amU
Aix Marseille Université

NNT : 2024AIXM0001

THÈSE DE DOCTORAT

Soutenue à Aix-Marseille Université
le 13 décembre 2024 par

Alexandre EPALLE

Stratégies parallèles dédiées au raffinement local de maillages structurés en mécanique du contact

Discipline
Sciences pour l'Ingénieur

Spécialité
Mécanique des solides

École doctorale
ED 353 SCIENCES POUR L'INGÉNIEUR : Mécanique, Physique, Micro et Nanoélectronique

Laboratoire/Partenaires de recherche
Laboratoire de Mécanique et d'Acoustique (LMA)
Commissariat à l'énergie atomique et aux énergies alternatives (CEA)

Composition du jury

Jean-François REMACLE Professeur, Université catholique de Louvain	Rapporteur
Vladislav YASTREBOV Chargé de recherche CNRS, Centre des Matériaux, Mines Paris	Rapporteur
Anthony GRAVOUIL Professeur, INSA Lyon	Président du jury
Marc MASSOT Professeur, École Polytechnique	Examinateur
Mikaël BARBOTEU Professeur, Université de Perpignan Via Domitia	Examinateur
Frédéric LEBON Professeur, Aix-Marseille Université	Directeur de thèse
Guillaume LATU Directeur de recherche, CEA Cadarache	Co-Directeur de thèse
Isabelle RAMIÈRE Directrice de recherche, CEA Cadarache	Co-Encadrante de thèse

cea iresne
MISTRAL LabCom
LMA Laboratoire de Mécanique et d'Acoustique

Affidavit

Je soussigné, Alexandre Epalle, déclare par la présente que le travail présenté dans ce manuscrit est mon propre travail, réalisé sous la direction scientifique de Frédéric Lebon et la co-direction de Guillaume Latu, dans le respect des principes d'honnêteté, d'intégrité et de responsabilité inhérents à la mission de recherche. Les travaux de recherche et la rédaction de ce manuscrit ont été réalisés dans le respect à la fois de la charte nationale de déontologie des métiers de la recherche et de la charte d'Aix-Marseille Université relative à la lutte contre le plagiat.

Ce travail n'a pas été précédemment soumis en France ou à l'étranger dans une version identique ou similaire à un organisme examinateur.

Fait à Cadarache le 27/09/2024



Cette œuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 4.0 International.

Liste de publications et/ou brevets et participation aux conférences

Liste des publications et/ou brevets réalisées dans le cadre du projet de thèse :

1) Alexandre Epalle, Isabelle Ramière, Guillaume Latu et Frédéric Lebon. "Parallel simulation and adaptive mesh refinement for 3D elastostatic contact mechanics problems between deformable bodies", soumis, 2024.

Participation aux conférences et écoles d'été au cours de la période de thèse :

1) "Massively Parallel Simulation and Adaptive Mesh Refinement for 3D Elastostatics Contact Mechanics Problems", 11th ECCOMAS Thematic Conference on Adaptive Modeling and Simulation : ADMOS (19 - 21 juin 2023, Göteborg) avec abstract, 2023. Présentation orale.
2) "Combination of adaptive mesh refinement and high performance computing for accurate solution of elastostatics contact mechanics problems", 7th ECCOMAS Thematic Conference on Computational Contact Mechanics : ICCCM (5 - 7 juillet 2023, Turin) avec abstract, 2023. Présentation orale.
3) "Simulation massivement parallèle et raffinement adaptatif de maillage pour les problèmes de mécanique du contact 3D", 16e Colloque national en calcul des structures : CSMA (13 - 17 mai 2024, Giens) avec proceeding, 2024. Présentation orale.
4) "High Performance Computing Strategy for 3D Contact Mechanics Problems with Adaptive Mesh Refinement", 9th European Congress on Computational Methods in Applied Sciences and Engineering : ECCOMAS (3 - 7 juin 2024, Lisbonne) avec abstract, 2024. Présentation orale.
5) "Parallel node-to-surface strategy for 3D contact mechanics problems with adaptive mesh refinement", 16th World Congress on Computational Mechanics and 4th Pan American Congress on Computational Mechanics : WCCM (21 - 26 juillet 2024, Vancouver) avec abstract, 2024. Présentation orale.

Résumé

Cette thèse s'intéresse à la simulation de problèmes de grandes tailles en mécanique du contact et vise à améliorer les performances obtenues par rapport à l'état de l'art. Ce sujet est motivé par la volonté de réaliser des simulations multiphysiques complexes en 3D pour des problèmes issus de l'industrie nucléaire, en particulier le comportement thermo-mécanique sous irradiation des crayons de combustible. L'objectif de ce travail est de mettre en place une stratégie numérique de raffinement adaptatif de maillage (AMR) pour la résolution de problèmes en mécanique du contact élasto-statique sur des calculateurs parallèles. Dans cette thèse, la méthode des éléments finis basée sur des éléments quadrangulaires (2D) ou hexaédriques (3D) est choisie comme méthode de discrétisation.

La première partie de ce travail porte sur une mise en place avancée d'un raffinement de maillage h-adaptatif local en mécanique du contact pour des problèmes avec des frontières de contact courbes. L'usage de l'AMR permet d'atteindre localement des tailles de mailles très fines, inaccessibles avec un raffinement uniforme de maillage. Pour prendre en compte précisément la courbure des géométries de contact au cours du raffinement adaptatif hiérarchique de maillage, des éléments finis super-paramétriques sont choisis. Le problème de contact est quant à lui traité par pénalisation pour limiter le nombre d'inconnues du système à résoudre. Un appariement de type nœud-à-nœud est appliqué pour la détection des éléments en contact. L'algorithme combiné AMR-contact est régi par deux boucles itératives imbriquées : la boucle externe gère le processus AMR, la boucle interne traite de la solution de contact par une méthode des statuts (recherche de contacts actifs).

Pour des simulations 3D impliquant au minimum plusieurs dizaines de millions d'inconnues, le calcul séquentiel ne suffit plus et la parallélisation des calculs s'avère nécessaire. Ainsi, nous proposons trois stratégies de partitionnement de maillage adaptées aux problèmes de contact pour distribuer la charge de calcul entre les processeurs. Une des forces de ces stratégies est que les nœuds de contact appairés sont assignés aux mêmes tâches MPI, réduisant ainsi le nombre d'échanges entre les processus et les surcoûts dus au parallélisme. Elles se distinguent néanmoins par leur manière de répartir les éléments du maillage sur les différents processus. Une adaptation spécifique aux problèmes de contact de la méthode Space Filling Curve, méthode de partitionnement géométrique classique, est notamment introduite. Les performances parallèles de l'algorithme AMR-contact sont évaluées sur des milliers de cœurs de calcul sur des problèmes de contact classiques 3D.

La troisième partie du travail se concentre sur l'extension de l'algorithme parallèle résolvant des problèmes de contact avec AMR à un appariement de type nœud-à-surface dans le but de traiter des géométries et chargements plus complexes. L'enjeu de la parallélisation du contact avec ce type d'appariement réside dans le fait que les nœuds peuvent intervenir dans plusieurs relations de contact en simultanée. Une stratégie de partitionnement de maillage adaptée à ce type de problèmes est proposée et évaluée sur des centaines de cœurs de calcul sur un problème de contact classique 3D.

Mots clés : mécanique du contact, raffinement adaptatif de maillage, calcul haute performance, élasto-staticité, méthode des éléments finis, frontière courbe

Abstract

This PhD thesis focuses on the simulation of large-scale problems in contact mechanics, with the aim of improving performance compared with the state of the art. This topic is motivated by the ambition to perform complex 3D multiphysics simulations for problems derived from the nuclear industry, in particular the thermo-mechanical behavior of fuel rods under irradiation. The goal of this work is to implement an adaptive mesh refinement (AMR) numerical strategy for solving elastostatic contact mechanics problems on parallel computers. In this PhD thesis, the finite element method based on quadrangular (2D) or hexahedral (3D) elements is chosen as the discretization method.

The first part of this work deals with an advanced implementation of local h-adaptive mesh refinement for contact mechanics problems with curved contact boundaries. The use of AMR makes it possible to locally reach very fine mesh sizes, inaccessible with uniform mesh refinement. To accurately discretize the curvature of contact geometries during hierarchical adaptive mesh refinement, super-parametric finite elements are chosen. The contact problem is solutioned by a penalty approach to limit the number of unknowns of the system to be solved. Node-to-node pairing is applied to detect elements in contact. The combined AMR-contact algorithm is ruled by two nested iterative loops : the external loop concerns the AMR process while the internal one deals with the contact solution by an active set method (search for active contacts).

For 3D simulations involving at least several tens of millions of unknowns, sequential computation is no longer sufficient, and parallelization of calculations becomes necessary. Three mesh partitioning strategies dedicated to contact problems are then introduced to distribute the computational load between processors. One of the strengths of these strategies is that paired contact nodes are on the same MPI tasks, thus reducing the number of exchanges between processes and the overheads due to parallelism. However, the approaches differ in the way they distribute the mesh elements between the different processes. In particular, a specific adaptation of the Space Filling Curve method, a classical geometric partitioning method, to contact problems is introduced. The parallel performance of the AMR-contact algorithm is evaluated on thousands of computing cores on classic 3D contact problems.

The third part of the work is devoted to the extension of the parallel algorithm solving contact problems with AMR to node-to-surface pairing, in order to handle more complex geometries and loads. The challenge of contact solution with this type of pairing in a parallel framework lies in the fact that nodes can be involved in several contact relationships. A mesh partitioning strategy suited to this type of problem is proposed and evaluated on hundreds of computing cores on a classic 3D contact problem.

Keywords: contact mechanics, adaptive mesh refinement, high performance computing, elastostaticity, finite element method, curved boundaries

L'éducation est l'arme la plus puissante que l'on puisse utiliser pour changer le monde.

Nelson Mandela
Johannesburg (Afrique du Sud), 16 juillet 2003

À toute ma famille

Préface

La rédaction de cette préface me permet d'exprimer ma profonde gratitude envers toutes les personnes qui ont contribué, de près comme de loin, à la réalisation de cette thèse. Cette préface est probablement le seul endroit dans ce manuscrit où je me livre à cœur ouvert et est un endroit privilégié entre vous, lecteurs, et moi.

La question qu'il me semble la plus importante lorsque l'on réalise un doctorat n'est pas celle liée à notre sujet, scientifique dans mon cas. C'est : pourquoi est-ce que j'effectue un doctorat ? Et c'est à cela que je vais essayer de répondre. Il y a des choses encrées en moi qui font que je réalise un doctorat. Tout d'abord, le fait que je sois quelqu'un de curieux, qui ai toujours envie d'apprendre. J'ai envie d'en apprendre tout d'abord plus sur les thématiques scientifiques qui m'intéressent, mais aussi sur le monde de la recherche. Mais j'ai aussi envie d'en apprendre plus sur moi, sur mes capacités mais aussi mes limites. J'aime les défis qui me permettent, par mon travail, de me prouver à moi-même que je suis capable de mener à bien mon doctorat. J'aime l'inconnu, n'ayant pas de parents docteurs, j'aime m'y confronter et je voulais me montrer que je sais faire preuve de courage en me confrontant à l'inconnu et à la difficulté, aux challenges, et rester calme, lucide et positif même dans mes jours sans. Enfin, je réalise un doctorat parce que j'ai envie de grandir en tant que personne chaque jour et de progresser quotidiennement sur des aspects aussi bien scientifiques qu'humains. Et de me confronter au regard des autres avec bienveillance, sans en avoir peur.

Mais alors, pourquoi réaliser ce doctorat ? Sur ce sujet ? Au Service d'Études et de Simulation du comportement des Combustibles du CEA de Cadarache ? Tout d'abord, parce que le sujet qui m'a été proposé m'a intéressé. Et plus je me suis avancé dans ce doctorat, plus ce sujet pluridisciplinaire comme je l'aime, combinant à la fois les mathématiques appliquées, l'informatique et la mécanique, et générique, car pouvant se décliner pour une multitude d'applications industrielles dans des secteurs d'activités très variés et à fort enjeu comme celui du nucléaire, m'a passionné et m'a donné envie de le creuser autant que possible. Enfin, parce que c'est toujours un honneur de travailler pour mon pays qui m'a tant donné depuis que je suis là.

Je souhaite, en premier lieu, adresser mes plus sincères remerciements à Isabelle Ramière, Guillaume Latu et Frédéric Lebon pour leur collaboration très enrichissante. Je leur suis très reconnaissant pour la confiance qu'ils m'ont accordée et pour m'avoir entraîné dans l'aventure passionnante de cette thèse de doctorat, largement pluridisciplinaire.

Je souhaiterais tout d'abord remercier chaleureusement Isabelle pour sa volonté de toujours viser à me faire progresser et me tirer vers le haut. Ses compétences indéniables et sa disponibilité presque à toute épreuve me resteront gravés pour longtemps.

Je souhaiterais exprimer toute ma gratitude envers Guillaume. Sa capacité à être toujours à l'écoute ainsi qu'à toujours distiller de bon conseils qui ont permis de faire avancer mes travaux sont inestimables et a considérablement enrichi mon expérience de doctorant. Le fait de m'avoir fait découvrir et apprécier le calcul haute performance, domaine qui m'était totalement inconnu au commencement de mon doctorat, traduit ses qualités indéniables de pédagogie.

Je tiens à remercier également Frédéric pour nos discussions passionnantes, pour son soutien et ses conseils toujours judicieux qui ont été très instructifs pour l'avancée des travaux tout au long de ces trois années.

Je tiens à exprimer toute ma reconnaissance à Vladislav Yastrebov et Jean-François Remacle pour leur précieux travail en tant que rapporteurs de ma thèse. Je souhaiterais par ailleurs remercier en particulier Anthony Gravouil d'avoir accepté volontiers de présider mon jury avec bienveillance. Mes remerciements s'adressent également à Mikaël Barboteu et Marc Massot pour les échanges intéressants que nous avons eu au cours de la soutenance. La présence et la contribution de chacun d'eux ont sans aucun doute rendu cet évènement plus agréable.

De nombreuses autres personnes ont contribué au bon déroulement de cette thèse tout au long de ces trois années et je leur en suis reconnaissant. Je souhaiterais remercier le LMA et le CEA pour leur accueil, et en particulier Bruno Collard et Josiane Girard pour le CEA, et Stéphanie Lieutaud pour le LMA qui se sont toujours rendus disponibles quand j'en ai ressenti le besoin. Je souhaiterais remercier nombre de mes collègues, Raphaël Prat, Daria Koliesnikova, Maxence Wangermez, Michel Freyss, Vincent Marino ou encore Gentien Marois, pour ne citer qu'eux, ainsi que l'ensemble du LSC, LDOP et LMCP, pour leur soutien et leur bonne humeur. Je souhaiterais remercier l' ensemble des doctorants, CDD et CDI que j'ai pu côtoyer dont les échanges ont pu être animés mais enrichissants et toujours dans la bonne humeur.

Je souhaite en particulier remercier ceux qui ont partagé le bureau numéro 42 avec moi, un numéro symbolique pour moi puisqu'il s'agit de mon numéro de département d'origine. Je souhaite donc remercier Louis Meyrand, Marius Duvillard et Guillaume Bareight pour avoir mis de la bonne humeur dans ce bureau avec moi. Pour des raisons très différentes, j'ai adoré leur compagnie.

Je souhaite spécialement remercier mes deux « camarades » doctorants, Marius Duvillard et Salomon El Bez, que j'ai pu côtoyer tout au long de ces trois années. J'ai été impressionné par la force de résilience de Marius, qui a plus que mérité de devenir Docteur au vu du déroulé de ces trois années. Je me souviens de la présentation fin 2021 de Carole Valot, que je remercie par ailleurs, où elle a fait le parallèle entre le parcours d'un doctorant et les montagnes russes, et ce parallèle fait sens te concernant. J'ai adoré les randonnées et les moments extraprofessionnels que nous avons pu partager. Salomon est également devenu un ami durant ces trois années de thèse et j'ai apprécié tous les nombreux moments qu'on a pu passer ensemble. En particulier, nos parties de football et de tennis mais aussi nos discussions passionnantes et passionnées sur le football et d'autres sujets me resteront gravées pour longtemps.

Je souhaiterais ensuite remercier mes deux amis, de plus longue date, Julien et Jordan, qui m'ont toujours soutenu et fait l'effort de se lever à pas d'heure pour assister à ma soutenance et entendre quelqu'un en costume leur parler en chinois. Ma sincère gratitude leur est destinée.

C'est avec émotion que je souhaite exprimer mes derniers mots de gratitude à l'ensemble

de ma famille proche pour leur soutien. Grégoire et Arthur, je vous remercie vivement pour les moments heureux que nous partageons depuis 16 et 19 ans. Un merci particulier à mes grands-parents et à mon oncle dont la fierté me comble de bonheur. Et enfin, je souhaiterais remercier le plus chaleureusement possible mes parents pour l'éducation qu'ils m'ont transmise et pour leur soutien et leur amour inconditionnels depuis maintenant plus de 25 ans. Je vous dédie à tous cette thèse.

Table des matières

Liste de publications et/ou brevets et participation aux conférences 3

Résumé 5

Abstract 6

Préface 8

Table des figures 13

Liste des tableaux 19

Introduction 21

1. Contexte, état de l'art et objectifs de la thèse 23
- 1.1. Contexte industriel . . . 24
- 1.2. Motivations de l'étude . . . 28
- 1.3. Mécanique du contact . . . 29
- 1.4. Raffinement adaptatif de maillage (AMR) . . . 36
- 1.5. Calcul Haute Performance (HPC) . . . 39
- 1.6. Solveurs linéaires et pré-conditionneurs . . . 45
- 1.7. État de l'art sur les développements contact-AMR-HPC actuels . . . 49
- 1.8. Cadre de travail . . . 53
- 1.9. Contributions de la thèse . . . 55

2. Raffinement adaptatif de maillage en mécanique du contact avec frontières courbes 59
- 2.1. Modélisation du contact . . . 60
- 2.2. Processus de raffinement adaptatif de maillage . . . 63
- 2.3. Application au problème du contact Hertzien . . . 70
- 2.4. Conclusion . . . 82

3. Partitionnement, équilibrage de charge et performances parallèles 85
- 3.1. Introduction . . . 86
- 3.2. Partitionnement d'un maillage non-conforme dans l'environnement logiciel MFEM . . . 86
- 3.3. Partitionnements séparant éléments de contact et de non-contact . . . 87
- 3.4. Partitionnement dérivé de la méthode Space Filling Curve (SFC) "SFC-contact" . . . 97
- 3.5. Analyse préliminaire sur le cas du contact Hertzien . . . 99
- 3.6. Étude des stratégies de partitionnement sur le contact Hertzien . . . 110
- 3.7. Extension à des problèmes de grande taille . . . 120
- 3.8. Conclusion . . . 128

4. Extension de la stratégie parallèle à un contact nœud-à-surface 131
- 4.1. Algorithme d'appariement nœud-à-surface en 3D avec AMR . . . 131

4.2. Parallélisation de la stratégie 137
4.3. Conclusion 149

Conclusion et perspectives **151**

Annexes **155**

A. Résultats complémentaires sur l'apport des méthodes de partitionnement "pie chart" et "SFC-contact" **156**

B. Résultats complémentaires sur l'empreinte mémoire et les performances solveur **158**

C. Comparaison de l'ordre de la transformation super-paramétrique des éléments **163**

Bibliographie **165**

Table des figures

1.1. Principe de fonctionnement d'un Réacteur à Eau Pressurisée (source : [*Connaissance des Énergies* s. d.]) . . . 25
1.2. Schéma d'un crayon combustible en (a) - Dimensions d'une pastille de combustible en (b) (source de (b) : [Koliesnikova 2021]) . . . 26
1.3. Illustration du comportement d'un élément combustible de REP au cours de l'irradiation : pastilles et gaine à l'état initial à gauche, fracturation des pastilles et mise sous forme diabolo au milieu, puis fermeture complète du jeu entre les pastilles et la gaine à droite (inspirée de [CEA 2008]) . . . 27
1.4. Schéma du problème de contact . . . 30
1.5. Représentation schématique en 2D des appariements nœud-à-nœud en (a) et nœud-à-surface en (b) (inspirées de [Yastrebov 2010]) . . . 31
1.6. Représentation schématique de l'appariement surface-à-surface en 2D (source : [Yastrebov 2010]) . . . 32
1.7. Lois de contact de Signorini en (a) et pénalisée en (b) . . . 33
1.8. Représentation schématique d'un contact pénalisé et détermination de la zone active de contact . . . 34
1.9. Description des principales méthodes AMR (source : [Koliesnikova 2021]) 37
1.10. Application de la méthode de décomposition spatiale de domaine à un domaine rectangulaire 2D (source : [Prat 2019]) . . . 42
1.11. Exemples de partitionnement RCB en dimension 2 et avec des poids de 1 sur les éléments (source : [Boman et al. 2012]) . . . 43
1.12. Exemple de deux courbes de Hilbert en (a) et (b) sur une grille en dimension 2 avec et sans AMR, et d'une courbe en Z sans AMR en (c) (source de (a) et (c) : [Červený et al. 2019]) . . . 44
1.13. Exemple de maillages triangulaires 2D localement raffinés pour des problèmes de contact (source : [Gustafsson et al. 2020]) . . . 50
1.14. Algorithme de raffinement adaptatif de maillage générique (source : [Koliesnikova et al. 2021]) . . . 54

2.1. Raffinement de maillage isotrope quasi-hiérarchique. . . . 67
2.2. AMR hiérarchique pour problèmes de contact à bords courbes avec solution éléments finis d'ordre un : éléments isoparamétriques en (b) versus super-paramétriques en (c). . . . 67
2.3. Exemple d'un maillage raffiné h-adaptatif non-conforme. . . . 68
2.4. Problème du contact Hertzien 2D. . . . 71
2.5. Problème du contact Hertzien 2D : maillages quadrangulaires initial dans (a) et raffiné dans (b). . . . 72
2.6. Problème du contact Hertzien 2D : distribution de la pression de contact sur la zone de contact potentielle. Aperçu général dans (a) et zoom sur le centre de la zone de contact dans (b) et sur la zone de changement de statut de contact dans (c). . . . 73
2.7. Problème du contact Hertzien 2D : influence de k_N et du pas de maillage h_{min} sur l'interpénétration des surfaces. . . . 74

2.8. Problème du contact Hertzien 3D : vue de coupe du maillage initial en (a), maillage final pour la combinaison AMR 1 avec $e_\Omega = 2\%$ et $k_N = 10^2$ E en (b) - ordre de la transformation super-paramétrique de 6. 75
2.9. Distribution de la pression de contact (en Pascal) sur la zone de contact (demi-disques de Hertz 2D) - appariement nœud-à-surface - combinaison AMR 1. 77
2.10. Erreurs de discrétisation (demi-disques de Hertz) en fonction du pas, en (a), et du nombre de nœuds, en (b), du maillage pour des approches de raffinement uniforme et d'AMR. 78
2.11. Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 2\%$). Vue de coupe dans (à gauche) et zoom sur la zone du centre du contact (à droite) - ordre de la transformation super-paramétrique de 6. 80
2.12. Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\text{LOC}} = 7\%$, $\delta = 0,1\%$). Vue de coupe dans (à gauche) et zoom sur la zone du centre du contact (à droite) - ordre de la transformation super-paramétrique de 6. 80
2.13. Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 2\%$). Zones encore marquées pour raffinement à la fin du processus AMR en rouge. Vue de coupe (à gauche) et zoom sur la zone du centre du contact (à droite) - ordre de la transformation super-paramétrique de 6. 81
2.14. Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\text{LOC}} = 7\%$, $\delta = 0,1\%$). Zones encore marquées pour raffinement à la fin du processus AMR en rouge. Vue de coupe (à gauche) et zoom sur la zone du centre du contact (à droite) - ordre de la transformation super-paramétrique de 6. 81
2.15. Contact Hertzien 3D avec deux solides de propriétés mécaniques différentes - combinaison AMR 1 ($e_\Omega = 2\%$). Vue de coupe (à gauche) et zoom sur la zone du centre du contact (à droite) - ordre de la transformation super-paramétrique de 6. 82

3.1. Exemple de courbes de Hilbert SFC en dimension 2 avec et sans AMR (source : [Campbell et al. 2003]). 87
3.2. Illustration du principe du traitement séparé des éléments de contact et de non-contact. 89
3.3. Illustration du principe de partitionnement pour 8 processus MPI. vert = "super-éléments", blanc = éléments de non-contact. Le nombre de cases par processus correspond au nombre de "super-éléments" et d'éléments de non-contact que possèdent les processus après application de l'AMR et avant rééquilibrage des éléments. Le numéro indiqué dans chaque case spécifie le numéro du processus auquel sera attribué le "super-élément" de contact ou l'élément de non-contact. 91
3.4. Illustration de l'équilibrage de charge "equidistributed" basé sur le contact pour un maillage 2D, typique du problème Hertzien - Chaque couleur identifie une tâche MPI spécifique - Partitionnements de maillage lors de la première et de la deuxième itération AMR en (a) et (b) respectivement. . . 92
3.5. Illustration de l'équilibrage de charge "equidistributed" basé sur le contact pour un maillage 3D, typique du problème Hertzien - Couleur représentant la tâche MPI. 93
3.6. Illustration d'un équilibrage de charge spécifique au contact via la stratégie "pie chart" pour un maillage 3D, typique du problème de Hertz. 94

3.7. Partitionnement "pie chart" : exemple de la répartition des éléments après application de l'AMR et avant rééquilibrage en (a), de sous-ensembles obtenus en (b) et de son partitionnement résultant en (c) pour un maillage 2D rectangulaire - Pour la figure en (b), une couleur correspond à un sous-ensemble ; pour les figures en (a) et en (c), une couleur correspond à un processus. 96
3.8. Illustration de l'adaptation du partitionnement SFC au contact. 97
3.9. Illustration d'un équilibrage de charge spécifique au contact via la stratégie "SFC-contact" pour un maillage 3D, typique du problème de Hertz. 99
3.10. Temps de calcul, étude paramétrique pour différentes valeurs de c - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 2\%$). 101
3.11. Résultats des performances parallèles de l'algorithme et de ses principales étapes - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $9,0$ millions - ordre de la transformation super-paramétrique 6. 103
3.12. Résultats des performances parallèles de l'algorithme et de ses principales étapes - Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\mathrm{LOC}} = 4\%$, $\delta = 0,1\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $5,1$ millions - ordre de la transformation super-paramétrique 6. 105
3.13. Résultats des performances parallèles de l'algorithme en (a) et de ses principales étapes : résolution par le solveur en (b), application de l'approche AMR estimate-mark-refine (fonction MFEM `Apply`) en (c) et rééquilibrage du maillage (fonction MFEM `Rebalance`) en (d) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $18,3$ millions - ordre de la transformation super-paramétrique 6. 112
3.14. Résultats des performances parallèles de l'algorithme en (a) et de ses principales étapes : résolution par le solveur en (b), application de l'approche AMR estimate-mark-refine (fonction MFEM `Apply`) en (c) et rééquilibrage du maillage (fonction MFEM `Rebalance`) en (d) - Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\mathrm{LOC}} = 2\%$, $\delta = 0,1$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $39,4$ millions - ordre de la transformation super-paramétrique 6. . . 113
3.15. Temps total passé dans l'étape de résolution du système matriciel (s) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - "pie chart" en pointillés et "SFC-contact" en trait plein - solveur HyprePCG - nombre de degrés de liberté du maillage conforme final : $18,3$ millions - ordre de la transformation super-paramétrique 6. 114
3.16. Empreinte mémoire totale (à gauche et en traits pleins, en GB) et supplémentaire allouée lors des étapes de résolution du système linéaire à la dernière itération AMR (à droite et en pointillés, en % de l'empreinte mémoire totale) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG - nombre de degrés de liberté du maillage conforme final : $18,3$ millions - ordre de la transformation super-paramétrique 6. . . 114

3.17. Nombre d'itérations du solveur HyprePCG avec pour pré-conditionneur HypreILU en (a) et HypreBoomerAMG en (b) - valeurs pour le dernier passage dans l'étape de résolution par le solveur des deux dernières itérations AMR - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - nombre de degrés de liberté du maillage conforme final : $18,3$ millions - ordre de la transformation super-paramétrique 6. 115
3.18. Temps totaux - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG - nombre de degrés de liberté du maillage conforme final : $18,3$ millions - ordres de la transformation super-paramétrique 3 (en pointillés) et 6 (en trait plein). 116
3.19. Empreinte mémoire (GB) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG - nombre de degrés de liberté du maillage conforme final : $18,3$ millions - ordres de la transformation super-paramétrique 3 (en pointillés) et 6 (en trait plein). 117
3.20. Empreinte mémoire totale (à gauche et en traits pleins, en GB) et supplémentaire allouée lors des étapes de résolution du système linéaire à la dernière itération AMR (à droite et en pointillés) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG - nombre de degrés de liberté du maillage conforme final : $18,3$ millions - ordre de la transformation super-paramétrique de 3. 118
3.21. Problème du contact Hertzien 3D : vue de coupe du maillage initial - ordre de la transformation super-paramétrique de 3. 120
3.22. Résultats des performances parallèles de l'algorithme en (a) et des performances du solveur en (b) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 0,5\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $146,9$ millions - ordre de la transformation super-paramétrique de 3. 121
3.23. Contact d'indentation pour des simulations avec $\alpha = 0,008$, en (a) et (b), et $\alpha = 0,03$, en (c) et (d), en vue de coupe - combinaison AMR 1 ($e_\Omega = 4\%$). 123
3.24. Cas test de l'indentation 3D : vue de coupe du maillage initial - éléments isoparamétriques. 124
3.25. Résultats des performances parallèles de l'algorithme en (a) et des performances du solveur en (b) - Indentation 3D - $\alpha = 0,008$ et combinaison AMR 1 ($e_\Omega = 0,9\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $295,9$ millions - ordre transformation 1. 125
3.26. Résultats des performances des étapes principales de l'algorithme (fonctions MFEM `Assemble` en (a), `FormLinearSystem` en (b), `Apply` pour l'application de l'approche estimate-mark-refine en (c) et `Rebalance` pour Rebalance mesh (rééquilibrage du maillage) en (d)) - Indentation 3D - $\alpha = 0,008$ et combinaison AMR 1 ($e_\Omega = 0,9\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $295,9$ millions - ordre transformation 1. 126

4.1. Illustration du principe de l'appariement nœud-à-surface sur un exemple 2D : deuxième étape en (a) et troisième étape en (b) - nœud esclave considéré représenté par un losange orange, nœuds maîtres par des croix vertes et position moyenne des surfaces par des croix bleues obliques. 133

4.2. Problème du contact Hertzien entre disque et rectangle 2D en (a) et entre sphère et pavé 3D en (b) : vue de coupe des maillages quadrangulaire et hexaédrique initiaux - ordre de la transformation super-paramétrique de 3. 134
4.3. Distribution de la pression de contact (en Pascal) sur la zone de contact (contact de Hertz 2D) - appariement nœud-à-surface - combinaison AMR 1. 135
4.4. Distribution de la pression de contact (en Pascal) sur la zone de contact (contact de Hertz 3D) - appariement nœud-à-surface - combinaison AMR 1. 136
4.5. Problème du contact Hertzien 2D : maillage final obtenu pour la combinaison AMR 1 avec $e_\Omega = 4\%$ et $k_N = 10^4$ E - ordre de la transformation super-paramétrique de 3. 136
4.6. Problème du contact Hertzien 3D : maillage final obtenu pour la combinaison AMR 1 avec $e_\Omega = 4\%$ et $k_N = 10^4$ E - Vue de la section de coupe en (a) et en biais en (b) - ordre de la transformation super-paramétrique de 3. . . . 137
4.7. Exemples de stockage de la discrétisation de l'interface maître par extension de la couche fantôme sur 3 processus : cas redondant en (a) et distribué (basé sur une approche appelée "binning scheme") en (b) - $\Gamma_\star^{\text{sl}}$ et $\Gamma_\star^{\text{ma}}$ sont les frontières de contact esclave et maître respectivement - les lignes de couleur en trait plein regroupent les nœuds possédés par chacun des processus initialement ; les lignes en pointillés traduisent l'extension de la couche fantôme élargissant les données disponibles à chaque processus. 138
4.8. Principe de l'extension de la couche fantôme maître d'un niveau - les éléments hachurés (en noir) appartiennent (ainsi que leurs nœuds) au processus courant ; les éléments en damier (en bleu) caractérisent l'extension de la couche fantôme maître dans le processus courant pour prendre en compte de nouveaux ddls fantômes. 139
4.9. Illustration des sous-domaines concordants. 140
4.10. Exemple de partitionnement 3D obtenu à partir de 4 paires de sous-domaines concordants. 141
4.11. Illustration d'un équilibrage de charge spécifique au contact de solides discrétisés avec des maillages non concordants via la stratégie "pie chart", typique du problème de Hertz - figures sans la représentation du maillage en (a) et en (b) ; figures en vue de coupe en (b) et en (c). 143
4.12. Problème du contact Hertzien entre sphère et pavé 3D : vue de coupe du maillage initial - ordre de la transformation super-paramétrique de 3. . . . 144
4.13. Problème du contact Hertzien 3D : maillage final obtenu pour la combinaison AMR 1 avec $e_\Omega = 1,5\%$ et $k_N = 10^4$ E - Vue de la section de coupe en (a) et en biais en (b) - transformation d'ordre 3. 144
4.14. Résultats des performances de l'algorithme et de ses principales étapes - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - solveur HyprePCG ; pré-conditionneur HypreILU avec niveau de remplissage de 1 en (a) et en (b), de 3 en (c) et en (d), et pré-conditionneur HypreBoomerAMG en (e) et en (f) - nombre de degrés de liberté du maillage conforme final : $8,8$ millions - ordre de la transformation super-paramétrique de 3 146
4.15. Contact 3D entre deux cubes étudiés dans [Mayr et al. 2023] - Cas test en (a) et résultats en strong scaling obtenus dans [Mayr et al. 2023] en (b) - Temps total de la résolution du contact en fonction du nombre de cœurs de calcul. 147

4.16. Résultats des performances de l'algorithme pour différents choix de pré-conditionnement - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - solveur HyprePCG ; pré-conditionneur HypreILU avec niveau de remplissage de 1 avec les triangles en orange, de 3 avec les ronds en bleu, et pré-conditionneur HypreBoomerAMG avec les pentagones en vert - nombre de degrés de liberté du maillage conforme final : $8,8$ millions - ordre de la transformation super-paramétrique de 3 148
4.17. Nombre d'itérations de la dernière étape de résolution du système matriciel par le solveur HyprePCG avec les trois configurations étudiées pour le pré-conditionneur - avant-dernière itération AMR en (a) ; dernière itération AMR en (b) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - nombre de degrés de liberté du maillage conforme final : $8,8$ millions - ordre de la transformation super-paramétrique de 3 148

Liste des tableaux

2.1. Influence du coefficient de pénalisation k_N sur la simulation AMR avec une erreur cible globale de $e_\Omega = 2\%$ - Temps moyens sur 3 calculs - Le symbole "-" indique que la valeur est identique à la dernière entrée dans la colonne. 75
2.2. Contact Hertzien 3D - combinaison AMR 1 - résultats en erreurs et nombre d'itérations AMR 79
2.3. Contact Hertzien 3D - combinaison AMR 2 - résultats en erreurs et nombre d'itérations AMR 80

3.1. Empreinte mémoire - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - solveur HyprePCG avec différents pré-conditionneurs - nombre de degrés de liberté du maillage conforme final : $9,0$ millions - ordre de la transformation super-paramétrique 6. 102
3.2. Temps de résolution - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - solveur HyprePCG avec différents pré-conditionneurs - nombre de degrés de liberté du maillage conforme final : $9,0$ millions - ordre de la transformation super-paramétrique 6. 106
3.3. Temps de résolution et nombre d'itérations de HyprePCG avec le pré-conditionneur HypreBoomerAMG - Strong Threshold $= 0,5$ - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - nombre de degrés de liberté du maillage conforme final : $9,0$ millions - 1024 cœurs CPU - ordre de la transformation super-paramétrique 6. 107
3.4. Temps de résolution et nombre d'itérations de HyprePCG avec le pré-conditionneur HypreBoomerAMG - Strong Threshold $= 0,7$ - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - nombre de degrés de liberté du maillage conforme final : $9,0$ millions - 1024 cœurs CPU - ordre de la transformation super-paramétrique 6. 107
3.5. Temps de résolution et nombre d'itérations de HyprePCG avec le pré-conditionneur HypreBoomerAMG - Strong Threshold $= 0,7$ - Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\mathrm{LOC}} = 4\%$, $\delta = 0,1\%$) - nombre de degrés de liberté du maillage conforme final : $5,1$ millions - 512 cœurs CPU - ordre de la transformation super-paramétrique 6. 109
3.6. Temps de résolution et nombre d'itérations de HyprePCG avec le pré-conditionneur HypreBoomerAMG - Strong Threshold $= 0,7$ - Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\mathrm{LOC}} = 4\%$, $\delta = 0,1\%$) - nombre de degrés de liberté du maillage conforme final : $5,1$ millions - 1024 cœurs CPU - ordre de la transformation super-paramétrique 6. 109
3.7. Comparaison des solutions éléments finis pour différents ordres de transformation - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG - Le symbole "-" indique que la valeur est identique à la dernière entrée dans la colonne. 119
3.8. Proportion des temps passés dans les différentes parties du code pour trois simulations différentes - solveur HyprePCG ; pré-conditionneur HypreILU - méthode de partitionnement "SFC-contact" - calculs sur 256 cœurs CPU. . 127

3.9. Proportion des temps passés dans les différentes parties du code pour trois simulations différentes - solveur HyprePCG ; pré-conditionneur HypreILU - méthode de partitionnement "SFC-contact" - calculs sur 4096 cœurs CPU. 127

A.1. Temps de résolution et nombre d'itérations de HyprePCG avec le pré-conditionneur HypreBoomerAMG - StrongThreshold $= 0,5$ - Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\text{LOC}} = 4\%$, $\delta = 0,1\%$) - Strong Threshold $= 0,5$ - nombre de degrés de liberté du maillage conforme final : $5,1$ millions - 512 cœurs CPU - ordre de la transformation super-paramétrique 6. 156
A.2. Temps de résolution et nombre d'itérations de HyprePCG avec le pré-conditionneur HypreBoomerAMG - Strong Threshold $= 0,5$ - Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\text{LOC}} = 4\%$, $\delta = 0,1\%$) - nombre de degrés de liberté du maillage conforme final : $5,1$ millions - 1024 cœurs CPU - ordre de la transformation super-paramétrique 6. 157

B.1. Temps de résolution et nombre d'itérations - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $U = 18,3$ millions - ordre de la transformation super-paramétrique 6. 159
B.2. Temps de résolution et nombre d'itérations - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG ; pré-conditionneur HypreBoomerAMG - Strong Threshold $= 0,7$ - nombre de degrés de liberté du maillage conforme final : $U = 18,3$ millions - ordre de la transformation super-paramétrique 6. 160
B.3. Temps de résolution et nombre d'itérations - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $U = 18,3$ millions - ordre de la transformation super-paramétrique 3. 161
B.4. Temps de résolution et nombre d'itérations - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG ; pré-conditionneur HypreBoomerAMG - Strong Threshold $= 0,7$ - nombre de degrés de liberté du maillage conforme final : $U = 18,3$ millions - ordre de la transformation super-paramétrique 3. 162

C.1. Contact Hertzien 3D ($e_\Omega = 1\%$, 4 cœurs par processus) - solveur HyprePCG ; pré-conditionneur HypreILU - méthode de partitionnement "pie chart" - calculs sur 1024 processus. 163
C.2. Contact Hertzien 3D ($e_\Omega = 1\%$, 4 cœurs par processus) - solveur HyprePCG ; pré-conditionneur HypreILU - méthode de partitionnement "SFC-contact" - calculs sur 2048 processus. 164

Introduction

La simulation numérique s'est imposée comme un outil incontournable dans de multiples secteurs d'activité, tels que l'industrie, les sciences de la vie, les télécommunications ou encore la météorologie. En raison de la forte augmentation de la puissance de calcul et de l'émergence de nouveaux outils et méthodes numériques, il est désormais possible de proposer une modélisation plus précise de phénomènes et de comportements toujours plus complexes. La simulation numérique offre ainsi l'opportunité aux scientifiques de représenter des situations réelles mais aussi de prévoir des comportements potentiels difficiles d'accès par l'expérience.

Dans le secteur de l'industrie nucléaire, il est primordial d'appréhender avec efficacité les phénomènes se produisant durant l'irradiation et la montée en puissance des réacteurs pour des enjeux de sûreté. La caractérisation du comportement du combustible nucléaire, et en particulier de la gaine, première barrière de sûreté nucléaire, représente un enjeu majeur. C'est pour cela que la modélisation précise du comportement mécanique de l'interaction pastille-gaine (IPG) est un sujet d'étude important. Cependant, la forte non-linéarité du phénomène induit de mailler très finement la zone de contact. Toutefois, la taille des maillages uniformément raffinés les mieux adaptés n'est pour le moment pas accessible en raison de coûts en temps de calcul et en espace mémoire trop élevés.

Dans le but d'atteindre des tailles d'éléments suffisamment fines localement sans avoir à raffiner uniformément le maillage, des méthodes de raffinement adaptatif de maillage ont été développées dans la littérature. De manière générale, ces méthodes sont combinées avec des estimateurs d'erreur *a posteriori* pour la détection automatique des zones d'intérêt nécessitant un ajustement du maillage. D'autre part, pour réduire le temps de calcul des simulations, tout en représentant fidèlement les comportements mécaniques, une des orientations actuelles consiste en l'utilisation du calcul parallèle qui permet de répartir les calculs entre les processeurs pour être exécutés en simultanée.

Les méthodes de raffinement de maillage et le calcul parallèle sont ainsi deux approches d'optimisation des performances des simulations visant à allier solutions précises et temps de calcul raisonnables et peuvent s'appliquer de manière complémentaire. Cependant, leur combinaison pour la résolution de problèmes en mécanique du contact entre solides déformables demeure encore un challenge scientifique. En effet, de nombreuses études ont proposé une solution partielle à cet enjeu. D'une part, certains travaux se sont servis des bibliothèques parallèles basées sur la méthode des éléments finis pour implémenter des stratégies de raffinement adaptatif de maillage applicables efficacement dans un cadre de calcul haute performance sans s'intéresser à la modélisation du contact. D'autres études se sont concentrées sur la mise en place d'une solution séquentielle de raffinement adaptatif de maillage pour les problèmes de contact, ou encore sur la parallélisation du contact pour des maillages uniformément raffinés. Enfin, des travaux ont visé à combiner raffinement adaptatif de maillage et calcul parallèle mais uniquement pour des problèmes de contact rigide-déformable.

Le travail présenté dans ce manuscrit vise à combiner stratégies de raffinement adaptatif de maillage, calcul haute performance et mécanique du contact entre solides déformables. Nous nous plaçons dans un cadre de contact élastique sans frottement. Ainsi, nous visons

à optimiser les temps de calcul et à réduire la taille mémoire nécessaire aux simulations. Afin de traiter efficacement diverses configurations en mécanique du contact unilatéral, la méthode de pénalisation, technique d'approximation numérique des problèmes sous contrainte, est appliquée ici car elle permet de régulariser le problème de contact. Une des forces de cette approche est qu'elle ne traite que des inconnues dites "primales" et n'augmente donc pas la taille de la simulation considérée. En ce qui concerne le raffinement adaptatif de maillage, une approche de raffinement hiérarchique de type h-adaptatif local, combinée à un estimateur d'erreur *a posteriori* classique (de Zienkiewicz et Zhu), est implémentée. En outre, l'application d'un raffinement hiérarchique de maillages des solides en contact, ayant le plus souvent des frontières de contact courbes, pose la question de l'approximation de la géométrie de cette dernière lors de l'adaptation automatique du maillage. Dans ces travaux, elle est ainsi bien approximée grâce à l'utilisation d'éléments finis super-paramétriques. Une étude sur la précision de cette approche à la résolution de problèmes de contact avec un appariement de type nœud-à-nœud est menée. Différentes analyses de convergence et études paramétriques permettent de s'assurer de l'efficacité, de la généricité et de l'applicabilité de la stratégie AMR-contact pour des problèmes de contact avec frontières courbes. De plus, dans cette thèse, des stratégies de répartition du maillage éléments finis sur différentes unités de calcul, conçues spécifiquement pour des problèmes de contact, sont mises en œuvre. Elles s'avèrent très performantes pour des simulations comportant plusieurs centaines de millions d'inconnues et utilisant jusqu'à plus de 8000 unités de calcul. Ces travaux représentent ainsi une réelle avancée vis-à-vis des stratégies parallèles de résolution du contact actuellement proposées dans la littérature dont les performances parallèles ne dépassent que très rarement le millier de cœurs de calcul. Une extension de la stratégie proposée au contact entre maillages non concordants via une procédure d'appariement nœud-à-surface est proposée dans le dernier chapitre. Des analyses de convergence permettent de confirmer son efficacité pour des problèmes de contact avec AMR et frontières courbes. Une méthode de distribution parallèle du maillage éléments finis est introduite et se révèle prometteuse pour des simulations comportant plusieurs millions d'inconnues et utilisant des centaines de cœurs. Ce sujet est encore exploratoire et souligne le challenge que représente une parallélisation efficiente dans le cas de l'appariement nœud-à-surface.

Ces travaux de recherche se sont effectués en collaboration entre le CEA de Cadarache et le Laboratoire de Mécanique et d'Acoustique, dans le cadre du laboratoire commun MISTRAL (Modelling Inspection and characterization of materials and STRuctures for Advanced Low carbon energies) entre le CEA, le CNRS, Aix-Marseille Université et l'École Centrale Marseille.

Chapitre 1

Contexte, état de l'art et objectifs de la thèse

Ce chapitre introduit, tout d'abord, le contexte industriel lié à la simulation du comportement du combustible nucléaire. Ensuite, plusieurs solutions numériques et informatiques sont passées en revue : résolution de problèmes en mécanique du contact, raffinement adaptatif de maillage, calcul haute performance, solveurs et pré-conditionneurs. Le positionnement de la thèse dans un contexte scientifique est annoncé vis-à-vis des approches combinant résolution de problèmes en mécanique du contact avec raffinement adaptatif de maillage et calcul parallèle. Enfin, les différents apports de cette thèse sont décrits.

Sommaire

1.1. Contexte industriel . . . 24
1.1.1. Réacteur à eau pressurisée . . . 24
1.1.2. Crayon de combustible . . . 25
1.1.3. Modélisation et simulation des interactions mécaniques . . . 27
1.2. Motivations de l'étude . . . 28
1.3. Mécanique du contact . . . 29
1.3.1. Problème de contact . . . 29
1.3.2. Appariement des surfaces de contact . . . 30
1.3.3. Résolution du problème de contact . . . 33
1.3.4. Méthode des statuts . . . 36
1.4. Raffinement adaptatif de maillage (AMR) . . . 36
1.4.1. Intérêt des maillages structurés . . . 37
1.4.2. Méthodes usuelles de raffinement de maillage . . . 37
1.4.3. Estimation d'erreur . . . 39
1.5. Calcul Haute Performance (HPC) . . . 39
1.5.1. Architecture matérielle . . . 40
1.5.2. Architecture logicielle . . . 41
1.5.3. Méthodes de partitionnement spatial de domaine et d'équilibrage de charge . . . 42
1.6. Solveurs linéaires et pré-conditionneurs . . . 45
1.6.1. Solveurs linéaires directs . . . 45
1.6.2. Solveurs linéaires itératifs . . . 46
1.6.3. Pré-conditionnement . . . 47
1.7. État de l'art sur les développements contact-AMR-HPC actuels . . . 49
1.7.1. Techniques de raffinement adaptatif de maillage appliquées au problème de contact . . . 49
1.7.2. Stratégies de parallélisation adaptées au raffinement adaptatif de maillage . . . 51
1.7.3. Méthodes de parallélisation du problème de contact . . . 52

1.7.4. Parallélisation du problème de contact rigide-déformable avec raffinement adaptatif de maillage . . . 53
1.8. Cadre de travail . . . 53
1.8.1. Choix de l'environnement logiciel de travail . . . 53
1.8.2. Choix de la méthode de raffinement . . . 54
1.8.3. Choix de l'estimateur d'erreur . . . 54
1.8.4. Choix des stratégies de détection et de résolution du contact . . . 55
1.8.5. Caractéristiques du supercalculateur Topaze du CCRT/CEA . . . 55
1.9. Contributions de la thèse . . . 55
1.9.1. Application du contact dans MFEM . . . 56
1.9.2. Adaptation de l'approche AMR dans MFEM . . . 56
1.9.3. Adaptation des méthodes de partitionnement au contact . . . 56
1.9.4. Apports de la thèse . . . 57

1.1. Contexte industriel

D'après les données du Réseau de Transport d'Électricité [1], la part d'électricité produite avec l'énergie nucléaire en France est de 70 % au premier semestre 2024 ; ce qui en fait une source d'énergie importante pour le mix énergétique français actuel.

La compréhension de l'intégralité des phénomènes survenant à l'intérieur d'un réacteur nucléaire en fonctionnement constitue un enjeu d'étude central. Cependant, dans un objectif de prédiction du comportement des installations, la difficile, voire impossible, appréhension des phénomènes dans leur totalité avec des moyens expérimentaux usuels rend l'usage de la modélisation et de la simulation numérique indispensable.

1.1.1. Réacteur à eau pressurisée

Plusieurs filières de réacteurs ont été déployées à travers le monde. Elles se caractérisent par des choix de technologies mais aussi par trois caractéristiques principales distinctes. La première est la composition de la matière fissile composant le combustible nucléaire. Ce sont le plus souvent des pastilles cylindriques de dioxyde d'uranium (UO_2) enrichi de noyaux d'uranium 235 fissiles ou de MOX (Mixed OXide), oxyde mixte d'uranium et de plutonium ($(U,Pu)O_2$). La deuxième est le modérateur (s'il y en a) permettant de freiner les neutrons libérés par la réaction de fission. La troisième est le fluide caloporteur qui vient évacuer la chaleur produite dans le cœur du réacteur pour la transférer à la turbine et à l'alternateur qui la transforment en électricité via un générateur de vapeur.

À l'échelle mondiale, plus de 80% du parc nucléaire en fonctionnement actuellement est constitué de réacteurs issus de la filière où l'eau joue le rôle de fluide caloporteur et modérateur, et 70% sont des Réacteurs à Eau Pressurisée (REP).

Un REP, dont le principe de fonctionnement est illustré figure 1.1, est composé de 3 circuits. Tout d'abord, le circuit primaire, contenu dans le bâtiment réacteur, constitue le cœur du réacteur nucléaire. Il est le lieu où les crayons combustibles sont regroupés en faisceaux dans plus d'une centaine d'assemblages et où les réactions de fission en chaîne exothermiques se produisent. L'énergie libérée, récupérée sous forme de chaleur par l'eau, qui atteint une température de plus de 300°C et qui est maintenue sous forme liquide grâce à une pression très élevée (155 bars). La chaleur de l'eau du circuit primaire est ensuite

1. https ://assets.rte-france.com/prod/public/2024-08/2024-08-02-bilan-s1-2024-fr.pdf

transmise à l'eau du circuit secondaire par échange thermique grâce à un générateur de vapeur. Soumise à une plus faible pression (70 bars), l'eau du circuit secondaire se vaporise et entraîne la rotation d'une turbine, elle-même couplée à un turboalternateur qui convertit l'énergie mécanique en énergie électrique. La vapeur est ensuite re-liquéfiée à l'aide d'un condenseur, refroidi par de l'eau de mer ou de rivière (circuit tertiaire).

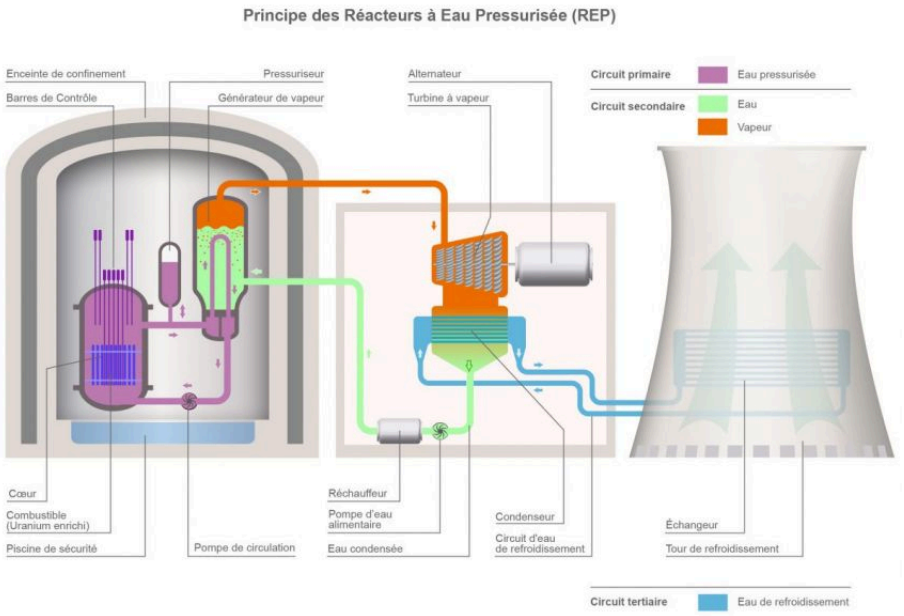


FIGURE 1.1. – Principe de fonctionnement d'un Réacteur à Eau Pressurisée (source : [*Connaissance des Énergies* s. d.])

Pour garantir sa sûreté, le réacteur nucléaire est composé de trois barrières de confinement ayant pour but d'éviter la fuite de produits radioactifs vers l'extérieur : la gaine métallique maintenant les pastilles de combustible nucléaire, l'enveloppe métallique qui emprisonne le fluide caloporteur, et l'armature épaisse en béton du bâtiment réacteur.

1.1.2. Crayon de combustible

Le crayon de combustible est un élément central du réacteur nucléaire. Le crayon de combustible est composé de deux éléments principaux : la pastille de combustible et la gaine. Dans les REP, il possède plusieurs centaines de pastilles de combustibles (généralement 300) empilées les unes sur les autres. Elles sont maintenues radialement par une gaine métallique cylindrique de quelques mètres, et aux extrémités de la gaine par des ressorts de maintien. Le crayon de combustible est schématisé en figure 1.2a. Lors de sa conception, un jeu radial initial entre les pastilles et la gaine d'environ 85 μm est dimensionné dans le but de faciliter l'introduction des pastilles dans la gaine. Lors de sa fabrication, le crayon est pressurisé à l'hélium entre 2 et 2,5 MPa, gaz noble possédant la meilleure conductivité thermique. La gaine entourant les pastilles de combustible, elle constitue la première barrière de sûreté des matières radioactives. Ainsi, la connaissance du comportement du crayon combustible sous irradiation est ainsi primordiale afin de s'assurer de l'intégrité de la gaine.

1.1.2.1. Pastille de combustible

Les pastilles de combustible de REP sont des céramiques qui se présentent sous forme de cylindres de 8,2 mm de diamètre et 13,5 mm de hauteur (voir figure 1.2b). Elles sont produites avec un évidement hémisphérique et un chanfrein à chacune de leurs extrémités dans le but de faciliter leur introduction dans la gaine notamment. Après fabrication, ces pastilles ont une contrainte de rupture comprise entre 100 et 150 MPa (matériau fragile).

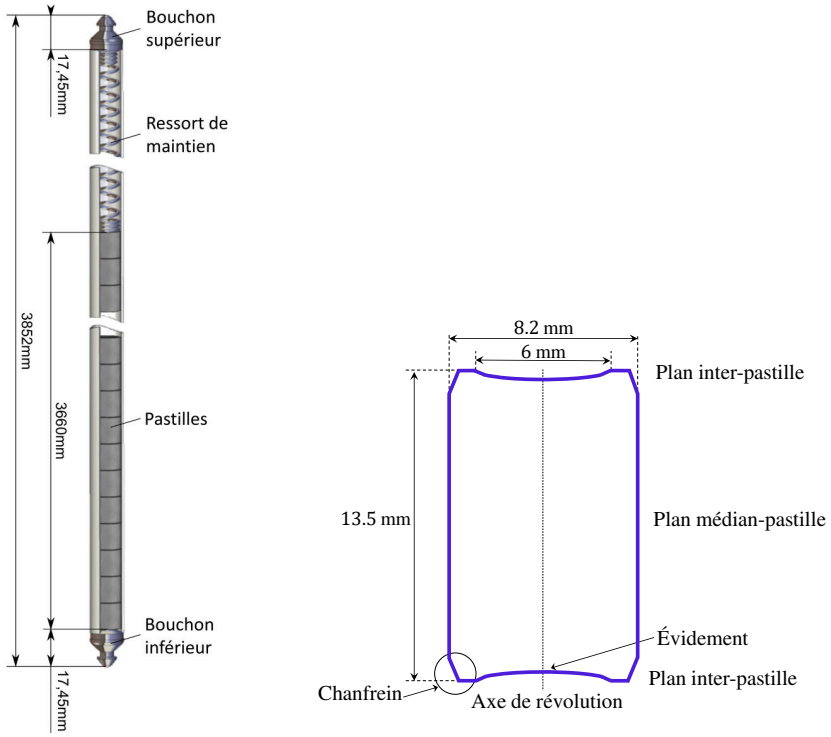


FIGURE 1.2. – Schéma d'un crayon combustible en (a) - Dimensions d'une pastille de combustible en (b) (source de (b) : [KOLIESNIKOVA 2021])

1.1.2.2. Interaction pastille-gaine

Lors du fonctionnement du réacteur, le comportement du crayon combustible est régi par celui des deux éléments qui le compose. Les comportements de la pastille de combustible et de la gaine sont fortement pilotés par l'irradiation en réacteur. Sous l'influence de la haute température présente au sein du cœur du réacteur, la pastille de combustible va se fragmenter dès la première montée en puissance du réacteur sous l'effet des contraintes thermiques. De plus, en début d'irradiation, la pastille de combustible voit sa porosité baisser et se densifie en conséquence. Puis, la pastille tend à se dilater thermiquement. Le fort gradient thermique entre le centre et la périphérie de la pastille induit une dilatation thermique différentielle qui conduit la pastille fragmentée à prendre la forme d'un "diabolo". Par ailleurs, le diamètre de la gaine tend à se réduire par fluage d'irradiation, du fait de la forte pression du fluide caloporteur (155 bars).

Les comportements thermomécaniques des deux composants du crayon combustible entraînent, lors de la première montée en puissance du réacteur, la réduction du jeu pastille-gaine jusqu'à sa fermeture locale via l'apparition de contacts pastille-gaine au niveau des plans inter-pastille. Puis, au cours du cycle de vie du réacteur et de ses diverses

montées en puissance, le jeu entre la pastille et la gaine se ferme complètement et le contact s'étend sur toute la hauteur de la pastille. Ces différents comportements sont illustrés par la figure 1.3.

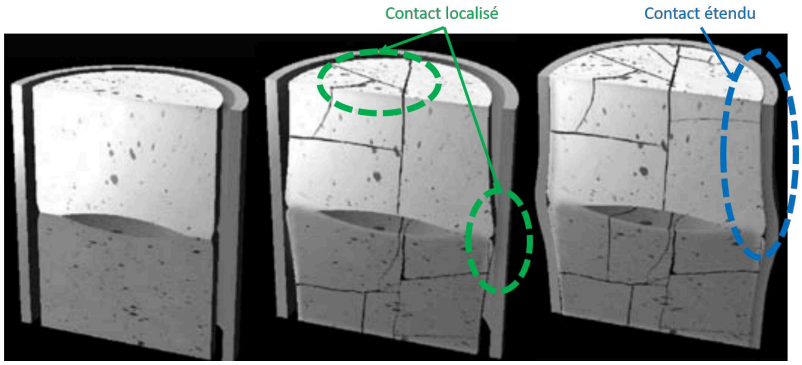


FIGURE 1.3. – Illustration du comportement d'un élément combustible de REP au cours de l'irradiation : pastilles et gaine à l'état initial à gauche, fracturation des pastilles et mise sous forme diabolo au milieu, puis fermeture complète du jeu entre les pastilles et la gaine à droite (inspirée de [CEA 2008])

En fonctionnement nominal, les interactions mécaniques entre les pastilles et la gaine n'induisent pas de rupture de la gaine. Cependant, la probabilité de rupture s'avère variable selon les situations. Ainsi, les règles générales de sûreté visant à garantir qu'aucun produit de radioactif ne soit rejeté vers l'extérieur du réacteur nucléaire et que le refroidissement du combustible soit possible en toute situation, l'intégrité mécanique de la gaine doit être préservée quelle que soit la catégorie de fonctionnement (incidentel et accidentel).

1.1.3. Modélisation et simulation des interactions mécaniques

Pour saisir les phénomènes prédominants qui affectent les performances du combustible, les simulations avancées du combustible nucléaire incluent généralement des formulations de mécanique des solides telles que le fluage, la fissuration, le gonflement gazeux, l'application de conditions limites de pression, des transferts thermiques, mais surtout la modélisation de contacts mécaniques. Il faut aussi noter que le gonflement du combustible et la pression qu'il impose à la gaine lorsqu'ils entrent en contact provoquent une déformation permanente de la gaine. La caractérisation précise de l'interaction entre combustible et gaine d'une part, du contact entre fragments de combustible d'autre part, sont essentielles pour simuler avec précision l'évolution temporelle du crayon combustible en réacteur.

L'interaction mécanique pastille-gaine étant une cause fréquente de la fissuration de la gaine et donc de sa perte d'étanchéité, il est donc primordial de caractériser avec précision le contact entre la pastille et la gaine, en particulier lorsque ce contact est localisé (apparition de concentration de contraintes au niveau du point triple) et donc le jeu non entièrement refermé.

Ainsi, une compréhension fine du comportement de la gaine passe à la fois par une modélisation précise du comportement mécanique de l'interaction (c'est-à-dire ici le contact) pastille-gaine mais aussi par des méthodes, stratégies et solveurs numériques efficaces pour caractériser ce phénomène.

Pour ce faire, une résolution par éléments finis est le plus souvent envisagée. Une modélisation pertinente du contact mécanique frottant est employée et les non-linéarités mécaniques des matériaux constituant la pastille et la gaine sont considérées. De plus, une discrétisation précise du maillage éléments finis, adéquate au problème mécanique

ciblé et pouvant évoluer temporellement, doit être appliquée afin d'optimiser le nombre d'inconnues des simulations réalisées tout en maîtrisant l'erreur commise.

Au CEA, la plateforme de simulation du comportement du combustible sous irradiation PLEIADES [Bernaud et al. 2024] (Plateforme Logicielle pour les Éléments Irradiés dans les Assemblages, en Démonstration, en Expérimentation ou en Service) est utilisée pour la simulations du comportement du combustible sous irradiation. Développée par le CEA et ses partenaires EDF et Framatome depuis 2003, cette plateforme met notamment à disposition des outils de calcul scientifique (OCS), dédiés aux différentes filières de réacteurs et intégrés dans la plateforme, une architecture logicielle, des méthodes numériques, des composants physiques et des lois et propriétés matériaux. L'OCS ALCYONE [Introïni et al. 2024] est consacré au comportement des REP, tandis que GERMINAL [Lainet et al. 2019] à celui des RNR-Na. Enfin, pour la résolution du comportement thermomécanique du crayon de combustible, le solveur Cast3M [CEA 2023] fait office de référence et est appelé par les OCS à cette fin.

1.2. Motivations de l'étude

Il a été montré dans de nombreuses études qu'un maillage fin composé d'éléments de la taille du micromètre, dans la zone critique *a minima*, permet de répondre à l'enjeu de capturer avec précision la concentration de contraintes due au contact au niveau du point triple dans le cas tridimensionnel [Sercombe et al. 2013]. Cela requiert en conséquence l'utilisation de maillages volumineux impliquant un grand nombre d'inconnues dans les simulations.

Les méthodes numériques présentes dans Cast3M n'étant pas particulièrement adaptées à une utilisation sur des architectures multiprocesseurs modernes, les performances parallèles du solveur s'en trouvent très fortement réduites. Par conséquent, l'usage de Cast3M ne nous permet pas de simuler en des temps de calcul acceptables les problèmes que nous souhaitons mettre en œuvre. En effet, le nombre de degrés de liberté maximum que nous pouvons prendre en compte avec Cast3M est de quelques millions, soit plusieurs ordres de grandeurs en dessous de ce qui est souhaité afin d'obtenir des simulations suffisamment précises, sachant qu'il faut notamment considérer la thermomécanique globale des pastilles et de la gaine. Ainsi, dès que les problèmes traités sont de grandes tailles (supérieur à 10 millions d'inconnues), le calcul parallèle s'avère nécessaire afin de s'assurer de coûts en temps de calcul raisonnables. C'est pourquoi, envisager l'utilisation de librairies éléments finis permettant de paralléliser efficacement les stratégies numériques développées fait sens ici dans le but de repousser la limitation induite par Cast3M concernant la taille des problèmes que nous pouvons considérer.

Ainsi, l'objectif global de cette thèse est de concevoir une solution parallèle qui gère aisément à la fois le raffinement adaptatif de maillage et le contact entre solides déformables dans le but de traiter des problèmes comportant un grand nombre de degrés de liberté. Ces travaux doivent, par conséquent, servir de base pour les futurs développements entrepris dans le projet PLEIADES-HPC (projet qui vise à adapter la plateforme PLEIADES pour les supercalculateurs actuels et à venir) mené par le Service d'Études et de Simulation du comportement des Combustibles (SESC) du CEA.

De plus, la modélisation du contact dans une approche combinant à la fois le calcul haute performance et le raffinement adaptatif de maillage dépasse le cadre des applications développées au CEA dans le domaine du combustible et traduit un enjeu scientifique plus global pour des secteurs industriels très divers (automobile, aéronautique notamment).

1.3. Mécanique du contact

Nous avons vu dans les sections 1.1 et 1.2 l'importance d'étudier finement le contact pastille-gaine pour des questions de sûreté des installations. Dans cette section, nous présentons les notions générales et les stratégies numériques les plus courantes pour traiter un problème en mécanique du contact.

1.3.1. Problème de contact

En mécanique des solides déformables, la méthode des éléments finis [Ciarlet 1978] (finite element method FEM en anglais) est la technique de discrétisation et de résolution des équations aux dérivées partielles la plus couramment utilisée. En ce qui concerne la mécanique du contact, cela est également le cas [Wriggers 2006], bien que d'autres approches ont été plus récemment introduites. Nous pouvons citer la VEM [Veiga et al. 2013 ; Wriggers et al. 2023] (virtual element method en anglais) et son extension au contact [Wriggers et al. 2016] et la SBFEM [Song et al. 1997] (scaled boundary finite element method en anglais), qui combine les avantages de la méthode des éléments finis et la méthode des éléments de frontière (boundary element method BEM en anglais) et est récemment adaptée pour l'analyse du contact dans [Hirshikesh et al. 2021]. Dans ce qui suit, la méthode des éléments finis est choisie comme technique de discrétisation.

Deux solides élastiques linéaires recouvrant les domaines $\Omega^s\,(s=1,2) \subset \mathbb{R}^m$ (avec m la dimension de l'espace) sont considérés potentiellement en contact sans frottement sur Γ_C^s, frontières de contact des corps. Une représentation schématique possible est donnée en figure 1.4. Ici, ni l'adhérence ni le glissement par frottement ne sont pris en compte.

Dans $\Omega = \Omega^1 \cup \Omega^2$ (tel que $\Omega^1 \cap \Omega^2 = \emptyset$), le problème de contact unilatéral élasto-statique sans frottement considéré sous l'hypothèse des petites perturbations s'écrit

$$(\mathcal{P}) : \begin{cases} -\mathrm{div}\left(\boldsymbol{\sigma}\left(\mathbf{u}\right)\right) = \mathbf{0} & \text{sur } \Omega \\ \boldsymbol{\varepsilon}\left(\mathbf{u}\right) = \dfrac{1}{2}\left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T\right) & \text{sur } \Omega \\ \boldsymbol{\sigma}\left(\mathbf{u}\right) = \mathbf{C} : \boldsymbol{\varepsilon}\left(\mathbf{u}\right) & \text{sur } \Omega \\ \text{Condition de contact} & \text{sur } \Gamma_C \\ \text{Conditions aux limites} & \text{sur } \Gamma_{BC} \end{cases} \tag{1.1}$$

avec $\mathbf{u}$ la solution du problème en déplacement, $\boldsymbol{\sigma}$ et $\boldsymbol{\varepsilon}$ les tenseurs de contrainte et de déformation respectivement, et $\mathbf{C}$ le tenseur d'élasticité d'ordre 4. Les bords Γ_C et Γ_{BC} sont définis tels que $\begin{cases} \Gamma_{BC} \cup \Gamma_C &= \partial\Omega \quad \text{(frontière de } \Omega) \\ \Gamma_{BC} \cap \Gamma_C &= \emptyset \\ \Gamma_{BC} &= \Gamma_{BC}^1 \cup \Gamma_{BC}^2 \\ \Gamma_C &= \Gamma_C^1 \cup \Gamma_C^2. \end{cases}$

Des conditions aux limites classiques (Dirichlet ou Neumann) sont appliqués sur Γ_{BC} de manière à ce que le problème $(\mathcal{P})$ soit bien posé.

Dans la théorie usuelle de la mécanique du contact, des conditions de contact de Signorini (voir [Moreau 1974 ; Curnier 1999]) sont prescrites sur Γ_C. Elles sont aussi parfois appelées Hertz-Signorini, Hertz-Signorini-Moreau ou Karush-Kuhn-Tucker (en théorie de l'optimisation) selon les sources.

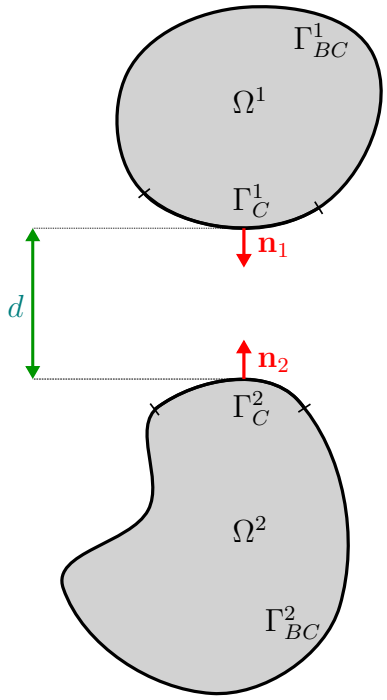


FIGURE 1.4. – Schéma du problème de contact

Elles peuvent s'écrire sous la forme des trois équations scalaires

$$u_N - d \leq 0, \tag{1.2}$$

$$\sigma_N \leq 0\ , \tag{1.3}$$

$$\sigma_N\,(u_N - d) = 0, \tag{1.4}$$

avec $u_N = \mathbf{u}^1 \cdot \mathbf{n}^1 + \mathbf{u}^2 \cdot \mathbf{n}^2$ le déplacement normal, $\sigma_N = (\boldsymbol{\sigma}^1 \cdot \mathbf{n}^1) \cdot \mathbf{n}^1 = (\boldsymbol{\sigma}^2 \cdot \mathbf{n}^2) \cdot \mathbf{n}^2$ la contrainte normale de contact et d la distance initiale (jeu) entre les solides. Nous désignons par $\mathbf{u}^s$ le champ de déplacement dans Ω^s : $\mathbf{u}^s = \mathbf{u}|_{\Omega^s}$, et par $\boldsymbol{\sigma}^s$ le champ de contrainte dans Ω^s : $\boldsymbol{\sigma}^s = \boldsymbol{\sigma}|_{\Omega^s}$. Le symbole '$\cdot$' indique le produit scalaire, tandis que $\mathbf{n}^s$ représente la normale unitaire extérieure à Ω^s sur Γ^s_C. De plus, l'hypothèse des petites perturbations implique que $\mathbf{n} = \mathbf{n}^1 = -\mathbf{n}^2$.

La relation (1.2) traduit la condition de non-interpénétration des solides en contact. L'équation (1.3) permet de s'assurer que le contact est non-adhésif. Enfin, la condition (1.4) est souvent appelée condition de complémentarité et permet d'imposer uniquement deux statuts de contact : contact et décollement.

Les conditions (1.2), (1.3) et (1.4), représentées graphiquement en forces de contact F_N sur la figure 1.7a (page 33), rendent le problème de contact fortement non-régulier et non-linéaire. En effet, quand $u_N - d$ est négatif, il n'y a pas de contact (et par conséquent $\sigma_N = 0$ et $F_N = 0$). Dans le cas contraire, il y a contact (donc $u_N - d = 0$), et σ_N et F_N sont négatifs.

1.3.2. Appariement des surfaces de contact

De manière générale, deux étapes peuvent être distinguées dans la stratégie permettant de résoudre un problème en mécanique du contact : l'étape de détection du contact et

celle de résolution. La première étape consiste à déterminer de manière fiable et robuste quelles parties des bords des deux solides sont susceptibles d'entrer en contact. Pour ce faire, une opération d'appariement de ces bords est réalisée. Cette opération est une tâche algorithmique qui repose sur la recherche des éléments (nœuds, segments ou surfaces d'éléments finis) les plus proches entre les deux solides.

Il existe diverses manières d'appairer les surfaces de contact telles que les appariements nœud-à-nœud, nœud-à-surface ou surface-à-surface. Elles sont décrites dans cette section.

1.3.2.1. Appariement nœud-à-nœud

La technique d'appariement de type nœud-à-nœud [Francavilla et al. 1975], dont une représentation schématique est donnée figure 1.5a, consiste à définir sur les deux bords de contact les paires de nœuds les plus proches. Très simple d'implémentation et d'utilisation, elle requiert l'usage de maillages concordants. Cela signifie, d'une part, que ces maillages doivent avoir le même nombre de nœuds situés sur les bords de contact. D'autre part, ils doivent rester en vis-à-vis y compris en configuration déformée ; ce qui limite le problème à de petites déformations et de petits glissements. Cette technique d'appariement satisfait le patch test de contact de Taylor [R. L. Taylor et al. 1991 ; El-Abbasi et al. 2001] qui vérifie que les forces de contact se transmettent correctement d'un solide à l'autre.

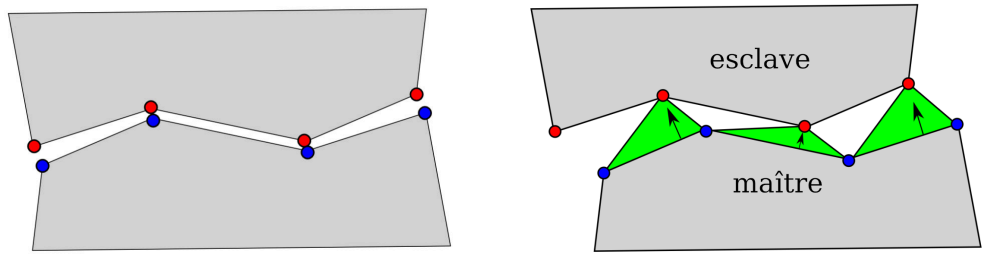


Figure 1.5. – Représentation schématique en 2D des appariements nœud-à-nœud en (a) et nœud-à-surface en (b) (inspirées de [Yastrebov 2010])

1.3.2.2. Appariement nœud-à-surface

La méthode d'appariement nœud-à-surface (aussi appelée nœud-à-facette ou encore nœud-à-segment en 2D) [Hughes et al. 1977] est la plus répandue pour la détection du contact. Avec cette technique, une des deux surfaces est choisie maître et l'autre esclave. Cette méthode permet de traiter des maillages non concordants (en particulier non-symétriques) en 2D et 3D, et de gérer les problèmes avec grandes déformations et glissement. Avec cette stratégie d'appariement, une recherche de minimum (en distance) entre les nœuds du bord de contact esclave et les éléments de l'interface maître est conduite et il y a autant de conditions de contact que de nœuds sur le bord esclave. Son principe est illustré sur la figure 1.5b.

De plus, les bords maître et esclave ne sont pas choisis aléatoirement car un choix incorrect mène à des solutions physiquement erronées. En général, trois règles sont utilisées pour cette définition.

— Règle n°1 : la surface englobante est toujours choisie maître.

— Règle n°2 : s'il n'y a pas de surface englobante, alors l'interface de contact ayant la plus grande densité d'éléments finis est toujours considérée esclave.
— Règle n°3 : si la règle n°2 ne permet pas de statuer (les densités sont égales), alors est défini esclave le bord qui se déforme le plus.

Néanmoins, cette technique d'appariement n'est pas stable (potentielle présence d'interpénétrations parasites quelle que soit la méthode de résolution employée) et ne vérifie pas le patch test de Taylor dans le cas de maillages non concordants. Afin de faire face à cette limitation, plusieurs travaux ont été entrepris comme ceux de Hallquist et al. [Hallquist et al. 1985] et de Zavarise et al. [Zavarise et al. 2009].

D'autre part, la comparaison de la distance de chaque nœud du bord esclave avec l'ensemble (ou plus généralement une sous-partie) des surfaces des éléments situés sur le bord maître peut entraîner une complexité algorithmique élevée (impliquant donc des coûts en temps de calcul importants), en particulier lorsque l'appariement doit être mis à jour fréquemment, par exemple suite à l'application d'un raffinement localisé du maillage.

Enfin, son implémentation dans un contexte de calcul haute performance reste un défi scientifique et les développements à ce sujet (voir [Attaway et al. 1998 ; Mayr et al. 2023] par exemple) se limitent à quelques centaines de cœurs de calcul.

1.3.2.3. Autres approches

La technique d'appariement de type surface-à-surface, introduite par Simo et al. [Simo et al. 1985] pour le cas 2D, est également parfois utilisée pour la détection du contact. Afin de l'étendre efficacement à des problèmes 3D, elle est souvent couplée avec la méthode mortar (le lecteur pourra se référer à [Bernardi et al. 1990 ; Belgacem et al. 1998 ; Devitt et al. 2000] en ce qui concerne la méthode mortar et son application en mécanique du contact). En plus de présenter les mêmes avantages que pour la stratégie d'appariement de type nœud-à-surface, la technique d'appariement surface-à-surface vérifie le patch test de Taylor et permet d'éviter les problèmes d'interpénétrations parasites. Néanmoins, en plus d'être fortement intrusive dans la formulation du problème éléments finis, sa combinaison avec la méthode mortar reste compliquée à implémenter. Pour plus de détails, le lecteur peut se référer à [Puso 2004 ; Wriggers 2006]. Cette technique d'appariement est décrite schématiquement avec la figure 1.6.

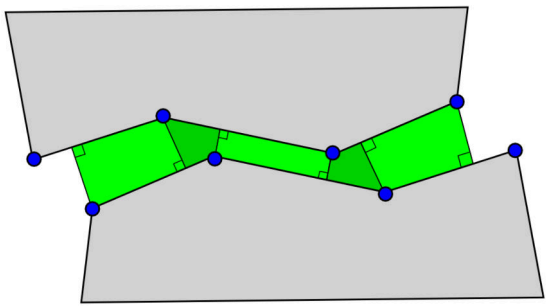


Figure 1.6. – Représentation schématique de l'appariement surface-à-surface en 2D (source : [Yastrebov 2010])

1.3.2.4. Détection du contact dans un cadre de calcul haute performance

Par ailleurs, la détection du contact est une tâche très coûteuse en temps d'exécution et sa parallélisation efficace reste un enjeu. En effet, comme les zones de contact sont le

plus souvent inconnues (car définies via la solution du problème de contact), sa détection efficiente dépend fortement des hypothèses du problème. Ainsi, cette étape de détection implique généralement d'équilibrer la charge dynamiquement mais son optimisation algorithmique dans un cadre parallèle et générique reste complexe. Des travaux se sont penchés sur cet enjeu et nous pouvons citer notamment [Attaway et al. 1998 ; Yastrebov et al. 2011 ; Chien et al. 2018 ; Liu et al. 2021] ou encore plus récemment [Motte et al. 2024].

1.3.3. Résolution du problème de contact

De nombreuses techniques ont été développées pour caractériser avec précision le contact entre deux corps déformables en prenant en compte des contraintes aux interfaces de contact [Kikuchi et al. 1988 ; Laursen 2002 ; Wriggers 2006]. Quand la méthode des éléments finis est utilisée, la méthode de pénalisation, la méthode des multiplicateurs de Lagrange, celle du Lagrangien augmenté sont communément employées pour concevoir des algorithmes robustes et efficaces pour résoudre des problèmes de contact. Pour plus de détails sur ces stratégies, le lecteur peut se référer à [Wriggers 2006].

1.3.3.1. Méthode de pénalisation

Le principe de pénalisation [Kikuchi et al. 1981] de la condition de contact consiste à ajouter au problème un potentiel pénalisant l'interpénétration des deux surfaces. En pratique, cela revient à imposer à chaque nœud élément fini des deux surfaces en contact une force proportionnelle à l'interpénétration de ces surfaces. Cette force est pilotée par le paramètre de pénalisation, généralement noté k_N, et permet de régulariser le problème de contact. En effet, dans la partie correspondant au contact de la figure 1.7b, la pente de la fonction n'est plus infinie mais est désormais égale au coefficient de pénalisation k_N. De plus, le problème à résoudre devient un problème à égalité variationnelle.

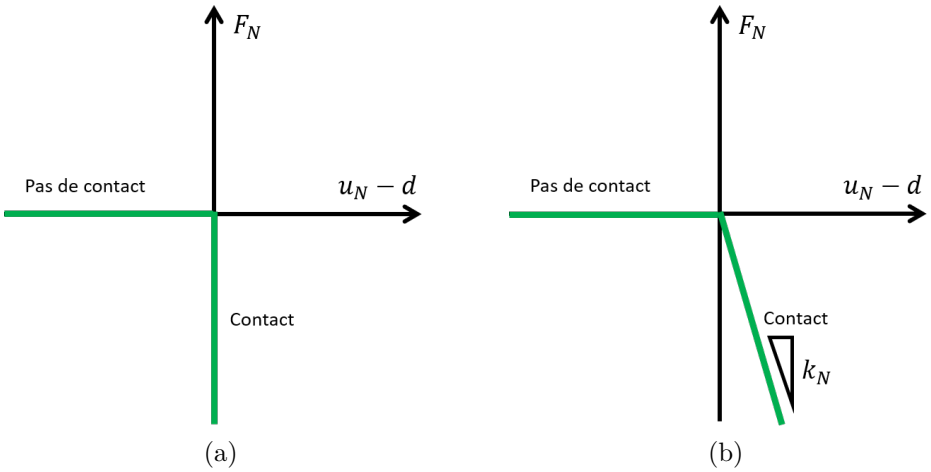


Figure 1.7. – Lois de contact de Signorini en (a) et pénalisée en (b)

Cette approche de résolution du problème de contact est très simple à implémenter dans un code de calcul. Elle ne modifie pas la formulation du problème de manière intrusive et n'augmente pas le nombre d'inconnues du problème. Cependant, elle présente l'inconvénient de ne pas vérifier exactement les conditions de contact de Signorini. De plus, le choix d'un paramètre de pénalisation optimal reste la principale limitation de

cette méthode. En effet, l'interpénétration, induite par la méthode et schématisée dans la figure 1.8, est directement reliée au coefficient de pénalisation k_N et tend vers zéro dans un cas asymptotique uniquement, c'est-à-dire quand k_N tend vers l'infini. Ce coefficient doit donc être suffisamment grand pour réduire l'interpénétration des surfaces. Cependant, une valeur trop élevée de ce coefficient produit un mauvais conditionnement de la matrice du système à résoudre (en implicite), entraînant des problèmes de convergence dans la résolution du problème pénalisé.

Pour réduire la valeur du coefficient de pénalisation tout en contrôlant l'interpénétration, des études ont été réalisées. Le lecteur pourra se référer aux études proposées dans [Chouly et al. 2013b ; Zavarise 2015] en guise d'exemple.

La méthode de la pénalisation a été l'une des premières à être utilisée pour résoudre des problèmes en mécanique du contact et est encore aujourd'hui répandue dans les codes industriels.

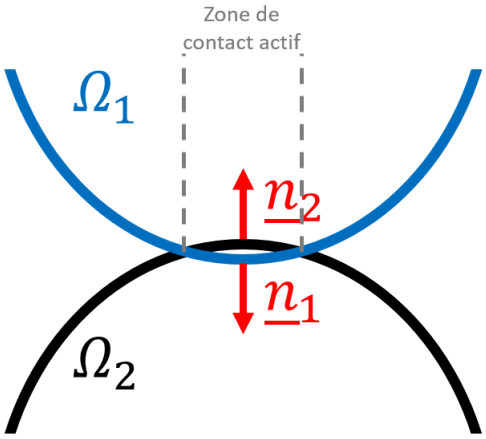


Figure 1.8. – Représentation schématique d'un contact pénalisé et détermination de la zone active de contact

1.3.3.2. Méthode des multiplicateurs de Lagrange

Dans un souci de respecter avec exactitude les conditions de Signorini, la méthode des multiplicateurs de Lagrange [Kikuchi et al. 1988] a été introduite.

Elle implique de nouvelles variables, les multiplicateurs de Lagrange, notés le plus souvent $\boldsymbol{\lambda}$. Ils caractérisent les forces de contact empêchant toute interpénétration des solides en contact. Ces multiplicateurs peuvent être décomposés en deux parties : une normale, souvent notée λ_N, traduisant la composante normale du contact appliquée aux solides et une tangentielle, souvent notée λ_T, caractérisant le frottement entre les solides (dans le cas d'un contact frottant).

En pratique, l'emploi de multiplicateurs de Lagrange permet de transformer un problème de minimisation sous contraintes en un problème de point-selle avec contrainte de non négativité sur les multiplicateurs.

Par ailleurs, même si la technique de résolution par les multiplicateurs de Lagrange permet de résoudre le problème de contact avec précision par la formation du problème de point-selle, et ce sans la présence de paramètres à ajuster, cette stratégie ajoute de nouvelles variables (non-régulières et fortement contraintes) au problème ; ce qui engendre des coûts en matière de temps de calcul supplémentaires non négligeables. Cependant, les divers avantages de cette approche font d'elle l'une des plus utilisées pour la résolution

de problèmes en mécanique du contact, en particulier dans les codes de calcul industriels pour les calculs implicites.

1.3.3.3. Méthode du Lagrangien augmenté

La formulation du Lagrangien augmenté [Simo et al. 1992] peut être vue comme une méthode de multiplicateurs de Lagrange régularisée par une stratégie de pénalisation. En définissant un paramètre de pénalisation, aussi parfois appelé paramètre d'augmentation, de valeur finie, cette approche permet de résoudre avec exactitude les conditions de contact traduites dans un problème entièrement non contraint issu d'une fonctionnelle à minimiser régulière. Ceci rend cette méthode ainsi plus facile à utiliser en pratique que celle des multiplicateurs de Lagrange (avec l'utilisation d'une procédure algorithmique dédiée).

Les travaux présentés dans [Simo et al. 1985 ; Alart 1988 ; Glowinski et al. 1989 ; Alart et al. 1991 ; Simo et al. 1992] ont été précurseurs dans l'application de la technique du Lagrangien augmenté à des problèmes en mécanique du contact avec ou sans frottement. Elle a ensuite été étendue à la résolution de problèmes de contact soumis à de grandes déformations (voir les travaux décrits dans [Pietrzak 1997 ; Pietrzak et al. 1999] par exemple). Enfin, la définition d'une valeur convenable du coefficient de pénalisation au cours de la résolution (majeur pour la convergence de la méthode) a fait l'objet d'études (voir les travaux exposés dans [Mijar et al. 2004a ; Mijar et al. 2004b ; Bussetta et al. 2009]).

1.3.3.4. Autres méthodes

D'autres stratégies de résolution des conditions de contact discrétisées sont disponibles dans la littérature, comme par exemple la méthode de Nitsche. Cette approche, initialement proposée en 1971 [Nitsche 1971] pour l'application de conditions aux limites de Dirichlet sans l'utilisation de multiplicateurs de Lagrange, a été étendue à des problèmes en mécanique du contact [Wriggers et al. 2008]. Il s'agit, comme pour l'approche par pénalisation, d'une méthode dite "primale", c'est-à-dire qu'aucune nouvelle inconnue "duale" n'est introduite comme cela est le cas pour les multiplicateurs de Lagrange ou le Lagrangien augmenté. Cette approche est basée sur la modification de la formulation variationnelle du problème éléments finis afin de satisfaire les conditions de contact de Signorini [Chouly et al. 2015]. Néanmoins, son caractère fortement intrusif, mais aussi le fait que cette méthode s'avère plutôt sensible aux paramètres et aux choix des fonctions de forme utilisées, la rend beaucoup moins répandue dans les codes de calcul que les approches de pénalisation ou des multiplicateurs de Lagrange. Pour plus de détails sur cette méthode, le lecteur peut se référer à [Chouly et al. 2013a ; Chouly et al. 2022].

D'autres méthodes spécifiques ont également été introduites dans la littérature pour résoudre un problème de contact. Il est notamment possible de citer celle qui s'appuie sur l'application d'un troisième milieu hyper-élastique fortement déformable qui vient immerger les corps en contact et tend à se rigidifier d'autant plus que les corps se rapprochent. Ce troisième milieu permet donc, lorsqu'il est fortement comprimé dans la zone de contact supposée, de transférer les forces de contact entre les corps. Cette approche présente l'avantage de faciliter grandement la détection du contact entre les deux corps. Néanmoins, l'amélioration de cette approche pour la résolution de problèmes de contact avec frottement reste un sujet de recherche. Pour plus de détails sur cette méthode, le lecteur peut se référer à [Wriggers et al. 2013 ; Frederiksen et al. 2024].

La méthode mortar peut être aussi utilisée pour résoudre des problèmes de contact [Belgacem et al. 1998 ; Devitt et al. 2000]. Inspirée des méthodes de décomposition de domaine, elle consiste à introduire une surface de contact mortar (qui peut être soit une des deux surfaces des solides en contact soit une surface de contact intermédiaire entre les deux solides) et à reformuler le problème de contact afin le résoudre sur cette surface, généralement avec des multiplicateurs de Lagrange. Elle se révèle adaptée lorsque les maillages ne sont pas concordants même si sa formulation est intrusive.

Par ailleurs, des procédures de relaxation peuvent également être employées pour résoudre un problème de contact frottant [Bisegna et al. 2001 ; Bisegna et al. 2004]. Elles consistent à obtenir une écriture duale condensée du problème de contact dont les nouvelles inconnues du problème sont les forces de contact normales et tangentielles. Cette formulation permet ainsi de construire la matrice de rigidité de contact à coût réduit et de faciliter la convergence des simulations. Néanmoins, ces procédures restent limitées à des problèmes de contact de type rigide-déformable.

1.3.4. Méthode des statuts

La méthode de type "statuts" (aussi appelée "active set strategy" en anglais) [Luenberger et al. 1984 ; Murty 1988 ; Luenberger 2003 ; Wriggers 2006] est généralement combinée aux méthodes de résolution (pénalisation, multiplicateurs de Lagrange, etc., voir les sections 1.3.3.1 et 1.3.3.2 par exemple) dans le but de transformer le problème non linéaire de contact en des problèmes linéaires. Elle consiste à définir la zone (définie par les nœuds) comme étant en contact actif, comme illustré sur la figure 1.8, et à résoudre un problème de contact sur cette zone connue *a priori* de contact actif. Après résolution du problème, la zone de contact actif est réactualisée et ce jusqu'à ce que les nœuds détectés en contact actif n'évoluent plus d'une itération à l'autre.

1.4. Raffinement adaptatif de maillage (AMR)

La forte non-linéarité locale des phénomènes découlant de la mécanique du contact, concentrant des contraintes très élevées dans la zone de contact, nous conduisent à devoir utiliser des maillages de pas d'espace très fins pour pouvoir simuler avec précision les phénomènes modélisés. Ainsi, un maillage uniforme fin pour de tels problèmes est encore très coûteux en termes de calcul et d'empreinte mémoire, en particulier pour des géométries 3D. De plus, sa génération et son stockage peuvent se révéler difficiles voire impossibles. Un des moyens de surmonter cette limitation est de s'appuyer sur des méthodes de raffinement adaptatif du maillage (Adaptive Mesh Refinement (AMR) en anglais) [Verfürth 1996]. Ces techniques, bien connues depuis de nombreuses années, permettent d'ajuster localement et automatiquement le maillage à une précision désirée vis-à-vis d'une quantité donnée, réduisant ainsi le nombre total de degrés de liberté par rapport à des maillages uniformément raffinés.

Par ailleurs, un intérêt majeur des techniques de raffinement de maillages est qu'elles permettent d'obtenir des taux de convergence optimaux même pour des problèmes présentant des singularités géométriques [Babuška et al. 1983 ; Hennig et al. 2016] et sont très employées dans la communauté de la mécanique des solides.

1.4.1. Intérêt des maillages structurés

Un maillage éléments finis structuré est un maillage dont chaque nœud intérieur (c'est-à-dire non situé sur les bords) a toujours le même nombre de nœuds voisins.

Les maillages structurés composés d'éléments quadrangulaires en 2D ou hexaédriques en 3D (fréquemment appelés éléments quad/hexa) sont couramment utilisés pour simuler différents problèmes (en mécanique des solides ou des fluides), qu'ils soient académiques ou industriels. Ces éléments sont intéressants en raison de leur structure tensorielle [Belytschko et al. 2003 ; Ramière et al. 2007 ; Arndt et al. 2021], de leur meilleure précision (par rapport aux triangles ou aux tétraèdres) pour le même ordre d'espace d'approximation [Arndt et al. 2021], et offrent de grands avantages pour de nombreux problèmes de mécanique des solides tels que la mécanique du contact [Michel et al. 2013 ; Liu et al. 2017]. Ainsi, se caractérisant par leur simplicité de génération et leur robustesse, des maillages comportant exclusivement des éléments quad/hexa sont considérés dans ces travaux.

1.4.2. Méthodes usuelles de raffinement de maillage

Parmi les nombreuses méthodes AMR développées dans la littérature, les techniques adaptatives les plus répandues pour réduire l'erreur de discrétisation sont les suivantes : h-adaptive, p-adaptive, hp-adaptive et multi-grille locale. Ces méthodes sont illustrées sur la figure 1.9. En outre, le raffinement adaptatif du maillage est généralement couplé à des estimateurs d'erreur *a posteriori* [Verfürth 1996] afin de déterminer automatiquement les éléments à raffiner, et parfois aussi d'arrêter automatiquement le processus de raffinement [O. C. Zienkiewicz et al. 1987 ; Liu et al. 2017].

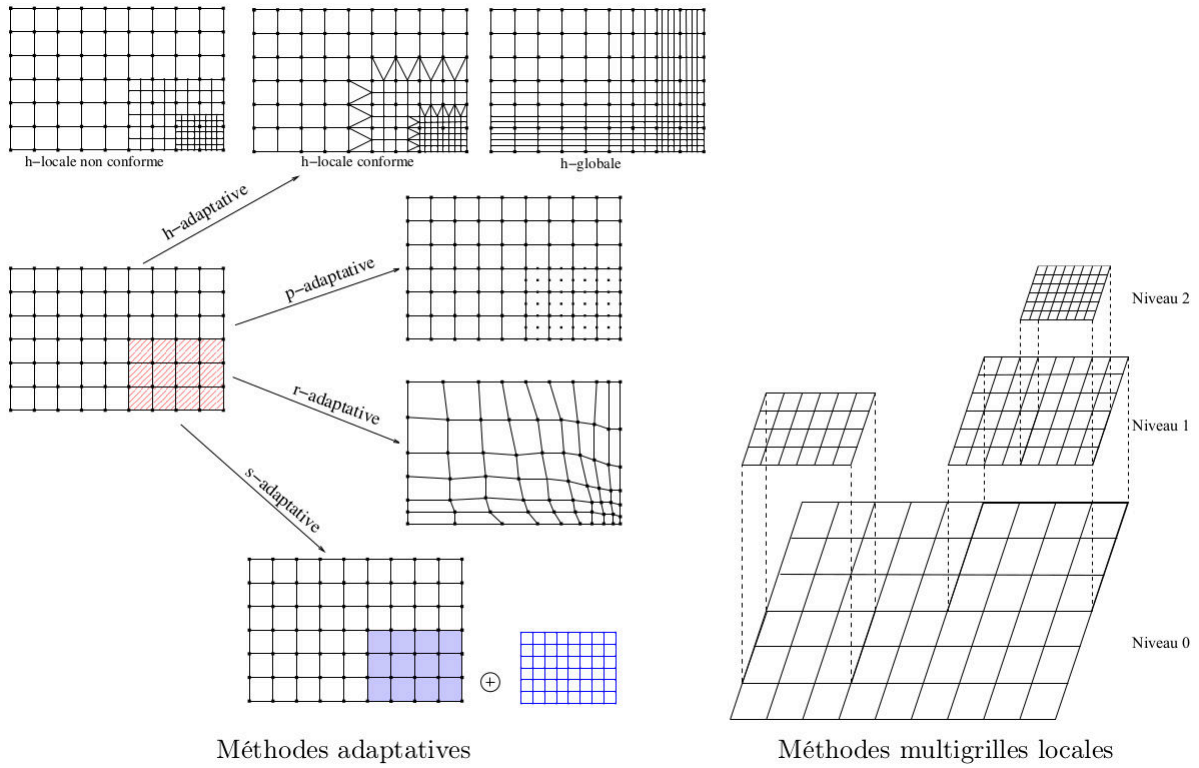


Figure 1.9. – Description des principales méthodes AMR (source : [Koliesnikova 2021])

1.4.2.1. Méthode h-adaptative

Les méthodes h-adaptatives [Demkowicz et al. 1985 ; Belytschko et al. 1993 ; Díez et al. 1999] consistent à raffiner les éléments dans des zones d'intérêt par la création de nouveaux éléments plus fins tout en conservant le degré des fonctions de forme. Elles sont connues pour leur simplicité et leur efficacité ; ce qui en fait des méthodes de raffinement de maillage largement répandues.

La procédure de génération d'un maillage raffiné avec une stratégie de type h-adaptatif la plus connue et mise en œuvre est le schéma de raffinement hiérarchique.

Le raffinement hiérarchique repose sur l'enrichissement du maillage existant par la subdivision des éléments nécessitant d'être raffinés en des éléments plus fins. De plus, le raffinement hiérarchique conduit généralement à des maillages raffinés optimaux [Zhu et al. 1993 ; Ródenas et al. 2017 ; Koliesnikova et al. 2021] par rapport au nombre total de degrés de liberté et permet un meilleur contrôle du processus d'adaptation de maillage grâce à une structure de données hiérarchique explicite. Lors de l'utilisation d'éléments quad/hexa, cette approche AMR conduit à des maillages non-conformes avec des nœuds non-conformes [Koliesnikova et al. 2021] (hanging nodes en anglais) pour lesquels la continuité doit généralement être assurée.

Plusieurs techniques ont été proposées pour éviter l'apparition de non conformités lors du raffinement hiérarchique de maillage mais ces dernières présentent d'autres inconvénients : maillages non optimaux vis-à-vis du nombre de degrés de liberté, combinaison de tous types d'éléments (voir la figure 1.9), dégradation des éléments ou moins aisés à générer ou à gérer dans un code de calcul. Pour plus de détails sur ces approches, le lecteur peut consulter [Schneiders 1998 ; Tchon et al. 2002 ; Ledoux et al. 2009 ; Nicolas et al. 2016].

1.4.2.2. Méthode p-adaptative

L'approche p-adaptative [Babuška et al. 1982 ; Salagame et al. 1997] repose sur l'augmentation locale du degré d'interpolation des fonctions de forme tout en gardant la discrétisation du domaine initial inchangée. Généralement, ce type de raffinement est appliqué avec l'emploi de fonctions de forme hiérarchiques [Szabo et al. 1990 ; Düster et al. 2001] pour assurer la continuité inter-élément.

1.4.2.3. Méthode hp-adaptative

L'approche hp-adaptative [Demkowicz et al. 1989 ; O. Zienkiewicz et al. 1989 ; Mitchell et al. 2014] consiste à raffiner localement le maillage en combinant les méthodes h-adaptative et p-adaptative. Elle est la plus connue et répandue parmi les approches combinées. Elle permet entre autres d'obtenir des taux de convergence exponentiels dans le cas de certains problèmes. Cette méthode requiert cependant l'utilisation de critères renseignant le type de raffinement à appliquer sur les éléments du maillage.

1.4.2.4. Méthode multi-grille locale

La méthode multi-grille locale [Brandt 1977a ; Brandt 1977b ; Bai et al. 1987 ; Khadra et al. 1996] consiste non pas à modifier le maillage initial mais à l'"enrichir" dans les zones d'intérêt en ajoutant des maillages (habituellement uniformes) locaux de pas de maillage de plus en plus fins s'imbriquant généralement les uns dans les autres. Le problème est donc divisé en plusieurs problèmes sur des maillages différents et résolus

séparément. Les solutions sont ensuite reliées entre elles par le biais d'un processus itératif basé sur l'utilisation d'opérateurs de transfert entre niveaux de maillage. Ces opérateurs de transfert présentent l'avantage d'agir lors de la phase de pré- et du post-traitement du calcul.

Parmi les méthodes multi-grilles locales existantes, la méthode de Correction Locale du Résidu (Local Defect Correction (LDC) en anglais) semble être la plus efficace pour l'étude de problèmes en mécanique du solide (avec possiblement du contact) avec des effets localisés résolus par la méthode des éléments finis [Barbié et al. 2014 ; Barbié et al. 2015 ; Liu et al. 2017 ; Koliesnikova et al. 2021].

1.4.2.5. Autres méthodes

En plus des méthodes citées précédemment, d'autres ont été introduites dans la littérature. Par exemple, l'approche r-adaptative [Ghosh et al. 1993 ; Cao et al. 1999] consiste à modifier la position des nœuds pour les relocaliser dans les zones d'intérêt et ainsi réduire l'erreur de discrétisation. L'approche s-adaptative [Fish 1992 ; Yue et al. 2005] (qui présente des similitudes avec les méthodes de type multi-grille locale) revient à enrichir le maillage avec un ensemble de niveaux de plus en plus fins localisés dans les zones d'intérêt, le problème composite (incluant tous les niveaux) étant ensuite résolu. D'autre approches combinatoires, hr-adaptative [Askes et al. 2001 ; Piggott et al. 2005], rp-adaptatives [Nübel et al. 2007 ; Franke et al. 2010] ou encore rs-adaptatives [Ghosh et al. 1996], peuvent également être utilisées pour adapter localement le maillage.

1.4.3. Estimation d'erreur

Un ingrédient clé pour une adaptation automatique du maillage est l'estimateur d'erreur. Son objectif est d'évaluer la qualité d'un calcul éléments finis en vue de l'améliorer.

Les estimations d'erreur *a priori* consistent à prédire l'ordre de convergence de la solution en fonction de la régularité de la solution exacte. Cependant, les estimations *a priori* ne donnent pas la possibilité de contrôler localement la qualité de la solution.

Il existe pour cela des estimateurs d'erreur dits *a posteriori* qui, à partir uniquement d'une solution approchée (obtenue par éléments finis), donne une estimation élémentaire de l'erreur commise. L'intérêt de ce type d'estimation est que l'utilisateur peut fixer le niveau d'erreur souhaité et, si ce seuil d'erreur n'est pas atteint, adapter/raffiner le maillage dans les zones où les erreurs sont les plus fortes. Néanmoins, son inconvénient principal est, qu'en dehors de cas particuliers, qu'il ne donne pas de bornes garanties généralement.

Les principales méthodes d'estimation d'erreur sont les méthodes basées sur les résidus de l'équation d'équilibre [Babuška et al. 1983], celles basées sur l'erreur en relation de comportement [Ladevèze et al. 1997] et celles basées sur le lissage des contraintes [O. C. Zienkiewicz et al. 1992].

Par ailleurs, de nombreux critères sont disponibles pour analyser les performances et l'efficacité des estimateurs d'erreur.

1.5. Calcul Haute Performance (HPC)

Résoudre un problème en mécanique du contact respectant l'exigence d'un maillage raffiné finement tel que décrit en section 1.2 conduit à des simulations très coûteuses en temps de calcul et en espace mémoire. Cela peut même se révéler impossible sur des

ordinateurs standards, ce même avec de l'adaptation locale de maillage. L'usage des supercalculateurs modernes permet de repousser ces limitations.

L'architecture de supercalculateur la plus commune actuellement se compose d'un ensemble de nœuds de calcul interconnectés par un réseau. Chaque nœud est constitué de mémoire vive, de périphériques d'entrée/sortie, d'un processeur (Unité Centrale de Traitement, Central Proccessing Unit ou CPU en anglais) et éventuellement d'un GPU (Unité de Traitement Graphique, Graphics Processing Unit en anglais) ou accélérateur. Chaque processeur est doté de plusieurs cœurs chargés de traiter des flux d'opérations de manière indépendante.

L'intégration d'algorithmes parallèles dans les codes de simulation permet de distribuer la charge entre les différentes unités de calcul du supercalculateur. Les développeurs utilisent des paradigmes de programmation pour mettre en œuvre cette parallélisation. D'une part, les calculs peuvent être répartis entre les nœuds grâce à une parallélisation à mémoire distribuée. D'autre part, à l'intérieur de chaque nœud, la parallélisation à mémoire partagée est souvent privilégiée pour la distribution des calculs entre les cœurs. Enfin, au niveau de chaque cœur, une autre forme de parallélisme, la vectorisation, est employée.

Ainsi, l'idéal visé par la mise en place du calcul parallèle est de réduire par deux le temps de calcul lorsque le nombre d'unités de calcul est multiplié par deux : c'est qui est défini comme le speedup idéal.

Dans cette section, une description succincte de l'architecture des processeurs, des paradigmes de parallélisation mais aussi des approches de partitionnement spatial de domaine et d'équilibrage de charge est proposée. Un zoom est effectué sur ce qui est utile pour comprendre les autres chapitres de ce manuscrit. Pour plus de détails, le lecteur peut notamment se référer aux travaux de thèse présentés dans [Prat 2019].

1.5.1. Architecture matérielle

1.5.1.1. Architecture d'un nœud de calcul

Depuis une vingtaine d'années, l'utilisation de processeurs multicœurs se répand de plus en plus. Chaque cœur est constitué d'ALUs (Unités Arithmétiques et Logiques, arithmetic logic units en anglais, effectuant des opérations élémentaires telles que des additions ou des multiplications), de registres (pour stocker les données récupérées depuis la mémoire vive ainsi que les instructions à exécuter) et de caches mémoire, lui permettant d'exécuter indépendamment des autres cœurs une liste d'instructions.

1.5.1.2. Hiérarchie mémoire des processeurs multicœurs

Avec les avancées technologiques des processeurs, la mémoire qui leur est associée a évolué vers une structure de plus en plus hiérarchique, comportant plusieurs niveaux : mémoire vive mais aussi caches L1, L2, L3, etc. Ces caches ont une capacité de stockage de quelques mégaoctets et offrent des temps d'accès plus courts que pour la mémoire vive. Lorsqu'une donnée est chargée depuis la mémoire vive, elle est automatiquement copiée dans les caches mémoire, car susceptible d'être réutilisée. Plus le niveau de cache est élevé, plus sa capacité de stockage est importante, mais aussi plus la latence d'accès (délai entre

le moment où une demande d'accès à une donnée est initiée et le moment où cette donnée est accessible) l'est également.

La latence des accès à la mémoire est directement liée à la proximité entre cette dernière et les ALUs. Cependant, des accès mémoire plus rapides impliquent généralement une capacité de stockage réduite : le cache L1 est généralement spécifique à chaque cœur, tandis que les caches L2 et L3 sont partagés entre les cœurs.

1.5.2. Architecture logicielle

1.5.2.1. Parallélisation en mémoire distribuée

La parallélisation en mémoire distribuée consiste à répartir les ressources de la simulation (maillages, tâches des calculs, données) entre plusieurs processus, chacun ayant accès aux capacités de calcul d'un cœur, d'un ensemble de cœurs ou même d'un nœud complet du système informatique.

Chaque processus ne peut accéder qu'à ses propres données et non directement aux données des autres processus. Il a pour fonction d'exécuter le flux d'instructions qui lui est assigné. Lorsqu'un processus nécessite des données ou des résultats produits par un autre processus, ces informations sont en général échangées via un système de communication par transfert de messages.

Depuis 1993, le Message Passing Interface (MPI) [Clarke et al. 1994] est devenu la norme prédominante pour la gestion des échanges d'informations. Sur un supercalculateur, les ressources de la simulation sont distribuées entre les processus MPI. Chaque processus MPI est caractérisé par son rang MPI. Ces processus peuvent communiquer entre eux via des communicateurs. Le standard MPI offre des méthodes de communication point à point entre deux processus MPI mais aussi des communications collectives lorsqu'un groupe de processus MPI doit diffuser ou récupérer des données ou des informations disponibles sur tous les processus. Cependant, lorsque le nombre de processus MPI est très élevé, dépassant environ 10 000 processus MPI, les communications collectives deviennent coûteuses.

1.5.2.2. Parallélisation intra-nœud

Chaque processus MPI exploite une couche additionnelle de parallélisme, appelée parallélisme par thread en mémoire partagée [McCool et al. 2012], pour paralléliser les calculs entre les cœurs. Cela consiste à attacher un ou plusieurs threads, des unités d'exécution indépendantes qui permettent au processus MPI d'exécuter plusieurs tâches en même temps, à chaque cœur. Chaque thread, qui dispose de sa propre mémoire ainsi que d'une pile d'instructions, a accès aux données de la mémoire vive et exécute une séquence d'instructions.

1.5.2.3. Vectorisation

La vectorisation consiste, sur chaque cœur, à transformer des opérations de calcul scalaires (additions, multiplications scalaires terme à terme sur les coefficients dans les structures de données telles que des tableaux ou des matrices par exemple) en opérations de calcul vectorielles pour optimiser les performances de calcul.

1.5.3. Méthodes de partitionnement spatial de domaine et d'équilibrage de charge

En mécanique des structures, la méthode de partitionnement spatial de domaine [Karypis 2003] est la technique la plus simple et la plus couramment utilisée pour distribuer les calculs entre les nœuds d'un supercalculateur. Elle consiste à subdiviser un domaine (un maillage éléments finis par exemple) en plusieurs sous-domaines tout en prenant en compte les conditions aux interfaces entre ces sous-domaines, comme illustré par la figure 1.10. Ainsi, lors de la subdivision, des couches fantômes sont créées sur les bords des différents sous-domaines dans le but de récupérer les informations des bords des sous-domaines voisins. Chaque subdivision est traitée sur un nœud de calcul du supercalculateur, auquel est assigné un processus MPI.

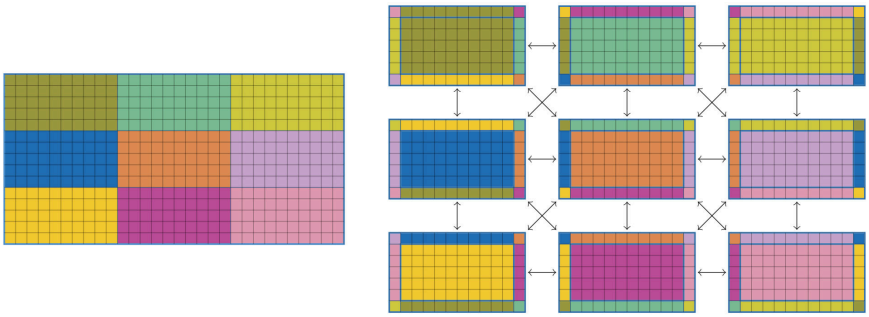


Figure 1.10. – Application de la méthode de décomposition spatiale de domaine à un domaine rectangulaire 2D (source : [Prat 2019])

Une distribution spatiale homogène des calculs entre les ressources d'un supercalculateur ne permet pas de toujours assurer une répartition équitable de la charge entre les sous-domaines. En effet, dans le cas de simulations réelles, la charge de calcul peut varier considérablement d'un sous-domaine à un autre. Par conséquent, certains nœuds de calcul se retrouvent avec une charge de travail nettement moins importante, ce qui les amène à attendre la fin des tâches de calcul sur les autres nœuds. Pour remédier à cette disparité, il est possible d'utiliser des méthodes plus avancées de partitionnement spatial de domaine.

Ces méthodes d'équilibrage de charge fonctionnent via l'attribution d'un poids à chaque élément à partitionner. Le principe de l'algorithme de partitionnement consiste alors à définir des sous-domaines dont la somme des poids est à peu près uniforme. Elles permettent ainsi de quantifier la charge de chaque élément et de minimiser les disparités de charge entre les sous-domaines.

Deux grandes familles de partitionnement existent : les partitionnements géométriques et les partitionnements de graphes.

1.5.3.1. Partitionnement géométrique

Les méthodes de partitionnement géométrique reposent sur la découpe géométrique d'un domaine. De nombreux partitionnements géométriques existent dans la littérature. Les plus couramment utilisés sont le partitionnement Recursive Coordinate Bisection (RCB), le partitionnement Recursive Inertial Bisection (RIB) et le partitionnement Space Filling Curve (SFC), et plus particulièrement le partitionnement Hilbert SFC. Ces méthodes de partitionnement géométriques sont implémentées dans des bibliothèques ou plateformes spécifiques telles que Zoltan [Rajamanickam et al. 2011 ; Boman et al. 2012] ou Coupe [Chevalier et al. 2023].

Coordinate Bisection (RCB)

Le partitionnement RCB [Williams 1991] est dérivé de la méthode Partition Binaire de l'Espace [Berger et al. 1987]. Dans cette approche, le domaine de simulation est partagé en deux sous-domaines par un hyperplan (une surface en 3D) orthogonal à l'un des axes de coordonnées, de manière à répartir équitablement la charge de travail (avec possiblement des poids affectés aux éléments) entre les deux sous-domaines. Ces sous-domaines sont ensuite subdivisés de manière récursive jusqu'à ce que le nombre de sous-domaines corresponde au nombre de processus MPI.

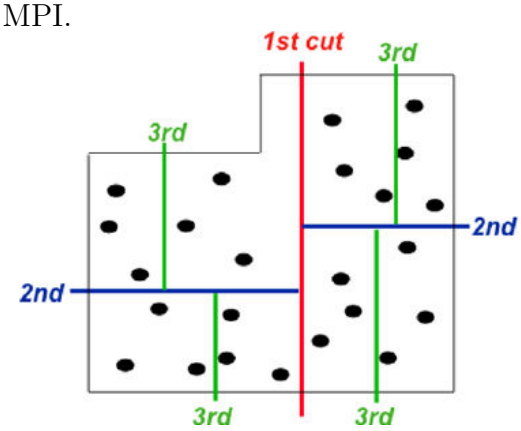


Figure 1.11. – Exemples de partitionnement RCB en dimension 2 et avec des poids de 1 sur les éléments (source : [Boman et al. 2012])

Recursive Inertial Bisection (RIB)

Variante du partitionnement RCB, le partitionnement RIB [Simon 1991 ; Keyser et al. 1992 ; V. E. Taylor et al. 1994] se distingue par le choix des hyperplans utilisés pour subdiviser de manière récursive les sous-domaines. En effet, le plan de coupe est ici défini via un axe appelé axe principal d'inertie du maillage.

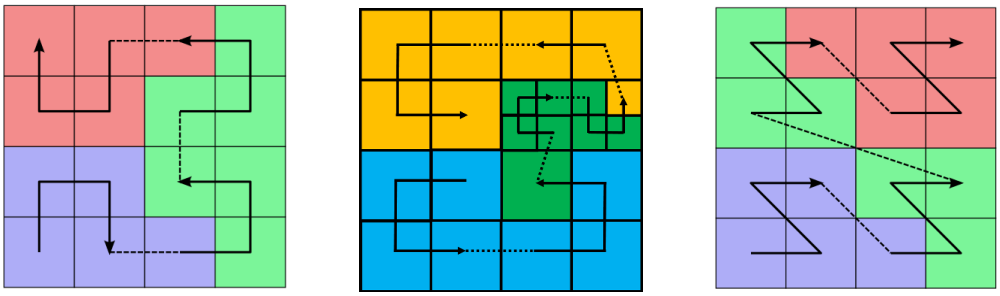
Tout comme dans le cas du partitionnement RCB, cette méthode de partitionnement est adaptée pour un équilibrage dynamique.

Hilbert Space Filling Curve (SFC)

Contrairement aux méthodes de partitionnement précédemment décrites, l'approche de partitionnement SFC [Pilkington et al. 1994 ; Aluru et al. 1997 ; Bader 2012] ne divise pas explicitement le domaine avec un hyperplan, mais ordonne de manière séquentielle les éléments de l'espace de dimension quelconque. Le principe de base de cette technique de partitionnement consiste à attribuer un poids à chaque élément et à construire une courbe ne passant jamais deux fois par le même élément. Un sous-domaine est déterminé lorsque la somme cumulée des poids le long de la courbe excède la charge totale divisée par le nombre souhaité de sous-domaines.

Parmi les courbes les plus fréquemment utilisées figure la courbe de Hilbert [Butz 1968] (voir figure 1.12a). Cette courbe appartient à la famille des fractales, également nommée courbe de Peano [Peano 1890]. Cette famille de courbes est connue pour remplir l'espace considéré. Ainsi, elle possède la spécificité de parcourir n'importe quel maillage, avec ou sans AMR (voir figure 1.12b), de n'importe quelle dimension, sans jamais passer deux fois

par le même élément de la grille et en assurant généralement que deux éléments successifs soient voisins. Dans le cas particulier de la courbe de Hilbert, chaque élément du domaine est numéroté selon son indice de Hilbert et se voit attribuer un poids correspondant à sa charge (défini en cours de calcul, par défaut fixé à 1). Ces éléments sont ensuite répartis sur les sous-domaines en utilisant le critère de découpage de la courbe défini précédemment (changement de sous-domaine dès que la somme des poids des éléments visités par la courbe de Hilbert dépasse la charge totale divisée par le nombre souhaité de sous-domaines). De plus, toute courbe de Peano est invariante de motif dans le sens où elle est construite de manière récursive à partir d'un motif de base, voir la figure 1.12. En effet, lorsqu'il est nécessaire d'appliquer une étape de récursivité (comme c'est le cas pour un maillage raffiné localement par exemple), le motif de base est copié à échelle réduite et substitue certains segments de courbes.



(a) (b) (c)

FIGURE 1.12. – Exemple de deux courbes de Hilbert en (a) et (b) sur une grille en dimension 2 avec et sans AMR, et d'une courbe en Z sans AMR en (c) (source de (a) et (c) : [ČERVENÝ et al. 2019])

Il existe de nombreuses variantes utilisant d'autres courbes de Peano, comme la courbe de Lebesgue également appelée courbe en Z, illustrée par la figure 1.12c. Ici, un indexage dit de Morton [MORTON 1966] permet de parcourir les éléments d'une structure de données arborescente. D'autres courbes remplisseuses d'espaces sont possibles telles que celles de Sierpiński. Toutefois, la courbe de Hilbert demeure la plus répandue car elle ne présente pas de discontinuité (les éléments successifs sont adjacents).

Par ailleurs, l'approche de partitionnement SFC est connue pour s'adapter avec efficacité à des maillages raffinés localement ou encore non-structurés. Ce type d'approche se révèle également pertinent lorsque un re-partitionnement dynamique de la grille ou du maillage est souhaité.

1.5.3.2. Partitionnement de graphe et d'hypergraphe

Contrairement aux méthodes géométriques, les méthodes de partitionnement de graphe et d'hypergraphe [RAJAMANICKAM et al. 2011] considèrent les relations entre les éléments d'un maillage. Ces méthodes partitionnent le maillage en se servant d'un graphe, un ensemble de sommets reliés par des arêtes (orientées ou non), ou d'un hypergraphe, graphes permettant à une arête de relier plus de deux sommets. En mécanique des structures, les éléments finis du maillage sont les sommets du graphe et les frontières partagées entre les éléments finis (arêtes en 2D, surfaces en 3D) représentent les arêtes du graphe reliant ces sommets. Ainsi, les méthodes de partitionnement de graphe visent à regrouper les sommets en sous-domaines sans considérer la forme géométrique de ces sous-domaines.

La répartition de la charge prend en considération le poids de chaque sommet du graphe tout en cherchant à minimiser le nombre d'arêtes coupées, pondérées par leurs valeurs, traduisant la longueur de l'interface entre les sous-domaines.

Ces méthodes ont pour avantage de mieux répartir la charge en général, de considérer la connectivité entre les éléments du graphe et permettent potentiellement de réduire les communications entre les sous-domaines. Néanmoins, les approches de graphe et d'hypergraphe sont respectivement dix et cent fois plus coûteuses environ que les approches géométriques et donc mal adaptées pour des simulations pour lesquelles il y a un besoin fréquent de re-partitionnement (ré-équilibrage). En effet, le temps de partitionnement avec ces approches est en général au moins équivalent au temps de résolution du système linéaire ; ce qui affecte grandement donc le temps total de simulation.

Plusieurs bibliothèques logicielles permettent de partitionner un maillage par une stratégie de graphe ou d'hypergraphe telles que Metis [Karypis et al. 1998] (pour du partitionnement de graphes), sa version parallélisée en mémoire distribuée (MPI) et en mémoire partagée ParMetis [Karypis et al. 1997] et son extension aux hypergraphes avec hMetis [Karypis et al. 1999 ; Karypis et al. 2000]. Pour du partitionnement de graphes, il peut également être cité Scotch [Pellegrini et al. 1996], et son extension parallèle PT-Scotch (performances similaires à ParMetis), ou encore Zoltan [Rajamanickam et al. 2011 ; Boman et al. 2012] et Chaco [Hendrickson 2011].

1.6. Solveurs linéaires et pré-conditionneurs

Lorsque l'on cherche à traiter des problèmes éléments finis en mécanique quasi-statique des structures via une résolution implicite, de grands systèmes linéaires creux de la forme

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{1.5}$$

est le plus souvent nécessaire, avec $\mathbf{A}$ matrice carrée inversible de taille $n \times n$, $\mathbf{x}$ le vecteur solution recherché et $\mathbf{b}$ le vecteur second membre. Ces systèmes sont dits creux car, de manière générale, la matrice à inverser comporte une forte proportion de coefficients nuls et, par conséquent, le système à résoudre est constitué d'équations dont chacune fait intervenir un très faible nombre d'inconnues.

Des algorithmes et structures de données spécifiques à la résolution de systèmes creux sont donc généralement utilisés pour réduire les coûts en calcul et en mémoire. Ainsi, pour résoudre de tels systèmes, il existe deux grandes familles de méthodes :

- les méthodes directes pour lesquelles la précision de la solution est de l'ordre de la précision machine ;
- les méthodes itératives basées sur un processus itératif de recherche de solutions approchées jusqu'à une précision donnée.

1.6.1. Solveurs linéaires directs

De manière générale, les solveurs directs présentent les avantages d'être génériques, précis et robustes. De plus, ils exploitent très efficacement les performances de processeurs modernes et sont très parallélisables car les ressources en temps et en mémoire peuvent être prédites aisément [George et al. 1981 ; Duff et al. 1986]. Ils sont donc fréquemment utilisés dans les codes industriels dans un but d'obtenir des solutions fiables. De nombreuses

méthodes directes, parallélisées ou non, sont disponibles pour résoudre des système linéaires. Elles sont souvent basées sur le processus d'élimination de Gauss par des stratégies de factorisation de $\mathbf{A}$ (par exemple, en factorisant $\mathbf{A}$ en matrices $\mathbf{LU}$ facilement inversibles, avec $\mathbf{L}$ et $\mathbf{U}$ des matrices triangulaires inférieures et supérieures respectivement et appelés facteurs), ou via la factorisation de Cholesky ou la factorisation $\mathbf{QR}$. Le lecteur peut se référer à [George et al. 1981 ; Dongarra et al. 1998] pour plus de détails sur ces méthodes.

Cependant, la résolution de problèmes de grandes tailles est gourmande avec ce type de solveur, notamment à cause des coûts en nombre d'opérations pour factoriser la matrice à inverser et des coûts en empreinte mémoire qui augmentent fortement lorsque le système est gros. Ainsi, nécessitant moins d'opérations et de mémoire que les solveurs directs, les solveurs linéaires itératifs sont généralement préférés lorsqu'il s'agit de résoudre des systèmes découlant de problèmes de grandes tailles issus de la discrétisation d'équations aux dérivées partielles. Ainsi, dans ce manuscrit, les résultats obtenus découleront toujours de solutions acquises par résolution avec un solveur linéaire itératif.

1.6.2. Solveurs linéaires itératifs

Les solveurs linéaires itératifs consistent à résoudre les systèmes linéaires comme (1.5) en calculant de manière successive les valeurs d'une suite de vecteurs $\mathbf{x}^k$ convergeant vers $\mathbf{x}$ lorsque $k \to \infty$, à partir d'un vecteur solution initial $\mathbf{x}^0$.

De manière générale, la matrice $\mathbf{A}$ est décomposée sous la forme $\mathbf{A} = \mathbf{M} - \mathbf{N}$ avec $\mathbf{M}$ une matrice facile à inverser (diagonale, triangulaire ou diagonale par bloc par exemple) dans le but de pouvoir construire aisément $\mathbf{x}^{k+1}$ en fonction de $\mathbf{x}^k$. En effet, le problème (1.5) peut s'écrire

$$\mathbf{M}\mathbf{x} = \mathbf{N}\mathbf{x} + \mathbf{b}; \tag{1.6}$$

ce qui permet d'obtenir $\mathbf{x}^{k+1}$ via

$$\mathbf{x}^{k+1} = \mathbf{M}^{-1}\mathbf{N}\mathbf{x}^k + \mathbf{M}^{-1}\mathbf{b}. \tag{1.7}$$

Les méthodes itératives recensent un large panel de techniques. Les plus connues et les plus largement utilisées sont les méthodes Krylov (en particulier la méthode du gradient conjugué) mais il existe aussi les méthodes multi-grille ou encore celles de Jacobi, de Gauss-Seidel et de relaxation. Pour plus de détails sur ces méthodes, le lecteur peut se référer à [Takahashi et al. 2014].

De manière générale, les méthodes itératives ne sont, par contre, pas aussi robustes que les méthodes directes. En effet, pour un certain nombre de problèmes, elles ne convergent pas ou difficilement ou en un temps prohibitif. Cependant, une technique très répandue pour améliorer la convergence des méthodes itératives est de pré-conditionner le système matriciel initial. Cela revient à transformer ce système en un système équivalent dont les propriétés sont plus favorables à la convergence. Grâce au pré-conditionnement, présenté en section 1.6.3, les méthodes itératives gagnent en robustesse. Par contre, trouver un compromis entre robustesse, généricité et efficacité reste un enjeu.

Dans ces travaux de thèse, la méthode du gradient conjugué est utilisée car elle est particulièrement adaptée au systèmes creux de grandes tailles à résoudre et est parallélisable efficacement. Dans un cadre parallèle, les produits matrice-vecteur qui sont effectués à chaque itération globale du solveur à gradient conjugué peuvent être réalisés au niveau de

chaque processus, tandis que les calculs de produits scalaires nécessaires pour former les nouveaux vecteurs utilisés lors d'une itération globale du gradient conjugué sont effectués globalement via des communications entre les processus.

1.6.3. Pré-conditionnement

Pour un problème donné, la convergence du solveur dépend fortement du conditionnement de la matrice $\mathbf{A}$ et peut s'avérer lente. Ainsi, lorsque la matrice $\mathbf{A}$ est mal conditionnée (par exemple pour des problèmes avec pénalisation), l'idée est de remplacer le système (1.5) par un système pré-conditionné ayant la même solution mais avec une matrice à inverser mieux conditionnée. En définissant les matrices carrées de pré-conditionnement $\mathbf{M}_G$ (à gauche) et $\mathbf{M}_D$ (à droite) inversibles de taille $n \times n$, le système pré-conditionné s'écrit généralement

$$\mathbf{M}_G^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}_G^{-1}\mathbf{b} \tag{1.8}$$

pour un pré-conditionnement effectué à gauche,

$$\begin{cases} \mathbf{A}\mathbf{M}_D^{-1}\mathbf{y} = \mathbf{b} \\ \mathbf{x} = \mathbf{M}_D^{-1}\mathbf{y} \end{cases} \tag{1.9}$$

pour un pré-conditionnement effectué à droite ou

$$\begin{cases} \mathbf{M}_G^{-1}\mathbf{A}\mathbf{M}_D^{-1}\mathbf{y} = \mathbf{M}_G^{-1}\mathbf{b} \\ \mathbf{x} = \mathbf{M}_D^{-1}\mathbf{y} \end{cases} \tag{1.10}$$

pour un pré-conditionnement effectué des deux côtés.

Ainsi, l'enjeu réside dans le choix des matrices de pré-conditionnement $\mathbf{M}_G$ et $\mathbf{M}_D$ qui doivent permettre d'améliorer le conditionnement de la matrice à inverser et par conséquent la convergence du solveur. Il s'agit donc de construire une matrice $\mathbf{M}$ ($\mathbf{M}_G$, $\mathbf{M}_D$ ou $\mathbf{M}_G\mathbf{M}_D$) proche de la matrice $\mathbf{A}$ telle que $\mathbf{M}^{-1}\mathbf{A}$ soit mieux conditionnée que $\mathbf{A}$ et donc plus facile à inverser. Il n'existe cependant pas de solution générique pour former un pré-conditionneur optimal. Par exemple, si $\mathbf{M}_G = \mathbf{A}$ alors la convergence est obtenue en une seule itération mais le calcul de l'inverse du pré-conditionneur devient aussi complexe que si le système n'était pas pré-conditionné. Pour des problèmes de contact frottant, les auteurs de [Alart et al. 1995] ont montré que l'utilisation du pré-conditionneur par factorisation LU incomplète (ILU) mais aussi celui de type multi-grille (grossier/fin) permettent de diviser le nombre d'itérations du solveur par 4 et de réduire les temps de résolution par près de 3.

Étant donné que les pré-conditionneurs influent fortement sur la convergence des solveurs linéaires itératifs, de nombreuses stratégies de pré-conditionnement ont été introduites dans la littérature et appliquées pour la résolution d'une multitude de systèmes linéaires. Il existe donc différentes classes de pré-conditionneurs dits algébriques telles que les méthodes itératives stationnaires (SSOR, symmetric SOR en anglais, par exemple), celles basées sur des factorisation incomplètes, les méthodes algébriques multi-grilles et celles basées sur le calcul d'inverses approchés creux de $\mathbf{A}$. Il peut être cité également le pré-conditionnement EBE, adapté efficacement à la vectorisation des calculs (voir la section 1.5.2.3 et les travaux présentés dans [Hughes et al. 1987]) et dont l'algorithme se révèle parallélisable dans le cas d'un contact frottant (voir [Alart et al. 1997]). Dans les sections 1.6.3.1 et 1.6.3.2

sont introduites les deux méthodes de pré-conditionnement utilisées dans ces travaux de thèse, en l'occurrence BoomerAMG et la factorisation LU incomplète.

1.6.3.1. Pré-conditionnement avec BoomerAMG

BoomerAMG [Henson et al. 2002] est une implémentation parallèle et scalable efficace de la méthode algébrique multi-grille pour la résolution d'équations aux dérivées partielles elliptiques, sur des maillages structurés ou non. BoomerAMG peut être employé aussi bien comme solveur que comme pré-conditionneur.

La méthode multi-grille est basée sur trois principales étapes permettant la résolution du système linéaire (si utilisé comme solveur) ou la création du pré-conditionnement : une étape de grossissement, une d'interpolation et une autre de relaxation. De plus, BoomerAMG possède un nombre important d'options et de paramètres et il est donc possible de choisir entre différentes techniques pour le grossissement, l'interpolation et la relaxation.

Par ailleurs, d'après les développeurs de la plateforme MOOSE [INL 2023], l'option la plus importante pour ce pré-conditionneur est le seuil fort (Strong Threshold en anglais). Cette option donne la capacité de contrôler le mécanisme de grossissement en éliminant les composantes de la matrice qui "n'ont pas d'importance". De manière générale, l'élimination d'un plus grand nombre de composantes de la matrice est bénéfique pour la vitesse de passage dans BoomerAMG et par conséquent dans une itération du solveur mais peut impacter très fortement la qualité du pré-conditionneur. En effet, l'élimination d'un trop grand nombre de composantes de la matrice conduit à une perte de performance globale de BoomerAMG qui nécessitera en conséquence un plus grand nombre d'itérations du solveur linéaire.

1.6.3.2. Pré-conditionnement par factorisation LU incomplète (ILU)

La méthode de factorisation LU incomplète peut être utilisée comme solveur itératif autonome, comme pré-conditionneur pour les méthodes itératives de Krylov ou comme lisseur pour les méthodes multi-grilles telles que BoomerAMG.

Le principe de la factorisation incomplète réside dans la limitation du remplissage des facteurs en éliminant certaines composantes lors de la factorisation [Gaidamour 2009]. Ainsi, le niveau de remplissage des facteurs mais aussi les critères permettant de déterminer comment ils doivent être remplis influencent la robustesse du pré-conditionneur. Le plus souvent, les algorithmes de factorisation incomplètes se basent sur deux types de critères indépendants : un critère analysant la position des coefficients dans la matrice $\mathbf{A}$ et un autre s'appuyant sur la valeur numérique de ces derniers. Les méthodes de factorisations incomplètes les plus courantes sont les factorisations ILUK (qui maîtrise le remplissage via un critère combinatoire) et ILUT (qui agit sur le remplissage en regardant les valeurs numériques des termes de la matrice) [Saad 2003].

Dans un cadre parallèle, les inconnues du vecteur solution $\mathbf{x}$ sont divisées en des points intérieurs et d'autres d'interface, où les points d'interface séparent les points intérieurs dans des domaines voisins. Les factorisations ILU locales reviennent à diviser les lignes de la matrice en blocs locaux (appartenant au processus) et externes (n'appartenant pas au processus). Selon les données d'entrée choisies, plusieurs combinaisons sont possibles pour la résolution du problème. Ainsi, le problème est résolu localement (dans les blocs locaux) avec la méthode ILU (ILUK ou ILUT), et globalement par des solveurs itératifs (GMRES

ou méthode de Jacobi par blocs notamment) via l'utilisation du système de complément de Schur.

Dans ces travaux de thèse, la méthode ILUK est choisie pour la résolution locale du problème et la méthode de Jacobi par blocs l'est pour la résolution globale.

1.7. État de l'art sur les développements contact-AMR-HPC actuels

Dans cette section, les travaux de recherche sont recentrés par rapport aux développements récents relatés dans la littérature. Ainsi, ceux s'approchant le plus, à notre connaissance, de la combinaison contact entre solides déformables, raffinement adaptatif de maillage et calcul haute performance sont donc exposés. Il s'articulent autour de différents axes. D'une part, des approches de raffinement adaptatif de maillage appliquées au problème de contact ont été proposées. Des stratégies de parallélisation de méthodes de raffinement adaptatif de maillage ont été développées en mécanique des structures sans contact. Enfin, des travaux ont été proposés autour de la parallélisation de problèmes considérant le contact entre solides déformables maillés uniformément ou le contact de type rigide-déformable avec un maillage adaptatif.

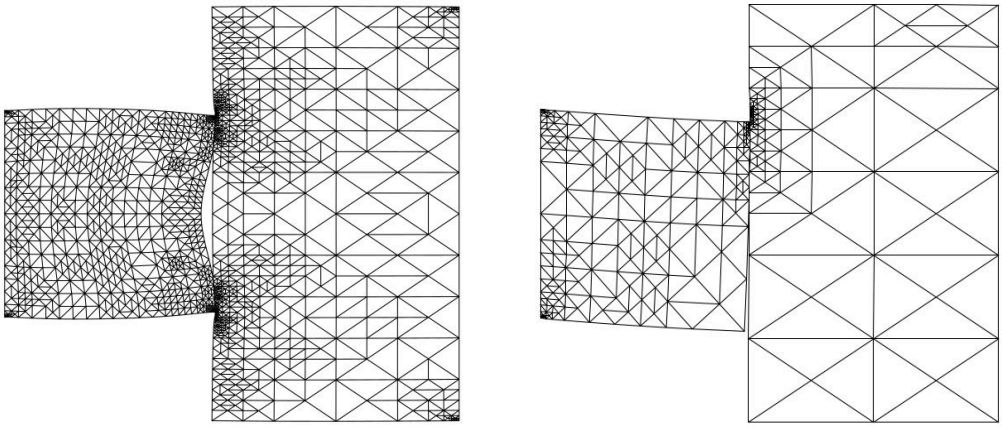
1.7.1. Techniques de raffinement adaptatif de maillage appliquées au problème de contact

De nombreux travaux se sont concentrés sur l'application des techniques d'AMR dans le contexte des problèmes en mécanique du contact, principalement dans le cadre de la méthode des éléments finis.

L'étude présentée dans [Wriggers et al. 1998] propose de modifier l'estimateur d'erreur de Zienkiewicz et Zhu (ZZ) [O. C. Zienkiewicz et al. 1987] afin de prendre en compte les contributions de contact pour l'application d'une stratégie h-adaptative conforme. Par ailleurs, une méthode multi-grille de Correction Locale de Résidu (LDC) est introduite dans [Liu et al. 2017] pour des problèmes de contact frottant. Ici, les auteurs ont appliqué avec succès l'estimateur ZZ standard dans chaque corps séparément, comme pour les problèmes multi-matériaux. De plus, l'adaptation de méthodes multi-grille locale pour les problèmes de contact a aussi été étudiée dans [Boffy et al. 2012]. Il est également possible de citer [Braess et al. 2007 ; Braess et al. 2009] où un estimateur d'erreur *a posteriori* spécifique pour la résolution de problèmes de contact de type rigide-déformable est introduit et analysé sur des exemples 2D.

Dans [Gustafsson et al. 2020], une étude de convergence de la méthode de Nitsche pour la résolution de problèmes de contact 2D sans frottement entre deux corps élastiques y est réalisée pour des maillages triangulaires raffinés avec une stratégie h-adaptative conforme basée sur un estimateur d'erreur *a posteriori* spécifique. Cette stratégie, présentée dans le cadre d'un contact de type nœud-à-nœud et illustrée dans la figure 1.13, a été étendue dans [Gustafsson et al. 2021] à un contact de type nœud-à-surface. Pour leur part, les auteurs de [Araya et al. 2023] ont proposé une stratégie d'estimation de l'erreur *a posteriori* pour des problèmes de contact de type rigide-déformable 2D avec maillages triangulaires via, là également, une résolution du problème de contact par la méthode de Nitsche. Similairement à [Araya et al. 2023], les travaux présentés dans [Hild et al. 2009] proposent deux estimateurs d'erreur *a posteriori* basés sur le résidu adaptés à la résolution

de problèmes de contact frottant de type rigide-déformable. Ces estimateurs ont eux aussi été appliqués sur des cas d'études 2D à maillages triangulaires et raffinés via une méthode h-adaptative conforme. La différence avec les travaux qui viennent d'être décrits est que la méthode utilisée ici pour résoudre le problème de contact est celle des multiplicateurs de Lagrange.



(a) (b)

FIGURE 1.13. – Exemple de maillages triangulaires 2D localement raffinés pour des problèmes de contact (source : [GUSTAFSSON et al. 2020])

Les travaux de [RODA-CASANOVA et al. 2018 ; RADEMACHER 2019] se concentrent sur de l'adaptation de maillage avec des méthodes de h-raffinement non-conformes. Dans [RODA-CASANOVA et al. 2018], une approche semi-analytique pour résoudre des problèmes de contact élastique sans frottement à l'aide d'une stratégie de raffinement adaptatif du maillage a été proposée. Ici, le raffinement de maillage h-adaptatif non-conforme est couplé à un indicateur de contact basé sur la pression de contact. Cette approche a été appliquée sur des problèmes de contact rigide-déformable 3D. Dans [RADEMACHER 2019], un estimateur d'erreur *a posteriori* spécifique à la résolution de problèmes de contact 2D avec frottement a été proposé.

Il faut aussi noter que, dans la majorité des travaux exposés ici, les méthodes proposées ont été appliquées uniquement sur des géométries 2D. D'autre part, certains travaux ne concernent que des problèmes de contact rigide-déformable (problèmes d'obstacles), [BRAESS et al. 2007 ; RODA-CASANOVA et al. 2018 ; RADEMACHER 2019 ; ARAYA et al. 2023] pour ne citer qu'eux.

Pour sa part, l'extension de la SBFEM aux problèmes de contact [HIRSHIKESH et al. 2021] a été directement introduite dans un cadre de maillages h-adaptatifs non-conformes 2D basé sur une décomposition en quadtree avec un indicateur d'erreur SBFEM dédié [SONG et al. 2018].

En ce qui concerne la VEM, [ALDAKHEEL et al. 2020] fournit une méthodologie pour combiner l'AMR et les problèmes de contact 2D. Deux types de techniques AMR sont étudiés sur les problèmes de contact en 2D : les maillages h-adaptatifs non-conformes ou les maillages de cellules de Voronoï localement raffinés conformes. Le maillage est localement raffiné dans une zone de contact prédéfinie jusqu'à ce qu'un critère géométrique sur la zone de contact raffinée soit atteint. L'extension aux problèmes de contact 3D du VEM avec AMR basé sur des éléments de type Voronoï est proposée dans [CIHAN et al. 2022].

De manière générale, les travaux proposés basés sur l'application d'une stratégie de raffinement adaptatif de maillage pour la résolution du contact se sont principalement restreints à des problèmes 2D avec contact de type rigide-déformable. Ainsi, les travaux présentés dans ce manuscrit visent à dépasser ces limitations pour résoudre des problèmes 3D de contact déformable-déformable.

1.7.2. Stratégies de parallélisation adaptées au raffinement adaptatif de maillage

Pour les problèmes très singuliers tels que ceux issus de la mécanique du contact, la discrétisation peut devenir très fine localement et le nombre d'inconnues prohibitif, même avec l'AMR. Il est donc également souhaitable d'utiliser des méthodes de calcul haute performance (HPC) pour un traitement plus rapide des calculs.

La résolution rapide d'équations aux dérivées partielles avec la FEM sur des maillages adaptativement raffinés via des stratégies de parallélisation a déjà été largement explorée. Par exemple, il a récemment été présenté dans [Munch et al. 2022] un algorithme pour résoudre les contraintes des nœuds non-conformes dans le contexte de la méthode des éléments finis matrix-free sur des unités centrales de traitement (CPUs) et des unités de traitement graphique (GPUs). L'article [Africa et al. 2022] détaille un processus parallèle d'adaptation de maillage non-conforme piloté par un estimateur *a posteriori* basé sur la reconstruction des flux pour des problèmes scalaires de réaction-diffusion-advection. En outre, les travaux exposés dans [Červený et al. 2019 ; Anderson et al. 2020] traitent d'une procédure parallèle générique pour le raffinement local de maillages 2D et 3D via une stratégie h-adaptative non-conforme. Cette approche s'articule notamment à partir d'opérateurs de prolongation et de restriction pour la bonne gestion des non-conformités de maillage, d'une stratégie AMR estimate-mark-refine et d'un partitionnement du maillage via la méthode Hilbert Space Filling Curve. La scalabilité de l'algorithme a été étudiée pour des simulations avec AMR allant jusqu'à 393000 cœurs CPU.

Le développement d'algorithmes AMR parallèles scalables a également fait l'objet d'études antérieures, voir [Flaherty et al. 1997 ; Tu et al. 2005 ; Mitchell 2007 ; Sundar et al. 2008 ; Burstedde et al. 2011 ; Burstedde 2020] pour n'en citer que quelques unes, en particulier autour de méthodes de rééquilibrage de charges utilisant des partitionnements basés sur des structures de données de type arbre (octree). Ces stratégies reposent généralement sur des méthodes de partitionnement basées sur des arbres de raffinement et donc sur des approches h-adaptatives non-conformes (avec des nœuds non-conformes). En effet, les travaux présentés dans [Flaherty et al. 1997] s'articulent autour de la résolution parallèle des lois de conservation, évoluant dans le temps, sur un maillage tridimensionnel non structuré via l'utilisation de la méthode de Galerkin discontinue. L'équilibrage de charges proposé est comparé à une technique de partitionnement classique de type Recursive Inertial Bisection (RIB, voir la section 1.5.3.1). Les travaux issus de [Tu et al. 2005] proposent une stratégie parallèle de construction et d'adaptation dynamique de maillages. Dans cette article, des résultats allant jusqu'à 2000 cœurs CPU sont présentés pour des modélisations de séismes couvrant des problèmes adaptés au terascale (10^{12} flops de puissance de calcul). Les travaux de [Flaherty et al. 1997] sont ré-utilisés dans [Mitchell 2007] pour proposer une méthode de partitionnement de grilles similaire où la principale nouveauté réside dans la génération de l'arbre dans le but de fournir des partitions connectées entre elles lors de l'usage de maillages composés

d'éléments triangulaires ou tétraédriques. De plus, dans [Sundar et al. 2008] est présentée une stratégie parallèle pour la génération et le grossissement d'octrees dans le cas de simulations avec AMR, allant jusqu'à 1024 cœurs CPU, qui respectent la règle du "2 :1 balance" (un seul nœud non-conforme par arête n'est autorisé). Par ailleurs, les travaux de Burstedde et al. [Burstedde et al. 2011 ; Burstedde 2020] reposent sur la conception d'algorithmes scalables sur des dizaines de milliers de cœurs CPU permettant de distribuer des calculs éléments finis avec AMR dynamique par le biais de forêts d'octrees, composées d'octrees (ou quadtree en 2D) connectés entre-eux. L'apport de ces travaux réside sur le développement des connectivités entre les octrees.

En outre, ces méthodes et algorithmes ont été intégrés dans plusieurs bibliothèques open-sources scalables utilisant la méthode des éléments finis telles que deal.II [Arndt et al. 2021], MFEM [Anderson et al. 2020], MoFEM [Kaczmarczyk et al. 2020], FEniCS [Alnæs et al. 2015], FreeFem++ [Hecht 2012] ou libMesh [Kirk et al. 2006] par exemple.

1.7.3. Méthodes de parallélisation du problème de contact

Des stratégies de résolution de problèmes de contact de grandes tailles, adaptées à l'architecture multiprocesseur des supercalculateurs, sont à l'étude depuis de nombreuses années. En effet, des travaux pionniers [Barboteu et al. 2001 ; Dureisseix et al. 2001 ; Alart 2007] introduisent des méthodes de décomposition de domaine et des algorithmes itératifs associés, adaptés aux ordinateurs parallèles. En particulier, ceux présentés dans [Barboteu et al. 2001 ; Alart 2007] se concentrent sur des problèmes "multi-contacts", tandis que ceux introduits dans [Dureisseix et al. 2001] résout des problèmes de contact sans frottement avec des multiplicateurs de Lagrange. Cependant, les performances de ces méthodes ont été démontrées sur un petit nombre de cœurs de calcul, pas plus d'une vingtaine. La méthode de pénalisation appliquée aux problèmes de contact rigide-déformable [Har et al. 2003] permet d'atteindre de bonnes performances jusqu'à soixante unités de calcul.

Des études récentes [Wiesner et al. 2021 ; Franceschini et al. 2022 ; Guillamet et al. 2022 ; Dostál et al. 2023 ; Mayr et al. 2023] proposent des stratégies permettant des calculs scalables sur un plus grand nombre de processeurs. Les travaux détaillés dans [Wiesner et al. 2021 ; Franceschini et al. 2022 ; Mayr et al. 2023] se concentrent sur des problèmes de point-selle (découlant des multiplicateurs de Lagrange ou des méthodes mortar) avec des pré-conditionneurs dédiés. Ces approches sont scalables jusqu'à des centaines de cœurs. En particulier, dans [Franceschini et al. 2022], la stratégie parallèle de résolution s'appuie notamment sur la construction d'un pré-conditionnement basé sur l'approximation du complément de Schur et les résultats de scalabilité y sont présentés pour des calculs allant jusqu'à 64 cœurs CPU. Dans [Mayr et al. 2023], les auteurs proposent une stratégie scalable jusqu'à 480 processeurs permettant de résoudre un problème en mécanique du contact dynamique avec un appariement de type nœud-à-surface via la méthode de mortar. Cette proposition est basée sur une technique de décomposition de domaine chevauchante de la discrétisation de l'interface, dans le but de garantir l'accès à toutes les paires maître/esclave nécessaires à la recherche de contacts et à l'évaluation des termes de mortar. La communication des données issues des couches fantômes est donc ici un point majeur de ces travaux. Dans [Guillamet et al. 2022], un algorithme parallèle qui s'étend jusqu'à 2400 cœurs pour résoudre un problème de contact rigide-déformable comportant jusqu'à 250 millions d'inconnues est décrit. La stratégie proposée est basée sur la méthode de décomposition de domaine Dirichlet-Neumann pour respecter les conditions

de contact. Dans la même idée, les travaux présentés dans [Dostál et al. 2023] utilisent des techniques de décomposition de domaine de type FETI pour résoudre un problème de contact rigide-déformable en parallèle.

Par ailleurs, lors de la sortie de la version de l'environnement logiciel MFEM de Mai 2024, une mini-application s'appuyant sur la bibliothèque externe Tribol[2] a été introduite pour résoudre un problème de contact. Elle met en contact deux cubes élastiques linéaires alignés et soumis à l'hypothèse des petites perturbations (HPP), et résout en parallèle le problème avec la méthode de mortar, pour garantir le patch test de contact. Cependant, à notre connaissance, ces développement n'ont pas encore fait l'œuvre d'une publication ; ce qui ne nous permet pas de connaître les performances parallèles de cette stratégie.

1.7.4. Parallélisation du problème de contact rigide-déformable avec raffinement adaptatif de maillage

Combiner la parallélisation des méthodes AMR avec la mécanique du contact demeure un défi scientifique. Comme nous l'avons déjà dit précédemment, les développements actuels se concentrent principalement sur des méthodes adaptatives non parallélisées pour les problèmes de contact, ou sur des méthodes de parallélisation pour des problèmes de contact résolus sur des maillages uniformément raffinés.

À notre connaissance, seule l'étude présentée dans [Frohne et al. 2016] analyse la performance parallèle de la résolution d'un problème en mécanique du contact avec un raffinement adaptatif du maillage local. Un problème conventionnel de contact à un corps (contact rigide-déformable) est résolu avec une stratégie d'ensemble actif (méthode des statuts) primal-dual basée sur les multiplicateurs de Lagrange qui peut être considérée comme une méthode de Newton semi-lisse. Ce problème de contact rigide-déformable simplifie grandement le problème général de contact déformable-déformable, notamment en matière de détection du contact et de raffinement du maillage. La stratégie est mise en œuvre dans la bibliothèque éléments finis deal.II. Le raffinement du maillage est effectué par une variante de l'estimateur d'erreur de Kelly [Kelly et al. 1983] prêt à être utilisé dans le cadre de deal.II. Pour davantage de détails sur la manière dont est raffiné localement un maillage éléments finis via la procédure utilisée dans deal.II, le lecteur peut se référer à [deal.II 2023]. Un raffinement hiérarchique h-adaptatif est ensuite appliqué aux éléments finis hexaédriques linéaires et quadratiques. Les résultats de la simulation de contact avec une scalabilité jusqu'à 1 024 processus sont montrés.

1.8. Cadre de travail

Dans cette section, les différents choix effectués pour mener à bien ces travaux de thèse sont présentés et justifiés.

1.8.1. Choix de l'environnement logiciel de travail

Comme décrit dans la section 1.7.2, plusieurs environnements logiciels combinant HPC et AMR dans le cadre de la méthode des éléments finis existent déjà. Nous avons donc décidé d'ajouter le traitement de la mécanique du contact dans un tel environnement.

2. https ://github.com/LLNL/Tribol

L'environnement logiciel MFEM [Anderson et al. 2020] a été choisi pour les principales raisons suivantes : scalabilité prouvée pour un grand nombre de cœurs, prise en charge mature des hexaèdres et des tétraèdres, prise en charge de l'AMR non-conforme dans un cadre parallèle, grande communauté active, ensemble minimal d'outils pour la modélisation de la mécanique du contact et capacité à être intégré dans une pile logicielle. En outre, les opérateurs parallélisés de résolution de systèmes contraints disponibles dans MFEM peuvent être directement utilisés pour construire le système linéaire obtenu suite à l'application des termes dus à l'approximation numérique du problème de contact.

Par ailleurs, dans MFEM, l'algorithme parallélisé pour le raffinement de maillage h-adaptatif non-conforme [Červený et al. 2019], basé sur l'estimateur Zienkiewicz-Zhu *a posteriori* [O. C. Zienkiewicz et al. 1987] et prêt à l'emploi, peut être enrichi par nos propres procédures estimate-mark-stop-refine.

Dans ces travaux, les versions 4.4 et 4.5.2 de MFEM ont constitué la base de nos développements.

1.8.2. Choix de la méthode de raffinement

Les maillages structurés considérés dans ces travaux sont basés sur des éléments finis quadrangulaires ou hexaédriques pour les raisons que nous avons explicitées dans la section 1.4.1.

En ce qui concerne la stratégie AMR, une solution de raffinement de maillage h-adaptatif hiérarchique non-conforme [Zhu et al. 1993 ; Ródenas et al. 2017] est appliquée. Cette approche a été choisie car elle s'est déjà avérée fortement scalable dans d'autres contextes [Červený et al. 2019 ; Anderson et al. 2020] au sein de l'environnement logiciel MFEM mais aussi pour les avantages du raffinement h-adaptatif non-conforme, déjà décrits dans la section 1.4.2.1.

Le processus de raffinement adaptatif du maillage est réalisé à l'aide d'un algorithme itératif tel que décrit dans la figure 1.14. En effet, le problème est d'abord résolu sur un maillage grossier avant d'appliquer le processus de raffinement selon le critère de détection choisi. Cette approche, appelée stratégie estimate-mark-refine [Verfürth 1996 ; Nochetto et al. 2012 ; Koliesnikova et al. 2021], est répétée de manière itérative jusqu'à ce que le critère d'arrêt défini soit satisfait.

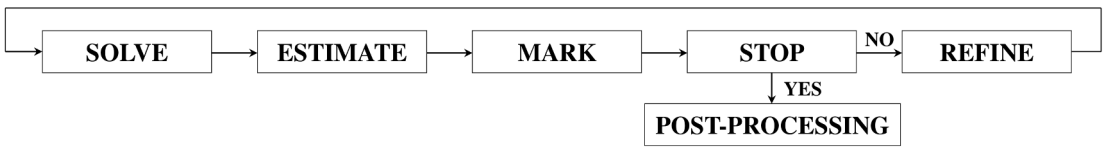


Figure 1.14. – Algorithme de raffinement adaptatif de maillage générique (source : [Koliesnikova et al. 2021])

En outre, le raffinement est choisi isotrope avec un ratio de raffinement de 2 qui respecte la règle du "un seul nœud non-conforme par arête", aussi appelée "single hanging node rule", car elle s'est montrée très efficace dans des travaux précédents (voir [Koliesnikova et al. 2021]).

1.8.3. Choix de l'estimateur d'erreur

Dans ces travaux, l'estimateur d'erreur *a posteriori* introduit par Zienkiewicz et Zhu dans [O. C. Zienkiewicz et al. 1987 ; O. C. Zienkiewicz et al. 1992], appelé estimateur

ZZ, a été choisi pour évaluer l'erreur de discrétisation et donc régir l'adaptation du maillage. Cet estimateur d'erreur a été retenu car il est déjà implémenté dans la bibliothèque MFEM mais aussi parce qu'il est largement répandu en mécanique des solides car il est générique (il ne se limite pas à un problème spécifique), peu coûteux à calculer, facile à mettre en œuvre et fonctionne bien en pratique [Farrell et al. 2011].

Dans ces travaux, cet estimateur est le seul testé car notre objectif principal est l'analyse de la stratégie AMR générique parallélisée pour la mécanique du contact et non une étude approfondie des estimateurs d'erreur.

1.8.4. Choix des stratégies de détection et de résolution du contact

Le problème de contact sera traité par une résolution par pénalisation combinée à la méthode des statuts. Celle-ci permet de ne traiter que les variables primales et de s'appuyer sur les avantages qui découlent de cette méthode (voir 1.3.3.1). En outre, les opérateurs parallélisés pour la résolution de systèmes contraints disponibles dans MFEM peuvent être directement utilisés. De plus, il faut souligner que l'obtention de solveurs scalables et de pré-conditionneurs associés pour les problèmes de point-selle est encore un travail en cours [Wiesner et al. 2021 ; Mayr et al. 2023 ; Nataf et al. 2023] ; ce qui nous a conforté dans notre choix de résoudre notre problème de contact par la méthode de la pénalisation.

Un appariement nœud-à-nœud des éléments en contact permet de bien définir le système matriciel [Luenberger et al. 1984] et est considéré dans une première approche (voir les chapitres 2 et 3). Ici, la détection du contact n'étant pas l'objet de ces travaux, elle n'est pas réalisée via l'utilisation d'un algorithme généraliste. En effet, dans ces travaux, la normale du contact est fixée *a priori*. La méthode d'appariement est ensuite étendue lors du chapitre 4 à l'utilisation d'une stratégie nœud-à-surface.

1.8.5. Caractéristiques du supercalculateur Topaze du CCRT/CEA

Dans le but de simuler des problèmes de grandes tailles sur un nombre important de processeurs, il est nécessaire de nous appuyer sur des architectures de calcul répondant à nos besoins. C'est pourquoi, dans ce manuscrit, l'essentiel des calculs présentés ont été effectués sur les nœuds de calcul du supercalculateur Topaze du CCRT/CEA (Centre de Calcul Recherche et Technologie). Ce supercalculateur de 8 PFlops de puissance crête, assemblé par ATOS/BULL, est composé de plusieurs partitions de nœuds de calcul. La partition que nous avons utilisée a une puissance de 4,5 PFlops et est composée de 864 nœuds, avec 128 cœurs CPU et 256 Go de mémoire chacun. Les cœurs en question sont des processeurs AMD Milan de fréquence d'horloge 2,45 GHz. Enfin, les nœuds sont interconnectés par un réseau d'interconnexion haute performance ayant un débit de 12,5 GB/s. Le lecteur peut se référer à [CEA 2021a ; CEA 2021b] pour plus de détails.

1.9. Contributions de la thèse

Dans cette section, le positionnement des travaux présentés dans ce manuscrit vis-à-vis de l'état de l'art exposé dans la section 1.7 est explicité.

L'originalité de nos travaux porte sur l'introduction d'une stratégie de calcul haute performance dédiée à la résolution des problèmes en mécanique du contact élasto-statique déformable-déformable en 3D avec AMR et des frontières de contact potentiellement courbes.

1.9.1. Application du contact dans MFEM

Tout d'abord, une stratégie globale de résolution du problème de contact dans l'environnement logiciel MFEM est proposée. Comme abordé précédemment, elle repose sur trois piliers : la méthode d'appariement des nœuds présents sur les surfaces de contact, la manière dont sont définis les nœuds en contact actif et la technique d'approximation des conditions de contact.

Concernant l'appariement des nœuds en contact potentiel, c'est-à-dire actif ou non, deux stratégies sont mises en place. Les algorithmes d'appariement nœud-à-nœud, décrit dans la section 2.1.3, et nœud-à-surface, décrit dans la section 1.3.2.2, permettent ainsi de regrouper les nœuds entre eux pour l'application des conditions de Signorini. Aussi, afin de connaître quels sont les nœuds en contact actif, un algorithme basé sur la méthode des statuts est développé.

Enfin, le système linéaire résultant est résolu via l'appel de la classe MFEM `PenaltyConstrainedSolver` permettant de résoudre des systèmes contraints avec la méthode de pénalisation. Dans ces travaux, cette classe a été enrichie par l'ajout de solveurs et pré-conditionneurs pour l'analyse de la stratégie (voir les sections 3.5.4 et 3.6.2 notamment).

La mini-application qui va vous être présentée, adaptée à l'environnement logiciel MFEM et à ses classes et opérateurs, est validée par des études de convergence en maillage par le biais d'estimateurs *a priori* et vis-à-vis de de solutions analytiques disponibles sur des benchmarks académiques (contact de Hertz).

1.9.2. Adaptation de l'approche AMR dans MFEM

Dans ces travaux de thèse, la méthode de raffinement de maillage h-adaptatif déjà implémentée dans MFEM est enrichie par notre propre procédure estimate-mark-refine. En effet, l'approche AMR proposée est adaptée avec nos propres critères, notamment locaux, pour la résolution des problèmes de contact considérés. Ici, l'estimateur ZZ est appliqué indépendamment sur chaque solide.

Par ailleurs, la combinaison de l'AMR et des problèmes de contact met en évidence les problématiques des bords courbes hiérarchiquement raffinés [Wriggers et al. 1998 ; Liu et al. 2017 ; Aldakheel et al. 2020]. Afin d'éviter des procédures manuelles (résultant en un raffinement quasi-hiérarchique) garantissant que les nœuds de contact potentiels soient situés sur la frontière de la courbe pendant le processus de raffinement [Liu et al. 2017 ; Koliesnikova et al. 2021], nous proposons d'utiliser des éléments super-paramétriques afin de préserver la forme de la frontière de la géométrie initiale, même pour les solutions d'éléments finis d'ordre un. Il s'agit d'éléments dont le degré d'interpolation de la géométrie est supérieur à celui des fonctions de forme de la solution. L'approche proposée est vérifiée notamment par des études de convergence en maillage.

1.9.3. Adaptation des méthodes de partitionnement au contact

Dans l'environnement logiciel MFEM sur lequel nous avons choisi de travailler, plusieurs stratégies de répartition des éléments sur les unités de calcul sont disponibles. Elles sont basées sur les méthodes de partitionnement géométrique via Hilbert SFC (voir la section 1.5.3.1) ou de graphe via la bibliothèque Metis (voir la section 1.5.3.2). Néanmoins, appliquées telles quelles, elle ne sont pas adaptées au partitionnement pour des problèmes de contact.

Dans les travaux présentés dans ce manuscrit, trois stratégies de partitionnement du maillages sont mises en œuvre et comparées via des cas d'étude représentatifs. Les stratégies proposées sont spécifiquement adaptées au traitement de problèmes de contact, garantissent la répartition équilibrée de la charge entre les processus et ont un principe commun : les nœuds appairés de contact sont sur les mêmes processus. Ainsi, l'algorithme qui construit la matrice locale d'appariement du contact ne dépend pas de communications avec d'autres processus. Par conséquent, les coûts de communication sont réduits et le code est considérablement simplifié.

La différence entre les trois approches de partitionnement de maillage repose cependant sur la manière dont sont répartis les éléments. Deux d'entre elles ont été spécifiquement développées lors de nos travaux autour du contact, tandis que la troisième est une variante de la méthode de partitionnement Hilbert SFC adaptée spécialement pour résoudre des problèmes de contact entre deux solides déformables.

1.9.4. Apports de la thèse

La stratégie parallèle proposée permet de résoudre des problèmes de contact déformable 3D en élastostatique avec raffinement adaptatif de maillage. Elle est applicable pour traiter le contact entre deux solides de mêmes matériaux ou de propriétés matériau différentes. De plus, elle se révèle appropriée pour le raffinement de maillages structurés par blocs présentant des frontières courbes, ce même pour des solutions éléments finis d'ordre 1. Des critères AMR locaux et globaux peuvent être appliqués. De plus, la stratégie de partitionnement de maillage adaptée au contact s'est avérée adaptée et très efficace sur des supercalculateurs. En effet, elle se révèle fortement scalable jusqu'à 8192 cœurs de calcul sur des problèmes de contact avec appariement nœud-à-nœud très volumineux comportant jusqu'à 400 millions de degrés de liberté. Ces résultats constituent ainsi une réelle avancée scientifique concernant le traitement de problèmes de contact déformable 3D en nœud-à-nœud dans un cadre parallèle puisque les simulations les plus probantes à notre connaissance ont été effectuées au mieux sur 2400 cœurs de calcul et n'excèdent pas 250 millions d'inconnues. De plus, la stratégie parallèle s'est présentée prometteuse pour la résolution de problèmes de contact avec appariement nœud-à-surface puisque ses performances parallèles se révèlent intéressantes jusqu'à 512 cœurs de calcul et pour des maillages localement raffinés comportant une dizaine de millions de degrés de liberté. Ces résultats repoussent là aussi les limitations actuelles en ce qui concerne le traitement du contact avec appariement nœud-à-surface dans un cadre de calcul haute performance. En effet, à notre connaissance, les résultats très récents faisant office de référence sur cet enjeu ont présenté une scalabilité intéressante jusqu'à 250 cœurs de calcul pour des solutions uniformément raffinées de quelques dizaines de millions de degrés de liberté. Enfin, la généricité de la stratégie contact-AMR-HPC pour tous problèmes en élasto-statique où la détection du contact est simple à réaliser (contact selon une normale prédéfinie fixe) permet de l'étendre à un large panel de problèmes, qu'ils soient dans le domaine spécifique du combustible nucléaire ou non.

Le travail exposé dans ce manuscrit est intégré pour partie dans un article soumis [Epalle et al. 2024c]. Il a aussi fait l'objet de plusieurs présentations dans des conférences : ADMOS [Epalle et al. 2023b], ICCCM [Epalle et al. 2023a], CSMA [Epalle et al. 2024d], ECCOMAS [Epalle et al. 2024a] et WCCM [Epalle et al. 2024b]. L'article porte sur les travaux introduits dans le chapitre 2 et sur les premières analyses de

performances parallèles du chapitre 3 (concernant la première méthode de partitionnement considérée). Les travaux exposés dans les conférences ADMOS [Epalle et al. 2023b] et ICCCM [Epalle et al. 2023a] se sont focalisés sur les résultats du chapitre 2. Les présentations aux conférences CSMA [Epalle et al. 2024d], ECCOMAS [Epalle et al. 2024a] et WCCM [Epalle et al. 2024b] se sont davantage concentrées sur les deux dernières approches de partitionnement considérées, décrites dans le chapitre 3.

Chapitre 2

Raffinement adaptatif de maillage en mécanique du contact avec frontières courbes

Ce chapitre décrit un algorithme générique, robuste et efficace permettant de traiter des problèmes en mécanique élasto-statique du contact avec raffinement adaptatif de maillage que nous avons implémenté dans MFEM. Cette stratégie vise ainsi à résoudre des problèmes 2D et 3D mettant en jeu un contact entre solides déformables présentant des frontières courbes. L'apport de l'approche proposée réside dans le fait qu'elle est applicable pour un raffinement h-adaptatif non-conforme de maillages structurés quadrangulaires ou hexaédriques et adaptable pour différents critères de détection et d'arrêt. La section 2.1 décrit la manière dont est traité le problème de contact. L'approche estimate-mark-refine considérée, adaptée à la gestion de bords courbes, et l'adaptation proposée pour la résolution du contact dans l'environnement logiciel MFEM (incluant la gestion des non-conformités) sont ensuite présentées dans la section 2.2. Enfin, la vérification de la solution de contact et l'intérêt de l'emploi de l'AMR sont évalués sur des problèmes de contact Hertzien en 2D et 3D dans la section 2.3. Aussi, différents critères de détection de l'AMR sont comparés et diverses analyses de convergence sont effectuées.

Sommaire

2.1. Modélisation du contact . . . 60
2.1.1. Forme discrète . . . 60
2.1.2. Forme discrète pénalisée . . . 60
2.1.3. Appariement nœud-à-nœud . . . 61
2.1.4. Traitement algorithmique du problème de contact pénalisé . . . 62
2.2. Processus de raffinement adaptatif de maillage . . . 63
2.2.1. Estimation de l'erreur : module ESTIMATE . . . 63
2.2.2. Étape de marquage : module MARK . . . 64
2.2.3. Critères d'arrêt : module STOP . . . 65
2.2.4. Raffinement du maillage : module REFINE . . . 66
2.2.5. Résolution non-conforme : module SOLVE . . . 68
2.2.6. Traitement algorithmique du processus combinant AMR et résolution du contact . . . 69
2.3. Application au problème du contact Hertzien . . . 70
2.3.1. Justification du paramètre de pénalisation k_N . . . 71
2.3.2. Comparaison avec la solution analytique de Hertz . . . 76
2.3.3. Sur les stratégies AMR . . . 77
2.4. Conclusion . . . 82

2.1. Modélisation du contact

Comme abordé en section 1.8.4, la méthode de pénalisation a été retenue comme technique d'approximation numérique des conditions de contact de Signorini (voir les équations (1.2)-(1.4) en page 30) et un appariement nœud-à-nœud est considéré.

2.1.1. Forme discrète

De manière générique, un problème de mécanique de contact sous forme discrète peut s'écrire

$$(\mathcal{P}') : \begin{cases} \mathbf{L}^{int}(\mathbf{U}) - \mathbf{L}^{ext} - \mathbf{L}^{C}(\mathbf{U}) & = \mathbf{0} \\ \mathbf{BU} & \leq \mathbf{D} \end{cases}, \tag{2.1}$$

où $\mathbf{U}$ dénote la solution du problème discret, tandis que $\mathbf{L}^{int}$, $\mathbf{L}^{ext}$ et $\mathbf{L}^{C}$ désignent les forces interne, externe et de contact respectivement. De plus, $\mathbf{B}$ est la matrice d'appariement des nœuds de contacts potentiels utilisée pour définir les déplacements relatifs discrets et $\mathbf{D}$ les jeux initiaux entre les nœuds appairés.

De ce fait, dans l'équation (2.1), le symbole '$\leq$' est utilisé pour représenter l'inégalité composante par composante.

Remarque : le terme $\mathbf{L}^{C}(\mathbf{U})$ doit être compris dans un sens générique. Lorsqu'une stratégie numérique basée sur les multiplicateurs de Lagrange (voir la section 1.3.3.2) ou ses dérivés (Lagrangien augmenté, etc.) est appliquée, la dépendance en $\mathbf{U}$ est remplacée par une dépendance en des variables additionnelles associées à la méthode.

Dans le cas élasto-statique considéré, les forces internes se réduisent à

$$\mathbf{L}^{int}(\mathbf{U}) = \mathbf{KU}, \tag{2.2}$$

avec $\mathbf{K}$ la matrice de raideur.

2.1.2. Forme discrète pénalisée

Cette sous-section s'appuie sur les hypothèses décrites en section 1.3.1 et les relations discrètes formulées en section 2.1.1. La méthode de pénalisation consiste à réécrire le problème d'inégalité de contact (2.1) comme une égalité variationnelle, beaucoup plus facile à résoudre par les solveurs classiques. Par ailleurs, une méthode des statuts (ou des contacts actifs) [Luenberger et al. 1984 ; Murty 1988 ; Luenberger 2003 ; Wriggers 2006] est utilisée pour linéariser le problème. En effet, la zone à pénaliser (zone de contact actif) dépend de la solution du problème. Cette méthode des statuts est une méthode itérative qui consiste à identifier l'ensemble des conditions actives de contact actif, désigné $\mathcal{A}$, défini par

$$\mathcal{A} = \{i \in [\![1; N_C]\!] \; ; \; \sum_{j=1}^{N} \mathbf{B}_{ij}\mathbf{U}_j \geq \mathbf{D}_i\} \tag{2.3}$$

avec N le nombre de degrés de liberté (inconnues primales), et N_C le nombre de contacts potentiels. Il est à noter qu'en pratique la somme se réduit aux degrés de liberté associés aux nœuds situés sur la surface de contact. Alors, la matrice d'appariement active $\hat{\mathbf{B}}$ est obtenue en restreignant les lignes de $\mathbf{B}$ à l'ensemble $\mathcal{A}$, voir l'équation (2.4). Le vecteur

des jeux "actifs" $\hat{\mathbf{D}}$ est défini de manière similaire :

$$\begin{cases} \hat{\mathbf{B}} & = \mathbf{B}\left[\mathcal{A}\right] \\ \hat{\mathbf{D}} & = \mathbf{D}\left[\mathcal{A}\right] \end{cases}. \tag{2.4}$$

La force de réaction de contact est définie dans le but de pénaliser l'interpénétration de la surface. Ainsi, la formulation de contact est régularisée, voir la figure 1.7b en page 33, et les forces de contact pénalisées discrètes $\mathbf{L}^C(\mathbf{U})$ sont données par l'équation (2.5).

$$\mathbf{L}^C(\mathbf{U}) = k_N \hat{\mathbf{B}}^T \left(\hat{\mathbf{D}} - \hat{\mathbf{B}}\mathbf{U}\right) \tag{2.5}$$

avec $k_N \gg 1$ le coefficient de pénalisation en $\mathrm{N.m^{-1}}$, ce qui peut être interprété comme une raideur de ressort dans l'interface de contact, voir [Ohtake et al. 1980 ; Ben Dhia 1988]. Enfin, l'équilibre mécanique s'écrit

$$\left(\mathbf{K} + k_N \hat{\mathbf{K}}_C\right) \mathbf{U} = \mathbf{L}^{ext} + k_N \hat{\mathbf{B}}^T \hat{\mathbf{D}} \tag{2.6}$$

avec $\hat{\mathbf{K}}_C = \hat{\mathbf{B}}^T \hat{\mathbf{B}}$.

Une fois la solution $\mathbf{U}$ trouvée, l'ensemble actif est mis à jour via l'équation (2.3), et le système linéaire (2.6) est de nouveau résolu. Ce processus est itéré jusqu'à ce que la zone active de contact ne varie plus en itération. Comme mentionné en section 1.8.4, le problème (2.6) implique uniquement des inconnues primales. D'autre part, avec l'approche par pénalisation, une légère interpénétration entre les solides est permise [Ohtake et al. 1980 ; Ben Dhia 1988], étant donné qu'il s'agit du terme moteur de la force de contact.

La classe `PenaltyConstrainedSolver` de l'environnement logiciel MFEM est utilisée pour former le système linéaire (2.6) à partir du coefficient de pénalisation k_N, des matrices et des vecteurs $\mathbf{K}$, $\hat{\mathbf{B}}$, $\mathbf{L}^{ext}$ et $\hat{\mathbf{D}}$ fournis en entrée.

2.1.3. Appariement nœud-à-nœud

Pour résoudre le problème de contact décrit par l'équation (2.6), une étape préliminaire consistant à appairer les nœuds de contact potentiels est nécessaire. En effet, cette étape permet de construire la matrice $\mathbf{B}$ et le vecteur $\mathbf{D}$, à partir des positions des nœuds du maillage situés sur les bords de contact des solides. Le processus consiste à appairer les nœuds de contact potentiels les plus proches à l'aide d'un critère de position.

En mécanique du contact numérique, il existe de plusieurs manières d'appairer les nœuds de l'interface de contact, voir la section 1.3.2. Les plus connues sont les algorithmes d'appariement nœud-à-nœud, nœud-à-surface et surface-à-surface. Ici, l'approche d'appariement nœud-à-nœud est choisie pour sa facilité d'implémentation et ses coûts de calcul très faibles. Cependant, elle restreint le problème et la géométrie à prendre en compte. Les maillages initiaux doivent avoir des interfaces de contact concordantes. En outre, cette approche d'appariement n'est valide que sous l'hypothèse des petites perturbations. Dans ce cadre, les nœuds de contact potentiels se font toujours face.

Ainsi, le logiciel MFEM est enrichi de notre propre algorithme d'appariement nœud-à-nœud qui consiste à trouver, pour chaque nœud du premier solide, le nœud le plus proche du second solide dans la direction de la normale au contact (donnée *a priori*). L'algorithme d'appariement des nœuds situés sur les bords de contact mis en œuvre se décompose en

deux boucles imbriquée :

— Une boucle externe consistant à parcourir tous les nœuds de la première surface de contact et à répertorier leur position.

— Une boucle interne comparant les coordonnées tangentielles au contact (selon la normale orthogonale à la normale de contact) de chaque nœud de la seconde surface de contact avec le nœud de la première surface considérée. Dès que les positions tangentielles correspondent, la paire de nœuds est répertoriée et la boucle interne s'arrête.

2.1.4. Traitement algorithmique du problème de contact pénalisé

Le processus proposé pour résoudre le problème de contact est régi par un processus itératif, voir l'algorithme 1 ci-après. L'appariement nœud-à-nœud est réalisé avant d'entrer dans la boucle, tandis que l'ensemble des paires de nœuds en contact actif $\mathcal{A}$ est mis à jour à la fin de chaque itération de l'algorithme des statuts en examinant l'interpénétration entre les solides, voir la définition (2.3). L'ensemble des nœuds en contact actif peut donc varier à chaque itération, et la matrice et le vecteur de contact, $\hat{\mathbf{K}}_C$ et $\hat{\mathbf{B}}^T\hat{\mathbf{D}}$ respectivement, sont reconstruits en conséquence. Pour leur part, la matrice de rigidité $\mathbf{K}$ et le vecteur $\mathbf{L}^{ext}$ sont formés en amont de cette boucle itérative. Le système linéaire résultant, voir l'équation (2.6), est résolu par un solveur itératif linéaire de type gradient conjugué pré-conditionné (PCG) car il est adapté aux problèmes pénalisés résultant en de grands systèmes symétriques définis positifs. Le critère d'arrêt du processus de résolution des contacts exige que l'ensemble des nœuds de contact actifs ne varie plus d'une itération à l'autre.

Algorithme 1 Processus de résolution du contact

Entrée : Données du problème, matrice de rigidité K, vecteur $\mathbf{L}^{ext}$, nombre maximum d'itérations de boucle de contact l_{max}
Sortie : Solution éléments finis *U*
Initialisation
  Procéder à l'appariement nœud-à-nœud
  Initialiser les paramètres de boucle de contact : $l \leftarrow 1$, $\boldsymbol{stop_C} \leftarrow faux$
  Définir l'ensemble initial des nœuds en contact actif (s'il y en a)
tant que ($\boldsymbol{stop_C} = faux$ et $l \leq l_{max}$) **faire**
  Former le système linéaire pénalisé
  Résoudre le système linéaire
  Définir le nouvel ensemble des nœuds en contact actif
  si pas d'évolution de l'ensemble des nœuds en contact actif **alors**
    $\boldsymbol{stop_C} \leftarrow vrai$
  fin si
  $l \leftarrow l + 1$
fin tant que

Remarques :

— Par défaut, il n'y a pas de nœuds en contact actif.

— Former le système linéaire pénalisé consiste à construire la matrice d'appariement active $\hat{\mathbf{B}}$ et le vecteur des jeux "actifs" $\hat{\mathbf{D}}$. Puis, au sein de la classe

`PenaltyConstrainedSolver` de MFEM, la matrice $\hat{\mathbf{K}}_C$ est formée et les termes de contact sont ajoutés au système linéaire (sans contact) pour former le système linéaire pénalisé.

— Le nouvel ensemble des nœuds en contact actif est défini via l'application de la méthode des statuts.

2.2. Processus de raffinement adaptatif de maillage

Comme évoqué à la page 36, la forte non linéarité locale du problème de contact exige un maillage suffisamment fin dans la zone de contact pour l'obtention de solutions précises. Cependant, l'utilisation de maillages uniformément raffinés fins s'avère très coûteux, en particulier en 3D, et des techniques de raffinement adaptatif de maillage sont donc utiles pour limiter le nombre de degrés de liberté des simulations considérées. Ainsi, l'intérêt est ici de combiner l'AMR à la résolution du problème de contact et d'en dresser une procédure algorithmique présentée dans la section 2.2.6.

Pour rappel, le raffinement adaptatif de maillage consiste à modifier ou enrichir localement et de manière automatique le maillage initial afin qu'il respecte une précision donnée sur la solution.

Dans les sections suivantes, les principaux éléments de chaque module (aussi appelé étape de calcul) de l'algorithme AMR combiné avec la résolution du contact sont décrits (voir la figure 1.14 en page 54).

2.2.1. Estimation de l'erreur : module ESTIMATE

L'ingrédient clé pour l'adaptation automatique du maillage est l'estimateur d'erreur. Comme décrit en section 1.8.3, ces travaux n'ont pas pour but de comparer l'efficacité des estimateurs d'erreur, et l'estimateur d'erreur *a posteriori* de Zienkiewicz et Zhu [O. C. Zienkiewicz et al. 1987] est défini comme l'outil permettant de piloter le raffinement du maillage. Cet estimateur est basé sur la différence entre un champ de contraintes lissé $\boldsymbol{\sigma}^\star$ et celui obtenu par résolution éléments finis $\boldsymbol{\sigma}^h$. Ce dernier est, pour une résolution éléments finis en déplacement, généralement discontinu entre les éléments. Pour la construction du champ lissé, la technique de projection nodale introduite dans [O. C. Zienkiewicz et al. 1987] est mise en œuvre. Soit Ω^h la quadrangulation de Ω, chaque élément de maillage est dénoté $T_i, i \in [\![1; N_E]\!]$ tel que $\overline{\Omega^h} = \cup_i^{N_E} \overline{T_i}$. Ainsi, N_E représente le nombre total d'éléments du maillage. La distribution d'erreur élémentaire locale en norme énergie s'écrit pour un élément de maillage T_i

$$\xi_{T_i} = \left(\int_{T_i} \left(\boldsymbol{\sigma}^\star - \boldsymbol{\sigma}^h \right) : \left(\boldsymbol{\varepsilon}^\star - \boldsymbol{\varepsilon}^h \right) dT_i \right)^{1/2}. \tag{2.7}$$

Remarque : en pratique, cette intégrale est évaluée aux point d'intégration où $\boldsymbol{\sigma}^h$ est définie.

À l'instar de [Liu et al. 2017], l'estimateur ZZ classique est appliqué dans chaque solide indépendamment sans aucune autre modification. L'erreur estimée absolue globale sur Ω^h est ensuite écrite par sommation des contributions élémentaires

$$\xi_\Omega = \left(\sum_{T_i \in \Omega^h} \xi_{T_i}^2 \right)^{1/2}. \tag{2.8}$$

Dans la bibliothèque logicielle MFEM, les $\{\xi_{T_i}\}_i$ sont calculées en utilisant la fonction `GetLocalErrors` appliquée à l'estimateur d'erreur `ZienkiewiczZhuEstimator`. La valeur scalaire ξ_Ω est ensuite obtenue à partir des $\{\xi_{T_i}\}_i$ grâce à la fonction `GetNorm`.

2.2.2. Étape de marquage : module MARK

2.2.2.1. Critères d'optimalité du maillage : sous-module OPTIMALITY

Dans la littérature, la stratégie de sélection des éléments à raffiner repose essentiellement sur une utilisation qualitative du champ d'erreur élémentaire estimé. Les approches qualitatives impliquent une détection empirique des éléments à raffiner. Les plus connues sont le marquage par quantile, le marquage de Dörfler ou le marquage par maximum, voir [Dörfler 1996] pour plus de détails. Dans ce cas, le choix optimal des paramètres requis n'est pas évident, étant donné que ces paramètres ne sont pas directement liés à une précision souhaitée et peuvent dépendre du problème considéré, voir [Barbié et al. 2015 ; Ramière et al. 2019] par exemple.

Dans nos développements, nous nous appuyons plutôt sur des approches quantitatives de marquage, liées à une précision prescrite par l'utilisateur. À cette fin, une erreur maximale admissible par élément doit être définie. Deux critères d'optimalité du maillage, axés sur une erreur globale ou locale, sont étudiés.

Critère ZZ : equidistribution de l'erreur spécifique

Dans [O. C. Zienkiewicz et al. 1987], les auteur ont proposé un critère d'optimalité du maillage basé sur le principe d'équidistribution de l'erreur :

$$\xi_{T_i}^{\max,\mathrm{ZZ}} = \text{const} \quad \forall T_i \in \Omega^h. \tag{2.9}$$

L'erreur absolue globale estimée sur Ω^h est alors définie par

$$\xi_\Omega^{\max,\mathrm{ZZ}} = e_\Omega \omega_\Omega = \sqrt{N_E}\, \xi_{T_i}^{\max,\mathrm{ZZ}} \tag{2.10}$$

avec N_E le nombre d'éléments du maillage, e_Ω la précision prescrite par l'utilisateur et ω_Ω la racine carrée de l'énergie de déformation globale. Cette dernière valeur est obtenue en additionnant les contributions élémentaires

$$\omega_\Omega = \left(\sum_{T_i \in \Omega^h} \omega_{T_i}^2 \right)^{1/2} \tag{2.11}$$

où l'énergie de déformation $\omega_{T_i}^2$ sur l'élément T_i, introduite dans [O. C. Zienkiewicz et al. 1987], est donnée par

$$\omega_{T_i}^2 = \int_{T_i} \boldsymbol{\sigma}^h : \boldsymbol{\varepsilon}^h dT_i + \int_{T_i} \left(\boldsymbol{\sigma}^\star - \boldsymbol{\sigma}^h\right) : \left(\boldsymbol{\varepsilon}^\star - \boldsymbol{\varepsilon}^h\right) dT_i \;\forall\; T_i \in \Omega^h. \tag{2.12}$$

Ainsi, l'erreur élémentaire absolue maximale tolérée est définie par

$$\xi_{T_i}^{\max,\mathrm{ZZ}} = e_\Omega \frac{\omega_\Omega}{\sqrt{N_E}}. \tag{2.13}$$

Le critère (2.13) est le critère d'optimalité du maillage le plus répandu à ce jour.

Critère LOC : erreur élémentaire locale

Dans [Ramière et al. 2019], un critère d'optimalité locale (LOC) est introduit et il a été démontré dans [Koliesnikova et al. 2021] qu'il était plus fiable que le critère ZZ pour le contrôle de l'erreur locale dans le cadre des problèmes de mécanique des structures. En outre, ce critère a été évalué avec succès dans [Liu et al. 2017] pour les problèmes de contact. La distribution élémentaire maximale admissible de l'erreur est directement régie par l'erreur élémentaire locale admissible

$$\xi_{T_i}^{\max,\mathrm{LOC}} = e_{\Omega,\mathrm{LOC}}\, \omega_{T_i}. \tag{2.14}$$

Comme prouvé dans [Ramière et al. 2019], cela implique automatiquement que l'erreur globale relative est majorée par $e_{\Omega,\mathrm{LOC}}$.

Dans la bibliothèque logicielle MFEM, la fonction `Apply` dédiée à marquer les éléments pour raffinement a été enrichie avec les implémentations nécessaires pour évaluer les erreurs maximales autorisées : calculs de ω_{T_i}, ω_Ω, $\xi_{T_i}^{\max,\mathrm{ZZ}}$ et $\xi_{T_i}^{\max,\mathrm{LOC}}$.

2.2.2.2. Détection des éléments à raffiner : sous-module DETECT

À partir de l'erreur élémentaire maximale admissible, l'ensemble des éléments marqués est simplement défini par

$$\mathcal{M}^0 = \left\{ T_i \in \Omega^h \;/\; \xi_{T_i} > \xi_{T_i}^{\max} \right\}. \tag{2.15}$$

Afin de garantir que les bords de contact raffinés restent concordants (ce qui est nécessaire pour l'appariement de contact nœud-à-nœud), l'ensemble $\mathcal{M}^0$ doit généralement être élargi. Les éléments situés sur les frontières de contact potentielles dont l'élément appairé est marqué pour raffinement sont alors également marqués pour raffinement, comme proposé dans [Liu et al. 2017]. Cet ensemble s'écrit donc

$$\mathcal{M} = \mathcal{M}^0 \cup \left\{ T_j \in \Omega^h \;/\; T_i \in \mathcal{M}^0 \text{ appairé avec } T_j \right\}. \tag{2.16}$$

Ici, nous définissons des éléments finis comme "appairés" lorsque tous leurs nœuds situés sur les bords de contact sont mutuellement appairés.

À cette fin, la fonction MFEM `Apply` a été enrichie pour recevoir des éléments appairés en entrée.

2.2.3. Critères d'arrêt : module STOP

Le processus AMR s'arrête de manière évidente lorsque l'ensemble des éléments marqués devient vide, c'est-à-dire lorsque

$$\mathcal{M} = \emptyset. \tag{2.17}$$

Cependant, pour les problèmes présentant des singularités, tels que ceux de la mécanique du contact, le critère (2.17) peut ne jamais être atteint et un raffinement infini dans des zones très localisées peut avoir lieu. D'autres critères d'arrêt sont donc couramment ajoutés, ce qui permet de mettre fin au processus de raffinement une fois que le maillage est considéré comme acceptable.

Dans ces travaux, la vérification des précisions globales ou locales est d'intérêt. Le processus de raffinement est donc analysé à l'aide de deux critères d'arrêt différents.

Critère d'arrêt global

Le critère d'arrêt le plus couramment utilisé dans la littérature est un critère global basé sur l'erreur globale absolue estimée (voir [O. C. Zienkiewicz et al. 1987]). Il est défini par

$$\frac{\xi_\Omega}{e_\Omega \omega_\Omega} \leq 1. \tag{2.18}$$

Critère d'arrêt local

Lorsque le contrôle de l'erreur locale est d'intérêt (voir critère de détection (2.14)), un critère d'arrêt local, introduit dans [Ramière et al. 2019] et basé sur des considérations géométriques, est utilisé pour arrêter le processus de raffinement. En effet, le respect du critère d'arrêt global (2.18) ne garantit aucune précision au niveau local. Le critère d'arrêt local permet ainsi d'assurer que la mesure de l'ensemble des éléments détectés $\mathcal{M}$ est inférieure à une proportion (δ) de la mesure totale du domaine. Il s'écrit

$$\mu\left(\Omega_{\mathcal{M}}\right) \leq \delta \cdot \mu\left(\Omega^h\right) \tag{2.19}$$

avec $\mu\left(\Omega_{\mathcal{M}}\right)$ la mesure de l'aire détectée $\overline{\Omega_{\mathcal{M}}} = \left\{\cup \overline{T_i} \ / \ T_i \in \mathcal{M}\right\}$, $\mu\left(\Omega^h\right)$ la mesure du maillage global, et δ le paramètre de contrôle d'erreur locale défini par l'utilisateur.
Avec ce critère d'arrêt, la solution obtenue satisfait le critère d'optimalité locale (2.14) sur $(1-\delta) \times 100$ pourcent du domaine. Ce critère d'arrêt local a donc l'avantage de déterminer automatiquement une approximation discrète de la zone critique en présence de singularités de contact, ce qui est d'un grand intérêt.

Remarque : En fixant $\delta = 0$, le critère (2.19) revient au critère (2.17). De plus, le choix de $\delta \ll 1$ empêche une représentation approximative de la zone critique.

2.2.4. Raffinement du maillage : module REFINE

Le raffinement du maillage est basé sur un raffinement hiérarchique isotrope, consistant à diviser chaque élément quad/hexa marqué (également appelé élément parent) en 2^m (avec m la dimension de l'espace) sous-éléments du même type (aussi appelés éléments enfants), sur la base de la division de chaque côté en 2 parties égales. Le ratio de raffinement est choisi égal à 2 en suivant les recommandations fournies par l'étude menée dans [Koliesnikova et al. 2021].

En outre, la "règle d'une unique non-conformité" (également appelée "règle de l'équilibre 2 :1"), détaillée dans [Nochetto et al. 2012 ; Koliesnikova et al. 2021] par exemple et largement utilisée dans la littérature, est appliquée. Cette règle a pour effet de limiter à un le nombre de nœuds non-conformes (*hanging nodes* en anglais) sur chaque arête.

Cependant, pour des éléments finis d'ordre un, ce raffinement hiérarchique du maillage limite la représentation de la géométrie lorsque nous considérons une frontière courbe, ce qui est généralement le cas des frontières de contact. Dans la littérature (voir [Liu et al. 2017] par exemple), des procédures artisanales de raffinement de maillage dites "quasi-hiérarchiques" ont été proposées. Elles consistent principalement à mettre à jour

la discrétisation de la frontière courbe en déplaçant les nœuds ajoutés par raffinement sur la géométrie courbe exacte, voir figure 2.1. Cette procédure donne des résultats très satisfaisants, mais elle est intrusive et non générique car elle nécessite la connaissance de l'équation de forme fermée de la frontière courbe pendant l'étape de raffinement.

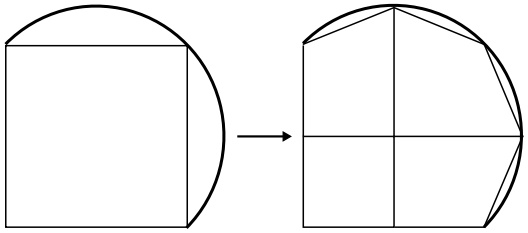
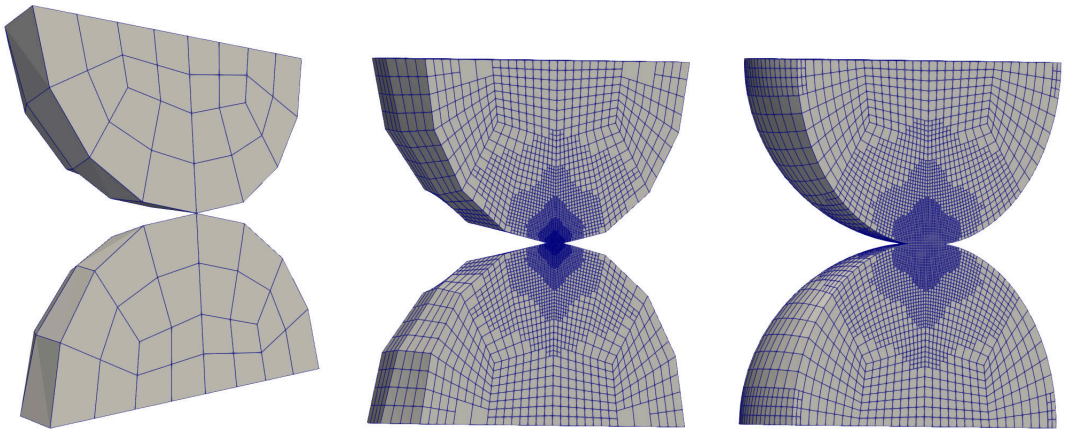


FIGURE 2.1. – Raffinement de maillage isotrope quasi-hiérarchique.

Dans ces travaux, la préservation de la géométrie du domaine de calcul est automatiquement garantie par l'utilisation d'éléments finis super-paramétriques. Ces éléments sont basés sur une transformation géométrique d'un degré d'interpolation plus élevé que les fonctions de forme de la solution. Avec ce type d'élément, l'erreur d'approximation de la surface pendant le raffinement est fortement réduite par rapport à celle obtenue avec une transformation d'ordre un, voir la figure 2.2. Ces éléments sont donc parfaitement adaptés aux problèmes en mécanique du contact où la zone active de contact doit être caractérisée avec précision.



(a) Maillage grossier initial

(b) AMR hiérarchique avec éléments isoparamétriques (transformation linéaire)

(c) AMR hiérarchique avec éléments super-paramétriques (transformation d'ordre 6)

FIGURE 2.2. – AMR hiérarchique pour problèmes de contact à bords courbes avec solution éléments finis d'ordre un : éléments isoparamétriques en (b) versus super-paramétriques en (c).

Ce type d'élément super-paramétrique est défini lors de la phase de génération du maillage par le mailleur Gmsh [GEUZAINE et al. 2009] et est correctement pris en charge par l'environnement logiciel MFEM. Par conséquent, la géométrie discrète n'a pas besoin d'être mise à jour au cours du processus de raffinement.

La procédure de raffinement de MFEM, `Apply`, est invoquée pour gérer la stratégie de raffinement de maillage h-adaptative non-conforme choisie [ČERVENÝ et al. 2019 ; ANDERSON et al. 2020]. La règle d'une unique non-conformité par arête est activée en fixant le paramètre MFEM `nc_limit` à 1.

2.2.5. Résolution non-conforme : module SOLVE

Le raffinement d'un maillage composé d'éléments quadrangulaires ou hexaédriques à l'aide de la méthode hiérarchique h-adaptive entraîne l'apparition de nœuds non-conformes, voir la figure 2.3. Les degrés de liberté (ddls) non-conformes doivent donc être contraints afin d'obtenir une solution continue.

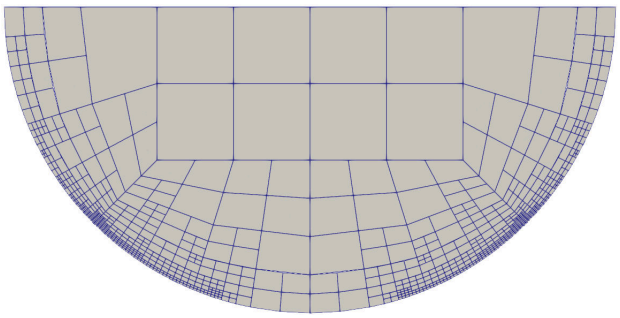


FIGURE 2.3. – Exemple d'un maillage raffiné h-adaptatif non-conforme.

Une tâche centrale dans l'utilisation de ce type de méthode de raffinement est de construire le système linéaire à inverser. Le choix fait dans MFEM et détaillé dans [ČERVENÝ et al. 2019] est de définir (et de résoudre) un problème restreint aux seuls ddls conformes. Son principe est rappelé ci-après afin de présenter son extension aux problèmes de contact.

La matrice de raideur globale (forme bilinéaire) $\tilde{\mathbf{K}}$ et le vecteur second membre du système linéaire éléments finis $\tilde{\mathbf{L}}^{ext}$ sont assemblés en prenant en compte tous les nœuds du maillage, sans s'inquiéter des non-conformités. Ainsi, le système linéaire résultant s'écrit

$$\tilde{\mathbf{K}}\tilde{\mathbf{U}} = \tilde{\mathbf{L}}^{ext} \tag{2.20}$$

avec $\tilde{\mathbf{U}}$ le vecteur contenant tous les ddls.
Dans le but de restreindre ce système aux ddls conformes, un opérateur de prolongation $\mathbf{P}$ est construit. Il s'écrit

$$\mathbf{P} = \begin{pmatrix} \mathbf{Id} \\ \mathbf{W} \end{pmatrix} \tag{2.21}$$

avec $\mathbf{Id}$ la matrice identité et $\mathbf{W}$ la matrice dite d'interpolation. Cette matrice d'interpolation $\mathbf{W}$ exprime la relation reliant les ddls non-conformes aux ddls conformes. Il s'agit de combinaisons linéaires utilisant les fonctions de forme éléments finis (voir [ČERVENÝ et al. 2019 ; ANDERSON et al. 2020] pour plus de détails sur la procédure pour la construction de cette matrice). Par conséquent, nous avons

$$\tilde{\mathbf{U}} = \mathbf{P}\mathbf{U} \tag{2.22}$$

avec $\mathbf{U}$ les ddls conformes. De la même manière, les fonctions test sont projetées sur l'espace conforme grâce à l'opérateur de prolongation $\mathbf{P}$. Par conséquent, le système linéaire (2.20) devient

$$\mathbf{P}^T\tilde{\mathbf{K}}\mathbf{P}\mathbf{U} = \mathbf{P}^T\tilde{\mathbf{L}}^{ext} \tag{2.23}$$

En définissant la matrice de raideur conforme

$$\mathbf{K} = \mathbf{P}^T\tilde{\mathbf{K}}\mathbf{P} \tag{2.24}$$

et le vecteur second membre conforme

$$\mathbf{L}^{ext} = \mathbf{P}^T \tilde{\mathbf{L}}^{ext}, \tag{2.25}$$

le système linéaire à résoudre pour obtenir la solution $\mathbf{U}$ sur les ddls conformes s'écrit

$$\mathbf{K}\mathbf{U} = \mathbf{L}^{ext}. \tag{2.26}$$

Une fois que la solution éléments finis $\mathbf{U}$ est trouvée, elle est étendue à tous les ddls du maillage par le biais de l'opérateur de prolongation (2.22).

Les mêmes opérations de projection doivent être réalisées sur les termes de contact apparaissant dans le système linéaire pénalisé (2.6). Comme déjà indiqué dans la section 2.1.2, nous avons choisi de nous appuyer sur la classe `PenaltyConstrainedSolver` de MFEM qui fonctionne sur des matrices et vecteurs associés à des ddls conformes. Ainsi, il est suffisant de définir la matrice d'appariement de contact actif sur les ddls conformes par

$$\hat{\mathbf{B}} = \tilde{\mathbf{B}}[\mathcal{A}]\mathbf{P} \tag{2.27}$$

où $\tilde{\mathbf{B}}$ est la matrice d'appariement de contact potentiel assemblée en prenant en compte tous les ddls, et $\mathcal{A}$ l'ensemble des nœuds en contact actif, voir la section 2.1.2.
Par conséquent, la matrice de raideur de contact construite par la classe `PenaltyConstrainedSolver` s'écrit

$$\hat{\mathbf{K}}_C = \mathbf{P}^T \tilde{\mathbf{B}}[\mathcal{A}]^T \tilde{\mathbf{B}}[\mathcal{A}]\mathbf{P}, \tag{2.28}$$

ce qui correspond à la projection sur les ddls conformes de la matrice de raideur de contact globale $\tilde{\mathbf{K}}_C = \tilde{\mathbf{B}}[\mathcal{A}]^T \tilde{\mathbf{B}}[\mathcal{A}]$. La projection du terme de contact situé à droite de l'égalité est aussi cohérente et s'écrit

$$\hat{\mathbf{B}}^T \hat{\mathbf{D}} = \mathbf{P}^T \tilde{\mathbf{B}}[\mathcal{A}]^T \hat{\mathbf{D}}. \tag{2.29}$$

Ainsi, le terme de contact situé à droite de l'égalité $\hat{\mathbf{D}}$ (celui donné en entrée de `PenaltyConstrainedSolver`) est directement

$$\hat{\mathbf{D}} = \tilde{\mathbf{D}}[\mathcal{A}]. \tag{2.30}$$

Il est à noter que ces dernières opérations algébriques ne sont pas directement disponibles dans les classes MFEM et ont été implémentées spécifiquement pour la résolution des problèmes de contact.

2.2.6. Traitement algorithmique du processus combinant AMR et résolution du contact

Le processus de résolution couplant AMR et contact est régi par deux boucles itératives imbriquées : la boucle interne pour la résolution du contact et la boucle externe pour l'application de l'approche AMR. Elles sont décrites dans l'algorithme 2. Cet algorithme appelle l'algorithme 1 dans lequel la définition de l'ensemble initial des nœuds en contact actif est généralement effectuée en considérant la solution à l'itération AMR précédente (pour des questions d'efficacité des calculs).

Algorithme 2 Processus de résolution combiné AMR-Contact

Entrée : Données du problème, nombre maximum d'itérations de boucle AMR n_{max}
Sortie : Solution éléments finis sur maillage raffiné non-conforme
Initialisation
$n \leftarrow 1$, $\boldsymbol{stop_{AMR}} \leftarrow faux$
tant que $\boldsymbol{stop_{AMR}} = faux$ et $n \leq n_{max}$ **faire**
Assembler les formes bilinéaire et linéaire
Étape de restriction.
Former le système linéaire conforme
Résoudre le contact (SOLVE) : appel de l'algorithme 1
Étape de prolongation.
Reconstruire la solution éléments finis sur tous les ddls
Détecter les éléments à raffiner (ESTIMATE-MARK)
Mettre à jour $\boldsymbol{stop_{AMR}}$ avec le critère d'arrêt (STOP)
si $\boldsymbol{stop_{AMR}} = faux$ **alors**
Raffiner les éléments détectés (REFINE) :
Créer le nouveau maillage raffiné non-conforme
fin si
$n \leftarrow n + 1$
fin tant que

Remarque : dans le cas où l'itération AMR courante est la première, alors il n'y a pas d'itération AMR précédente. Par conséquent, dans le cas où des conditions aux limites de Dirichlet sont appliquées, cet ensemble est nul. Ce ne sera pas le cas néanmoins si des conditions aux limites de Neumann sont appliquées.

2.3. Application au problème du contact Hertzien

Cette section est consacrée à l'illustration de l'efficacité de la stratégie proposée et de son adéquation aux problèmes en mécanique du contact avec des bords courbes. À cette fin, les problèmes de référence académiques de Hertz décrits dans [Hertz 1882] sont considérés en 2D et en 3D. Il s'agit de cas tests de référence bien connus en mécanique du contact, d'autant plus que des expressions analytiques du champ de contraintes dans la zone de contact sont disponibles pour réaliser des validations numériques. Le problème mécanique considéré est un problème de mécanique du contact élastique, dont la formation formulation forte a été introduite au chapitre 1 avec les équations (1.1)-(1.4). Sauf mention contraire, les deux demi-disques 2D (respectivement deux demi-sphères 3D) mis en contact sont des matériaux élastiques isotropes avec le même module d'Young $E = 210$ GPa et un coefficient de Poisson $\nu = 0,3$. Leur rayon est égal à $R = 2$ m. Des conditions aux limites de Dirichlet de valeur

$$u_D = \frac{\delta_0}{2} + \alpha R, \tag{2.31}$$

où $\delta_0 = 2$ m est le jeu minimal initial entre les solides et α un coefficient positif, sont imposées sur les bords plans tels que représentés par exemple en 2D dans la figure 2.4.

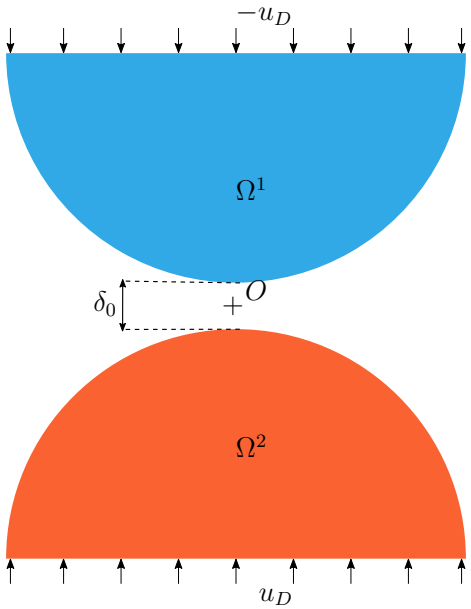


FIGURE 2.4. – Problème du contact Hertzien 2D.

Deux combinaisons de critères de détection et d'arrêt pour l'AMR sont considérées :

— **combinaison AMR 1 :** critère d'optimalité ZZ (2.13) et critère d'arrêt global (2.18) ;

— **combinaison AMR 2 :** critère d'optimalité LOC (2.14) et critère d'arrêt local (2.19).

Les grandeurs d'intérêt dans nos simulations sont l'erreur estimée globale relative en norme énergie $\gamma_\Omega = \xi_\Omega/\omega_\Omega$, la mesure de la zone encore marquée pour raffinement à la fin du processus AMR (évaluée par le ratio $\eta = \mu\left(\Omega_{\mathcal{M}}\right)/\mu\left(\Omega^h\right)$), le nombre d'éléments du maillage N_E et le nombre de ddls N. Ces grandeurs d'intérêt permettent de comparer quantitativement les deux combinaisons AMR.

Par ailleurs, la version de MFEM utilisée dans cette section pour ces études est la version 4.4.

2.3.1. Justification du paramètre de pénalisation k_N

Bien que la méthode de pénalisation soit une méthode classique et répandue pour le traitement numérique des problèmes de contact, voir par exemple [KIKUCHI et al. 1988 ; LAURSEN 2002 ; WRIGGERS 2006], le choix d'un paramètre de pénalisation optimal (à savoir k_N dans l'équation (2.5)) reste la principale limitation de cette méthode. En effet, comme déjà évoqué en section 1.3.3.1, l'interpénétration générée est directement liée au paramètre de pénalisation k_N, et ne devient nulle que dans un cas asymptotique, lorsque k_N tend vers l'infini. Cependant, un coefficient de pénalisation trop important conduit à un mauvais conditionnement de la matrice de rigidité de contact $\hat{\mathbf{K}}_C$, ce qui peut entraîner des difficultés de convergence des solveurs itératifs.

Pour limiter la valeur du coefficient de pénalisation tout en contrôlant l'interpénétration, il a été proposé dans [ZAVARISE 2015] d'ajouter un paramètre de décalage à la contribution de la pénalisation, qui est mis à jour de manière incrémentale avec les résidus obtenus lors de la résolution du système linéaire par un solveur itératif. Cette méthode, appelée "Shift Penalty Method" en anglais, permet de déplacer dynamiquement le point à partir duquel le potentiel de pénalisation est appliqué au problème vers l'intérieur du jeu initial, afin de réduire l'interpénétration due au contact sans augmenter le paramètre de pénalisation. Cette solution n'a pas été retenue dans notre travail, car elle implique de corriger le système

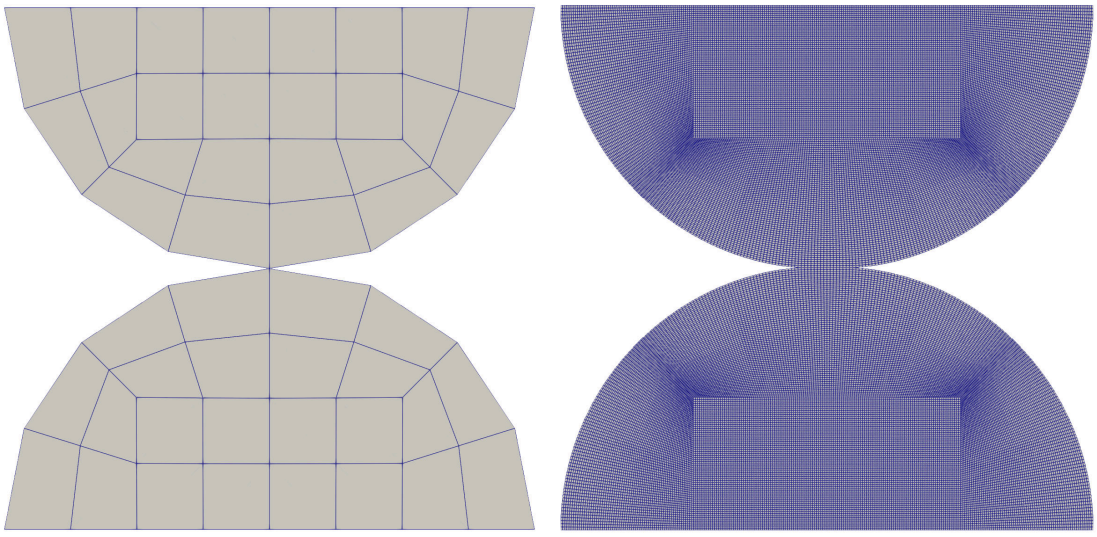
à résoudre au cours des itérations du solveur, ce qui n'est pas possible lors de l'utilisation de solveurs boîte noire tels que ceux de la bibliothèque hypre employée. D'autre part, une analyse de convergence de la méthode de pénalisation pour les problèmes de contact 2D et 3D a été réalisée dans [Chouly et al. 2013b]. Cette étude montre que les mêmes taux de convergence en maillage que ceux obtenus avec des méthodes qui résolvent directement le problème du point-selle découlant de la formulation du contact sont obtenus en choisissant k_N inversement proportionnel à h (la taille de maille).

Nous proposons de choisir le coefficient de pénalisation à travers deux études préliminaires. La première étape consiste à comparer les résultats obtenus via la méthode de pénalisation avec ceux obtenus avec la méthode des multiplicateurs de Lagrange. Pour cela, les forces de contact (voir $\mathbf{L}^C$ dans le Problème (2.1)) dérivées des deux méthodes sont comparées :

— $\mathbf{L}^C(\boldsymbol{\lambda}) = \hat{\mathbf{B}}^T\boldsymbol{\lambda}$, pour la méthode des multiplicateurs de Lagrange, où $\boldsymbol{\lambda}$ désigne la solution des multiplicateurs de Lagrange ;

— $\mathbf{L}^C(\mathbf{U}) = k_N\hat{\mathbf{B}}^T\left(\hat{\mathbf{D}} - \hat{\mathbf{B}}\mathbf{U}\right)$ pour la méthode de pénalisation.

Les forces de contact des multiplicateurs de Lagrange sont obtenues grâce à des simulations réalisées avec le logiciel éléments finis Cast3M [CEA 2023], développé par le CEA, qui est basé sur la méthode des multiplicateurs de Lagrange pour la mécanique du contact. Les forces de contact pénalisées sont obtenues avec notre solveur de contact pénalisé implémenté dans MFEM. Il convient de noter que pour toutes ces comparaisons, le maillage d'entrée utilisé dans Cast3M et MFEM, présenté dans la figure 2.5b, est identique. Ainsi, nous considérons un maillage 2D Q_1 (fonctions test bilinéaires) conforme finement discrétisé avec un pas de maille $h \approx R/2^8 = 2^{-7}$, soit 200 000 de nœuds de discrétisation au total (100 000 de nœuds dans chaque demi-disque). Ce maillage est directement obtenu via le mailleur Gmsh [Geuzaine et al. 2009] par le biais de 6 étapes de raffinement uniforme et isotrope d'un maillage initial grossier exposé dans la figure 2.5a. Le maillage raffiné est donc ici adapté à la géométrie.

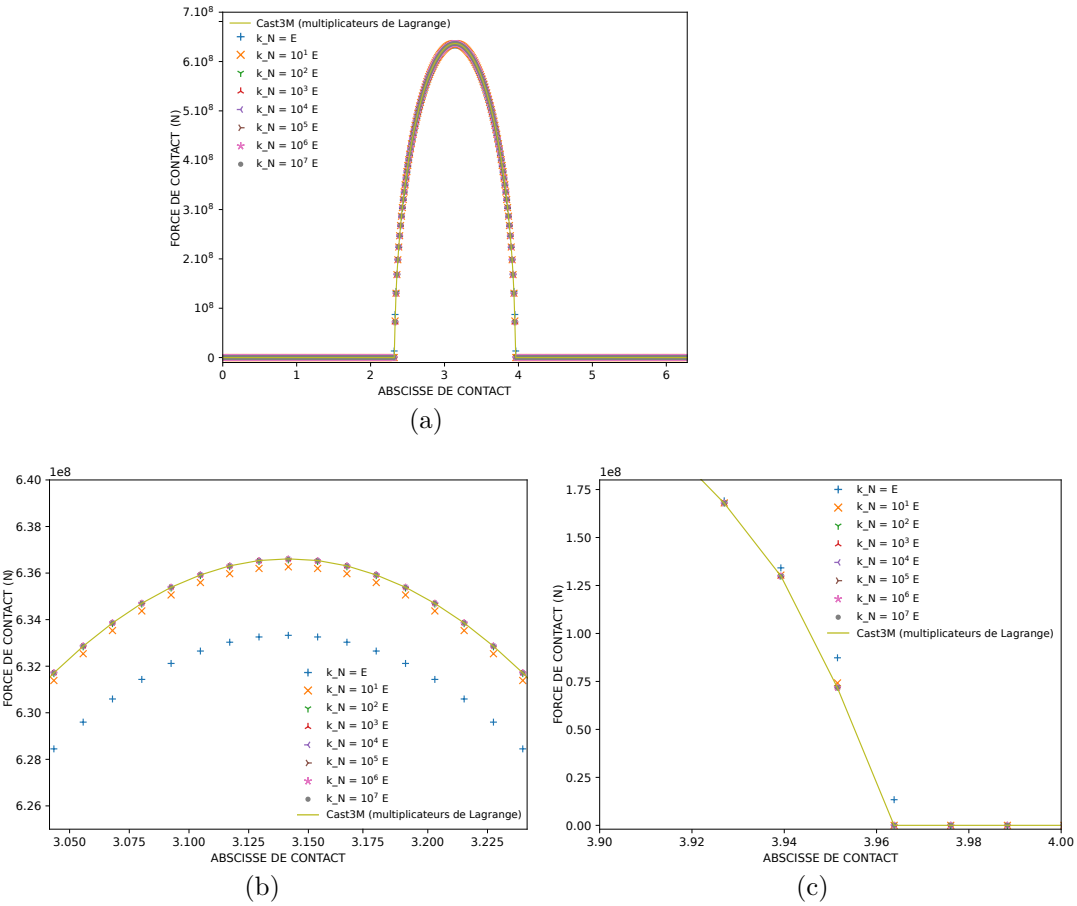


(a) (b)

Figure 2.5. – Problème du contact Hertzien 2D : maillages quadrangulaires initial dans (a) et raffiné dans (b).

La figure 2.6 montre les forces de contact obtenues avec notre algorithme MFEM de contact pour différentes valeurs du coefficient de pénalisation, allant de $k_N = E$ à

$k_N = 10^7 \cdot E$ par incrément d'un facteur de 10 (sans modifier le maillage d'entrée). Les forces de contact obtenues en utilisant la méthode des multiplicateurs de Lagrange, considérée comme la solution de référence, sont également reportées sur cette figure. Ici, α, défini dans l'équation (2.31) en page 70, est égal à $0,15$.



(a) (b) (c)

FIGURE 2.6. – Problème du contact Hertzien 2D : distribution de la pression de contact sur la zone de contact potentielle. Aperçu général dans (a) et zoom sur le centre de la zone de contact dans (b) et sur la zone de changement de statut de contact dans (c).

Comme attendu, les forces pénalisées convergent vers les forces des multiplicateurs de Lagrange lorsque le coefficient de pénalisation augmente. Les zooms effectués dans les figures 2.6b et 2.6c montrent clairement une mauvaise précision des résultats pour $k_N = \mathrm{E}$ et $k_N = 10 \cdot E$. Un coefficient de pénalisation d'au moins $10^2 \cdot E$ est donc nécessaire. En outre, le choix $k_N = \mathrm{E}/h = 128 \cdot E$ dérivé de [CHOULY et al. 2013b], est parfaitement conforme à cette conclusion et peut convenir dans le cas considéré.

Par ailleurs, la figure 2.7 représente l'influence du pas du maillage et du coefficient de pénalisation sur l'interpénétration des surfaces en contact pour deux maillages uniformément raffinés 6 (en bleu) et 7 (en orange) fois à partir du même maillage initial de pas h_{ini}. D'une part, cette figure montre que, pour une même simulation éléments finis, augmenter le coefficient de pénalisation tend logiquement à faire diminuer l'interpénétration des surfaces. D'autre part, cette figure permet d'observer que, pour un même coefficient de pénalisation, le ratio interpénétration sur pas de maillage reste constant. Cela montre qu'augmenter le coefficient de pénalisation n'est pas la seule

solution pour faire réduire l'interpénétration. En effet, diminuer le pas de maillage permet à coefficient de pénalisation fixé d'améliorer la précision de la solution. Ce résultat se justifie par le fait que plus il y a de nœuds sur les bords de contact, plus la valeur nodale de la force de contact sur chacun de ces nœuds est faible. Ainsi, faire le choix d'un coefficient de pénalisation constant durant les itérations AMR revient à réduire l'interpénétration avec le pas de maillage.

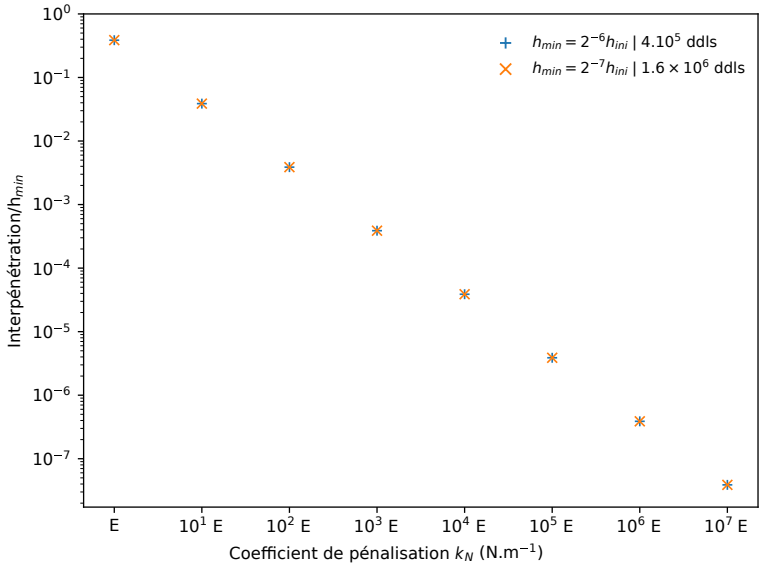
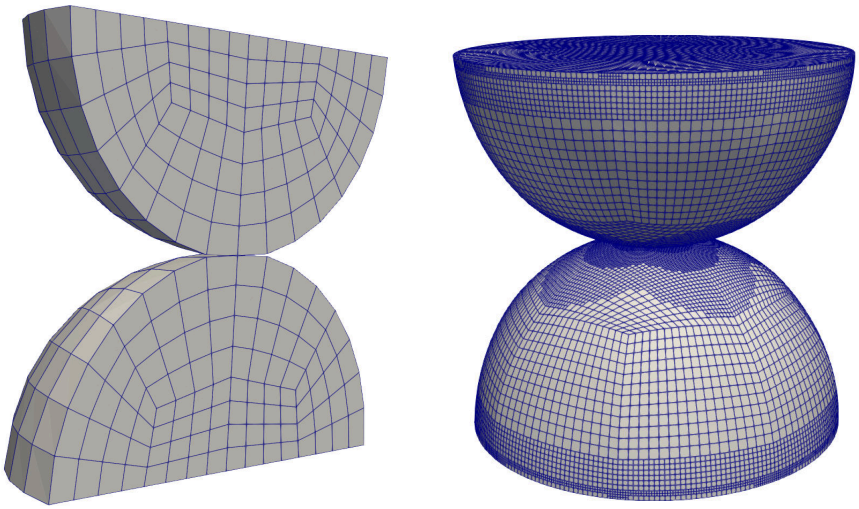


FIGURE 2.7. – Problème du contact Hertzien 2D : influence de k_N et du pas de maillage h_{min} sur l'interpénétration des surfaces.

La deuxième étape consiste à comparer les différents choix de k_N au cours d'une simulation AMR 3D par rapport à plusieurs quantités d'intérêt : le nombre d'éléments N_E et de ddls N, et les temps total et de résolution du système linéaire, T_{tot} et T_{solveur}, respectivement. Ici, la combinaison AMR 1 avec $e_\Omega = 2\%$ est considérée. Le maillage initial est construit dans le mailleur Gmsh à partir du maillage illustré dans la figure 2.2a (en vue de coupe) et auquel un niveau de raffinement uniforme a été appliqué (par le mailleur). Le maillage, présenté dans la figure 2.8a (en vue de coupe), ainsi obtenu contient 2484 nœuds et 2048 éléments finis et est adapté à la géométrie. Ici, pour l'AMR hiérarchique sur des maillages 3D, l'ordre de la transformation super-paramétrique est de 6. Le maillage localement raffiné final obtenu est représenté sur la figure 2.8b. Notons par ailleurs qu'ici, ainsi que pour la suite du manuscrit (sauf indication contraire), α vaut 0.015 (voir l'équation (2.31)).

Les simulations ont été effectuées sur quatre nœuds de calcul du supercalculateur Topaze du CCRT/CEA (processeurs AMD Milan) hébergeant chacun 128 cœurs par nœuds, donc 512 cœurs en tout. Le tableau 2.1 résume quelques résultats obtenus pour les différents choix de k_N. Le dernier choix correspond à une adaptation au cas d'une force nodale de contact avec AMR de la proposition de [CHOULY et al. 2013b] où k_N dépend de $h_{\min}$, la taille minimum de discrétisation à l'itération AMR courante. Cela consiste à augmenter k_N durant les itérations AMR, ce qui implique une solution encore plus précise (avec moins d'interpénétration) pour les maillages raffinés.



(a) (b)

FIGURE 2.8. – Problème du contact Hertzien 3D : vue de coupe du maillage initial en (a), maillage final pour la combinaison AMR 1 avec $e_\Omega = 2\%$ et $k_N = 10^2$ E en (b) - ordre de la transformation super-paramétrique de 6.

k_N (N.m^{-1})	N_E	N	T_{tot} (s)	T_{solveur} (s)
E	7.85×10^5	1.377×10^7	452	311
10 E	7.84×10^5	1.375×10^7	426	308
10^2 E	-	-	427	309
10^3 E	-	-	436	304
10^4 E	-	-	445	305
10^5 E	-	-	441	306
10^6 E	-	-	426	307
10^7 E	-	-	437	303
$\frac{\text{E}}{h_{\min}}$	-	-	431	306

TABLEAU 2.1. – Influence du coefficient de pénalisation k_N sur la simulation AMR avec une erreur cible globale de $e_\Omega = 2\%$ - Temps moyens sur 3 calculs - Le symbole "-" indique que la valeur est identique à la dernière entrée dans la colonne.

Il faut noter que toutes les simulations se sont terminées après 6 itérations AMR donc 5 étapes de raffinement, de sorte que le pas de maillage final h_{fin} est d'environ $2^{-5} h_{ini}$ avec h_{ini} le pas de maillage initial. Pour k_N supérieur à $10 \cdot E$, les résultats en termes d'AMR sont similaires en matière de nombre d'éléments et de nombre de ddls. De plus, les temps de calcul ne varient que très légèrement en fonction du coefficient de pénalisation choisi. En effet, des variations d'environ 6 % sont observées pour le temps de calcul total, et ne dépassent pas 3 % pour le temps de résolution du système linéaire. Ainsi, les résultats tendent à peu dépendre du coefficient de pénalisation dès lors que ce dernier est supérieur ou égal à $10 \cdot E$ en fin de calcul. De ce fait, en choisissant un coefficient de pénalisation variable tel que défini à la dernière ligne du tableau 2.1, il est observé que k_N dépasse $10 \cdot E$ à partir de la quatrième itération AMR (donc à la suite de trois étapes de

raffinement du maillage). Ainsi, à la fin du calcul, les nombres de nœuds et d'éléments du maillage sont les mêmes que pour un calcul avec un coefficient de pénalisation fixe et supérieur à $10 \cdot E$ et remplit les critères en matière de précision de la solution (même nombre de degrés de liberté que pour une simulation avec un coefficient de pénalisation fixe supérieur à $10 \cdot E$ notamment).

A partir de ces deux études, il a été décidé de fixer le coefficient de pénalisation à une valeur constante de $k_N = 10^4$ E pour obtenir un compromis satisfaisant entre interpénétration des surfaces et convergence des simulations, même pour des éléments de petites dimensions. En effet, le choix de 10^4 E ne semble pas plus coûteux en matière de temps de calcul que 10^2 E et réduit encore plus nettement l'interpénétration. Un coefficient de pénalisation variable aurait pu être un autre choix approprié, mais cette option est un peu plus intrusive.

2.3.2. Comparaison avec la solution analytique de Hertz

Pour évaluer la précision de la stratégie de solution proposée, il convient de comparer les résultats obtenus avec ceux issus de la théorie de Hertz (voir [Hertz 1882] pour plus de détails). Pour s'en tenir au cadre de la théorie de Hertz, les conditions suivantes doivent être satisfaites :

- les solides ont un comportement mécanique élastique, homogène et isotrope ;
- l'hypothèse des petites perturbations est appliquée ;
- les surfaces de contact sont petites comparativement aux solides entiers et les déformations à l'extérieur de la zone de contact sont négligeables par rapport à celles à l'intérieur ;
- pas de frottement.

Ainsi, pour cette analyse, le cas test 2D est considéré et les conditions aux limites de Dirichlet fixées ($u_D - \dfrac{\delta_0}{2} = 0,015\ R$) respectent l'hypothèse des petites perturbations. En outre, nous nous appuyons toujours sur la combinaison 1 pour le processus AMR.

La théorie de Hertz permet d'obtenir la solution analytique suivante pour la pression de contact :

$$p\,(r) = p_{\mathrm{O}}\sqrt{a^2 - r^2}/a\ ;\ r \leq a \tag{2.32}$$

où a est la distance caractéristique de contact, r la distance (positive) au centre O de la zone de contact et $p_{\mathrm{O}} = p\,(\mathrm{O})$ la pression de contact maximale au centre O. Pour calibrer les paramètres (a, p_{O}) de la solution de Hertz pour notre cas test, un maillage très raffiné a été utilisé pour construire une solution de référence, qui permet d'obtenir $a = 0,199$ et $p_{\mathrm{O}} = 11,5$ GPa. Ces valeurs sont ensuite utilisées pour définir la solution analytique de Hertz, qui est comparée dans la figure 2.9 avec les pressions de contact ($p = -\sigma_N$ sur Γ_C) obtenues à partir de différentes solutions AMR. Dans cette figure, l'origine de l'abscisse de contact est choisie comme étant le centre O de la zone de contact. Les résultats montrent clairement qu'en diminuant la tolérance spécifiée par l'utilisateur e_Ω pour l'application de l'AMR, la maillage est raffiné dans la zone de contact et la solution éléments finis tend vers la solution dérivée de la théorie de Hertz. En outre, cet exemple montre l'efficacité de la transformation super-paramétrique pour capturer la courbure de la frontière pendant les itérations d'AMR hiérarchique.

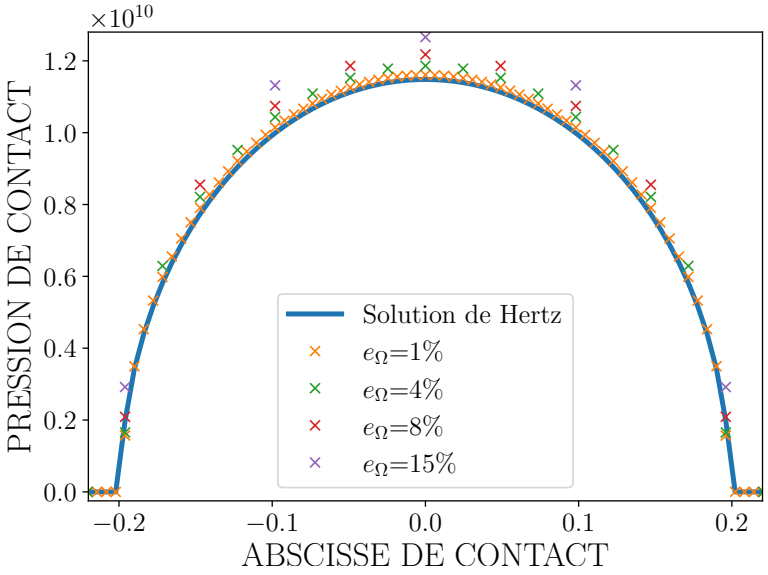


FIGURE 2.9. – Distribution de la pression de contact (en Pascal) sur la zone de contact (demi-disques de Hertz 2D) - appariement nœud-à-surface - combinaison AMR 1.

2.3.3. Sur les stratégies AMR

Afin d'évaluer l'intérêt d'effectuer de l'AMR au lieu d'un raffinement uniforme, la convergence en pas de maillage et en nombre de nœuds est réalisée pour le cas 2D. Pour cela, une solution de référence (sur toute la surface) est construite sur un maillage discrétisé très fin composé d'éléments finis linéaires avec un pas de maillage d'environ $R/2^{10}$, soit 3,2 millions de nœuds au total (1,6 million de nœuds pour chaque demi-disque). Pour cette étude de convergence en maillage, des éléments finis linéaires ou quadratiques isoparamétriques sont considérés. En effet, comme mentionné dans [DROUET et al. 2015] et les références citées dans ce dernier, la solution de contact appartient dans le meilleur des cas à l'espace de Hilbert $H^{5/2}$. Par conséquent, seules des méthodes éléments finis d'ordre 1 et 2 sont réellement d'intérêt. Pour évaluer l'erreur d'approximation sur le domaine de calcul, des maillages uniformément pré-raffinés de manière isotrope par le mailleur Gmsh sont utilisés. Un exemple de maillage initial est représenté sur la figure 2.5a.

Pour sa part, la méthode AMR est appliquée sur des éléments finis linéaires uniquement. En outre, comme les études de convergence de maillage impliquent des erreurs globales (sur l'ensemble du domaine), la combinaison AMR 1 est mise en œuvre. La figure 2.10 illustre la convergence en maillage des solutions uniformément raffinées ainsi que des solutions raffinées localement (avec de l'AMR).

D'une part, il est observé que les solutions uniformément raffinées respectent les taux de convergence prédits par la théorie des éléments finis en présence de contact entre deux solides, voir [WOHLMUTH et al. 2012 ; DROUET et al. 2015] par exemple. Pour la norme énergie, ces taux de convergence sont $\mathcal{O}\left(h^{\min(k,\tau-1)}\right)$ en fonction du pas de maillage et $\mathcal{O}\left(N^{-\min(k/m,(\tau-1)/m)}\right)$ vis-à-vis du nombre de degrés de liberté (directement proportionnel au nombre de nœuds du maillage). Dans ces formules, k désigne le degré d'interpolation des fonctions de forme, τ la régularité de la solution (ici $\tau = 5/2$) et m la dimension de l'espace. Par conséquent, pour des problèmes de contact résolus par des éléments finis d'ordre supérieur à 1, des taux de convergences sous-optimaux (ici en 3/2) sont obtenus,

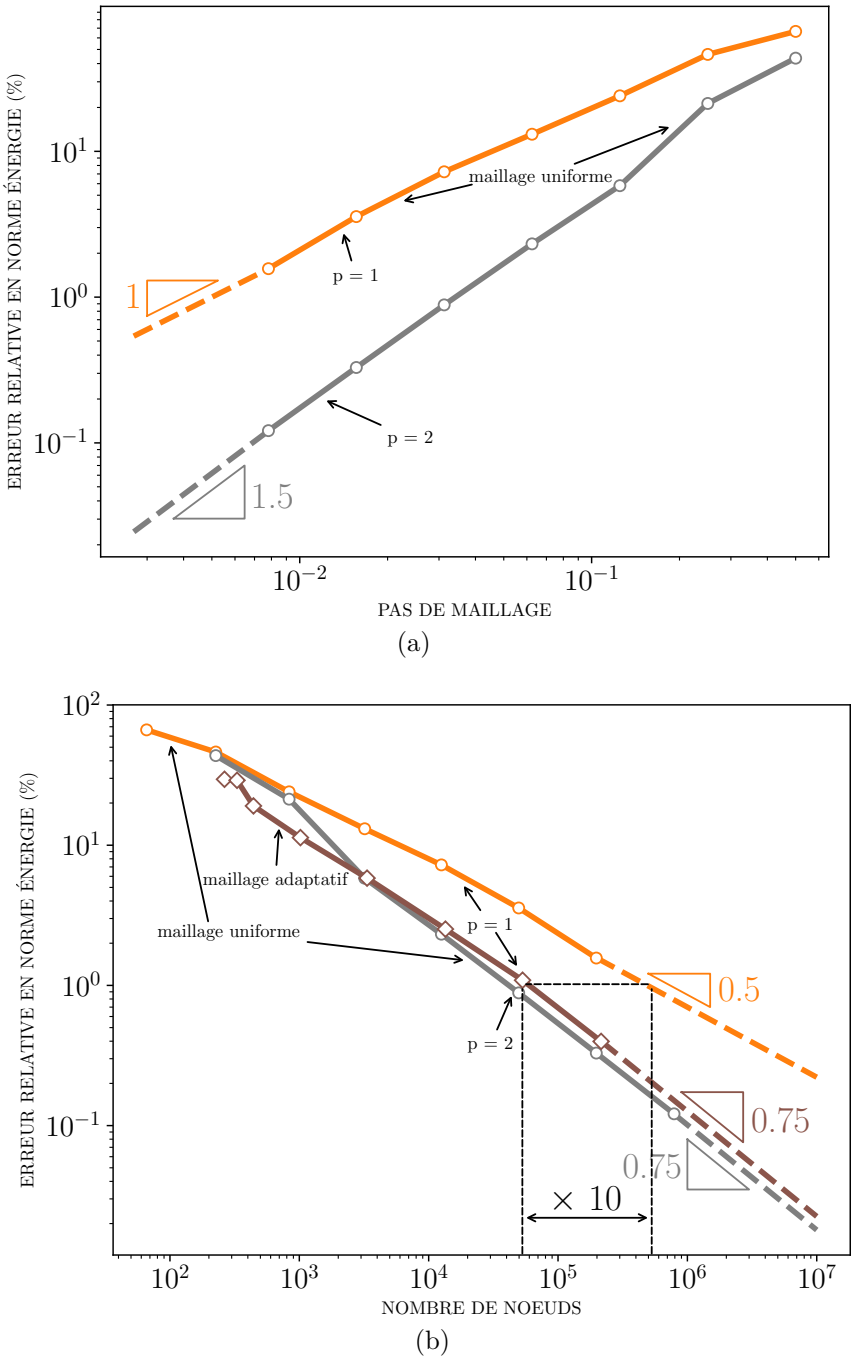


FIGURE 2.10. – Erreurs de discrétisation (demi-disques de Hertz) en fonction du pas, en (a), et du nombre de nœuds, en (b), du maillage pour des approches de raffinement uniforme et d'AMR.

voir la figure 2.10a.

D'autre part, les solutions d'ordre un obtenues avec AMR révèlent des meilleurs taux de convergence en ddls que les solutions obtenues avec un maillage uniformément raffiné, voir la figure 2.10b. Une super-convergence des solutions AMR d'ordre un est même observée, étant donné qu'elles sont tout aussi performantes que les solutions d'ordre deux uniformément raffinées. Comme attendu, pour un nombre de ddls donné et pour des éléments finis d'ordre un, les erreurs obtenues avec AMR

sont plus faibles que celles obtenues avec un raffinement uniforme. Par ailleurs, pour une erreur donnée, le nombre de ddls est au moins 10 fois plus petit pour la solution AMR. Ce gain substantiel justifie donc l'intérêt d'implémenter une stratégie AMR.

L'approche estimate-mark-refine considérée dans notre étude, combinant des critères quantitatifs de détection et d'arrêt, a récemment prouvé son efficacité dans un contexte de mécanique des structures [Koliesnikova et al. 2021] mais sans contact. Ainsi, l'enrichissement de la méthode h-adaptative actuellement implémentée dans MFEM avec cette stratégie est évaluée sur le problème du contact Hertzien ci-après. Pour ce cas d'étude, les demi-sphères 3D sont mises en contact et les deux combinaisons AMR introduites précédemment sont considérées. Le maillage initial pour des calculs avec 512 processus MPI est visible en figure 2.8a.

Remarque : Étant donné qu'un maillage 3D très fin et uniformément raffiné est très coûteux en termes de temps de calcul et d'empreinte mémoire, les résultats des erreurs estimées ne sont pas comparés ici à une solution de référence.

Le tableau 2.2 détaille les résultats obtenus avec la combinaison AMR 1 pour différentes valeurs de e_Ω allant de 2% à 1% tandis que la figure 2.11 montre le maillage raffiné obtenu pour $e_\Omega = 2\%$.

	$\gamma_\Omega(\%)$	$\eta(\%)$	N_E	Nombre d'itérations AMR
$e_\Omega = 2\%$	1.84	15.4	7.84×10^5	5
$e_\Omega = 1.7\%$	1.14	29.9	2.41×10^6	6
$e_\Omega = 1.5\%$	1.08	33.0	3.07×10^6	6
$e_\Omega = 1.2\%$	0.976	21.2	4.59×10^6	7
$e_\Omega = 1\%$	0.925	16.1	6.27×10^6	7

Tableau 2.2. – Contact Hertzien 3D - combinaison AMR 1 - résultats en erreurs et nombre d'itérations AMR

Similairement, le tableau 2.3 fournit les résultats obtenus avec la combinaison AMR 2 pour $e_{\Omega,\text{LOC}}$ allant de 7% à 2% et $\delta = 0.1\%$, tandis que la figure 2.12 affiche le maillage obtenu pour $e_{\Omega,\text{LOC}} = 7\%$ et $\delta = 0.1\%$. Ici, $e_{\Omega,\text{LOC}}$ est choisi globalement pour être plus élevé que e_Ω parce que la combinaison AMR 2 tend à plus raffiner que la combinaison AMR 1, voir les tableaux 2.2 et 2.3 pour la valeur d'erreur de 2% par exemple.

Tout d'abord, nous observons que, pour la combinaison AMR 1 (voir la figure 2.11), le raffinement de maillage est répandu dans la zone de contact, tandis que, pour la combinaison AMR 2, la figure 2.12 montre que les mailles les plus fines (correspondant au niveau de raffinement le plus élevé) sont localisées dans la zone où le statut de contact (contact/décollement) change. Les deux tableaux montrent que, quelle que soit la combinaison sélectionnée, le seuil d'erreur prescrit par l'utilisateur e_Ω (respectivement $e_{\Omega,\text{LOC}}$) est respecté, c'est-à-dire $\gamma_\Omega < e_\Omega$ (respectivement $\gamma_\Omega < e_{\Omega,\text{LOC}}$), en particulier pour la combinaison AMR 2 où cette condition n'est pas imposée. De plus, pour une erreur relative globale équivalente γ_Ω, le maillage obtenu contient moins d'éléments lorsque la combinaison AMR 1 est utilisée, mais l'utilisation de cette combinaison ne contrôle pas

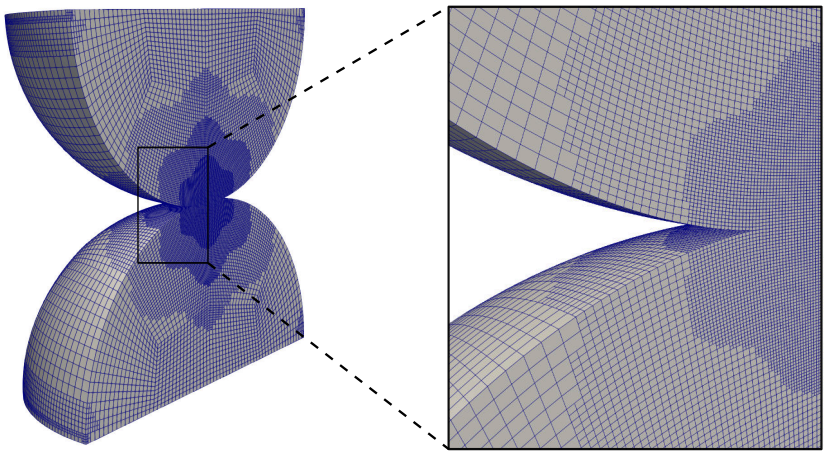


FIGURE 2.11. – Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 2\%$). Vue de coupe dans (à gauche) et zoom sur la zone du centre du contact (à droite) - ordre de la transformation super-paramétrique de 6.

	$\gamma_\Omega(\%)$	$\eta(\%)$	N_E	Nombre d'itérations AMR
$e_{\Omega,\text{LOC}} = 7\%$, $\delta = 0.1\%$	5.11	0.0537	2.88×10^5	5
$e_{\Omega,\text{LOC}} = 5\%$, $\delta = 0.1\%$	3.60	0.0182	1.05×10^6	6
$e_{\Omega,\text{LOC}} = 4\%$, $\delta = 0.1\%$	2.92	0.0330	1.73×10^6	6
$e_{\Omega,\text{LOC}} = 3\%$, $\delta = 0.1\%$	2.21	0.0108	4.64×10^6	7
$e_{\Omega,\text{LOC}} = 2\%$, $\delta = 0.1\%$	1.47	0.0363	1.34×10^7	7

TABLEAU 2.3. – Contact Hertzien 3D - combinaison AMR 2 - résultats en erreurs et nombre d'itérations AMR

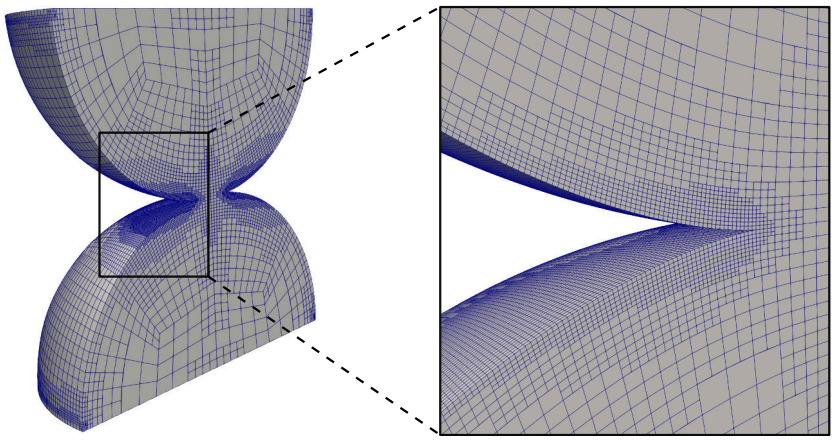


FIGURE 2.12. – Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\text{LOC}} = 7\%$, $\delta = 0,1\%$). Vue de coupe dans (à gauche) et zoom sur la zone du centre du contact (à droite) - ordre de la transformation super-paramétrique de 6.

efficacement la zone où l'erreur locale prescrite n'est pas respectée, voir η à la ligne 1 du tableau 2.2 par rapport à la ligne 5 du tableau 2.3. En effet, contrôler η, comme la combinaison AMR 2 le permet, contribue à limiter la proportion du maillage qui ne respecte

pas le seuil d'erreur local. Ainsi, choisir la combinaison AMR 2 pour l'application de l'AMR nous permet de caractériser plus efficacement les régions critiques, voir la figure 2.14 par comparaison avec la figure 2.13, où le nombre d'itérations AMR est le même.

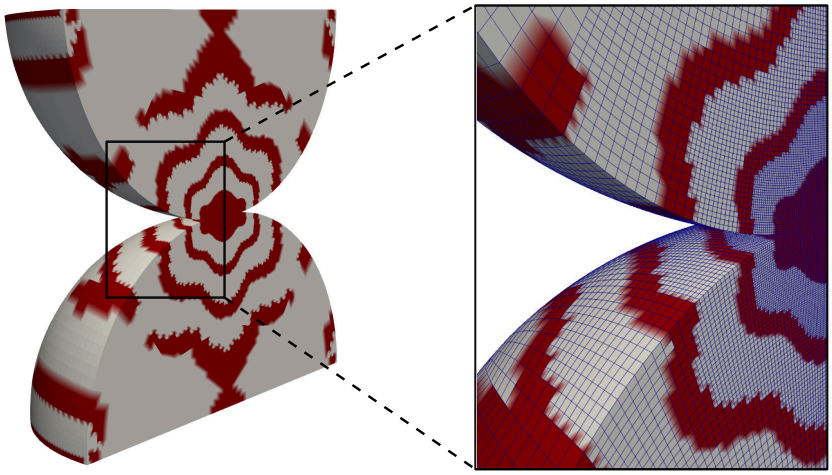


FIGURE 2.13. – Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 2\%$). Zones encore marquées pour raffinement à la fin du processus AMR en rouge. Vue de coupe (à gauche) et zoom sur la zone du centre du contact (à droite) - ordre de la transformation super-paramétrique de 6.

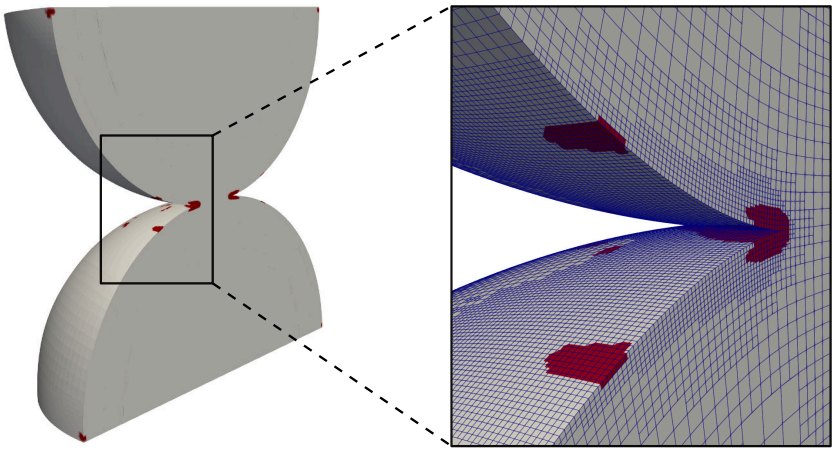


FIGURE 2.14. – Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\mathrm{LOC}} = 7\%$, $\delta = 0,1\%$). Zones encore marquées pour raffinement à la fin du processus AMR en rouge. Vue de coupe (à gauche) et zoom sur la zone du centre du contact (à droite) - ordre de la transformation super-paramétrique de 6.

Ainsi, le choix de la combinaison de critères AMR pertinente dépend de ce qui est d'intérêt à contrôler. Si l'objectif principal est de contrôler l'erreur globalement seulement, alors choisir la combinaison 1 permet d'obtenir un maillage acceptable avec un nombre limité de ddls. Si l'objectif du raffinement est de contrôler localement l'erreur, alors choisir la combinaison 2 est requise.

Par ailleurs, les figures 2.11 à 2.14 permettent d'observer que le caractère symétrique du problème (géométries identiques, mêmes matériaux) engendre un raffinement parfaitement identique des deux solides où les nœuds sont exactement en vis-à-vis.

Pour finir, le dernier cas de vérification concerne le cas du contact entre deux solides avec des propriétés mécaniques différentes. Des simulations ont été réalisées avec deux solides présentant un contraste en module d'Young de 10^4, le coefficient de Poisson restant le même. Le solide inférieur est alors plus rigide que le supérieur, qui conserve la caractéristique matériau des cas test précédents ($E_1 = 210$ GPa). Les mêmes conditions aux limites de Dirichlet que celles introduites au début de la section ont été appliquées et la combinaison AMR 1 a été choisie. Un exemple de résultats obtenus est reporté dans la figure 2.15. Plusieurs observations sont à noter. Tout d'abord, le fait d'appliquer distinctement le processus AMR sur chaque solide en contact, permet de constater que le solide le plus déformable est le plus raffiné. Par ailleurs, le maillage raffiné obtenu montre l'effet du critère d'élargissement des éléments marqués pour raffinement (voir l'équation (2.16)), étant donné que tous les éléments appairés (en contact potentiel) sont raffinés de la même manière.

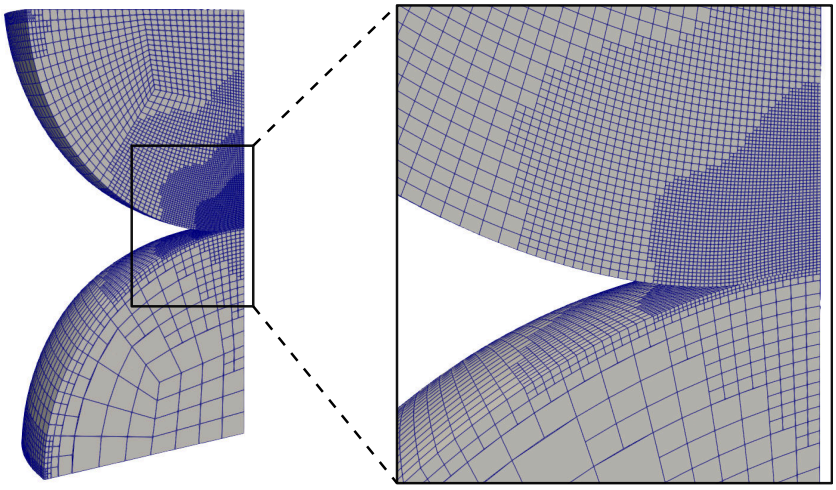


FIGURE 2.15. – Contact Hertzien 3D avec deux solides de propriétés mécaniques différentes - combinaison AMR 1 ($e_\Omega = 2\%$). Vue de coupe (à gauche) et zoom sur la zone du centre du contact (à droite) - ordre de la transformation super-paramétrique de 6.

2.4. Conclusion

Dans ce chapitre, une stratégie AMR h-adaptative non-conforme dédiée à des problèmes en mécanique du contact en élasto-statique a été introduite. Elle a été développée dans la bibliothèque éléments finis open-source MFEM. Elle est basée sur l'implémentation d'un appariement de type nœud-à-nœud combiné avec la technique de la pénalisation pour la résolution du problème de contact. L'algorithme MFEM pour le raffinement de type h non-conforme a été enrichi par notre propre stratégie estimate-mark-refine. Cette stratégie est en particulier conçue pour être efficace pour des problèmes de contact avec des frontières courbes grâce à l'emploi d'éléments finis super-paramétriques. Ainsi, des solutions éléments finis d'ordre un très précises peuvent être obtenues sur des maillages raffinés localement.

Des exemples académiques représentatifs 2D et 3D ont permis de valider l'approche proposée. Tout d'abord, nos solutions ont été validées vis-à-vis de la solution de Hertz, mais aussi en comparant les forces de contact issues de la pénalisation avec leurs équivalents en

multiplicateurs de Lagrange. Par ailleurs, le coefficient de pénalisation a été dimensionné via une étude paramétrique et le choix de $k_N = 10^4$ E s'est avéré pertinent dans le but d'obtenir un compromis satisfaisant entre interpénétration des surfaces et convergence des calculs, même pour des maillages fins.

Deux combinaisons de critères AMR ont été étudiées. Tout d'abord, en utilisant la combinaison AMR 1, combinant le critère de détection défini par Zienkiewicz et Zhu et un critère d'arrêt global basé sur l'erreur globale absolue estimée, l'intérêt de raffiner localement le maillage a été justifié par l'étude de la convergence en norme d'erreur énergie de déformation sur l'ensemble du domaine de calcul. En outre, la comparaison de cette première combinaison AMR avec la seconde, combinant le critère de détection LOC basé sur l'erreur élémentaire locale et un critère d'arrêt local, a permis de mettre en exergue les spécificités de chacune des combinaisons considérées. En effet, les zones raffinés se concentrent soit au niveau du centre du contact, soit au niveau de la zone limite contact/décollement, et la combinaison est donc à choisir selon l'objectif en erreur qui est visé.

En outre, la stratégie proposée a été confrontée au cas où les matériaux des solides mis en contact sont différents. Il a été observé que le problème n'est plus symétrique puisque les solides ont été raffinés différemment selon leur rigidité. De plus, même si un léger glissement entre les solides a pu être noté, ce glissement s'est révélé très petit par rapport au plus petit pas du maillage.

Par ailleurs, la parallélisation de la stratégie numérique AMR-contact proposée dans le but de pouvoir traiter des cas 3D plus complexes, plus représentatifs de cas réels, est l'objet du chapitre 3. Cette extension à un cadre de calcul parallèle constitue un apport réel par rapport aux solutions proposées jusqu'alors qui ne combinent AMR et résolution de problèmes de contact déformable-déformable que sur des calculateurs séquentiels.

Chapitre 3

Partitionnement, équilibrage de charge et performances parallèles

Ce chapitre se concentre sur l'adaptation de la stratégie de résolution du contact décrite dans le chapitre 2 (avec un appariement de type nœud-à-nœud) dans un cadre parallèle. Cela passe par la mise en place de méthodes de partitionnement de maillage permettant un équilibrage de charge efficace du domaine d'étude. En considérant les possibilités offertes par l'environnement logiciel MFEM, différentes stratégies de partitionnement sont détaillées. Elles sont adaptées à la gestion du contact et à l'application de l'AMR dans un cadre de calcul haute performance. Des analyses de performances parallèles sont effectuées sur le cas du contact Hertzien et sur un cas plus volumineux d'indentation afin d'évaluer les caractéristiques et la pertinence des stratégies mises en œuvre. La scalabilité de notre algorithme employant ces méthodes de partitionnement est validée jusqu'à plusieurs milliers de cœurs sur des grands domaines de calcul.

Sommaire

3.1. Introduction . . . 86
3.2. Partitionnement d'un maillage non-conforme dans l'environnement logiciel MFEM . . . 86
3.3. Partitionnements séparant éléments de contact et de non-contact . . . 87
3.3.1. Partitionnement "equidistributed" . . . 90
3.3.2. Partitionnement "pie chart" . . . 93
3.4. Partitionnement dérivé de la méthode Space Filling Curve (SFC) "SFC-contact" . . . 97
3.5. Analyse préliminaire sur le cas du contact Hertzien . . . 99
3.5.1. Présentation des simulations étudiées . . . 99
3.5.2. Étude de l'empreinte mémoire pour le partitionnement "equidistributed" . . . 101
3.5.3. Strong scaling pour le partitionnement "equidistributed" . . . 103
3.5.4. Étude du pré-conditionnement pour le partitionnement "equidistributed" . . . 105
3.5.5. Apport des méthodes de partitionnement "pie chart" et "SFC-contact" 106
3.6. Étude des stratégies de partitionnement sur le contact Hertzien . . . 110
3.6.1. Strong scaling pour les trois stratégies étudiées . . . 111
3.6.2. Comparaison de pré-conditionneurs pour les méthodes de partitionnement "pie chart" et "SFC-contact" . . . 112
3.6.3. Influence du maillage d'entrée choisi pour les méthodes de partitionnement "pie chart" et "SFC-contact" . . . 116
3.7. Extension à des problèmes de grande taille . . . 120
3.7.1. Contact Hertzien . . . 120
3.7.2. Cas d'indentation . . . 122
3.8. Conclusion . . . 128

3.1. Introduction

Obtenir une prédiction précise des solutions des problèmes en mécanique du contact avec adaptation locale du maillage nécessite de résoudre généralement un problème comportant un système linéaire creux de grande taille. Il est donc intéressant de répartir les tâches entre différents processus afin d'accélérer les simulations. La nécessité de paralléliser les codes devenant de plus en plus importante, des algorithmes ont été mis en œuvre dans des bibliothèques d'éléments finis telles que MFEM pour traiter de grands maillages non-conformes en parallèle [Anderson et al. 2020]. Cependant, l'adaptation de ces algorithmes aux problèmes en mécanique du contact reste une tâche à fort enjeu.

Comme vu en section 1.9.4, l'environnement logiciel MFEM a été choisi pour réaliser les développements informatiques relatifs à ces travaux de thèse. Il s'appuie sur un parallélisme basé sur la bibliothèque d'échanges de messages MPI. Pour débuter ce chapitre, la manière dont MFEM partitionne un maillage non-conforme est introduite dans la section 3.2. Dans les sections 3.3 et 3.4, nous décrivons et comparons des solutions parallèles qui s'attachent à adapter ces algorithmes de partitionnement aux problèmes de contact. Leurs performances parallèles sont ensuite analysées dans les sections 3.5, 3.6 et 3.7 sur les cas du contact Hertzien et de l'indentation. Le paradigme de traitement parallèle à mémoire distribuée le plus courant est ici considéré, à savoir le standard pour le passage de messages MPI (Message Passing Interface en anglais).

Comme évoqué en section 1.3.2.4, la détection du contact dans un cadre parallèle peut être une tâche très coûteuse en temps de calcul et sa parallélisation efficace reste un enjeu. Dans nos travaux, nous nous limitons à des problèmes en statique avec des détections du contact simples, sans chercher en première approche à élaborer une méthodologie de détection générique.

3.2. Partitionnement d'un maillage non-conforme dans l'environnement logiciel MFEM

Nous présentons brièvement les notions principales caractérisant la parallélisation du maillage réalisée dans MFEM. Cela est nécessaire pour comprendre les sections à venir dans ce chapitre.

Dans MFEM, un maillage parallèle est représenté par la classe d'objet `ParMesh` qui est dérivée de la classe `Mesh`. Les structures de données et fonctionnalités héritées de la classe `Mesh` caractérisent la partie locale (associée à la tâche MPI correspondante) du maillage considéré.

Une combinaison efficace de la mécanique du contact et des stratégies de raffinement h-adaptatives non-conformes dans un cadre de traitement parallèle à mémoire distribuée repose avant tout sur un partitionnement de maillage pertinent. Dans la bibliothèque MFEM, le partitionnement est basé sur les éléments, c'est-à-dire que chaque élément est assigné à un des processus MPI. Ainsi, différents processeurs ne peuvent pas être responsable du même élément. L'ensemble des éléments (finis) du maillage est donc divisé en R régions disjointes, où R est le nombre de tâches MPI. Ces régions peuvent cependant partager des entités de plus faible dimensions situées au niveau des frontières entre ces régions : des faces (en 3D), des arêtes (en 2D et 3D) et des sommets (en 1D, 2D et 3D). Les données liées à ces sommets, arêtes et faces à la frontière de chaque région de tâche sont dupliquées en ajoutant des éléments fantômes (copie d'éléments qui appartiennent

à d'autres processus MPI), de sorte que chaque région puisse être considérée comme un sous-maillage isolé qui peut être traité en parallèle. L'opérateur de prolongation $\mathbf{P}$ (défini pour les nœuds non-conformes dans le cas AMR, voir l'équation (2.21)) est étendu au traitement des degrés de liberté (ddls) fantômes de sorte que la solution du système linéaire, obtenue après l'étape de résolution dans le module SOLVE, concerne uniquement les degrés de liberté conformes et non fantômes. Le lecteur peut se référer à [Červený et al. 2019] pour plus de détails.

Par défaut, dans la bibliothèque MFEM, lorsque le maillage est raffiné de manière hiérarchique, les éléments enfants sont pré-assignés à la tâche MPI de leur élément parent. En conséquence, lors d'un raffinement h-adaptatif non conforme, un déséquilibre entre les processus MPI peut se produire, rendant utile un rééquilibrage des éléments. La fonction `Rebalance` peut être appelée pour produire un nouveau partitionnement du maillage. Utilisée ainsi (sans aucun argument), elle correspond à une fonction boîte noire qui construit une solution de partitionnement basée sur la méthode géométrique classique des courbes de remplissage d'espace (Space-Filling Curve, SFC) via les courbes de Hilbert (voir la section 1.5.3.1). Elles sont illustrées par la figure 3.1.

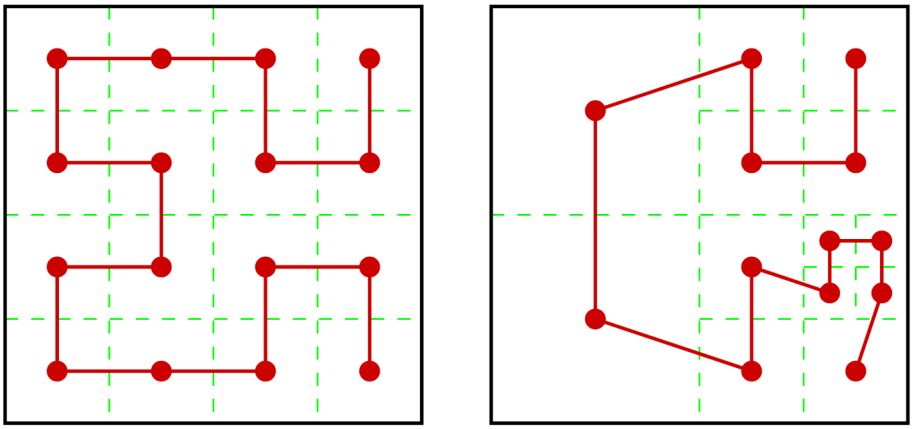


Figure 3.1. – Exemple de courbes de Hilbert SFC en dimension 2 avec et sans AMR (source : [Campbell et al. 2003]).

Dans la bibliothèque MFEM, les fonctions intégrées pour créer des partitionnements de maillage ne sont actuellement pas adaptées aux problèmes issus de la mécanique du contact. En effet, si elles sont appliquées telles quelles, les nœuds de contact appairés n'ont aucune raison particulière d'être sur les mêmes processus. Ainsi, la construction de la matrice d'appariement de contact $\mathbf{B}$ (voir la section 2.1.1) nécessiterait, au minimum, des communications MPI supplémentaires. Dans ces travaux, nous présentons des stratégies spécifiques qui soit fournissent de nouvelles fonctions de partitionnement, soit adaptent des fonctions de partitionnement de maillage proposées par MFEM aux problèmes de contact. En revanche, les autres fonctionnalités principales de MFEM (assemblage, formation du système linéaire, résolution, etc.) sont toujours utilisées telles quelles.

3.3. Partitionnements séparant éléments de contact et de non-contact

Les algorithmes dédiés proposés dans cette section, et plus globalement dans ce chapitre, garantissent un partitionnement du maillage tel que les nœuds de contact appairés se

trouvent sur les mêmes processus durant les itérations de raffinement. Concrètement, cela implique que la construction de la matrice locale d'appariement de contact $\mathbf{B}$ ne requiert pas de communications avec les autres processus. Ainsi, les matrices de rigidité de contact et les vecteurs de contact sont assemblés localement sur chaque tâche MPI, comme si chaque région était comme un maillage isolé, traitée par une partie séquentielle du code. Cela réduit considérablement la quantité de communications inter-processus pendant les itérations de raffinement, tout en améliorant l'accessibilité du code et la scalabilité parallèle. En effet, si les nœuds de contact appairés ne se trouvent pas sur les mêmes processus, plusieurs communications supplémentaires seraient, *a minima*, nécessaires. Premièrement, des communications seraient requises pour appairer les nœuds de contact (possiblement en passant au préalable par une recherche des processus voisins). D'autres communications seraient envisagées afin de calculer l'interpénétration entre les nœuds de contact appairés. Enfin, la construction de la matrice d'appariement active $\hat{\mathbf{B}}$ et le vecteur des jeux "actifs" $\hat{\mathbf{D}}$ exigeraient également des communications supplémentaires afin de bien prendre en compte l'ensemble des contributions des nœuds. Le calcul de l'interpénétration mais aussi de la matrice $\hat{\mathbf{B}}$ et du vecteur $\hat{\mathbf{D}}$ étant effectués à l'intérieur de la boucle interne de contact, le surcoût dû à ces communications devrait être significatif.

La spécificité des algorithmes de partitionnement présentés dans cette section réside dans le fait que les éléments s'appuyant sur des frontières de contact potentielles ne sont pas traités par les mêmes processus MPI que les autres éléments dans le but de simplifier l'écriture du code. Dans la suite, nous appellerons "éléments de contact" les éléments finis dont au moins un côté est situé sur un bord de contact potentiel, et "éléments de non-contact" les éléments finis n'ayant aucun bord sur la zone de contact potentiel. L'ensemble des éléments de contact et de non-contact représente une partition du maillage.

L'algorithme de partitionnement implémenté choisi est responsable de l'attribution de chaque élément à la tâche MPI à laquelle il appartiendra une fois le rééquilibrage effectué. Ici, l'algorithme proposé fournit une partition de maillage à la fonction MFEM `Rebalance`, cette dernière redistribuant ensuite les éléments entre les tâches MPI en conséquence. L'algorithme est conçu pour garantir qu'une fois le maillage redistribué, la charge de travail soit autant égale que possible entre les processus traitant les éléments de contact d'une part, et les autres tâches d'autre part.

Après l'application de l'algorithme AMR (et avant le rééquilibrage), les régions MPI traitant initialement les éléments de contact peuvent contenir à la fois des éléments de contact et des éléments non-contact. La formule

$$R_C = \left\lceil c \frac{N_{E,C}}{N_{E,tot}} R \right\rceil \tag{3.1}$$

définit le nombre de tâches MPI traitant les éléments de contact. Dans cette équation, $N_{E,C}$ représente le nombre (pair) d'éléments de contact, $N_{E,tot}$ le nombre total d'éléments du maillage (tous deux après l'application du raffinement), R le nombre total de processus MPI, $c > 0$ un coefficient donné *a priori*, et $\lceil x \rceil$ la fonction partie entière supérieure du scalaire x (plus petit entier supérieur ou égal à x). Dans cette formule, choisir $c > 1$ conduira à avoir moins d'éléments en moyenne sur les tâches MPI traitant les éléments de contact que sur les autres, et inversement pour $c < 1$. De plus, R_C est réévalué à chaque itération de raffinement du maillage, car le ratio $N_{E,C}/N_{E,tot}$ peut avoir changé.

Le résultat obtenu avec l'équation (3.1) doit être corrigé dans les cas suivants :

$$\begin{cases} R_C = \dfrac{N_{E,C}}{2} & \text{si } R_C > \dfrac{N_{E,C}}{2} \\ R_C = R - 1 & \text{si } R_C \geq R \text{ et } N_{E,tot} > N_{E,C}. \end{cases} \tag{3.2}$$

Le premier critère garantit qu'il y ait au moins une paire d'éléments de contact par tâche MPI traitant les éléments de contact. Le deuxième critère assure qu'au moins un processus MPI soit réservé pour traiter les éléments de non-contact. Évidemment, s'il n'y a que des éléments de contact, le deuxième critère ne s'applique pas.

Ensuite, pour chaque région r contenant $N_{E,\mathrm{r}}$ éléments, les $N_{E,C,\mathrm{r}}$ éléments de contact (avec $N_{E,C,\mathrm{r}} = 0$ dans les régions MPI non-contact) doivent être assignés aux R_C premiers processus MPI traitant les éléments de contact, tandis que les $N_{E,\mathrm{r}} - N_{E,C,\mathrm{r}}$ autres éléments doivent être assignés aux $(R - R_C)$ tâches MPI suivantes. Une illustration descriptive est présentée en figure 3.2.

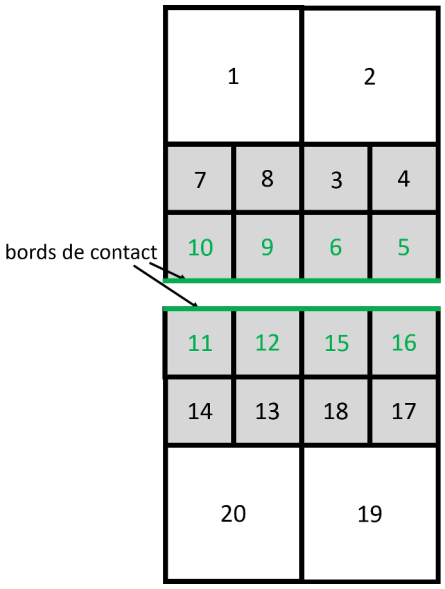


FIGURE 3.2. – Illustration du principe du traitement séparé des éléments de contact et de non-contact.

Dans cet exemple, deux solides carrés peuvent rentrer en contact et deux processus MPI sont considérés (le processus 0 en gris, le processus 1 en blanc). Suite à une étape de raffinement, le processus 0 possède à la fois des éléments de contact (éléments dont la numérotation est en vert) et des éléments de non-contact, tandis que le processus 1, qui ne possédait pas d'éléments de contact, ne possède toujours pas d'éléments de contact après le raffinement. Ainsi, après application de l'équation (3.1) et des critères (3.2), le processus 0 conserve les éléments de contact et idem pour le processus 1 qui conserve les éléments de non-contact. Ainsi, les éléments 5, 6, 9, 10, 11, 12, 15 et 16 doivent être assignés au processus 0 tandis que les autres éléments doivent être assignés au processus 1.

Dans cette section, deux approches de partitionnement de maillage basées sur le principe qui vient d'être décrit sont introduites. La première présente la caractéristique de ne pas faire intervenir de communications entre les processus au cours du partitionnement tandis que la seconde fait usage de communications entre les processus pour proposer un partitionnement où les éléments voisins sont assignés aux mêmes processus.

3.3.1. Partitionnement "equidistributed"

Un premier algorithme de partitionnement est présenté ici. Il est basé sur l'équidistribution des éléments sur les processus et est appelé "equidistributed" dans ce manuscrit. Concrètement, dans chaque région MPI r, les éléments de contact du premier solide sont distribués, le plus équitablement possible, en fonction de leur numéro local sur les R_C premiers processus MPI.

Ces éléments sont attribués localement aux processus MPI via une stratégie proche de la méthode SFC (voir la section 1.5.3.1) appliquée uniquement aux éléments de contact. Cependant, elle diffère de cette méthode sur deux points notables. Premièrement, contrairement à la méthode SFC via la courbe de Hilbert, qui effectue une renumérotation globale des éléments [1] pour les redistribuer ensuite (induisant ainsi des communications entre les tâches MPI), seule la numérotation locale des éléments est prise en compte ici pour le partitionnement. Ainsi, aucune communication entre les processus MPI n'est nécessaire. En effet, avec cet algorithme, un processus qui a $N_{E,C}$ éléments de contact à répartir sur R_C tâches MPI va distribuer les $\lceil N_{E,C}/R_C \rceil$ premiers éléments de la liste des $N_{E,C}$ éléments sur le premier processus qui traite les éléments de contact et ainsi de suite pour les autres éléments sur les autres processus concernés.

Deuxièmement, comme pour une région r, il pourrait y avoir moins d'éléments de contact (localement) que de tâches MPI traitant les éléments de contact ($\mathrm{N}_{E,C,\mathrm{r}}/2 < R_C$), le numéro du processus de départ pour le partitionnement ne commence pas à 0 mais est défini avec un décalage afin d'éviter les déséquilibres potentiels. Le numéro du processus de fin est alors lui aussi modifié.

Ensuite, les éléments de contact dans le second solide sont affectés au même processus MPI que leur élément appairé dans le premier solide.

Puis, les éléments restants (non-contact) doivent être distribués de manière équilibrée sur le sous-ensemble restant des tâches MPI. Pour cela, une routine basée sur une équidistribution des éléments est employée. Elle est similaire à celle appliquée pour les éléments de contact du premier solide.

Enfin, une fois ce nouveau partitionnement produit, la fonction `RedistributeElements` de MFEM, appelée au sein de la fonction `Rebalance`, s'occupe de transférer les éléments entre les tâches MPI tout en respectant le partitionnement spécifié. De plus, dans la stratégie de solution AMR-contact proposée, l'étape de rééquilibrage est effectuée à la fin de chaque itération AMR, car il s'agit d'une étape peu coûteuse lorsqu'une méthode de partitionnement géométrique est appliquée pour redistribuer les éléments, voir [Červený et al. 2019]. Cette étape pourrait également être réalisée à une fréquence moins importante mais son influence sur la scalabilité parallèle de la stratégie n'est pas évaluée dans ces travaux.

Un exemple de partitionnement produit par notre stratégie est illustré dans la figure 3.3. Dans cet exemple, une étape de raffinement local de maillage vient d'être effectuée et les éléments (représentés ici par des cases) doivent être répartis entre 8 tâches MPI, qui ont initialement un nombre variable d'éléments. À l'itération AMR actuelle, il y a 3 tâches MPI traitant les éléments de contact, qui, après l'application de l'étape de raffinement, contiennent à la fois des éléments de contact et des éléments de non-contact. Dans la figure 3.3, les éléments de contact sont regroupés par paires d'éléments de contact et seul

1. Le raffinement du maillage implique la création de nouveaux éléments et donc il est nécessaire de proposer une nouvelle numérotation des éléments du maillage nouvellement constitués, à la fois localement et globalement.

le numéro local de l'élément de contact du premier solide est pris en compte. De cette manière, un élément de contact du second solide n'est plus identifié par son numéro local, mais par le numéro local de l'élément de contact du premier solide auquel il est appairé. Chaque regroupement d'éléments caractérise un "super-élément" et est illustré en vert dans la figure 3.3. Les éléments de non-contact sont représentés en blanc.

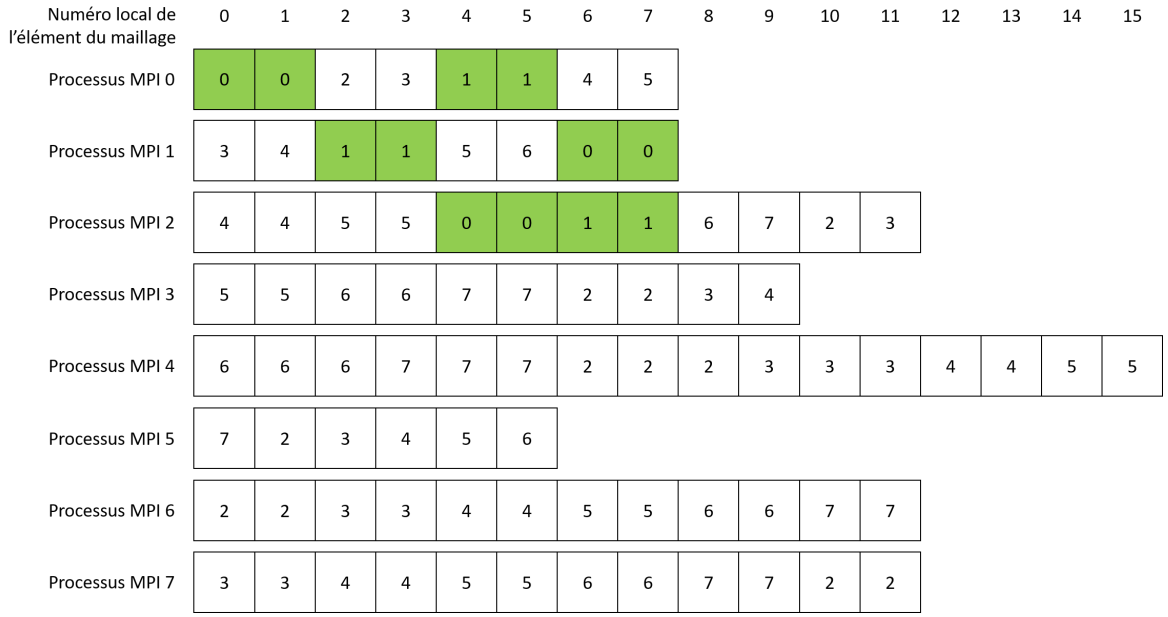


FIGURE 3.3. – Illustration du principe de partitionnement pour 8 processus MPI. vert = "super-éléments", blanc = éléments de non-contact. Le nombre de cases par processus correspond au nombre de "super-éléments" et d'éléments de non-contact que possèdent les processus après application de l'AMR et avant rééquilibrage des éléments. Le numéro indiqué dans chaque case spécifie le numéro du processus auquel sera attribué le "super-élément" de contact ou l'élément de non-contact.

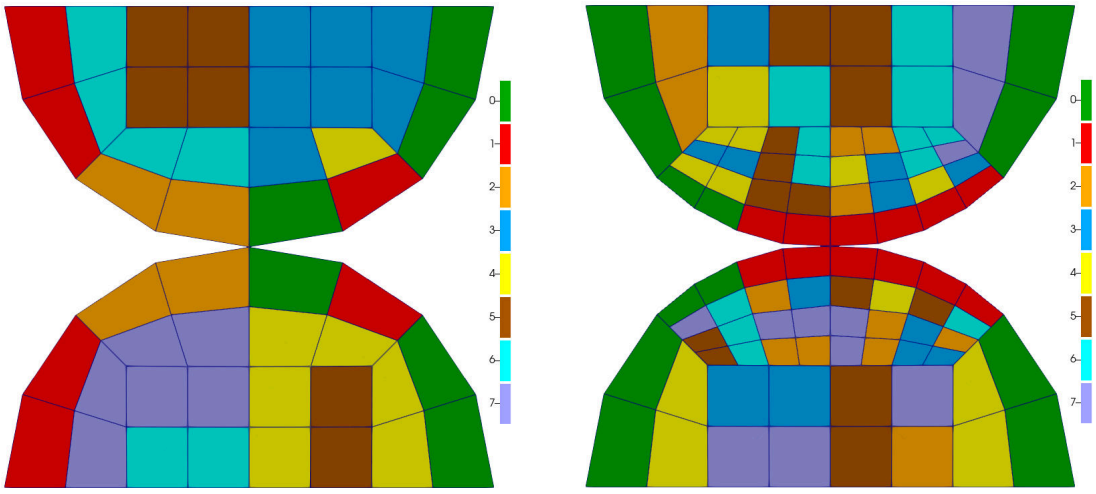
De plus, à l'itération AMR actuelle, il y a également 5 tâches MPI traitant des éléments de non-contact, qui, une fois l'étape de raffinement terminée, ne contiennent que des éléments de non-contact. Pour chaque processus MPI, les éléments (ou paires d'éléments de contact appairés) sont numérotés localement, et leur numéro local est défini en haut de la figure 3.3. Après l'application de l'équation (3.1) et des critères (3.2), R_C sera (dans cet exemple) de 2 à la prochaine itération AMR, et $R - R_C$ sera donc de 6. Les tâches MPI traitant les éléments de contact seront donc les processus 0 et 1, tandis que les processus traitant les éléments non-contact seront les processus 2, 3, 4, 5, 6 et 7. Le décalage du numéro du processus de départ pour le partitionnement est aussi mis en évidence avec l'exemple de partitionnement présenté dans la figure. En effet, pour chaque processus, le numéro du processus d'affectation de l'élément du maillage correspondant à la première case verte (respectivement blanche) en partant de la gauche vaut r mod R_C (respectivement $\mathrm{r} + R_C$ si $\mathrm{r} + R_C < R$ ou $\mathrm{r} + R_C$ mod $R - R_C$ sinon, avec 'mod' le modulo) : par exemple, la première case blanche du processus 4 a reçu le numéro 6 et celle du processus 5 a reçu le numéro 7. De même, cette figure permet d'observer que le numéro de processus de fin (pour les éléments de contact : $(\mathrm{r} + N_{E,C,\mathrm{r}}/2)$ mod R_C ; pour les éléments de non-contact : $\mathrm{r} + R_C + N_{E,\bar{C},\mathrm{r}}$ si $\mathrm{r} + R_C + N_{E,\bar{C},\mathrm{r}} < R$ ou $\mathrm{r} + R_C + N_{E,\bar{C},\mathrm{r}}$ mod $R - R_C$ sinon) diffère selon le processus considéré (et le nombre d'éléments qu'il a à redistribuer). Par exemple,

pour les éléments de contact du processus 1, le processus de fin vaut 0 et pour les éléments de non-contact du processus 4, le processus de fin vaut 5.

Par ailleurs, il est à noter que le cas où $N_{E,C,r}/2 < R_C$ (qui signifierait que le nombre de cases vertes est inférieur strictement à R_C) ne se produit pas dans l'exemple proposé ici car pour chacun des processus ayant des éléments de contact (des cases vertes), le nombre de cases vertes est supérieur à $R_C = 2$.

Ainsi, l'algorithme de partitionnement proposé fournit une nouvelle partition locale du maillage qui est celle donnée dans la figure 3.3, où le numéro inscrit dans chaque case indique le processus MPI auquel l'élément (ou la paire d'éléments de contact) est nouvellement assigné.

Appliquer la stratégie proposée sur l'exemple présenté dans la figure 3.3 conduit au partitionnement de maillage illustré dans la figure 3.4. La figure 3.4 affiche les numéros des tâches MPI attribués aux éléments d'un maillage 2D utilisé pour une simulation de contact avec AMR (ici, $c = 1$). Le calcul a été effectué en utilisant 8 cœurs pour faciliter la compréhension de la figure, et les deux premières itérations AMR sont présentées. La figure 3.4 montre que la numérotation globale initiale des éléments du maillage (effectuée par le générateur de maillage Gmsh [Geuzaine et al. 2009]) influence le partitionnement du maillage, tant lors de la première itération AMR que lors des itérations suivantes. En effet, la numérotation globale du maillage est utilisée pour le partitionnement initial et influe sur la nouvelle numérotation locale des éléments, obtenue une fois le partitionnement de maillage appliqué.

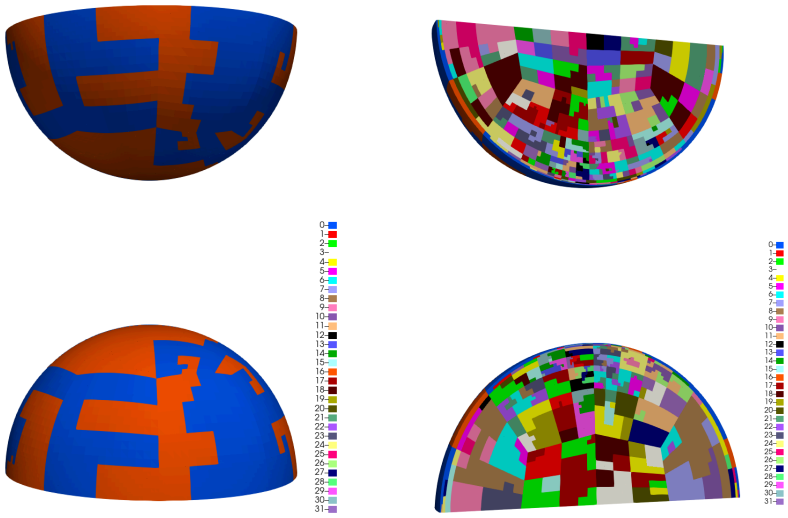


(a) (b)

Figure 3.4. – Illustration de l'équilibrage de charge "equidistributed" basé sur le contact pour un maillage 2D, typique du problème Hertzien - Chaque couleur identifie une tâche MPI spécifique - Partitionnements de maillage lors de la première et de la deuxième itération AMR en (a) et (b) respectivement.

Comme mentionné précédemment, ce partitionnement de maillage garantit que les termes de contact soient construits sans communication inter-processus MPI. Il est aussi à noter que la méthode de partitionnement proposée garantit un bon équilibrage des éléments sur les processus mais ne répartit pas ces éléments de manière parfaitement équilibrée. D'autre part, ce partitionnement de maillage induit des régions MPI composées d'éléments dispersés dans tout le maillage. Cela peut également être observé dans la figure 3.5, qui

affiche les numéros des tâches MPI attribués aux éléments d'un maillage 3D utilisé pour une simulation de contact avec AMR (ici, $c = 1$). Le calcul a été effectué en utilisant 32 cœurs pour obtenir un nombre acceptable de couleurs, et les arêtes du maillage ne sont pas représentées pour mieux distinguer les changements de couleur, ce qui signifie un changement de tâche MPI entre deux éléments voisins.



(a) (b)

FIGURE 3.5. – Illustration de l'équilibrage de charge "equidistributed" basé sur le contact pour un maillage 3D, typique du problème Hertzien - Couleur représentant la tâche MPI.

Comme prévu, les nœuds appairés pour le contact sont sur la même tâche MPI, mais de nombreuses interfaces entre sous-domaines MPI sont créées à l'intérieur de chaque solide.

3.3.2. Partitionnement "pie chart"

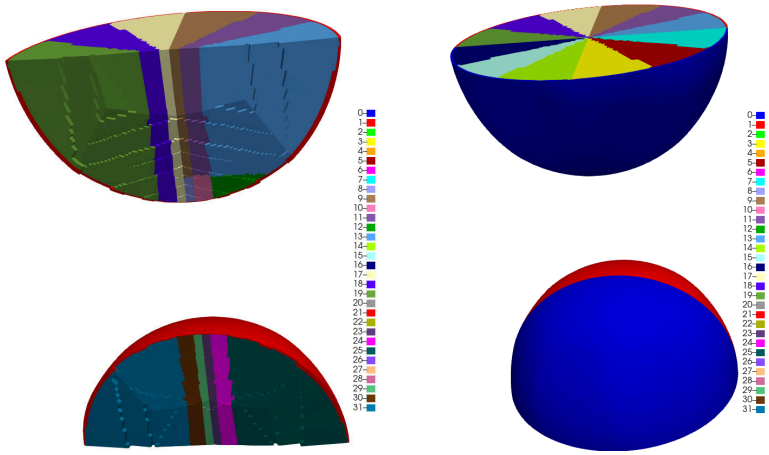
L'algorithme de partitionnement que l'on nomme "pie chart" repose sur un même principe général que celui du partitionnement "equidistributed" présenté en section 3.3.1. En effet, les éléments de contact et intérieurs aux volumes des deux solides (de non-contact) sont traités dans des processus distincts et les équations (3.1) et (3.2) permettent de déterminer le nombre de tâches MPI traitant les éléments de contact. La grande différence entre cet algorithme de partitionnement et l'algorithme "equidistributed" réside dans le choix d'affectation des éléments aux différents processus. Le but est ici de pouvoir regrouper les éléments voisins dans une même région parallèle et donc de limiter au maximum la dispersion des éléments d'une même région que nous avons pu constater avec la méthode de partitionnement "equidistributed" sur la figure 3.5 par exemple.

L'algorithme "pie chart" repose sur le même principe qu'il s'agisse de la redistribution des éléments de contact ou de non-contact et peut être résumé comme suit. Tout d'abord, des sous-ensembles d'éléments spatialement voisins sont créés. Ensuite, pour chaque sous-ensemble d'éléments voisins, un certain nombre de processus MPI vont se répartir les éléments du sous-ensemble qui leur est dédié. Il est à noter que ces processus sont liés à un seul et unique sous-ensemble et ne vont pas se voir assignés des éléments d'un autre sous-ensemble. Pour chaque sous-ensemble, afin que les éléments répartis sur chaque

processus MPI soient spatialement très peu dispersés, ces éléments sont préalablement triés selon leur position.

L'algorithme "pie chart" est composé de différentes étapes. Ces étapes s'appliquent à l'identique pour le partitionnement des éléments de contact et pour celui des autres éléments (non-contact).

La première étape consiste à calculer la position moyenne de chaque élément du maillage et à affecter cet élément à un sous-ensemble d'éléments voisins (la forme des sous-ensembles d'éléments voisins est un paramètre qui dépend de la géométrie des solides) tout en veillant à ce que les éléments de contact en vis-à-vis soient bel et bien dans le même sous-ensemble. Par exemple, pour le cas de deux sphères en contact, nous avons choisi de définir ces sous-ensembles via des parts de camembert, comme illustré sur la figure 3.6. Par ailleurs, dans ces travaux, le nombre de sous-ensembles a été choisi constant au cours du calcul et défini en fonction du nombre total de processus MPI utilisé dans le calcul considéré. [2] Ainsi, un certain nombre de processus MPI (défini dans la deuxième étape) se partageront les éléments appartenant à chacun des sous-ensembles.



(a) (b)

FIGURE 3.6. – Illustration d'un équilibrage de charge spécifique au contact via la stratégie "pie chart" pour un maillage 3D, typique du problème de Hertz.

La deuxième étape consiste, tout d'abord, à dénombrer le nombre d'éléments par sous-ensemble (par sommation sur tous les processus MPI confondus). Puis, le nombre de processus MPI qui traiteront chacun des sous-ensembles est défini. En effet, le nombre d'éléments par sous-ensemble pouvant varier entre les sous-ensembles, choisir d'affecter le même nombre de processus MPI à chaque sous-ensemble conduirait à un déséquilibrage des éléments sur les tâches MPI. Ainsi, pour deux sous-ensembles A et B, si le sous-ensemble A a plus d'éléments que le sous-ensemble B, alors le nombre de tâches MPI traitant les éléments du sous-ensemble A doit être supérieur au nombre de tâches MPI traitant les éléments du sous-ensemble B.

2. Le nombre de sous-ensembles est le plus souvent défini (très) inférieur au nombre de tâches MPI utilisées, mais doit être choisi (*a priori*) de manière à ce que l'équidistribution des éléments sur les processus soit la plus respectée possible, en particulier lorsqu'un faible nombre de cœurs de calcul est utilisé et où le risque de déséquilibre peut être le plus probable.

Dans le même esprit que pour la formule (3.1), ce nombre (entier) est calculé au prorata du nombre total d'éléments du maillage. Il est possible de le définir par la formule

$$R_{C,se} = \left\lceil \frac{N_{E,C,se}}{N_{E,C,tot}} R_C \right\rceil \tag{3.3}$$

pour les éléments de contact et par la formule

$$R_{\bar{C},se} = \left\lceil \frac{N_{E,\bar{C},se}}{N_{E,\bar{C},tot}} (R - R_C) \right\rceil \tag{3.4}$$

pour les autres éléments, avec pour chaque type i (contact ou non-contact) et chaque sous-ensemble se :

— $R_{i,se}$ le nombre de tâches MPI traitant les éléments du sous-ensemble se ;

— $N_{E,i,se}$ le nombre d'éléments dans le sous-ensemble se ;

— $N_{E,i,tot}$ le nombre total d'éléments (de contact ou de non-contact) ;

— R (respectivement $R - R_C$) le nombre total de tâches MPI traitant les éléments de contact (respectivement les éléments de non-contact).

Cette valeur est ensuite ajustée[3] afin que la somme des $R_{C,se}$ soit bien égale à R_C.

Pour illustrer le fonctionnement de cette deuxième étape, nous nous appuyons sur l'exemple simple décrit dans la figure 3.7 et inspiré de la figure 3.3. Dans cet exemple, les maillages rectangulaires présentés ont subi un raffinement local de maillage. Le maillage obtenu après raffinement est composé de 36 éléments au total que nous souhaitons répartir sur 12 processus MPI et est présenté en figure 3.7a. Après application de l'équation (3.1), 4 processus traitent les éléments de contact et 8 tâches MPI traitent les autres éléments ($c = 1$ ici). La première étape a permis de définir neuf sous-ensembles différents : 3 pour les éléments de contact et 6 pour les autres éléments. Ils sont distingués sur la figure 3.7b. En particulier :

— pour les éléments de contact : un bleu clair (composé de 4 éléments), un blanc (composé de 4 éléments) et un orange (composé de 4 éléments) ;

— pour les éléments de non-contact : un rouge (composé de 6 éléments), un violet (composé de 3 éléments), un gris (composé de 3 éléments), un vert (composé de 6 éléments), un jaune (composé de 3 éléments) et un bleu foncé (composé de 3 éléments).

Ainsi, dans cet exemple et en appliquant les équations (3.3) et (3.4), il y aura :

— 2 tâches MPI qui vont recevoir les éléments du sous-ensemble bleu clair ;

— 1 tâche MPI qui va recevoir les éléments du sous-ensemble blanc ;

— 1 tâche MPI qui va recevoir les éléments du sous-ensemble orange ;

— 2 tâches MPI qui vont recevoir les éléments du sous-ensemble rouge ;

— 1 tâche MPI qui va recevoir les éléments du sous-ensemble violet ;

— 1 tâche MPI qui va recevoir les éléments du sous-ensemble gris ;

— 2 tâches MPI qui vont recevoir les éléments du sous-ensemble vert ;

— 1 tâche MPI qui va recevoir les éléments du sous-ensemble jaune ;

— 1 tâche MPI qui va recevoir les éléments du sous-ensemble bleu foncé.

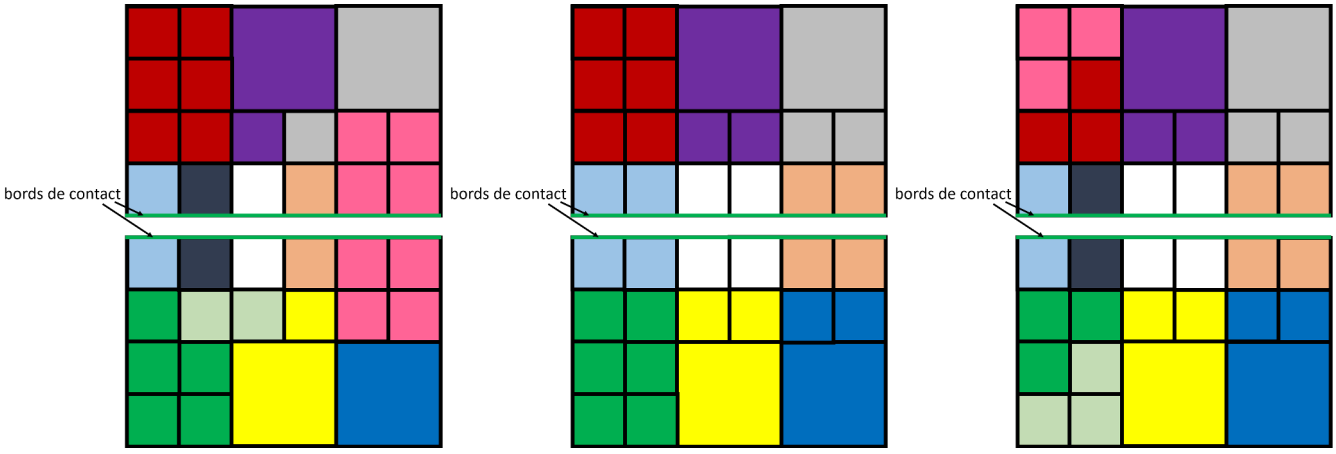


FIGURE 3.7. – Partitionnement "pie chart" : exemple de la répartition des éléments après application de l'AMR et avant rééquilibrage en (a), de sous-ensembles obtenus en (b) et de son partitionnement résultant en (c) pour un maillage 2D rectangulaire - Pour la figure en (b), une couleur correspond à un sous-ensemble ; pour les figures en (a) et en (c), une couleur correspond à un processus.

La troisième étape consiste, pour un sous-ensemble donné, à trier les éléments en regardant leur position selon une direction prédéfinie. Dans tous les cas étudiés en utilisant cette méthode de partitionnement, la direction choisie sera l'axe de la normale au contact (voir figure 3.6). De plus, l'algorithme utilisé pour trier les éléments selon leur position est un algorithme de "tri rapide" [4].

Enfin, une dernière étape permet, pour un sous-ensemble donné, d'affecter les éléments aux tâches MPI de manière la plus équirépartie possible, en utilisant le résultat du tri. Ainsi, un sous-ensemble qui a $N_{E,i,se}$ éléments triés à distribuer sur $R_{i,se}$ tâches MPI va répartir les $\lceil N_{E,i,se}/R_{i,se} \rceil$ premiers éléments de la liste triée sur le premier processus et ainsi de suite pour les autres éléments sur les autres processus.

Le partitionnement du maillage découlant de l'exemple décrit dans la figure 3.7 est affiché dans la figure 3.7c où chaque couleur correspond à un processus. À noter que les formules (3.3) et (3.4) permettent de minimiser le déséquilibre potentiel du nombre d'éléments entre les processus.

Ainsi, le partitionnement proposé permet de réduire drastiquement la dispersion des éléments dans le maillage mais aussi, comme pour l'algorithme "equidistributed", de s'assurer que les éléments en contact se trouvent sur les mêmes processus, comme montré sur la figure 3.6. Cette dispersion peut se quantifier globalement via les deux métriques suivantes : le rapport surface-volume (*surface-to-volume ratio* en anglais) et le nombre de sommets de frontière (*boundary vertices* en anglais). Le *surface-to-volume ratio* permet de quantifier la surface d'un objet en fonction d'une unité de volume. Pour leur part, les *boundary vertices* sont, en théorie des graphes, les sommets du graphe (les éléments finis

3. Si la somme des $R_{i,se}$ est différente de R (ou R_C) et $|R_{i,se} - R| = D_R$ (ou $|R_{i,se} - R| = D_R$), alors la valeur $R_{i,se}$ est incrémentée ou décrémentée de 1 pour les D_R premiers sous-ensembles.

4. https ://fr.wikipedia.org/wiki/Tri_rapide

du maillage dans notre cas) qui ont des sommets voisins (reliés par une arête) assignés à d'autres processus. Optimiser (minimiser) ces deux métriques tend à limiter la dispersion des éléments dans le maillage et permet ainsi de réduire le volume des communications. En observant les répartitions des éléments sur les processus présentés dans les figures 3.5 et 3.6, il est possible de se rendre compte que la méthode de partitionnement "pie chart" optimise beaucoup mieux ces métriques que celle "equidistributed". Pour plus de détails sur ces métriques, le lecteur peut se référer à [Hendrickson 1998 ; Meyerhenke et al. 2006].

De plus, avec cet algorithme, défini par un paramétrage (nombre et forme des sous-ensembles en particulier), il est possible de faire en sorte que les éléments intérieurs aux volumes du premier et du second solide soient affectés à des processus MPI différents, comme c'est le cas ici pour le partitionnement des deux sphères représentées dans la figure 3.6. La figure 3.6 montre les numéros des tâches MPI assignés aux éléments du maillage 3D utilisé pour une simulation avec AMR (ici, $c = 1$). Comme pour la figure 3.5, le calcul représenté a été effectué en utilisant 32 cœurs CPU afin d'avoir un nombre raisonnable de couleurs, et les arêtes des éléments du maillage ne sont pas représentées dans le but de mieux distinguer les changements de couleur qui représentent un changement de tâche MPI entre deux éléments voisins.

3.4. Partitionnement dérivé de la méthode Space Filling Curve (SFC) "SFC-contact"

Le troisième algorithme de partitionnement mis en œuvre, nommé "SFC-contact", consiste à modifier *a priori* le partitionnement SFC déjà implémenté dans MFEM dans le but de l'adapter à la résolution de nos problèmes en mécanique du contact. Dans cet algorithme, les éléments de contact en vis-à-vis sont regroupés en un seul "super-élément" dans le but de s'assurer que les éléments de contact soient en vis-à-vis et *a fortiori* que les nœuds de contact appairés soient sur les mêmes processus. La définition de ces "super-éléments" est représentée schématiquement par la figure 3.8.

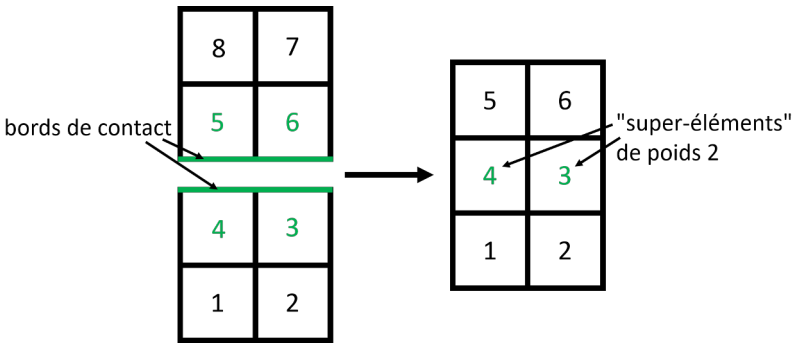


Figure 3.8. – Illustration de l'adaptation du partitionnement SFC au contact.

Comme c'est le cas pour le partitionnement "equidistributed" (voir la section 3.3.1), seul les numéros locaux et globaux de l'élément de contact du premier solide sont pris en compte. Ainsi, un élément de contact du second solide est identifié par le même indiçage que l'élément de contact du premier solide auquel il est appairé.

Ensuite, une fois que les "super-éléments" sont définis (ce qui implique une renumé-

rotation locale des éléments[5]), une méthode de partitionnement de type SFC utilisant la courbe de Hilbert est appliquée, en prenant en compte ces "super-éléments" en lieu et place des éléments de contact. Dans cet algorithme, les "super-éléments" se voient affecter un poids de 2 (car ils regroupent, pour un contact nœud-à-nœud, deux éléments du maillage) tandis que les éléments intérieurs aux volumes des solides ont un poids de 1. Cette technique d'affectation des poids sur les éléments pour l'application de la méthode de partitionnement SFC a déjà été testé dans d'autres contextes (voir [Lieber et al. 2018 ; Borrell et al. 2018 ; Borrell et al. 2020 ; Fernando et al. 2020] pour plus de détails).

Ces "super-éléments" sont répertoriés localement (pour chaque processus MPI) par l'appariement des éléments de contact en vis-à-vis. La routine utilisée pour cela est la même que celle appliquée pour l'appariement de ces éléments lors de l'application de l'une des deux premières méthodes de partitionnement introduites dans ce manuscrit.

Dans la bibliothèque MFEM, ces "super-éléments" sont ensuite transmis à la fonction `Rebalance` de la classe `ParNCMesh` (dans le cas où le maillage a subi un raffinement local et non-conforme du maillage au préalable), qui a été étendue pour prendre en compte ces "super-éléments".

En ce qui concerne les maillages uniformément raffinés fournis en entrée du code, le constructeur `ParMesh` de la classe `ParMesh` permet de créer un maillage parallèle à partir du maillage fourni ainsi que d'un partitionnement soit fourni en entrée de la fonction, soit créé en "boîte noire". Cependant, aucune fonction utilisant la méthode SFC n'est implémentée dans la bibliothèque pour le partitionnement d'un maillage conforme et ne peut être appelée par la fonction `ParMesh`. En effet, par défaut, c'est-à-dire si aucun partitionnement n'est fourni en entrée et dans un cas classique sans contact, le maillage est partitionné via la bibliothèque Metis. Ainsi, afin de considérer le contact dans le partitionnement initial du maillage, le constructeur `ParMesh` de la classe `ParMesh` est enrichi par l'appel de la fonction `GenerateSFCPartitioning` implémentée par nos soins au sein du code source de MFEM. Cette fonction fournit ainsi un partitionnement utilisant la méthode SFC et dédiée à la définition de "super-éléments". Cette fonction est analogue à la fonction `Rebalance` de la classe `ParNCMesh`.

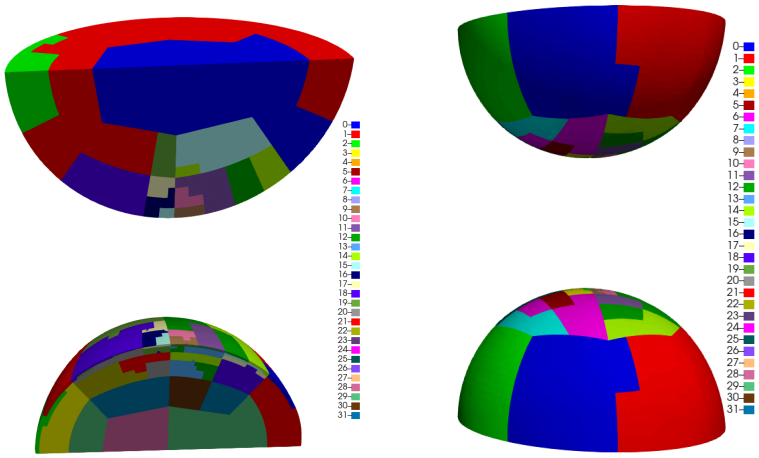
Le principe générique de l'algorithme de redistribution de type SFC, dédié au contact, des éléments du maillage pour la création d'un nouveau maillage parallèle est illustré par l'algorithme 3.

Algorithme 3 Algorithme de la redistribution des éléments avec le partitionnement "SFC-contact"

Entrée : Position spatiale des éléments finis de contact
Sortie : Nouveau maillage parallèle
Appariement des éléments de contact en vis-à-vis des deux solides
Appel de la fonction MFEM `ParMesh` ou `Rebalance` (enrichies par nos soins)
  Identification des éléments de contact du solide 1, du solide 2 et éléments de non-contact
  Partitionnement Hilbert SFC avec uniquement les "super-éléments" (avec la numérotation locale des éléments de contact du solide 1) et les éléments de non-contact
  Ajout des rangs MPI aux éléments de contact du solide 2
  Redistribution des éléments (appel de la fonction MFEM `RedistributeElements`)

5. La numérotation locale des éléments de contact du second solide est supprimée. Les éléments qui sont encore numérotés sont renumérotés dans le même ordre qu'initialement mais en s'assurant qu'il n'y ait pas de "saut" de numérotation entre deux éléments numérotés.

La figure 3.9 montre les numéros des 32 tâches MPI assignés aux éléments du maillage 3D pour une simulation avec AMR qui utilise cette approche de partitionnement. Tout d'abord, il est observé que plus de processus MPI traitent les éléments de contact que lorsque les deux autres stratégies de partitionnement sont utilisées. De plus, la figure permet de remarquer que les processus auxquels ont été assignés des éléments de contact possèdent également des éléments de non-contact. Il est à signaler que cette stratégie basée sur la méthode SFC est générique pour deux solides en contact et assure un bon regroupement spatial des éléments appartenant à un même processus.



(a) (b)

FIGURE 3.9. – Illustration d'un équilibrage de charge spécifique au contact via la stratégie "SFC-contact" pour un maillage 3D, typique du problème de Hertz.

3.5. Analyse préliminaire sur le cas du contact Hertzien

Cette section s'attache à étudier les performances parallèles de la stratégie AMR-contact (décrite dans le chapitre 2) sur un premier cas d'application en utilisant la méthode de partitionnement "equidistributed". Ainsi, des études de strong scaling (temps de passage dans une portion ou la globalité de l'algorithme en fonction du nombre de cœurs CPU), de consommation mémoire mais aussi de l'influence du pré-conditionnement y sont présentées. Enfin, l'intérêt de la mise en œuvre des méthodes de partitionnement "pie-chart" et "SFC-contact" y sont justifiées.

De plus, il est à noter que la version de MFEM utilisée dans cette section ainsi que dans toute la suite du chapitre est la version 4.4.

3.5.1. Présentation des simulations étudiées

Pour étudier la scalabilité parallèle de notre méthode, nous avons réalisé des études numériques avec un nombre variable de cœurs (en strong scaling) pour des simulations représentatives du problème du contact Hertzien sphérique en 3D tel décrit en section 2.3.

En plus des temps de calcul totaux, les temps des principales fonctions suivantes du programme sont intégrées au profilage (*profiling* en anglais) :

— assemblage de la matrice de rigidité éléments finis (fonction MFEM `Assemble`) ;

— formation du système linéaire sans prise en compte des termes de contact (fonction MFEM `FormLinearSystem`, équation (2.26)) ;

— ajout des termes de contact et résolution du système matriciel pénalisé (application du solveur issu de la classe MFEM `PenaltyPCGSolver`, équation (2.6)) ;

— raffinement des éléments (approche AMR estimate-mark-refine, fonction MFEM `Apply`) ;

— équilibrage et redistribution de l'ensemble des éléments finis du maillage après raffinement (incluant la fonction MFEM `Rebalance`).

Remarque : l'assemblage du second membre éléments finis n'est pas inclus dans ce profiling car son coût est négligeable par rapport à celui de la matrice de rigidité éléments finis.

Tout d'abord, la combinaison 1 pour l'AMR est appliquée au cas-test. Le seuil e_Ω est fixé à 2% et le maillage résultant est illustré dans la figure 2.11, page 80. Ce maillage est obtenu après 6 itérations AMR (donc 5 étapes de raffinement du maillage, sachant que le maillage initial, visible dans la figure 2.2a en page 67, a été pré-raffiné uniformément une fois par le mailleur Gmsh avant d'être fourni à MFEM) et la solution discrète du problème $\mathbf{U}$ comporte $2,3$ millions de degrés de liberté conformes. De plus, en incluant les degrés de liberté non-conformes et ceux fantômes, les calculs ont pu atteindre jusqu'à près de 13,8 millions de degrés de liberté, pour un calcul utilisant 512 unités de calcul. L'empreinte mémoire maximum des calculs (obtenue pour le calcul utilisant 512 cœurs) vaut 430 gigabytes (GB en anglais). Par ailleurs, il est à noter que, dans tout le chapitre, les éléments finis utilisés sont bilinéaires (Q_1).

Remarque : les valeurs d'empreinte mémoire renseignées dans ce manuscrit correspondent à la somme des empreintes mémoires sur tous les processus.

Comme vu à la section 3.3, le coefficient c de l'équation (3.1) permet de définir le nombre de processus MPI qui traitent les éléments de contact. Pour fixer ce coefficient c dans la formule d'équilibrage de charge considérée (3.1), des calculs sont effectués en faisant varier ce coefficient pour un nombre constant de tâches MPI. Les résultats obtenus sont présentés dans la figure 3.10 pour 512 tâches MPI. Ils montrent que la valeur 1 de ce paramètre permet de réduire le temps d'exécution du calcul. De plus, il est observé que le temps passé dans la fonction MFEM `FormLinearSystem` est minimisé lorsque $c = 1$ et au contraire devient très important (dans certains cas supérieur au temps passé dans la fonction `Assemble`) lorsque la valeur choisie pour c s'éloigne de 1. Cela s'explique par un déséquilibrage de charge lors de la réalisation du double produit matriciel $\mathbf{P}^T\tilde{\mathbf{K}}\mathbf{P}$ (voir équation (2.24), page 68). En effet, la taille de la matrice $\tilde{\mathbf{K}}$ s'avère près de 4 fois plus élevée sur certains processus pour $c = 4$ (en l'occurrence ici ceux qui ne traitent pas les éléments de contact puisque $c > 1$) par rapport à $c = 1$.

Il est observé que choisir une valeur de coefficient de c autre que 1 peut conduire a des temps de calcul plus élevés. D'autres calculs non présentés ici ont montré que le choix d'un nombre différent de processus MPI produit des mesures de temps conduisant à la même conclusion pour le choix de c.

Il est aussi intéressant d'étudier les performances parallèles de notre stratégie sur un maillage plus fin. C'est pourquoi, dans l'analyse de scalabilité réalisée et présentée dans les figures 3.11 et 3.12 dans les sous-sections qui suivent, le problème de contact est résolu en appliquant l'AMR avec la combinaison 1 et $e_\Omega = 1,5\%$, et avec la combinaison 2 et

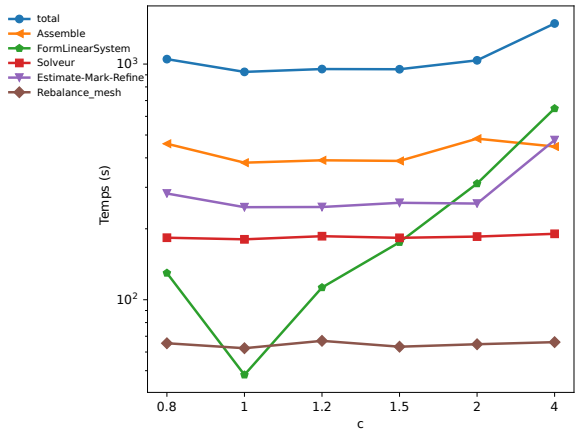


FIGURE 3.10. – Temps de calcul, étude paramétrique pour différentes valeurs de c - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 2\%$).

$e_{\Omega,\text{LOC}} = 4\%$ et $\delta = 0,1\%$. Le maillage initial uniformément raffiné est conçu de manière à ce qu'il y ait toujours plus d'éléments que de tâches parallèles et a donc a été pré-raffiné uniformément une fois par le mailleur Gmsh. Les solutions résultantes sont obtenues après 7 itérations AMR (donc 6 étapes de raffinement) et la solution discrète $\mathbf{U}$ du dernier problème "conforme" à résoudre comporte $9,0$ millions et $5,1$ millions de degrés de liberté respectivement. Les calculs ont été effectués sur les nœuds de calcul du supercalculateur Topaze du CCRT/CEA (processeurs AMD Milan, voir la section 1.8.5), chacun hébergeant 128 cœurs par nœud. Les résultats suivants sont obtenus en résolvant le système linéaire conforme en utilisant le solveur de gradient conjugué de hypre (HyprePCG), pré-conditionné par la stratégie de factorisation LU incomplète disponible dans hypre (HypreILU). Le lecteur peut notamment se référer à la section 1.6.3 ainsi qu'à [TAKAHASHI et al. 2014] pour plus de détails sur ces méthodes.

Pour la simulation avec la combinaison 1 d'AMR et $e_\Omega = 1,5\%$, les calculs ont pu atteindre jusqu'à près de 55,3 millions de degrés de liberté en incluant les degrés de liberté non-conformes et ceux fantômes, pour un calcul utilisant 1024 cœurs. En outre, l'empreinte mémoire maximum des calculs (obtenue pour le calcul utilisant 1024 cœurs) vaut 1549 GB.

Pour la simulation avec la combinaison 2 d'AMR et $e_{\Omega,\text{LOC}} = 4\%$ et $\delta = 0,1\%$, les calculs ont pu atteindre jusqu'à près de 34,5 millions de degrés de liberté en incluant les degrés de liberté non-conformes et ceux fantômes, pour un calcul utilisant 1024 cœurs. En outre, l'empreinte mémoire maximum des calculs (obtenue pour le calcul utilisant 1024 cœurs) vaut 1058 GB.

3.5.2. Étude de l'empreinte mémoire pour le partitionnement "equidistributed"

Lors des simulations, il a été observé que l'empreinte mémoire des calculs était élevée et constitue dans certains cas un facteur limitant. Cela semble principalement venir de la gestion distribuée du maillage au sein de MFEM. Ainsi, afin de réaliser nos études en scalabilité forte (strong scaling en anglais), il est décidé d'allouer 4 cœurs à chaque processus utilisé et d'augmenter ainsi l'empreinte mémoire maximum permise à chaque processus. Le tableau 3.1 rapporte l'empreinte mémoire utilisée pour les calculs AMR avec la combinaison 1 avec $e_\Omega = 1,5\%$ pour 256, 512 et 1024 cœurs CPU. Cette empreinte mémoire est listée pour les calculs utilisant les pré-conditionneurs ILU et BoomerAMG de

hypre, mais aussi pour la configuration sans pré-conditionneur (répertorié par le symbole ∅ dans le tableau). Pour chacune de ces simulations, le pourcentage de l'empreinte mémoire totale utilisée consommée par les étapes suivantes est indiqué dans ce tableau :

— la dernière étape de raffinement (approche AMR estimate-mark-refine) ;

— la dernière étape d'équilibrage de charge ;

— les étapes de résolution du système matriciel durant la dernière itération AMR.

Ce pourcentage correspond à la différence d'empreinte mémoire (définie grâce à la fonction `getrusage` de la structure de données `rusage`[6] et exprimée par rapport l'empreinte mémoire maximale totale de la simulation) entre l'instant où l'on sort de la fonction en question et l'instant où l'on y rentre.

Pré-conditionneur	∅			HypreBoomerAMG		HypreILU		
Cœurs CPU	256	512	1024	256	512	256	512	1024
Empreinte mémoire totale (GB)	662	973	1545	875	1218	671	977	1546
Dernière étape de raffinement (%)	40, 4	34, 5	28, 2	22, 4	21, 9	40, 7	34, 9	28, 3
Dernière étape de rééquilibrage (%)	8, 2	11, 0	18, 8	6, 3	8, 6	8, 1	10, 9	18, 8
Dernière itération AMR - Étapes solveur (%)	0	0	0	20, 5	17, 0	0	0	0

Tableau 3.1. – Empreinte mémoire - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - solveur HyprePCG avec différents pré-conditionneurs - nombre de degrés de liberté du maillage conforme final : 9, 0 millions - ordre de la transformation super-paramétrique 6.

Notons que la simulation pour 1024 cœurs avec le pré-conditionneur HypreBoomerAMG n'a pas convergé, pour une raison qui sera expliquée plus loin. Le tableau 3.1 permet d'observer que l'empreinte mémoire augmente avec le nombre de cœurs. Cela est dû principalement à la duplication de structures de données associées aux couches fantômes. Le tableau 3.1 montre que, comme prévu en raison de l'augmentation du nombre de ddls, la dernière étape de raffinement consomme beaucoup d'espace mémoire. Il en est de même pour la dernière étape de rééquilibrage de la charge, mais qui est un peu moins consommatrice que la dernière étape de raffinement (entre 1,5 et 5 fois moins d'après le tableau 3.1 selon le nombre de cœurs CPU utilisé). Enfin, les dernières résolutions du système matriciel induisent un coût important en matière d'espace mémoire lorsque le système est résolu avec le pré-conditionneur HypreBoomerAMG, mais pas du tout avec HypreILU ou sans pré-conditionneur. Ce coût en espace mémoire pour l'étape de résolution avec le pré-conditionneur HypreBoomerAMG est dû au stockage de la matrice de pré-conditionnement, qui est plus volumineuse que celle pour le pré-conditionnement HypreILU. En effet, durant les étapes de résolution du système linéaire de la dernière itération AMR, le nombre de coefficients non nuls dans la matrice de pré-conditionnement est environ 2

6. Les tailles maximales de mémoire résidente utilisées de tous les processus participant à la simulation ont été sommées pour fournir cette valeur.

fois plus élevé avec HypreBoomerAMG qu'avec HypreILU comme pré-conditionneur pour le calcul utilisant 256 cœurs CPU, et est 4 fois plus important pour le calcul utilisant 512 cœurs CPU. Lors de la mesure de la mémoire, il a été observé avec HypreBoomerAMG que l'empreinte mémoire est plus importante que l'empreinte mémoire due au stockage théorique de la matrice de pré-conditionnement résultant du nombre de coefficients non nuls trouvés dans hypre. Cela est probablement lié aux choix d'implémentation concernant la parallélisation de la matrice de pré-conditionnement.

En conclusion, que les pré-conditionneurs HypreBoomerAMG et HypreILU soient utilisés ou non, les fonctions MFEM `Apply` (pour l'application de l'approche AMR estimate-mark-refine) et `Rebalance` (pour la redistribution des éléments du maillage) sont fortement consommatrices en empreinte mémoire pour les calculs présentés ici. Cela est en particulier le cas lors des dernières itérations AMR lorsque le nombre d'éléments finis du maillage (et donc le nombre de degrés de liberté) y est important. Enfin, pour la méthode "equi-distributed", si le pré-conditionneur HypreBoomerAMG est employé, la résolution du système linéaire par le solveur engendre une forte hausse de la mémoire allouée au cours des itérations AMR (en l'occurrence 20% de l'empreinte mémoire totale rien que pour la dernière itération AMR pour les simulations utilisant 256 et 512 cœurs étudiées).

3.5.3. Strong scaling pour le partitionnement "equidistributed"

Pour du raffinement de maillage avec la combinaison 1, la figure 3.11 montre le speedup en strong scaling ainsi que le temps total et les temps des étapes intermédiaires de l'algorithme en fonction du nombre de cœurs CPU.

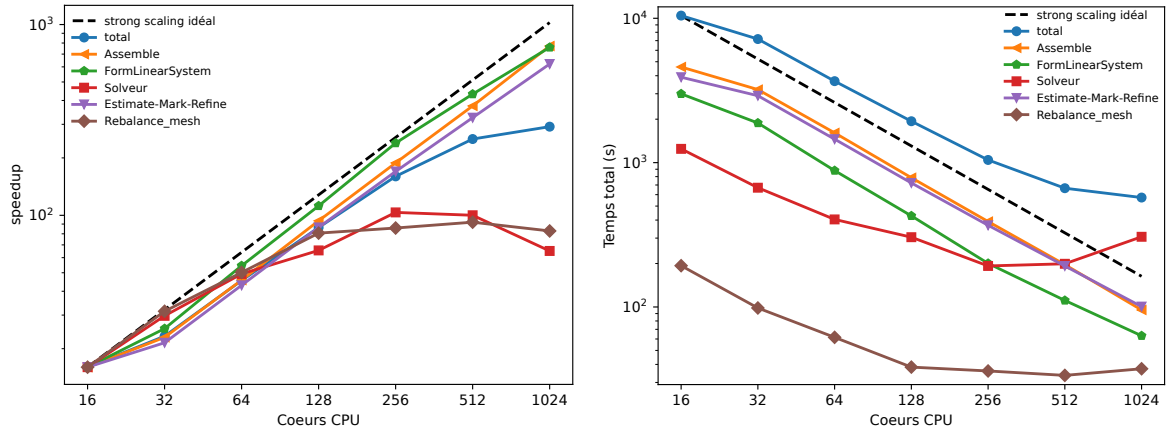


FIGURE 3.11. – Résultats des performances parallèles de l'algorithme et de ses principales étapes - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : 9,0 millions - ordre de la transformation super-paramétrique 6.

En comparant les résultats obtenus avec la courbe idéale, où la parallélisation serait parfaite et ne générerait aucun surcoût, nous pouvons observer que la scalabilité globale de l'algorithme obtenu est satisfaisante, en particulier jusqu'à 512 cœurs. D'autre part, au-delà de 512 cœurs, une perte de performance significative est observée dans l'étape de résolution du système linéaire par le solveur pré-conditionné. Le problème a été investigué à l'aide de l'outil de profilage Scalasca [BÖHME et al. 2012]. Il est dû à des schémas de

communication déséquilibrés et à des volumes de communication importants causés par le partitionnement du maillage. La raison en est que l'ensemble des éléments appartenant à un processus donné est composé d'éléments dispersés sur l'ensemble du maillage. Il en résulte des frontières importantes entre les régions parallèles, et donc des communications qui n'auraient pas lieu si chaque processus gérait un plus grand nombre d'éléments voisins les uns des autres. Une étude de profilage approfondie a montré que le volume en temps de communication est deux fois plus élevé que le volume en temps de calcul vis-à-vis de temps mesurés par la métrique de déséquilibre du chemin critique (*critical path imbalance metric* en anglais, exprimé en seconde dans l'outil Scalasca). De plus, les opérations les plus coûteuses sont les opérations de réduction et de communication MPI_Allreduce() et MPI_Waitall(), qui occupent près de 70 % du temps passé dans le solveur. C'est pourquoi il est d'intérêt de proposer une solution permettant de générer une partition du maillage créant des régions parallèles composées d'éléments mieux regroupés.

De plus, une perte de performance est également observée dans la figure 3.11 entre 16 et 32 cœurs CPU. La raison la plus probable est que la bande passante intra-nœud vue par le processeur est plus élevée pour 16 cœurs CPU que pour 32.

Pour de l'AMR avec la combinaison 2, des analyses de scalabilité similaires ont été réalisées et sont présentées dans la figure 3.12. Les valeurs seuil $e_{\Omega,\mathrm{LOC}} = 4\%$ et $\delta = 0,1\%$ ont été fixées pour mettre en évidence la spécificité de cette combinaison de critères pour le même nombre d'étapes de raffinement que précédemment pour la combinaison AMR 1. Les observations et les conclusions concernant la scalabilité globale sont les mêmes que pour le choix précédent de critères AMR. Cependant, la fonction MFEM qui permet la formation du système linéaire (`FormLinearSystem`) est plus performante qu'idéalement en particulier entre 32 et 128 cœurs CPU, ce qui est certainement dû à la présence d'effets de cache puisque nous utilisons globalement plus de mémoire cache avec un nombre croissant de nœuds de calcul. En effet, si un nombre croissant de cœurs est utilisé, alors il est typiquement possible d'employer de plus en plus de caches rapides en simultané. Dans notre cas, sur le supercalculateur Topaze, le cache L2 est de 512 KB par cœur. Il y a donc plus de place pour les données dans le cache L2 pour un nombre de cœurs qui augmente. Dans le cas d'une forte mise à l'échelle, la quantité totale de données pouvant être placées dans la mémoire cache L2 est de plus en plus grande avec le nombre de cœurs. Les accélérations superlinéaires signalées peuvent s'expliquer par ces effets de cache. Plus précisément, la routine de la bibliothèque hypre `hypre_BoomerAMGBuildCoarseOperator` est responsable de cette superlinéarité et consiste à réaliser l'opération matricielle définie par l'équation (2.24). Par ailleurs, soulignons que, comme le nombre final de degrés de liberté conformes est presque la moitié de celui de la simulation pour la combinaison AMR 1 précédente, les temps totaux sont réduits en conséquence.

En conclusion, la parallélisation de la stratégie avec AMR s'avère efficace jusqu'à 512 cœurs mais des améliorations sont possibles, notamment sur l'algorithme de partitionnement, afin d'obtenir de très bonnes performances parallèles pour un plus grand nombre de processeurs.

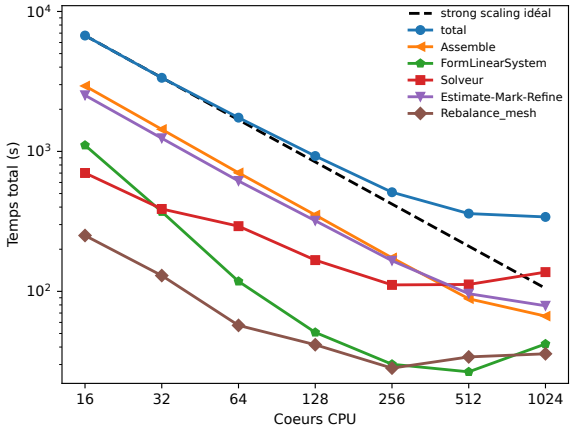


FIGURE 3.12. – Résultats des performances parallèles de l'algorithme et de ses principales étapes - Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\mathrm{LOC}} = 4\%$, $\delta = 0,1\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : 5,1 millions - ordre de la transformation super-paramétrique 6.

3.5.4. Étude du pré-conditionnement pour le partitionnement "equidistributed"

Une étude comparative des performances du solveur en fonction de choix du pré-conditionneur est présentée dans le tableau 3.2. Le tableau montre le nombre d'itérations et les temps d'exécution passés dans l'étape de résolution du système linéaire pour deux étapes de résolution particulières : la dernière étape de résolution (du problème de contact) pour les deux dernières itérations AMR. Ces résultats sont obtenus pour 256, 512 et 1024 cœurs CPU pour les pré-conditionneurs HypreBoomerAMG et HypreILU, de même que sans pré-conditionneur (caractérisé par le symbole ∅ dans le tableau).

Comme dit précédemment, la simulation avec le pré-conditionneur HypreBoomerAMG ne converge pas avec 1024 cœurs. Cela est dû à un problème de matrice de pré-conditionnement non symétrique définie-positive. Ce type de problème a également été rencontré en modifiant l'ensemble des paramètres du cas d'étude, par exemple en utilisant la combinaison 2 pour l'AMR ou en changeant la valeur du coefficient de pénalisation. La modification du seuil fort (Strong Threshold en anglais, noté ST dans le tableau) [7] ne nous permet pas de résoudre ce problème. Étant donné que le solveur de gradient conjugué développé dans MFEM conduit exactement au même problème lorsqu'il est pré-conditionné avec HypreBoomerAMG, le problème vient de l'association de ce pré-conditionneur avec un solveur de type gradient conjugué mais pas du solveur en lui-même. En outre, les simulations avec HypreILU utilisé comme pré-conditionneur ou en n'employant pas de pré-conditionneur se déroulent sans incident. Par ailleurs, nous avons vérifié que l'emploi de solveurs ne nécessitant pas de matrice de pré-conditionnement symétrique-définie-positive, tels que GMRES ou BiCGSTAB, permet bien d'utiliser HyperBoomerAMG, au prix d'un temps de calcul plus important.

Le tableau 3.2 montre que l'utilisation d'un pré-conditionneur de type HypreILU réduit le nombre d'itérations du solveur d'un facteur de 2 à 3 par rapport au cas sans pré-

7. Ce paramètre de HypreBoomerAMG est défini comme l'option la plus importante par les développeurs de la plateforme MOOSE [INL 2023].

Pré-conditionneur		∅			HypreBoomerAMG (ST = 0.7)			HypreILU		
Cœurs CPU		256	512	1024	256	512	1024	256	512	1024
Total - Étapes solveur	Temps (s)	205,9	242,7	504,2	264,4	341,5	∅	192,4	199,2	306,4
Avant-dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour CG	2580	2571	2352	36	36	42	890	974	1039
	Temps (s)	16,0	27,2	58,1	29,6	37,9	44,2	13,8	20,6	22,4
Dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour CG	4487	3815	3810	38	37	∅	1038	1059	1253
	Temps (s)	61,4	49,3	111,6	57,6	72,0	∅	39,2	29,8	67,1

Tableau 3.2. – Temps de résolution - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - solveur HyprePCG avec différents pré-conditionneurs - nombre de degrés de liberté du maillage conforme final : 9,0 millions - ordre de la transformation super-paramétrique 6.

conditionneur, et semble également performant en termes de réduction des temps de calcul. En effet, pour 1024 cœurs, le temps global passé dans le solveur y est 1,6 fois moins élevé que pour le cas sans pré-conditionneur et en particulier la résolution du système linéaire y est environ 2 à 2,5 fois plus rapide en fin de processus AMR, lorsque le nombre de degrés de liberté est important. De plus, il est observé que le nombre d'itérations du solveur obtenu varie en fonction du nombre de cœurs CPU utilisé, y compris pour le pré-conditionneur HypreBoomerAMG, réputé pour être robuste et efficace. Une explication à cela pour HypreBoomerAMG réside dans le fait que le pré-conditionneur a construit des matrices de pré-conditionnement sensiblement différentes selon le nombre de cœurs pour résoudre le problème pénalisé. Par ailleurs, le tableau montre que, globalement, les temps de résolution du système linaire sont environ 2 à 2,5 fois moins élevés en utilisant HypreILU comme pré-conditionneur plutôt que HypreBoomerAMG. Par conséquent, bien que l'utilisation du pré-conditionneur HypreBoomerAMG puisse réduire considérablement le nombre d'itérations du solveur, la combinaison du pré-conditionneur HypreILU avec le solveur HyprePCG s'avère plus efficace pour les cas d'étude traités et a été préférée pour la résolution de nos problèmes de contact. Elle est ainsi considérée dans la suite du manuscrit.

3.5.5. Apport des méthodes de partitionnement "pie chart" et "SFC-contact"

L'utilisation de la méthode de partitionnement "equidistributed" provoquant des problèmes de convergence de nos simulations lorsque ces dernières sont résolues par un solveur de type gradient conjugué pré-conditionné par HypreBoomerAMG, il est d'intérêt de connaître ce qu'il en est lorsque les méthodes de partitionnement "pie chart" ou bien "SFC-contact" sont appliquées. Ainsi, les simulations présentant les problèmes de convergence

explicités dans la section précédente sont réalisées avec l'emploi de ces deux méthodes de partitionnement.

Méthode de partitionnement		"equidistributed"	"pie chart"	"SFC-contact"
Total - Étapes solveur	Temps (s)	∅	22,3	16,8
Avant-dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour CG	45	30	31
	Temps (s)	54,0	2,1	1,5
Dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour CG	∅	31	31
	Temps (s)	∅	4,0	2,6

TABLEAU 3.3. – Temps de résolution et nombre d'itérations de HyprePCG avec le préconditionneur HypreBoomerAMG - Strong Threshold = 0,5 - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - nombre de degrés de liberté du maillage conforme final : 9,0 millions - 1024 cœurs CPU - ordre de la transformation super-paramétrique 6.

Méthode de partitionnement		"equidistributed"	"pie chart"	"SFC-contact"
Total - Étapes solveur	Temps (s)	∅	21,3	15,0
Avant-dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour PCG	42	35	33
	Temps (s)	59,3	1,9	1,3
Dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour PCG	∅	37	36
	Temps (s)	∅	3,9	2,3

TABLEAU 3.4. – Temps de résolution et nombre d'itérations de HyprePCG avec le préconditionneur HypreBoomerAMG - Strong Threshold = 0,7 - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - nombre de degrés de liberté du maillage conforme final : 9,0 millions - 1024 cœurs CPU - ordre de la transformation super-paramétrique 6.

Pour la combinaison AMR 1, la simulation qui nous intéresse a pour cible $e_\Omega = 1,5\%$ et 1024 cœurs CPU. Ces calculs ont été effectués en appliquant deux valeurs de Strong Threshold différentes : 0,5, valeur définie par défaut dans MFEM, et 0,7, valeur conseillée par les développeurs de la plateforme MOOSE [INL 2023]. Les résultats sont répertoriés dans les tableaux 3.3 et 3.4 respectivement. Dans ces deux tableaux sont notés, pour chaque méthode de partitionnement, le temps passé dans l'étape de résolution du système linéaire pénalisé ainsi que le nombre d'itérations et le temps total passé dans l'étape de

résolution pour la dernière itération de résolution du système linéaire des deux dernières itérations AMR.

Tout d'abord, en regardant le nombre de degrés de liberté du maillage mais aussi l'erreur estimée à la fin du calcul, il est à noter que les solutions éléments finis semblent être indépendantes du pré-conditionneur utilisé pour la résolution du système linéaire [8]. Cela est dû au fait que le critère de convergence du solveur est plutôt exigeant [9]. En outre, les résultats obtenus permettent de vérifier que, pour les deux valeurs de Strong Threshold fixées, l'utilisation de la méthode de partitionnement "equidistributed" conduit à une non convergence de la simulation lors de la dernière itération AMR, tandis que la simulation converge normalement avec les deux autres méthodes de partitionnement pour les deux valeurs de Strong Threshold choisies. Il est, de plus, observé que le temps de résolution est légèrement plus faible (25 à 30 %) avec la méthode de partitionnement "SFC-contact" qu'avec la méthode de partitionnement "pie chart" mais aussi que le choix du Strong Thresold influe peu sur les résultats que ce soit en nombre d'itérations de solveur (légèrement moins d'itérations en choisissant plutôt un Strong Threshold de 0,5) ou en temps de résolution (moins de 5 % de gain en temps de résolution avec Strong Threshold de 0,7). Cette variation en nombre d'itérations du solveur est directement liée au partitionnement du maillage réalisé et par conséquent à la méthode de partitionnement retenue. En effet, même si le système global (que nous aurions obtenu si nous résolvions le système linéaire de manière séquentielle) demeure identique quel que soit le partitionnement local du maillage, la manière dont sont répartis localement les éléments du maillage influe sur la manière dont est résolu le système matriciel. [10]

Par ailleurs, lors de l'avant-dernière itération AMR, il est observé que le nombre d'itérations du solveur est au moins 15% plus faible avec l'utilisation de la méthode de partitionnement "pie chart" ou "SFC-contact" par rapport à la méthode "equidistributed". De plus, lors de cette même avant-dernière itération AMR, le temps de résolution est réduit d'un facteur d'au moins 25 avec les stratégies "pie chart" ou "SFC-contact" par rapport à la méthode "equidistributed". Cela permet ainsi de valider l'intérêt des stratégies de partitionnement "pie chart" et "SFC-contact" pour la résolution de nos problèmes en mécanique du contact.

Pour la combinaison AMR 2, les simulations qui nous intéressent sont pour $e_{\Omega,\mathrm{LOC}} = 4\%$, $\delta = 0,1\%$ et 512 et 1024 cœurs CPU. Ces calculs ont été effectués en appliquant les deux valeurs de Strong Threshold choisies : 0,5 et 0,7. Les résultats sont respectivement répertoriés dans les tableaux A.1 (en annexe A) et 3.5 pour 512 cœurs CPU et A.2 (en annexe A) et 3.6 pour 1024 cœurs CPU. Ces tableaux possèdent la même structure que les tableaux 3.3 et 3.4 étudiés précédemment.

Ces résultats nous permettent de consolider les observations et de confirmer les conclusions faites pour la simulation pour la combinaison AMR 1 que ce soit en matière de convergence de la simulation, de nombre d'itérations solveur, de temps de résolution du système linéaire et d'influence du Strong Threshold dans les résultats obtenus. Ils permettent ainsi de vérifier l'intérêt de l'utilisation de stratégies de partitionnement de maillage telles que "pie chart" ou "SFC-contact" vis-à-vis de "equidistributed", où les

8. Les valeurs sont strictement identiques en ce qui concerne le nombre de ddls et la variation d'erreur estimée est négligeable par rapport à l'erreur estimée, quel que soit le nombre de cœurs utilisés, le pré-conditionneur ou la méthode de partitionnement.

9. Seuil fixé à 10^{-20} en erreur relative dans les simulations considérées.

10. La bibliothèque hypre fournit le nombre d'itérations global.

Méthode de partitionnement		"equidistributed"	"pie chart"	"SFC-contact"
Total - Étapes solveur	Temps (s)	∅	17,0	14,0
Avant-dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour PCG	∅	36	36
	Temps (s)	∅	1,5	1,3
Dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour PCG	∅	37	36
	Temps (s)	∅	3,0	2,2

TABLEAU 3.5. – Temps de résolution et nombre d'itérations de HyprePCG avec le préconditionneur HypreBoomerAMG - Strong Threshold = 0,7 - Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\text{LOC}} = 4\%$, $\delta = 0,1\%$) - nombre de degrés de liberté du maillage conforme final : 5,1 millions - 512 cœurs CPU - ordre de la transformation super-paramétrique 6.

Méthode de partitionnement		"equidistributed"	"pie chart"	"SFC-contact"
Total - Étapes solveur	Temps (s)	∅	14,7	11,8
Avant-dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour PCG	∅	36	35
	Temps (s)	∅	1,3	1,0
Dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour PCG	∅	38	36
	Temps (s)	∅	2,2	1,6

TABLEAU 3.6. – Temps de résolution et nombre d'itérations de HyprePCG avec le préconditionneur HypreBoomerAMG - Strong Threshold = 0,7 - Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\text{LOC}} = 4\%$, $\delta = 0,1\%$) - nombre de degrés de liberté du maillage conforme final : 5,1 millions - 1024 cœurs CPU - ordre de la transformation super-paramétrique 6.

domaines partitionnés obtenus sont éparpillés dans le domaine global de l'étude.

En outre, pour la simulation étudiée ici (nombre de degrés de liberté du maillage conforme final valant 5,1 millions), les résultats permettent d'observer que, contrairement aux simulations où la méthode de partitionnement "equidistributed" est employée (voir les valeurs correspondantes à HypreBoomerAMG dans le tableau 3.2), l'utilisation d'un partitionnement "pie chart" ou "SFC-contact" permet de réduire aussi bien le temps total de résolution du système linéaire que le temps de la dernière étape lorsque l'on augmente le nombre de cœurs CPU de 512 à 1024 (tableau 3.5 versus tableau 3.6). En effet, les résultats de la section 3.5.3 ont permis d'observer que, de manière globale, les performances parallèles de notre stratégie, mais aussi en particulier de l'étape de résolution du système

linéaire, se dégradent entre 512 et 1024 cœurs CPU lorsque la méthode de partitionnement "equidistributed" est employée. Par contre, notre stratégie se révèle plus performante entre 512 et 1024 cœurs lorsque les méthodes"pie chart" ou "SFC-contact" sont appliquées pour redistribuer les éléments dans le maillage. Cependant, pour une meilleure représentativité de l'efficacité parallèle des stratégies de partitionnement "pie chart" et "SFC-contact", des études plus approfondies sont effectuées et présentées dans les sections 3.6 et 3.7 sur des simulations plus volumineuses et impliquant l'utilisation d'un plus grand nombre de cœurs CPU.

3.6. Étude des stratégies de partitionnement sur le contact Hertzien

Ayant observé, d'une part, un bon comportement parallèle jusqu'à 512 cœurs CPU en utilisant la stratégie de partitionnement "equidistributed", mais aussi, d'autre part, une perte de performance au-delà, il est d'intérêt de confronter l'ensemble de nos stratégies de partitionnement afin de distinguer leurs performances parallèles à plus large échelle. Pour ce faire, deux simulations différentes du contact Hertzien 3D sont considérées : une simulation avec la combinaison 1 d'AMR et $e_\Omega = 1\%$ (nombre de degrés de liberté du maillage conforme final : $18,3$ millions) et une simulation avec la combinaison 2 d'AMR et $e_{\Omega,\text{LOC}} = 2\%$ et $\delta = 0,1$ (nombre de degrés de liberté du maillage conforme final : $39,4$ millions). De plus, le maillage initial uniformément raffiné est conçu de manière à ce qu'il y ait toujours plus d'éléments que de tâches parallèles et a donc a été pré-raffiné uniformément deux fois par le mailleur Gmsh. Ce maillage initial comporte 16384 éléments. Les calculs ont convergé après 6 itérations AMR, donc 5 raffinements du maillage.

Pour la simulation avec la combinaison 1 d'AMR et $e_\Omega = 1\%$, les calculs ont pu atteindre jusqu'à près de 34,4 millions de degrés de liberté avec la méthode de partitionnement "pie chart" et 26,4 millions de degrés de liberté avec l'approche de partitionnement "SFC-contact" en incluant les degrés de liberté non-conformes et ceux fantômes, pour un calcul utilisant 4096 cœurs. Cette différence de valeur de nombre de ddls est due à un nombre d'éléments fantômes qui diffère selon la méthode de partitionnement choisie. En outre, l'empreinte mémoire maximum des calculs (obtenue pour le calcul utilisant 4096 cœurs) vaut 2877 GB pour la méthode de partitionnement "pie chart" et 2664 GB pour l'approche de partitionnement "SFC-contact".

Pour la simulation avec la combinaison 2 d'AMR et $e_{\Omega,\text{LOC}} = 2\%$ et $\delta = 0,1$, les calculs ont pu comporter jusqu'à près de 72,6 millions de degrés de liberté avec la méthode de partitionnement "pie chart" et 57,3 millions de degrés de liberté avec l'approche de partitionnement "SFC-contact" en incluant les degrés de liberté non-conformes et ceux fantômes, pour un calcul utilisant 4096 cœurs. En ce qui concerne l'empreinte mémoire maximum des calculs (obtenue pour le calcul utilisant 4096 cœurs), elle vaut 4589 GB pour la méthode de partitionnement "pie chart" et 3983 GB pour l'approche de partitionnement "SFC-contact".

Pour obtenir les résultats présentés, le système linéaire est résolu à l'aide du solveur de gradient conjugué de hypre (HyprePCG), pré-conditionné avec HypreILU. De plus, pour des questions d'empreinte mémoire (voir les sections 3.5.2 et 3.6.3 à ce sujet), le nombre de cœurs par processus MPI est fixé à 4. Enfin, pour les calculs effectués en utilisant un nombre de cœurs CPU "faible" (< 512), les problématiques en empreinte mémoire nous ont obligés à augmenter le nombre de cœurs par processus pour ces calculs-ci uniquement.

En effet, lorsqu'un "faible" nombre de cœurs CPU est utilisé, l'espace mémoire allouée est plus restreint et une correction du nombre de cœurs par processus est nécessaire. Dans les simulations considérées, le nombre de cœurs par processus MPI est fixé à 16 pour le calcul utilisant 128 processus et est fixé à 8 pour celui employant 256 processus.

3.6.1. Strong scaling pour les trois stratégies étudiées

Pour la combinaison AMR 1, la figure 3.13 montre le temps total et les temps des étapes intermédiaires de l'algorithme en fonction du nombre de cœurs CPU pour les trois stratégies de partitionnement de maillage considérées. En figure 3.13a est tracée l'évolution du temps de calcul (en secondes) total de la simulation avec $e_\Omega = 1\%$ (nombre de degrés de liberté du maillage conforme final : $18,3$ millions) en fonction du nombre de processeurs tandis que sur les figures 3.13b, 3.13c et 3.13d sont représentées les évolutions du temps de résolution du système linéaire, d'application de l'approche AMR et du rééquilibrage des éléments du maillage respectivement en fonction également du nombre de processeurs. Dans ces figures, les résultats obtenus sont comparés avec la courbe idéale, courbe représentant une parallélisation parfaite. De plus, pour le partitionnement "equidistributed", le point pour 4096 cœurs CPU n'est pas représenté car la simulation n'a pas convergé en un temps raisonnable.

Comme déjà observé, les performances parallèles de la simulation utilisant le partitionnement "equidistributed" sont bonnes jusqu'à 512 processeurs mais tendent à se dégrader fortement au-delà, en particulier pour plus de 1024 cœurs CPU. Concernant les résultats avec un partitionnement "pie chart" ou "SFC-contact", un comportement hautement scalable est observé jusqu'à plus de 4000 cœurs CPU ; ce qui met en avant la très bonne qualité des partitionnements "pie chart" et "SFC-contact" mis en œuvre. De plus, ces résultats se confirment sur l'étape de résolution qui présente également une très bonne scalabilité jusqu'à plus de 4000 cœurs CPU. Par ailleurs, pour un nombre d'unités de calcul de 128 ou 256, l'étape de résolution du système matriciel n'est pas prépondérante par rapport à d'autres parties de la stratégie, comme l'étape d'application de l'AMR par exemple (voir la figure 3.13b par rapport à la figure 3.13c). Ainsi, le comportement parallèle de l'algorithme est impacté principalement par le temps passé dans la fonction d'application de la stratégie estimate-mark-refine, et par conséquent l'usage de la méthode "equidistributed" s'avère efficace pour ces faibles nombres de cœur CPU. De plus, la figure 3.13d montre que redistribuer les éléments du maillage avec l'approche "SFC-contact" se révèle moins coûteuse que les autres approches considérées.

La figure 3.14 montre les résultats obtenus en strong scaling pour les trois stratégies de partitionnement de maillage considérées pour de l'AMR avec la combinaison 2. De même, dans la figure 3.14a est tracée l'évolution du temps de calcul total de la simulation (en secondes) en fonction du nombre de cœurs CPU tandis que dans les figures 3.14b, 3.14c et 3.14d sont représentées les évolutions du temps de résolution du système linéaire, de l'application de l'AMR et du rééquilibrage de charge en fonction du nombre de cœurs CPU. Le calcul avec la méthode de partitionnement "equidistributed" n'a également pas convergé en des temps de calcul acceptables pour 4096 processeurs.

Les résultats obtenus permettent d'observer le même type de comportement parallèle que pour la simulation avec la combinaison 1 pour l'AMR ; ce qui confirme que les partitionnements "pie chart" et "SFC-contact" sont de très bonne qualité, quelle que soit la combinaison de critères AMR considérée.

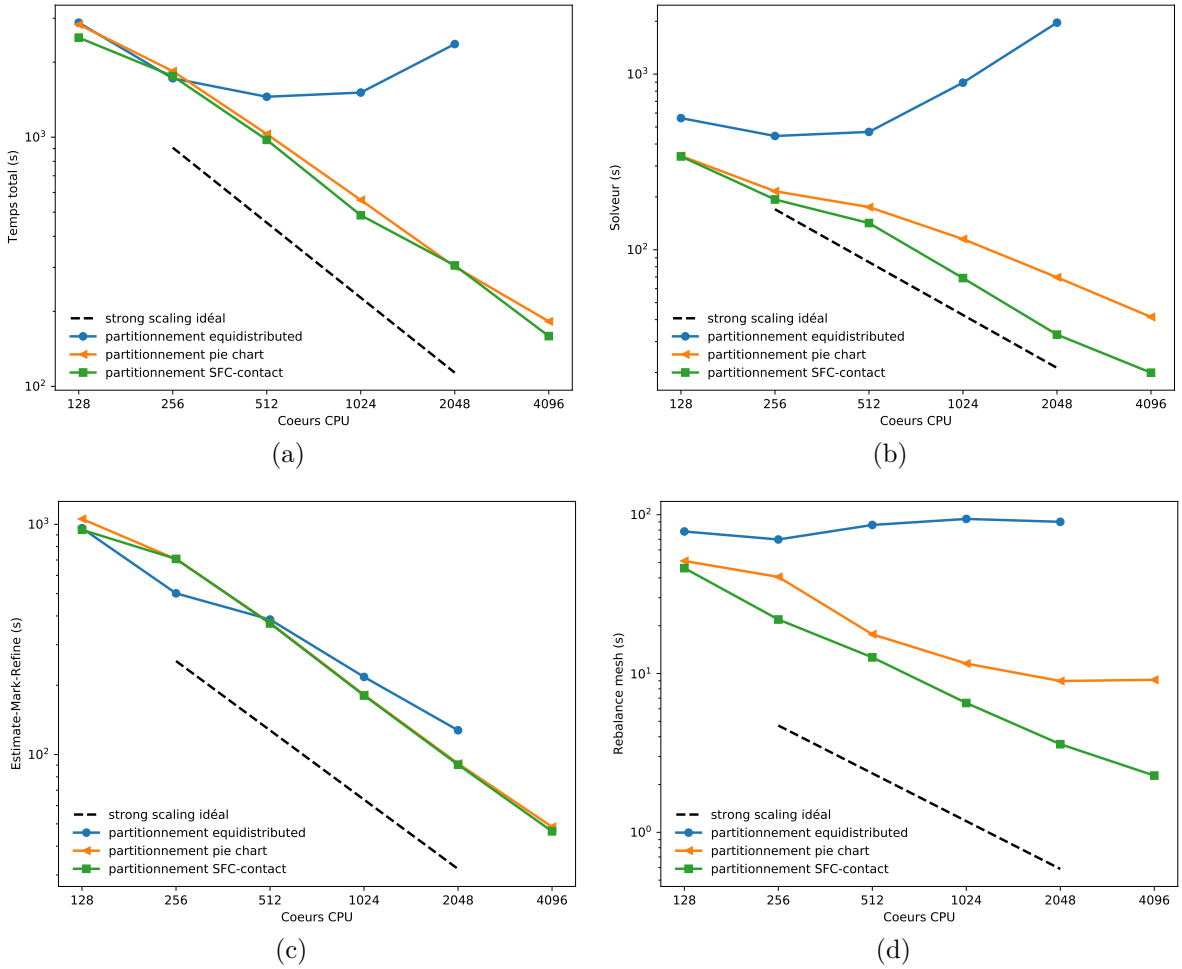


FIGURE 3.13. – Résultats des performances parallèles de l'algorithme en (a) et de ses principales étapes : résolution par le solveur en (b), application de l'approche AMR estimate-mark-refine (fonction MFEM `Apply`) en (c) et rééquilibrage du maillage (fonction MFEM `Rebalance`) en (d) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $18,3$ millions - ordre de la transformation super-paramétrique 6.

3.6.2. Comparaison de pré-conditionneurs pour les méthodes de partitionnement "pie chart" et "SFC-contact"

Nous avons vu à la section 3.5.4 qu'utiliser le pré-conditionneur HypreILU est à privilégier par rapport au pré-conditionneur HypreBoomerAMG lorsque l'algorithme de partitionnement "equidistributed" est appliqué, notamment en raison de problématiques de matrice non-symétrique définie positive rencontrées. Cependant, étant donné que les études menées dans la section 3.6.1 nous ont permis d'observer les très satisfaisantes performances de notre stratégie parallèles via l'application des algorithmes de partitionnement de maillage "pie chart" et "SFC-contact" et l'utilisation du pré-conditionneur HypreILU, il est tout de même d'intérêt d'analyser les performance parallèles de notre simulation en employant le pré-conditionneur HypreBoomerAMG afin de les comparer avec les résultats obtenus avec HypreILU. Ainsi, nous avons comparé nos résultats avec HypreILU et HypreBoomerAMG (avec un Strong Threshold ST de $0,7$) comme pré-conditionneurs

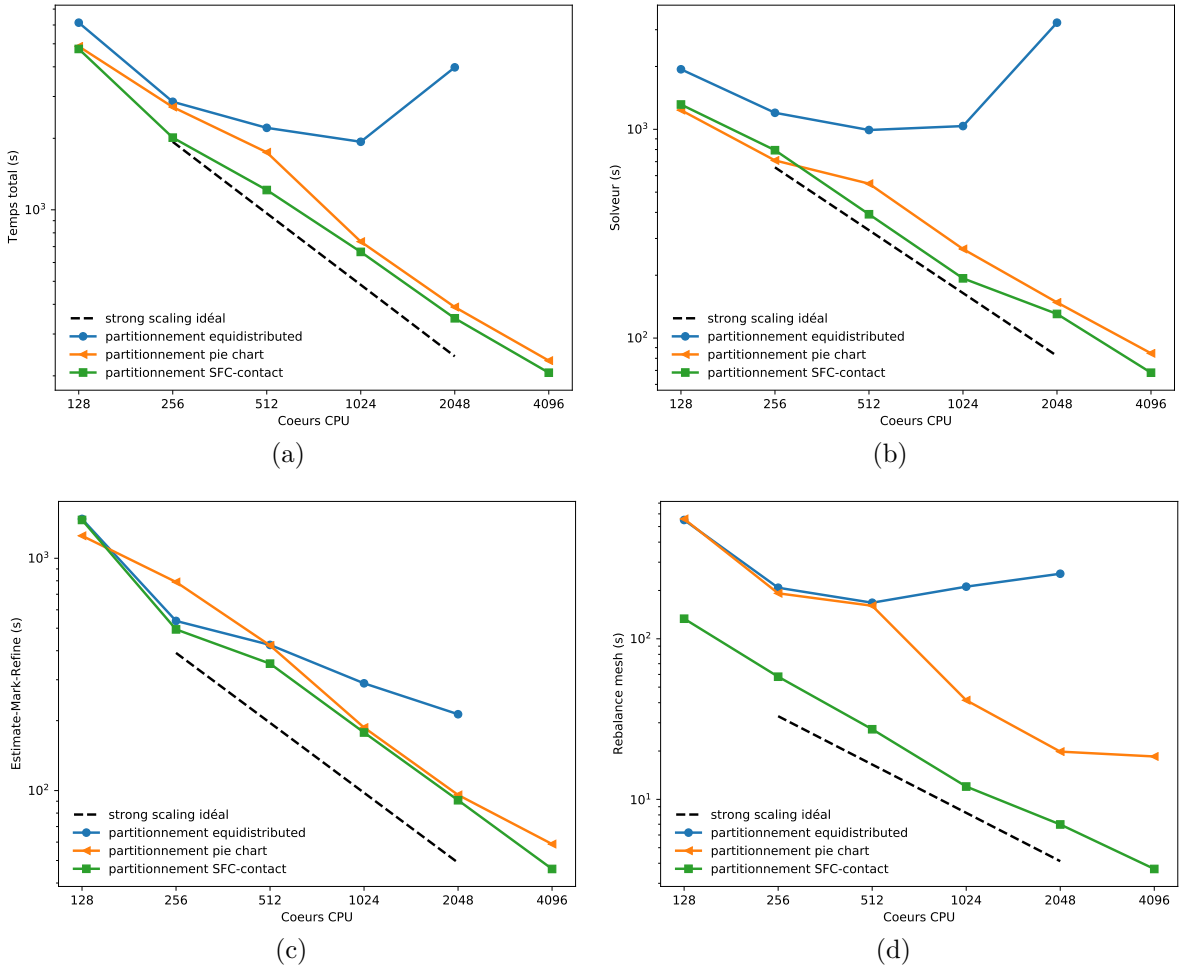


FIGURE 3.14. – Résultats des performances parallèles de l'algorithme en (a) et de ses principales étapes : résolution par le solveur en (b), application de l'approche AMR estimate-mark-refine (fonction MFEM `Apply`) en (c) et rééquilibrage du maillage (fonction MFEM `Rebalance`) en (d) - Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\mathrm{LOC}} = 2\%$, $\delta = 0,1$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $39,4$ millions - ordre de la transformation super-paramétrique 6.

(HyprePCG est toujours le solveur) pour 1024, 2048 et 4096 processeurs, et en utilisant les algorithmes de partitionnement "pie chart" et "SFC-contact". De plus, le maillage d'entrée utilisé est à transformation super-paramétrique d'ordre 6. Pour simplifier notre étude, seule la simulation avec la combinaison 1 pour l'AMR est présentée ici.

Ces simulations nous permettent de tracer les tendances concernant les temps de résolution du système linéaire, des nombres d'itérations du solveur pour la dernière étape de solveur en fonction du nombre de cœurs CPU utilisé. Elles sont représentées sur les figures 3.15, 3.16, 3.17a et 3.17b. Par ailleurs, les tableaux donnant toutes les indications chiffrées sont reportés en annexe B, en l'occurrence les tableaux B.1 pour la résolution avec HypreILU comme pré-conditionneur et B.2 pour la résolution avec HypreBoomerAMG comme pré-conditionneur.

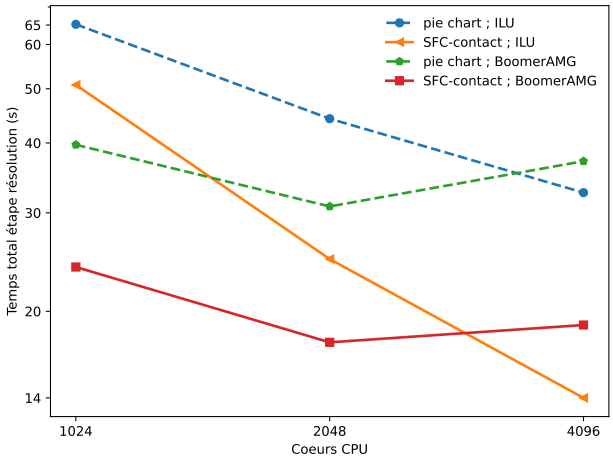


FIGURE 3.15. – Temps total passé dans l'étape de résolution du système matriciel (s) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - "pie chart" en pointillés et "SFC-contact" en trait plein - solveur HyprePCG - nombre de degrés de liberté du maillage conforme final : 18,3 millions - ordre de la transformation super-paramétrique 6.

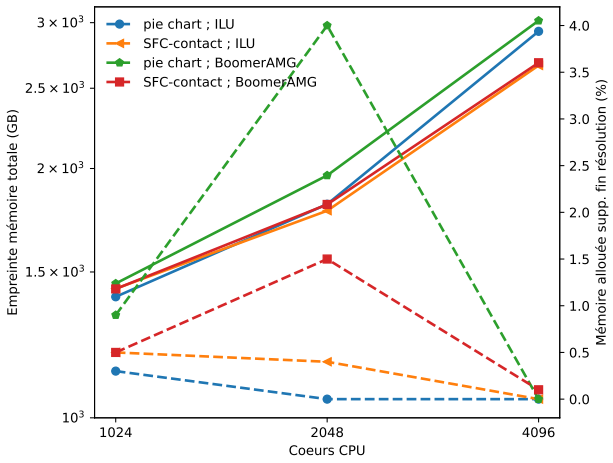
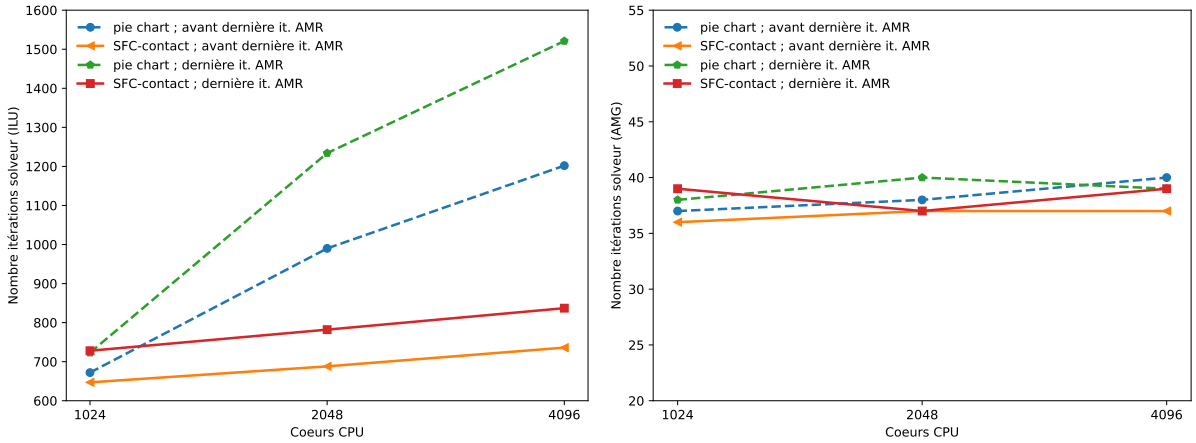


FIGURE 3.16. – Empreinte mémoire totale (à gauche et en traits pleins, en GB) et supplémentaire allouée lors des étapes de résolution du système linéaire à la dernière itération AMR (à droite et en pointillés, en % de l'empreinte mémoire totale) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG - nombre de degrés de liberté du maillage conforme final : 18,3 millions - ordre de la transformation super-paramétrique 6.

Les résultats obtenus nous permettent tout d'abord d'observer que, de manière globale, les temps de calcul et de résolution du système linéaire sont plus élevés avec la stratégie de partitionnement "pie chart" qu'avec celle "SFC-contact" (voir la figure 3.15). De plus, la figure 3.16 permet de remarquer que l'empreinte mémoire globale du calcul est également un peu plus élevée avec la stratégie de partitionnement "pie chart" qu'avec celle "SFC-contact". Ensuite, le nombre d'itérations du solveur à gradient conjugué pré-conditionné avec HypreILU tend à augmenter légèrement avec le nombre de processeurs avec l'approche de partitionnement "SFC-contact" tandis que cette augmentation est beaucoup plus forte avec la méthode de partitionnement "pie chart" (voir la figure 3.17a). D'autre part, la figure 3.17b montre que ce nombre d'itérations est très stable mais aussi drastiquement



(a) (b)

FIGURE 3.17. – Nombre d'itérations du solveur HyprePCG avec pour pré-conditionneur HypreILU en (a) et HypreBoomerAMG en (b) - valeurs pour le dernier passage dans l'étape de résolution par le solveur des deux dernières itérations AMR - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - nombre de degrés de liberté du maillage conforme final : $18,3$ millions - ordre de la transformation super-paramétrique 6.

moins important lorsque le pré-conditionneur HypreBoomerAMG est employé, quelle que soit la stratégie de partitionnement utilisée. Cette différence de comportement en nombre d'itérations solveur est dû au fait que ILU est une méthode de pré-conditionnement partielle, tandis que BoomerAMG est réputée pour être robuste et efficace. De plus, la différence de comportement entre les simulations avec "pie chart" et "SFC-contact" réside dans la taille des couches fantômes. En effet, la méthode de partitionnement "pie chart" engendre globalement une couche fantôme comportant plus de ddls que l'approche "SFC-contact" et cela implique la présence de plus de coefficients non nuls situés loin de la diagonale de la matrice à inverser. Nous observons aussi, via la figure 3.16 (axe de droite des ordonnées), que l'empreinte mémoire des étapes de résolution du système linéaire pour la dernière itération AMR, et *a fortiori* pour toutes les itérations AMR (voir section 3.5.2), est très faible par rapport à l'empreinte mémoire totale du calcul.

Par ailleurs, en analysant, à la figure 3.15, le strong scaling de nos résultats concernant la résolution du système linéaire, une perte de performance parallèle est observée lorsque HypreBoomerAMG est utilisé comme pré-conditionneur, contrairement aux résultats avec HypreILU qui se sont révélés plus scalables (mais plus coûteux à 1024 et 2048 cœurs).

En conclusion, l'utilisation du pré-conditionneur HypreILU engendre une hausse du nombre global d'itérations du solveur, en particulier avec la méthode "pie chart", qui est beaucoup plus élevé qu'avec HypreBoomerAMG comme pré-conditionneur. Cependant, les temps de calculs se révèlent compétitifs pour un grand nombre de cœurs de calcul (en l'occurrence 4096).

3.6.3. Influence du maillage d'entrée choisi pour les méthodes de partitionnement "pie chart" et "SFC-contact"

Comme nous avons pu nous en rendre compte dans la section 3.5.2, l'empreinte mémoire des calculs est un facteur limitant dans nos simulations et nous oblige à utiliser un nombre de cœurs par processus important (en l'occurrence 4 dans la majorité des calculs réalisés jusqu'ici). Nous avons donc cherché à analyser l'influence du maillage fourni en entrée de la simulation sur l'empreinte mémoire de celle-ci mais aussi sur son résultat éléments finis concernant la précision de la solution. Pour cela, nous avons opté pour une transformation super-paramétrique d'ordre moins élevé. L'ordre 3 choisi ici permet de conserver une bonne approximation de la courbure des géométries sphériques mises en jeu. Il est à noter également que, comme pour tous les résultats présentés dans ce chapitre, les éléments finis utilisés pour calculer la solution restent d'ordre 1 (Q_1). Des calculs identiques à ceux étudiés dans la section 3.6.2 (avec un ordre de transformation de 6) sont réalisés. Il est ainsi pertinent de comparer ces résultats avec ceux de la section 3.6.2 vis-à-vis du temps de calcul mais aussi de l'empreinte mémoire allouée. Pour cela, les évolutions du temps de calcul total mais aussi de l'empreinte mémoire totale en fonction du nombre de processeurs mis en jeu sont représentées dans les figures 3.18 et 3.19, pour toutes les configurations (partitionnements, pré-conditionneurs, ordre de la transformation) étudiées.

Remarque : Le lecteur pourra se référer à l'annexe B, en particulier dans les tableaux B.3 pour la résolution avec HypreILU comme pré-conditionneur et B.4 pour la résolution avec HypreBoomerAMG comme pré-conditionneur, afin d'avoir toutes les indications chiffrées sur les résultats. [11]

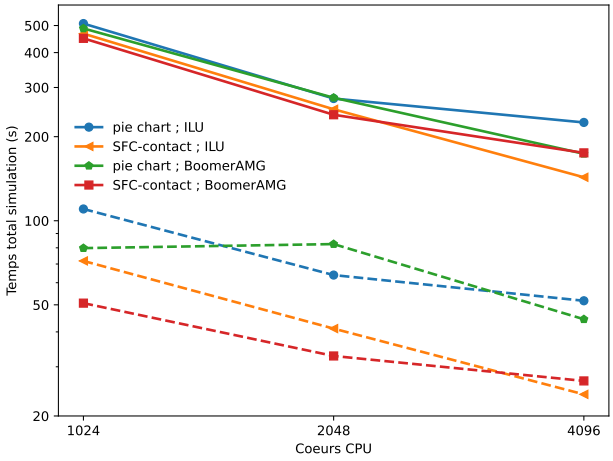


FIGURE 3.18. – Temps totaux - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG - nombre de degrés de liberté du maillage conforme final : 18, 3 millions - ordres de la transformation super-paramétrique 3 (en pointillés) et 6 (en trait plein).

Tout d'abord, les résultats obtenus, que ce soit avec HypreILU ou HypreBoomerAMG comme pré-conditionneur, nous permettent d'observer que l'application de la méthode de partitionnement "pie chart" se révèle de manière générale un peu plus gourmande en temps de calcul, en temps de résolution du système linéaire ou encore en empreinte

11. Ces tableaux sont construits de la même manière que ceux de la section 3.6.2.

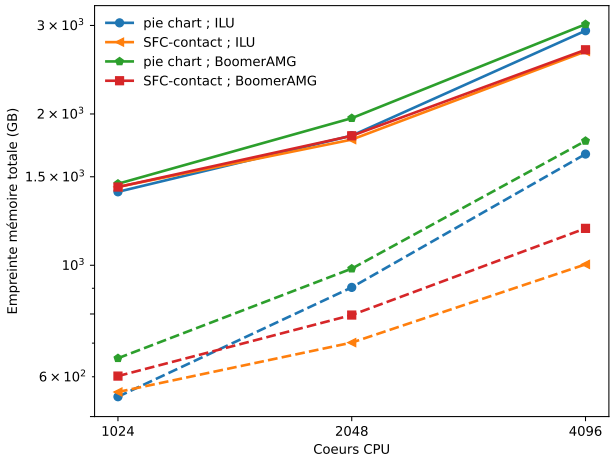


FIGURE 3.19. – Empreinte mémoire (GB) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG - nombre de degrés de liberté du maillage conforme final : 18,3 millions - ordres de la transformation super-paramétrique 3 (en pointillés) et 6 (en trait plein).

mémoire allouée. Comme cela a déjà été constaté dans la section 3.6.2, ces résultats mettent en évidence que l'emploi de HypreBoomerAMG permet un gain en temps de calcul pour un nombre de cœurs CPU de 1024 ou 2048 par rapport HypreILU mais tend à perdre en performance parallèle pour un plus grand nombre de processeurs.

Ensuite, la figure 3.18 montre que l'utilisation d'un maillage à transformation super-paramétrique d'ordre 3 au lieu de 6 permet de faire diminuer le temps de calcul d'un facteur 5 dans le cas testé. De plus, cela semble indépendant du nombre de processeurs et de la méthode de partitionnement choisis, ce facteur 5 se retrouvant globalement pour 1024, 2048 et 4096 cœurs CPU. En réalisant une analyse plus détaillée, il a été observé que cette diminution du temps de calcul est due essentiellement aux fonctions spécifiques de MFEM d'assemblage de la matrice de raideur globale $\hat{\mathbf{K}}$ (`Assemble`), de formation du système linéaire sans le contact (`FormLinearSystem`, équation (2.24) en particulier) et d'application de l'approche estimate-mark-refine choisie (`Apply`). En effet, pour ces trois fonctions MFEM en particulier, les résultats présentés en annexe C permettent d'observer que la diminution du temps passé dans ces dernières dû à l'utilisation d'un maillage à transformation super-paramétrique d'ordre 3 au lieu de 6 est d'un facteur 20 environ tandis que l'étape de résolution du système linéaire est (logiquement car indépendante de l'ordre de la transformation) globalement identique.

Ensuite, la figure 3.18 nous permet de confirmer que choisir la stratégie de partitionnement "SFC-contact" fournit des résultats meilleurs en temps de calcul et en performance parallèle, en particulier avec HypreILU comme pré-conditionneur. En effet, en observant les résultats obtenus pour un ordre de transformation de 3 sur la figure 3.18, le temps total passé dans la simulation se révèle environ 1,5 à 2 fois plus faible lorsque l'approche de partitionnement "SFC-contact" est appliquée plutôt que "pie chart", quel que soit le pré-conditionneur employé ou le nombre de cœurs CPU choisi. L'apport de l'algorithme de partitionnement "SFC-contact" est aussi d'intérêt vis-à-vis des empreintes mémoire allouées au cours de la simulation, comme le montre la figure 3.19. En effet, alors que l'empreinte mémoire totale semble globalement équivalente pour 1024 cœurs, quelle que soit la méthode de partitionnement appliquée pour un ordre de transformation

choisi, elle croît plus fortement entre 2048 et 4096 processeurs lorsque la méthode de partitionnement "pie chart" est employée, que l'ordre de la transformation soit fixé égal à 3 ou 6. En comparant les résultats obtenus en utilisant ces deux méthodes de redistribution des éléments, il est observé que le nombre de degrés de liberté total de la simulation (incluant les degrés de liberté fantômes mais aussi issus des nœuds non-conformes) est légèrement plus élevé avec la méthode "pie chart" qu'avec la méthode "SFC-contact" et cette différence tend à s'accroître avec le nombre de cœurs de calcul. Ainsi, cela permet de se rendre compte que la couche fantôme est plus importante avec "pie chart", ce qui peut expliquer cette différence dans l'évolution de l'empreinte mémoire totale.

Par ailleurs, comme tend à le confirmer la figure 3.20, visible à la page 118, par comparaison avec les résultats visibles sur la figure 3.16 qui étudie l'empreinte mémoire de la même manière mais pour des simulations dont l'ordre de la transformation super-paramétrique diffère, l'empreinte mémoire allouée aux étapes de résolution du système linéaire, et en particulier en fin de simulation, impacte plus fortement l'empreinte mémoire totale du calcul lorsque l'ordre de la transformation vaut 3 au lieu de 6. En effet, en particulier pour 1024 et 2048 processeurs, la proportion d'empreinte mémoire allouée aux étapes de résolution du système linéaire est importante lorsque l'algorithme de partitionnement "SFC-contact" est appliqué (par exemple, autour de 20% de l'empreinte mémoire maximale totale du calcul pour un calcul sur 1024 cœurs). Cependant, cette proportion tend à décroître lorsque le nombre de cœurs CPU augmente[12], notamment parce que les empreintes mémoire allouées pour les autres fonctions de la stratégie globale couplant contact et AMR deviennent prépondérantes.

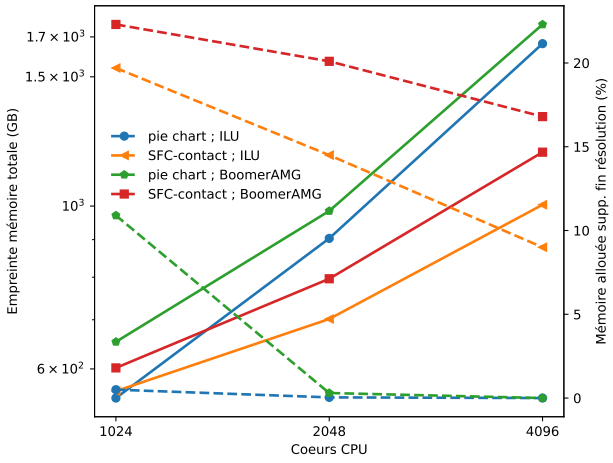


FIGURE 3.20. – Empreinte mémoire totale (à gauche et en traits pleins, en GB) et supplémentaire allouée lors des étapes de résolution du système linéaire à la dernière itération AMR (à droite et en pointillés) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG - nombre de degrés de liberté du maillage conforme final : $18,3$ millions - ordre de la transformation super-paramétrique de 3.

De plus, la figure 3.20 permet de conclure sur l'étude de l'empreinte mémoire. En effet, la combinaison de la méthode de redistribution des éléments "SFC-contact", du

12. Pour les calculs utilisant 4096 cœurs et la méthode de partitionnement "SFC-contact", la proportion d'empreinte mémoire pour l'étape de résolution du système linéaire descend à 9 % avec HypreILU comme pré-conditionneur et 17 % avec HypreBoomerAMG comme pré-conditionneur.

pré-conditionneur HypreILU et de l'ordre 3 pour la transformation super-paramétrique des éléments permet de minimiser la mémoire allouée à l'étape de résolution du système linéaire mais aussi à l'ensemble de la simulation. Ainsi, pour réduire les coûts en espace mémoire, une solution consiste à réduire l'ordre de la transformation super-paramétrique des éléments.

Il faut de plus s'assurer de la précision du résultat de la simulation éléments finis obtenue. C'est pour cela que, à travers le tableau 3.7 comparatif, les résultats éléments finis obtenus sont analysés pour les maillages d'entrée à transformation super-paramétrique d'ordres 3 et 6. Ici, les grandeurs d'intérêt, qui sont explicitées dans le tableau, sont :

- la taille du système linéaire (avec les degrés de liberté conformes) à résoudre lors de la dernière itération AMR (noté ici U par abus de notation) ;
- le nombre d'éléments finis du maillage obtenu à la fin du calcul N_E ;
- l'erreur estimée globale relative en norme énergie $\gamma_\Omega = \xi_\Omega / \omega_\Omega$;
- la mesure de la zone encore marquée pour raffinement à la fin du processus AMR, évaluée par le ratio $\eta = \mu\left(\Omega_{\mathcal{M}}\right) / \mu\left(\Omega^h\right)$.

Partitionnement / Pré-conditionneur	Ordre de la transformation	taille de **U** ($\times 10^6$)	N_E	γ_Ω	η
"pie chart" / ILU	3	18,262296	6187024	0,924102	15,931785
	6	18,261960	6186912	-	15,931783
"pie chart" / BoomerAMG (ST = 0,7)	3	18,262296	6187024	-	15,931785
	6	18,261960	6186912	-	15,931783
"SFC-contact" / ILU	3	18,262296	6187024	-	15,931785
	6	18,261960	6186912	-	15,931783
"SFC-contact" / BoomerAMG (ST = 0,7)	3	18,262296	6187024	-	15,931785
	6	18,261960	6186912	-	15,931783

TABLEAU 3.7. – Comparaison des solutions éléments finis pour différents ordres de transformation - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG - Le symbole "-" indique que la valeur est identique à la dernière entrée dans la colonne.

Les résultats obtenus montrent que le passage d'un maillage à transformation super-paramétrique d'ordre 6 à un autre d'ordre 3 a une faible influence sur la solution éléments finis obtenue pour le cas test 3D du contact Hertzien. En effet, l'écart relatif observé est de $1,8 \times 10^{-3}\%$ pour U et N_E, de $1,3 \times 10^{-5}\%$ pour η et inférieur à $10^{-5}\%$ pour γ_Ω. Il est donc possible de choisir un maillage d'entrée à transformation super-paramétrique d'ordre 3 pour notre étude du contact Hertzien. Cependant, ces résultats sont fortement dépendants des géométries et de la courbure des surfaces mises en jeu. De fait, utiliser un maillage d'entrée à transformation super-paramétrique d'ordre élevé peut s'avérer judicieux dans le cas d'un contact entre deux surfaces rugueuses ou présentant une courbure complexe à définir (elliptique par exemple). En effet, la géométrie de ce type de surfaces n'est pas caractérisable avec des fonctions d'ordre peu élevé (affines, quadratiques, etc.), donc est le plus souvent impossible à définir avec exactitude. Ainsi, augmenter l'ordre permettrait de réduire l'erreur de définition de discrétisation de la surface.

3.7. Extension à des problèmes de grande taille

Pouvoir effectuer des simulations sur de larges domaines de calcul comportant un grand nombre d'inconnues représente un enjeu scientifique majeur. Cette section s'attache par conséquent à étudier les performances parallèles de notre stratégie sur ce type de simulation. Ici, en prenant en considération les observations réalisées à partir des résultats obtenus dans les sections 3.5 et 3.6, le système linéaire est résolu avec le solveur HyprePCG, pré-conditionné avec HypreILU.

3.7.1. Contact Hertzien

Le cas du contact Hertzien 3D est considéré et le maillage fourni à l'environnement logiciel MFEM est à transformation super-paramétrique d'ordre 3. Le maillage initial est construit dans le mailleur Gmsh à partir du maillage illustré dans la figure 2.2a (en vue de coupe, visible en page 67) et auquel deux niveaux de raffinement uniforme ont été appliqués (par le mailleur) de manière à ce que chaque processus ait au moins un élément à gérer. Le maillage, présenté dans la figure 3.21 (en vue de coupe), ainsi obtenu contient 18020 nœuds et 16384 éléments finis et est adapté à la géométrie.

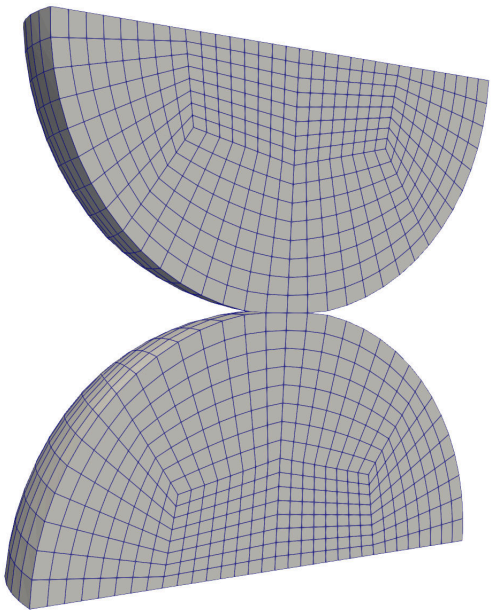
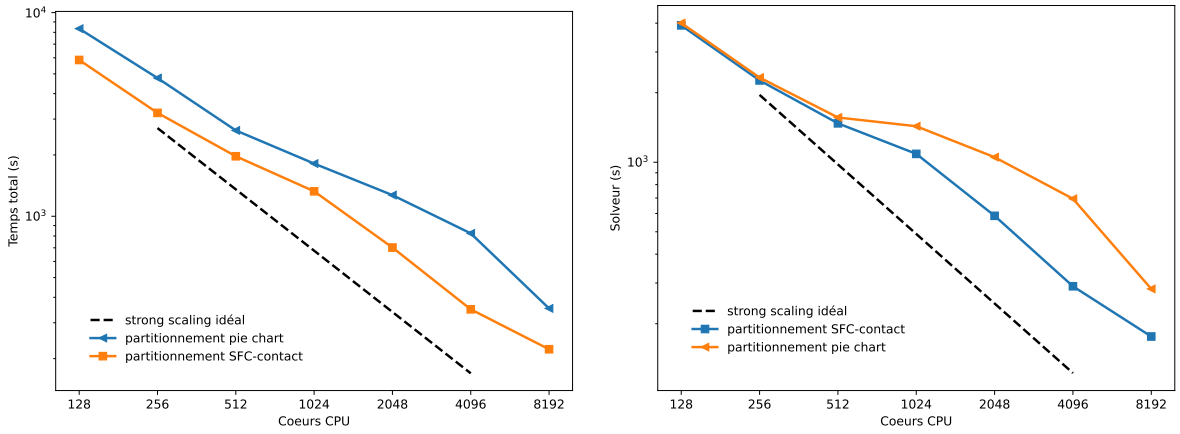


FIGURE 3.21. – Problème du contact Hertzien 3D : vue de coupe du maillage initial - ordre de la transformation super-paramétrique de 3.

Dans cette étude, la combinaison AMR 1 est considérée avec $e_\Omega = 0,5\%$; ce qui conduit à une simulation dont le nombre de degrés de liberté du système linéaire à résoudre (n'incluant pas les degrés de liberté issus des nœuds non conformes) vaut 146,9 millions. Les méthodes de partitionnement analysées ici sont les stratégies "pie chart" et "SFC-contact" et les résultats (temps total du calcul et temps de résolution du système linéaire) en strong scaling obtenus sont présentés en figures 3.22a pour les temps totaux et 3.22b en ce qui concernant les temps de résolution du système linéaire.

En outre, l'utilisation d'un maillage à transformation super-paramétrique d'ordre moins élevé que dans la section 3.6 nous permet de fixer un nombre de cœurs par processus moins



(a) (b)

FIGURE 3.22. – Résultats des performances parallèles de l'algorithme en (a) et des performances du solveur en (b) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 0,5\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : 146, 9 millions - ordre de la transformation super-paramétrique de 3.

important car nous avons vu en section 3.6.3 que les coûts en empreinte mémoire y sont plus réduits. Ainsi, le nombre de cœurs par processus est choisi égal à 2.

Les calculs ont convergé après 7 itérations AMR donc 6 raffinements du maillage. Les calculs ont pu atteindre jusqu'à près de 245 millions de degrés de liberté avec la méthode de partitionnement "pie chart" et 190 millions de degrés de liberté avec l'approche de partitionnement "SFC-contact" en incluant les degrés de liberté non-conformes et ceux fantômes, pour un calcul utilisant 8192 cœurs. L'empreinte mémoire maximum des calculs (obtenue pour le calcul utilisant 8192 cœurs) vaut 15 328 GB pour la méthode de partitionnement "pie chart" et 4421 pour l'approche de partitionnement "SFC-contact". Ce facteur 3,5 observé entre ces deux empreintes mémoires est cohérent avec les résultats analysés dans la section précédente, notamment par le biais de la figure 3.20. En effet, il a été observé, d'une part, que l'application de l'algorithme de partitionnement "pie chart" est plus coûteux en empreinte mémoire pour un grand nombre de cœurs CPU (à partir de 1024 à 2048 cœurs CPU selon le pré-conditionneur utilisé) et que, d'autre part, cet écart en empreinte mémoire entre les deux stratégies de partitionnement s'accroît avec le nombre de processeurs.

La figure 3.22a montrent que la stratégie développée couplant AMR et contact se révèle très fortement scalable sur cette simulation y compris pour plus de 8000 cœurs CPU. De plus, ces résultats de scalabilité très satisfaisants se confirment aussi lorsque nous analysons les performances de l'étape de résolution, illustrées dans la figure 3.22b, car présentant également une très bonne scalabilité pour plus de 8000 cœurs CPU. Notre stratégie parallèle est donc pleinement validée sur le cas du contact Hertzien.

Remarque : la perte de performance parallèle de l'étape de résolution observée entre 256 et 512 cœurs CPU sur la figure 3.22b (et dans une moindre mesure entre 128 et 256 unités de calcul) est liée au fait que le nombre de cœurs par processus a été augmenté (et vaut en l'occurrence 4, 8 et 16 pour les calculs sur 512, 256 et 128 cœurs CPU respectivement) afin d'allouer suffisamment de mémoire pour que les calculs puissent aller à leur terme. Le

nombre de nœuds de calcul, la bande passante mémoire et la mémoire cache L3 cumulées associées sont donc constants pour les trois premiers points de chaque courbe.

3.7.2. Cas d'indentation

Après avoir vérifié les performances de notre stratégie en matière de convergence vis-à-vis de résultats de référence mais aussi de scalabilité forte sur le cas du contact Hertzien, s'assurer de sa validité et de ses performances vis-à-vis de géométries différentes est également important. Ainsi, le cas d'un contact entre un cube de côté H et un pavé droit de dimensions $L \times L \times H$ (avec $L = 1,5$ m et $H = 0,5$ m), et de propriétés matériaux différentes est considéré dans cette section.

3.7.2.1. Définition du problème

Il s'agit ici d'étudier un cas test représentatif de l'indentation d'un matériau. Les maillages choisis sont concordants dans la zone de contact potentielle, de manière à ce que l'appariement nœud-à-nœud s'applique correctement. Sur ce cas d'étude, des conditions aux limites de Dirichlet d'intensité $u_D = \frac{\delta_0}{2} + \alpha H$, avec $\delta_0 = 1$ m est le jeu minimal initial entre les solides et α un coefficient (strictement positif), sont appliquées aux surfaces supérieure du cube et inférieure du pavé droit de manière à ce qu'ils rentrent en contact. Le cube est choisi plus rigide que le pavé droit. Les propriétés mécaniques du cube sont $\mathrm{E}_1 = 210$ GPa et $\nu_1 = 0,3$ tandis que celles du pavé droit sont $\mathrm{E}_2 = 2,1$ GPa et $\nu_2 = 0,3$. Ici, la combinaison AMR 1 est considérée et e_Ω est fixé à 4% dans un premier temps. Enfin, le coefficient de pénalisation est, dans un premier temps, choisi constant et égal à $10^4 \mathrm{E}_1$, en nous appuyant sur l'étude paramétrique réalisée en section 2.3.1. Un exemple du contact considéré est visible en figure 3.23. En particulier, les figures 3.23a et 3.23b ainsi que les figures 3.23c et 3.23d représentent des vues de coupe du cas test pour des simulations avec $\alpha = 0,008$ et $\alpha = 0,03$ respectivement. Pour l'exemple présenté dans la figure, 128 cœurs du supercalculateur Topaze ont été utilisés et le système linéaire final obtenu comporte 1,73 millions de ddls (en ne dénombrant que les degrés de liberté conformes) pour la simulation avec $\alpha = 0,008$ et 1,87 millions de ddls pour celle avec $\alpha = 0,03$.

De plus, les calculs ont convergé après 7 itérations AMR (donc 6 raffinements du maillage) et ont atteint 2,57 millions de degrés de liberté en incluant les degrés de liberté non-conformes et ceux fantômes. L'empreinte mémoire du calcul vaut ici (pour un calcul avec 128 cœurs utilisés) 41 GB. Cette simulation s'est avérée très peu coûteuse en empreinte mémoire par rapport aux simulations présentées dans les section précédentes. Cela est dû au fait d'avoir appliqué des éléments finis à transformation isoparamétriques au maillage.

Tout d'abord, les résultats obtenus dans la figure 3.23 montrent que, comme lors de la section 2.3.3, le solide qui se déforme le plus est celui qui subit un raffinement plus important.

3.7.2.2. Strong scaling pour le partitionnement "SFC-contact"

Il est d'intérêt d'évaluer l'efficacité parallèle de notre stratégie sur ce cas d'étude pour s'assurer que les performances très satisfaisantes, observées dans les sections 3.6.1 et 3.7, se conservent lorsque les géométries des solides en contact changent. Ici, le fait de considérer des solides à bords droits permet :

— de ne pas avoir à utiliser d'éléments finis à transformation super-paramétrique ;

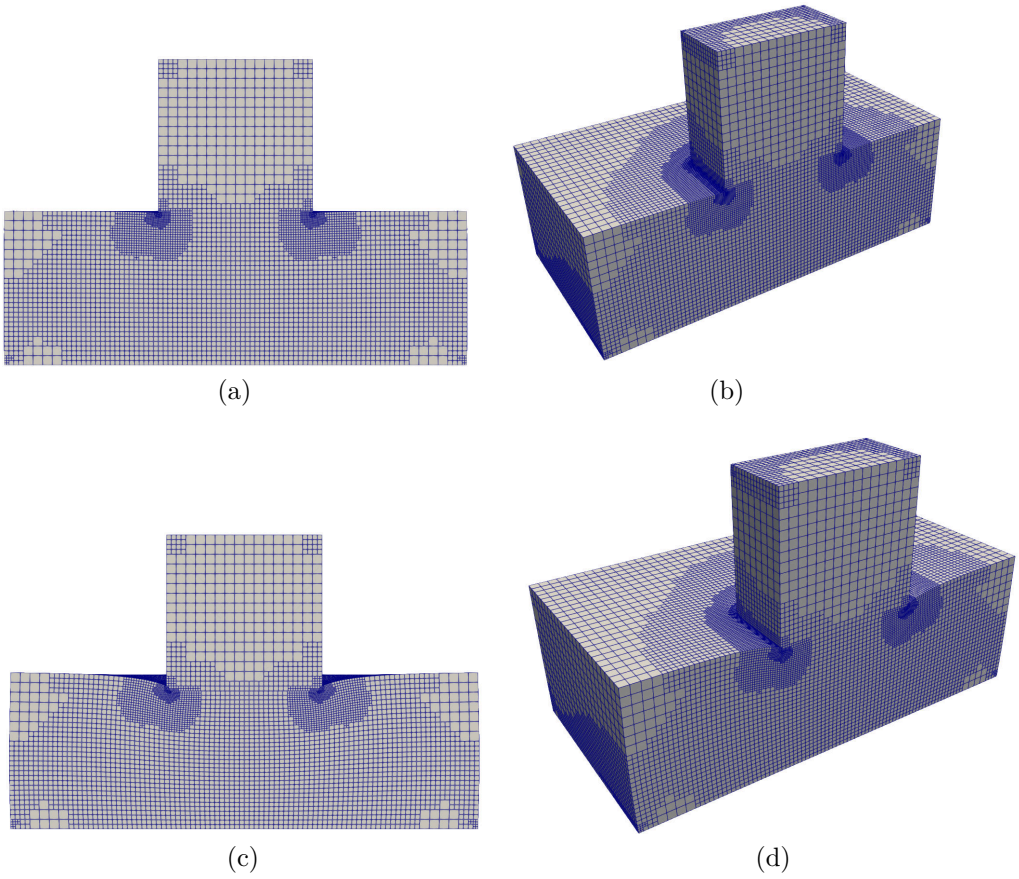


FIGURE 3.23. – Contact d'indentation pour des simulations avec $\alpha = 0,008$, en (a) et (b), et $\alpha = 0,03$, en (c) et (d), en vue de coupe - combinaison AMR 1 ($e_\Omega = 4\%$).

— de limiter les coûts en espace mémoire ;

— de traiter des problèmes de plus grandes tailles, avec un plus grand nombre de degrés de liberté.

Comme, pour ce cas test, l'intérêt principal est de traiter des simulations de plus grande taille, l'analyse de la qualité de solution physique calculée ne sera pas réalisée ici.

Pour ce cas test, la combinaison AMR 1 avec $e_\Omega = 0,9\%$ (nombre de degrés de liberté du maillage conforme final : 295,9 millions) est appliquée et l'étude de strong scaling est présentée dans les figures 3.25a et 3.25b. Les calculs ont été réalisés en utilisant deux cœurs par processus.

Le maillage initial uniforme, présenté dans la figure 3.24 (en vue de coupe), est construit dans le mailleur Gmsh de manière à ce que chaque processus ait au moins un élément à gérer. Le maillage ainsi obtenu contient 45733 nœuds et 40960 éléments finis.

Les calculs ont convergé après 9 itérations AMR donc 8 raffinements du maillage. De plus, les calculs ont pu atteindre jusqu'à près de 370 millions de degrés de liberté en incluant les degrés de liberté non-conformes et ceux fantômes (pour le calcul sur 8192 processeurs). L'empreinte mémoire maximum du calcul (obtenue pour le calcul utilisant 8192 cœurs) vaut 8034 GB. Dans cette simulation, les paramètres matériaux E_1, ν_1, E_2 et ν_2 sont inchangés par rapport à ceux définis à la section précédente et nous fixons

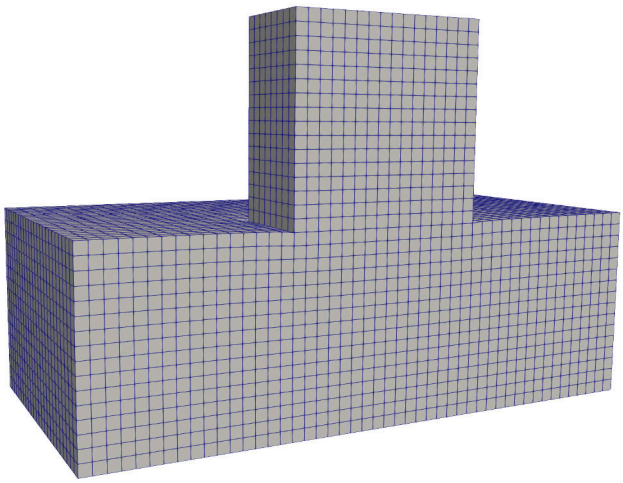


FIGURE 3.24. – Cas test de l'indentation 3D : vue de coupe du maillage initial - éléments isoparamétriques.

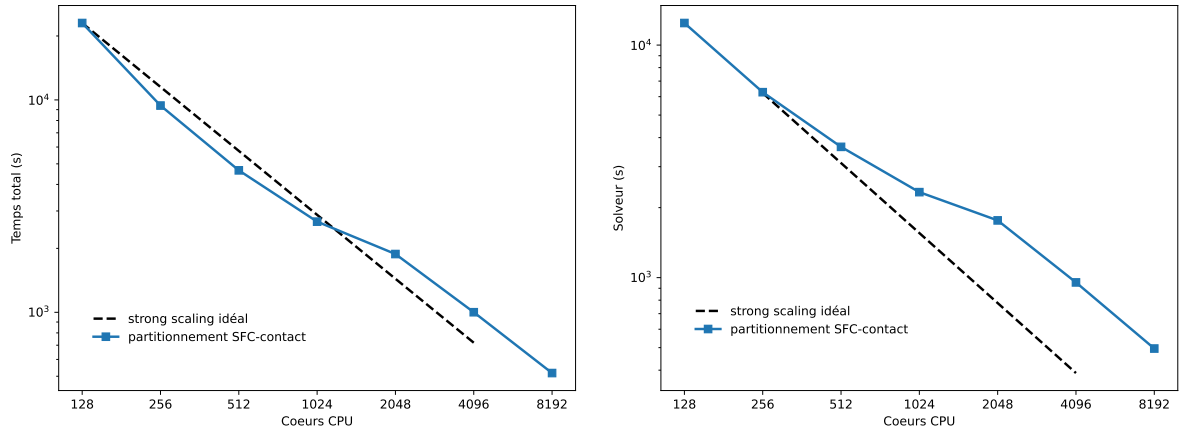
$k_N = 10^4 \mathrm{E}_1$ et $\alpha = 0,008$.

Les temps des principales fonctions suivantes du programme qui font l'objet d'un profilage sont toujours les suivantes :

- assemblage de la matrice de rigidité éléments finis (fonction MFEM `Assemble`) ;
- formation du système linéaire sans prise en compte des termes de contact (fonction MFEM `FormLinearSystem`, équation (2.26)) ;
- ajout des termes de contact et résolution du système matriciel (application du solveur issu de la classe MFEM `PenaltyPCGSolver`, équation (2.6)) ;
- raffinement des éléments (approche AMR estimate-mark-refine, fonction MFEM `Apply`) ;
- équilibrage et redistribution de l'ensemble des éléments finis du maillage après raffinement (incluant la fonction MFEM `Rebalance`) .

Les résultats obtenus montrent une très bonne scalabilité parallèle de notre stratégie pour plus de 8000 cœurs CPU lorsque nous l'appliquons à ce cas d'étude. Par ailleurs, il est observé que les performances parallèles se révèlent meilleures qu'idéalement pour 256, 512 et 1024 cœurs CPU. Cela est dû au comportement superlinéaire observé sur le comportement parallèle de certaines fonctions. C'est effectivement le cas ici pour les étapes de formation du système linéaire, de raffinement et de redistribution de l'ensemble des éléments finis du maillage après raffinement (fonctions MFEM `FormLinearSystem`, `ApplyImpl` et `Rebalance` respectivement), comme il est possible de s'en rendre compte sur la figure 3.26. Cette super linéarité est observée ici vraisemblablement en raison d'effet cache L2 dans les calculs. En effet, ici, le nombre de cœurs par processus a été augmenté et vaut en l'occurrence ici 4, 8, 16 et 32 pour les calculs sur 1024, 512, 256 et 128 cœurs CPU respectivement. Le nombre de nœuds de calcul est donc constant dans ces quatre configurations. Cela augmente par conséquent la taille mémoire cumulée du cache L2 accessible. Par ailleurs, il est observé sur la figure 3.25b que le solveur perd en performance parallèle entre 1024 et 2048 cœurs de calcul. Une étude de profilage plus approfondie serait nécessaire afin d'affiner l'analyse.

Enfin, la comparaison des figures 3.25b et 3.26 mais aussi des tableaux 3.8 et 3.9 (colonne de droite) permet de remarquer que l'étape de résolution du système linéaire est un peu



(a) (b)

FIGURE 3.25. – Résultats des performances parallèles de l'algorithme en (a) et des performances du solveur en (b) - Indentation 3D - $\alpha = 0,008$ et combinaison AMR 1 ($e_\Omega = 0,9\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : 295,9 millions - ordre transformation 1.

moins prépondérante (en matière de temps passé dans la fonction) sur les autres fonctions de l'algorithme AMR-contact à faibles nombres de cœurs CPU (128 à 512 notamment) par rapport aux calcul pour un plus grand nombre de cœurs (2048 à 8192 en particulier). Ainsi, si pour un faible nombre de cœurs CPU les autres fonctions de l'algorithme AMR-contact peuvent influer notablement sur le comportement parallèle de la stratégie (voir paragraphe précédent), ce dernier devient quasi exclusivement piloté par les performances parallèles du solveur pour un plus grand nombre d'unités de calcul. Cette observation est confirmée par les résultats reportés sur les tableaux 3.8 et 3.9 qui sont présentés par la suite. Ce comportement global, à faible et haut nombre d'unités de calcul, explique ainsi l'allure de la courbe de strong scaling présentée dans la figure 3.25a.

De plus, des calculs comportant un maillage conforme final allant jusqu'à 400 millions de degrés de liberté (combinaison AMR 1 avec $e_\Omega = 0,8\%$, nombre de degrés de liberté du maillage conforme final : 403,5 millions) ont été réalisés. Les calculs ont convergé après 9 itérations AMR (donc 8 raffinements, sachant que le maillage initial fourni en entrée est celui illustré en figure 3.24). De plus, les calculs ont pu atteindre près de 490 millions de degrés de liberté en incluant les degrés de liberté non-conformes et ceux fantômes (pour le calcul sur 8192 processeurs). L'empreinte mémoire maximum du calcul (obtenue pour le calcul utilisant 8192 cœurs) vaut 13189 GB. Cette valeur élevée d'empreinte mémoire est causée par le fait que le maillage final obtenu est composé d'un grand nombre d'éléments finis et que par conséquent les étapes de raffinement du maillage et de rééquilibrage de charge (fonctions MFEM `Apply` et `Rebalance`, voir la section 3.5.2) sont consommatrices en espace mémoire.

Par ailleurs, il est d'intérêt de connaître quelles sont les fonctions prépondérantes dans le calcul pour un grand nombre de cœurs de calcul utilisé afin d'optimiser les performances parallèles de la stratégie AMR-contact proposée. Ainsi, les tableaux 3.8 et 3.9 montrent les proportions (par rapport au temps total du calcul, en pourcentage) de temps passé dans les parties parties principales du code pour trois simulations différentes employant

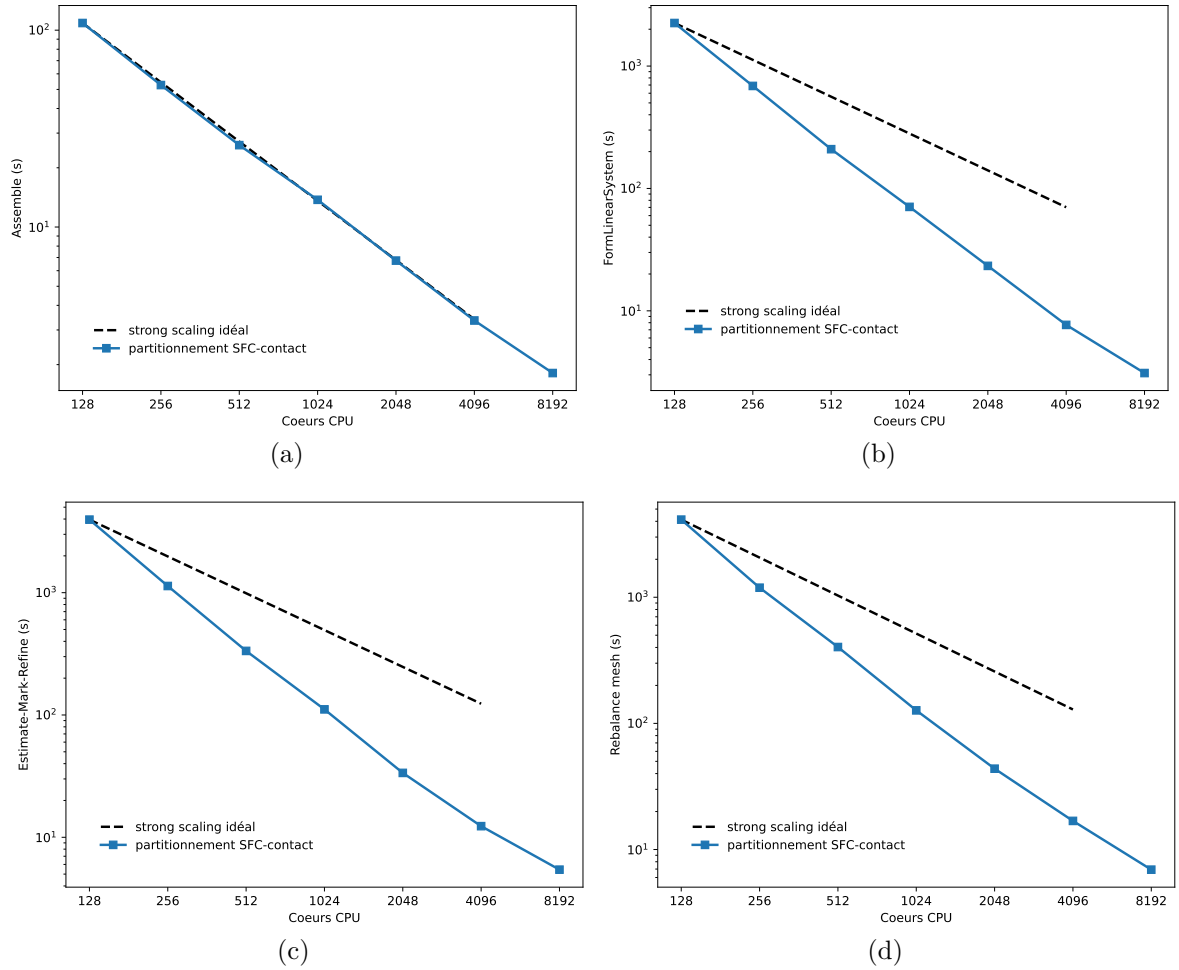


FIGURE 3.26. – Résultats des performances des étapes principales de l'algorithme (fonctions MFEM `Assemble` en (a), `FormLinearSystem` en (b), `Apply` pour l'application de l'approche estimate-mark-refine en (c) et `Rebalance` pour Rebalance mesh (rééquilibrage du maillage) en (d)) - Indentation 3D - $\alpha = 0,008$ et combinaison AMR 1 ($e_\Omega = 0,9\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : 295, 9 millions - ordre transformation 1.

chacune le même nombre de cœurs CPU : le contact Hertzien avec combinaison AMR 1 avec $e_\Omega = 1\%$ (ordre de la transformation 6 avec 4 cœurs par processus), avec $e_\Omega = 0,5\%$ (ordre de la transformation 3 avec 2 cœurs par processus) et le cas de l'indentation avec combinaison AMR 1 avec $e_\Omega = 0,9\%$ (ordre de la transformation 1 avec 2 cœurs par processus). Pour ces trois simulations, le tableau 3.8 décrit les résultats pour des calculs utilisant 256 cœurs CPU tandis que le tableau 3.9 les reporte pour des calculs sur 4096 processeurs. De plus, il est à noter que la méthode de partitionnement choisie est la même pour ces six calculs, et il s'agit de "SFC-contact".

Un point important observé en comparant ces deux tableaux est que l'étape de résolution du système linéaire devient largement prépondérante en matière de temps de calcul sur les autres parties de la simulation lorsque le maillage d'entrée est composé d'éléments finis à transformation isoparamétrique (pour le cas de l'indentation) ou super-paramétrique d'ordre "moins élevé" (par exemple 3 pour le cas du contact Hertzien)

% du temps total	Hertz ($e_\Omega = 1\%$, ordre de la transformation de 6, 4 cœurs par processus)	Hertz ($e_\Omega = 0,5\%$, ordre de la transformation de 3, 2 cœurs par processus)	Indentation ($e_\Omega = 0,9\%$, ordre de la transformation de 1, 2 cœurs par processus)
Assemblage	44,3	9,43	0,56
Formation du système linéaire (sans le contact)	15,8	4,60	7,31
Ajout du contact et résolution du système matriciel	11,0	70,2	66,7
Estimate-Mark-Refine	40,2	8,86	12,1
Rééquilibrage de charge	1,24	4,54	12,7

TABLEAU 3.8. – Proportion des temps passés dans les différentes parties du code pour trois simulations différentes - solveur HyprePCG ; pré-conditionneur HypreILU - méthode de partitionnement "SFC-contact" - calculs sur 256 cœurs CPU.

% du temps total	Hertz ($e_\Omega = 1\%$, ordre de la transformation de 6, 4 cœurs par processus)	Hertz ($e_\Omega = 0,5\%$, ordre de la transformation de 3, 2 cœurs par processus)	Indentation ($e_\Omega = 0,9\%$, ordre de la transformation de 1, 2 cœurs par processus)
Assemblage	38,4	8,73	0,33
Formation du système linéaire (sans le contact)	19,8	1,82	0,77
Ajout du contact et résolution du système matriciel	12,5	83,5	95,3
Estimate-Mark-Refine	29,2	6,03	1,23
Rééquilibrage de charge	1,43	1,34	1,68

TABLEAU 3.9. – Proportion des temps passés dans les différentes parties du code pour trois simulations différentes - solveur HyprePCG ; pré-conditionneur HypreILU - méthode de partitionnement "SFC-contact" - calculs sur 4096 cœurs CPU.

et lorsque le nombre de cœurs de calcul utilisés devient important. Au contraire, lorsque l'ordre de la transformation super-paramétrique est plus élevée (et vaut 6 par exemple), l'étape de résolution du système linéaire est moins influente sur les performance parallèles

de la simulation et ce sont les fonctions s'occupant de l'assemblage matriciel ou encore de l'application de l'approche AMR estimate-mark-refine qui régissent davantage le comportement parallèle de l'algorithme AMR-contact. En outre, l'étude de ces tableaux permet de remarquer que la proportion de temps passé dans les étapes de formation du système linéaire (sans le contact), d'application de la stratégie AMR ou encore de redistribution des éléments du maillage sur les processus tend à diminuer avec le nombre de cœurs CPU utilisés et l'ordre de la transformation super-paramétrique du maillage. Le phénomène contraire est observé pour l'étape de résolution du système matriciel. Ainsi, lorsqu'un maillage avec des éléments finis à transformation super-paramétrique d'ordre "faible" (de l'ordre de 2 ou 3 comme pour le cas Hertz $(0,5\%)$) voire isoparamétrique est choisi, les performances parallèles du solveur deviennent beaucoup plus importantes pour obtenir une scalabilité satisfaisante de l'algorithme. Concernant les calculs utilisant un maillage avec des éléments super-paramétrique d'ordre élevé (colonne de gauche dans les tableaux), les proportions semblent rester globalement les mêmes quel que soit le nombre de processeurs.

En conclusion, les résultats vus dans cette section (figure 3.25b en particulier), mais aussi en section 3.7 (figure 3.22b notamment) montrent que la combinaison HyprePCG comme solveur, HypreILU comme pré-conditionneur remplit parfaitement la mission de résoudre le problème de contact de manière hautement scalable avec les méthodes de partitionnement "pie chart" et "SFC-contact". De plus, le fait de choisir des éléments finis à transformation isoparamétriques a permis d'effectuer des simulations très volumineuses comportant 300 à 400 millions de ddls. Cela a également permis d'observer que les performances parallèles du solveur influent encore plus fortement sur la scalabilité globale de la stratégie que lorsque des éléments finis à transformation super-paramétrique sont utilisés.

3.8. Conclusion

Dans ce chapitre, trois stratégies de partitionnement ont été introduites et étudiées : les algorithmes "equidistributed", "pie chart" et "SFC-contact". Ces trois méthodes de redistribution des éléments se caractérisent, tout d'abord, par le fait que les éléments de contact en vis-à-vis sont sur les mêmes processus durant les itérations AMR. Ces stratégies ont pu être évaluées sur deux cas d'études différents, de grandes tailles en allant jusqu'à 400 millions de degrés de liberté conformes, avec des solides en contact de mêmes ou de différentes propriétés matériaux et pour des simulations allant jusqu'à 8192 cœurs CPU.

Les simulations considérées ont mis en exergue les performances parallèles très satisfaisantes des algorithmes de partitionnement proposés pour résoudre nos problèmes en mécanique du contact par la méthode de pénalisation avec AMR. Tout d'abord, la stratégie "equidistributed" s'est révélée scalable pour des calculs allant jusqu'à 512 processeurs mais présente des limites au-delà, en particulier en raison d'un volume de communication important. Par contre, les stratégies "pie chart" et "SFC-contact" se sont avérées très performantes pour plus de 8000 cœurs CPU et des simulations comportant près de 400 millions de degrés de liberté. Ces résultats représentent une réelle avancée en ce qui concerne la parallélisation de problèmes de contact puisque les études récemment proposées n'ont pas excédé 2400 cœurs de calculs et 250 millions d'inconnues, et se sont le plus souvent restreints à des contacts rigide-déformable et sans AMR (voir le chapitre 1, sections 1.7.3 et 1.7.4).

Les résultats obtenus ont permis de mettre en évidence certaines propriétés intrinsèques à nos simulations. Tout d'abord, certaines fonctions de MFEM utilisées dans notre algorithme entraînent de fortes hausses de consommation mémoire et nous obligent à utiliser plusieurs cœurs par processus. Cette consommation mémoire a pu néanmoins être réduite lorsque nous diminuons l'ordre de la transformation des éléments finis du maillage. Ceci a eu pour conséquence de diminuer les temps de calcul mais aussi de rendre prépondérantes les performances du solveur pour optimiser la stratégie globale.

Afin de vérifier les performances de la classe MFEM employée pour résoudre le système matriciel, la combinaison du solveur HyprePCG avec les pré-conditionneur HypreILU ou HypreBoomerAMG est analysée. L'utilisation de HypreBoomerAMG a permis de réduire fortement le nombre d'itérations du solveur mais a présenté à la fois des problèmes de matrice symétrique définie positive lorsque la stratégie de partitionnement "equidistributed" est appliquée mais aussi quelques pertes de performances parallèles au-delà de 4000 processeurs avec les deux autres méthodes de partitionnement. Ne présentant ni de perte majeure de performance parallèle pour plus de 4000 tâches MPI (avec "pie chart" et "SFC-contact") ni de problématique de matrice symétrique définie positive, la combinaison du solveur itératif HyprePCG avec le pré-conditionneur HypreILU a été identifiée comme la plus efficace dans les configurations considérées.

Il a également été observé que la scalabilité des simulations est légèrement moins intéressante mais aussi que le nombre d'itérations du solveur tend à croître fortement avec le nombre de processeurs lorsque la méthode de partitionnement "pie chart" est appliquée plutôt que "SFC-contact".

La méthode de partitionnement "pie chart" peut être adaptée à faibles coûts à d'autres géométries par le biais de son paramétrage (nombre et forme des sous-ensembles). En effet, la seule modification majeure dans l'algorithme de mise en œuvre de la méthode est lors de la première étape et en particulier lors de la création des sous-ensembles d'éléments voisins. Il s'agira donc, pour chaque étude, de définir les formes choisies pour ces sous-ensembles et de formuler une routine d'une dizaine de lignes (une boucle FOR ou WHILE incluant quelques structures conditionnelles IF, ELSE IF, ELSE) pour l'adapter aux géométries considérées. Par ailleurs, une extension de la méthode de partitionnement "SFC-contact" à d'autres courbes SFC (Peano, Morton, etc.) est possible mais les courbes en question ne sont pas implémentées pour le moment dans le code source de l'environnement logiciel MFEM et il n'est pas certain que l'implémentation de ces courbes apportent une plus value majeure sur les performances parallèle de la stratégie. Leur implémentation et analyse pourrait néanmoins être une tâche future.

Pour la gestion d'un maillage initial structuré et conforme en mémoire distribuée, le code source de la bibliothèque MFEM a été enrichi de la fonction de génération du partitionnement ainsi que d'une fonction qui permet de récupérer les informations sur les "super-éléments". Pour la gestion de maillages non-conformes dans un cadre parallèle, le code source de la bibliothèque MFEM a été enrichi de deux fonctions afin de récupérer les informations sur les "super-éléments". De plus, la fonction de redistribution des éléments a été adaptée à la gestion des éléments de contact. Enfin, concernant l'étape de résolution du système linéaire, certains solveurs et pré-conditionneurs ont été ajoutés dans le code source de l'environnement logiciel MFEM et leurs paramètres ont pu être adaptés en conséquence.

Les stratégies de redistribution des éléments du maillage peuvent être facilement transposables à d'autres environnements logiciels éléments finis. Néanmoins, le principal effort à fournir pour cette transposition serait d'adapter les objets spécifiques à MFEM et leurs

fonctions spécifiques dans les environnements logiciels cibles.

Par ailleurs, une perspective envisageable à cette proposition de stratégie parallèle pourrait être d'analyser ses performances au-delà de 16000 cœurs de calcul sur des simulations atteignant un milliard de degrés de liberté. Une autre piste d'étude pourrait être d'étendre la stratégie contact-AMR-HPC proposée dans ces deux derniers chapitres à des cas test comportant plusieurs zones de contact en simultanée.

Enfin, l'extension de la stratégie HPC-AMR-contact proposée à des simulations mettant en jeu des maillages non concordants via la mise en œuvre d'un algorithme parallèle d'appariement nœud-à-surface est l'objet du chapitre 4. Cela doit permettre ainsi de pouvoir traiter des cas 3D plus variés. Cette extension constitue un apport réel par rapport aux solutions proposées jusqu'alors puisque, à notre connaissance, seuls quelques travaux se sont concentrés sur l'implémentation d'une stratégie parallèle pour résoudre un problème en mécanique du contact entre maillages non concordants. De plus ces travaux se limitent à des discrétisations uniformes des solides.

Chapitre 4

Extension de la stratégie parallèle à un contact nœud-à-surface

Dans le chapitre 2, la stratégie de résolution du contact par pénalisation et raffinement adaptatif de maillage est couplée à un algorithme d'appariement des nœuds situés sur les bords de contact de type nœud-à-nœud. La parallélisation de cette stratégie a été présentée dans le chapitre 3. Le présent chapitre traite de l'extension de la stratégie parallèle globale au cas où l'algorithme d'appariement des nœuds situés sur les bords de contact est désormais de type nœud-à-surface afin de pouvoir traiter des problèmes avec maillages non concordants. Cela passe par le choix d'une méthode de partitionnement du maillage qui permette de résoudre avec précision le problème de contact tout en équilibrant efficacement la charge du domaine d'étude. Tout d'abord, la question de l'enrichissement de notre stratégie, résolvant des problèmes de contact 3D déformable avec AMR, par l'implémentation d'un algorithme de type nœud-à-surface est abordée en section 4.1. Dans cette section, l'enjeu scientifique ainsi que la solution proposée est dans un premier temps présentée, puis validée sur le cas du contact Hertzien. Dans un second temps, la question de sa parallélisation efficace est traitée dans la section 4.2. Ensuite, les performances parallèles de la solution proposée sont analysées.

Sommaire

4.1. Algorithme d'appariement nœud-à-surface en 3D avec AMR 131
4.1.1. Traitement algorithmique de l'appariement nœud-à-surface 132
4.1.2. Validation de la solution : comparaison avec la solution analytique de Hertz 133
4.2. Parallélisation de la stratégie 137
4.2.1. Extension dans un cadre parallèle : stratégies d'accès aux données . 138
4.2.2. Partitionnement dérivé du partitionnement "pie chart" 141
4.2.3. Analyse de la scalabilité et du pré-conditionnement 142
4.3. Conclusion 149

4.1. Algorithme d'appariement nœud-à-surface en 3D avec AMR

Mettre en œuvre un algorithme d'appariement nœud-à-surface pour détecter les relations de contact permet l'utilisation de maillages non concordants (voir la figure 4.2b) et ainsi traiter un plus large panel de géométries de problèmes en mécanique du contact qu'en considérant une approche de type nœud-à-nœud. En outre, mettre en place cette stratégie d'appariement permettra, à terme, de considérer des solides ayant de grandes déformations au niveau de la zone de contact.

Dans l'environnement logiciel MFEM, aucune procédure d'appariement de type nœud-à-surface n'est disponible. Ainsi, il s'agit de proposer ici une stratégie adaptée à son implémentation dans MFEM, en conservant l'approche AMR estimate-mark-refine déjà décrite.

4.1.1. Traitement algorithmique de l'appariement nœud-à-surface

Dans nos travaux, une formulation de type maître-esclave est considérée pour l'appariement des nœuds en contact potentiel.

L'algorithme d'appariement des nœuds situés sur les bords de contact implémenté se décompose en plusieurs étapes :

— La première étape consiste à déterminer et répertorier pour chaque élément fini situé sur la surface de contact maître la position moyenne de la surface maître en question ainsi que les numéros des nœuds associés à cette surface. Le calcul de la position moyenne de la surface maître a pour but de faciliter la recherche des surfaces maître/esclave les plus proches. En pratique, cette première étape consiste à remplir un premier tableau où chaque ligne correspond à un élément fini et les colonnes dénotent les numéros des nœuds maîtres de l'élément fini en question et leur position.

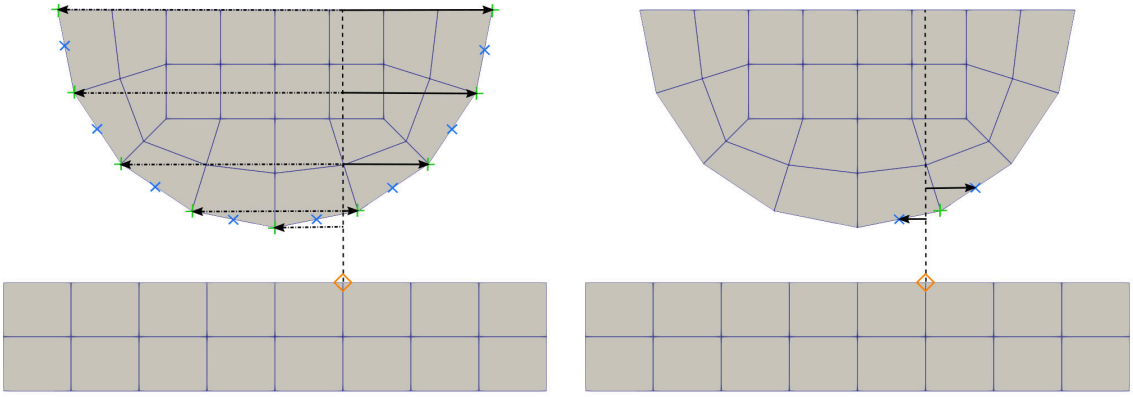
— La deuxième étape vise à déterminer, pour chaque nœud de la frontière de contact esclave, quel est le nœud maître le plus proche. Il est à noter que, l'appariement du contact n'étant pas la thématique de ces travaux de thèse, les détections du contact sont "simples" dans le sens où nous supposons que la normale de contact est connue *a priori*. Ainsi, pour déterminer le nœud maître le plus proche d'un nœud esclave, l'ensemble des nœuds de la surface maître est parcouru et leur distance (orthogonale à la normale de contact) au nœud esclave est calculée. Enfin, le nœud maître ayant la plus petite distance est retenu.

— Une fois que le nœud maître le plus proche d'un nœud esclave est trouvé, la valeur de la distance (orthogonale à la normale de contact) calculée est analysée. Si la distance est nulle, les nœuds sont en vis-à-vis et une procédure similaire à l'algorithme nœud-à-nœud est appliquée. Sinon, l'élément fini appartenant à la surface maître et situé le plus proche du nœud esclave selon la distance orthogonale à la normale de contact est recherché. Pour cela, l'ensemble des éléments finis maîtres ayant le nœud maître trouvé comme étant le plus proche du nœud esclave est parcouru. Pour chaque élément fini maître, sa distance au nœud esclave est calculée. Le calcul est ici effectué en considérant la position moyenne de la surface maître, déterminée lors de la première étape. Ainsi, l'élément fini maître ayant la plus petite distance devient l'élément maître appairé au nœud esclave.

— Enfin, pour chaque nœud de la surface esclave, les différents coefficients de la matrice d'appariement sont déterminés par projection de sa fonction de forme sur les nœuds maîtres associés à l'élément maître appairé.

Le principe de l'algorithme d'appariement nœud-à-surface est illustré sur un cas 2D par la figure 4.1. Sur cet exemple, la surface de contact potentiel du rectangle est définie comme surface esclave et celle du demi-disque comme surface maître pour des raisons de meilleure lisibilité de la figure. Sur cette figure, le nœud esclave repéré par un losange orange est à appairer aux nœuds marqués par une croix vert clair d'une surface maître dont la position moyenne est marquée par une croix bleue oblique. La normale au contact est ici orthogonale au rectangle (donc dans la direction du trait en pointillés). Sur la figure 4.1a,

les flèches noires, reliant le trait en pointillés aux croix vertes, traduisent ce qui est réalisé durant la deuxième étape de l'algorithme : les distances, orthogonales à la normale de contact, entre le nœud esclave et les nœuds maîtres sont calculées et la distance minimale est recherchée. La troisième étape peut être décrite de la même manière en traçant des flèches entre le trait en pointillés et les croix bleues obliques correspondant aux surfaces sur lesquelles se trouve le nœud maître le plus proche, comme décrit sur la figure 4.1b.



(a) (b)

FIGURE 4.1. – Illustration du principe de l'appariement nœud-à-surface sur un exemple 2D : deuxième étape en (a) et troisième étape en (b) - nœud esclave considéré représenté par un losange orange, nœuds maîtres par des croix vertes et position moyenne des surfaces par des croix bleues obliques.

Il est à noter que dans la stratégie proposée, l'algorithme d'appariement nœud-à-surface est appliqué intégralement à chaque itération AMR puisque le maillage et le partitionnement évoluent.

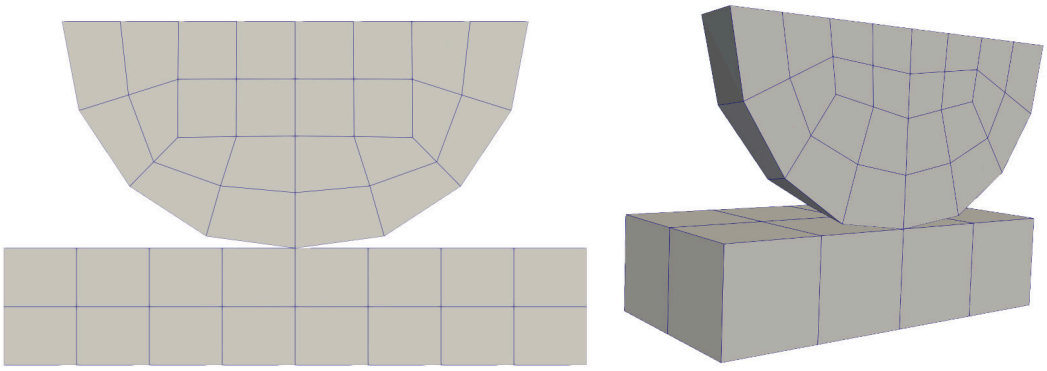
Par ailleurs, la mise en place d'une stratégie d'appariement de ce type implique que les nœuds maîtres peuvent intervenir dans plusieurs conditions de contact. Cela implique que la matrice d'appariement de contact potentiel $\tilde{\mathbf{B}}$ (voir la section 2.2.5) possède davantage de coefficients non nuls que si un appariement nœud-à-nœud était considéré. Ainsi, la matrice de raideur de contact $\hat{\mathbf{K}}_C$ (voir l'équation (2.28) en page 69) possède également davantage de coefficients non nuls qu'avec l'utilisation d'un appariement nœud-à-nœud.

4.1.2. Validation de la solution : comparaison avec la solution analytique de Hertz

4.1.2.1. Présentation des simulations

Cette section a pour but d'illustrer la pertinence de la stratégie proposée et son adéquation aux problèmes en mécanique du contact avec des bords courbes et des maillages non concordants. À cette fin, le problème de référence académique de Hertz est considéré en 2D et 3D. Comme pour la section 2.3, des expressions analytiques du champ de contraintes dans la zone de contact sont disponibles pour vérifier la précision de la stratégie numérique. Elle sont définies par l'équation (4.1) dont la forme est identique à la distribution de pression définie dans le chapitre 2. Le problème mécanique considéré est toujours un problème élasto-statique de contact (voir les équations (1.1)-(1.4)). Ici, les deux solides mis

en contact sont supposés être des matériaux élastiques isotropes dont le module d'Young présente un contraste de facteur 100 et les coefficients de Poisson sont identiques ($\nu = 0,3$ en l'occurrence). Le solide inférieur (rectangle en 2D, parallélépipède rectangle en 3D), de module d'Young $E = 210$ GPa, est alors choisi plus rigide que le supérieur (demi-disque en 2D, demi-sphère en 3D). Dans cette étude, le disque et la sphère étant les moins rigides, leur surface de contact potentielle est choisie comme surface esclave et la surface du rectangle ou du pavé en contact potentiel est considérée maître. Le rayon du disque et de la sphère est égal à $R = 2$ m, tandis que les dimensions du rectangle et du pavé droit sont $l \times H$ et $L \times L \times H$ respectivement avec $l = 5$ m, $L = 4$ m et $H = 1$ m. Le cas d'étude est similaire à celui introduit en section 2.3 : des conditions aux limites de Dirichlet de valeur $u_D = \dfrac{\delta_0}{2} + \alpha R$, où δ_0 est le jeu initial entre les solides et α un coefficient positif, sont imposées sur les bords plans orthogonaux à la normale du contact, donc le bord plan supérieur du demi-disque ou de la demi-sphère et inférieur du rectangle ou du pavé droit. Dans cette étude, $\alpha = 0,015$ pour le cas 2D et $\alpha = 0,007$ pour le cas 3D de manière à ce que la pression de contact maximale soit proche pour les deux cas. Le maillage initial est composé d'éléments quadrangulaires ou hexaédriques et est représenté sur les figures 4.2a et 4.2b.



(a) (b)

FIGURE 4.2. – Problème du contact Hertzien entre disque et rectangle 2D en (a) et entre sphère et pavé 3D en (b) : vue de coupe des maillages quadrangulaire et hexaédrique initiaux - ordre de la transformation super-paramétrique de 3.

Dans cette section, seule la combinaison AMR 1 est considérée : critère d'optimalité ZZ (2.13) et critère d'arrêt global (2.18).

De plus, la version de MFEM utilisée dans cette section est la version 4.5.2.

4.1.2.2. Comparaison avec la solution analytique de Hertz

Pour évaluer la précision de l'approche, il convient de comparer les résultats obtenus avec la théorie. Pour s'en tenir au cadre de la théorie de Hertz, les conditions présentées dans la section 2.3.2 doivent être satisfaites. Elles sont rappelées comme suit :

- les solides ont un comportement mécanique élastique, homogène et isotrope ;
- l'hypothèse des petites perturbations est appliquée ;
- les surfaces de contact sont petites comparativement aux solides entiers et les déformations à l'extérieur de la zone de contact sont négligeables par rapport à celles à l'intérieur ;

— pas de frottement.

Ainsi, pour cette analyse, les cas tests 2D et 3D sont considérés et les conditions limites de Dirichlet fixées ($u_D - \frac{\delta_0}{2} = 0,015\ R$ pour le 2D et $u_D - \frac{\delta_0}{2} = 0,007\ R$ pour le 3D) respectent les critères de l'hypothèse des petites perturbations.

Nous rappelons la distribution de pression de contact définie par la théorie de Hertz (qui s'applique de la même manière pour le cas 3D que pour celui 2D) :

$$p(r) = p_{\mathrm{O}}\sqrt{a^2 - r^2}/a\ ;\ r \leq a \tag{4.1}$$

où a est la distance caractéristique (ou demi-largeur) de contact, r la distance (positive) au centre O de la zone de contact et $p_{\mathrm{O}} = p(\mathrm{O})$ la pression de contact maximale au centre O. Pour calibrer notre solution de Hertz, deux maillages très raffinés ont été utilisés pour construire des solutions de référence, qui donnent $a = 0,302$ et $p_{\mathrm{O}} = 174$ MPa pour le cas 2D, et $a = 0,242$ et $p_{\mathrm{O}} = 176$ MPa pour le cas 3D. Ces valeurs sont ensuite utilisées pour définir la distribution de pression de Hertz, qui est comparée dans les figures 4.3 et 4.4 avec les pressions de contact obtenues via la solution par éléments finis $p = -\sigma_N = -(\boldsymbol{\sigma}\mathbf{n}) \cdot \mathbf{n}$ sur Γ_C. Dans ces figures, l'origine de l'abscisse de contact est choisie par le centre O de la zone de contact et, pour le cas 3D, les pressions de contact tracées correspondent à celles obtenues au niveau du plan de coupe défini par la figure 4.2b. Dans les figures 4.5 et 4.6 sont présentés les maillages obtenus pour une simulation avec $e_\Omega = 4\%$.

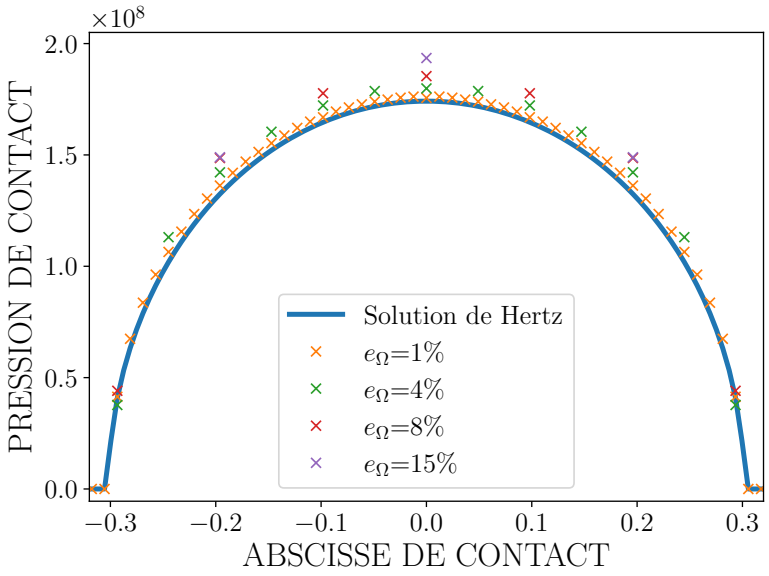


FIGURE 4.3. – Distribution de la pression de contact (en Pascal) sur la zone de contact (contact de Hertz 2D) - appariement nœud-à-surface - combinaison AMR 1.

Les résultats présentés dans les figures 4.3 et 4.5 pour le cas 2D mais aussi dans les figures 4.4 et 4.6 pour le cas 3D montrent clairement, d'une part, qu'en diminuant la précision spécifiée par l'utilisateur e_Ω pour l'application AMR, la solution par éléments finis est raffinée dans la zone de contact et converge vers la solution dérivée de la théorie de Hertz. D'autre part, la transformation super-paramétrique se révèle encore une fois efficace pour capturer la courbure de la frontière pendant les itérations d'AMR hiérarchique.

Par ailleurs, les figures 4.5 et 4.6 permettent de remarquer, tout d'abord que le raffinement du maillage est principalement concentré sur le solide qui se déforme le plus, en l'occurrence le demi-disque pour le cas 2D et la demi-sphère pour le cas 3D. Ces observations sont

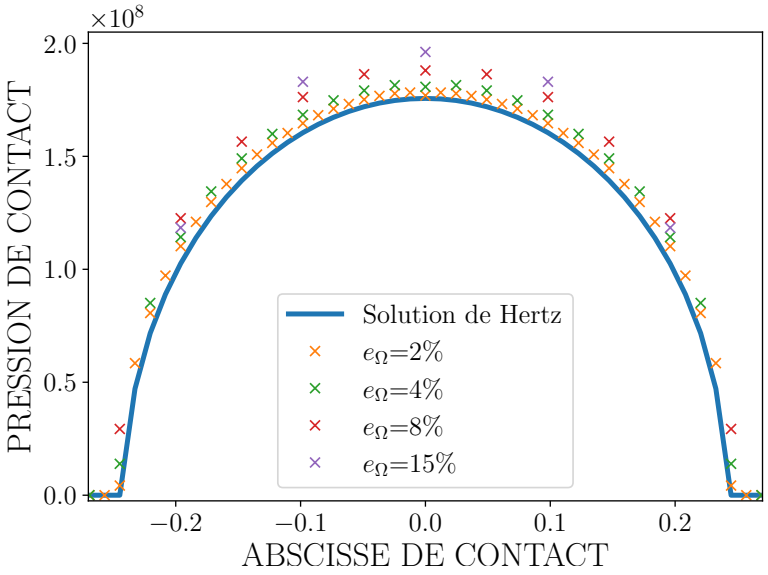


FIGURE 4.4. – Distribution de la pression de contact (en Pascal) sur la zone de contact (contact de Hertz 3D) - appariement nœud-à-surface - combinaison AMR 1.

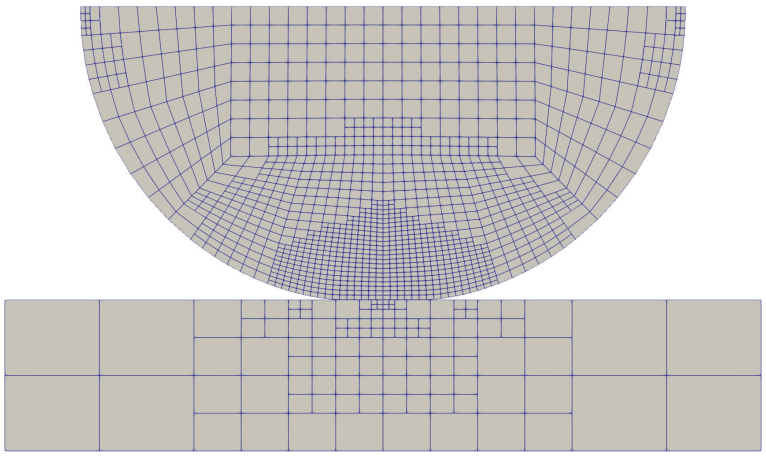
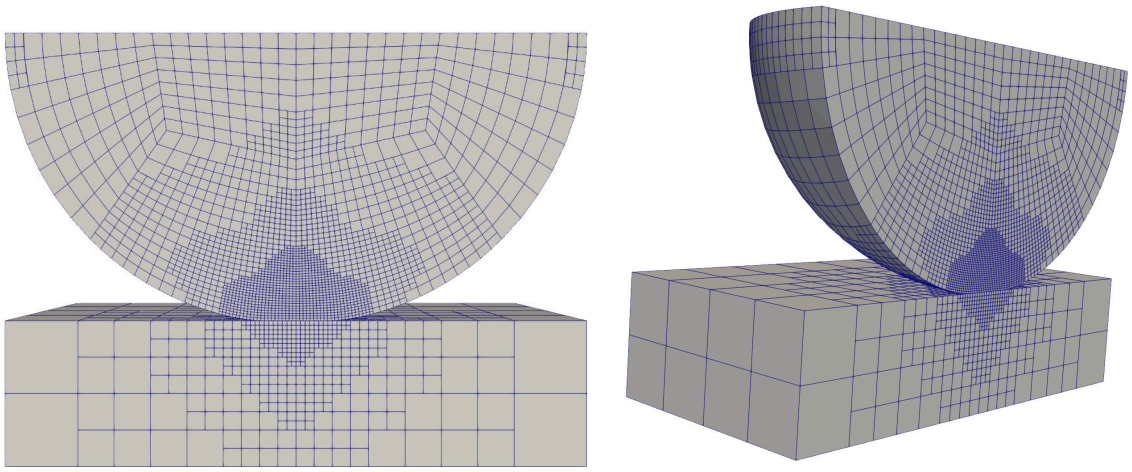


FIGURE 4.5. – Problème du contact Hertzien 2D : maillage final obtenu pour la combinaison AMR 1 avec $e_\Omega = 4\%$ et $k_N = 10^4$ E - ordre de la transformation super-paramétrique de 3.

cohérentes avec ce qui a été constaté dans le chapitre 2 (voir l'étude avec deux matériaux différents dans la section 2.3.3). Cependant, le raffinement du maillage diffère quelque peu selon la simulation considérée pour un même seuil d'erreur e_Ω. En effet, dans le cas 2D présenté, le demi-disque est beaucoup plus raffiné que le rectangle et la règle du "2 :1 balance" (un seul nœud esclave entre deux nœuds maîtres) n'est plus respectée dans la zone de contact, tandis que, dans le cas 3D, la différence de raffinement entre les deux solides est moins marquée. Cela est dû au fait que la demi-largeur de contact a soit plus grande pour le cas 2D que pour le cas 3D et, par conséquent, que le demi-disque se déforme plus que la demi-sphère.



(a) (b)

FIGURE 4.6. – Problème du contact Hertzien 3D : maillage final obtenu pour la combinaison AMR 1 avec $e_\Omega = 4\%$ et $k_N = 10^4$ E - Vue de la section de coupe en (a) et en biais en (b) - ordre de la transformation super-paramétrique de 3.

4.2. Parallélisation de la stratégie

La mise en œuvre d'une stratégie de partitionnement de maillages non concordants soumis à un problème de contact dans un cadre de calcul parallèle est un défi scientifique notable. En effet, étant donné que les nœuds maîtres interviennent dans plusieurs relations de contact, proposer un partitionnement du maillage et des structures de données qui permettent d'avoir accès à toutes les contributions des degrés de liberté (ddls) du maillage est nécessaire à la résolution du problème. En effet, les maillages n'étant plus concordants, rien n'empêche que des nœuds maîtres devant être appairés à un même nœud esclave soient définis sur des processus différents et que les données permettant de procéder à l'appariement des nœuds ne soient pas directement accessibles. De plus, ce partitionnement doit s'assurer que la charge soit bien équilibrée entre les cœurs tout au long du calcul pour que la parallélisation soit efficace.

Dans cette section, nous nous décrivons une stratégie de parallélisation de notre algorithme combiné AMR-contact avec maillages non concordants. L'extension de la stratégie à un cadre de calcul parallèle doit être basée sur deux éléments principaux : une technique pertinente d'accès aux données de discrétisation et une méthode de partitionnement efficace qui permette d'optimiser les surcoûts potentiels dus aux besoins supplémentaires d'accès à ces données. Tout d'abord, la stratégie envisagée d'accès aux données de discrétisations nécessaire pour réaliser efficacement l'appariement nœud-à-surface dans un cadre parallèle est présentée dans la section 4.2.1. De plus, une première solution permettant de s'affranchir des problématiques d'accès aux données est présentée. Puis, une approche de partitionnement du maillage, proche de la méthode "pie chart" présenté en section 3.3.2 et permettant d'équilibrer la charge tout en minimisant la dispersion des éléments appartenant à un même processus, est introduite dans la section 4.2.2. Enfin, les performances parallèles de la stratégie proposée sont analysées par des études de strong scaling et de comparaison des pré-conditionnements, et sont recentrés vis-à-vis de travaux les plus récents se focalisant sur le contact avec appariement nœud-à-surface en parallèle.

4.2.1. Extension dans un cadre parallèle : stratégies d'accès aux données

À notre connaissance, les travaux présentés dans [Mayr et al. 2023], déjà évoqués au chapitre 1 (voir la section 1.7.3), sont à ce jour les plus récents et les plus avancés en ce qui concerne la résolution de problèmes de contact en parallèle avec maillages non concordants. Dans ces travaux, des stratégies parallèles sont proposées pour un stockage efficace des paires d'éléments maître/esclave dans le but résoudre les problèmes de contact entre les nœuds et les surfaces en utilisant la méthode de type mortar (avec des multiplicateurs de Lagrange). En particulier, des stratégies de décomposition de domaine avec sous-domaines se recouvrant sont proposées (voir la figure 4.7), et ceci afin de garantir l'accès à toutes les paires maître/esclave nécessaires à la recherche de contacts et à l'évaluation des termes de mortar. L'approche "binning scheme" décrite dans la figure 4.7b consiste à découper l'interface maître en des intervalles réguliers et à étendre la couche fantôme d'autant d'intervalles que nécessaire, en partant des nœuds maîtres jusqu'à ce que l'extension de couche fantôme proposée permette de réaliser correctement l'appariement des nœuds en contact, c'est-à-dire que chaque nœud esclave soit appairé à l'ensemble des nœuds auquel il devrait.

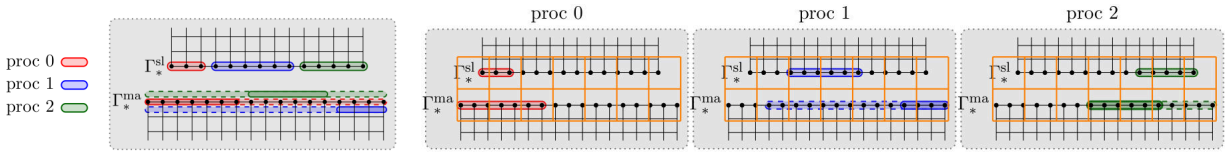


Figure 4.7. – Exemples de stockage de la discrétisation de l'interface maître par extension de la couche fantôme sur 3 processus : cas redondant en (a) et distribué (basé sur une approche appelée "binning scheme") en (b) - $\Gamma^{sl}_\star$ et $\Gamma^{ma}_\star$ sont les frontières de contact esclave et maître respectivement - les lignes de couleur en trait plein regroupent les nœuds possédés par chacun des processus initialement ; les lignes en pointillés traduisent l'extension de la couche fantôme élargissant les données disponibles à chaque processus.

La scalabilité de ces deux approches (redondante et distribuée avec "binning scheme") est vérifiée par des simulations utilisant jusqu'à 480 cœurs CPU pour des problèmes dépendants en temps et soumis à de grandes déformations. L'idée est donc de s'appuyer sur ce principe d'extension de la couche fantôme pour proposer une méthode de partitionnement et une stratégie scalable résolvant des problèmes de contact entre maillages non concordants. La principale limitation ici concerne le partitionnement initial du maillage qui peut engendrer une couche fantôme conséquente, y compris en l'étendant via l'approche appelée "binning scheme". En effet, comme il est possible de le remarquer sur la figure 4.7b pour le processus 1, aucun des nœuds non fantômes esclave (en haut) affectés à ce processus ne se situe en face de la partie de frontière maître (en bas) assignée à ce même processus. Une solution pour optimiser l'extension de la couche fantôme serait donc de proposer un partitionnement initial du maillage qui, pour chaque processus, permette d'avoir une majorité de nœuds esclaves qui soit déjà située en face de la frontière maître.

Par ailleurs, les auteurs de [Guillamet et al. 2022] ont également cherché à résoudre des problèmes de contact entre solides discrétisés avec des maillages non concordants. Néanmoins, les travaux présentés se sont limités à un contact de type rigide-déformable et

proposent une stratégie d'échange de données par MPI qui peut s'avérer coûteuse dans le cas d'un calcul sur un grand nombre de cœurs (car hausse des communications entre les processus). De plus, sa mise en œuvre dans le cas déformable-déformable et où la recherche des contacts est une opération fréquente (par exemple à chaque itération AMR comme décrit dans ces travaux de thèse), non présentée par les auteurs de ces travaux, pourrait engendrer des coûts de communication supplémentaires et non-négligeables à grand nombre de cœurs.

Concernant l'accès aux données de discrétisation des frontières de contact potentiel du maillage, étendre la couche fantôme maître, dans la même idée que la stratégie proposée dans [Mayr et al. 2023], est envisagée. Néanmoins, cette extension n'a pas été mise en œuvre pour le traitement des simulations présentées dans la section 4.2.3. Ici, le partitionnement proposé permet de limiter l'extension de la couche fantôme maître à un seul niveau (voir la figure 4.8) ; ce qui donne accès à toutes les données nécessaires pour la résolution du problème de contact en parallèle. Ainsi, dans ces simulations, une attention particulière est accordée au fait que, pour chaque processus possédant des nœuds en contact actif, toutes les données de discrétisation soient accessibles localement (*a minima* dans la couche fantôme telle que définie actuellement) pour que le système à résoudre soit aisé à construire sans communications.

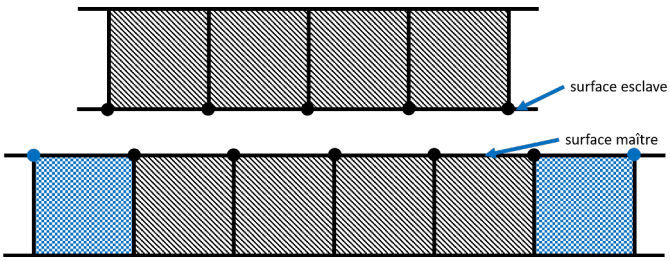
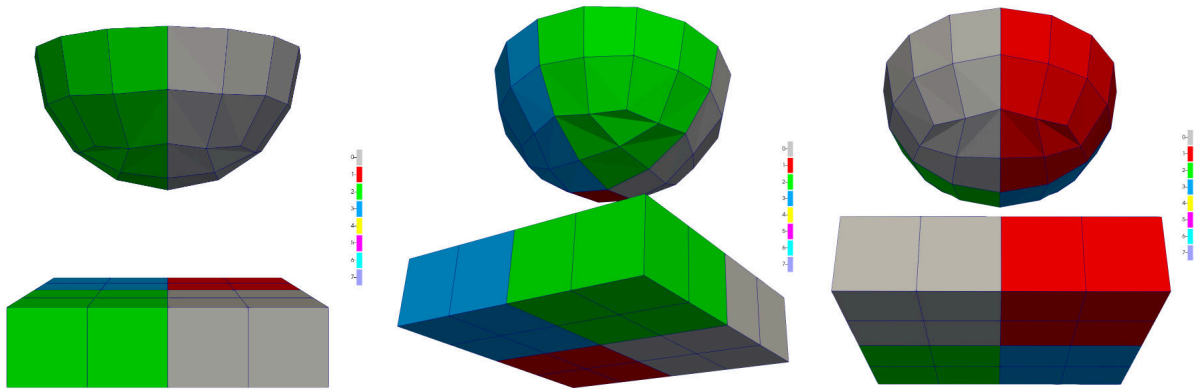


Figure 4.8. – Principe de l'extension de la couche fantôme maître d'un niveau - les éléments hachurés (en noir) appartiennent (ainsi que leurs nœuds) au processus courant ; les éléments en damier (en bleu) caractérisent l'extension de la couche fantôme maître dans le processus courant pour prendre en compte de nouveaux ddls fantômes.

Dans ces travaux de thèse, une stratégie est proposée ci-après pour s'affranchir du besoin d'étendre beaucoup la couche fantôme tout en permettant à plusieurs processus de traiter les nœuds en contact actif. Cette solution, bien que constituant une étape intermédiaire vers une stratégie générique basée sur l'extension automatisée de la couche fantôme, permet de résoudre avec robustesse et en parallèle le problème de contact avec maillages non concordants.

La spécificité de la solution proposée dans cette section réside dans la manière dont les éléments sont répartis sur les processus. Dans la suite, nous appellerons "arêtes concordantes" les arêtes des deux frontières de contact qui se situent en vis-à-vis, s'il y en a. Par conséquent, nous appellerons "sous-domaines concordants" les sous domaines délimitées par des arêtes concordantes qui se font face. La figure 4.9 illustre ces définitions. Dans cette figure, il est possible de dénombrer 4 paires d'arêtes concordantes qui permettent de délimiter 4 paires de sous-domaines concordants, en l'occurrence une bleue, une grise, une rouge et une vert clair.

La méthode proposée consiste à définir autant de processus qui traitent les éléments en contact actif qu'il y a de paires de sous-domaines concordants. Ainsi, définir un processus



(a) (b) (c)

FIGURE 4.9. – Illustration des sous-domaines concordants.

traitant les éléments en contact actif permet de s'affranchir du besoin d'étendre la couche fantôme créée dans MFEM suite au partitionnement du maillage[1]. La solution proposée s'appuie sur la consistance de l'opérateur de prolongation de MFEM parallélisé $\mathbf{P}$ qui gère efficacement les degrés de liberté fantôme, dans le sens où les contributions des degrés de liberté fantômes sont conservées dans la matrice $\mathbf{P}$ lors de la restriction du problème aux ddls conformes et non fantômes (voir les équations (2.24), (2.28) et (2.29) en page 68). En effet, dans un cadre de calcul en mémoire distribuée, cet opérateur permet relier directement un degré de liberté appartenant à un processus à son pendant fantôme d'un autre processus. Il s'écrit

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_0 & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} \\ \mathbf{Wg}_{1,0} & \mathbf{P}_1 & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ \mathbf{Wg}_{r,0} & \cdots & \mathbf{Wg}_{r,r-1} & \mathbf{P}_r & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \mathbf{0} \\ \mathbf{Wg}_{R-1,0} & \cdots & \cdots & \cdots & \mathbf{Wg}_{R-1,R-2} & \mathbf{P}_{R-1} \end{pmatrix} \tag{4.2}$$

avec $\mathbf{P}_r$ l'opérateur de prolongation spécifique à la région parallèle r (voir la section 3.2 pour ce qui est de la division du maillage en régions parallèles ou régions de tâches) et $\mathbf{Wg}_{r,i}$, $i \in [\![0; r-1]\!]$, l'opérateur d'interpolation des degrés de liberté fantômes sur les degrés de liberté (non fantômes) de la région parallèle i.

Remarques :

— le processus 0 ne possède pas d'éléments fantômes dans l'environnement logiciel MFEM ;

— les matrices d'interpolation $\mathbf{Wg}_{r,i}$ sont très creuses car ne possèdent que les contributions des degrés de liberté fantômes. Elles peuvent même être nulles dans le cas où aucun degré de liberté d'un processus ne correspond à un degré de liberté fantôme d'un autre processus.

1. Une couche fantôme minimale de degrés de liberté est construite par défaut dans MFEM dès lors que le calcul est distribué sur plusieurs processus (voir la section 3.2).

Ainsi, lorsque le problème de contact est restreint aux degrés de liberté conformes et non fantômes (voir les équations (2.27) et (2.29) en page 69 en particulier), toutes les contributions des nœuds en contact sont conservées.

Mettre en place un partitionnement du maillage qui ne soit pas basé sur l'utilisation de sous-domaines concordants n'est actuellement pas viable sans étendre la couche fantôme actuellement créée par MFEM dans le cas d'un appariement de type nœud-à-surface. En effet, dans ce cas-là, certains nœuds esclaves ne seraient pas appairés à l'ensemble des nœuds maîtres auxquels ils devraient l'être. Néanmoins, la solution décrite et mise en application dans les simulations considérées dans la section 4.2.3 présente une contrainte qui ne la rend pas générique. En effet, pour chaque paire de sous-domaines concordants, il ne peut y avoir qu'un seul processus qui traite les éléments en contact actif, tous les autres processus gérant les éléments de cette paire de sous-domaines doivent traiter uniquement des éléments en contact non-actif. Cela implique donc la nécessité de connaître *a priori* la zone en contact actif. Cependant, la solution introduite est robuste et convient en première approche.

Remarque : pour chaque paire de sous-domaines concordants, il peut y avoir un ou plusieurs processus qui traitent les éléments en contact potentiel qui ne sont pas en contact actif, comme illustré sur la figure 4.10. Sur cette figure, issue d'un calcul sur 48 cœurs CPU, il y a 4 sous-domaines concordants. Il y a donc 4 processus qui traitent l'ensemble des éléments en contact actif (en l'occurrence les processus 0, 2, 5 et 8 en gris, vert, rose et marron respectivement), et les 7 autres processus qui gèrent les éléments de contact potentiel ne traitent que des éléments en contact non actif.

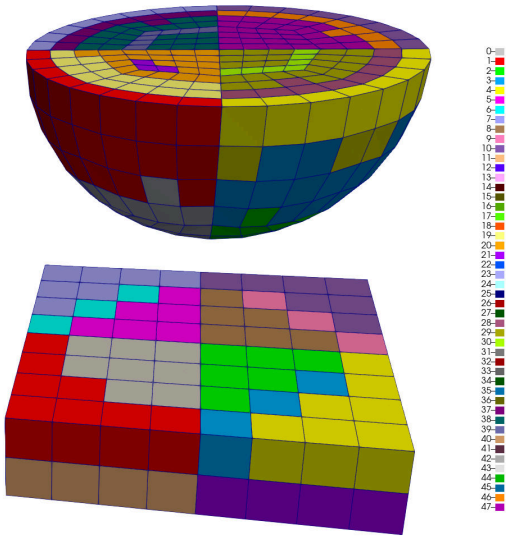


FIGURE 4.10. – Exemple de partitionnement 3D obtenu à partir de 4 paires de sous-domaines concordants.

4.2.2. Partitionnement dérivé du partitionnement "pie chart"

Comme évoqué dans la section précédente, le principal challenge de la mise en place d'une stratégie d'appariement nœud-à-surface dans un cadre de calcul haute performance réside dans le stockage et l'échange éventuel de données sur la discrétisation de l'interface maître. Pour cette raison, il semble important de proposer une méthode de partitionnement

du maillage qui permette de répondre efficacement à cet enjeu en minimisant les surcoûts dûs au stockage et aux communications de données. L'enjeu est double : le partitionnement proposé doit être en accord avec la solution d'accès aux données basée sur les paires de sous-domaines concordants introduite dans la section précédente et doit être adapté à la mise en œuvre d'une stratégie d'accès aux données plus générique, par exemple par l'extension de la couche fantôme comme envisagé dans la section précédente. Pour ce faire, l'approche de partitionnement proposée doit :

— minimiser le nombre de processus traitant les éléments de contact (afin de regrouper au maximum les éléments en contact actif) ;

— faire en sorte que les nœuds de contact de la surface esclave et que les nœuds de contact de la surface maître auxquels ces derniers sont appairés soient le plus possible sur les mêmes processus [2].

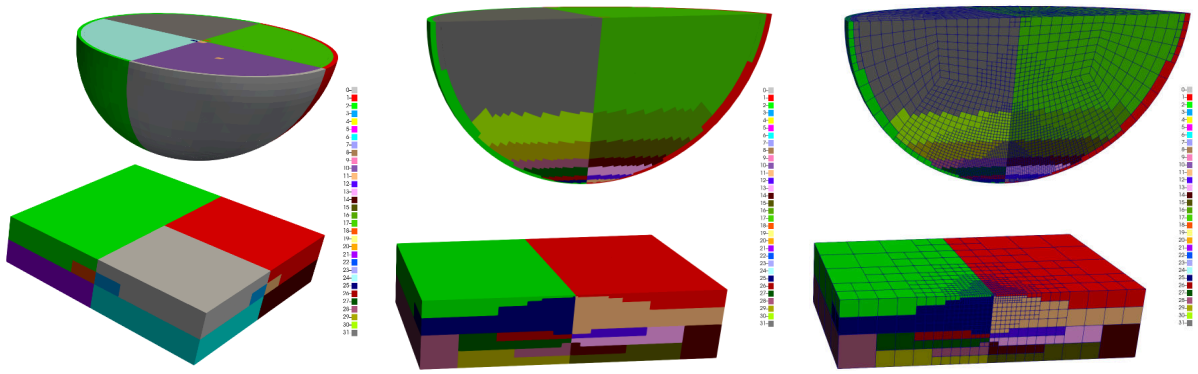
Il a été observé dans le chapitre 3 (en particulier par le biais des figures 3.5, page 93, 3.6, page 94, et 3.9, page 99) que le nombre de processus traitant les éléments de contact était moins important avec les deux approches de partitionnement séparant éléments de contact et de non-contact plutôt qu'avec la stratégie "SFC-contact". Pour cette raison, la méthode de partitionnement "SFC-contact", pourtant très scalable pour des contacts avec maillages concordants, n'est pas considérée ici. Ainsi, l'approche de partitionnement proposée pour distribuer efficacement le maillage sur les processus dans le cas d'un problème avec un appariement nœud-à-surface est basée sur la méthode de partitionnement "pie chart", dont le principe est présenté en section 3.3.2. Cette dernière s'est révélée fortement scalable jusqu'à plus de 8000 cœurs CPU et pour des simulations comportant près de 200 millions d'inconnues (voir la section 3.7), dans le cas d'un problème avec des maillages concordants.

De plus, l'utilisation d'une stratégie de partitionnement basée sur la méthode "pie chart" présente deux avantages notables. D'une part, le partitionnement étant basé uniquement sur le regroupement dans un même processus d'éléments voisins, les éléments maîtres et esclaves qui se font face seront déjà efficacement regroupés sur les mêmes processus. Ainsi, l'application d'une stratégie d'extension de la couche fantôme comme proposé dans [Mayr et al. 2023] peut potentiellement être optimisée. D'autre part, l'algorithme de redistribution des éléments étant basé sur la position spatiale moyenne des éléments du maillage, la seule adaptation nécessaire à apporter réside dans la définition de la forme des sous-ensembles d'éléments voisins (première étape de l'algorithme décrit dans la section 3.3.2). Cette adaptation reste par ailleurs indépendante de la discrétisation des maillages (concordants ou non) ou encore du fait d'appliquer de l'AMR au problème considéré. Un exemple de partitionnement de maillage 3D obtenu avec cette nouvelle méthode est illustré dans la figure 4.11. La figure 4.11 affiche les numéros des tâches MPI assignés aux éléments du maillage 3D utilisé pour une simulation avec AMR et adapté à un appariement de type nœud-à-surface. Dans cette figure, le calcul a été effectué en utilisant 32 cœurs CPU afin d'avoir un nombre raisonnable de couleurs et ainsi faciliter la lecture.

4.2.3. Analyse de la scalabilité et du pré-conditionnement

Pour étudier la scalabilité parallèle de notre méthode, des études en strong scaling et de l'influence du pré-conditionnement sont exposées pour le problème du contact Hertzien

2. Dans l'approche proposée, il est nécessaire que les éléments en contact actif de deux sous-domaines concordants soient sur les mêmes processus. Néanmoins, les éléments en contact non actif ne contribuent pas au contact dans le système linéaire et peuvent appartenir à d'autres processus. Néanmoins, ils restent regroupés par éléments voisins sur les processus afin de réduire les communications.



(a) (b) (c)

FIGURE 4.11. – Illustration d'un équilibrage de charge spécifique au contact de solides discrétisés avec des maillages non concordants via la stratégie "pie chart", typique du problème de Hertz - figures sans la représentation du maillage en (a) et en (b) ; figures en vue de coupe en (b) et en (c).

sphérique en 3D introduit en section 4.1.2.1. Ces études sont similaires à celles présentées dans le chapitre 3. Par ailleurs, il est à noter que la version de MFEM utilisée pour ces analyses est la version 4.4.

Pour réaliser ces études de scalabilité, le temps de calcul total mais aussi les temps passés au sein des principales fonctions suivantes du programme sont intégrées au profilage :

- application de l'algorithme d'appariement nœud-à-surface ;
- assemblage de la matrice de rigidité éléments finis (fonction MFEM `Assemble`) ;
- formation du système linéaire sans prise en compte des termes de contact (fonction MFEM `FormLinearSystem`, équation (2.26)) ;
- ajout des termes de contact et résolution du système matriciel pénalisé (application du solveur issu de la classe MFEM `PenaltyPCGSolver`, équation (2.6)) ;
- raffinement des éléments (approche AMR estimate-mark-refine, fonction MFEM `Apply`) ;
- équilibrage et redistribution de l'ensemble des éléments finis du maillage après raffinement (incluant la fonction MFEM `Rebalance`).

Remarque : identiquement aux études de strong scaling présentées dans le chapitre 3, l'assemblage du second membre éléments finis n'est pas inclus car son coût est négligeable par rapport aux autres fonctions du code.

Pour ce cas test 3D, seules la combinaison 1 pour l'AMR et la stratégie parallèle présentée dans cette section sont appliquées. Les solides considérés ont des propriétés matériaux identiques : $E = 210$ GPa et $\nu = 0,3$ et le maillage initial fourni en entrée de la simulation a été pré-raffiné deux fois par le mailleur Gmsh afin que chaque processus possède *a minima* un élément. Il est illustré par la figure 4.12. De plus, l'ordre de la transformation super-paramétrique des éléments est fixée à 3. Par ailleurs, le maillage fourni en entrée nous permet d'affecter 4 processus maximum pour le traitement des éléments en contact actif sans avoir à modifier la couche fantôme créée par MFEM lors du partitionnement du maillage.

Le seuil e_Ω est fixé à $1,5\%$ et le maillage résultant est illustré dans la figure 4.13. Ce maillage est obtenu après 6 itérations AMR, donc 5 étapes de raffinement du maillage.

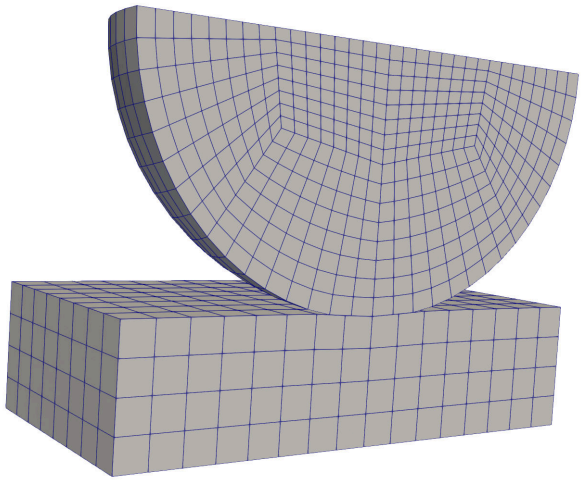
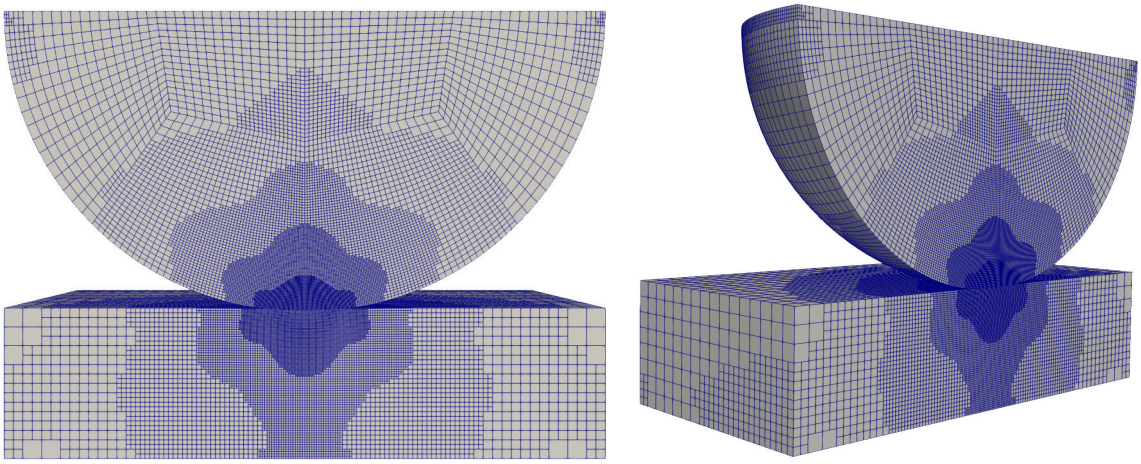


FIGURE 4.12. – Problème du contact Hertzien entre sphère et pavé 3D : vue de coupe du maillage initial - ordre de la transformation super-paramétrique de 3.

La solution discrète du problème $\mathbf{U}$ comporte 8,8 millions de ddls conformes. De plus, en incluant les ddls non-conformes et ceux fantômes, les calculs ont pu atteindre jusqu'à près de 17,1 millions de degrés de liberté, pour un calcul utilisant 2048 cœurs CPU. L'empreinte mémoire maximum totale des calculs (obtenue pour le calcul utilisant 2048 cœurs) vaut 694 GB. Par ailleurs, il est à noter que, dans tout le chapitre, les éléments finis utilisés restent trilinéaires (Q_1).



(a) (b)

FIGURE 4.13. – Problème du contact Hertzien 3D : maillage final obtenu pour la combinaison AMR 1 avec $e_\Omega = 1,5\%$ et $k_N = 10^4$ E - Vue de la section de coupe en (a) et en biais en (b) - transformation d'ordre 3.

Les calculs ont été effectués sur les nœuds de calcul du supercalculateur Topaze du CCRT/CEA (cf. section 1.8.5). Les résultats suivants sont obtenus en résolvant le système linéaire conforme en utilisant le solveur HyprePCG. Dans cette étude, les pré-conditionneurs HypreILU et HypreBoomerAMG sont à nouveau comparés. Pour HypreILU, l'influence de l'option caractérisant le niveau de remplissage des matrices de pré-conditionnement sur les performances du solveur est analysé. Par défaut, le niveau de remplissage de la matrice de pré-conditionnement est fixé à 1 dans l'environnement logiciel

MFEM et l'ensemble des analyses présentées dans le chapitre 3 l'ont été avec la valeur par défaut. Dans cette section, les résultats obtenus pour un choix de niveau de remplissage des matrice de pré-conditionnement de 1 sont comparés avec ceux pour un choix de niveau de remplissage valant 3. Par ailleurs, pour le pré-conditionneur HypreBoomerAMG, le paramètre de Strong Threshold (ST) est fixé à $0,7$.

L'étude de strong scaling est présentée dans la figure 4.14. Cette figure montre le speedup ainsi que le temps total et les temps des étapes de l'algorithme intégrées au profilage en fonction du nombre de cœurs CPU. Les figures 4.14a et 4.14b concernent les résultats obtenus avec HypreILU comme pré-conditionneur avec un niveau de remplissage de 1, tandis que les figures 4.14c et 4.14d illustre les performances parallèles de la stratégie avec ce même pré-conditionneur avec un niveau de remplissage de 3. Enfin, les figures 4.14e et 4.14f montrent les résultats obtenus en choisissant HypreBoomerAMG comme pré-conditionneur. Ces résultats sont comparés au strong scaling idéal, représenté en traits pointillés noirs. Il est également à noter que pour des problématiques d'empreinte mémoire, si le nombre de cœurs par processus est fixé à 2 pour les calculs allant entre 256 et 2048 cœurs, ce nombre vaut 4 et 8 pour les calculs utilisant 128 et 64 cœurs CPU respectivement. Cette augmentation du nombre de cœurs par processus pour ces calculs a donc pour conséquence d'améliorer les performances du solveur ; ce qui explique la perte de scalabilité de l'algorithme entre 64 et 256 cœurs CPU visible sur les figures.

Par ailleurs, des résultats obtenus sur la figure 4.14, il est possible de relever quelques points notables. Tout d'abord, la scalabilité globale de la stratégie (courbe bleue avec les ronds) s'avère satisfaisante jusqu'à 512 cœurs mais présente une perte importante de performance au-delà. Cette perte de performance est due à un mauvais équilibrage des éléments sur les processus, sachant que seuls 4 processus sont chargés de traiter les éléments en contact actif. En effet, le nombre d'éléments affectés aux processus traitant les éléments en contact actif devient nettement plus élevé que sur les autres processus au-delà de 512 cœurs. En conséquence, les performances parallèles de notre stratégie deviennent limitées par les performances des processus traitant ces éléments. Néanmoins, les performances parallèles obtenues constituent un apport vis-à-vis des résultats de la littérature. En effet, comme évoqué dans les sections 1.7.3 et 4.2.1, les résultats présentés dans [Mayr et al. 2023], qui sont les plus avancés sur ce sujet à notre connaissance, font état de performances parallèles satisfaisantes pour un nombre bien inférieur de cœurs CPU, pour un cas test avec géométries droites et sans AMR (voir la figure 4.15). Ainsi, l'application de notre stratégie permet d'atteindre une scalabilité satisfaisante pour des calculs utilisant deux fois plus de cœurs.

De plus, les résultats obtenus dans les figures 4.14a, 4.14c et 4.14e permettent de remarquer que les performances parallèles de la stratégie parallèle proposée sont légèrement meilleures lorsque le pré-conditionneur HypreILU est utilisé et que le niveau de remplissage des matrices de pré-conditionnement est fixé à 3 plutôt qu'avec HypreBoomerAMG ou HypreILU avec un niveau de remplissage de 1.

D'autre part, les résultats illustrés dans les figures 4.14b, 4.14d et 4.14f permettent d'observer que la quasi totalité du temps passé dans l'algorithme concerne les étapes de résolution du système linéaire par le solveur. Cette observation est d'autant plus notable que, lors des analyses de performances présentées dans le chapitre 3, les étapes de résolution du système matriciel par le solveur étaient prépondérantes dans le coût total de la stratégie (en temps de calcul) mais pas au même niveau qu'ici (voir les tableaux 3.8 et 3.9 (colonne

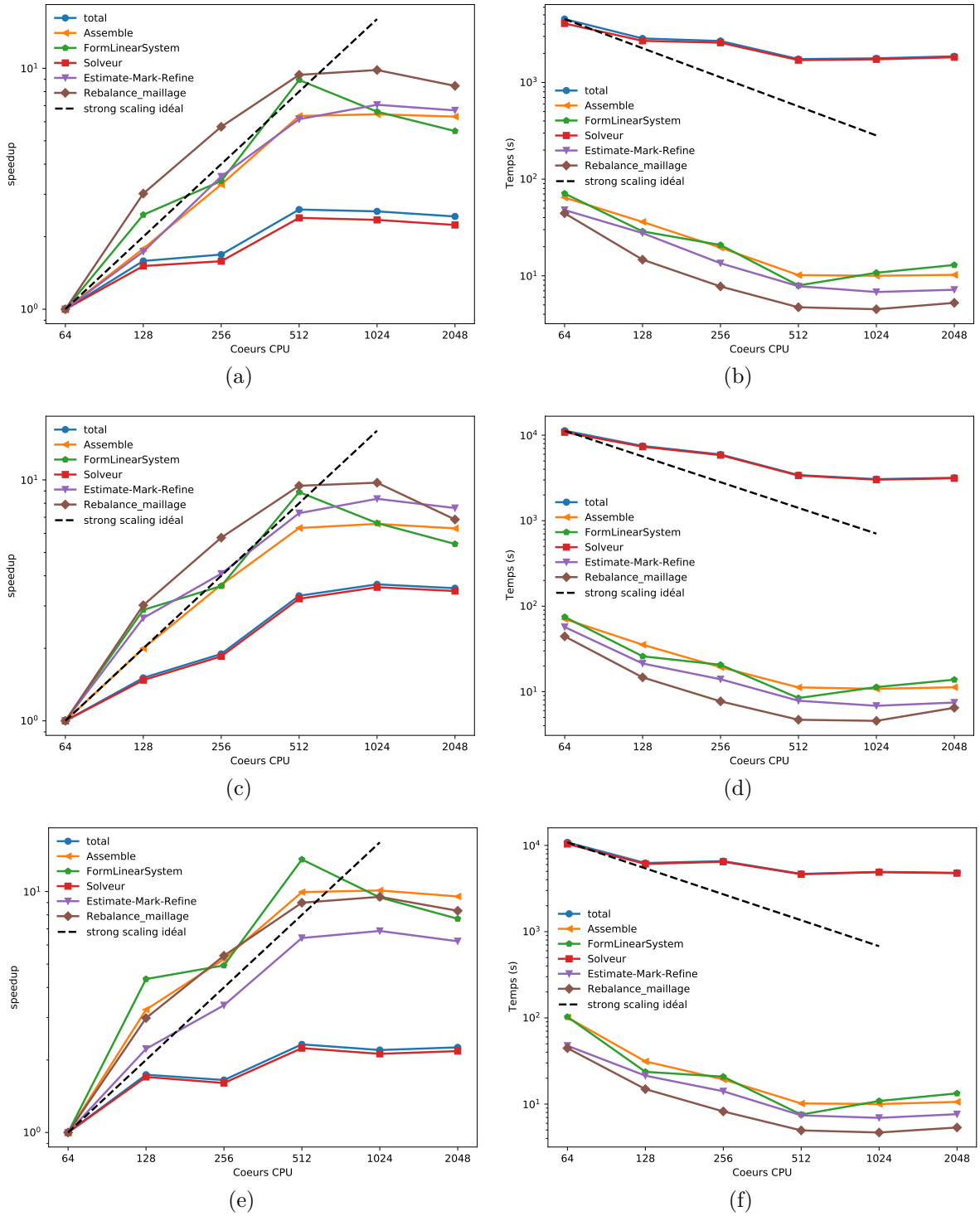
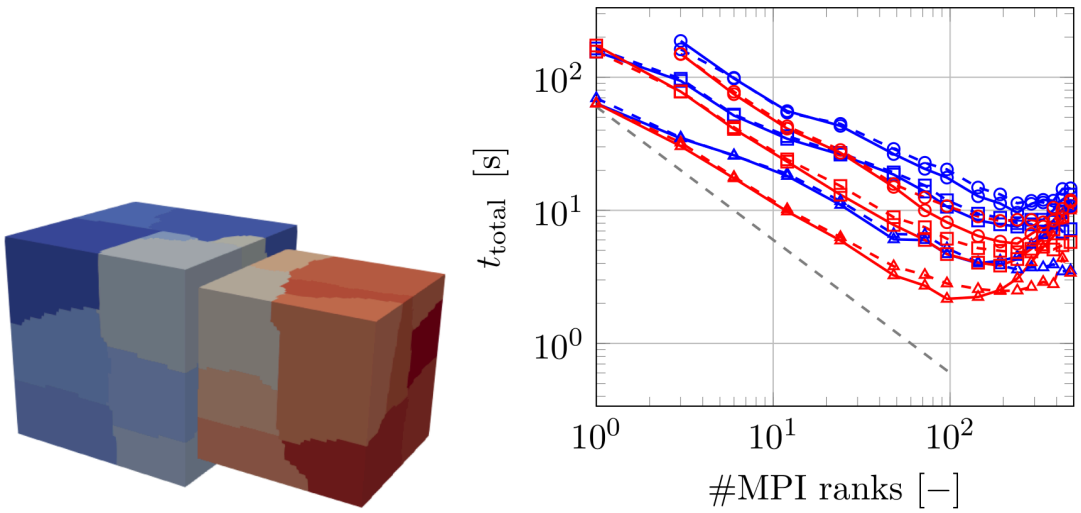


FIGURE 4.14. – Résultats des performances de l'algorithme et de ses principales étapes - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - solveur HyprePCG ; pré-conditionneur HypreILU avec niveau de remplissage de 1 en (a) et en (b), de 3 en (c) et en (d), et pré-conditionneur HypreBoomerAMG en (e) et en (f) - nombre de degrés de liberté du maillage conforme final : 8,8 millions - ordre de la transformation super-paramétrique de 3

du milieu) en page 127 ou encore la figure 3.22 en page 121 notamment). Cela est dû au



(a) (b)

FIGURE 4.15. – Contact 3D entre deux cubes étudiés dans [MAYR et al. 2023] - Cas test en (a) et résultats en strong scaling obtenus dans [MAYR et al. 2023] en (b) - Temps total de la résolution du contact en fonction du nombre de cœurs de calcul.

fait que la matrice de contact $\hat{\mathbf{K}}_C$ (voir l'équation (2.28) en page 69 en particulier) contient davantage de coefficients non nuls, et potentiellement situés plus éloignés de la diagonale. En effet, lors de l'appariement nœud-à-surface, les nœuds maîtres sont potentiellement impliqués dans plusieurs relations de contact ; ce qui n'était pas le cas lorsque l'appariement nœud-à-nœud était considéré. Ces nœuds contribuent ainsi davantage dans la matrice de contact. De plus, le fait que le seuil d'erreur relatif pour la convergence du solveur soit plutôt exigeant (10^{-20} dans les simulations présentées) cause ces temps de résolution très élevés.

Les figures 4.14b, 4.14d et 4.14f permettent également de remarquer que l'application de l'algorithme d'appariement nœud-à-surface est une étape plutôt coûteuse de la stratégie. En effet, il s'agit de la deuxième étape la plus coûteuse après l'étape de résolution du système linéaire par le solveur mais légèrement devant les fonctions d'assemblage matriciel et de formation du système linéaire.

Enfin, les figures 4.14b, 4.14d, 4.14f mais aussi 4.16, où sont comparés les choix de pré-conditionnement dans la même figure, permettent de montrer que choisir HypreILU comme pré-conditionneur avec un niveau de remplissage de 1 permet de réduire le temps de calcul total de l'algorithme d'un rapport 1,5 à 2 par rapport aux deux autres configurations de pré-conditionneur étudiées.

Il est également d'intérêt d'étudier le nombre d'itérations global du solveur au cours du calcul afin d'analyser la robustesse des configurations de pré-conditionnement avec le nombre de cœurs CPU. Pour ce faire, le nombre d'itérations du solveur lors du dernier passage dans l'étape de résolution du système matriciel est illustré en figure 4.17 pour les deux dernières itérations AMR : l'avant-dernière itération AMR dans la figure 4.17a et la dernière itération AMR dans la figure 4.17b.

Ces résultats montrent que, comme lors de l'application d'un appariement de type nœud-à-nœud (voir les figures 3.17a et 3.17b en page 115), le pré-conditionneur

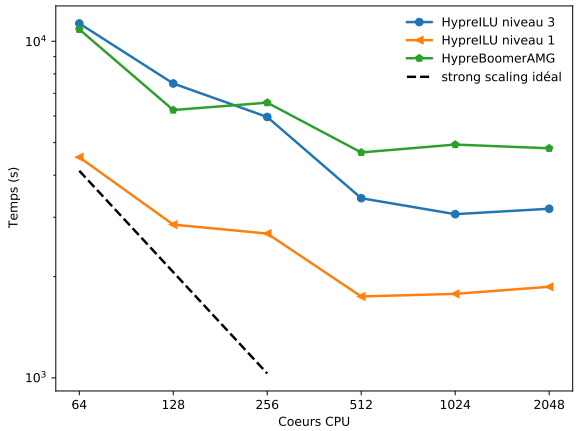


FIGURE 4.16. – Résultats des performances de l'algorithme pour différents choix de pré-conditionnement - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - solveur HyprePCG ; pré-conditionneur HypreILU avec niveau de remplissage de 1 avec les triangles en orange, de 3 avec les ronds en bleu, et pré-conditionneur HypreBoomerAMG avec les pentagones en vert - nombre de degrés de liberté du maillage conforme final : 8,8 millions - ordre de la transformation super-paramétrique de 3

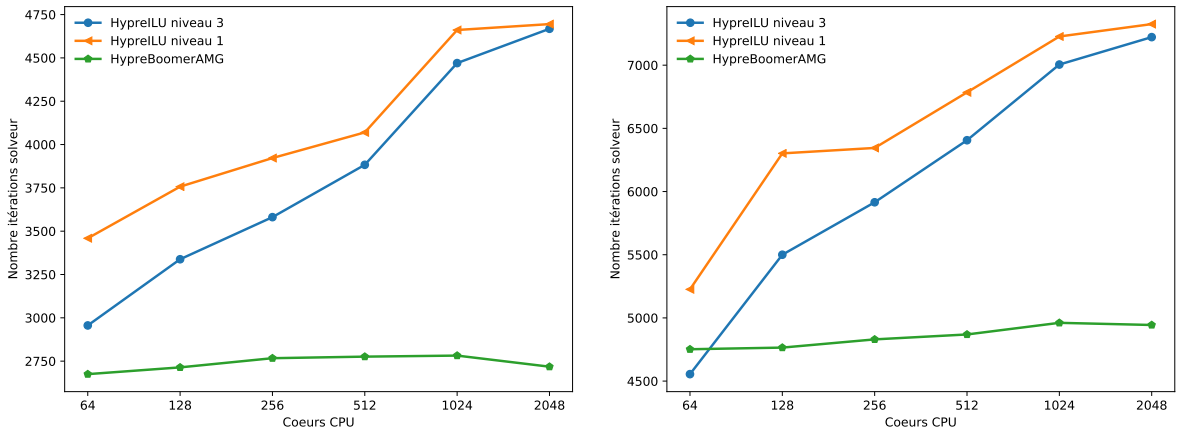


FIGURE 4.17. – Nombre d'itérations de la dernière étape de résolution du système matriciel par le solveur HyprePCG avec les trois configurations étudiées pour le pré-conditionneur - avant-dernière itération AMR en (a) ; dernière itération AMR en (b) - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1,5\%$) - nombre de degrés de liberté du maillage conforme final : 8,8 millions - ordre de la transformation super-paramétrique de 3

HypreBoomerAMG s'avère robuste car le nombre global d'itérations du solveur ne varie que très peu en fonction du nombre de cœurs CPU tandis que le pré-conditionneur HypreILU ne l'est pas puisque le nombre d'itérations du solveur augmente fortement avec le nombre de cœurs. Néanmoins, avec HypreBoomerAMG comme pré-conditionneur, le gain en nombre d'itérations du solveur par rapport aux calculs avec HypreILU est beaucoup moins marqué que lorsque un appariement de type nœud-à-nœud était appliqué (le lecteur pourra se référer aux figures 3.17a et 3.17b en page 115 pour la comparaison), en particulier lorsque le nombre de cœurs utilisés est faible. En effet, lorsque

un appariement de type nœud-à-nœud était appliqué, l'écart en nombre d'itérations entre HypreBoomerAMG et HypreILU était d'un rapport de 20 environ, tandis qu'avec l'application d'un appariement de type nœud-à-surface ce rapport est désormais plutôt de l'ordre de 1,5 pour le calcul utilisant 2048 cœurs CPU. De plus, en comparant les valeurs absolues, nous remarquons que le fait d'avoir des maillages non concordants (et donc d'appliquer une stratégie d'appariement de type nœud-à-surface) engendre une très forte augmentation du nombre d'itérations du solveur avec HypreBoomerAMG comme pré-conditionneur, passant de quelques dizaines d'itérations à quelques milliers. Ainsi, même si le pré-conditionneur HypreBoomerAMG est robuste (le nombre d'itérations augmente peu avec le nombre de cœurs), il s'avère moins efficace pour l'optimisation du nombre d'itérations du solveur lorsque les maillages ne sont plus concordants et qu'un appariement de type nœud-à-surface est appliqué.

En conclusion, les performances parallèles de la stratégie proposée se révèlent satisfaisantes jusqu'à 512 cœurs et sont prometteuses pour une extension plus générique de l'approche. De plus, il a été observé dans cette section que l'étape de résolution du système linéaire par le solveur est centrale lorsqu'un algorithme d'appariement nœud-à-surface est appliqué puisque le temps passé dans cette fonction est très largement prépondérante sur l'ensemble du reste de l'algorithme. Enfin, l'usage du pré-conditionneur HypreILU permet au solveur HyprePCG de résoudre le système linaire en des temps de calcul compétitifs lorsque le niveau de remplissage des matrices de pré-conditionnement est choisi égal à 1 mais présente une légère amélioration de la scalabilité lorsque ce niveau est fixé à 3. Néanmoins, choisir une valeur trop élevée (supérieure à 3) pour ce niveau de remplissage n'est pas judicieux car cela augmente le temps passé dans l'étape de résolution du système linéaire.

4.3. Conclusion

Dans ce chapitre, la stratégie combinant AMR et résolution du problème de contact en 3D élasto-statique exposé dans le chapitre 2 a été étendue à la gestion de maillages non concordants par la définition d'un algorithme d'appariement nœud-à-surface. Des exemples représentatifs 2D et 3D ont permis de valider la précision de l'approche proposée. En effet, nos solutions ont été vérifiées vis-à-vis de la solution de Hertz où la convergence a été obtenue. Cette stratégie s'est donc avérée une nouvelle fois efficace pour des problèmes de contact avec des frontières courbes et des éléments finis de premier ordre quadrangulaires ou hexaédriques super-paramétriques. Lors de cette étude, l'extension proposée a également été confrontée au cas où les matériaux des solides mis en contact sont différents.

La parallélisation de ce type d'appariement s'appuie sur deux éléments majeurs : une stratégie efficiente de stockage ou d'échange de données et une méthode pertinente de redistribution des éléments. Une première approche inspirée de la méthode "pie chart", décrite dans le chapitre 3, est implémentée pour répondre à ce challenge. Cette approche s'appuie sur la particularité de traiter les éléments de contact dans des processus MPI différents et partitionne les éléments selon leur position géographique. La différence apportée dans cette méthode de partitionnement, afin de la rendre adaptée au contact entre maillages non concordants, consiste à faire la distinction entre les éléments en contact actif et ceux qui ne le sont pas, et à mettre ces premiers éléments sur les mêmes tâches MPI. Cette stratégie a pu être évaluée sur le cas du contact Hertzien 3D et une simulation comportant jusqu'à 8,8 millions de degrés de liberté conformes et pour des calculs comportant jusqu'à

2048 cœurs CPU.

Les simulations considérées ont mis en évidence les prometteuses performances parallèles de l'approche de partitionnement implémentée pour résoudre nos problèmes en mécanique du contact par la méthode de pénalisation avec AMR. D'une part, elle s'est avérée scalable jusqu'à 512 processeurs mais présente des limites au-delà, en raison d'un mauvais équilibrage des éléments et par conséquent de la charge de calcul. Ce déséquilibrage est inhérent au principe de ce partitionnement, car tous les éléments en contact actif d'une même zone de contact sont assignés à un nombre restreint de tâches MPI qui ont donc une lourde charge de calcul. Néanmoins, les résultats obtenus ont permis d'obtenir des performances en scalabilité meilleures que celles exposées dans les travaux qui font, à notre connaissance, office de référence sur la parallélisation du contact entre maillages non concordants (algorithme scalable jusqu'à 250 cœurs environ). La stratégie parallèle proposée dans ce chapitre pour résoudre des problèmes de contact avec appariement de type nœud-à-surface dans un contexte de calcul haute performance peut ainsi servir de base pour le développement futur d'une proposition plus générique adaptée à des simulations engageant des centaines de millions d'inconnues et utilisant plusieurs milliers de cœurs. D'autre part, les résultats obtenus ont permis de constater que l'étape de résolution du système linéaire par le solveur pré-conditionné devient critique lorsque les maillages sont non concordants (où l'algorithme nœud-à-surface doit être appliqué). Enfin, la comparaison des pré-conditionneurs ont permis de montrer que le pré-conditionneur HypreILU reste le plus compétitif pour résoudre le problème de contact étudiés dans ce chapitre. De plus, pour le pré-conditionneur HypreILU, augmenter le paramètre de niveau de remplissage des matrices de pré-conditionnement ne présente pas d'avantage flagrant.

Tout comme pour le chapitre 3, la stratégie de redistribution des éléments du maillage implémentée peut facilement être transposable à d'autres environnements logiciels éléments finis. Néanmoins, la principale difficulté de cette transposition réside dans l'adaptation des objets et fonctions spécifiques à MFEM (`ParFiniteElementSpace` et `ParMesh` notamment) au sein des environnements logiciels cibles.

Conclusion et perspectives

Ce manuscrit présente une solution parallèle combinant aisément le raffinement adaptatif de maillage (AMR) et le contact élasto-statique entre deux solides déformables. L'objectif est donc atteint de pouvoir mailler finement dans la zone de contact pour obtenir une solution précise et de gérer de manière parallélisée, grâce à l'environnement logiciel de travail MFEM, des problèmes comportant un très grand nombre de degrés de liberté, c'est-à-dire plusieurs centaines de millions d'inconnues.

Le premier axe de ce travail introduit un algorithme AMR h-adaptatif non-conforme dédié à la résolution de problèmes en mécanique du contact comportant des frontières de contact courbes. Ces problèmes de contact sont tout d'abord résolus, sur des maillages initiaux quadrangulaires ou hexaédriques structurés, par la méthode de pénalisation combinée à celle des statuts et s'appuyant sur un appariement des éléments de contact avec un algorithme de type nœud-à-nœud. L'analyse de cas 2D et 3D représentatifs ont permis de s'assurer que l'approche proposée est précise et performante, grâce à des études de convergence par rapport à des solutions de référence dans la zone de contact mais aussi dans l'ensemble du domaine d'étude. En effet, ces analyses ont permis de montrer que l'utilisation de l'AMR permet d'optimiser efficacement le nombre de degrés de liberté des simulations considérant un problème de contact. Ainsi, la mise en œuvre d'une stratégie AMR est bénéfique pour le traitement de problèmes de contact 3D en élasto-statique, en particulier quand une solution éléments finis très fine est nécessaire. En outre, l'usage d'éléments finis super-paramétriques a permis de conserver avec efficacité la courbure des frontières de contact mises en jeu, en particulier pour des solutions éléments finis d'ordre 1. De plus, la comparaison des combinaisons des critères AMR appliqués a permis d'observer leur particularité. L'association des critères de détection ZZ et d'arrêt global ont conduit à un raffinement du maillage localisé au niveau de la zone de contact et à une erreur possiblement importante localement sur le domaine d'étude à la fin du calcul. La combinaison des critères de détection LOC et d'arrêt local ont, quant à eux, engendré un raffinement de maillage au niveau de la zone limite contact/décollement et l'erreur locale à la fin du calcul respecte une précision définie localement. Néanmoins, à seuil d'erreur donné, cette dernière combinaison de critères produit un maillage comportant un plus grand nombre de degrés de liberté qu'avec la combinaison des deux premiers critères. L'approche proposée a également été appliquée avec succès sur des simulations avec des solides en contact ayant des propriétés mécaniques différentes.

Le deuxième axe de ce travail s'articule autour de la parallélisation de l'algorithme combinant AMR et résolution du problème de contact. À cette fin, l'implémentation et l'analyse de trois approches de partitionnement différentes, ayant pour principe commun d'assigner les éléments de contact en vis-à-vis à un même processus, ont permis de mettre en exergue leurs forces mais aussi leurs limites. L'algorithme "equidistributed", présenté tout d'abord, a pour spécificité de traiter les éléments de contact dans des tâches MPI différentes et de distribuer les éléments de contact et les éléments de non-contact de manière équidistribuée localement (pour chaque processus). La stratégie "pie chart" s'appuie sur la particularité de traiter également les éléments de contact dans des tâches MPI différentes mais partitionne, pour sa part, les éléments selon leur position géographique avec une géométrie de découpe prescrite. Enfin, l'algorithme "SFC-contact"

s'approche du partitionnement Hilbert SFC classique mais se distingue de ce dernier par la création de "super-éléments" regroupant deux-à-deux les éléments de contact des deux solides de manière à ce que les éléments de contact en vis-à-vis soient sur les mêmes tâches MPI. Tout d'abord, la méthode de partitionnement "equidistributed" s'est révélée performante jusqu'à 512 cœurs CPU mais est limitée au-delà pour des raisons de coûts de communication importants car les régions parallèles étaient composées d'éléments dispersés dans le domaine. Le regroupement d'éléments voisins sur les mêmes processus s'avérant primordial pour l'obtention de bonnes performances parallèles, les stratégies de partitionnement "pie chart" et "SFC-contact" ont permis de réduire les surcoûts observés pour "equidistributed". Ainsi, avec ces deux autres méthodes de partitionnement, la stratégie combinant AMR et contact s'est révélée fortement scalable pour des simulations comportant jusqu'à 400 millions de degrés de liberté et utilisant jusqu'à 8192 unités de calcul. De plus, ces deux approches sont suffisamment génériques pour être adaptables à d'autres études et à d'autres environnements logiciels éléments finis sans modifications extensives. En outre, les études réalisées ont permis, d'une part, de remarquer que les fonctions spécifiques de MFEM d'application de l'approche estimate-mark-refine choisie, de rééquilibrage de charge et de résolution du système linéaire sont gourmandes en empreinte mémoire, en particulier lorsque le maillage devient volumineux. D'autre part, ces travaux ont permis de souligner que choisir des éléments à transformation super-paramétrique d'ordre très élevé implique une forte consommation de mémoire au cours des calculs mais aussi d'allonger notablement le temps passé dans les fonctions MFEM d'assemblage de la matrice de raideur du problème, de formation du système linéaire et d'application de l'approche AMR. Par ailleurs, la comparaison des pré-conditionneurs HypreILU et HypreBoomerAMG a montré que le premier est plus robuste et moins coûteux en matière de temps de calcul que le second pour le traitement des problèmes de contact avec appariement nœud-à-nœud que nous avons considérés, au prix d'un nombre d'itérations de solveur plus élevé.

Le troisième axe de ce travail vise à étendre l'approche combinant AMR et contact à la mise en place d'un algorithme d'appariement nœud-à-surface pour la détection des éléments en contact. L'analyse de cas d'études 2D et 3D représentatifs ont permis de valider la précision de l'extension (contact en nœud-à-surface avec AMR) proposée et de s'assurer que la courbure des frontières de contact est bien préservée efficacement grâce aux éléments finis super-paramétriques. Une méthode de partitionnement du maillage dédiée au contact a été introduite. En plus d'avoir la particularité de traiter les éléments de contact et de non-contact dans des processus distincts, différencier les éléments en contact actif et ceux qui ne le sont pas a permis de la rendre appropriée à la résolution du contact entre maillages non concordants. Les performances parallèles de l'approche proposée se sont avérées prometteuses sur des simulations comportant des millions d'inconnues et plusieurs centaines de cœurs.

Ainsi, les travaux entrepris durant cette thèse et exposés dans ce manuscrit peuvent contribuer à la feuille de route du projet PLEIADES-HPC mené par le Service d'Études et de Simulation du comportement des Combustibles du CEA, en particulier en ce qui concerne l'obtention de solutions plus précises pour des problématiques d'interaction mécanique pastille-gaine.

Les travaux entrepris dans cette thèse permettent d'envisager leur extension à un large

panel de problèmes. Tout d'abord, dans le cadre de la mécanique des structures, un des axes proches de développement pourrait concerner les comportements mécaniques non-linéaires. Une piste de travail pourrait être de prendre en compte des comportements mécaniques anisotropes, orthotrope par exemple, ou visco-élastique car se rapprochant du comportement mécanique du combustible nucléaire. Dans ce cas, le projet MFEM-MGIS [Helfer et al. 2021 ; Helfer s. d.] peut être utilisé à cette fin, étant donné que la bibliothèque MFEM autonome actuelle traite seulement des comportements mécaniques élastiques ou hyper-élastiques. Par ailleurs, l'introduction de notre algorithme parallèle AMR-contact dans l'environnement logiciel MFEM-MGIS ne semble pas être un challenge informatique majeur.

En ce qui concerne la mécanique du contact, le prolongement logique de ces travaux serait le traitement du contact frottant. Pour caractériser la force de contact tangentielle, une version pénalisée du frottement, qui régulariserait la loi de frottement dans la même idée que pour le contact normal, pourrait être considérée. Le challenge serait ainsi axé sur l'implémentation numérique d'une loi de frottement dans l'environnement logiciel MFEM. De plus, la distinction efficace des phases dites d'adhérence et celles de glissement entre les solides est nécessaire dans le but d'obtenir une solution AMR précise.

L'élargissement de la stratégie proposée à des problèmes de multi-contacts ou mettant en jeu des surfaces potentiellement rugueuses peut également être un axe d'étude futur. En effet, les algorithmes d'appariement implémentés jusqu'ici ne permettant d'appairer les nœuds que de deux frontières à la fois et nécessitant de fixer au préalable la normale de contact, ils ne sont pas encore adaptés aux problèmes évoqués. Une piste de travail pourrait être de parcourir l'algorithme d'appariement des nœuds situés sur les frontières de contact autant de fois qu'il y a de zones en contact potentiel. Pour appliquer efficacement cette adaptation, les normales des différents contacts devront être renseignées avec soin. De plus, son implémentation dans un cadre parallèle est un enjeu et les méthodes de partitionnement mises en œuvre dans ces travaux de thèse devront être adaptées à ce type de problèmes.

Générique pour tous problèmes de contact élasto-statique avec une détection du contact simple, la stratégie combinée contact-AMR-HPC proposée pourrait être étendue à des problèmes de contact présentant des zones de détections non définies au préalable. Ces travaux de thèses pourraient notamment combinés avec ceux de A. Motte, réalisés au CEA de Paris-Saclay en collaboration avec l'Onera, qui se concentrent sur la détection du contact en mémoire distribuée visant à proposer une solution générique accélérant le processus de détection du contact.

Enfin, l'adaptation de la stratégie à des problèmes dépendants en temps est un axe d'étude d'intérêt dans le cas de modélisations avec comportement mécanique non linéaire. Les enjeux se concentreront ici sur deux aspects principaux. Tout d'abord, une attention spéciale à la question du transfert de champ entre les pas de temps sera essentielle pour la réalisation d'une adaptation dynamique et automatique du maillage pertinente. Ensuite, les problématiques d'équilibrage de charge entre les pas de temps seront à considérer avec soin. En effet, il ne serait probablement pas judicieux de redistribuer le maillage à chaque pas de temps en raison des coûts que cela représente et, par conséquent, décider d'une fréquence adéquate de rééquilibrage de charge, par le biais de critères appropriés, serait pertinent.

Annexes

Annexe A

Résultats complémentaires sur l'apport des méthodes de partitionnement "pie chart" et "SFC-contact"

Les tableaux A.1 et A.2, cités à la section 3.5.5 du chapitre 3, s'attachent à comparer les résultats obtenus en matière de temps passé dans la dernière étape de résolution du système matriciel dans sa globalité pour chacune des deux dernières itérations AMR pour les méthodes de partitionnement "equidistributed", "pie chart" et "SFC-contact", et des calculs sur 512 et 1024 cœurs CPU. L'objectif visé par ces résultats est de montrer l'apport des approches de partitionnement "pie chart" et "SFC-contact" vis-à-vis de "equidistributed". Ici, les nombres d'itérations du solveur correspondantes sont également mentionnés. HypreBoomerAMG est choisi comme pré-conditionneur et le paramètre de StrongThreshold est fixé à $0,5$.

Méthode de partitionnement		"equidistributed"	"pie chart"	"SFC-contact"
Total - Étapes solveur	Temps (s)	∅	$17,4$	$14,8$
Avant-dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour PCG	∅	30	33
	Temps (s)	∅	$1,6$	$1,5$
Dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour PCG	∅	35	32
	Temps (s)	∅	$3,1$	$2,4$

TABLEAU A.1. – Temps de résolution et nombre d'itérations de HyprePCG avec le pré-conditionneur HypreBoomerAMG - StrongThreshold $= 0,5$ - Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\mathrm{LOC}} = 4\%$, $\delta = 0,1\%$) - Strong Threshold $= 0,5$ - nombre de degrés de liberté du maillage conforme final : $5,1$ millions - 512 cœurs CPU - ordre de la transformation super-paramétrique 6.

Méthode de partitionnement		"equidistributed"	"pie chart"	"SFC-contact"
Total - Étapes solveur	Temps (s)	∅	15,5	12,9
Avant-dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour PCG	∅	35	34
	Temps (s)	∅	1,4	1,2
Dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour PCG	∅	32	33
	Temps (s)	∅	2,2	1,7

TABLEAU A.2. – Temps de résolution et nombre d'itérations de HyprePCG avec le préconditionneur HypreBoomerAMG - Strong Threshold $= 0,5$ - Contact Hertzien 3D - combinaison AMR 2 ($e_{\Omega,\mathrm{LOC}} = 4\%$, $\delta = 0,1\%$) - nombre de degrés de liberté du maillage conforme final : 5,1 millions - 1024 cœurs CPU - ordre de la transformation super-paramétrique 6.

Annexe B

Résultats complémentaires sur l'empreinte mémoire et les performances solveur

Les tableaux B.1, B.2 B.3 et B.4, cités aux sections 3.6.2 (pour les tableaux B.1 et B.2) et 3.6.3 (pour les tableaux B.3 et B.4) du chapitre 3, reportent les résultats obtenus en matière de temps passé dans la dernière étape de résolution du système matriciel dans sa globalité pour chacune des deux dernières itérations AMR pour les méthodes de partitionnement "pie chart" et "SFC-contact", et des calculs sur 1024, 2048 et 4096 cœurs CPU. Les résultats obtenus ont pour objectif de comparer les pré-conditionneurs mais aussi l'ordre de la transformation super-paramétrique des éléments finis pour les deux méthodes de partitionnement considérées. Ici, les nombres d'itérations du solveur correspondantes sont également mentionnés ainsi que des données sur l'empreinte mémoire (globale sur le calcul et spécifique à aux passages dans le solveur de la dernière itération AMR). Les pré-conditionneurs HypreILU et HypreBoomerAMG (avec paramètre de Strong Threshold fixé à $0,7$) sont comparés, tout comme l'ordre de la transformation super-paramétrique des éléments finis.

Dans chaque tableau sont répertoriés (de haut en bas) :

— le temps total du calcul (en secondes) ;

— l'empreinte mémoire totale (en gigabyte, GB) ;

— le temps total passé dans les étapes de résolution du système linéaire (en secondes) ;

— les nombres d'itérations du solveur et les temps de résolution pour la dernière étape de résolution du système linéaire (en se focalisant sur l'avant-dernière et la dernière itération AMR) ;

— l'empreinte mémoire des étapes de résolution du système linéaire de la dernière itération AMR.

cœurs CPU		1024		2048		4096	
Méthode de partitionnement		"pie chart"	"SFC-contact"	"pie chart"	"SFC-contact"	"pie chart"	"SFC-contact"
Total simulation	Temps (s)	$508,2$	$467,7$	$273,9$	$250,7$	$224,9$	$143,1$
Empreinte mémoire totale (GB)		$1400,0$	$1431,9$	$1811,0$	$1778,4$	$2928,6$	$2663,3$
Total - Étapes solveur	Temps (s)	$65,2$	$50,8$	$44,2$	$24,8$	$32,6$	$14,0$
Avant-dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour CG	672	647	990	688	1202	736
	Temps (s)	$3,9$	$3,6$	$3,5$	$2,0$	$2,3$	$1,1$
Dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour CG	723	728	1234	782	1521	837
	Temps (s)	$13,9$	$11,5$	$8,9$	$5,6$	$6,2$	$2,7$
Dernière itération AMR - Étapes solveur	Empreinte mémoire (%)	$0,3$	$0,5$	0	$0,4$	0	0

TABLEAU B.1. – Temps de résolution et nombre d'itérations - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $U = 18,3$ millions - ordre de la transformation super-paramétrique 6.

cœurs CPU		1024		2048		4096	
Méthode de partitionnement		"pie chart"	"SFC-contact"	"pie chart"	"SFC-contact"	"pie chart"	"SFC-contact"
Total simulation	Temps (s)	488, 2	449, 9	275, 7	239, 6	173, 7	175, 0
Empreinte mémoire totale (GB)		1452, 7	1431, 0	1961, 9	1809, 7	3016, 7	2683, 5
Total - Étapes solveur	Temps (s)	39, 7	24, 0	30, 8	17, 6	37, 1	18, 9
Avant-dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour CG	37	36	38	37	40	37
	Temps (s)	3, 1	2, 0	2, 9	1, 4	3, 2	1, 4
Dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour CG	38	39	40	37	39	39
	Temps (s)	9, 3	4, 6	5, 3	2, 8	6, 0	2, 0
Dernière itération AMR - Étapes solveur	Empreinte mémoire (%)	0, 9	0, 5	4, 0	1, 5	0	0, 1

TABLEAU B.2. – Temps de résolution et nombre d'itérations - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG ; pré-conditionneur HypreBoomerAMG - Strong Threshold $= 0, 7$ - nombre de degrés de liberté du maillage conforme final : $U = 18, 3$ millions - ordre de la transformation super-paramétrique 6.

cœurs CPU		1024		2048		4096	
Méthode de partitionnement		"pie chart"	"SFC-contact"	"pie chart"	"SFC-contact"	"pie chart"	"SFC-contact"
Total simulation	Temps (s)	$110,2$	$71,8$	$63,9$	$41,1$	$51,7$	$23,9$
Empreinte mémoire totale (GB)		$547,6$	$559,6$	$903,6$	$701,7$	$1664,4$	$1004,1$
Total - Étapes solveur	Temps (s)	$72,6$	$45,4$	$42,3$	$26,3$	$34,4$	$14,1$
Avant-dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour CG	778	649	842	687	1203	736
	Temps (s)	$4,4$	$3,7$	$2,4$	$1,6$	$2,4$	$0,8$
Dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour CG	925	729	946	780	1439	835
	Temps (s)	$16,7$	$10,3$	$9,7$	$6,4$	$6,4$	$3,2$
Dernière itération AMR - Étapes solveur	Empreinte mémoire (%)	$0,5$	$19,7$	$0,04$	14.5	0	$9,0$

TABLEAU B.3. – Temps de résolution et nombre d'itérations - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG ; pré-conditionneur HypreILU - nombre de degrés de liberté du maillage conforme final : $U = 18,3$ millions - ordre de la transformation super-paramétrique 3.

cœurs CPU		1024		2048		4096	
Méthode de partitionnement		"pie chart"	"SFC-contact"	"pie chart"	"SFC-contact"	"pie chart"	"SFC-contact"
Total simulation	Temps (s)	79, 8	50, 7	82, 5	32, 8	44, 4	26, 7
Empreinte mémoire totale (GB)		653, 2	601, 9	984, 7	796, 1	1767, 7	1184, 2
Total - Étapes solveur	Temps (s)	40, 6	23, 9	59, 0	17, 4	27, 8	16, 4
Avant-dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour CG	36	36	38	38	38	38
	Temps (s)	3, 5	2, 1	4, 9	1, 4	2, 4	1, 7
Dernière itération AMR - Dernière étape solveur	Nombre d'itérations pour CG	38	37	39	38	38	39
	Temps (s)	9, 4	4, 7	7, 7	3, 4	4, 3	1, 9
Dernière itération AMR - Étapes solveur	Empreinte mémoire (%)	10, 9	22, 3	0, 3	20, 1	0	16, 8

TABLEAU B.4. – Temps de résolution et nombre d'itérations - Contact Hertzien 3D - combinaison AMR 1 ($e_\Omega = 1\%$) - solveur HyprePCG ; pré-conditionneur HypreBoomerAMG - Strong Threshold $= 0, 7$ - nombre de degrés de liberté du maillage conforme final : $U = 18, 3$ millions - ordre de la transformation super-paramétrique 3.

Annexe C

Comparaison de l'ordre de la transformation super-paramétrique des éléments

Les tableaux C.1 et C.2, cités à la section 3.6.3 du chapitre 3, comparent les temps passés dans les principales fonctions de l'algorithme combiné contact-AMR-HPC selon l'ordre de la transformation super-paramétrique des éléments finis pour les méthodes de partitionnement "pie chart" et "SFC-contact", et des calculs sur 1024 et 2048 cœurs CPU. Les résultats présentés dans ces tableaux et la section correspondante ont pour objectif de comparer l'ordre de la transformation super-paramétrique des éléments finis pour les deux méthodes de partitionnement choisies. Ici, le pré-conditionneur HypreILU est choisi.

Temps passé dans la fonction (s)	ordre de la transformation de 3	ordre de la transformation de 6
Total	110	508
Assemblage	9,96	194
Formation du système linéaire (sans le contact)	9,80	106
Ajout du contact et résolution du système matriciel	72,6	65,2
Estimate-Mark-Refine	7,70	180
Rééquilibrage de charge	5,91	8,97

TABLEAU C.1. – Contact Hertzien 3D ($e_\Omega = 1\%$, 4 cœurs par processus) - solveur HyprePCG ; pré-conditionneur HypreILU - méthode de partitionnement "pie chart" - calculs sur 1024 processus.

Temps passé dans la fonction (s)	ordre de la transformation de 3	ordre de la transformation de 6
Total	41,1	251
Assemblage	4,67	97,4
Formation du système linéaire (sans le contact)	1,80	42,2
Ajout du contact et résolution du système matriciel	26,3	24,8
Estimate-Mark-Refine	3,86	90,4
Rééquilibrage de charge	0,92	3,22

TABLEAU C.2. – Contact Hertzien 3D ($e_\Omega = 1\%$, 4 cœurs par processus) - solveur HyprePCG ; pré-conditionneur HypreILU - méthode de partitionnement "SFC-contact" - calculs sur 2048 processus.

Bibliographie

N. El-Abbasi et K.-J. Bathe (2001). Stability and patch test performance of contact discretizations and a new solution algorithm. *Computers & Structures* 79.16, p. 1473-1486. doi : https://doi.org/10.1016/S0045-7949(01)00048-7 (cf. p. 31).

P. C. Africa, S. Perotto et C. de Falco (2022). Scalable recovery-based adaptation on quadtree meshes for advection-diffusion-reaction problems (cf. p. 51).

P. Alart (1988). *Multiplicateurs 'augmentés' et méthode de Newton généralisée pour contact avec frottement*. Rapp. tech. Report, Document LMA-DME-EPFL, Lausanne (cf. p. 35).

– (2007). *Contact on Multiprocessor Environment : from Multicontact Problems to Multiscale Approaches*. Computational Contact Mechanics, Springer, Vienna, p. 163-217 (cf. p. 52).

P. Alart, M. Barboteu et F. Lebon (sept. 1997). Solution of frictional contact problems by an EBE preconditioner. *Computational Mechanics* 20.4, p. 370-378. doi : 10.1007/s004660050258 (cf. p. 47).

P. Alart et A. Curnier (1991). A mixed formulation for frictional contact problems prone to Newton like solution methods. *Comput Methods Appl Mech Eng* 92, p. 353-375 (cf. p. 35).

P. Alart et F. Lebon (mars 1995). Solution of frictional contact problems using ILU and coarse/fine preconditioners. *Computational Mechanics* 16, p. 98-105. doi : 10.1007/BF00365863 (cf. p. 47).

F. Aldakheel, B. Hudobivnik, E. Artioli, L. Beirão da Veiga et P. Wriggers (2020). Curvilinear virtual elements for contact mechanics. *Computer Methods in Applied Mechanics and Engineering* 372, p. 113394. doi : https://doi.org/10.1016/j.cma.2020.113394 (cf. p. 50, 56).

M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes et G. N. Wells (2015). The FEniCS project version 1.5. *Archive of Numerical Software* 3, p. 9-23 (cf. p. 52).

S. Aluru et F. Sevilgen (1997). Parallel domain decomposition and load balancing using space-filling curves. *Proceedings Fourth International Conference on High-Performance Computing*. IEEE Comput. Soc., p. 230-235. doi : 10.1109/HIPC.1997.634498 (cf. p. 43).

R. Anderson, J. Andrej, A. Barker, J. Bramwell, J.-S. Camier, J. Červený, V. Dobrev, Y. Dudouit, A. Fisher, T. Kolev, W. Pazner, M. Stowell, V. Tomov, J. Dahm, D. Medina et S. Zampini (2020). MFEM : a modular finite element methods library. *Computers & Mathematics with Applications* (cf. p. 51, 52, 54, 67, 68, 86).

R. Araya et F. Chouly (août 2023). Residual a Posteriori Error Estimation for Frictional Contact with Nitsche Method. *Journal of Scientific Computing* 96.87. doi : 10.1007/s10915-023-02300-8 (cf. p. 49, 50).

D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin et D. Wells (2021). The deal.II finite element library : Design, features, and insights. *Computers & Mathematics with Applications* 81. Development and Application of Open-source Software for Problems with Numerical PDEs, p. 407-422. doi : 10.1016/j.camwa.2020.02.022 (cf. p. 37, 52).

H. Askes et A. Rodríguez-Ferran (2001). A combined rh-adaptive scheme based on domain subdivision. formulation and linear examples. *International Journal for Numerical Methods in Engineering* 51.3, p. 253-273 (cf. p. 39).
S. Attaway, B. Hendrickson et S. P. et al. (1998). A parallel contact detection algorithm for transient solid dynamics simulations using PRONTO3D. *Computational Mechanics* 22, p. 143-159 (cf. p. 32, 33).
I. Babuška, A. Miller et M. Vogelius (1983). Adaptive methods and error estimation for elliptic problems of structural mechanics. *Adaptive Computational Methods for Partial Differential Equations*, p. 57-73 (cf. p. 36, 39).
I. Babuška et B. Szabo (1982). On the rates of convergence of the Finite Element method. *International Journal for Numerical Methods in Engineering* 18.3, p. 323-341 (cf. p. 38).
M. Bader (2012). *Space-Filling Curves : An Introduction with Applications in Scientific Computing*. Springer Publishing Company, Incorporated (cf. p. 43).
D. Bai et A. Brandt (mars 1987). Local mesh refinement multilevel techniques. *SIAM Journal on Scientific and Statistical Computing* 8, p. 109-134 (cf. p. 38).
L. Barbié, I. Ramière et F. Lebon (2014). Strategies involving the Local Defect Correction multilevel refinement method for solving three-dimensional linear elastic problems. *Computers & Structures* 130, p. 73-90 (cf. p. 39).
– (2015). An automatic multilevel refinement technique based on nested local meshes for nonlinear mechanics. *Computers and Structures* 147, p. 14-25 (cf. p. 39, 64).
M. Barboteu, P. Alart et M. Vidrascu (2001). A domain decomposition strategy for non-classical frictional multi-contact problems. *Comp. Meth. Appl. Mech. Engrg* 190, p. 4785-4803 (cf. p. 52).
F. B. Belgacem, P. Hild et P. Laborde (1998). The mortar finite element method for contact problems. *Mathematical and Computer Modeling* 28, p. 263-271 (cf. p. 32, 36).
T. Belytschko, C. Parimi, N. Moës, N. Sukumar et S. Usui (2003). Structured extended finite element methods for solids defined by implicit surfaces. *International Journal for Numerical Methods in Engineering* 56, p. 609-635 (cf. p. 37).
T. Belytschko et M. Tabbara (1993). H-adaptive Finite Element methods for dynamic problems, with emphasis on localization. *International Journal for Numerical Methods in Engineering* 36.24, p. 4245-4265 (cf. p. 38).
H. Ben Dhia (1988). Modelling and solution by panalty-duality method of unilateral contact problems. *Calcul des structures et intelligence artificielle* 2, p. 1-18 (cf. p. 61).
M. J. Berger et S. H. Bokhari (1987). A partitioning strategy for nonuniform problems on multiprocessors. *IEEE Transactions on Computers* 5, p. 570-580 (cf. p. 43).
C. Bernardi, N. Debit et Y. Maday (1990). Coupling finite element and spectral methods : first results. *Mathematics of Computation* 54, p. 21-39 (cf. p. 32).
S. Bernaud, I. Ramière, G. Latu et B. Michel (2024). PLEIADES : A numerical framework dedicated to the multiphysics and multiscale nuclear fuel behavior simulation. *Annals of Nuclear Energy* 205, p. 110577. doi : https://doi.org/10.1016/j.anucene.2024.110577 (cf. p. 28).
P. Bisegna, F. Lebon et F. Maceri (2004). Relaxation procedures for solving Signorini–Coulomb contact problems. *Advances in Engineering Software* 35.10. Engineering Computational Technology, p. 595-600. doi : https://doi.org/10.1016/j.advengsoft.2004.03.018 (cf. p. 36).
P. Bisegna, F. Lebon et F. Maceri (déc. 2001). D-PANA : A convergent block-relaxation solution method for the discretized dual formulation of the Signorini-Coulomb contact

problem. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics* 333, p. 1053-1058. DOI : 10.1016/S0764-4442(01)02153-X (cf. p. 36).

H. Boffy, M.-C. Baietto, P. Sainsot et T. Lubrecht (avr. 2012). An Efficient 3D Model of Heterogeneous Materials for Elastic Contact Applications Using Multigrid Methods. *Journal of Tribology-Transactions of the Asme* 134.2, p. 021401. DOI : 10.1115/1.4006296 (cf. p. 49).

D. Böhme, F. Wolf, B. R. de Supinski, M. Schulz et M. Geimer (2012). Scalable Critical-Path Based Performance Analysis. *26th International Parallel and Distributed Processing Symposium*. IEEE, p. 1330-1340. DOI : 10.1109/IPDPS.2012.120 (cf. p. 103).

E. G. Boman, Ü. V. Çatalyürek, C. Chevalier et K. D. Devine (avr. 2012). The Zoltan and Isorropia parallel toolkits for combinatorial scientific computing : Partitioning, ordering and coloring. *Sci. Program.* 20.2, p. 129-150. DOI : 10.1155/2012/713587 (cf. p. 42, 43, 45).

R. Borrell, J. Cajas, D. Mira, A. Taha, S. Koric, M. Vázquez et G. Houzeaux (2018). Parallel mesh partitioning based on space filling curves. *Computers & Fluids* 173, p. 264-272. DOI : https://doi.org/10.1016/j.compfluid.2018.01.040 (cf. p. 98).

R. Borrell, G. Oyarzun, D. Dosimont et G. Houzeaux (2020). *Parallel SFC-based mesh partitioning and load balancing*. arXiv : 2007.03518 [physics.comp-ph] (cf. p. 98).

D. Braess, C. Carstensen et R. Hoppe (sept. 2007). Convergence analysis of a conforming adaptive finite element method for an obstacle problem. *Numerische Mathematik* 107.3, p. 455-471. DOI : 10.1007/s00211-007-0098-6 (cf. p. 49, 50).

– (mars 2009). Error Reduction in Adaptive Finite Element Approximations of Elliptic Obstacle Problems. *International Journal of Computer Mathematics - IJCM* 27.2, p. 148-169 (cf. p. 49).

A. Brandt (1977a). Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation* 31, p. 333-390 (cf. p. 38).

– (1977b). *Multi-Level Adaptive Techniques (MLAT) for Partial Differential Equations : Ideas and Software*. Sous la dir. d'e. M. S. J. R. Rice. Academic Press, p. 277-318 (cf. p. 38).

C. Burstedde (2020). Parallel tree algorithms for AMR and non-standard data access. *ACM Transactions on Mathematical Software* 46, p. 1-31 (cf. p. 51, 52).

C. Burstedde, L. C. Wilcox et O. Ghattas (2011). p4est : Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing* 33, p. 1103-1133 (cf. p. 51, 52).

P. Bussetta, D. Marceau et J.-P. Ponthot (mai 2009). Résolution du problème de contact mécanique frottant : méthode du lagrangien augmenté adapté. *Proceedings of 9th National Symposium on Computational Structural Mechanics, Giens, France*. T. 2, p. 623-628 (cf. p. 35).

A. R. Butz (1968). Space filling curves and mathematical programming. *Information and Control* 12.4, p. 314-330. DOI : https://doi.org/10.1016/S0019-9958(68)90367-7 (cf. p. 43).

P. M. Campbell, K. D. Devine, J. E. Flaherty, L. G. Gervasio et J. D. Teresco (2003). *Dynamic Octree Load Balancing Using Space-Filling Curves*. Rapp. tech. CS-03-01, Williams College Department of Computer Science (cf. p. 87).

W. Cao, W. Huang et R. Russell (1999). An r-adaptive Finite Element method based upon moving mesh PDEs. *Journal of Computational Physics* 149, p. 221-244 (cf. p. 39).

CEA (2008). *Les combustibles nucléaires*. Sous la dir. de Moniteur, p. 147 (cf. p. 27).

CEA (2021a). *CCRT - MOYENS DE CALCUL : TOPAZE*. URL : https://www-hpc.cea.fr/fr/CCRT-moyens.html (cf. p. 55).
– (2021b). *CCRT TOPAZE*. URL : https://www-ccrt.cea.fr/fr/moyen_de_calcul/index.htm (cf. p. 55).
– (2023). *Cast3M*. URL : https://www.cast3m.cea.fr (cf. p. 28, 72).
J. Červený, V. Dobrev et T. Kolev (2019). Non-conforming mesh refinement for high-order finite elements. *SIAM J. Sci. Comput.* 41.4, p. C367-C392 (cf. p. 44, 51, 54, 67, 68, 87, 90).
C. Chevalier, H. Hirtz, F. Ledoux et S. Morais (2023). Coupe : a mesh partitioning platform. *SIAM International Meshing Roundtable* (cf. p. 42).
T.-H. Chien, J.-W. Lin et R.-G. Chang (août 2018). Parallel Collision Detection with OpenMP. *Journal of Physics : Conference Series, 3rd Annual International Conference on Information System and Artificial Intelligence (ISAI2018)*. T. 1069. 1. IOP Publishing. DOI : 10.1088/1742-6596/1069/1/012180 (cf. p. 33).
F. Chouly et P. Hild (2013a). A Nitsche-based method for unilateral contact problems : numerical analysis. *SIAM J. Numer. Anal.* 51.2, p. 1295-1307 (cf. p. 35).
– (2013b). On convergence of the penalty method for unilateral contact problems. *Applied Numerical Mathematics* 65, p. 27-40 (cf. p. 34, 72-74).
F. Chouly, P. Hild, V. Lleras et Y. Renard (2022). Nitsche method for contact with Coulomb friction : Existence results for the static and dynamic finite element formulations. *J. Comput. Appl. Math.* 416.114557 (cf. p. 35).
F. Chouly, P. Hild et Y. Renard (2015). A Nitsche finite element method for dynamic contact : 2. Stability of the schemes and numerical experiments. *ESAIM Math. Model. Numer. Anal.* 49.2, p. 503-528 (cf. p. 35).
P. G. Ciarlet (1978). *The finite element method for elliptic problems*. North-Holland, Amsterdam, New York, Oxford (cf. p. 29).
M. Cihan, B. Hudobivnik, J. Korelc et P. Wriggers (2022). A virtual element method for 3D contact problems with non-conforming meshes. *Comput. Methods Appl. Mech. Engrg.* 402.115385 (cf. p. 50).
L. Clarke, I. Glendinning et R. Hempel (1994). *The mpi message passing interface standard*. In Programming environments for massively parallel distributed systems, Springer, p. 213-218 (cf. p. 41).
Connaissance des Énergies (s. d.). URL : https://www.connaissancedesenergies.org/ (cf. p. 25).
A. Curnier (1999). *Unilateral contact : Mechanical modelling*. P. Wriggers et P. Panagiotopoulos, editors, New Developments in Contact Problems, Springer, Wien, p. 1-54 (cf. p. 29).
deal.II (2023). *step 6*. URL : https://www.dealii.org/current/doxygen/deal.II/step%5C_6.html (cf. p. 53).
L. Demkowicz, P. Devloo et J. Oden (1985). On an h-type mesh-refinement strategy based on minimization of interpolation errors. *Computer Methods in Applied Mechanics and Engineering* 53, p. 67-89 (cf. p. 38).
L. Demkowicz, J. Oden, W. Rachowicz et O. Hardy (1989). Toward a universal h-p adaptive Finite Element strategy, part 1. constrained approximation and data structure. *Computer Methods in Applied Mechanics and Engineering* 77.1, p. 79-112 (cf. p. 38).
T. W. M. Devitt et T. A. Laursen (2000). A mortar-finite element formulation for frictional contact problems. *International Journal for Numerical Methods in Engineering* 48, p. 1525-1547 (cf. p. 32, 36).

P. DÌEZ et A. HUERTA (1999). A unified approach to remeshing strategies for Finite Element h-adaptivity. *Computer Methods in Applied Mechanics and Engineering* 176, p. 215-229 (cf. p. 38).

J. J. DONGARRA, L. S. DUFF, D. C. SORENSEN et H. A. van der VORST (1998). Numerical Linear Algebra for High-Performance Computers. SIAM, Philadelphia, PA, USA (cf. p. 46).

W. DÖRFLER (1996). A convergent adaptive algorithm for poisson's equation. *SIAM Journal on Numerical Analysis* 33, p. 1106-1124 (cf. p. 64).

Z. DOSTÁL, T. KOZUBEK, M. SADOWSKÁ et V. VONDRÁK (2023). *Scalable algorithms for contact problems*. Springer (cf. p. 52, 53).

G. DROUET et P. HILD (2015). Optimal Convergence for Discrete Variational Inequalities Modelling Signorini Contact in 2D and 3D without Additional Assumptions on the Unknown Contact Set. *SIAM Journal on Numerical Analysis* 53.3, p. 1488-1507. DOI : 10.1137/140980697 (cf. p. 77).

I. S. DUFF, A. M. ERISMAN et J. K. REID (1986). *Direct Methods for Sparse Matrices*. Oxford University Press, London (cf. p. 45).

D. DUREISSEIX et C. FARHAT (2001). A numerically scalable domain decomposition method for the solution of frictionless contact problems. *Int. J. Num. Meth. Engrg* 50, p. 2643-2666 (cf. p. 52).

A. DÜSTER, H. BRÖKER et E. RANK (2001). The p-version of the finite element method for three-dimensional curved thin walled structures. *International Journal for Numerical Methods in Engineering* 52.7, p. 673-703 (cf. p. 38).

A. EPALLE, I. RAMIÈRE, G. LATU et F. LEBON (juill. 2023a). Combination of adaptive mesh refinement and high performance computing for accurate solution of elastostatics contact mechanics problems. *Proceedings of the Conference Held in Turin* (cf. p. 57, 58).

– (juin 2023b). Massively Parallel Simulation and Adaptive Mesh Refinement for 3D Elastostatics Contact Mechanics Problems. *Proceedings of the Conference Held in Göteborg*. DOI : 10.23967/admos.2023.061 (cf. p. 57, 58).

– (juin 2024a). High Performance Computing Strategy for 3D Contact Mechanics Problems with Adaptive Mesh Refinement. *Proceedings of the Conference Held in Lisbon* (cf. p. 57, 58).

– (juill. 2024b). Parallel node-to-surface strategy for 3D contact mechanics problems with adaptive mesh refinement. *Proceedings of the Conference Held in Vancouver* (cf. p. 57, 58).

– (sept. 2024c). Parallel simulation and adaptive mesh refinement for 3D elastostatic contact mechanics problems between deformable bodies (cf. p. 57).

– (mai 2024d). Simulation massivement parallèle et raffinement adaptatif de maillage pour les problèmes de mécanique du contact 3D. *Proceedings of the Conference Held in Giens* (cf. p. 57, 58).

P. E. FARRELL, S. MICHELETTI et S. PEROTTO (2011). An anisotropic Zienkiewicz–Zhu-type error estimator for 3D applications. *International Journal for Numerical Methods in Engineering* 85.6, p. 671-692. DOI : 10.1002/nme.2980 (cf. p. 55).

M. FERNANDO et H. SUNDAR (2020). *Scalable Local Timestepping on Octree Grids*. arXiv : 2011.10570 [cs.MS] (cf. p. 98).

J. FISH (1992). The s-version of the Finite Element method. *Computers & Structures* 43.3, p. 539-547 (cf. p. 39).

J. FLAHERTY, R. LOY, M. SHEPHARD, B. SZYMANSKI, J. TERESCO et L. ZIANTZ (1997). Adaptive Local Refinement with Octree Load Balancing for the Parallel Solution of

Three-Dimensional Conservation Laws. *J. Parallel Distrib. Comput.* 47, p. 139-152 (cf. p. 51).
A. Francavilla et O. C. Zienkiewicz (1975). A note on numerical computation of elastic contact problems. *International Journal for Numerical Methods in Engineering* 9.4, p. 913-924. doi : 10.1002/NME.1620090410 (cf. p. 31).
A. Franceschini, M. Ferronato, M. Frigo et C. Janna (2022). A reverse augmented constraint preconditioner for Lagrange multiplier methods in contact mechanics. *Comput. Methods Appl. Mech. Eng.* 392 (cf. p. 52).
D. Franke, A. Düster, V. Nübel et E. Rank (2010). Erratum to : A comparison of the h-, p-, hp-, and rp-version of the FEM for the solution of the 2d hertzian contact problem. *Computational Mechanics* 45, p. 513-522 (cf. p. 39).
A. Frederiksen, OleSigmund et K. Poulios (2024). Topology Optimization of self-contacting structures. *Computational Mechanics* 73, p. 967-981. doi : 10.1007/s00466-023-02396-7 (cf. p. 35).
J. Frohne, T. Heister et W. Bangerth (2016). Efficient numerical methods for the large-scale, parallel solution of elastoplastic contact problems. *Int. J. Numer. Methods Eng.* 105, p. 416-439 (cf. p. 53).
J. Gaidamour (déc. 2009). Conception d'un solveur linéaire creux parallèle hybride direct-itératif. Theses. Université Sciences et Technologies - Bordeaux I (cf. p. 48).
A. George et J. W. H. Liu (1981). *Computer Solution of Large Sparse Positive Definite Systems*. Prentice Hall Professional Technical Reference (cf. p. 45, 46).
C. Geuzaine et J.-F. Remacle (2009). Gmsh : a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* 79.11, p. 1309-1331 (cf. p. 67, 72, 92).
S. Ghosh et S. Manna (1993). R-adapted Arbitrary Lagrangian-Eulerian Finite Element method in metal-forming simulation. *Journal of Materials Engineering and Performance* 2.2, p. 271-282 (cf. p. 39).
S. Ghosh et S. Raju (1996). R–s adapted Arbitrary Lagrangian-Eulerian Finite Element method for metal-forming problems with strain localization. *International Journal for Numerical Methods in Engineering* 39.19, p. 3247-3272 (cf. p. 39).
R. Glowinski et P. L. Tallec (1989). *Augmented Lagrangians and Operator-Splitting Methods in Nonlinear Mechanics*. SIAM, Philadelphia (cf. p. 35).
G. Guillamet, M. Rivero, M. Zavala-Aké, M. Vázquez, G. Houzeaux et S. Oller (2022). A parallel algorithm for unilateral contact problems. *Computers & Structures* 271 (cf. p. 52, 138).
T. Gustafsson, R. Stenberg et J. Videman (2020). On Nitsche's Method for Elastic Contact Problems. *SIAM Journal on Scientific Computing* 42.2, B425-B446. doi : 10.1137/19M1246869 (cf. p. 49, 50).
– (2021). Nitsche's Master-Slave Method for Elastic Contact Problems. *Numerical Mathematics and Advanced Applications ENUMATH 2019*. Sous la dir. de F. J. Vermolen et C. Vuik. Springer International Publishing, p. 899-908 (cf. p. 49).
J. O. Hallquist, G. L. Goudreau et D. J. Benson (1985). Sliding interfaces with contact-impact in large-scale Lagrangian computations. *Computer Methods in Applied Mechanics and Engineering* 51.1-3, p. 107-137 (cf. p. 32).
J. Har et R. E. Fulton (2003). A parallel finite element procedure for contact-impact problems. *Engineering with Computers* 19, p. 67-84 (cf. p. 52).
F. Hecht (2012). New development in Freefem++. *J. Numer. Math.* 20, p. 251-265 (cf. p. 52).

T. Helfer et G. Latu (oct. 2021). The MFEM-MGIS project : coupling MFEM and MGIS for high performance nonlinear simulations. *Conference : 7th MFront User Meeting* (cf. p. 153).

T. Helfer (s. d.). *The MFEM/MGIS project*. URL : https://thelfer.github.io/mfem-mgis/ (cf. p. 153).

B. Hendrickson (2011). Chaco. *Encyclopedia of Parallel Computing*. Springer US, p. 248-249 (cf. p. 45).

B. Hendrickson (1998). Graph partitioning and parallel solvers : Has the emperor no clothes ? *Solving Irregularly Structured Problems in Parallel*. Sous la dir. d'A. Ferreira, J. Rolim, H. Simon et S.-H. Teng. Berlin, Heidelberg : Springer Berlin Heidelberg, p. 218-225 (cf. p. 97).

P. Hennig, M. Kastner, P. Morgenstern et D. Peterseim (2016). Adaptive Mesh Refinement Strategies in Isogeometric Analysis - A Computational Comparison. *Computer Methods in Applied Mechanics and Engineering* 316, p. 424-448 (cf. p. 36).

V. E. Henson et U. M. Yang (avr. 2002). BoomerAMG : a parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.* 41.1, p. 155-177. DOI : 10.1016/S0168-9274(01)00115-5 (cf. p. 48).

H. Hertz (1882). *On the Contact of Rigid Elastic Solids and on Hardness*. Macmillan et Co., New York (cf. p. 70, 76).

P. Hild et V. Lleras (2009). Residual Error Estimators for Coulomb Friction. *SIAM Journal on Numerical Analysis* 47.5, p. 3550-3583. DOI : 10.1137/070711554 (cf. p. 49).

H. Hirshikesh, A. L. N. Pramod, E. T. Ooi, C. Song et S. Natarajan (2021). An adaptive scaled boundary finite element method for contact analysis. *Eur. J. Mech. A. Solids* 86.104180 (cf. p. 29, 50).

T. J. R. Hughes, R. M. Ferencz et J. O. Hallquist (1987). Large-scale vectorized implicit calculations in solid mechanics on a Cray X-MP/48 utilizing EBE preconditioned conjugate gradients. *Computer Methods in Applied Mechanics and Engineering* 61.2, p. 215-248. DOI : https://doi.org/10.1016/0045-7825(87)90005-3 (cf. p. 47).

T. J. R. Hughes, R. L. Taylor et W. Kanoknukulchai (1977). *A finite element method for large displacement contact and impact problems*. Sous la dir. de K. Bathe, J. T. Oden, W. Wunderlich et E. W. (eds). Formulations et Computational Algorithms in FE Analysis, MIT Press, p. 468-495 (cf. p. 31).

INL (2023). *MOOSE*. URL : https://mooseframework.inl.gov/application%5C_usage/hypre.html (cf. p. 48, 105, 107).

C. Introïni, I. Ramière, J. Sercombe, B. Michel, T. Helfer et J. Fauque (2024). ALCYONE : the fuel performance code of the PLEIADES platform dedicated to PWR fuel rods behavior. *Annals of Nuclear Energy* 207, p. 110711. DOI : https://doi.org/10.1016/j.anucene.2024.110711 (cf. p. 28).

L. Kaczmarczyk, Z. Ullah, K. Lewandowski, X. Meng, X.-Y. Zhou, I. Athanasiadis, H. Nguyen, C.-A. Chalons-Mouriesse, E. J. Richardson, E. Miur, A. G. Shvarts, M. Wakeni et C. J. Pearce (2020). MoFEM : An Open Source, Parallel Finite Element Library. *Journal of Open Source Software* 5, p. 1441/1-8 (cf. p. 52).

G. Karypis, R. Aggarwal, V. Kumar et S. Shekhar (1999). Multilevel hypergraph partitioning : applications in vlsi domain. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 7.1, p. 69-79 (cf. p. 45).

G. Karypis et V. Kumar (2000). Multilevel k-way hypergraph partitioning. *VLSI design* 11.3, p. 285-300 (cf. p. 45).

G. Karypis, K. Schloegel et V. Kumar (1997). *Parmetis : Parallel graph partitioning and sparse matrix ordering library. Version 1.0*. Rapp. tech. Dept. of Computer Science, University of Minnesota, p. 22 (cf. p. 45).
G. Karypis (2003). *Multi-constraint mesh partitioning for contact/impact computations*. Rapp. tech. Department of Computer Science et Engineering, Minneapolis, MN (cf. p. 42).
G. Karypis et V. Kumar (1998). *METIS : A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*. Rapp. tech. University of Minnesota, Department of Computer Science / Army HPC Research Center Minneapolis, MN 55455 (cf. p. 45).
D. W. Kelly, J. P. De S. R. Gago, O. C. Zienkiewicz et I. Babuska (1983). A posteriori error analysis and adaptive processes in the finite element method : Part I—error analysis. *International Journal for Numerical Methods in Engineering* 19.11, p. 1593-1619. doi : 10.1002/nme.1620191103 (cf. p. 53).
J. D. Keyser et R. D (1992). *Grid partitioning by inertial recursive bisection*. Rapp. tech. Report TW 174, Katholieke Universiteit Leuven, Belgium (cf. p. 43).
K. Khadra, P. Angot, J. P. Caltagirone et P. Morel (1996). Concept de zoom adaptatif en architecture multigrille locale ; étude comparative des méthodes L.D.C., F.A.C. et F.I.C. *ESAIM : Mathematical Modelling and Numerical Analysis Modélisation Mathématique et Analyse Numérique* 30.1, p. 39-82 (cf. p. 38).
N. Kikuchi et J. T. Oden (1988). *Contact problems in elasticity : a study of variational inequalities and finite element methods*. T. 8. SIAM Studies in Applied Mathematics, Society for Industrial et Applied Mathematics (SIAM), Philadelphia, PA (cf. p. 33, 34, 71).
N. Kikuchi et Y. Song (1981). Penalty/finite-element approximations of a class of unilateral problems in linear elasticity. *Quart. Appl. Math.* 39 (1), p. 1-22 (cf. p. 33).
B. S. Kirk, J. W. Peterson, R. H. Stogner et G. F. Carey (déc. 2006). libMesh : A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. *Engineering with Computers* 22, p. 237-254 (cf. p. 52).
D. Koliesnikova (2021). Raffinement adaptatif automatique de maillage à précision contrôlée dans un contexte multiéchelle pour la mécanique des solides non linéaire. Thèse de doct. Aix-Marseille Université (cf. p. 26, 37).
D. Koliesnikova, I. Ramière et F. Lebon (2021). A unified framework for the computational comparison of adaptive mesh refinement strategies for all-quadrilateral and all-hexahedral meshes : Locally adaptive multigrid methods versus h-adaptive methods. *Journal of Computational Physics* 437 (cf. p. 38, 39, 54, 56, 65, 66, 79).
P. Ladevèze et P. Rougeot (1997). New advances on a posteriori error on constitutive relation in finite element analysis. *Comput. Meth. Appl. Mech. Engrg.* 150, p. 239-249 (cf. p. 39).
M. Lainet, B. Michel, J.-C. Dumas, M. Pelletier et I. Ramière (2019). GERMINAL, a fuel performance code of the PLEIADES platform to simulate the in-pile behaviour of mixed oxide fuel pins for sodium-cooled fast reactors. *Journal of Nuclear Materials* 516, p. 30-53 (cf. p. 28).
T. A. Laursen (2002). *Computational contact and impact mechanics*. Springer-Verlag, Berlin (cf. p. 33, 71).
F. Ledoux et J. F. Shepherd (2009). Topological modifications of hexahedral meshes via sheet operations : a theoretical study. *Engineering with Computers* 26, p. 433-447 (cf. p. 38).

M. Lieber et W. E. Nagel (2018). Highly scalable SFC-based dynamic load balancing and its application to atmospheric modeling. *Future Generation Computer Systems* 82, p. 575-590. doi : https://doi.org/10.1016/j.future.2017.04.042 (cf. p. 98).

H. Liu, I. Ramière et F. Lebon (2017). On the coupling of local multilevel mesh refinement and ZZ methods for unilateral frictional contact problems in elastostatics. *Computer Methods in Applied Mechanics and Engineering* 323, p. 1-26 (cf. p. 37, 39, 49, 56, 63, 65, 66).

H. Liu, Q. Liu, H. Ma et J. Fish (2021). A novel GPGPU-parallelized contact detection algorithm for combined finite-discrete element method. *International Journal of Rock Mechanics and Mining Sciences* 144. doi : https://doi.org/10.1016/j.ijrmms.2021.104782 (cf. p. 33).

D. G. Luenberger (2003). *Linear and Nonlinear Programming, 2nd ed.* Kluwer Academic Publishers (cf. p. 36, 60).

D. G. Luenberger et Y. Ye (jan. 1984). *Linear and Nonlinear Programming*. T. 67. 2. doi : 10.2307/1240727 (cf. p. 36, 55, 60).

M. Mayr et A. Popp (2023). Scalable computational kernels for mortar finite element methods. *Engineering with Computers* 39, p. 3691-3720 (cf. p. 32, 52, 55, 138, 139, 142, 145, 147).

M. McCool, J. Reinders et A. Robison (2012). *Structured parallel programming : patterns for efficient computation*. Elsevier (cf. p. 41).

H. Meyerhenke, B. Monien et S. Schamberger (2006). Accelerating shape optimizing load balancing for parallel FEM simulations by algebraic multigrid. *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*. doi : 10.1109/IPDPS.2006.1639295 (cf. p. 97).

B. Michel, C. Nonon, J. Sercombe, F. Michel et V. Marelle (2013). Simulation of pellet-cladding interaction with the pleiades fuel performance software environment. *Nuclear Technology* 182, p. 124-137 (cf. p. 37).

A. R. Mijar et J. S. Arora (2004a). An augmented Lagrangian optimization method for contact analysis problems, 1 : formulation and algorithm. *Structural and Multidisciplinary Optimization* 28, p. 99-112 (cf. p. 35).

– (2004b). An augmented Lagrangian optimization method for contact analysis problems, 2 : numerical evaluation. *Structural and Multidisciplinary Optimization* 28, p. 113-126 (cf. p. 35).

W. F. Mitchell (2007). A refinement-tree based partitioning method for dynamic load balancing with adaptively refined grids. *J. Parallel Distrib. Comput.* 67, p. 417-429 (cf. p. 51).

W. F. Mitchell et M. A. McClain (2014). A comparison of hp-adaptive strategies for Elliptic Partial Differential equations. *ACM Transactions on Mathematical Software* 41.1, p. 1-39 (cf. p. 38).

J. J. Moreau (1974). *On unilateral constraints, friction and plasticity*. G. Capriz et G. Stampacchia, editors, New Variational Techniques in Mathematical Physics, CIME, Roma, p. 175-322 (cf. p. 29).

G. M. Morton (1966). *A computer oriented geodetic data base and a new technique in file sequencing*. Rapp. tech. International Business Machines Company New York (cf. p. 44).

A. Motte, B. Prabel, C. Bovet, V. Chiaruttini et O. Jamond (mai 2024). Détection des contacts en mémoire massivement distribuée par un double parcours de Bounding Volume Hierarchies. *16ème Colloque National en Calcul de Structures*. CNRS, CSMA, ENS Paris-Saclay, CentraleSupélec. Giens, France (cf. p. 33).

P. Munch, K. Ljungkvist et M. Kronbichler (2022). Efficient application of hanging-node constraints for matrix-free high-order FEM computations on CPU and GPU. *Proceedings of the High performance computing : 37th international conference, ISC High Performance, Lecture Notes in Computer Science*. T. 13289. Springer (cf. p. 51).
K. G. Murty (1988). *Linear Complementarity, Linear and Nonlinear Programming*. Helderman-Verlag (cf. p. 36, 60).
F. Nataf et P.-H. Tournier (2023). A GenEO Domain Decomposition method for Saddle Point problems. en. *Comptes Rendus. Mécanique* 351.S1, p. 667-684. doi : 10.5802/crmeca.175 (cf. p. 55).
G. Nicolas, T. Fouquet, S. Geniaut et S. Cuvilliezam (2016). Improved adaptive mesh refinement for conformal hexahedral meshes. *Advances in Engineering Software* 102, p. 14-28 (cf. p. 38).
J. Nitsche (1971). Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen in der Mathematik an der Universität Hamburg* 36, p. 9-15 (cf. p. 35).
R. H. Nochetto et A. Veeser (2012). *Primer of Adaptive Finite Element Methods*. Springer Berlin Heidelberg (cf. p. 54, 66).
V. Nübel, A. Düster et E. Rank (2007). An rp-adaptive finite element method for the deformation theory of plasticity. *Computational Mechanics* 39.5, p. 557-574 (cf. p. 39).
K. Ohtake, J. Oden et N. Kikuchi (1980). Analysis of certain unilateral problems in von karman plate theory by a penalty method-part 2. Approximation and numerical analysis. *Computer Methods in Applied Mechanics and Engineering* 24.3, p. 317-337. doi : https://doi.org/10.1016/0045-7825(80)90068-7 (cf. p. 61).
G. Peano (1890). Sur une courbe, qui remplit toute une aire plane. *Mathematische Annalen* 36.1, p. 157-160 (cf. p. 43).
F. Pellegrini et J. Roman (1996). Scotch : A Software Package for Static Mapping by Dual Recursive Bipartitioning of Process and Architecture Graphs. *High-Performance Computing and Networking : International Conference and Exhibition HPCN EUROPE 1996 Brussels, Belgium, April 15–19, 1996 Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, p. 493-498 (cf. p. 45).
G. Pietrzak (1997). Continuum mechanics modelling and augmented Lagrangian formulation of large deformation frictional contact problems. Thèse de doct. École Polytechnique Fédérale de Lausanne (cf. p. 35).
G. Pietrzak et A. Curnier (1999). Large deformation frictional contact mechanics : continuum formulation and augmented Lagrangian treatment. *Computer Methods in Applied Mechanics and Engineering* 177, p. 351-381 (cf. p. 35).
M. Piggott, C. Pain, G. Gorman, P. W. Power et A. Goddard (2005). h, r, and hr adaptivity with applications in numerical ocean modelling. *Ocean Modelling* 10, p. 95-113 (cf. p. 39).
J. R. Pilkington et S. B. Baden (1994). *Partitioning with Spacefilling Curves*. Rapp. tech. Department of Computer Science et Engineering, University of California, San Diego (cf. p. 43).
R. Prat (2019). Équilibrage dynamique de charge sur supercalculateur exaflopique appliqué à la dynamique moléculaire. Thèse de doct. Université de Bordeaux (cf. p. 40, 42).
M. A. Puso (2004). A 3D mortar method for solid mechanics. *IJNME* 59.3, p. 315-336 (cf. p. 32).

A. Rademacher (juill. 2019). Mesh and model adaptivity for frictional contact problems. *Numerische Mathematik* 142.3, p. 465-523. doi : 10.1007/s00211-019-01044-8 (cf. p. 50).

S. Rajamanickam, E. G. Boman, K. D. Devine, V. J. Leung, L. A. Riesen, U. Catalyurek, D. Bozdag, C. Chevalier et M. Wolf (2011). *Tutorial : The zoltan toolkit*. Rapp. tech. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States) (cf. p. 42, 44, 45).

I. Ramière, P. Angot et M. Belliard (2007). A fictitious domain approach with spread interface for elliptic problems with general boundary conditions. *Computer Methods in Applied Mechanics and Engineering* 196, p. 766-781 (cf. p. 37).

I. Ramière, H. Liu et F. Lebon (2019). Original geometrical stopping criteria associated to multilevel adaptive mesh refinement for problems with local singularities. *Computational Mechanics* (cf. p. 64-66).

V. Roda-Casanova et F. Sanchez-Marin (2018). An adaptive mesh refinement approach for solving non-Hertzian elastic contact problems. *Meccanica* 53, p. 2013-2028. doi : 10.1007/s11012-017-0806-y (cf. p. 50).

J. Ródenas, J. Albelda, M. Tur et F. Fuenmayor (2017). A hierarchical h-adaptivity methodology based on element subdivision. *Revista UIS Ingenierías* 16, p. 263-280 (cf. p. 38, 54).

Y. Saad (2003). *Iterative methods for sparse linear systems*. SIAM, Philadelphia, PA, second edition (cf. p. 48).

R. R. Salagame et A. D. Belegundu (1997). A simple p-adaptive refinement procedure for structural shape optimization. *Finite Element in Analysis and Design* 24.3, p. 133-155 (cf. p. 38).

R. Schneiders (1998). Refining quadrilateral and hexahedral element meshes. *Fifth International Meshing Roundtable, Germany*. T. 1 (cf. p. 38).

J. Sercombe, J. Julien, F. Michel, B. Michel et E. Fédérici (sept. 2013). 3D modelling of strain concentration due to PCI within the fuel code ALCYONE (cf. p. 28).

J. C. Simo et T. A. Laursen (1992). An augmented Lagrangian treatment of contact problems involving friction. *Comput. Struct.* 42, p. 97-116 (cf. p. 35).

J. C. Simo, P. Wriggers et R. L. Taylor (1985). A perturbed Lagrangian formulation for the finite element solution of contact problems. *Computer Methods in Applied Mechanics and Engineering* 50, p. 163-180 (cf. p. 32, 35).

H. D. Simon (1991). Partitioning of unstructured problems for parallel processing. *Computing systems in engineering* 2.2-3, p. 135-148 (cf. p. 43).

C. Song et J. P. Wolf (1997). The scaled boundary finite-element method—alias consistent infinitesimal finite-element cell method—for elastodynamics. *Computer Methods in Applied Mechanics and Engineering* 147.3, p. 329-355. doi : 10.1016/S0045-7825(97)00021-2 (cf. p. 29).

C. Song, E. T. Ooi, A. L. N. Pramod et S. Natarajan (2018). A novel error indicator and an adaptive refinement technique using the scaled boundary finite element method. *Engineering Analysis with Boundary Elements* 94, p. 10-24. doi : https://doi.org/10.1016/j.enganabound.2018.05.010 (cf. p. 50).

H. Sundar, R. Sampath et G. Biros (2008). Bottom-up construction and 2 :1 balance refinement of linear octrees in parallel. *SIAM Journal on Scientific Computing* 30, p. 2675-2708 (cf. p. 51, 52).

B. Szabo et I. Babuška (1990). *Finite Element analysis*. T. 1. Wiley, New York (cf. p. 38).

T. Takahashi, M. Pierre et A. Henrot (2014). *Analyse Numérique*. Nancy, France (cf. p. 46, 101).
R. L. Taylor et O. Papadopoulos (1991). *On a patch test for contact problems in two dimensions*. Sous la dir. de P. Wriggers et W. W. (eds). Nonlinear Computational Mechanics, Springer, p. 690-702 (cf. p. 31).
V. E. Taylor et B. Nour-Omid (1994). A study of the factorization fill-in for a parallel implementation of the finite element method. *International journal for numerical methods in engineering* 37.22, p. 3809-3823 (cf. p. 43).
K. Tchon, J. Dompierre et R. Camarero (2002). Conformal refinement of all-quadrilateral and all-hexahedral meshes according to an anisotropic metric. *11th International Meshing Roundtable*, p. 231-242 (cf. p. 38).
T. Tu, D. O'Hallaron et O. Ghattas (2005). Scalable parallel octree meshing for terascale applications. *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*. IEEE Computer Society (cf. p. 51).
L. B. da Veiga, F. Brezzi et L. Marini (2013). Virtual elements for linear elasticity problems. *SIAM J. Numer. Anal.* 51 (cf. p. 29).
R. Verfürth (1996). A review of a posteriori error estimation and adaptive mesh-refinement techniques. *Advances in Numerical Mathematics* (cf. p. 36, 37, 54).
T. A. Wiesner, M. Mayr, A. Popp, M. W. Gee et W. A. Wall (2021). Algebraic multigrid methods for saddle point systems arising from mortar contact formulations. *International Journal for Numerical Methods in Engineering* 122.15, p. 3749-3779. doi : 10.1002/nme.6680 (cf. p. 52, 55).
R. D. Williams (1991). Performance of dynamic load balancing algorithms for unstructured mesh calculations. *Concurr. Pract. Exp.* 3, p. 457-481 (cf. p. 43).
B. Wohlmuth, A. Popp, M. Gee et W. Wall (juin 2012). An abstract framework for a priori estimates for contact problems in 3D with quadratic finite elements. *Computational Mechanics* 49.6. doi : 10.1007/s00466-012-0704-z (cf. p. 77).
P. Wriggers (2006). *Computational contact mechanics. Second edition*. Springer-Verlag (cf. p. 29, 32, 33, 36, 60, 71).
P. Wriggers, F. Aldakheel et B. Hudobivnik (2023). *Virtual Element Methods in Engineering Sciences*. Springer (cf. p. 29).
P. Wriggers, W. Rust et B. Reddy (2016). A virtual element method for contact. *Comput. Mech.* 58 (cf. p. 29).
P. Wriggers et O. Scherf (1998). Different a posteriori error estimators and indicators for contact problems. *Mathematical and Computer Modelling* 28.4-8, p. 437-447 (cf. p. 49, 56).
P. Wriggers, J. Schröder et A. Schwarz (2013). A finite element method for contact using a third medium. *Computational Mechanics* 52.4, p. 837-847 (cf. p. 35).
P. Wriggers et G. Zavarise (2008). A formulation for frictionless contact problems using a weak form introduced by Nitsche. *Comput. Mech.* 41, p. 407-420 (cf. p. 35).
V. A. Yastrebov, G. Cailletaud et F. Feyel (2011). A Local Contact Detection Technique for Very Large Contact and Self-Contact Problems : Sequential and Parallel Implementations. *Trends in Computational Contact Mechanics*. Sous la dir. de G. Zavarise et P. Wriggers. Berlin, Heidelberg : Springer Berlin Heidelberg, p. 227-251. doi : 10.1007/978-3-642-22167-5_13 (cf. p. 33).
V. A. Yastrebov (juin 2010). *Introduction to Computational Contact Mechanics - Part I. Basics*. WEMESURF short course on contact mechanics and tribology. Paris, France (cf. p. 31, 32).

Z. Yue et D. Robbins (2005). Adaptive superposition of Finite Element meshes in elastodynamic problems. *International Journal for Numerical Methods in Engineering* 63.11, p. 1604-1635 (cf. p. 39).

G. Zavarise (2015). The shifted penalty method. *Comput. Mech.* 56, p. 1-17 (cf. p. 34, 71).

G. Zavarise et L. D. Lorenzis (2009). A modified node-to-segment algorithm passing the contact patch test. *International Journal for Numerical Methods in Engineering* 79, p. 379-416 (cf. p. 32).

J. Z. Zhu, E. Hinton et O. C. Zienkiewicz (1993). Mesh enrichment against mesh regeneration using quadrilateral elements. *Communications in Numerical Methods in Engineering* 9, p. 547-554 (cf. p. 38, 54).

O. C. Zienkiewicz et J. Z. Zhu (1987). A simple error estimator and adaptive procedure for practical engineerng analysis. *International Journal for Numerical Methods in Engineering* 24, p. 337-357 (cf. p. 37, 49, 54, 63, 64, 66).

– (1992). The superconvergent patch recovery and a posteriori error estimates. Part 1 : The recovery technique. *International Journal for Numerical Methods in Engineering* 33, p. 1331-1364 (cf. p. 39, 54).

O. Zienkiewicz, J. Zhu et N. Gong (1989). Effective and practical h–p-version adaptive analysis procedures for the Finite Element method. *International Journal for Numerical Methods in Engineering* 28.4, p. 879-891 (cf. p. 38).