

Stratégie hiérarchique de sous-structuration et maillage pour la simulation HPC

Y. El Gharbi^{1,2}, P. Gosselet¹, A. Parret-Fréaud², C. Bovet³

¹ LMT, École Normale Supérieure Paris-Saclay, {elgharbi,gosselet}@lmt.ens-cachan.fr

² Safran S.A., augustin.parret-freaud@safrangroup.com

³ Onera — The French Aerospace Lab F-92322 Chatillon, France, christophe.bovet@onera.fr

Résumé — Une nouvelle approche de sous-structuration hiérarchique est développée et étudiée pour pallier le manque de robustesse des solveurs parallèles de décomposition de domaine sur des problèmes industriels. Cette méthode permet, en plus de l'amélioration du conditionnement des problèmes à résoudre, une parallélisation des étapes de pré-traitement du maillage. Sur une première étude académique, cette méthode a montré une convergence plus rapide de l'algorithme FETI par rapport à l'emploi d'une approche classique avec décomposition automatique.

Mots clés — décomposition de domaine, FETI, maillage hiérarchique, déformation de maillage, HPC, calcul parallèle.

1 Introduction

Les méthodes de décomposition de domaine sans recouvrement, comme FETI¹ [4] et BDD² [11], offrent un cadre favorable pour la définition de solveurs hybrides adaptés à l'architecture massivement parallèle des supercalculateurs récents et à même de résoudre des problèmes linéaires et non linéaires en mécanique des structures. Une revue de ces méthodes est disponible dans [6]. Ces méthodes ont montré une maturité incontestable sur une large classe de problèmes. Cependant, comme toute méthode itérative, elles sont sensibles au conditionnement du système à résoudre, ce qui a freiné leur adoption dans des codes de calcul industriels et même leur application dans ce même contexte.

La résolution classique d'un problème éléments finis par une approche de décomposition de domaines peut se résumer de la manière suivante (branche supérieure de la figure 2) :

1. génération d'un maillage global de la structure, potentiellement de grande taille, obtenu à l'aide d'un mailleur séquentiel ;
2. décomposition en sous-structures à l'aide d'un logiciel de partitionnement de graphe de connectivités associé (Metis [10], Scotch [13], Chaco [8]...).

Dans le cas de problèmes hétérogènes, fréquemment rencontrés dans un cadre industriel, des méthodes efficaces de préconditionnement ont été développées au début des années 2000 [14, 9], permettant d'obtenir de bonnes propriétés de convergence dès lors que les hétérogénéités sont localisées au sein des sous-domaines. Cependant, l'utilisation d'un logiciel de partitionnement de graphe automatique afin d'équilibrer la charge et de réduire les communications entre les machines conduit bien souvent à l'obtention de sous-domaines présentant des formes élançées ainsi que des interfaces irrégulières (*cf.* figure 1) et mal positionnées vis-à-vis des hétérogénéités.

De récents développements, comme les algorithmes de la famille de AMPFETI³ fondées sur le multi-préconditionnement [2, 7], ou des problèmes aux valeurs propres généralisées résolus préalablement (GenEO⁴ [16]) ont permis à ces solveurs de gagner en robustesse sur des cas très hétérogènes et mal conditionnés mais leur consommation en terme de mémoire se révèle être encore très importante.

S'il paraît réalisable de produire des sous-domaines homogènes en optimisant le placement des interfaces par rapport aux hétérogénéités par ajout de pondérations sur les arêtes du graphe, il paraît en

1. *Finite Elements Tearing and Interconnecting*
2. *Balanced Domain Decomposition*
3. *Adaptive Multipreconditioned-FETI*
4. *Generalized Eigenproblems in the Overlaps*

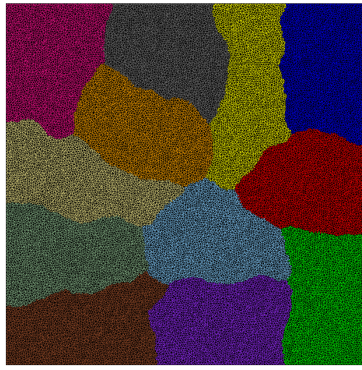


FIGURE 1 – Partitionnement d’un maillage 2D issu du logiciel de partitionnement de graphe Metis [10].

revanche impossible de garantir un aspect géométrique régulier de ces interfaces car le maillage n’est pas structuré et les logiciel de partitionnement de graphe ne prennent pas en compte la géométrie. D’autre part, cette approche traditionnelle de résolution présente une autre faiblesse en ce sens que la phase de génération du maillage et des sous-domaines est largement dominée par des étapes séquentielles qui représentent une portion significative du temps de calcul total. En particulier, la génération d’un maillage conforme de la mésostructure d’un composite tissé 3D à partir de données de géométrie est un processus qui peut devenir très coûteux et qui exploiterait avantageusement une parallélisation.

L’objectif de la méthode ici présentée est de contourner l’ensemble des difficultés précédentes en obtenant de manière automatique, et de manière la plus parallèle possible, des décompositions présentant des interfaces régulières et éventuellement adaptées aux hétérogénéités structurales permettant de réduire le conditionnement sans recourir aux techniques avancées mentionnées précédemment. Pour ce faire, cette nouvelle méthode, qui apparaît comme complètement originale pour la simulation en mécanique des solides, repose schématiquement sur une inversion des étapes de maillage et de sous-structuration associée à une discrétisation possiblement hiérarchique.

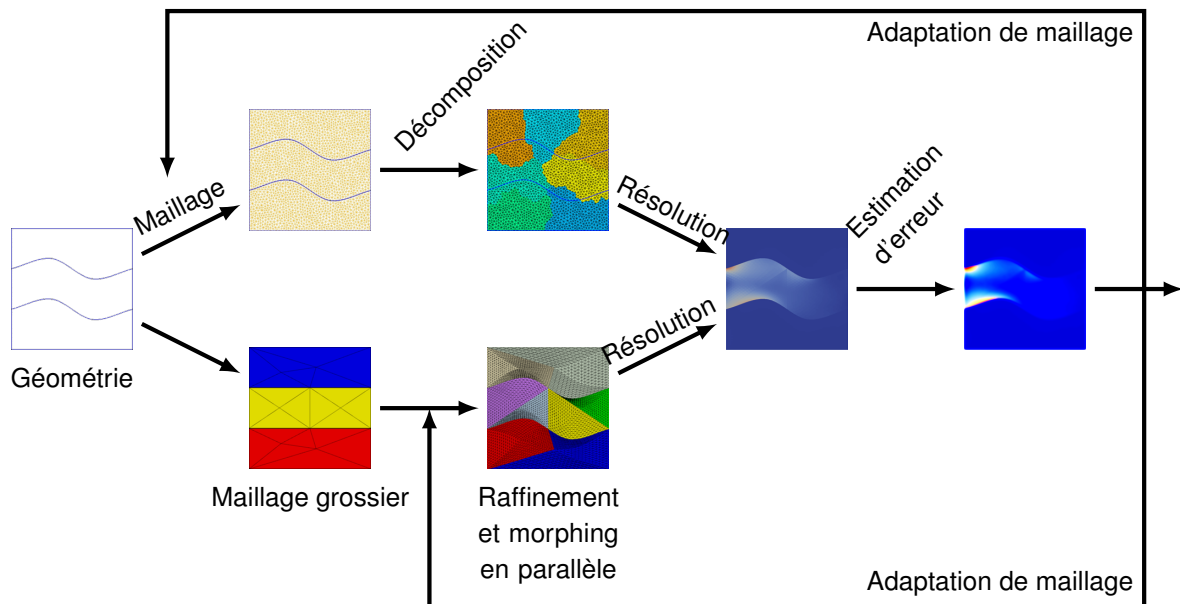


FIGURE 2 – Démarche de calcul en décomposition de domaine : - Branche supérieure : approche classique - Branche inférieure : approche hiérarchique développée.

Comme décrit sur la branche inférieure de la figure 2, à partir d’une géométrie de la structure (p.ex. CAO/CAO discrète), un premier maillage grossier est généré, sans besoin de représenter finement la géométrie initiale, afin d’en définir un partitionnement bien proportionné. Ensuite, les sous-domaines, définis comme éléments ou union d’éléments grossiers, sont distribués sur les processeurs de calcul avec l’information de la CAO sous-jacente de chaque sous-domaine. Le maillage fin de la structure est alors réalisé en parallèle sur chaque sous-domaine tout en conservant la compatibilité aux interfaces.

L'application de techniques de déformation de maillage permet enfin de s'assurer que ce maillage fin respecte au mieux la géométrie sous-jacente.

Le développement d'un prototype de cette méthode est réalisé à l'aide de l'interface Python du logiciel éléments finis Z-set [1, 5], dans lequel l'algorithme de décomposition de domaine FETI est implémenté. Une première partie s'attachera à rappeler les fondements de l'algorithme FETI utilisé pour la résolution des problèmes sous-structurés obtenus à l'aide de la méthode décrite dans une seconde partie.

2 Rappels sur la méthode FETI

On rappelle ici les bases de la méthode FETI utilisée dans le cadre de ce travail et décrite plus en détail dans [4, 6]. La discrétisation d'un problème de mécanique linéaire sur un domaine Ω peut se mettre sous la forme du système symétrique défini positif $Ku = f$. On suppose que le domaine peut être partitionné en N sous-domaines conformes $\Omega^{(s)}$ et que le problème associé peut s'écrire : $\forall s \in \llbracket 1; N \rrbracket$,

$$\begin{aligned} K^{(s)}u^{(s)} &= f^{(s)} + t^{(s)T}B^{(s)T}\lambda, \\ \sum_s B^{(s)}t^{(s)}u^{(s)} &= 0, \end{aligned} \quad (1)$$

où $t^{(s)}$ sont les opérateurs de trace, $B^{(s)}$ les opérateurs d'assemblage booléen et λ le champ d'inter-efforts entre les sous-domaines.

On définit classiquement le complément de Schur primal local $S_p^{(s)} = K_{bb}^{(s)} - K_{bi}^{(s)}K_{ii}^{(s)-1}K_{ib}^{(s)}$ (i représente les degrés de liberté internes et b les degrés de liberté de bord), le complément de Schur dual local $S_d^{(s)} = t^{(s)}K^{(s)+}t^{(s)T}$ ($K^{(s)+}$ désigne la pseudo-inverse de $K^{(s)}$) et la base des mouvements de corps rigides $R^{(s)} = \text{Ker}(K^{(s)})$. On peut aussi noter la relation entre les compléments de Schur locaux : $S_d^{(s)} = S_p^{(s)+}$. Avec ces notations, on peut écrire :

$$\begin{aligned} e &= \left[\dots \quad f^{(s)T}R^{(s)} \quad \dots \right]^T, & G &= \left[\dots \quad B^{(s)}t^{(s)}R^{(s)} \quad \dots \right]^T, \\ S_d &= \sum_s B^{(s)}S_d^{(s)}B^{(s)T}, & b_d &= \sum_s B^{(s)}t^{(s)}K^{(s)+}f^{(s)}. \end{aligned}$$

Le système (1) conduit à la formulation duale :

$$\begin{bmatrix} S_d & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} -b_d \\ -e \end{bmatrix}. \quad (2)$$

La contrainte $G^T\lambda = -e$ est imposée à l'aide d'une solution initiale λ_0 respectant cette condition ($G^T\lambda_0 = -e$) et d'un projecteur P sur l'espace orthogonal à G ($G^TP = 0$). L'inconnue λ est donc recherchée comme $\lambda = \lambda_0 + P\tilde{\lambda}$ avec :

$$\begin{aligned} \lambda_0 &= -QG(G^TQG)^{-1}e, \\ P &= I - QG(G^TQG)^{-1}G^T, \end{aligned} \quad (3)$$

et $\tilde{\lambda}$ est la solution du système :

$$P^T S_d P \tilde{\lambda} = P^T (-b_d - S_d \lambda_0). \quad (4)$$

La matrice Q intervenant dans (3) est symétrique définie positive et peut être choisie comme le préconditionneur \tilde{S}_p , l'identité ou une matrice de pondération [15].

Le système (4), qui est équivalent au système (2), est résolu par un solveur itératif de type gradient conjugué préconditionné dont le préconditionneur \tilde{S}_p est un assemblage de quantités primales locales telles que :

$$\tilde{S}_p = \sum_s \tilde{B}^{(s)} \tilde{S}_p^{(s)} \tilde{B}^{(s)T},$$

5. Cette contrainte est aussi appelée problème grossier : elle permet à des sous-domaines non voisins d'interagir sans la nécessité de transmettre des informations à travers d'autres sous-domaines.

avec $\tilde{B}^{(s)}$ des opérateurs d’assemblage pondérés tels que $\sum_s B^{(s)} \tilde{B}^{(s)T} = I$ et où $\tilde{S}_p^{(s)}$ peuvent être les compléments de Schur primaux locaux ou des approximations de ceux-ci ($K_{bb}^{(s)}$ ou $\text{diag}(K_{bb}^{(s)})$). La pondération choisie dans $\tilde{B}^{(s)}$ peut être fondée sur la multiplicité des degrés de liberté de l’interface lorsque le problème est homogène ou sur les coefficients diagonaux de $K_{bb}^{(s)}$ lorsqu’il est hétérogène.

La méthode FETI équipée de son problème grossier et du préconditionneur optimal de la littérature possède une extensibilité théorique car le conditionnement κ , qui est directement lié à la vitesse de convergence, dépend faiblement du nombre N de sous-domaines du problème :

$$\kappa \simeq C \left(1 + \log \left(\frac{H}{h} \right) \right)^2,$$

où H est la taille caractéristique des sous-domaines, h celle des éléments du maillage et C une constante qui dépend uniquement de la forme et de l’homogénéité des sous-domaines.

3 Démarche mise en œuvre

Dans cette partie, la méthode pour obtenir un maillage sous-structuré adapté pour les méthodes de décomposition de domaine sans recouvrement est décrite. L’objectif est ici de former une décomposition possédant les propriétés suivantes :

- l’interface entre les sous-domaines doit être la plus régulière possible (elle doit suivre la courbure de l’interface entre les matériaux ou être constituée de grands morceaux de plans) ;
- chaque sous-domaine doit être composé d’un seul matériau (cette condition pouvant être relâchée si la géométrie est trop complexe).

À partir d’une géométrie (CAO/CAO discrète), on choisit d’en extraire un maillage fin de peau (maillage coque qui pourrait s’apparenter à une description similaire à celle du format STL ⁶) représentant finement la géométrie et un maillage volumique grossier. Ce maillage grossier permet l’obtention d’une sous-structuration dont l’interface est constituée de plans de grande dimension. Les étapes pour parvenir à obtenir une décomposition hiérarchique sont les suivantes :

1. génération du maillage grossier représentant les entités physiques ;
2. raffinement des sous-domaines en parallèle en gardant la conformité à l’interface ;
3. déformation des sous-domaines en parallèle pour s’adapter à la géométrie de la structure.

3.1 Décomposition grossière homogène

À partir du maillage grossier, les ensembles physiques d’éléments sont extraits dans des maillages indépendants. Une fois ces ensembles isolés, un logiciel de partitionnement de graphe est utilisé sur chacun d’eux. À cette étape, chaque domaine est constitué d’un assemblage d’éléments grossiers.

À partir d’ici, chaque sous-domaine peut être traité de façon indépendante en parallèle. Comme les maillages obtenus sont trop grossiers pour effectuer un calcul éléments finis acceptable, il est nécessaire de générer un maillage plus fin des sous-domaines. Dans un contexte parallèle, la génération d’un maillage ne constitue pas une tâche aisée. En effet, il faut veiller à garder la conformité du maillage de part et d’autre de l’interface. La stratégie de remaillage hiérarchique qui a été mise en place ici permet de pouvoir garantir *a priori* cette conformité en gardant un remaillage parallèle sans échange entre les sous-domaines. Pour ce faire, un h-raffinement hiérarchique (consistant à diviser les éléments existants pour en créer de nouveaux) est réalisé dans chaque sous-domaine en parallèle de sorte de préserver automatiquement la conformité à l’interface. Par construction, les nouveaux éléments construits sont semblables à l’élément initial.

3.2 Déformation des sous-domaines

À ce stade, les maillages obtenus ne respectent pas la géométrie sous-jacente. En effet, le maillage des surfaces des entités physiques ne sont pas localisés sur les entités (ou surfaces physiques), qu’ils sont sensés représenter. On effectue alors une phase de déformation de maillage en deux étapes comme suit :

6. *Stereo-Lithography* : ce format décrit un objet par sa surface externe facettisée en éléments triangulaires

1. projection des surfaces physiques sur le maillage fin de la peau ;
2. déformation du volume intérieur en utilisant la projection précédente en tant que « conditions limites ».

La surface externe du sous-domaine est projetée sur la géométrie fine à l'aide de l'algorithme d'intersection rayon-triangle de Möller-Trumbore [12] qui consiste à trouver, pour tous les nœuds de la peau du sous-domaine, le triangle du maillage fin qui intersecte la droite passant par ce nœud et de direction donnée. Un traitement différent est appliqué aux arêtes physiques (celles qui représentent de vraies arêtes de la géométrie). Il est fondé sur une correspondance d'abscisses curviligne entre les arêtes.

La déformation du maillage est réalisée à l'aide d'une technique d'interpolation fondée sur les fonctions de base radiales (méthode RBF décrite par [3]). Chaque composante du champ de déplacement u_i appliqué aux nœuds du maillage à la position \underline{X} peut-être exprimée indépendamment des autres en fonction de la position \underline{X}_{b_j} des n_b nœuds de bord par la relation :

$$u_i(\underline{X}) = \sum_{j=1}^{n_b} \alpha_j \phi \left(\left\| \underline{X} - \underline{X}_{b_j} \right\| \right) + p(\underline{X}), \forall i \in \llbracket 1; d \rrbracket, \quad (5)$$

avec p un polynôme ayant comme coefficients les $(\beta_i)_{0 \leq i \leq \delta(p)}$, d la dimension de l'espace et ϕ une base de fonctions. Le polynôme p est un polynôme d'ordre 1 qui permet de décrire exactement les mouvements de corps rigides.

En imposant la condition d'interpolation $u_i(\underline{X}_{b_j}) = u_{b_j} \cdot x_i, \forall b_j \in \llbracket 1, n_b \rrbracket$, on arrive au système linéaire, valable pour chaque direction de l'espace :

$$\begin{bmatrix} M & P_b \\ P_b^T & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} u_b \\ 0 \end{bmatrix}, \quad (6)$$

avec M une matrice symétrique définie positive⁷ de taille $n_b \times n_b$ définie par $M_{ij} = \phi \left(\left\| \underline{X}_{b_i} - \underline{X}_{b_j} \right\| \right)$ et P_b une matrice de taille $n_b \times 4$ telle que $P_{b_i} = [1 \quad X_{b_i}]$. X_b étant une matrice de taille $n_b \times d$ contenant la liste des coordonnées des nœuds de bord dont le déplacement est connu. La fonction ϕ est la fonction de base radiale telle que donnée dans [3]. Une fois le problème (6) résolu, il suffit de construire, pour tous les nœuds intérieurs et pour chaque direction, le déplacement comme décrit dans l'équation (5). La transformation permet de modifier le maillage d'un sous-domaine (figure 3a) en un maillage déformé ayant les nœuds intérieurs répartis dans le volume comme illustré sur la figure 3b.

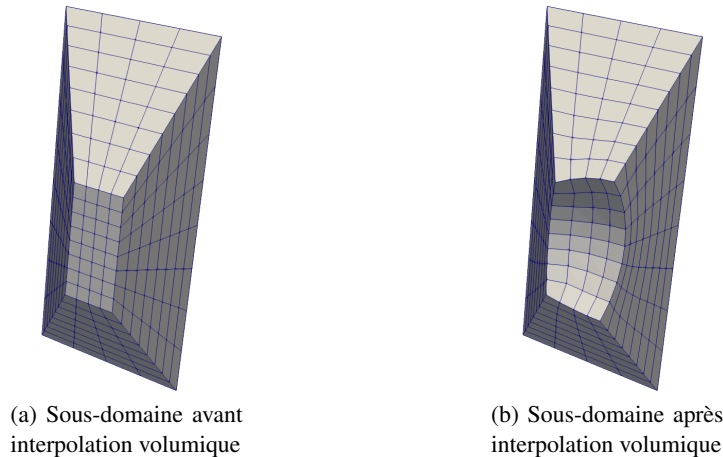


FIGURE 3 – Déformation des nœuds intérieurs d'un sous-domaine (vue en coupe)

Une fois la décomposition obtenue, la méthode FETI peut être appliquée pour résoudre le problème sous-structuré sur le maillage partitionné créé.

7. à condition que ϕ soit définie positive

4 Résultats numériques

Le potentiel de la méthode est étudié sur le cas test académique d'une structure cubique de côté $H = 10$ (cf. figure 4c) munie d'un comportement linéaire élastique isotrope, constitué d'une inclusion sphérique rigide de module d'élasticité $E_1 = 200$ GPa et de rayon $H/4$ entourée d'une matrice de module E_2 tel que $E_1/E_2 = 10^5$, et soumise à un chargement de type traction/cisaillement à l'aide de conditions aux limites de Dirichlet. Le maillage grossier initial est constitué de 13 éléments hexahédriques linéaires (six dans la matrice et sept dans l'inclusion).

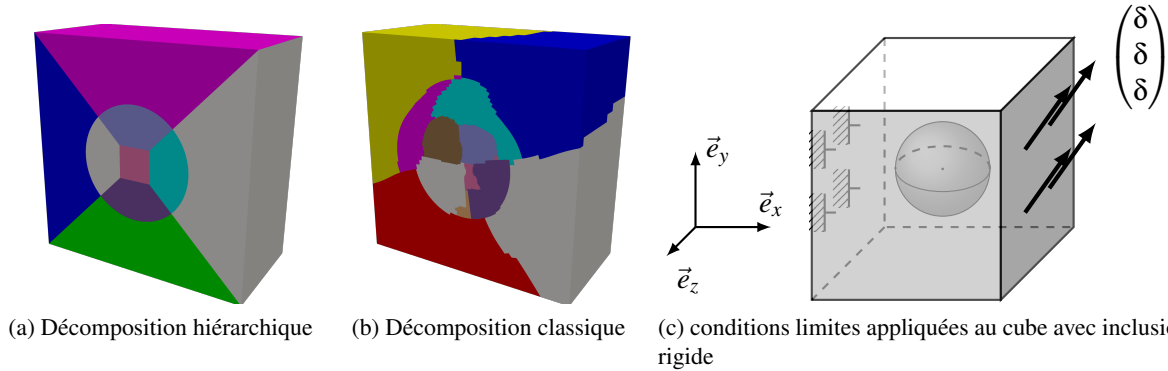


FIGURE 4 – Exemple de cube avec inclusion et vue en coupe des décompositions à 13 sous-domaines et cinq raffinements hiérarchiques

Le problème associé est résolu sur la base d'une décomposition issue de la méthode hiérarchique présentée précédemment (cf. figure 4a), à l'issue de cinq raffinements hiérarchiques successifs et en présence de 13 sous-domaines avec un total de 1.2 million de degrés de liberté. Ce calcul est ensuite comparé à celui associé à une décomposition automatique par le logiciel Scotch du maillage obtenu précédemment (cf. figure 4b). Le fait de récupérer le maillage issu de la décomposition hiérarchique permet d'avoir un problème global possédant exactement le même nombre de degrés de liberté et des maillages de même caractéristiques. Cela permet d'observer uniquement l'influence de la décomposition sur la convergence.

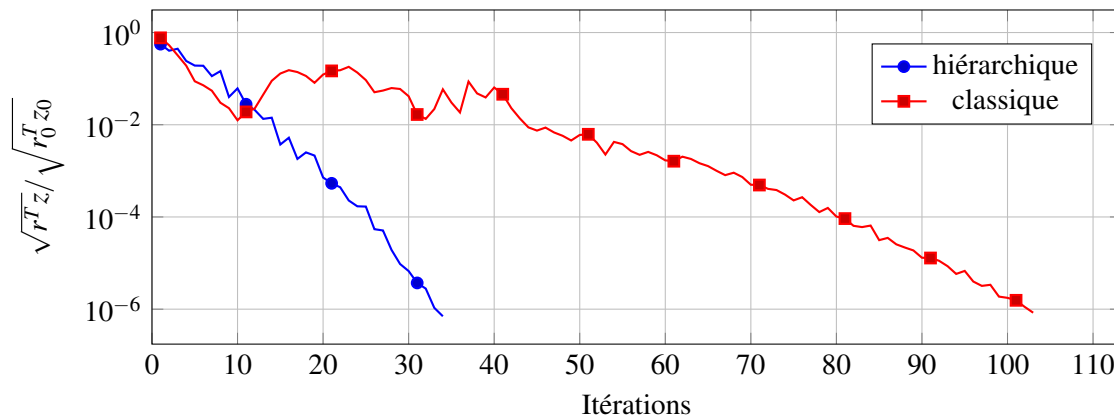


FIGURE 5 – Courbe de convergence pour le résidu préconditionné $\sqrt{r^T z} / \sqrt{r_0^T z_0}$ pour les décompositions hiérarchique et classique

Le critère de convergence étant fixé à 10^{-6} pour le rapport entre le résidu préconditionné et le résidu préconditionné initial, la courbe de convergence est représentée sur la figure 5. Le solveur converge en seulement 34 itérations pour la décomposition hiérarchique contre 103 pour la décomposition classique, soit un gain de 67 % sur le cas considéré. De plus, le temps CPU pour atteindre ce critère est de 29.2 s pour l'approche décrite ici contre 96.6 s pour l'approche classique avec des temps CPU moyens par itération semblables (0.86 s et 0.94 s respectivement).

D'autres problèmes ont pu être testés sur le même cas en faisant varier le nombre de raffinements de 1 à 5 et le nombre de sous-domaines de 2 à 13 et de manière générale, la majorité des calculs issus de la décomposition hiérarchique permettent des gains d'itérations compris entre 20 % et 60 % par rapport aux calculs issus d'une décomposition classique. De plus, la décomposition hiérarchique conduit à une convergence moins dépendante à la taille du maillage.

5 Conclusion

Une approche originale de sous-structuration hiérarchique permettant d'obtenir des interfaces les plus régulières possible et de création des sous-domaines homogènes a été introduite et implémentée à l'aide du langage Python dans le cadre de la plateforme éléments finis Z-set. Les résultats semblent prometteurs même si la méthode n'a encore été appliquée qu'à une géométrie simple.

Par la suite, les propriétés de la décomposition obtenue, en particulier la régularité des interfaces, seront exploitées pour définir des stratégies de résolution optimisées.

Références

- [1] J. Besson and R. Foerch. Large scale object-oriented finite element code design. *Computer Methods in Applied Mechanics and Engineering*, 142(1-2) :165–187, March 1997.
- [2] Christophe Bovet, Augustin Parret-Fréaud, Nicole Spillane, and Pierre Gosselet. Adaptive multipreconditioned FETI : Scalability results and robustness assessment. *Computers & Structures*, 193 :1–20, December 2017.
- [3] A. de Boer, M.S. van der Schoot, and H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers & Structures*, 85(11-14) :784–795, June 2007.
- [4] Charbel Farhat and Francois-Xavier Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6) :1205–1227, October 1991.
- [5] Feyel, Frédéric. Some new technics regarding the parallelisation of ZéBuLoN, an object oriented finite element code for structural mechanics. *ESAIM : M2AN*, 36(5) :923–935, 2002.
- [6] Pierre Gosselet and Christian Rey. Non-overlapping domain decomposition methods in structural mechanics. *Archives of Computational Methods in Engineering*, 13(4) :515–572, December 2006.
- [7] 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, May 2015.
- [8] Bruce Hendrickson. Chaco. In David Padua, editor, *Encyclopedia of Parallel Computing*, pages 248–249. Springer US, Boston, MA, 2011.
- [9] Axel Klawonn and Olof Widlund. FETI and Neumann-Neumann iterative substructuring methods : Connections and new results. *Communications on Pure and Applied Mathematics*, 54(1) :57–90, January 2001.
- [10] Dominique LaSalle, Md Mostofa Ali Patwary, Nadathur Satish, Narayanan Sundaram, Pradeep Dubey, and George Karypis. Improving graph partitioning for modern graphs and architectures. pages 1–4. ACM Press, 2015.
- [11] Jan Mandel. Balancing domain decomposition. *Communications in Numerical Methods in Engineering*, 9(3) :233–241, March 1993.
- [12] Tomas Möller and Ben Trumbore. Fast, Minimum Storage Ray-Triangle Intersection. *Journal of Graphics Tools*, 2(1) :21–28, January 1997.
- [13] François Pellegrini and Jean Roman. Scotch : A software package for static mapping by dual recursive bipartitioning of process and architecture graphs. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Heather Liddell, Adrian Colbrook, Bob Hertzberger, and Peter Sloot, editors, *High-Performance Computing and Networking*, volume 1067, pages 493–498. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996.
- [14] Daniel J. Rixen and Charbel Farhat. A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *International Journal for Numerical Methods in Engineering*, 44(4) :489–516, February 1999.
- [15] Daniel J. Rixen, Charbel Farhat, Radek Tezaur, and Jan Mandel. Theoretical comparison of the FETI and algebraically partitioned FETI methods, and performance comparisons with a direct sparse solver. *International Journal for Numerical Methods in Engineering*, 46(4) :501–533, October 1999.

- [16] Nicole Spillane and Daniel J. Rixen. Automatic spectral coarse spaces for robust FETI and BDD algorithms. November 2012.