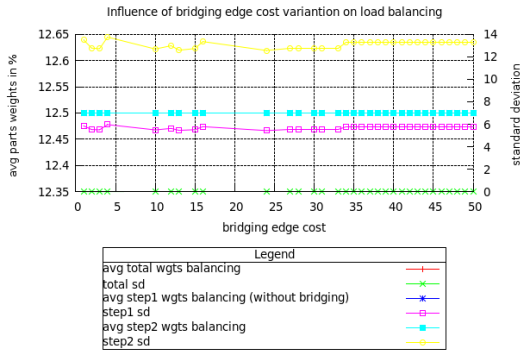
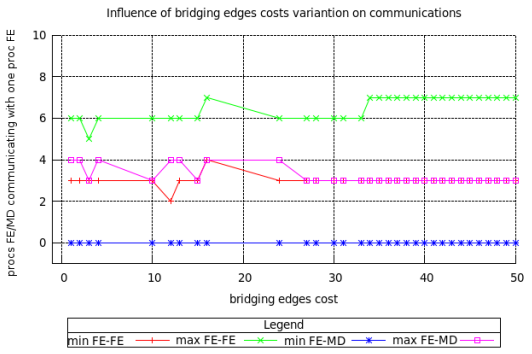
F.1.3 Script pour tracer les courbes avec Gnuplot

```
#tracer avec la commande $> gnuplot plot_wgts.plt -persist
 set title 'Influence of bridging vertex weight and b_hedge/c_hedge cost variantion on load balancing
 set key below Left title 'Legend' box 7
 set autoscale
 set x2range [-1:500]
 unset log
 unset label

 set xtic auto
 set x2tic auto
 set ytic auto
 set y2tic auto
 set grid
 set border 3
 set xtics nomirror
 set x2tics
 set ytics nomirror
 set y2tics

 set xlabel "bridging vertices weight and bridging hedge cost"
 set x2label "coupling hedge cost"
 set ylabel "avg parts weights in %"
 set y2label "standard deviation"

#pour rediriger la sortie vers un fichier eps
#set terminal postscript #portrait color
```

```
#set output 'LoadBalance_vWgtseCostVar.eps'

# tracer avec interpolation lineaire des points
plot "cheCosts-vWgts-heCosts.dat" using 3:6 with linespoints title '...' axes x1y1
replot "cheCosts-vWgts-heCosts.dat" using 3:7 with linespoints title '...' axes x1y2
replot "cheCosts-vWgts-heCosts.dat" using 3:8 with linespoints title '...' axes x1y1
replot "cheCosts-vWgts-heCosts.dat" using 3:9 with linespoints title '...' axes x1y2
replot "cheCosts-vWgts-heCosts.dat" using 3:10 with linespoints title '...' axes x1y1
replot "cheCosts-vWgts-heCosts.dat" using 3:11 with linespoints title '...' axes x1y2
$
```

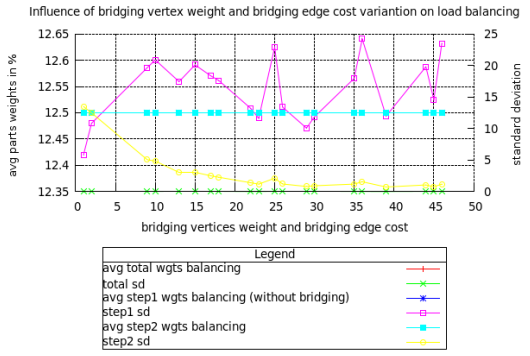
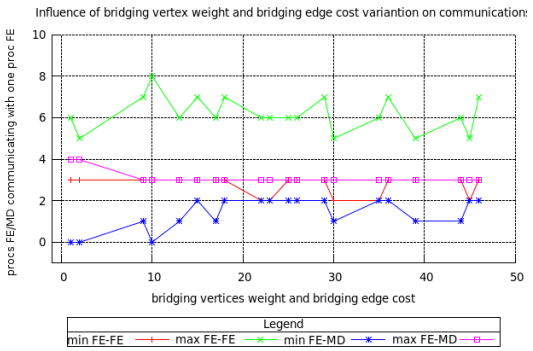
F.2 Résultat des diagnostiques

F.2.1 Cas d'une modélisation avec un graphe

Influence sur l'équilibrage de charge	Influence sur les communications
 Tab F.1 (a) : Influence de la variation du poids des sommets de la zone de recouvrement du graphe sur l'équilibrage de charge.	 Tab F.1 (b) Influence de la variation du poids des sommets de la zone de recouvrement du graphe sur les communications.
 Tab F.1 (c) Influence de la variation du cout des arêtes de la zone de recouvrement du graphe sur l'équilibrage de charge.	 Tab F.1 (d) Influence de la variation du cout des arêtes de la zone de recouvrement du graphe sur les communications.

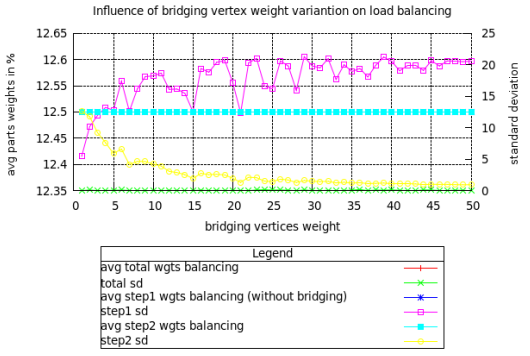
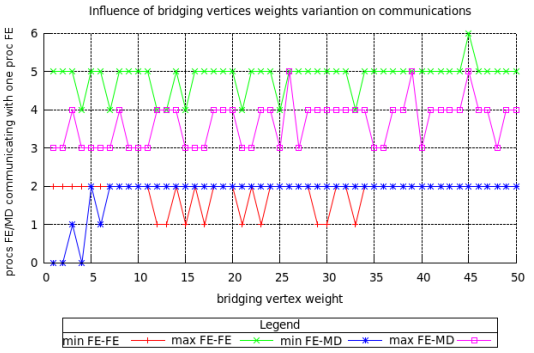
... suite à la page suivante ...

... suite de la page précédente ...

Influence sur l'équilibrage de charge	Influence sur les communications
 Tab F.1 (e) Influence de la variation du poids des sommets et du cout des arêtes de la zone de recouvrement du graphe sur l'équilibrage de charge.	 Tab F.1 (f) Influence de la variation du poids des sommets et du cout des arêtes de la zone de recouvrement du graphe sur les communications.

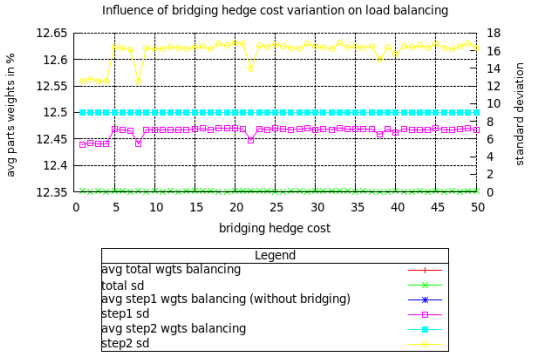
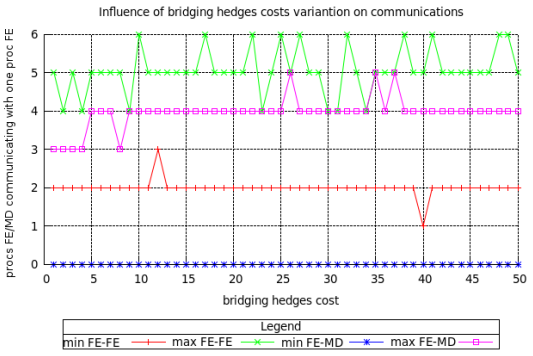
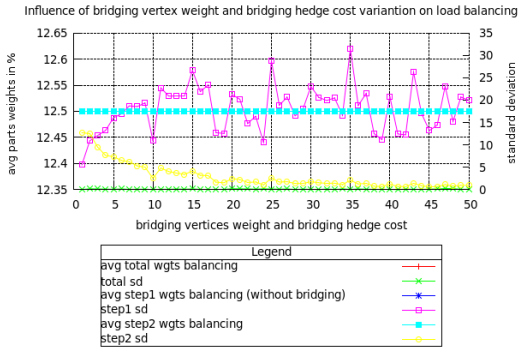
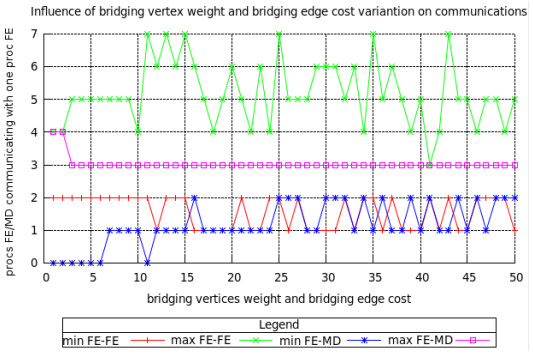
Tab. F.1: Influence de la variation du poids des sommets et du cout des arêtes de la zone de recouvrement du graphe sur l'équilibrage de charge et les communications.

F.2.2 Cas d'une modélisation avec un hypergraphe

Influence sur l'équilibrage de charge	Influence sur les communications
 Tab F.2 (a) : Influence de la variation du poids des sommets de la zone de recouvrement de l'hyper-graphe sur l'équilibrage de charge.	 Tab F.2 (b) : Influence de la variation du poids des sommets de la zone de recouvrement de l'hyper-graphe sur les communications.

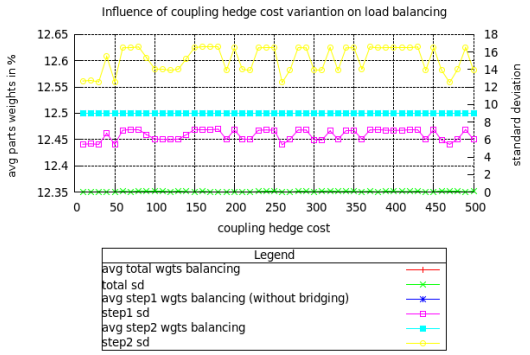
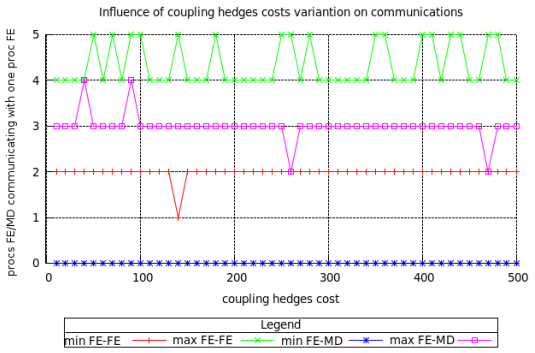
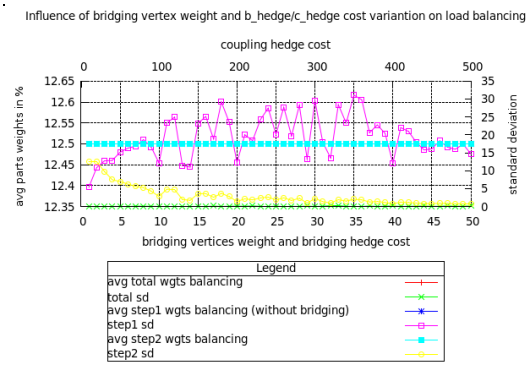
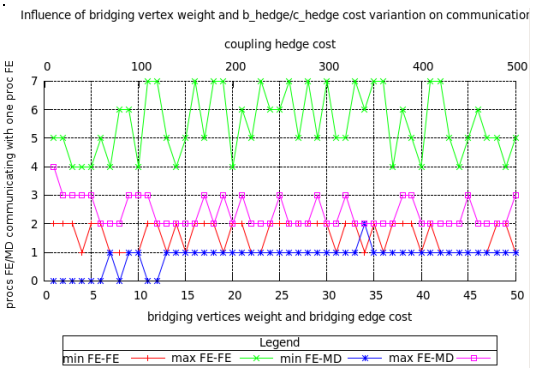
... suite à la page suivante ...

... suite de la page précédente ...

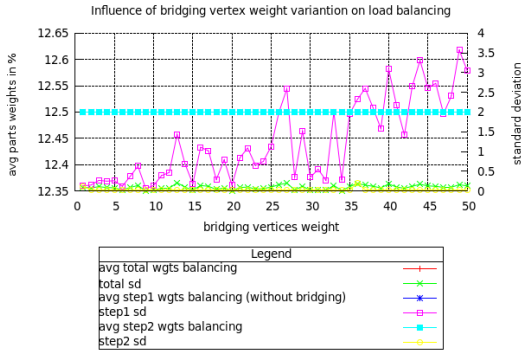
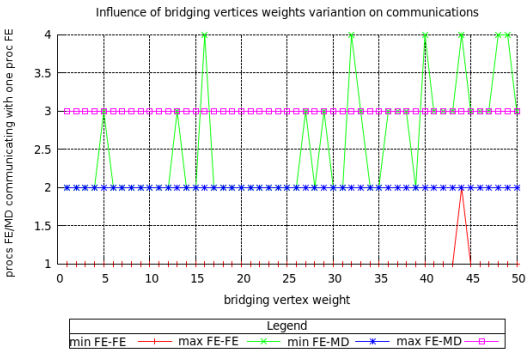
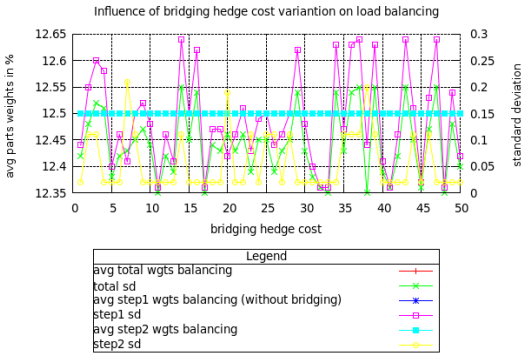
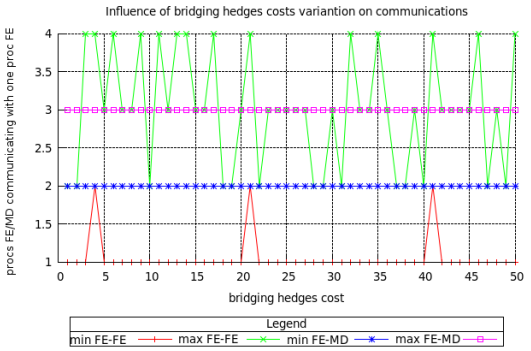
Influence sur l'équilibrage de charge	Influence sur les communications
 Tab F.2 (c) : Influence de la variation du cout des hyper-arêtes de la zone de recouvrement de l'hyper-graphe sur l'équilibrage de charge.	 Tab F.2 (d) : Influence de la variation du cout des hyper-arêtes de la zone de recouvrement de l'hyper-graphe sur les communications.
 Tab F.2 (e) : Influence de la variation du poids des sommets et du cout des hyper-arêtes normales de la zone de recouvrement de l'hyper-graphe sur l'équilibrage de charge.	 Tab F.2 (f) : Influence de la variation du poids des sommets et du cout des hyper-arêtes normales de la zone de recouvrement de l'hyper-graphe sur les communications.

... suite à la page suivante ...

... suite de la page précédente ...

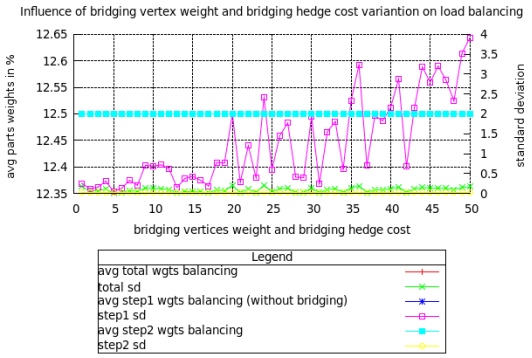
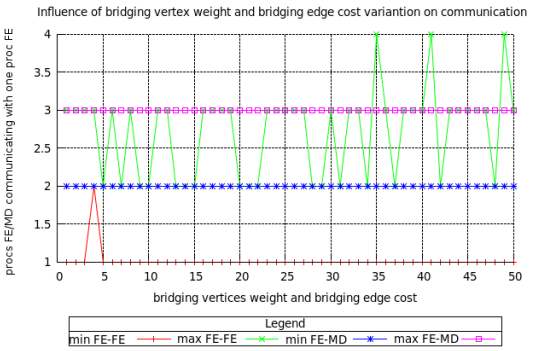
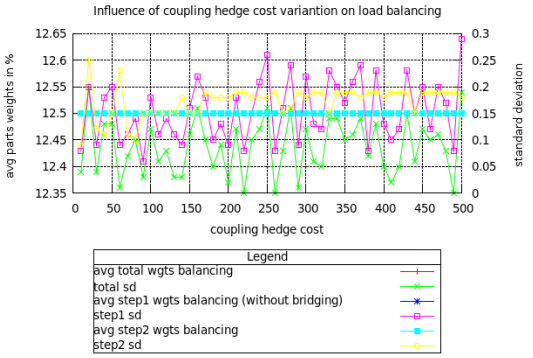
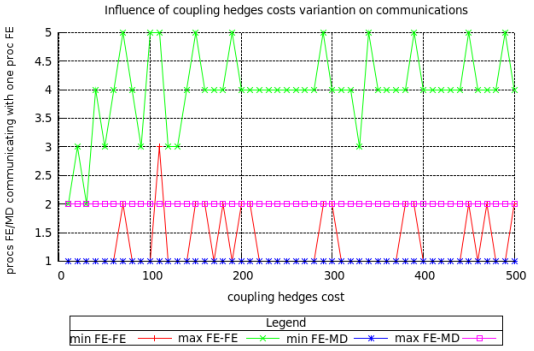
Influence sur l'équilibrage de charge	Influence sur les communications
 Tab F.2 (g) : Influence de la variation du cout des hyper-arêtes de couplage de l'hyper-graphe sur l'équilibrage de charge.	 Tab F.2 (h) : Influence de la variation du cout des hyper-arêtes de couplage de l'hyper-graphe sur les communications.
 Tab F.2 (i) : Influence de la variation du poids des sommets et du cout des hyper-arêtes normales et de couplage de la zone de recouvrement de l'hyper-graphe sur l'équilibrage de charge.	 Tab F.2 (j) : Influence de la variation du poids des sommets et du cout des hyper-arêtes normales et de couplage de la zone de recouvrement de l'hyper-graphe sur les communications.

Tab. F.2: Influence de la variation du poids des sommets et du cout des hyper-arêtes normales et de couplage de la zone de recouvrement de l'hyper-graphe sur l'équilibrage de charge et les communications.

Influence sur l'équilibrage de charge	Influence sur les communications
.  Tab F.3 (a) : Influence de la variation du poids des sommets de la zone de recouvrement de l'hyper-graphe sur l'équilibrage de charge (avec technique des points fixes).	.  Tab F.3 (b) : Influence de la variation du poids des sommets de la zone de recouvrement de l'hyper-graphe sur les communications (avec technique des points fixes).
.  Tab F.3 (c) : Influence de la variation du cout des hyper-arêtes de la zone de recouvrement de l'hyper-graphe sur l'équilibrage de charge (avec technique des points fixes).	.  Tab F.3 (d) : Influence de la variation du cout des hyper-arêtes de la zone de recouvrement de l'hyper-graphe sur les communications (avec technique des points fixes).

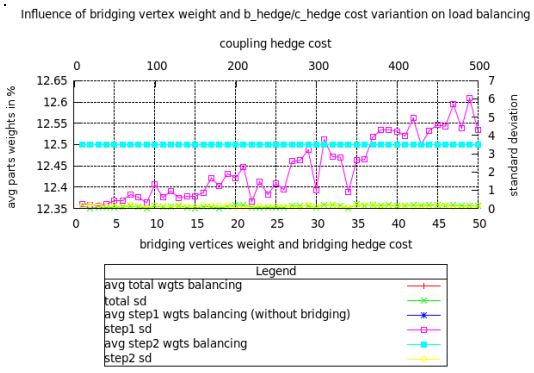
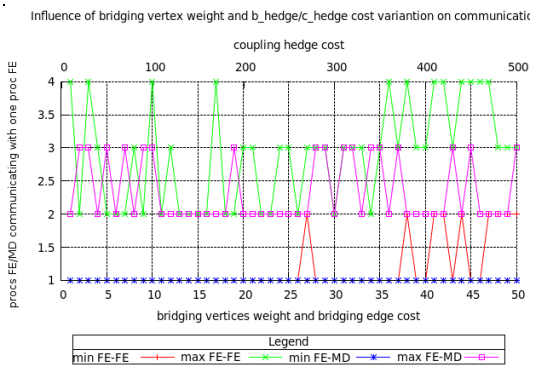
... suite à la page suivante ...

... suite de la page précédente ...

Influence sur l'équilibrage de charge	Influence sur les communications
 Tab F.3 (e) : Influence de la variation du poids des sommets et du cout des hyper-arêtes normales de la zone de recouvrement de l'hyper-graphe sur l'équilibrage de charge (avec technique des points fixes).	 Tab F.3 (f) : Influence de la variation du poids des sommets et du cout des hyper-arêtes normales de la zone de recouvrement de l'hyper-graphe sur les communications (avec technique des points fixes).
 Tab F.3 (g) : Influence de la variation du cout des hyper-arêtes de couplage de l'hyper-graphe sur l'équilibrage de charge (avec technique des points fixes).	 Tab F.3 (h) : Influence de la variation du cout des hyper-arêtes de couplage de l'hyper-graphe sur les communications (avec technique des points fixes).

... suite à la page suivante ...

... suite de la page précédente ...

Influence sur l'équilibrage de charge	Influence sur les communications
 Tab F.3 (i) : Influence de la variation du poids des sommets et du cout des hyper-arêtes normales et de couplage de la zone de recouvrement de l'hyper-graphe sur l'équilibrage de charge (avec technique des points fixes).	 Tab F.3 (j) : Influence de la variation du poids des sommets et du cout des hyper-arêtes normales et de couplage de la zone de recouvrement de l'hyper-graphe sur les communications (avec technique des points fixes).

Tab. F.3: Influence de la variation du poids des sommets et du cout des hyper-arêtes normales et de couplage de la zone de recouvrement de l'hyper-graphe sur l'équilibrage de charge et les communications (avec technique des points fixes).

Bibliographie

[1] Hmetis - serial hypergraph partitioning. http ://glaros.dtc.umn.edu/gkhome/metis/hmetis/publications. Technical report. .

[2] Metis - serial graph partitioning. http ://glaros.dtc.umn.edu/gkhome/metis/metis/publications. Technical report. .

[3] wiki Grid5000 : https ://www.grid5000.fr. .

[4] Wikipédia. http ://www.wikipedia.org/. .

[5] Umit V. Catalyurek al. Hypergraph-based dynamic load balancing for adaptive scientific computations.

[6] M.P. Allen and D.J. Tildesley. Computer simulation of liquids. Oxford University Press, 1989.

[7] Guillaume ANCIAUX. LibMultiScale, AMEL User Guide. Université Bordeaux 1, septembre 2007. .

[8] Guillaume ANCIAUX. Simulation multi-échelles des solides par une approche couplée dynamique moléculaire/éléments finis. De la modélisation à la simulation haute performance. PhD thesis, Université de Bordeaux 1, 2007. .

[9] A. Athanasiadis. Three-dimensional hybrid grid generation and application to Navier-Stokes computations. PhD thesis, Université Libre de Bruxelles, 2004. PhD thesis.

[10] Jean-François Remacle Christophe Geuzaine. Gmsh Reference Manual, http ://geuz.org/gmsh/doc/texinfo/gmsh.html, 2009. .

[11] M. Ortiz E.B. Tadmor and R. Philips. Quasicontinuum analysis of defects in solids. Philosophical Magazine, 73(6) :1529–1563, 1996.

[12] Aurélien ESNARD. Analyse, conception et réalisation d'un environnement pour le pilotage et la visualisation en ligne de simulations numériques paralléles. PhD thesis, Université de Bordeaux 1, 2005.

[13] Aurélien ESNARD. Programmation sur la Grille (Grid'5000). LaBRI, Université Bordeaux 1, 2008. .

[14] Louis Frécon. Éléments de mathématiques discrètes, volume 1.

[15] M. R. Garey and D. S. Johnson. Computers and Intractablility : A Guide to the Theory of NP-Completeness. W.h. freeman, san francisco edition, 1979.

[16] Christophe Geuzaine and Jean-Francois Remacle. Gmsh, March 18 2009. , .

[17] Joseph E. Flaherty James D. Teresco, Karen D. Devine. Partitioning and dynamic load balancing for the numerical solution of partial differential equations.

[18] H. Jin and R.I. Tanner. Generation of unstructured tetrahedral meshes by advancing front technique. International Journal for Numerical Methods in Engineering. pages 1805-1823.

[19] George Karypis Karen D. Devine, Erik G. Boman. Partitioning and load balancing for emerging parallel applications and architectures. Sandia National Laboratories.

[20] Robert T. Heaphy Rob H. Bisseling Umit V. Catalyurek Karen D. Devine, Erik G. Boman. Parallel hypergraph partitioning for scientific computing.

[21] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. Technical report, University of Minnesota, Department of Computer Science, 1998.

[22] Denis LAPOIRE. Graphes et algorithmes. LaBRI, ENSEIRB, Université Bordeaux 1, 2009. .

[23] François PELLEGRINI. Scotch. http ://www.labri.fr/perso/pelegrin/scotch/, 2008.

[24] Parikh Rainald, Lohner and Paresh. Generation if three-dimentional unstructured grid by the advancing-front method. International Journal for Numerical Methods in Fluids. pages 1135-1149.

[25] Parikh Rainald, Lohner and Paresh. Generation if three-dimentional unstructured grid by the advancing-front method. International Journal for Numerical Methods in Fluids, pages 1135–1149, 1988.

[26] SANDIA. Zoltan, http ://www.cs.sandia.gov/zoltan/. http ://www.cs.sandia.gov/Zoltan/.

[27] S.P. Xiao T. Belytschko. Coupling methods for continuum model with molecular model. International Journal for Multiscale Computational Engineering, 2003. 1(1) :115-126, 2003.

[28] Cevdet Aykanat Umit V. Catalyurek. PaToH : Partitioning Tool for Hypergraphs. The Ohio State University Department of Biomedical Informatics, Bilkent University Computer Engineering Department, 1999. .

[29] N.P Weatherill. Delaunay triangulation in computational fluid dynamics. Computational Mathematics and Applications. pages 129-150.

[30] N.P Weatherill. Delaunay triangulation in computational fluid dynamics. Computational Mathematics and Applications, pages 129–150, 1992.

[31] N.P. Weatherill and O. Hassan. Efficient three-dimensional delaunay triangulation with automatic point creation and imposed boundary constraints. International Journal for Numerical Methods in Engineering. pages 2005-2039.

[32] N.P. Weatherill and O. Hassan. Efficient three-dimensional delaunay triangulation with auto-matic point creation and imposed boundary constraints. International Journal for Numerical Methods in Engineering, pages 2005–2039, 1994.

[33] O.C. Zienkiewicz and R.L. Taylor. The finite element method, volume 1. 4 edition.