

Calcul haute performance pour les problèmes de point-selle en mécanique du contact

V. Robert^{1,2}, P. Jolivet², O. Jamond¹, M. A. Badri³

¹ Université Paris-Saclay, CEA, Service d'Études Mécaniques et Thermiques, Gif-sur-Yvette, France, {vincent.robert;olivier.jamond}@cea.fr

² Sorbonne Université, CNRS, LIP6, Paris, France, pierre.jolivet@lip6.fr

³ Université Paris-Saclay, CEA, Service de Génie Logiciel pour la Simulation, Gif-sur-Yvette, France, mohd-afeef.badri@cea.fr

Résumé — Les problèmes de mécanique du contact conduisent à la résolution de systèmes linéaires de point-selle difficiles à résoudre. Dans le cadre d'une approche permettant de transformer le système de point-selle indéfini en système réduit défini positif, nous proposons l'introduction d'un facteur d'échelle qui améliore le conditionnement du système ainsi que plusieurs matrices de préconditionnement, dans le but de traiter des problèmes de grande dimension. Les premiers tests numériques présentés montrent les potentialités de la méthode et les futures directions de recherche.

Mots clés — HPC, contact, point-selle, résolution de système linéaire, préconditionnement.

1 Introduction

Dans le domaine de la mécanique des structures, les systèmes simulés impliquent souvent des structures déformables qui peuvent entrer en contact. Ceci se traduit généralement dans les modèles numériques par des contraintes cinématiques sur l'inconnue du problème (i.e. le champ de déplacement), traitées par l'introduction d'inconnues dites duales et d'inégalités les impliquant qui assurent le respect de la non-interpénétration des structures qui entrent en contact. Les problèmes de contact sont donc non linéaires, et nécessitent l'utilisation d'algorithmes itératifs de type Newton. À chaque itération de celui-ci, une première étape de détection des contacts est effectuée afin sélectionner les contraintes actives. Le système linéaire de point-selle peut ensuite être construit avec celles-ci, puis résolu.

Nous nous concentrons uniquement sur ce dernier point de résolution de ce système linéaire au sein d'une itération du solveur non linéaire : l'objectif est de développer des méthodes numériques efficaces de résolution de ce type de système, en très grande dimension (de l'ordre du milliard d'inconnues) et dans un contexte de calcul parallèle. Un des buts est de les intégrer à MANTA, un nouveau code de calcul multiphysique en cours de développement au CEA (voir [7]). MANTA se base sur la librairie PETSc (voir [2]) pour la résolution des systèmes linéaires, celle-ci est donc notre outil principal.

Ce résumé est structuré de la façon suivante : la section 2 introduit brièvement les systèmes de point-selle apparaissant en mécanique du contact, puis nous présentons la méthode centrale de cet article en section 3. Nos choix d'implémentation de cette méthode sont présentés à la section 4. Enfin, nous analysons en section 5 les résultats des premiers tests numériques réalisés.

2 Systèmes de point-selle

Dans le cas des problèmes de contact (voir [3] pour une introduction plus générale), un système de point-selle est un système linéaire symétrique de type bloc 2x2 de la forme

$$\begin{pmatrix} A & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \quad (1)$$

avec $A \in \mathbb{R}^{n \times n}$ symétrique définie positive (SPD) issue de la discrétisation de la forme bilinéaire de la formulation variationnelle du problème d'origine, $C \in \mathbb{R}^{m \times n}$ avec $m \leq n$ le bloc de contraintes actives, $x \in \mathbb{R}^n$ le vecteur de variables primales, $\lambda \in \mathbb{R}^m$ le vecteur des multiplicateurs de Lagrange, et $f \in \mathbb{R}^n$ et $g \in \mathbb{R}^m$ les vecteurs correspondants du membre de droite.

Une propriété importante des systèmes de point-selle apparaissant dans la majorité des problèmes en mécanique du contact, par rapport à ceux issus de la mécanique des fluides (le problème de Stokes par exemple), est que le bloc de contraintes y est beaucoup plus petit que le bloc principal, i.e. $m \ll n$. Les contraintes de contact ne sont définies que sur un bord du domaine simulé (une surface 2d pour un problème tridimensionnel), il en résulte en particulier une décroissance du ratio $\frac{m}{n}$ lorsqu'on raffine le maillage uniformément.

Les systèmes de point-selle sont difficiles à résoudre, notamment pour les raisons suivantes : ils sont indéfinis (ils ont à la fois des valeurs propres positives et négatives), et très souvent mal conditionnés. En conséquence, les méthodes de résolution pour systèmes définis positifs ne sont pas accessibles, et les méthodes itératives utilisables sont très lentes à converger, induisant ainsi la nécessité de préconditionner le système à résoudre. De façon générale, les préconditionneurs pour systèmes de point-selle efficaces sont conçus pour un type de problème particulier, ce qui motive nos recherches dans le cas du contact.

3 Méthode d'Ainsworth

3.1 Introduction

La méthode décrite par Mark Ainsworth dans [1] est une méthode de projection qui transforme le système de point-selle en un système de taille réduite $n \times n$, symétrique défini positif, qui a la même solution que le problème d'origine. Elle est potentiellement plus simple à résoudre, et d'autre part la taille du système obtenu étant indépendante du nombre de contraintes, le coût lié à celles-ci pourrait être plus progressif par rapport à leur nombre que dans une méthode de résolution du système de point-selle complet.

3.2 Description

On se base ici sur l'article [1]. On souhaite résoudre le système (1), avec A SPD, et l'on suppose que les hypothèses suivantes de problème bien posé sont remplies :

- i. $\text{Ker}(A) \cap \text{Ker}(C) = \{0\}$,
- ii. C est de rang plein.

Cette dernière condition permet de définir les projecteurs orthogonaux complémentaires

$$\begin{aligned} P &:= C^T(CC^T)^{-1}C, & (\text{Im}(P) = \text{Im}(C^T), \text{Ker}(P) = \text{Ker}(C)) \\ Q &:= I - P = I - C^T(CC^T)^{-1}C, & (\text{Im}(Q) = \text{Ker}(C), \text{Ker}(Q) = \text{Im}(C^T)) \end{aligned}$$

ainsi que $R := C^T(CC^T)^{-1}$. On peut ensuite construire les membres de gauche et de droite

$$\begin{aligned} A' &:= C^T C + Q^T A Q \in \mathbb{R}^{n \times n}, \\ f' &:= C^T g + Q^T (f - A R g) \in \mathbb{R}^n, \end{aligned}$$

du système réduit de dimension $n \times n$

$$A'x = f'. \quad (2)$$

Le théorème principal de [1] démontre alors :

- a) l'existence et l'unicité de la solution x de (2),
- b) que la solution x de (2) est solution du problème de minimisation à l'origine de (1),
- c) qu'une fois x obtenu, λ peut être calculé simplement : $\lambda = R^T (f - Ax)$.

Un autre résultat important de cet article est que les bornes du spectre de A' sont similaires à celles du spectre de A : soient $\bar{\lambda}(A)$ et $\bar{\lambda}(CC^T)$, resp. $\underline{\lambda}(A)$ et $\underline{\lambda}(CC^T)$, les valeurs propres maximales, resp. minimales, de A et CC^T , alors pour tout x de l'espace d'éléments finis considéré on a

$$\min\{\underline{\lambda}(A), \bar{\lambda}(CC^T)^{-1}\} \leq \frac{x^T A' x}{x^T x} \leq \max\{\bar{\lambda}(A), \underline{\lambda}(CC^T)^{-1}\}.$$

Les valeurs propres de CC^T étant indépendantes du paramètre de discrétisation du maillage, leur effet sur ces bornes est insignifiant. Le conditionnement de A' est donc similaire à celui de A .

3.3 Modification et système linéaire final

On considère la modification suivante des éléments constitutifs de (2) : pour un réel $\beta > 0$, nous pouvons définir les termes

$$\begin{aligned}\tilde{A} &:= \beta C^T C + Q^T A Q \in \mathbb{R}^{n \times n}, \\ \tilde{f} &:= \beta C^T g + Q^T (f - A R g) \in \mathbb{R}^n,\end{aligned}\tag{3}$$

puis le système

$$\tilde{A}x = \tilde{f},\tag{4}$$

et l'on a les résultats suivants :

- a) l'existence et l'unicité de la solution x de (4),
- b) la solution x de (4) est solution de (2),

Dans le but de résoudre (4) aussi efficacement que possible pour des problèmes de grande taille et en contexte de calcul parallèle, le recours au préconditionnement est indispensable. Le système linéaire final que l'on étudie est donc, pour $M \in \mathbb{R}^{n \times n}$ une matrice de préconditionnement non singulière à définir,

$$\boxed{M^{-1} \tilde{A} x = M^{-1} \tilde{f}}.\tag{5}$$

Rappelons que pour être efficace comme préconditionneur, M doit être d'une certaine façon proche de \tilde{A} (i.e. $M \approx \beta C^T C + Q^T A Q$), de sorte que le conditionnement $\kappa(M^{-1} \tilde{A})$ du système préconditionné soit meilleur (i.e. inférieur) que celui du système non préconditionné $\kappa(\tilde{A})$, et que les coûts liés à la construction de M et à la résolution d'un système linéaire avec M comme matrice à chaque itération du solveur linéaire principal doivent être compensés par le gain global en performance. Il s'agit donc de trouver la meilleure combinaison possible d'une matrice de préconditionnement M et d'une méthode numérique de résolution d'un système linéaire avec M comme matrice.

Nous observons lors de nos expériences que le choix du facteur β dans (3) a une forte incidence sur $\kappa(M^{-1} \tilde{A})$, la question se pose donc de la valeur optimale de β pour ce conditionnement. Dans [6], il est proposé une méthode de pénalisation qui modifie la matrice de point-selle avec l'introduction d'un terme similaire au $C^T C$ dans \tilde{A} et la proposition d'un facteur d'échelle empirique $\frac{\|A\|}{\|C\|^2}$ devant ce terme, améliorant le conditionnement du système. Nous nous basons sur cette proposition pour expérimenter plusieurs choix de β . À la suite de ces tests, nous fixons le facteur d'échelle devant les $C^T C$ dans \tilde{A} et M et devant $C^T g$ dans \tilde{f} à une valeur de $\beta = 5 \frac{\|Q^T A Q_*\|_1}{\|C^T C\|_1}$, où Q_* est une approximation de Q dans M définie à la section 4.3. Nous gardons cependant à l'esprit la nature empirique de cette valeur et souhaiterons notamment vérifier sa pertinence pour d'autres problèmes.

4 Méthode de résolution du système

4.1 Algorithme itératif principal et algorithme de préconditionnement

Dans tous nos tests, la méthode numérique choisie pour résoudre (5) est l'algorithme itératif du Gradient Conjugué (CG). Cette méthode de Krylov est l'algorithme de choix pour les systèmes SPD, ce qui est le cas pour (5) pour les valeurs retenues pour le facteur β et les M proposés (voir section 4.3). Le critère de convergence est une tolérance relative de 1e-8 sur la norme naturelle relative du résidu.

Le choix de l'algorithme de préconditionnement s'est porté sur la méthode multigrille algébrique (AMG), avec lequel nous avons obtenu les meilleures performances, et dont l'efficacité avec CG est reconnue, de même que sa grande capacité d'extensibilité (voir [5]). Une limite possible à ce choix, que dans la pratique nous n'avons pas encore rencontrée, est le fait qu'AMG a été conçue pour la résolution de problèmes issus d'équations aux dérivées partielles elliptiques scalaires (voir [8]), ce que le système modifié (5) n'est pas du fait des modifications inhérentes à la méthode. L'implémentation choisie est BoomerAMG de la librairie Hypre, qui offre notamment les algorithmes HMIS et PMIS de construction des opérateurs grossiers conçus pour être performants en contexte de calcul parallèle (voir [4]).

La résolution du système se divise en deux étapes : une phase préparatoire (appelée *setup* dans ce qui suit) pour la méthode d'Ainsworth et pour AMG pendant laquelle les divers opérateurs en jeu sont construits, puis la phase de résolution proprement dite, constituée des itérations successives de CG préconditionné par AMG jusqu'à convergence.

4.2 Implémentation de \tilde{A}

Pour une raison de consommation de mémoire, la matrice \tilde{A} n'est pas construite explicitement. En effet, même si CC^T est creuse et de petite dimension dans le cadre d'un problème de contact, son inverse est dense, et l'action de C^T sur $(CC^T)^{-1}C$ puis de Q sur A conduit à ce que \tilde{A} soit beaucoup plus dense que A (voir figure 1). L'utilisation de CG pour la résolution du système permet de se restreindre à définir une fonction qui réalise le produit matrice-vecteur de \tilde{A} (de même pour Q). Pour le terme $(CC^T)^{-1}$ dans Q , une factorisation de Cholesky de CC^T est effectuée une seule fois lors du setup, puis durant la résolution il suffit de procéder, à chaque itération, à une étape de descente et remontée peu coûteuse.

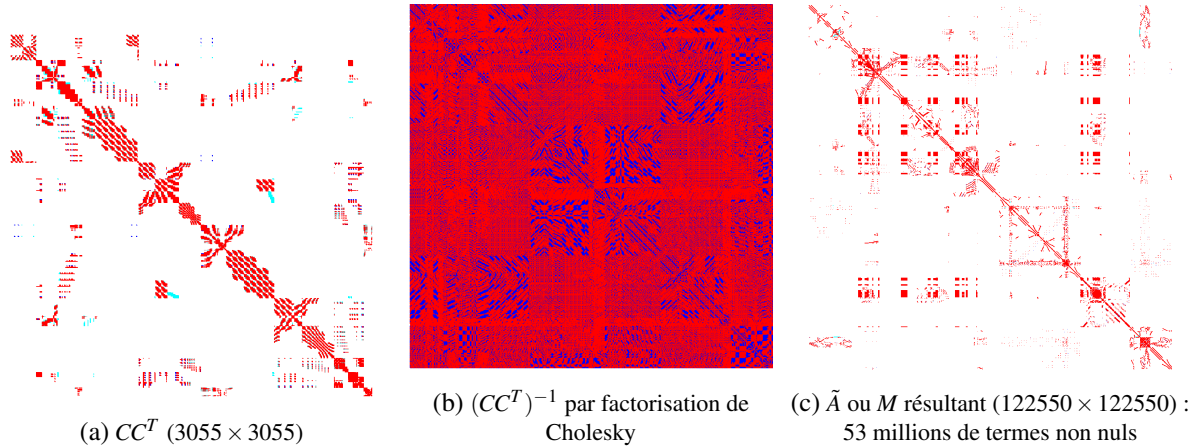


FIGURE 1 – Problème d'un inverse dense de CC^T dans \tilde{A} ou M : densité des matrices (A a 8 millions de termes non nuls).

4.3 Choix de la matrice de préconditionnement M

Afin de pouvoir utiliser les algorithmes de préconditionnement de PETSc, il est par contre nécessaire de construire M explicitement. Pour l'exacte même raison que pour \tilde{A} , il n'est pas souhaitable d'utiliser de méthode qui calcule une approximation pleine de l'inverse de CC^T , du fait de la consommation de mémoire qui en résulte. La recherche d'une approximation de $(CC^T)^{-1}$ aussi creuse que possible mais suffisamment précise pour que l'algorithme de préconditionnement reste efficace est donc notre premier moyen d'action sur la construction d'une matrice M . Nous expérimentons de nombreuses méthodes pour approcher cet inverse, qui conduisent à développer et sélectionner les différentes approximations Q_* de Q (donc différentes matrices de préconditionnement M_*) que nous présentons à présent.

Pour $C \in \mathcal{M}_{m \times n}(\mathbb{R})$, afin d'approcher le projecteur orthogonal $Q = I - C^T(CC^T)^{-1}C$, on définit plusieurs approximations

$$Q_* := I - C^T \text{Inv}(CC^T)C,$$

où dans chaque Q_* l'action de $\text{Inv}(\cdot) : \mathcal{M}_m(\mathbb{R}) \rightarrow \mathcal{M}_m(\mathbb{R})$ sur CC^T est différente et décrite dans le tableau ci-dessous, de même que les noms des différents Q_* et des M_* qui en résultent :

TABLE 1 – Définitions des Q_* , des $\text{Inv}(CC^T)$, et des M_*

Nom de Q_*	Description de la méthode utilisée dans $\text{Inv}(CC^T)$	Nom de M_*
Q_{CC}	Résolution directe via une factorisation de Cholesky de CC^T	M_{CC}
Q_{CG}	Gradient conjugué avec une tolérance relative élevée (défaut $1e-1$), préconditionné par une inversion diagonale de Jacobi	M_{CG}
Q_{BJ_CC}	Méthode de block Jacobi, avec une factorisation de Cholesky pour chacun des blocs	M_{BJ_CC}
Q_{BJ_CG}	Méthode de block Jacobi, avec pour chaque bloc une méthode de gradient conjugué avec tolérance relative élevée (défaut $1e-1$) préconditionné par une inversion diagonale de Jacobi	M_{BJ_CG}
Q_{JCB}	Inversion diagonale de Jacobi, où chacun des termes diagonaux utilisés est la norme $L1$ de la ligne correspondante de CC^T	M_{JCB}
Q_{ID}	Cas particulier dans lequel Q_{ID} est simplement la matrice identité	M_{ID}

Elles sont en quelque sorte classées de la plus précise mais coûteuse, à la moins précise mais plus légère. Q_{CG} et Q_{BJ_CG} exploitent une propriété de l'implémentation PETSc de CG : les termes non convergés sont laissés à zéro. Cela permet d'obtenir une approximation d'autant plus creuse (mais moins précise) que le critère de convergence de tolérance relative sur la norme du résidu est élevé.

Q_{BJ_CC} et Q_{BJ_CG} sont simplement des versions bloc-diagonales des deux premières. Pour Q_{JCB} , des tests ont montré que l'utilisation de la norme $L1$ des lignes CC^T au lieu de ses termes diagonaux aboutit à un bien meilleur conditionnement du système. Q_{TD} est née de l'idée d'avoir une version aussi légère que possible. Nous n'avons pas encore essayé d'autre préconditionneur que Jacobi pour Q_{CG} et Q_{BJ_CG} , l'idée étant d'avoir une méthode peu coûteuse et qui préserve la symétrie de l'opérateur.

5 Expériences numériques

5.1 Cas-test

Il s'agit de la modélisation par éléments finis d'une situation de contact entre deux cubes déformables Ω_1 et Ω_2 posés l'un sur l'autre, qui se veut à la fois simple et représentative de cas de contact réels que l'on veut traiter dans des applications industrielles, avec notamment des matériaux différents pour chaque cube et un maillage non conforme au niveau de la surface du contact. La formulation est celle de l'élasticité linéaire isotrope compressible avec deux conditions de Dirichlet (un déplacement vertical vers le bas imposé à la face supérieure du cube supérieur, et l'immobilité imposée à la face inférieure du cube inférieur), voir la figure 2a pour une description des domaines :

$$\begin{aligned} -\nabla \cdot \sigma_1(u) &= f_1 \text{ dans } \Omega_1, \\ -\nabla \cdot \sigma_2(u) &= f_2 \text{ dans } \Omega_2, \\ u &= (0 \quad 0 \quad -\gamma)^T, \text{ avec } \gamma > 0, \text{ sur } \Gamma_1^D, \\ u &= 0 \text{ sur } \Gamma_2^D, \end{aligned}$$

à laquelle s'ajoutent des contraintes de contact, que nous décrivons au niveau discret. Le maillage est hexahédral et les éléments de type Q1.

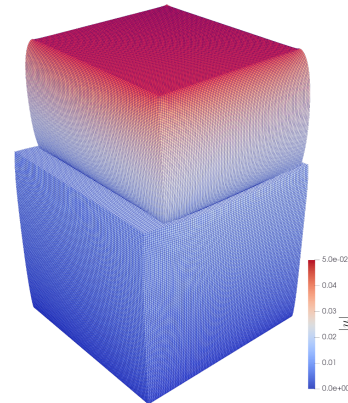
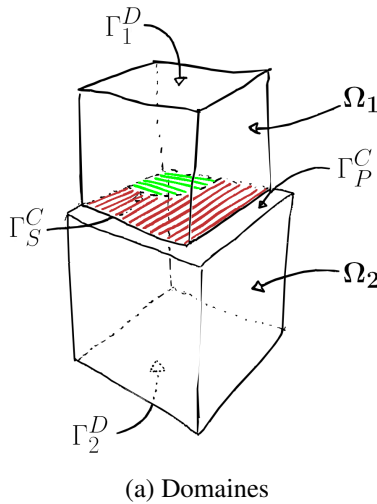


FIGURE 2 – Présentation du cas-test.

Les deux surfaces en contact sont nommées surface *primaire* Γ_P^C et surface *secondaire* Γ_S^C . La définition des contraintes suit la méthode *nœud-vers-surface* : les contraintes sur les déplacements u^s des nœuds de la surface secondaire Γ_S^C sont exprimées à l'aide de combinaisons convexes des déplacements u^p des nœuds des éléments de la surface primaire Γ_P^C sur lesquels ils sont projetés orthogonalement. La zone de contact est divisée en deux zones de proportion paramétrable, qui obéissent chacune à un type de contrainte :

- une condition d'*adhérence* : $u^s - \sum_{i=1}^4 \alpha_i u_i^p = 0$ (avec $\alpha_1, \dots, \alpha_4 \geq 0$ et $\sum_{i=1}^4 \alpha_i = 1$),
- une condition de *glissement* : $(u^s - \sum_{i=1}^4 \beta_i u_i^p) \cdot n = 0$ (avec $\beta_1, \dots, \beta_4 \geq 0$ et $\sum_{i=1}^4 \beta_i = 1$).

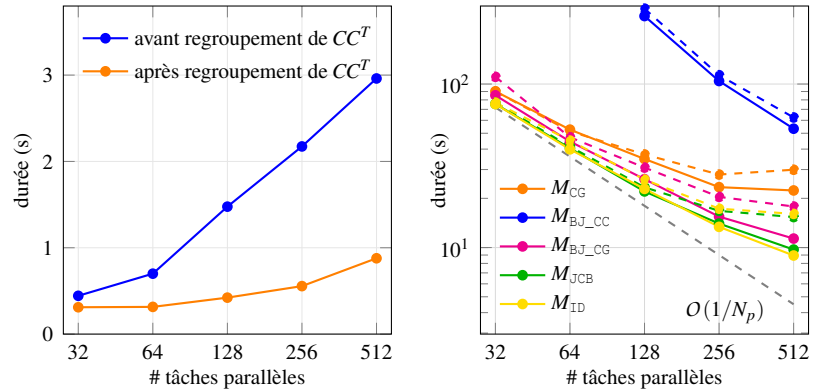
5.2 Extensibilité forte

Nous effectuons un premier test d’extensibilité forte, pour un problème de taille modérée : 5.5 millions d’inconnues, résultant de la discrétisation de chaque cube en 96^3 éléments (voir figure 2b), distribuées sur 32 à 512 tâches parallèles. Ce test permet d’aborder la question de la distribution parallèle de C (donc de CC^T) et des conséquences de celle-ci sur la performance de la méthode.

La raison est la suivante : l’algorithme de partitionnement du maillage optimise la distribution parallèle du maillage volumique, donc la charge de calcul par processeur liée à A , mais du fait que les contraintes de contact ne sont que surfaciques, C n’est distribuée que sur une petite partie des processeurs, avec de plus une charge de calcul par processeur liée à C possiblement très inégale (voir table 2). A chaque fois que l’on double le nombre total de processeurs au cours du test, bien que le nombre de processeurs concernés par C ne soit multiplié que par un facteur de l’ordre de $2^{\frac{2}{3}}$ (du fait du rapport de dimensions surface / volume), chacun d’eux reçoit de moins en moins de lignes de C , la quantité de communications par opération de calcul impliquant C augmente donc considérablement. Ceci explique notamment la mauvaise extensibilité de la factorisation de Cholesky de CC^T dans \tilde{A} pendant la phase de setup (voir figure 3a, courbe bleue). Dans ce test, C et CC^T n’ont que 27 000 lignes. La solution est donc de regrouper le CC^T de \tilde{A} sur un seul processeur, ce qui améliore considérablement l’extensibilité de cette factorisation (voir figure 3a, courbe orange) donc l’extensibilité globale (voir figure 3b). Dans le cas d’une matrice C de plus grande taille, on adapte cette stratégie en regroupant CC^T sur plusieurs processeurs au lieu d’un seul, en maximisant le travail de chaque processeur concerné jusqu’à une limite de 100 000 lignes (des tests ont montré qu’il s’agit d’un ordre de grandeur raisonnable).

# procs total	# lignes A/proc	# procs CC^T	# lignes CC^T /proc
32	170 000	8 (25%)	3340 (2452 à 4153)
64	85 000	13 (20%)	2055 (862 à 3747)
128	42 500	18 (14%)	1484 (64 à 2466)
256	21 250	29 (11%)	921 (239 à 1434)
512	10 625	46 (9%)	581 (176 à 1198)

TABLE 2 – Distribution parallèle (A , CC^T avant regroupement) : nombre total de processeurs, nombre moyen de lignes de A par processeur, nombre de processeurs ayant des lignes de CC^T , nombre moyen et minimal à maximal de lignes de CC^T par processeur concerné.



(a) Factorisation de Cholesky dans \tilde{A}

(b) Totaux (pointillés : avant / pleines : après)

FIGURE 3 – Test d’extensibilité forte : comparaison avant / après regroupement du CC^T de \tilde{A} sur un processus.

Par ailleurs, les résultats d’extensibilité des temps de construction des différents M_* ne sont pas bons. Cela nous semble être également causé par la distribution de C et les opérations qui en résultent, et nous incite donc à nous concentrer sur cette question dans nos recherches à venir. Les résultats positifs de ce test sont la bonne extensibilité de la phase de résolution, ainsi que la constance du nombre d’itérations (entre 30 et 33 selon le nombre de tâches).

5.3 Extensibilité faible

Nous imposons une moyenne constante de 83 000 inconnues environ par processus pour 8, 64 puis 512 processus en raffinant chaque cube en 48^3 , 96^3 et 192^3 éléments hexaédraux, et obtenons des systèmes avec respectivement environ 0.66, 5.3 et 42.5 millions d’inconnues. Les paramètres physiques du problème sont adaptés en conséquence afin que celui-ci reste similaire pour chaque processus.

Les temps obtenus lors de la phase de résolution avec M_{BJ_CG} , M_{JCB} et M_{ID} (voir figure 4a) semblent correspondre au comportement attendu de BoomerAMG et CG (voir [5]), mais nous devons effectuer des tests avec des systèmes de plus grande taille pour le confirmer. Les mauvaises performances obtenues en utilisant M_{BJ_CC} et M_{CG} nous incitent à nous délester de ces choix. Le nombre d’itérations ne croît que très modérément, entre 18 pour le plus petit cas et 23 pour le plus grand. Comme précédemment, l’extensibilité des temps nécessaires pour construire les M_* dans la méthode d’Ainsworth est mauvaise

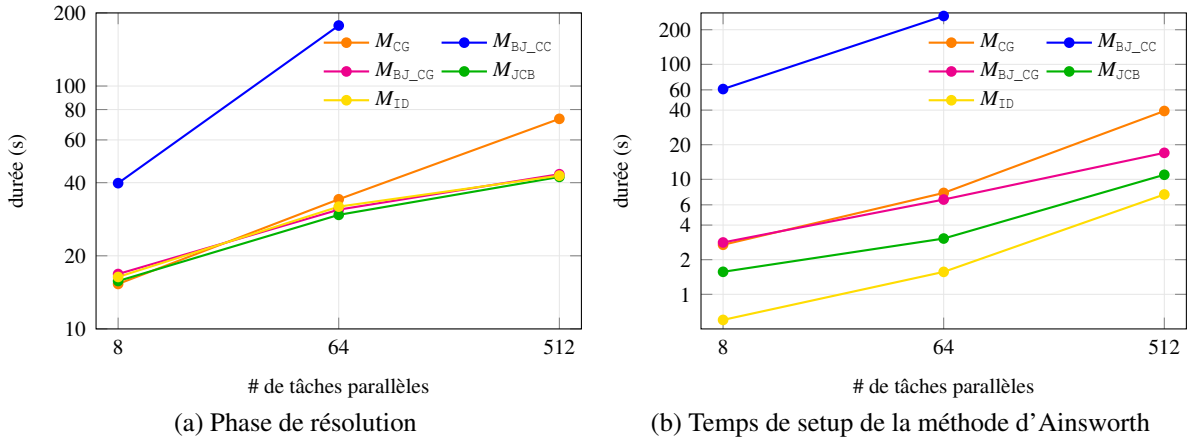


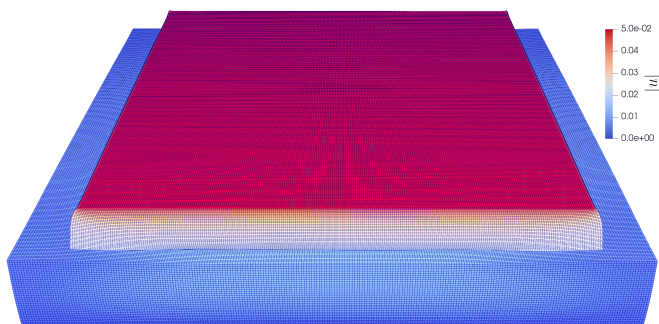
FIGURE 4 – Test d'extensibilité faible.

(voir figure 4b) : outre la distribution parallèle de C à améliorer et les coûts inhérents aux produits matriciels, il est d'autant plus important de fournir une estimation précise de la préallocation de mémoire des diverse matrices en jeu que le cas est de grande dimension.

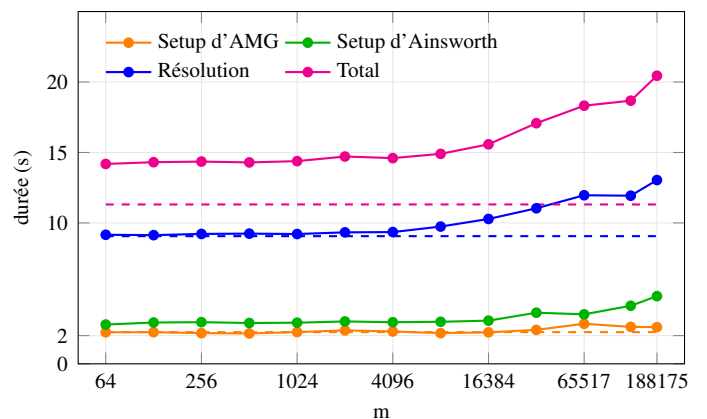
5.4 Coût de la méthode en fonction du nombre de contraintes

Le but de ce troisième test est d'évaluer le coût lié aux contraintes, en comparant une résolution du problème sans contact, c'est-à-dire de $Ax = f$ avec A et f issus de (1), avec des résolutions successives du même cas dans lequel on ajoute graduellement des contraintes, c'est-à-dire des lignes m de C , de façon aléatoire (le point de vue est ici uniquement celui de la résolution du système, et n'a pas de signification physique). Chacun des deux volumes en contact est ici divisé en $256 \times 256 \times 32$ éléments hexaédraux cubiques, les divisions en 256 parties étant effectuées dans le plan de la surface de contact afin de maximiser le nombre total d'éléments en contact (voir la figure 5a) : C a environ 188 000 lignes, pour un total d'environ 13 millions d'inconnues. L'approximation de Q utilisée ici est M_{JCB} .

Les résultats sont présentés en figure 5b. AMG est globalement insensible à la présence de contraintes, ainsi qu'à leur nombre. Les durées des autres étapes sont très stables pendant la première moitié du test avec moins de contraintes, et n'augmentent que lentement lorsque leur nombre devient non négligeable par rapport au nombre total d'inconnues. Dans cette première moitié, les temps de résolution sont similaires à ceux observés dans le cas sans contraintes et les durées de setup de la méthode d'Ainsworth sont très constantes. Une analyse plus fine montre que le temps de construction de M_{JCB} y est le coût dominant, et reste globalement constant quel que soit le nombre de contraintes. L'augmentation du temps pour les plus grands nombres de contraintes est essentiellement due au coût croissant de la factorisation de Cholesky de CC^T dans \tilde{A} . Par ailleurs, le nombre d'itérations de CG est très stable sur l'ensemble du test, de 22 à 24 itérations.



(a) Illustration du cas-test



(b) Résultats avec M_{JCB} (pointillés : cas sans contraintes)

FIGURE 5 – Test de la méthode d'Ainsworth en fonction du nombre de contraintes.

6 Conclusion

Dans l'état actuel de nos recherches, la méthode d'Ainsworth semble prometteuse :

- en utilisant une matrice de préconditionnement M_* avec une approximation judicieuse de Q (Q_{JCB} semble le meilleur candidat, étant à la fois peu coûteux en mémoire comme en calculs, rapide, et aboutissant à un conditionnement du système aussi bon que des versions plus coûteuses),
- avec le facteur d'échelle β proposé qui améliore grandement le conditionnement du système,
- avec CG comme solveur principal et AMG comme algorithme de préconditionnement,
- sous réserve d'améliorer les performances d'extensibilité du setup de la méthode.

Plusieurs questions restent en suspens, auxquelles nous devons répondre afin de pouvoir effectuer des tests à grande échelle et, espérons-le, conclure à l'utilité réelle de la méthode, au premier rang desquelles la question de la distribution parallèle de C . Des tests de robustesse de la méthode doivent également être conduits, afin d'estimer la classe de problèmes pour laquelle la méthode peut s'avérer efficace.

Références

- [1] Ainsworth Mark. *Essential boundary conditions and multi-point constraints in finite element analysis*, Comput. Methods Appl. Mech. Eng., Elsevier BV, 6323-6339, 2001.
- [2] Balay Satish, Abhyankar Shrirang et al. <https://petsc.org/>, 2025.
- [3] Benzi Michele, Golub Gene, Liesen, Jörg. *Numerical Solution of Saddle Point Problems*, Acta Numerica, Cambridge University Press, 1-137, 2005.
- [4] De Sterck Hans, Yang Ulrike Meier, Heys Jeffrey J. *Reducing complexity in parallel algebraic multigrid preconditioners*, SIAM J. Matrix Anal. Appl., Society for Industrial & Applied Mathematics (SIAM), 1019-1039, 2006.
- [5] Falgout Robert D. *An Introduction to Algebraic Multigrid*, Comput. Sci. Eng., IEEE Computer Society, 24-33, 2006.
- [6] Golub Gene, Greif Chen. *On solving block-structured indefinite linear systems*, SIAM J. Sci. Comput., Society for Industrial & Applied Mathematics (SIAM), 2076-2092, 2003.
- [7] Jamond Olivier, Lelong Nicolas et al. *MANTA : an industrial-strength open-source high performance explicit and implicit multi-physics solver*, 16ème Colloque National en Calcul de Structures (CSMA 2024), CNRS et CSMA et ENS Paris-Saclay et CentraleSupélec, 2024.
- [8] Trottenberg Ulrich, Schuller Anton. *Multigrid*, Academic Press, Inc., 2000.