CSMA 2026
17ème Colloque National en Calcul des Structures
18-22 Mai 2026, Presqu'île de Giens (Var)

Partitionnement Space Filling Curve pour la simulation en mécanique du contact 3D avec raffinement adaptatif de maillage

A. Epalle[1,2], G. Latu[1], I. Ramière[1], F. Lebon[2]

[1] *CEA, DES, IRESNE, DEC, SESC ; Cadarache, F-13108 Saint-Paul-Lez Durance, [guillaume.latu,isabelle.ramiere]@cea.fr*
[2] *Aix-Marseille Université, CNRS, Centrale Marseille, LMA ; F-13453 Marseille cedex 13, lebon@lma.cnrs-mrs.fr*

Résumé — La simulation de problèmes de grandes tailles en mécanique du contact reste un challenge numérique. Afin de représenter la solution avec une précision accrue, une stratégie combinant raffinement adaptatif de maillage et partitionnement parallèle est proposée. Un outil performant et extensible sur supercalculateur est ainsi obtenu pour du contact pénalisé nœud-à-nœud entre solides 3D déformables.
Mots clés — Contact élastostatique, Pénalisation, Solides déformables, Raffinement de maillage adaptatif, Partitionnement parallèle, Calcul Haute Performance.

1 Contexte et objectifs

Paralléliser efficacement des méthodes de raffinement adaptatif de maillage (AMR – *Adaptive Mesh Refinement* en anglais) et de résolution du contact mécanique 3D est une étape essentielle pour aller vers des simulations complexes qui dépassent les niveaux de performance actuels. Or, combiner AMR, contact et parallélisation reste un défi scientifique. Les développements proposés dans la littérature se concentrent généralement uniquement sur la combinaison de deux des trois ingrédients précédents.

Nous avons proposé récemment dans [1] une stratégie permettant de résoudre dans un environnement parallèle, au moyen d'AMR, des problèmes de contact 3D entre solides déformables comportant jusqu'à des dizaines de millions de degrés de liberté (DDL). Nous avons introduit la résolution du contact dans une librairie éléments finis possédant déjà des structures de données permettant de faire de l'AMR en parallèle, à savoir MFEM [2]. L'une des particularités de notre stratégie parallèle est que les nœuds de contact appairés sont hébergés par les mêmes tâches MPI, ce qui réduit le nombre d'échanges entre les processus pour construire l'opérateur de contact. Cependant les performances parallèles (scalabilité) obtenues dans [1] sont limitées à 512 processus CPU car le partitionnement utilisé conduit à regrouper sur un processus MPI des éléments dispersés (non contigus) dans le maillage. Cela entraîne des frontières importantes entre les régions parallèles, et donc des coûts de communication supplémentaires.

L'objectif de ce travail est d'étendre les capacités parallèles de notre stratégie de résolution en proposant un partitionnement parallèle mieux adapté à la résolution du contact entre solides déformables. Ce partionnement consiste en une extension du partitionnement SFC (*Space Filling Curve*, en anglais) [3] permettant de conserver les nœuds de contact appairés sur le même processus MPI [4].

Le reste de cet article est organisé comme suit. La section 2 reprend brièvement les éléments clés de la résolution du contact et de la stratégie AMR choisies. La section 3 est consacrée à la description des stratégies de partitonnement adaptées au contact, avec un focus sur la nouvelle approche proposée dans ce travail. Enfin, la section 4 permet d'apprécier les performances obtenues, avec une bonne scalabilité jusqu'à 8192 processus avec l'approche SFC dédiée au contact.

2 Résolution du contact et raffinement adaptatif du maillage

Méthode de pénalisation et appariemment nœud-à-nœud Dans l'optique de pouvoir utiliser des solveurs linéaires scalables existants, une stratégie de résolution du contact par pénalisation, avec mise à jour itérative des statuts de contact, est adoptée [5]. En effet, elle permet de se ramener à un problème comportant uniquement des DDL primaux et une structure standard de la matrice éléments finis du système linéaire. Le système à résoudre dans le cas du contact élastostatique s'écrit alors :

$$\left(K + k_N B^T B\right) U = L^{ext} + k_N B^T D \tag{1}$$

avec U le vecteur solution (déplacement) du problème discret, K la matrice de rigidité (ou raideur), B la matrice d'appariement de contact, D le vecteur des jeux initiaux entre nœuds de contact appairés, $k_N \gg 1$ N/m le coefficient de pénalisation et L^{ext} le vecteur des forces externes généralisées.
La définition de la matrice B s'appuie sur une stratégie d'appariemment de type nœud-à-nœud afin de pouvoir construire B de manière locale sur chaque processus MPI sans devoir faire appel à des communications inter-processus, cf. section 3. Le système linéaire (1) est alors résolu via la classe `PenaltyConstrainedSolver` de l'environnement logiciel MFEM.

Raffinement h-adaptatif sur maillages hexaédriques Les maillages considérés dans ce travail sont des maillages structurés composés uniquement d'éléments hexaédriques. Ces derniers possèdent en effet de bonnes propriétés numériques et informatiques [6]. Des éléments finis de type Q_1 sont utilisés.

En ce qui concerne la stratégie AMR, une solution de raffinement de maillage de type h-adaptatif hiérarchique non-conforme est appliquée. Cette approche a été choisie car elle s'est déjà avérée fortement scalable dans d'autres contextes [7, 2] au sein de l'environnement logiciel MFEM. Le processus de raffinement adaptatif du maillage est réalisé à l'aide d'un algorithme itératif de type ESTIMATE-MARK-REFINE [6]. Les ingrédients clés utilisés ici sont :

— Estimateur d'erreur *a posteriori* de Zienkiewicz et Zhu [8], appelé estimateur ZZ par la suite, appliqué sur chaque solide indépendamment.

— Critère de marquage de type ZZ :

$$\mathcal{M}^0 = \left\{ T_i \in \Omega^h \;/\; \xi_{T_i} > e_\Omega \frac{\omega_\Omega}{\sqrt{N_E}} \right\}. \tag{2}$$

avec Ω^h le domaine discrétisé, ξ_{T_i} l'erreur élémentaire locale pour un élément de maillage T_i définie par l'équation (3), 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 [8] rappelée en (4).

$$\xi_{T_i} = \left(\int_{T_i} \left(\sigma^\star - \sigma^h \right) : \left(\varepsilon^\star - \varepsilon^h \right) dT_i \right)^{1/2}, \tag{3}$$

avec σ^h, ε^h : champs de contraintes et déformations éléments finis, et $\sigma^\star$, $\varepsilon^\star$: champs de contraintes et déformation lissés.

$$\omega_\Omega = \left(\sum_{T_i \in \Omega^h} \omega_{T_i}^2 \right)^{1/2} ; \quad \omega_{T_i} = \left(\int_{T_i} \sigma^h : \varepsilon^h dT_i + \int_{T_i} \left(\sigma^\star - \sigma^h \right) : \left(\varepsilon^\star - \varepsilon^h \right) dT_i \right)^{1/2}. \tag{4}$$

— Élargissement de la zone détectée afin de garantir que les bords de contact raffinés restent concordants (ce qui est nécessaire pour l'appariement de contact nœud-à-nœud) :

$$\mathcal{M} = \mathcal{M}^0 \cup \left\{ T_j \in \Omega^h \;/\; T_i \in \mathcal{M}^0 \text{ appairé avec } T_j \right\}. \tag{5}$$

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

— Raffinement h-local hiérarchique 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". Cette règle s'est montrée très efficace dans des travaux précédents [6]. Le système linéaire est restreint aux DDL conformes via un opérateur de prolongation [7].

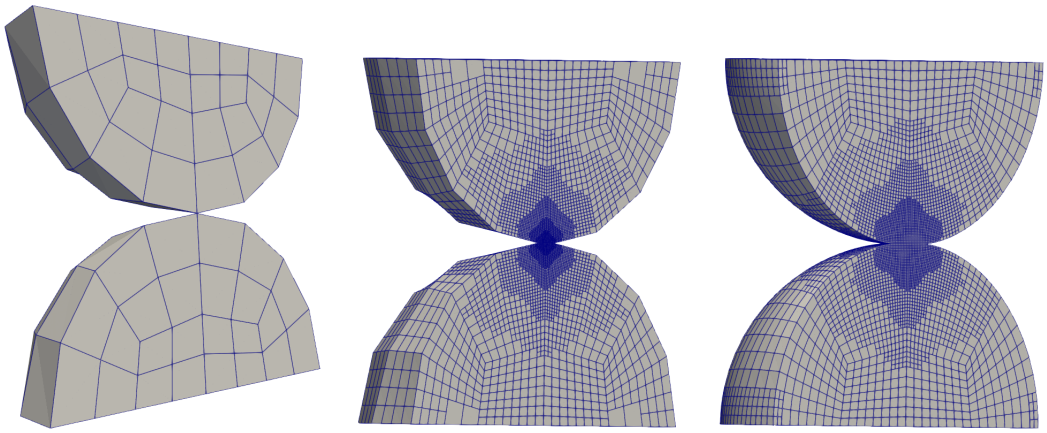
— Critère d'arrêt global :

$$\frac{\xi_\Omega}{e_\Omega \omega_\Omega} \leq 1, \quad \text{avec} \quad \xi_\Omega = \left(\sum_{T_i \in \Omega^h} \xi_{T_i}^2 \right)^{1/2}. \tag{6}$$

En effet, le critère AMR d'arrêt standard $\mathcal{M} = \emptyset$ n'est généralement jamais atteint pour des problèmes avec singularités tels que ceux de mécanique du contact considérés ici.

La procédure de raffinement de MFEM, `Apply`, est invoquée pour gérer la stratégie de raffinement de maillage h-adaptative non-conforme choisie.

Transformation super-paramétrique Le raffinement hiérarchique du maillage limite la représentation de la géométrie de frontières courbes, ce qui est généralement le cas des frontières de contact. 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 isoparamétrique d'ordre un, voir la figure 1. Ce type d'élément super-paramétrique est défini lors de la phase de génération du maillage par le mailleur Gmsh [9] et est correctement pris en charge par l'environnement logiciel MFEM.



(a) Maillage grossier initial (b) Éléments isoparamétriques (transformation linéaire) (c) Éléments super-paramétriques (transformation d'ordre 6)

FIGURE 1 – AMR hiérarchique pour problèmes de contact à bords courbes avec solution éléments finis d'ordre un et erreur voulue $e_\Omega = 4\%$: éléments isoparamétriques versus super-paramétriques.

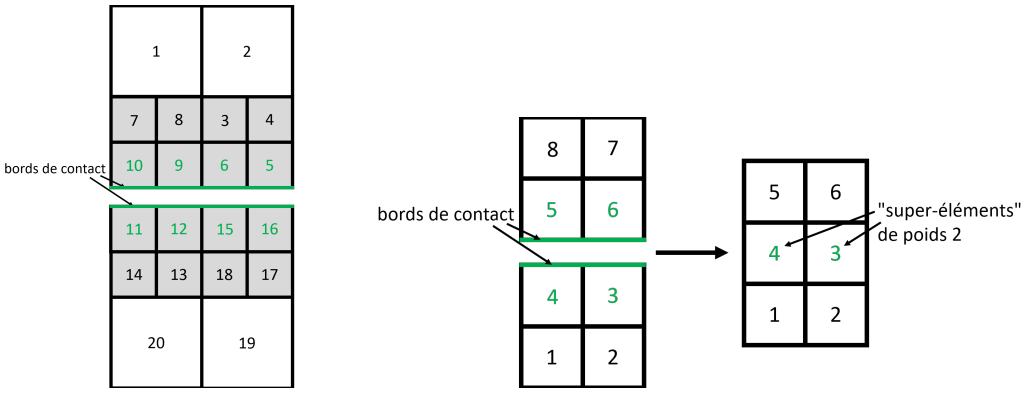
3 Approche parallèle

Partitionnement MFEM 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. Différents processeurs ne peuvent pas être responsables du même élément. L'ensemble des éléments du maillage est donc divisé en K régions disjointes, où K est le nombre de tâches MPI. Les données liées aux 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, ce qui facilite le calcul parallèle. L'opérateur de prolongation (défini pour les nœuds non-conformes dans le cas AMR) est étendu au traitement des DDL fantômes de sorte que la solution du système linéaire, obtenue après l'étape de résolution, concerne uniquement les DDL conformes et non fantômes, cf. [7] 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. La fonction `Rebalance` est appelée pour produire un nouveau partitionnement du maillage et rééquilibrer la charge. Utilisée ainsi (sans aucun argument), elle correspond à une fonction boîte noire qui construit une solution de partitionnement basée sur la méthode géométrique SFC classique via des courbes de Hilbert, voir [3] et paragraphe 'Partitionnement SFC' plus loin. Les fonctions de partitonnement de maillage intégrées dans MFEM ne sont actuellement pas adaptées aux problèmes de mécanique du contact.

Groupement des éléments en contact On souhaite combiner efficacement la stratégie de résolution du contact avec du raffinement h-adaptatif non conforme dans un paradigme de traitement parallèle à mémoire distribuée. Les algorithmes dédiés proposés ici garantissent un partitionnement du maillage tel que les nœuds de contact appairés se trouvent sur les mêmes processus. Concrètement, cela implique que la construction de la matrice locale d'appariement de contact B ne requiert pas de communications inter-processus. Ainsi, les matrices de rigidité de contact ($k_N B^T B$) et les vecteurs de contact ($k_N B^T D$)

sont assemblés localement sur chaque tâche MPI. Cela réduit considérablement la quantité de communications pendant les itérations de raffinement.

Partitionnement équidistribué Le partitionnement introduit dans [1] propose que les éléments s'appuyant sur des frontières de contact potentielles ne soient pas traités par les mêmes processus MPI que les autres éléments, voir figure 2a. Ainsi une première étape consiste à définir le nombre de tâches MPI traitant les éléments de contact. Il s'agit simplement d'avoir en moyenne autant d'élements sur les tâches MPI traitant les éléments de contact que sur les autres tâches (traitant des éléments "non-contact"). La deuxième étape opère une équidistribution des éléments sur les processus (sans chercher à regrouper spatialement les éléments). Les éléments de contact du premier solide sont distribués, équitablement, en fonction de leur numéro local sur les processus MPI affectés au contact. Uniquement la numérotation locale des éléments est prise en compte, aucune communication entre les processus MPI n'est nécessaire. 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) sont distribués de manière équilibrée sur le sous-ensemble restant des tâches MPI avec le même type de répartition équitable et locale. Enfin, une fois ce nouveau partitionnement produit, la fonction `Rebalance` de MFEM, s'occupe d'appliquer le partitionnement en transférant effectivement les éléments entre processus. Il est à noter que la numérotation globale initiale des éléments du maillage (donnée par le générateur de maillage Gmsh [9]) influence le partitionnement du maillage. Et cela, tant lors de la première itération AMR, que lors des itérations suivantes. En effet, la numérotation globale du maillage initiale a un impact sur la numérotation locale des éléments.



(a) Principe du traitement séparé des éléments de contact et de non-contact.

(b) Adaptation du partitionnement SFC au contact.

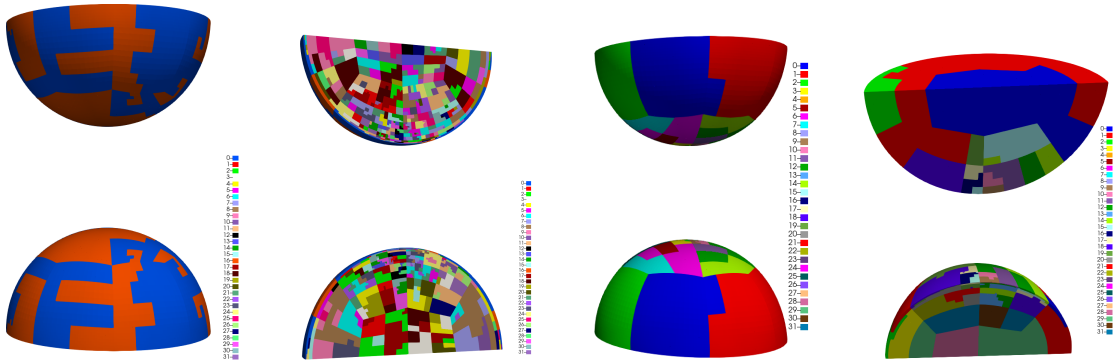
FIGURE 2 – Schémas illustrant le traitement et regroupement des éléments dans la zone de contact.

Partitionnement SFC Une courbe remplissant l'espace (ou SFC) est une fonction d'ordonnancement qui crée une bijection entre les nombres entiers et les points X situés dans une grille régulière à D dimensions ($D > 1$). Cet ordonnancement a la bonne propriété d'assurer que les points proches dans l'espace ont des indices entiers proches. La méthode SFC réalise une renumérotation globale des éléments et les redistribue ensuite. Cela induit des communications entre les tâches MPI.

L'algorithme de partitionnement SFC tenant compte du contact consiste à modifier le partitionnement SFC déjà implémenté dans MFEM pour le contraindre à garder les éléments de contact appairés sur les mêmes processus. Pour ce faire, les éléments de contact en vis-à-vis sont fusionnés en un seul *super-élément* de poids 2 (au lieu de 1 pour les autres éléments "non-contact"). La définition de ces *super-éléments* est représentée schématiquement sur la figure 2b. En pratique, nous avons enrichi le constructeur `ParMesh` par un appel à la fonction `GenerateSFCPartitioning`. Cette méthode fournit un partitionnement de type SFC variante Hilbert qui prend en compte les super-éléments de contact.

Caractéristiques des partitionnements Les deux partitionnements de maillage décrits ci-dessus garantissent que les termes de contact sont construits sans communication MPI. Cela se visualise dans les figures 3a et 3c sur lesquelles les zone en vis–à-vis sur les deux solides sont de la même couleur.

Il est aussi à noter que la méthode de partitionnement équidistribuée induit des régions MPI composées d'éléments dispersés dans tout le maillage, voir figure 3b versus figure 3d pour l'approche SFC-



(a) (b) (c) (d)

FIGURE 3 – Illustration d'un équilibrage de charge adapté à la prise en compte du contact via la stratégie équidistribuée (a,b) et SFC (c,d) pour un maillage 3D typique du problème de Hertz résolu avec AMR. La couleur représente le numéro de tâche MPI. Calculs effectués sur 32 processus.

contact. Les changements de couleur signifient en pratique un changement de tâche MPI et, par suite, des communications MPI induites dans certaines étapes de la résolution.

4 Benchmark parallèle et optimisations

Pour évaluer les performances de la stratégie proposée et l'impact des partitionnements, divers calculs ont été réalisés sur la machine topaze (CEA-CCRT, processeurs AMD Milan, 128 cœurs/nœud).

4.1 Benchmark Contact de Hertz sur demi-sphères

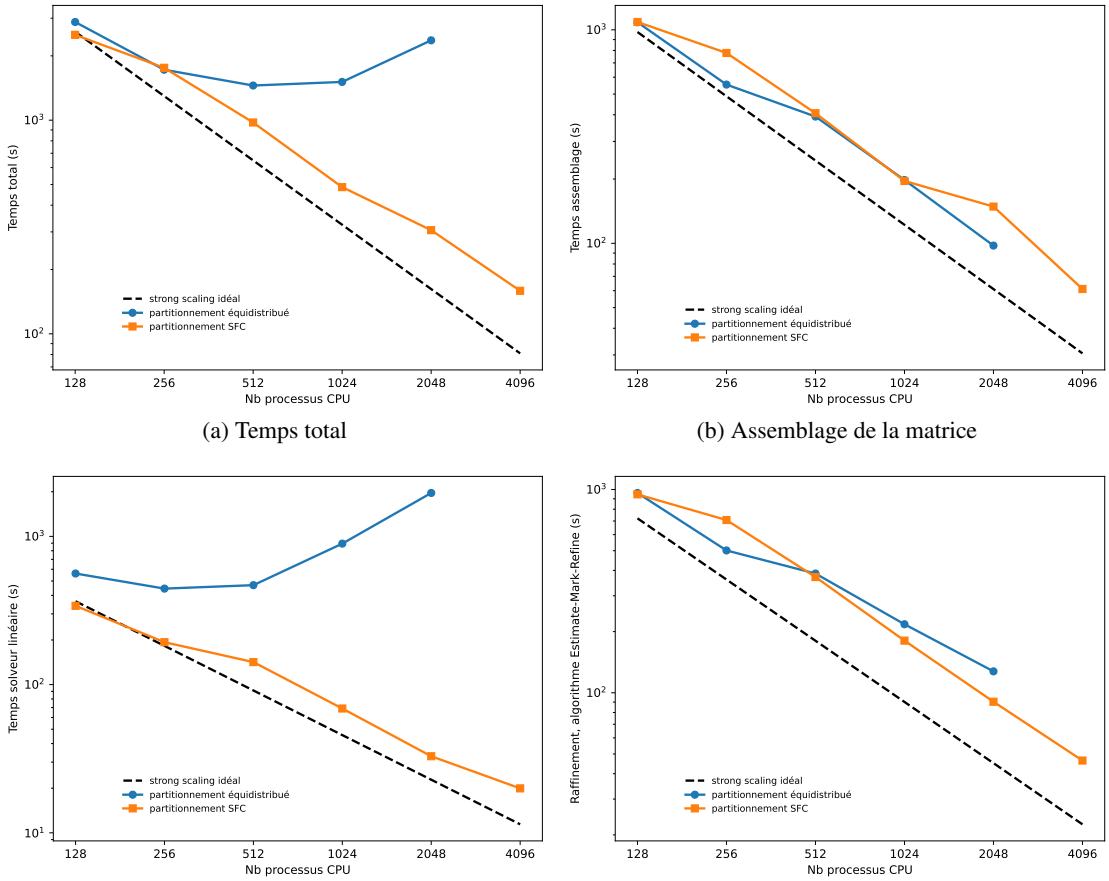
Strong scaling Pour le cas classique du contact hertzien 3D entre deux demi-sphères déformables, la figure 4 montre le temps total et le temps des étapes les plus significatives de notre algorithme AMR-contact en fonction du nombre de processus CPU. Pour le partitionnement équidistribué, le point pour 4096 processus n'est pas représenté car la simulation n'a pas convergé en un temps raisonnable.

Dans ces figures, les résultats obtenus sont comparés à la pente de la courbe de scaling idéale qui serait atteinte pour une parallélisation parfaite (sans aucun surcoût dû au parallélisme).

Les performances parallèles de la simulation utilisant le partitionnement *équidistribué* sont bonnes jusqu'à 512 processus mais tendent à se dégrader fortement au-delà. Cela provient du nombre important de sommets aux interfaces entre sous-domaines MPI qui a un impact sur les coûts en communication dans le solveur linéaire itératif. Cela engage de nombreuses multiplications impliquant des matrices creuses (*cf.* [10]) et échanges d'information aux interfaces entre sous-domaines. Pour caractériser cela, nous avons procédé à un profiling : on constate un fort poids dû aux communications avec 70 % du temps dans MPI_Allreduce et MPI_Waitall à 1024 processus au sein du solveur linéaire.

Concernant les résultats avec partitionnement *SFC*, un comportement hautement scalable est observé jusqu'à plus de 4000 processus CPU. Ceci s'explique par la bonne cohérence spatiale des sous-domaines. On constate ces bons résultats à toutes les étapes importantes de l'algorithme y compris l'étape de résolution du solveur linéaire itératif (voir figure 4c). Il est important de noter que certaines de ces sous-étapes ont peu d'impact sur le scaling : l'assemblage de la matrice et l'application de l'approche AMR, toutes deux scalables pour les deux partitionnements.

Effet de l'ordre de la transformation super-paramétrique et du préconditionneur L'empreinte mémoire constatée dans les simulations parallèles sur des maillages à plusieurs millions de nœuds est un facteur limitant. Nous avons été régulièrement amené à employer 4 cœurs par processus afin de mobiliser de la mémoire vive supplémentaire sur les supercalculateurs. Un axe d'amélioration est le choix optimisé de l'ordre de la transformation super-paramétrique. Aussi, le préconditionneur du solveur linéaire a une influence significative sur temps de calcul et mémoire. Parmi tous les préconditionneurs envisagés (et accessibles dans MFEM), HypreBoomerAMG et HypreILU sont ceux qui donnent les meilleurs résultats (en combinaison avec le solveur HyprePCG utilisé ici). La figure 5a présente une étude comparative sur l'ordre de la transformation (3 versus 6) et sur l'impact du préconditionneur en termes de temps de calcul



(a) Temps total

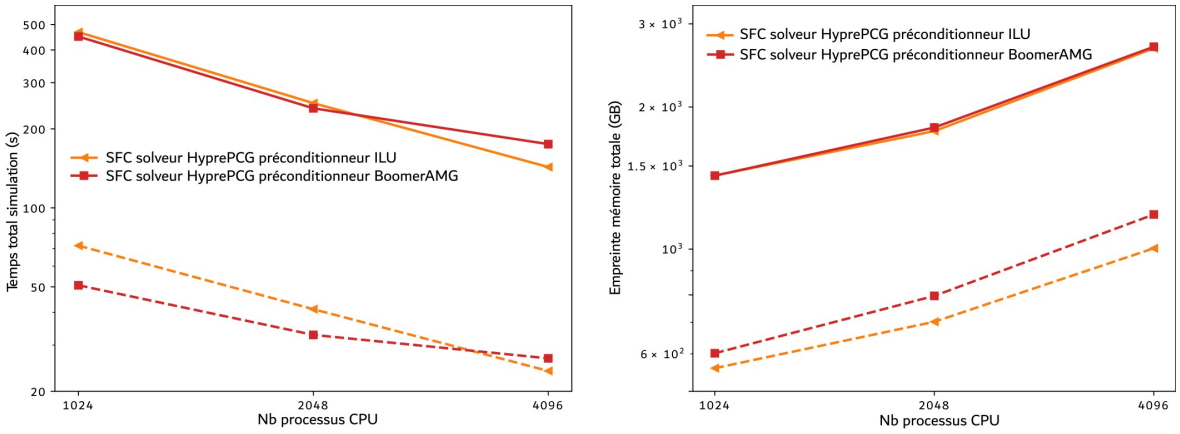
(b) Assemblage de la matrice

(c) Résolution par le solveur linéaire

(d) Approche AMR estimate-mark-refine

FIGURE 4 – Contact hertzien 3D : Performances parallèles de l'algorithme AMR-contact pour $e_\Omega = 1\%$, #DDL final= $18,3$ millions (5 raffinements), transformation super-paramétrique d'ordre 6.

et de mémoire consommée pour la stratégie avec partitionnement SFC-contact.



(a) Temps total

(b) Empreinte mémoire (GB)

FIGURE 5 – Contact Hertzien 3D : effet de l'ordre de la transformation super-paramétrique et du pré-condtionneur sur les temps de calculs et empreintes mémoire. Ordre 3 (en pointillés) et ordre 6 (en trait plein). Partitionnement SFC-contact. Méthode AMR avec $e_\Omega = 1\%$ et #DDL final = $18,3$ millions.

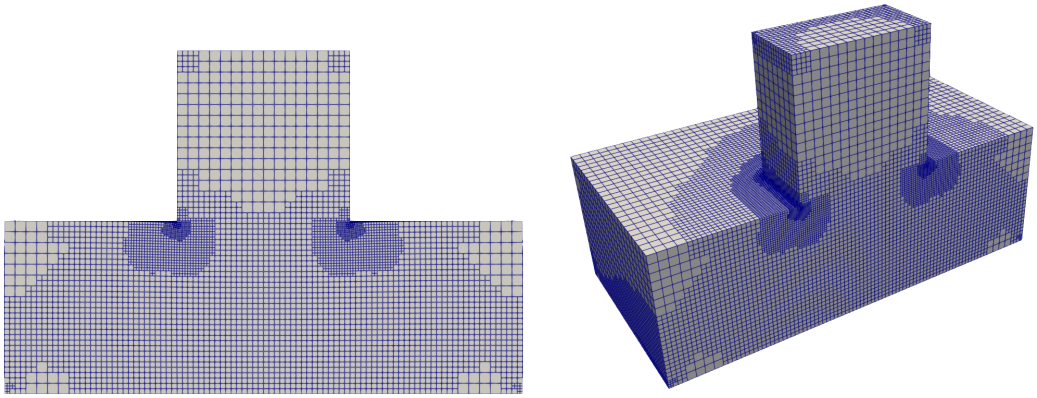
La figure 5a montre que l'utilisation d'un maillage à transformation super-paramétrique d'ordre 3 au lieu de 6 permet de faire diminuer le temps de calcul d'un facteur 5 dans le cas testé (pour 1024, 2048 et 4096 processus). En réalisant une analyse plus détaillée, il a été observé que cette diminution du temps de calcul est due essentiellement aux fonctions spécifiques de MFEM d'assemblage de la matrice de rigidité globale (`Assemble`), de formation du système linéaire sans le contact (`FormLinearSystem`) et d'application de l'approche ESTIMATE-MARK-REFINE choisie. En effet, le temps passé dans ces

trois fonctions est diminué d'un facteur 20 environ tandis que l'étape de résolution du système linéaire est globalement identique (ce qui est logique puisque la taille du système à résoudre est indépendante de l'ordre de la transformation). Le gain en empreinte mémoire est également important avec un facteur 2 observé lors de la réduction de l'ordre de la transformation, cf. figure 5b. Ainsi, pour réduire les coûts en espace mémoire et temps de calcul, une solution pragmatique consiste à réduire autant que possible (en fonction de la courbure à approcher) l'ordre de la transformation super-paramétrique des éléments.

Par ailleurs, la stratégie avec partitionnement *SFC* fournit des résultats légèrement meilleurs en temps de calcul, performances parallèles et mémoire consommée avec HypreILU comme pré-conditionneur.

4.2 Benchmark de type indentation

Pouvoir effectuer des simulations comportant un grand nombre d'inconnues représente un enjeu scientifique majeur. Ce paragraphe s'attache par conséquent à étudier les performances parallèles de notre stratégie numérique dans ce cadre. Il s'agit ici d'étudier un cas test représentatif de l'indentation d'un cube de côté H dans un pavé droit de dimensions $L \times L \times H$ ($L > H$). Le cube est plus rigide d'un facteur 100. Les maillages choisis sont concordants dans la zone de contact potentielle, de manière à ce que l'appariement nœud-à-nœud s'applique correctement. Par ailleurs, la géométrie cartésienne nous permet d'utiliser une transformation isoparamétrique d'ordre 1, ce qui permet de réduire considérablement l'empreinte mémoire de nos calculs, cf. étude en section précédente. Sur ce cas d'étude, des conditions aux limites de Dirichlet dans la direction verticale sont appliquées aux surfaces supérieure du cube et inférieure du pavé droit de manière à ce qu'ils rentrent en contact. Un exemple de résultat obtenu avec notre stratégie AMR-contact-HPC est visible en figure 6. Le système linéaire est résolu avec le solveur HyprePCG, pré-conditionné avec HypreILU.

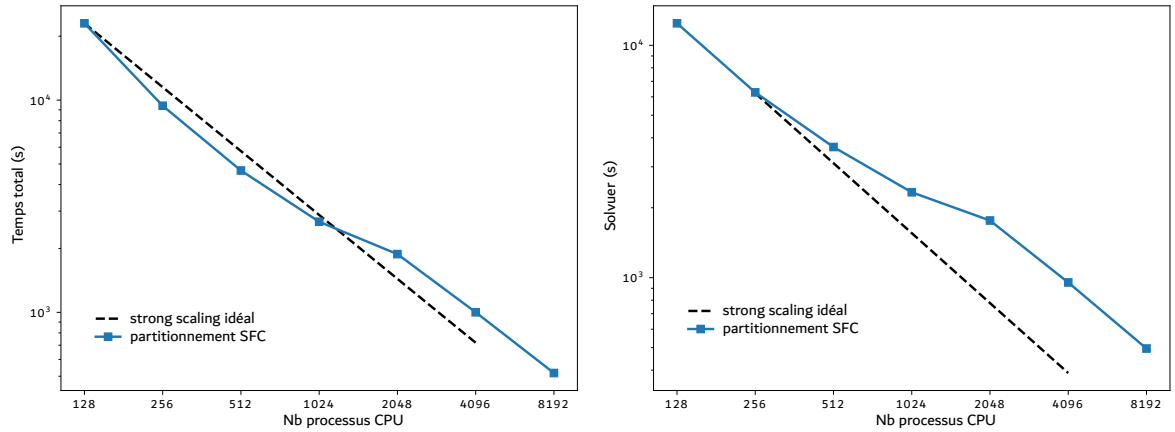


(a) (b)

FIGURE 6 – Contact d'indentation 3D : Stratégie AMR avec $e_\Omega = 4\%$, #DDL final = $1,73$ millions (6 raffinements), transformation d'ordre 1.

Un cas de plus grande taille, avec un nombre de degrés de liberté du maillage conforme final de $295,9$ millions, obtenu avec $e_\Omega = 0,9\%$ est maintenant considéré. L'étude de strong scaling est présentée en figure 7. Les calculs ont été réalisés en utilisant plusieurs cœurs par processus afin d'avoir la mémoire vive nécessaire sur le supercalculateur. Le nombre de cœurs par processus a dû être adapté et vaut en l'occurrence ici 4, 8, 16 et 32 pour les calculs sur 1024, 512, 256 et 128 processus respectivement ; il y a ensuite 2 cœurs par processus lorsque le nombre de processus est supérieur ou égal à 2048. L'empreinte mémoire maximum du calcul (obtenue pour le calcul utilisant 8192 processus) vaut 8034 GB.

Les résultats obtenus montrent une très bonne scalabilité globale de notre stratégie pour plus de 8000 processus lorsque nous l'appliquons à ce cas d'étude, cf figure 7a. Par ailleurs, il est observé que les performances parallèles se révèlent meilleures qu'idéalement pour 256, 512 et 1024 processus. Cela est dû à un comportement superlinéaire observé pour certaines étapes : formation du système linéaire, raffinement du maillage, redistribution de l'ensemble des éléments finis du maillage après raffinement. Cette super linéarité est vraisemblablement liée à une meilleure localité des données et à des effets cache lors des calculs. Par ailleurs, il peut être observé sur la figure 7b que le solveur perd en performance parallèle entre 1024 et 2048 processus. Une étude de profilage plus approfondie serait nécessaire afin d'affiner l'analyse.



(a) Temps total

(b) Temps solveur

FIGURE 7 – Contact d'indentation 3D : Performances parallèles de l'algorithme. Méthode AMR avec $e_\Omega = 0,9\%$, #DDL final = 295,9 millions (8 raffinements), transformation d'ordre 1..

5 Conclusion

Nous avons introduit un algorithme scalable, robuste et efficace pour traiter des problèmes de contact élastostatique en 3D entre solides déformables dans un cadre d'éléments finis. La solution proposée combine le traitement du problème de contact par pénalisation avec un algorithme d'appariement nœud-à-nœud et un raffinement h-adaptatif non conforme de maillages hexaédriques basé sur une approche estimate-mark-refine dans un cadre parallèle. La nouveauté présentée ici réside dans la construction d'un partitionnement de type Space Filling Curve adapté aux problèmes de contact (avec la définition de super-éléments) qui nous permet d'atteindre une scalabilité quasiment parfaite jusqu'à 8192 processus sur un problème d'environ 300 millions de degrés de liberté.

Références

[1] A. Epalle, I. Ramière, G. Latu, and F. Lebon. Parallel simulation and adaptive mesh refinement for 3d elastostatic contact mechanics problems between deformable bodies. In *Advances in Applied Mechanics*, volume 61. Elsevier, 2025.

[2] 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, and S. Zampini. MFEM : a modular finite element methods library. *Computers & Mathematics with Applications*, 2020.

[3] P. M. Campbell, K. D. Devine, J. E. Flaherty, L. G. Gervasio, and J. D. Teresco. Dynamic octree load balancing using space-filling curves. Technical report, CS-03-01, Williams College Department of Computer Science, 2003.

[4] A. Epalle. *Stratégies parallèles dédiées au raffinement local de maillages structurés en mécanique du contact*. PhD thesis, Aix-Marseille Université, 2024. Sciences pour l'ingénieur : spécialité Mécanique des Solides.

[5] P. Wriggers. *Computational contact mechanics. Second edition*. Springer-Verlag, 2006.

[6] D. Koliesnikova, I. Ramière, and F. Lebon. 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, 2021.

[7] J. Červený, V. Dobrev, and T. Kolev. Non-conforming mesh refinement for high-order finite elements. *SIAM J. Sci. Comput.*, 41(4) :C367–C392, 2019.

[8] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineerng analysis. *International Journal for Numerical Methods in Engineering*, 24 :337–357, 1987.

[9] C. Geuzaine and J.-F. Remacle. 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) :1309–1331, 2009.

[10] U.V. Catalyurek and C. Aykanat. Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Transactions on Parallel and Distributed Systems*, 10(7) :673–693, 1999.