

Adaptation à deux niveaux pour le solveur Adaptive Multipreconditioned FETI

C. Bovet¹, A. Parret-Fréaud², P. Gosselet³, N. Spillane⁴

¹ Onera – The French Aerospace Lab F-92322 Chatillon, France, christophe.bovet@onera.fr

² Safran Tech, Modelling & Simulation, Rue des Jeunes Bois, Châteaufort, 78114 Magny-Les-Hameaux, France, augustin.parret-freaud@safrangroup.com

³ LMT, ENS Cachan, CNRS, Université Paris-Saclay, 61 av. du Président Wilson, 94235 Cachan, France, pierre.gosselet@lmt.ens-cachan.fr

⁴ CMAP UMR 7641 Ecole Polytechnique CNRS, Route de Saclay, 91128 Palaiseau Cedex, France, nicole.spillane@polytechnique.edu

Résumé — Nous présentons l'une des dernières évolutions de la méthode de décomposition de domaine multipréconditionnée et adaptative AMPFETI, qui vise à résoudre des problèmes de très grandes tailles et de complexités industrielles. Dans AMPFETI, chaque agrégat de sous-domaines propose une direction de recherche. Ce préconditionnement multiple fournit la robustesse nécessaire pour résoudre les systèmes mal conditionnés induits par exemple par une forte hétérogénéité du matériau. Le critère d'adaptation permet de sélectionner uniquement les directions de recherche qui contribuent réellement à la convergence du solveur itératif. L'évolution présentée est l'ajout d'un second niveau d'adaptation dans le sens où les agrégats de multipréconditionnement évoluent pour accélérer encore la convergence. Les performances du solveur AMPFETI sur des cas industriels seront également présentées.

Mots clés — Décomposition de domaine, AMPFETI, préconditionneurs multiples, préconditionneurs adaptatifs, solveur de Krylov, *deflation*

1 Introduction

L'avènement du calcul haute performance offre des moyens de calculs toujours plus puissants et le calcul de structures tente naturellement d'exploiter ces nouvelles possibilités. La finesse des discrétisations s'accroît ce qui conduit à des systèmes linéaires de très grandes tailles; permettre la résolution massivement parallèle de ces derniers est indispensable. Les solveurs itératifs de type Krylov qu'ils soient appliqués directement sur les systèmes obtenus, ou suite à une formulation en décomposition de domaine [1, 2] sont parfaitement adaptés pour relever ce défi. Comme tout solveur itératif, ces méthodes restent sensibles au conditionnement du système à résoudre et des situations pathologiques comme un matériau très hétérogène, une déformation plastique importante ou des éléments finis de mauvaises qualités ralentissent particulièrement, ou même empêchent, leur convergence. Longtemps cantonnée aux exemples académiques, l'application de ces solveurs à des modèles provenant de problématiques industrielles est récente et ces derniers comportent bien souvent les situations pathologiques énoncées précédemment.

Accroître la robustesse des approches itératives est donc une problématique actuelle et deux ingrédients principaux se détachent. Le premier consiste à utiliser des méthodes blocs qui génèrent des espaces de recherche de plus grande dimension. L'approche bloc peut être obtenue en décomposant le second membre [3] ou en utilisant des préconditionneurs multiples, elle correspond aux solveurs de Krylov multipréconditionnés [4–7]. La méthode Adaptive Multipreconditioned FETI présentée appartient à cette première catégorie [8, 9]. Le second se traduit par l'ajout d'un problème grossier supplémentaire pour lequel la solution sera obtenue grâce à un solveur direct. Cette approche correspond aux solveurs de Krylov augmentés [10]. Connaissant un espace de faible dimension, dit d'*augmentation*, qui contient les modes problématiques de l'opérateur préconditionné, ce problème grossier supplémentaire permet de capter la partie de la solution appartenant à cet espace dès l'initialisation; la convergence du solveur en sera améliorée. Le choix de cet espace d'augmentation peut être issu de problèmes aux valeurs propres locaux [11–14], d'une connaissance *a priori*, ou d'une résolution précédente [15–17]. Cette dernière possibilité est particulièrement intéressante dans le cas de simulations non linéaires résolues avec une approche de type Newton-Raphson.

Ce papier est organisé comme suit : nous présentons la méthode Adaptive Multipreconditioned FETI dans un premier temps ; puis nous illustrons l'apport de l'adaptivité à deux niveaux. Une troisième partie présente quelques résultats issus de cas industriels. Ce travail est pour partie financé par le projet ANR SEMAFOR¹ qui vise à rendre accessible la simulation de propagation de fissures en plasticité généralisée. D'autre part, ces travaux ont bénéficié d'un accès aux moyens de calcul du CINES au travers de l'allocation de ressources 2017-A0040910442 attribuée par GENCI. Enfin, tous les algorithmes présentés sont implémentés dans la suite éléments finis Z-set².

2 AMPFETI : une méthode FETI multipréconditionnée et adaptative

2.1 Rappel de la méthode FETI [1]

Considérons le système linéaire de la forme $\mathbf{K}\mathbf{u} = \mathbf{f}$ issu de la discrétisation, par la méthode des éléments finis, d'un problème mécanique linéaire défini sur le domaine Ω . Le vecteur des déplacements généralisés est noté \mathbf{u} . Le vecteur des efforts généralisés \mathbf{f} comprend les efforts imposés sur la structure et éventuellement, des termes induits par des conditions aux limites de type Dirichlet non homogène. La matrice de rigidité \mathbf{K} est supposée symétrique définie positive. Soit une partition du domaine Ω en une famille de N_d sous-domaines $(\Omega^s)_{1 \leq s \leq N_d}$ non recouvrants telle que : $\bar{\Omega} = \bigcup_{s=1}^{N_d} \bar{\Omega}^s$ et $\Omega^s \cap \Omega^p = \emptyset, \forall s \neq p$. On note $\Upsilon^{sp} = \bar{\Omega}^s \cap \bar{\Omega}^p$ l'interface locale entre deux sous-domaines de la décomposition. L'union des interfaces du sous-domaine s est notée Υ^s . L'ensemble des interfaces est noté Υ . La formulation globale précédente est équivalente à la formulation sous-structurée (1)–(3), où l'exposant \square^s traduit la restriction d'une quantité à un sous domaine. Les trois systèmes d'équations suivants traduisent l'équilibre local de chaque sous structure, la continuité des déplacements et l'équilibre des inter efforts aux travers des interfaces entre sous-domaines.

$$\mathbf{K}^s \mathbf{u}^s = \mathbf{f}^s + \mathbf{t}^{s\top} \boldsymbol{\lambda}^s \quad \forall 1 \leq s \leq N_d \quad (1)$$

$$\sum_{s=1}^{N_d} \mathbf{B}^s \mathbf{t}^s \mathbf{u}^s = \mathbf{0} \quad (2)$$

$$\sum_{s=1}^{N_d} \mathbf{A}^s \boldsymbol{\lambda}^s = \mathbf{0} \quad (3)$$

Les opérateurs \mathbf{t}^s , \mathbf{A}^s et \mathbf{B}^s sont respectivement l'opérateur de trace discrète, l'opérateur d'assemblage primal et l'opérateur d'assemblage dual [18]. La méthode FETI consiste à travailler avec une unique inconnue d'interface $\boldsymbol{\lambda}$ qui assure l'équilibre des interfaces³ (3). Après condensation sur les *ddl*s d'interface, les équilibres condensés locaux sont injectés dans (2) et nous obtenons le système FETI :

$$\begin{pmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{G}^\top & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{e} \end{pmatrix} \quad (4)$$

$$\mathbf{e} = - \left(\mathbf{f}^{1\top} \mathbf{R}^1 | \dots | \mathbf{f}^{N\top} \mathbf{R}^N \right)^\top ; \quad \mathbf{d} = - \sum_s \mathbf{B}^s \mathbf{t}^s \mathbf{K}^{s+} \mathbf{f}^s$$

$$\mathbf{F} = \sum_s \mathbf{B}^s \mathbf{S}^{s\dagger} \mathbf{B}^{s\top} ; \quad \mathbf{G} = \left(\mathbf{B}^1 \mathbf{t}^1 \mathbf{R}^1 | \dots | \mathbf{B}^N \mathbf{t}^N \mathbf{R}^N \right)$$

Dans (4), les compléments de Schur primaux sont notés \mathbf{S}^s et leur (pseudo-)inverses $\mathbf{S}^{s\dagger} = \mathbf{t}^s \mathbf{K}^{s+} \mathbf{t}^{s\top}$. Le recours à une pseudo-inverse est dû à l'éventuelle présence de sous-structures flottantes pour lesquelles tous les mouvements de corps rigides ne sont pas bloqués, il existe donc des bases \mathbf{R}^s de $\text{Ker}(\mathbf{K}^s)$. La contrainte de solvabilité $\mathbf{G}^\top \boldsymbol{\lambda} = \mathbf{e}$, qui prévient de l'excitation des mouvements de corps rigides, est prise en compte grâce une stratégie d'initialisation–projection. Un projecteur $\boldsymbol{\Pi}$ est défini tel que $\mathbf{G}^\top \boldsymbol{\Pi} = \mathbf{0}$ et la solution est recherchée sous la forme :

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \boldsymbol{\Pi} \tilde{\boldsymbol{\lambda}} ; \quad \mathbf{G}^\top \boldsymbol{\lambda}_0 = \mathbf{e}$$

1. Référence : ANR-14-CE07-0037

2. <http://www.zset-software.com/>

3. Alors $\boldsymbol{\lambda}^s = \mathbf{B}^{s\top} \boldsymbol{\lambda}$.

En injectant cette forme de solution dans (4) et en prémultipliant par $\mathbf{\Pi}^\top$, le système final (5) est obtenu.

$$\mathbf{\Pi}^\top \mathbf{F} \mathbf{\Pi} \tilde{\boldsymbol{\lambda}} = \mathbf{\Pi}^\top (\mathbf{d} - \mathbf{F} \boldsymbol{\lambda}_0) \quad (5)$$

Dans la méthode FETI classique [1], le système (5), s'il est symétrique défini positif est résolu grâce à un Gradient conjugué préconditionné. Le préconditionneur est une somme pondérée de compléments de Schur locaux⁴.

$$\tilde{\mathbf{S}} = \sum_s \tilde{\mathbf{B}}^s \mathbf{S}^s \tilde{\mathbf{B}}^{s\top} \quad (6)$$

2.2 Multipreconditionnement et adaptation

La convergence du gradient conjugué est pilotée par la largeur du spectre de l'opérateur $\mathbf{\Pi}^\top \mathbf{F} \mathbf{\Pi}$. Certains cas pathologiques comme la présence de fortes hétérogénéités, d'éléments fins de mauvaises qualités, ou d'une déformation plastique élevée, génèrent de très grandes valeurs propres isolées qui ralentissent voire empêchent la convergence du solveur itératif. La méthode AMPFETI, grâce au multipreconditionnement, s'affranchit de ces difficultés de convergence en générant un espace de recherche de plus grande dimension [8, 9, 19, 20]. C'est principalement l'application du gradient conjugué multipreconditionné et adaptatif (AMPCG) [4, 6] à la formulation FETI.

Dans sa version initiale [9], l'algorithme génère, à chaque itération, autant de directions de recherche candidates qu'il y a de sous-domaines dans la décomposition. Celles-ci seront éventuellement sélectionnées en fonction du test d'adaptivité.

$$\mathbf{Z}_{i+1} = (\tilde{\mathbf{S}}^1 \mathbf{r}_{i+1} | \dots | \tilde{\mathbf{S}}^{N_d} \mathbf{r}_{i+1}) \quad (7)$$

Avec le multipreconditionnement (7), les besoins en termes de mémoire vive peuvent être importants, en particulier si le système est très mal conditionnés et que le nombre de sous-domaines est élevé. Cela a motivé le regroupement des directions de recherche candidates par agrégats de sous-domaines [19]. Le choix du nombre d'agrégat limite la consommation mémoire maximale possible et propose des directions de recherche moins « locales ». Le choix du nombre d'agrégats vient en complément du paramètre utilisateur qui pilote le critère d'adaptivité. Soit $(C_i)_{1 \leq i \leq N_c}$ un partitionnement du graphe des domaines, avec le multipreconditionnement par agrégats, l'algorithme propose à chaque itération les N_d directions de recherche candidates :

$$\mathbf{Z}_{i+1} = \left(\sum_{\Omega^s \in C_1} \tilde{\mathbf{S}}^s \mathbf{r}_{i+1} | \dots | \sum_{\Omega^s \in C_{N_c}} \tilde{\mathbf{S}}^s \mathbf{r}_{i+1} \right) \quad (8)$$

Le regroupement des sous-domaines par agrégats est issu d'un partitionnement automatique (via Scotch ou Metis) du graphe des domaines [19]. Ces algorithmes visent à minimiser le coût de coupe du partitionnement ce qui garantit une bonne localité du résidu préconditionné \mathbf{Z}_i . En effet, suite au regroupement par agrégats, un sous-domaine aura une contribution non nulle sur la colonne de \mathbf{Z}_i associée à sa couleur ainsi que sur celles correspondantes à la couleur de ses voisins. L'algorithme 1 résume les différentes étapes. Nous renvoyons à [9] pour plus de détails.

2.3 Adaptation à deux niveaux

Il semble difficile de déterminer *a priori* les agrégats de multipreconditionnement qui garantissent une convergence optimale. Sans information particulière, l'algorithme de partitionnement de graphe produit des agrégats équilibrés⁵, mais dans le cas d'inclusions localisées par exemple, nous avons montré que les directions de recherche proposées par les sous-domaines au voisinage des inclusions contiennent plus d'« information numérique » [9]. Dans ce travail, nous proposons donc d'exploiter le test d'adaptation locale pour faire évoluer les agrégats au court des itérations. Ainsi, le nombre d'agrégats reste fixe mais ces derniers évoluent en fonction du résultat du test d'adaptation locale. À chaque itération, les N_c agrégats sont séparés en trois groupes.

$$\mathcal{P}_{N_c} = \mathcal{S}_c \oplus \mathcal{J}_c \oplus \mathcal{U}$$

4. Ou d'approximations de ces derniers.

5. C'est à dire que le nombre de sous-domaines par agrégat est globalement uniforme.

Algorithme 1 : AMPFETI avec τ -test local et déflation par ajout d'un projecteur supplémentaire.

Input : Seuil de convergence $\varepsilon > 0$, le seuil $\tau > 0$ du processus adaptatif.

- 1 $\Pi = I - \mathbf{A}\mathbf{G}(\mathbf{G}^\top \mathbf{A}\mathbf{G})^{-1} \mathbf{G}^\top$
- 2 $\tilde{\lambda}_0 = \mathbf{A}\mathbf{G}(\mathbf{G}^\top \mathbf{A}\mathbf{G})^{-1} \mathbf{e}$
- 3 $\mathbf{r}_0 = \Pi^\top (\mathbf{d} - \mathbf{F}\tilde{\lambda}_0)$
- 4 $\mathbf{Z}_0 = \left(\sum_{\Omega^s \in \mathcal{C}_1} \tilde{\mathbf{S}}^s \mathbf{r}_0 \mid \dots \mid \sum_{\Omega^s \in \mathcal{C}_{N_c}} \tilde{\mathbf{S}}^s \mathbf{r}_0 \right)$
- 5 $\mathbf{W}_0 = \Pi \mathbf{Z}_0$
- 6 $\lambda_0 = \mathbf{0}, i = 0$
- 7 **while** $\|\mathbf{r}_i\| > \varepsilon$ **do**
- 8 $\mathbf{Q}_i = \mathbf{F}\mathbf{W}_i$
- 9 $\Delta_i = \mathbf{Q}_i^\top \mathbf{W}_i ; \gamma_i = \mathbf{Z}_i^\top \mathbf{r}_i ; \alpha_i = \Delta_i^\dagger \gamma_i$
- 10 $\lambda_{i+1} = \lambda_i + \mathbf{W}_i \alpha_i$
- 11 $\mathbf{r}_{i+1} = \mathbf{r}_i - \Pi^\top \mathbf{Q}_i \alpha_i$
- 12 $t_i^C = \frac{(\mathbf{W}_i \alpha_i)^\top \sum_{\Omega^s \in \mathcal{C}} \mathbf{F}^s (\mathbf{W}_i \alpha_i)}{\mathbf{r}_{i+1}^\top \sum_{\Omega^s \in \mathcal{C}} \tilde{\mathbf{S}}^s \mathbf{r}_{i+1}}$
- 13 $\mathbf{Z}_{i+1} = \text{concatenate}(\{\sum_{\Omega^s \in \mathcal{C}} \tilde{\mathbf{S}}^s \mathbf{r}_{i+1} ; t_i^C < \tau\}, \{\sum_{t_i^C > \tau} \sum_{\Omega^s \in \mathcal{C}} \tilde{\mathbf{S}}^s \mathbf{r}_{i+1}\})$
- 14 $\mathbf{W}_{i+1} = \Pi \mathbf{Z}_{i+1}$ **for** $0 \leq j \leq i$ **do** $\left\{ \begin{array}{l} \Phi_{i,j} = \mathbf{Q}_j^\top \mathbf{W}_{i+1} \\ \mathbf{W}_{i+1} \leftarrow \mathbf{W}_{i+1} - \mathbf{W}_j \Delta_j^\dagger \Phi_{i,j} \end{array} \right.$
- 15 **return** $\tilde{\lambda}_0 + \Pi \lambda_i$

L'ensemble $\mathcal{S}_c := \{C \in \mathcal{P}_{N_c} \mid t_i^C \ll \tau\}$ regroupe les agrégats candidats à la division (*split candidate*) car d'après le τ -test, les sous-domaines le peuplant contiennent vraisemblablement des informations utiles à la convergence du solveur. À l'inverse, l'ensemble $\mathcal{J}_c := \{C \in \mathcal{P}_{N_c} \mid t_i^C \gg \tau\}$ désigne les agrégats candidats au regroupement (*join candidate*). Les agrégats dans aucunes des deux situations précédentes restent inchangés $\mathcal{U} := \mathcal{P}_{N_c} \setminus (\mathcal{J}_c \cup \mathcal{S}_c)$. Une fois les agrégats classés, différentes stratégies de regroupement et de division sont possibles. L'algorithme 2 résume la procédure, on peut remarquer qu'il y a un décalage d'une itération entre la classification des agrégats et la modifications effectives de ces derniers.

Algorithme 2 : Algorithme d'adaptation des agrégats pour AMPFETI avec τ -test local.

Input : Définition initiale des clusters, stratégie de regroupement et de division des agrégats.

- 1 AMPFETI initialisation (11 – 16)
- 2 **while** $\|\mathbf{r}_i\| > \varepsilon$ **do**
- 3 AMPFETI début d'itération ($\ell 8 - \ell 11$)
- 4 Adaptation des clusters ($\mathcal{S}_c^i, \mathcal{J}_c^i, \mathcal{U}^i$)
- 5 Local test avec les nouveaux agrégats ($\ell 12$)
- 6 Classification des agrégats ($\mathcal{S}_c^{i+1}, \mathcal{J}_c^{i+1}, \mathcal{U}^{i+1}$)
- 7 AMPFETI fin d'itération ($\ell 13 - \ell 14$)

2.4 Résultats numériques

L'apport de l'adaptation à deux niveaux est présentée sur un cas académique, il s'agit d'un plaque 3D élastique où 400 inclusions très raides sont ajoutées. Celle-ci est encastrée sur un bord et un chargement de traction-flexion est imposé sur le bord opposé. Les inclusions se situent dans deux bandes prédéfinies (voir figure 1). Le module de Young des inclusions est tiré aléatoirement avec une loi de probabilité uniforme. Le rapport de module de Young entre la matrice et les inclusions est au maximum de 10^5 . Le coefficient de poisson est de 0.3. Le maillage est structuré, il se compose d'hexaèdres linéaires. Le problème global possède 32 millions de *ddls*; 480 sous-domaines sont utilisés pour la décomposition.

La figure 2 montre les courbes de convergence obtenues ainsi que les espaces de recherche générés pour différents nombre d'agrégats (60, 120, 240) et pour différentes stratégies : pas d'agrégats (courbe L noire), agrégats fixes (courbe LXX) et évolutifs (courbes U et B). La stratégies B couple un regroup-

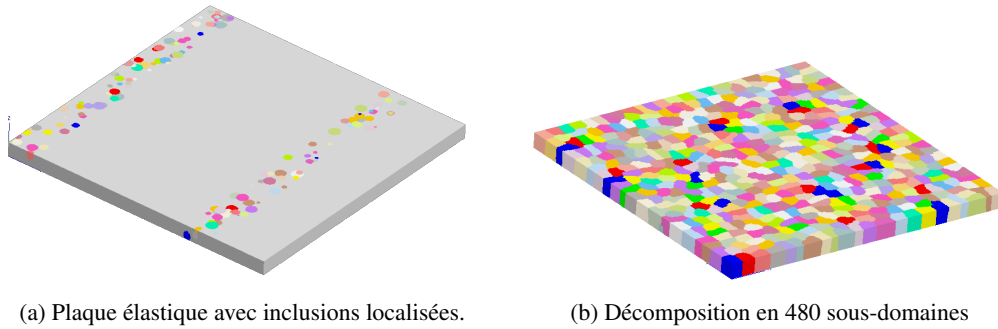


FIGURE 1 – Plaque avec des inclusions localisées

pement deux-à-deux avec une division par bisection; la stratégie notée U utilise un regroupement par composantes connexes et une division uniforme. L'apport de l'adaptation à deux niveaux est particulièrement visible pour les courbes B240 et U240 car elles convergent aussi rapidement que la courbe sans agrégats (qui dispose de la granularité de multipreconditionnement la plus fine), tout en générant un espace de recherche de plus faible dimension (de l'ordre de 20%). Cela se traduit par une réduction de la consommation mémoire du solveur. Les versions adaptatives avec 120 et 60 agrégats convergent plus lentement, elles proposent par contre un gain mémoire plus important.

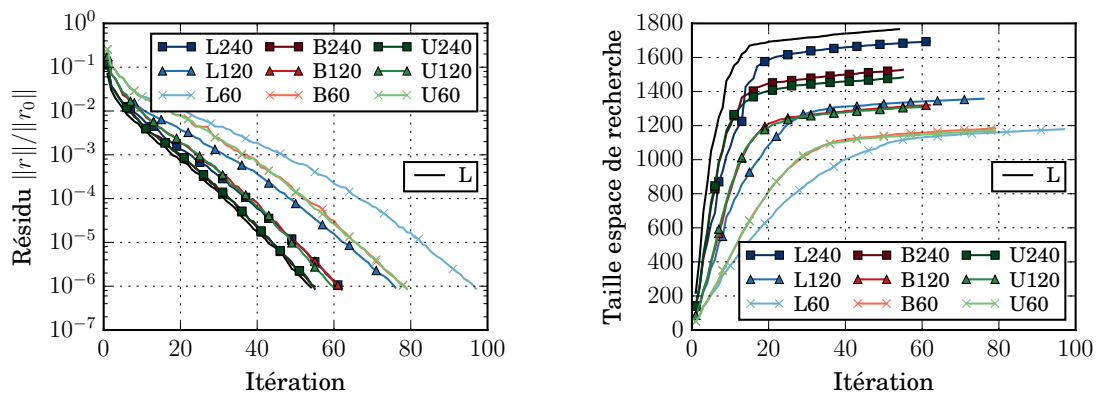


FIGURE 2 – Résultats obtenus pour la plaque avec des inclusions localisées. Le numéro indique le nombre d'agrégats. La courbe noire correspond au AMPFETI sans regroupement par agrégat. Les courbes notées L correspondent aux agrégats fixes, les courbes U et B réfèrent aux agrégats adaptatifs. Le paramètre utilisateur τ est fixé à 0.01.

3 Application à un cas réel : les propergols

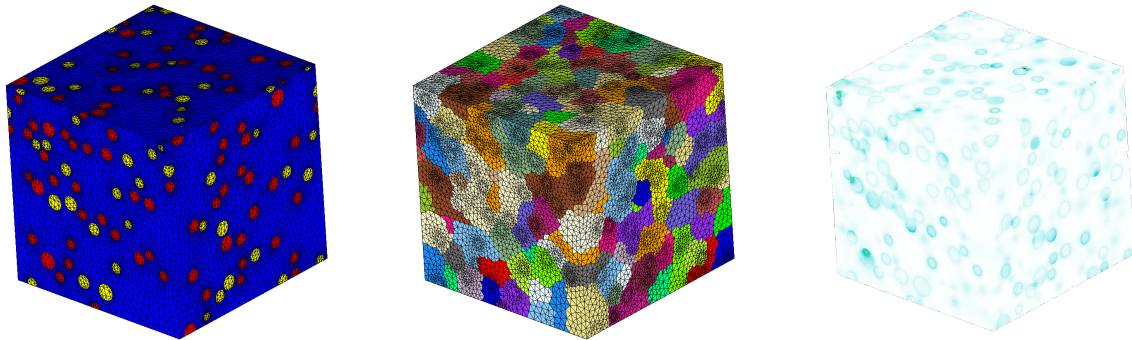
Les propergols sont des matériaux composites appartenant à la famille des matériaux énergétiques. Ils sont composés de charges énergétiques enrobées dans une matrice polyuréthane. L'objectif est de proposer une loi de comportement homogénéisée de ce matériau. Cependant, leur simulation s'avère complexe car ses spécificités physiques conduisent à de nombreuses difficultés d'un point de vue numérique. Ces spécificités et les conséquences d'un point de vue numérique sont énoncées ci-dessous :

- Le caractère fortement multi-échelle conduit, après discrétisation éléments finis à des systèmes linéaires de très grandes dimensions (plusieurs millions d'inconnues);

- La forte hétérogénéité entre la matrice et les inclusions (en termes de raideur) dégrade considérablement le conditionnement de ces systèmes linéaires ;
- Le caractère quasi-incompressible de la matrice impose l'utilisation d'éléments mixtes (déplacement, pression, variation de volume).

Le solveur AMPFETI a été utilisé pour résoudre les problèmes éléments finis nécessaires à l'homogénéisation. Dans ce travail, nous nous limitons au comportement linéarisé et restons dans le cadre des petites déformations. Néanmoins, l'utilisation du solveur pour résoudre les systèmes tangents issus de la simulation non linéaire est directe. L'hétérogénéité entre matrice et inclusions est de l'ordre de 10^5 et le coefficient de poisson de la matrice est de 0.498.

La figure 3a montre le maillage éléments finis ainsi que la décomposition en sous domaines utilisée pour l'un des packs générés. Ce derniers se compose de 5 494 528 de tétraèdres quadratiques et de 22 millions de *ddls*, la décomposition contient 448 sous domaines. Le calcul est effectué sur 448 processeurs (parallélisme de type MPI uniquement), il dure 231.5 secondes. La courbe de convergence La figure 4 montre la courbe de convergence obtenue ainsi que l'espace de recherche généré. Ici, étant donné le caractère diffus de l'hétérogénéité, c'est le test d'adaptation global qui a été utilisé avec 112 agrégats de multipréconditionnement. À titre d'exemple, la figure 3c présente la carte des contraintes de Von Mises pour le cas de compression hydrostatique.



(a) Maillage du VER, les couleurs rouges et jaunes correspondent à deux type de charges différentes.

(b) Décomposition en 448 sous-domaines obtenue avec Metis.

(c) Carte des contraintes de Von Mises, la couleur blanche correspond aux zones faiblement contraintes.

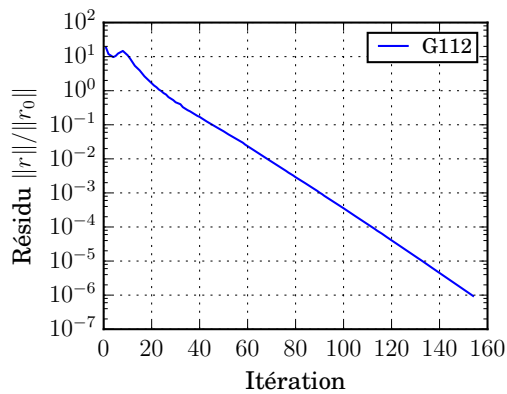
FIGURE 3 – Simulation d'un propergol.

4 Conclusion et perspectives

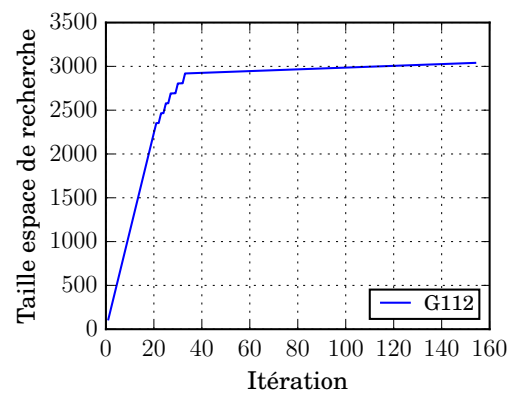
La méthode de décomposition de domaine multipréconditionnée et adaptative AMPFETI, ainsi que le principe d'adaptation à deux niveaux des agrégats de multipréconditionnement ont été présentés. L'adaptation à deux niveaux permet d'améliorer la qualité des directions de recherche candidates ce qui permet, suivant les cas, de converger aussi rapidement que la méthode à un seul niveau tout en générant un espace de recherche de plus faible dimension. L'empreinte mémoire du solveur est donc diminuée. Un cas d'application concret, la simulation de propergols, a mis en évidence les performances de la méthode.

Références

- [1] C. Farhat and F. X. Roux. The dual Schur complement method with well-posed local Neumann problems. *Contemporary Mathematics*, 157 :193, 1994.
- [2] Jan Mandel. Balancing domain decomposition. *Communications in Numerical Methods in Engineering*, 9(3) :233, 1993.
- [3] Dianne P. O'Leary. The block conjugate gradient algorithm and related methods. *Linear Algebra and its Applications*, 29 :293–322, 1980. Special Volume Dedicated to Alson S. Householder.
- [4] Robert Bridson and Chen Greif. A multipreconditioned conjugate gradient algorithm. *SIAM J. Matrix Anal. Appl.*, 27(4) :1056–1068 (electronic), 2006.



(a) Courbe de convergence du solveur itératif.



(b) Taille des espaces de recherche générés. La courbe noire notée G112 correspond au AMPFETI avec test global et 112 agrégats.

FIGURE 4 – Résultats obtenus pour le propergols. La courbe notée G112 correspondent aux agrégats fixes avec le test d’adaptation global. Le préconditionneur utilisé est de type Dirichlet avec un k -scaling. Le paramètre utilisateur τ est fixé à 0.01.

- [5] Chen Greif, Tyrone Rees, and Daniel B. Szyld. GMRES with multiple preconditioners. *SeMA Journal*, pages 1–19, 2016.
- [6] Nicole Spillane. An Adaptive Multipreconditioned Conjugate Gradient Algorithm. *SIAM J. Sci. Comput.*, 38(3) :A1896–A1918, 2016.
- [7] Christophe Bovet, Pierre Gosselet, and Nicole Spillane. Multipreconditioning for nonsymmetric problems : The case of orthomin and biCG. *Comptes Rendus Mathématique*, 355(3) :354–358, 2017.
- [8] Pierre Gosselet, Daniel Rixen, François-Xavier Roux, and Nicole Spillane. Simultaneous FETI and block FETI : Robust domain decomposition with multiple search directions. *International Journal for Numerical Methods in Engineering*, 104(10) :905–927, 2015. nme.4946.
- [9] Christophe Bovet, Augustin Parret-Fréaud, Nicole Spillane, and Pierre Gosselet. Adaptive multipreconditioned FETI : Scalability results and robustness assessment. *Computers & Structures*, 193 :1–20, 2017.
- [10] Y. Saad. Analysis of Augmented Krylov Subspace Methods. *SIAM Journal on Matrix Analysis and Applications*, 18(2) :435–449, April 1997.
- [11] Nicole Spillane and Daniel J. Rixen. Automatic spectral coarse spaces for robust FETI and BDD algorithms. *International Journal for Numerical Methods in Engineering*, 95(11) :953–990, 2013.
- [12] N. Spillane. *Robust domain decomposition methods for symmetric positive definite problems*. PhD thesis, Thèse de l’Ecole doctorale de Mathématiques de Paris centre, Laboratoire Jacques Louis Lions, Université Pierre et Marie Curie, Paris, 2014.
- [13] Jan Mandel and Bedřich Sousedík. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Comput. Methods Appl. Mech. Engrg.*, 196(8) :1389–1399, 2007.
- [14] Michael C. Leistner, Pierre Gosselet, and Daniel J. Rixen. Recycling of Solution Spaces in Multi-Preconditioned FETI Methods Applied to Structural Dynamics. *International Journal for Numerical Methods in Engineering*, 2018.
- [15] F. Risler and C. Rey. Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale non-linear problems. *Numerical algorithms*, 23 :1, 2000.
- [16] P. Gosselet. *Méthodes de décomposition de domaine et méthodes d’accélération pour les problèmes multi-champs en mécanique non-linéaire*. PhD thesis, Université Paris 6, 2003.
- [17] P. Gosselet, C. Rey, and J. Pebrel. Total and selective reuse of Krylov subspaces for the solution to a sequence of nonlinear structural problems. *International Journal for Numerical Methods in Engineering*, 94(1) :60–83, 2013.
- [18] P. Gosselet and C. Rey. Non-overlapping domain decomposition methods in structural mechanics. *Archives of Computational Methods in Engineering*, 13(4) :515–572, 2006.

- [19] Christophe Bovet, Augustin Parret-Fréaud, Nicole Spillane, and Pierre Gosselet. Méthode de décomposition de domaine multipréconditionnée et adaptative pour les problèmes mal conditionnés. In *13e colloque national en calcul des structures*, Giens, Var, France, May 2017. Université Paris-Saclay.
- [20] D. Rixen. *Substructuring and Dual Methods in Structural Analysis*. PhD thesis, Université de Liège, Belgium, Collection des Publications de la Faculté des Sciences appliquées, n.175, 1997.