

Méthode de décomposition de domaine multipréconditionnée et adaptative pour les problèmes mal conditionnés

C. Bovet¹, A. Parret-Fréaud², N. Spillane³, P. Gosselet⁴

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

² Safran Tech, Rue des Jeunes Bois, Chateaufort, CS 80112 78772 Magny-Les-Hameaux, France, augustin.parret-freaud@safrangroup.com

³ CMAP UMR 7641 Ecole Polytechnique CNRS, Route de Saclay, 91128 Palaiseau Cedex, France, nicole.spillane@polytechnique.edu

⁴ LMT, ENS Cachan, CNRS, Université Paris-Saclay, 61 av. du Président Wilson, 94235 Cachan, France, pierre.gosselet@lmt.ens-cachan.fr

Résumé — Nous présentons la méthode de décomposition de domaine multipréconditionnée et adaptative AMPFETI, qui vise à résoudre des problèmes de très grandes tailles et de complexités industrielles. La méthode AMPFETI est robuste et particulièrement efficace pour résoudre des problèmes mal conditionnés impliquant par exemple de fortes hétérogénéités. Ce travail présente une amélioration de la méthode où la « granularité » du multipréconditionnement n'est plus systématiquement le sous-domaine. Cette récente amélioration rend accessibles des problèmes à un grand nombre de sous-domaines.

Mots clés — Décomposition de domaine, FETI, préconditionneurs multiples, préconditionneurs adaptatifs, solveur de Krylov robuste.

1 Introduction

L'accroissement quasi-constant de la puissance des supercalculateurs mondiaux¹ rend envisageables les simulations aux giga-échelles. Le calcul de structure est naturellement attiré par cette tendance, qu'il s'agisse de décrire finement le comportement d'un matériau complexe (simulation de pièces polycristallines avec description des grains par exemple), ou bien de simuler le comportement de structures complexes comme le fuselage complet d'un avion. Résoudre des problèmes discrétisés de grandes tailles n'est plus le domaine réservé de la recherche académique et l'industrie exprime un réel besoin de méthodes de calcul haute performance.

Les méthodes de décomposition de domaine de type Schur, c'est-à-dire les méthodes FETI [1] ou BDD [2] et leurs variantes avec contraintes [3, 4] permettent d'exploiter pleinement l'architecture parallèle des supercalculateurs en limitant les communications globales entre processeurs. Comme tout solveur itératif, ces méthodes sont néanmoins sensibles au conditionnement du système à résoudre et il existe quelques situations pathologiques [5] qui ralentissent particulièrement, ou même empêchent, leur convergence. La présence d'interfaces « chahutées », de sous-domaines mal proportionnés ou d'hétérogénéités matérielles importantes, traversant une interface ou un sous domaine, sont quelques unes de ces situations pathologiques. Celles-ci sont malheureusement fréquemment observées lors de simulations de cas industriels qui recourent à une décomposition générée automatiquement. À cause de cette sensibilité peu d'industriels utilisent aujourd'hui les méthodes de décomposition de domaine pour réaliser leurs simulations.

Ces dernières années deux familles de solutions se détachent pour améliorer la robustesse des méthodes de décomposition de domaine. La première se traduit par l'ajout d'un problème grossier supplémentaire, dit d'augmentation, qui permet d'éliminer les modes problématiques de l'opérateur préconditionné. Ces modes peuvent être détectés grâce à des problèmes aux valeurs propres locaux [6, 7, 8]. La seconde se caractérise par l'usage de préconditionneurs multiples, elle correspond aux solveurs de Krylov multipréconditionnés [9, 10, 11]. La méthode Adaptive Multipreconditioned FETI présentée appartient à cette seconde catégorie.

Ce papier est organisé comme suit : nous rappelons la méthode FETI dans un premier temps ; puis nous introduisons la méthode *Adaptive Multipreconditioned FETI* et illustrons l'apport du multipréconditionnement et de l'adaptativité. Une récente amélioration qui consiste à regrouper les sous-domaines en

1. www.top500.org/statistics/perfdevel

clusters est ensuite présentée. Une étude de type extensibilité faible pour un cas fortement hétérogène met en évidence l'intérêt de cette amélioration. Ce travail est pour partie financé par le projet ANR SEMAFOR² qui regroupe, instituts de recherche, laboratoires et industriels autour de la simulation de propagation de fissures en plasticité généralisée. Nous tenons également à remercier le Grand Équipement National de Calcul Intensif et le Centre Informatique National de l'Enseignement Supérieur qui nous ont donné accès au supercalculateur OCCIGEN. Tous les algorithmes présentés sont implémentés dans la suite éléments finis Z-set 8.6³.

2 Rappel de la méthode FETI [1]

2.1 Formulation sous-structurée duale

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), 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 [12]. Une propriété importante des opérateurs d'assemblage primal et dual est leur orthogonalité $\sum_{s=1}^{N_d} \mathbf{B}^s \mathbf{A}^{s\top} = \mathbf{0}$. La méthode FETI consiste à choisir une unique inconnue d'interface $\boldsymbol{\lambda}$ telle que $\boldsymbol{\lambda}^s = \mathbf{B}^{s\top} \boldsymbol{\lambda}$. L'équilibre de l'interface est par construction satisfaite, le problème est reformulé.

$$\begin{aligned} \mathbf{K}^s \mathbf{u}^s &= \mathbf{f}^s + \mathbf{t}^{s\top} \mathbf{B}^{s\top} \boldsymbol{\lambda} \quad \forall 1 \leq s \leq N_d \\ \sum_s \mathbf{B}^s \mathbf{t}^s \mathbf{u}^s &= \mathbf{0} \end{aligned} \quad (4)$$

Ce problème est ensuite condensé sur l'interface. Notons i les degrés de liberté (*ddls*) internes au domaine et b les degrés de liberté sur les interfaces Υ . Les compléments de Schur sont notés $\mathbf{S}^s = \mathbf{K}_{ii}^s - \mathbf{K}_{bi}^s \mathbf{K}_i^{s-1} \mathbf{K}_{ib}^s$; et leur (pseudo-)inverses $\mathbf{S}^{s\dagger} = \mathbf{t}^s \mathbf{K}^{s\dagger} \mathbf{t}^{s\top}$. 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. Soit \mathbf{R}^s une base de $\text{Ker}(\mathbf{K}^s)$, posons :

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

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

2. Référence : ANR-14-CE07-0037. Partenaires : Armines CDM, LMT Cachan, Onera, Safran

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

En éliminant l'inconnue déplacement dans (4), nous obtenons la formulation FETI qui consiste à trouver (λ, β) tels que :

$$\begin{pmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{G}^\top & \mathbf{0} \end{pmatrix} \begin{pmatrix} \lambda \\ \beta \end{pmatrix} = \begin{pmatrix} d \\ e \end{pmatrix} \quad (5)$$

La contrainte de solvabilité $\mathbf{G}^\top \lambda = e$, qui prévient de l'excitation des mouvements de corps rigides, est prise en compte grâce une stratégie d'initialisation–projection. Un projecteur $\mathbf{\Pi}$ est défini tel que $\mathbf{G}^\top \mathbf{\Pi} = \mathbf{0}$ et la solution est recherchée sous la forme :

$$\lambda = \lambda_0 + \mathbf{\Pi} \tilde{\lambda} \quad ; \quad \mathbf{G}^\top \lambda_0 = e$$

En injectant cette forme de solution dans (5), et en prémultipliant par $\mathbf{\Pi}^\top$, le système final (6) est obtenu.

$$\mathbf{\Pi}^\top \mathbf{F} \mathbf{\Pi} \tilde{\lambda} = \mathbf{\Pi}^\top (d - \mathbf{F} \lambda_0) \quad (6)$$

2.2 Méthode FETI et solveur de Krylov

La méthode FETI est la combinaison de la formulation sous-structurée duale précédente avec un solveur itératif de type Krylov. Dans le cas symétrique défini positif, un Gradient conjugué est classiquement utilisé pour résoudre (6). Le préconditionneur choisi est une somme pondérée de compléments de Schur primaux⁴ locaux.

$$\tilde{\mathbf{S}} = \sum_s \tilde{\mathbf{B}}^s \mathbf{S}^s \tilde{\mathbf{B}}^{s\top} \quad (7)$$

Soit respectivement $\tilde{\lambda}_i$ et r_i , la solution approchée et le résidu à l'itération i . L'algorithme FETI-CG génère une base \mathbf{F} -orthogonale $(w_i)_i$ de l'espace de Krylov associé \mathcal{K}_i . La solution approchée à l'itération i vérifie (8) où λ^* est la solution du problème.

$$\tilde{\lambda}_i = \operatorname{argmin} \|\tilde{\lambda} - \mathbf{\Pi} \lambda^*\|_{\mathbf{F}} \quad \forall \tilde{\lambda} \in \bigoplus_{j=0}^{i-1} \operatorname{Vect}(w_j) \quad (8)$$

3 Présentation de la méthode *Adaptive Multipreconditioned FETI*

3.1 Multipréconditionnement & adaptivité

Il existe certains cas pathologiques, comme la présence de fortes hétérogénéités traversant une interface ou un sous-domaine, où la convergence de la méthode FETI est fortement ralentie [5]. Ces difficultés sont liées au caractère trop local du préconditionneur (7). La méthode Simultaneous FETI [13, 14] a été proposée pour résoudre ces problèmes fortement hétérogènes. En gardant distinctes les contributions de chaque sous-domaine lors du préconditionnement, elle construit un résidu préconditionné de la forme $\mathbf{Z}_{i+1} = (\tilde{\mathbf{S}}^1 r_{i+1} | \dots | \tilde{\mathbf{S}}^{N_d} r_{i+1})$ où $\tilde{\mathbf{S}}^s = \tilde{\mathbf{B}}^s \mathbf{S}^s \tilde{\mathbf{B}}^{s\top}$. À chaque itération, N_d directions de recherche sont générées. Cette méthode s'apparente en fait à l'application du gradient conjugué multipréconditionné (MPCG) [9] à la formulation FETI. Ce multipréconditionnement génère un espace de recherche de plus grande dimension et permet une convergence rapide. Le coût d'une itération, en terme d'opérations et de mémoire, est par contre plus élevé.

La méthode AMPFETI proposée suit la même démarche, c'est l'application du gradient multipréconditionné adaptatif récemment développé (AMPCG) [11] à la formulation FETI. À chaque itération, l'algorithme AMPCG choisit entre 1 et N_d directions de recherche à conserver. L'algorithme 1 présente l'application de l'algorithme AMPCG à la méthode FETI. Il regroupe les trois variantes proposées par AMPCG. Seule l'étape de préconditionnement diffère entre ces variantes (lignes 13–19). La première ne comporte pas de test (ligne 14), c'est exactement une itération de type Simultaneous FETI (ou MPFETI) ; N_d directions de recherche sont générées. La seconde, dénommée AMPFETIG, présente un comportement tout-ou-rien (ligne 15) ; suivant l'état du τ -test global l'algorithme réalise une itération de type FETI ou MPFETI. Le τ -test local de la dernière variante, AMPFETIL, permet une plus grande finesse ; pour chaque sous-domaine le test détermine si la direction de recherche doit être conservée ou non. Les

4. Ou d'approximations de ces derniers.

directions non conservées sont collectées et sommées. Le nombre de directions générées peut prendre toutes les valeurs entre 1 et N_d .

Algorithme 1 : Adaptive Multipreconditioned FETI. L'utilisateur fournit la tolérance $\varepsilon > 0$ et le seuil $\tau > 0$ du processus adaptatif. N vaut N_d pour la version classique et N_c lorsqu'un regroupement par *clusters* est utilisé.

```

1  $\tilde{\lambda}_0 = \mathbf{0}$ 
2  $\mathbf{r}_0 = \mathbf{\Pi}^\top (\mathbf{d} - \mathbf{F}\lambda_0)$ 
3  $\mathbf{Z}_0 = (\tilde{\mathbf{S}}^1 \mathbf{r}_0 | \dots | \tilde{\mathbf{S}}^N \mathbf{r}_0)$ 
4  $\mathbf{W}_0 = \mathbf{\Pi} \mathbf{Z}_0$ 
5  $i = 0$ 
6 tant que  $\sqrt{\mathbf{r}_i^\top \mathbf{Z}_i \mathbf{1}} > \varepsilon$  faire
7    $\mathbf{Q}_i = \mathbf{F} \mathbf{W}_i = \sum_s \mathbf{F}^s \mathbf{W}_i = \sum_s \mathbf{B}^s \mathbf{S}^{s\dagger} \mathbf{B}^{s\top} \mathbf{W}_i$ 
8    $\mathbf{\Delta}_i = \mathbf{Q}_i^\top \mathbf{W}_i$ 
9    $\boldsymbol{\gamma}_i = \mathbf{Z}_i^\top \mathbf{r}_i$ 
10   $\boldsymbol{\alpha}_i = \mathbf{\Delta}_i^\dagger \boldsymbol{\gamma}_i$ 
11   $\tilde{\lambda}_{i+1} = \tilde{\lambda}_i + \mathbf{W}_i \boldsymbol{\alpha}_i$ 
12   $\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{\Pi}^\top \mathbf{Q}_i \boldsymbol{\alpha}_i$ 
13  si No  $\tau$ -test alors
14     $\mathbf{Z}_{i+1} = (\tilde{\mathbf{S}}^1 \mathbf{r}_{i+1} | \dots | \tilde{\mathbf{S}}^N \mathbf{r}_{i+1})$ 
15  sinon si Global  $\tau$ -test alors
16     $\mathbf{Z}_{i+1} = \begin{cases} (\tilde{\mathbf{S}}^1 \mathbf{r}_{i+1} | \dots | \tilde{\mathbf{S}}^N \mathbf{r}_{i+1}) & \text{if } t_i := (\boldsymbol{\gamma}_i^\top \boldsymbol{\alpha}_i) / (\mathbf{r}_{i+1}^\top \tilde{\mathbf{S}} \mathbf{r}_{i+1}) < \tau, \\ \tilde{\mathbf{S}} \mathbf{r}_{i+1} & \text{otherwise.} \end{cases}$ 
17  sinon si Local  $\tau$ -test alors
18     $t_i^s = \frac{(\mathbf{W}_i \boldsymbol{\alpha}_i)^\top \mathbf{F}^s (\mathbf{W}_i \boldsymbol{\alpha}_i)}{\mathbf{r}_{i+1}^\top \tilde{\mathbf{S}} \mathbf{r}_{i+1}}$ 
19     $\mathbf{Z}_{i+1} = \text{concatenate}(\{\tilde{\mathbf{S}}^s \mathbf{r}_{i+1}; t_i^s < \tau\}, \sum_{t_i^s > \tau} \tilde{\mathbf{S}}^s \mathbf{r}_{i+1})$ 
20   $\mathbf{W}_{i+1} = \mathbf{\Pi} \mathbf{Z}_{i+1}$ 
21  pour  $0 \leq j \leq i$  faire  $\begin{cases} \Phi_{i,j} = \mathbf{Q}_j^\top \mathbf{W}_{i+1} \\ \mathbf{W}_{i+1} \leftarrow \mathbf{W}_{i+1} - \mathbf{W}_j \mathbf{\Delta}_j^\dagger \Phi_{i,j} \end{cases}$ 
22   $i \leftarrow i + 1$ 
23 retourner  $\tilde{\lambda}_i$ 

```

Le seuil τ contrôle la sélectivité du processus d'adaptivité : une grande valeur de τ conduit à des itérations multipréconditionnées, une faible valeur entraîne des itérations de type FETI. Le corollaire 3.3 de [11], propose une relation entre la valeur de τ et la diminution de l'erreur souhaitée ρ . La méthode AMPFETI vérifie les propriétés suivantes :

1. Les directions de recherche produites par AMPFETI sont \mathbf{F} -orthogonales :

$$\mathbf{W}_i^\top \mathbf{F} \mathbf{W}_j = \mathbf{W}_i^\top \mathbf{Q}_j = \mathbf{0} \quad \forall i \neq j \quad (9)$$

2. À l'itération i , la solution approchée $\tilde{\lambda}_i$ produite par AMPFETI minimise la \mathbf{F} -norme de l'erreur :

$$\tilde{\lambda}_i = \operatorname{argmin} \|\tilde{\lambda} - \mathbf{\Pi} \boldsymbol{\lambda}^*\|_{\mathbf{F}} \quad \forall \tilde{\lambda} \in \bigoplus_{j=0}^{i-1} \operatorname{Vect}(\mathbf{W}_j) \quad (10)$$

L'adaptivité permet d'anticiper si une itération de type FETI serait suffisante, elle permet d'élargir l'espace de recherche uniquement lorsque cela s'avère nécessaire.

3.2 Résultats numériques

Ce premier exemple concerne une plaque 3D d'éclancement 20 dans la direction x et 10 suivant y (cf. Fig. 1). Cette plaque est encastrée sur la surface $x = 0$ et une combinaison de pression et de cisaillement est imposée en $x = 20$. Le maillage est réglé et comporte $15 \times 250 \times 125$ éléments de type c3d20, ce

qui conduit à un problème global à 5 959 218 degrés de liberté. Une décomposition automatique en 127 sous-domaines est utilisée. Cette plaque est composée d'une matrice linéaire élastique de caractéristiques $(\nu_m, E_m) = (0.45, 1)$. Des inclusions, au comportement élastique linéaire, sont aléatoirement ajoutées dans cette matrice. Le coefficient de poisson des inclusions est $\nu_i = 0.3$ et les modules de Young $(E_i)_i$ sont également tirés aléatoirement en utilisant une loi de probabilité uniforme. La valeur maximale de E_i/E_m est 10^5 . Deux cas tests sont considérés ; les inclusions sont tirées dans le volume complet pour le premier (Fig. 1b), elles sont localisées dans le second (Fig. 1c).

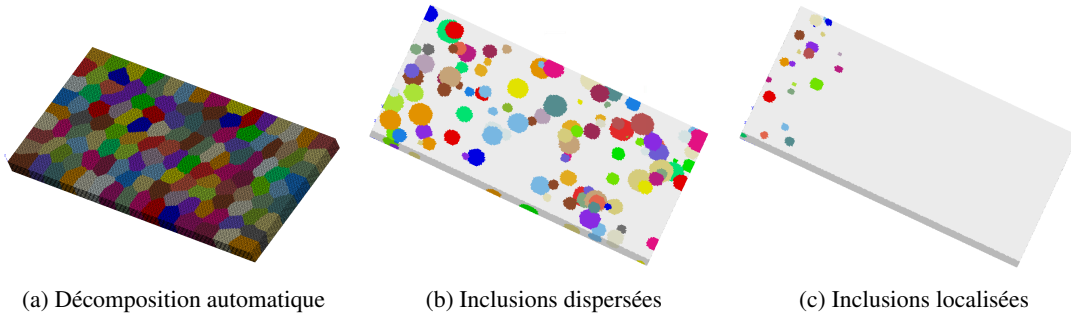


FIGURE 1 – Plaque 3D avec inclusions hétérogènes

Les résultats des méthodes FETI, MPFETI et AMPFETI, en termes de nombre d'itérations⁵, de taille d'espace de recherche et de temps de calcul, sont regroupés dans la Table 1. Les compteurs Précond. Opérateur et Orthog. correspondent respectivement aux lignes 13–19, ligne 7 et ligne 21. Quel que soit le cas considéré, les méthodes multipréconditionnées convergent plus rapidement que la méthode FETI, à la fois en terme de temps et de nombre d'itérations. Comparées à la méthodes MPFETI, les méthodes adaptatives produisent un espace de recherche de taille réduite, sans dégrader significativement la convergence du solveur. Le temps requis pour orthogonaliser les blocs de directions de recherche $(\mathbf{W}_i)_i$ est ainsi réduit, il devient comparable à celui de la méthode FETI. L'apport du τ -test local est particulièrement visible lorsque les inclusions sont localisées car seules les directions de recherche associées aux sous-domaines proches des inclusions sont conservées. Sur ce cas, le test global de la méthode AMPFETIG ne permet pas une telle sélection et l'espace généré semble surdimensionné.

	Inclusions dispersées ($\tau = 0.1$)					
	Compteurs		Temps (s)			
	# iter.	# dir. recherche	Total	Précond.	Opérateur	Orthog.
FETI	487	487	239.2	89.90 (37.6%)	75.68 (31.6%)	8.78 (3.7%)
MPFETI	30	3810	117.2	14.41 (12.3%)	19.73 (16.8%)	35.18 (30.0%)
AMPFETIG	38	1550	76.33	14.94 (19.6%)	14.40 (18.9%)	8.12 (10.6%)
AMPFETIL	36	1219	81.79	21.65 (26.5%)	14.15 (17.3%)	7.41 (9.1%)
	Inclusions localisées ($\tau = 0.1$)					
	# iter.	# dir. recherche	Total	Précond.	Opérateur	Orthog.
FETI	181	181	126.2	40.91 (32.4%)	33.33 (26.4%)	1.47 (1.2%)
MPFETI	24	3048	110.4	14.39 (13.0%)	20.52 (18.6%)	23.94 (21.7%)
AMPFETIG	28	1666	86.05	16.11 (18.7%)	16.17 (18.8%)	9.28 (10.8%)
AMPFETIL	27	486	73.3	18.78 (25.6%)	12.37 (16.9%)	2.45 (3.3%)

TABLE 1 – Inclusions hétérogènes : récapitulatif des résultats

4 Amélioration de AMPFETI

4.1 Choix de la granularité du multipréconditionnement

En générant un espace de recherche de plus grande dimension, les méthodes multipréconditionnées convergent plus rapidement que la méthode FETI. Elles sont néanmoins plus consommatrices de mé-

5. Le solveur est considéré convergé lorsque le résidu initial a été réduit d'un facteur 10^6 .

moire. En effet, si l'on considère une décomposition régulière du cube en damier visible sur la Figure 2, et si l'on note n le nombre de nœuds suivant une arête d'une case du damier ; le coût de stockage des blocs de vecteurs $(\mathbf{W}_i)_i$ et $(\mathbf{Q}_i)_i$, pour la méthode MPFETI, est en $O(n^2 N_d n_{iter})$ ⁶. Le coût de stockage des matrices denses $(\mathbf{\Delta}_i)_i$ est lui proportionnel à $N_d^2 n_{iter}$. Ce surcoût n'est plus négligeable lorsque N_d est de l'ordre de grandeur de n et peut s'avérer problématique lorsqu'un très grand nombre de sous-domaines est requis. Cette limite est particulièrement présente pour AMPFETIG à cause du comportement tout-ou-rien. Le τ -test local permet une meilleure finesse dans la sélection des directions de recherche. Cependant, bien que le paramètre utilisateur τ pilote la sélectivité du processus et influe sur nombre de directions de recherche conservées, le lien entre τ et la taille de l'espace de recherche généré n'est pas direct. Et, pour des problèmes mal conditionnés, le τ -test local sélectionne souvent toutes les directions de recherche lors des premières itérations et cela même avec une valeur de τ faible.

En complément du paramètre τ , nous proposons de choisir *a priori* la granularité du multiprécconditionnement. Les sous-domaines sont regroupés en N_c *clusters* et, au lieu de construire $\mathbf{Z}_{i+1} = (\tilde{\mathbf{S}}^1 \mathbf{r}_{i+1} | \dots | \tilde{\mathbf{S}}^{N_d} \mathbf{r}_{i+1})$ et de générer N_d directions de recherche candidates, nous formons (11) et proposons N_c directions de recherche. Ici, $(C_i)_i$ désigne un regroupement de sous-domaines.

$$\mathbf{Z}_{i+1} = \left(\sum_{\Omega^s \in C_1} \tilde{\mathbf{S}}^s \mathbf{r}_{i+1} | \dots | \sum_{\Omega^s \in C_{N_c}} \tilde{\mathbf{S}}^s \mathbf{r}_{i+1} \right) \quad (11)$$

Plus précisément, on considère le graphe valué $G = (\{\Omega_1, \dots, \Omega^{N_d}\}, \Upsilon)$ où les domaines Ω^s sont les sommets du graphe. Assez naturellement, une arête relie deux sommets Ω^s, Ω^p s'il existe une interface commune Υ^{sp} . Le poids de chaque sous-domaine est supposé unitaire. Soit une partition $P_{N_c} = \{C_1, \dots, C_{N_c}\}$ en N_c parties des sommets du graphe. Un algorithme de type Algo. 1, qui utilise le multiprécconditionnement (11), reste dans le cadre AMPCG, il vérifie les propriétés (9)(10). Les τ -test sont également toujours valides, ils sont par contre définis à l'échelle des *clusters*.

Une différence notable porte sur la localité du résidu précconditionné \mathbf{Z}_i . Dans la version classique, la colonne s de \mathbf{Z}_i est non nulle uniquement sur les *ddl*s concernant le sous-domaine s et ses voisins. Suite au regroupement par *clusters*, un sous-domaine s 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. Le caractère creux de \mathbf{Z}_i impacte directement le coût calcul de $\mathbf{Q}_i = \mathbf{F}\mathbf{W}_i$ et donc les performances du solveur. En pratique, le projecteur $\mathbf{\Pi}$ est décomposé (cf. [14]) et le calcul de \mathbf{Q}_i requiert le calcul de $\mathbf{F}\mathbf{Z}_i$. Chaque itération implique la résolution de problèmes de Neumann locaux à seconds membres multiples et le nombre de seconds membres dépend directement du caractère creux de \mathbf{Z}_i . Il est donc essentiel que la partition minimise le nombre de sous-domaines voisins des couleurs $(C_i)_i$. Cela correspond au coût de coupe de la partition (12). Minimiser ce coût de coupe est généralement l'un des objectifs des décomposeurs automatiques comme Scotch ou Metis.

$$\text{coupe}(P_{N_c}) = \sum_{i < j} \text{coupe}(C_i, C_j) = \sum_{i < j} \sum_{\substack{\Omega^k \in C_i \\ \Omega^l \in C_j}} \text{poids}(\Omega^k, \Omega^l) = \sum_{i < j} \text{card}\left(\left\{\Upsilon^{kl}, \Omega^k \in C_i, \Omega^l \in C_j\right\}\right) \quad (12)$$

4.2 Résultats numériques

Pour illustrer les propos précédents, les résultats d'une étude de type extensibilité faible sont présentés (voir Figure 2). Le problème considéré est un cube en damier hétérogène. Le comportement est élastique linéaire ($\nu = 0.3$) et le coefficient d'hétérogénéité est 10^6 . Le maillage est un maillage réglé, composé d'éléments c3d8. Une décomposition automatique est utilisée et le nombre de sous-domaines est égal au nombre de cases du damier. Une face est encastrée et une combinaison de pression et cisaillement est imposée sur la face opposée. Les sous-domaines comportent en moyenne 80 000 *ddl*s et le ratio h/H est de 29. Trois nombres de *clusters* sont proposés, 64, 125 et 216. Le nombre de sous-domaines utilisés est compris entre 2^3 et 9^3 .

Les résultats sont regroupés dans la Table 2 et sur la Figure 3. À nouveau, la méthode FETI requiert plus d'itérations pour converger, ce nombre augmente fortement avec le nombre de sous-domaines. Il est multiplié par cinq entre 64 et 729 sous-domaines. L'augmentation du nombre d'itérations est beaucoup

6. Par comparaison, le coût de stockage des directions de recherche de la méthode FETI est en $O(n^2 n_{iter})$.

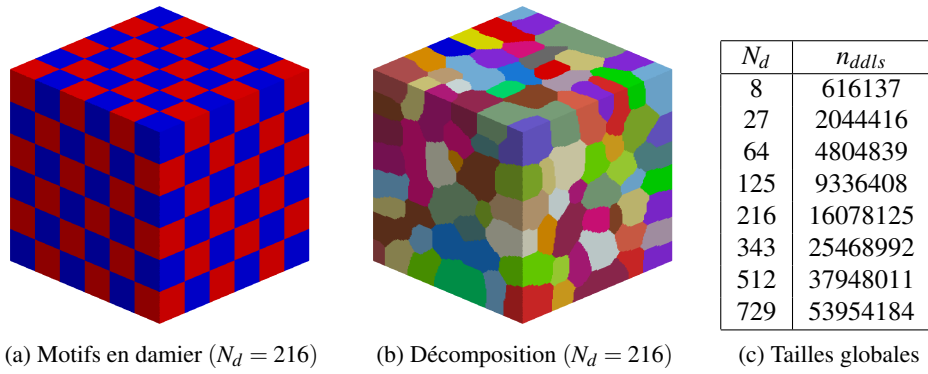


FIGURE 2 – Classe de cas tests pour l'étude d'extensibilité faible : damier cubique hétérogène

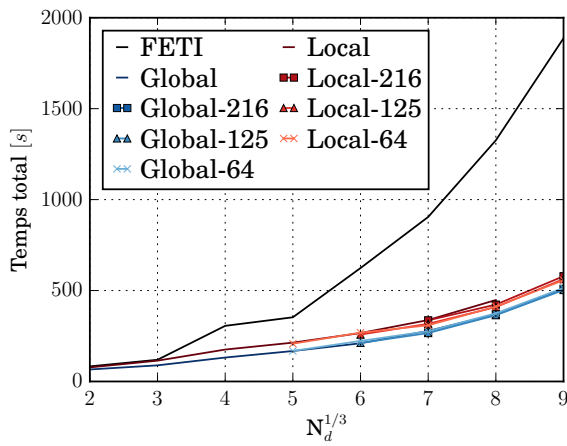
plus faible pour les méthodes multipréconditionnées. Quelle que soit l'adaptivité utilisée, l'utilisation de clusters permet de réduire la taille de l'espace de recherche sans significativement augmenter le nombre d'itérations. L'impact sur le temps de calcul est limité (cf. Fig 3a). La diminution du nombre de clusters augmente le nombre d'itérations et réduit la taille de l'espace de recherche. Ce phénomène est particulièrement visible sur le cas à 729 sous-domaines (Fig. 3b). Pour ce cas, la mémoire consommée par les méthodes sans regroupement par clusters excède la limite autorisée pour le calcul (5.2 Go/coeur). Il est évidemment possible d'augmenter cette limite ou de rendre le processus d'adaptivité plus sélectif. Pour ces problèmes mal conditionnés, le processus d'adaptivité sélectionne généralement l'ensemble des directions de recherche lors des premières itérations. Le regroupement par *clusters* permet cependant de piloter la croissance de l'espace de recherche (cf. Fig. 3b), en définissant l'augmentation maximale par itération.

N_d	Nombre d'itérations						Nombre de directions de recherche					
	64	125	216	343	512	729	64	125	216	343	512	729
FETI	381	435	732	1075	1474	1956	381	435	732	1075	1474	1956
AMPFETIG	96	121	105	105	117	M	663	865	1825	3183	4205	>4374
AMPFETIG-216				122	130	136				2487	3355	4436
AMPFETIG-125			111	125	138	143			1599	2233	3114	4111
AMPFETIG-64		118	130	141	150	170		811	1264	1968	2859	3635
AMPFETIL	85	92	94	94	96	M	668	898	1613	2724	4024	>4325
AMPFETIL-216				105	113	122				2487	3372	4320
AMPFETIL-125			104	116	127	137			1449	2238	3064	3991
AMPFETIL-64		103	117	132	147	162		789	1266	1961	2734	3670

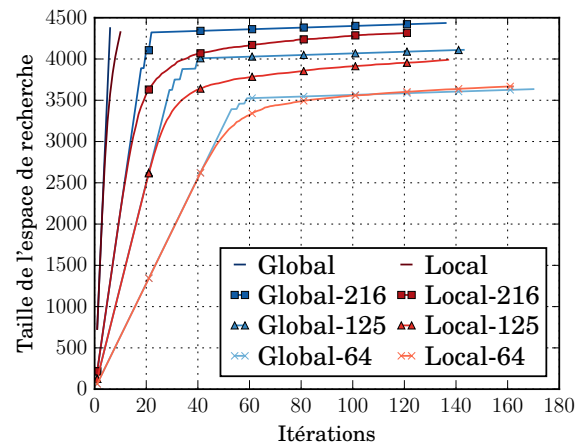
TABLE 2 – Extensibilité faible pour un cube en damier hétérogène : récapitulatif des résultats. Paramètres de l'étude : $E_1/E_2 = 10^6$, $\nu = 0.3$, $\tau = 10^{-2}$ et $h/H = 29$. La notation *M* signifie que la simulation a excédé la limite de mémoire allouée (5.2 Go/coeur).

5 Conclusion et perspectives

La méthode de décomposition de domaine multipréconditionnée et adaptative AMPFETI a été présentée. L'usage de préconditionneurs multiples permet aux méthodes de type MPFETI de converger en un nombre faible d'itérations. Grâce au processus d'adaptivité, la taille de l'espace de recherche s'adapte au conditionnement du problème à résoudre. Une modification de la méthode permet de contrôler la granularité du multipréconditionnement. En complément du paramètre utilisateur τ , ce regroupement par *clusters* permet de réduire l'empreinte mémoire du solveur et rend accessible des problèmes à un très grand nombre de sous-domaines. Deux cas tests dont une étude de type extensibilité faible pour un cas fortement hétérogène illustrent l'intérêt des solveurs proposés.



(a) Extensibilité faible, temps de calcul



(b) Évolution de l'espace de recherche au cours des itérations (cas à 729 sous-domaines)

FIGURE 3 – Extensibilité faible pour un cube en damier hétérogène. Paramètres de l'étude : $E_1/E_2 = 10^6$ et $\tau = 10^{-2}$, $h/H = 29$.

Références

- [1] C. Farhat, F. X. Roux. *Implicit parallel processing in structural mechanics*. Computational Mechanics Advances, 2(1) :1–124, 1994. North-Holland.
- [2] J. Mandel. *Balancing domain decomposition*. Communications in Numerical Methods in Engineering, 9(3) :233, 1993.
- [3] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, D. Rixen. *FETI-DP : a Dual-Primal Unified FETI Method - Part I : a Faster Alternative to the Two-Level FETI Method*. International Journal for Numerical Methods in Engineering, 50(7) :1523–1544, 2001.
- [4] C. R. Dohrmann. *A preconditioner for substructuring based on constrained energy minimization*. SIAM Journal for Scientific Computing, 25 :246, 2003.
- [5] C. Pechstein, R. Scheichl. *Analysis of FETI methods for multiscale PDEs. Part II : interface variation*. Numer. Math., 118(3) :485–529, 2011.
- [6] N. Spillane, D. J. Rixen. *Automatic spectral coarse spaces for robust FETI and BDD algorithms*. International Journal for Numerical Methods in Engineering, 95(11) :953–990, 2013.
- [7] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, R. Scheichl. *Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps*. Numer. Math., 126(4) :741–770, 2014.
- [8] J. Mandel, B. 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.
- [9] R. Bridson, C. Greif. *A multipreconditioned conjugate gradient algorithm*. SIAM J. Matrix Anal. Appl., 27(4) :1056–1068, 2006.
- [10] C. Greif, T. Rees, D. B. Szyld. *Gmres with multiple preconditioners*. SeMA Journal, pages 1–19, 2016.
- [11] N. Spillane. *An Adaptive Multipreconditioned Conjugate Gradient Algorithm*. SIAM J. Sci. Comput., 38(3) :A1896–A1918, 2016.
- [12] P. Gosselet, C. Rey. *Non-overlapping domain decomposition methods in structural mechanics*. Archives of Computational Methods in Engineering, 13(4) :515–572, 2006.
- [13] 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.
- [14] P. Gosselet, D. J. Rixen, F.-X. Roux, N. 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.