Adaptive multipreconditioned FETI: scalability results and robustness assessment

Christophe Bovet, Augustin Parret-Fréaud, Nicole Spillane, Pierre Gosselet

Adaptive multipreconditioned FETI: scalability results and robustness assessment

Christophe Bovet[a,b], Augustin Parret-Fréaud[c], Nicole Spillane[d], Pierre Gosselet[a]

[a] *LMT, ENS Cachan, CNRS, Université Paris-Saclay, 61 avenue du Président Wilson, 94235 Cachan, France*
[b] *Onera – The French Aerospace Lab, F-92322 Châtillon, France*
[c] *Safran Tech, Rue des Jeunes Bois, Châteaufort, CS 80112 78772 Magny-Les-Hameaux, France*
[d] *CMAP, École polytechnique, CNRS, Université Paris-Saclay, 91128, Palaiseau, France.*

Abstract

The purpose of this article is to assess the adaptive multipreconditioned FETI solvers (AMPFETI) on realistic industrial problems and hardware. The multipreconditioned FETI algorithm (first introduced as Simultaneous FETI [1]) is a non-overlapping domain decomposition method which exhibits good robustness properties without requiring the explicit knowledge of the original partial differential equation, or any *a priori* analysis of the algebraic system through eigenvalues problems. Multipreconditioned FETI solves critical problems in significantly fewer iterations than classical FETI but each iteration involves a larger computational effort. An adaptive strategy (known as the adaptive multipreconditioned conjugate gradient algorithm [2]) has been proposed to achieve balance between robustness and efficiency and we will observe that it provides an efficient solver for the problems considered here.

Keywords: Domain decomposition, FETI, Krylov subspace methods, multipreconditioning, adaptive multipreconditioning

1. Introduction

Domain decomposition methods provide a favorable framework to solve in parallel the linear systems resulting from the discretization of partial differential equations. Indeed the algorithms which can be derived, and in particular the Schur complement family of methods, namely Balancing Domain Decomposition (BDD) [3], Finite Element Tearing and Interconnecting (FETI) [4], and their constrained variants [5, 6], involve exchanges between neighbors and only limited all-to-all communications. Moreover, the sparse structure of the coarse

*Corresponding author
URL: christophe.bovet@onera.fr (Christophe Bovet)

problems introduced to ensure the scalability allows for multilevel large scale implementations [7, 8].

These methods suffer from a loss of performance in certain cases (typically jagged interfaces, bad aspect ratios, strong heterogeneities misplaced with respect to the interface, incompressibility) which makes them ineffective to solve a large class of systems encountered by mechanical engineers. Many fixes were proposed to correct these flaws but they often required the knowledge of the original PDE [9, 10], and they were thus not generic enough to be implemented in commercial engineering software.

It has been understood that the drop of performance occurred when the classical preconditioner - a weighted sum of the inverses of subdomain matrices - was not able to correctly take into account certain global features [11, 12]. New fixes were proposed based on an a priori analysis of the algebraic system without reference to the PDE. The prior analysis takes the form of localized generalized eigenvalues problems with minimal communication. In the GenEO methods [13, 14], the part of the spectrum of the preconditioned operator which causes bad conditioning, is detected and removed from the resolution by augmenting the Krylov solver; the same objective can be achieved by adding well chosen primal constraints [15, 16]. In the deluxe scaling methods, the partition of unity between subdomains which is involved in the preconditioner is optimized in order to avoid the dilution of the problematic local information [17].

Because of the strong connection between eigenvalue analysis and Krylov solvers, it was tempting to skip the prior analysis and try to detect the "bad modes" on the fly. This idea led to the Simultaneous-FETI algorithm (SFETI) [1], which turned out to be a particular case of the multipreconditioned Krylov solvers [18, 19, 20] with an optimized implementation. For clarity, in this article we will refer to the SFETI algorithm as the multipreconditioned FETI (MPFETI) algorithm. In MPFETI the additivity of the preconditioner is exploited in order to generate as many search directions as there are subdomains: the local information is preserved, and an optimal combination (scaling) is computed at each iteration. The drawback of the method is the large quantity of information generated at each iteration and the storage requirements caused by the loss of the short recurrence property. In order to limit the amount of numerical data, a selection criterion was proposed in [2] where the vectors whose contributions need not be individualized are summed up (as in the classical FETI algorithm). This is the adaptive multipreconditioned conjugate gradient algorithm (AMPCG) which gives rise to the adaptive MPFETI, or AMPFETI, algorithm.

The aim of this paper is to present a validation of the MPFETI and AMPFETI algorithms on large scale computations and realistic problems coming from the industry, computed on "standard" clusters (a few thousands cores). MPFETI requires the factorization of a dense $N \times N$ matrix at each iteration (where N is the number of subdomains) so in its present form we do not expect it to scale up to hundreds of thousands of subdomains as do some of the previously cited methods. We do anticipate that this bottleneck could be avoided with further adaptive or multilevel methods, for instance by aggregating subdomains

into clusters before multipreconditiong as proposed in [21] for the RAS domain decomposition method.

The article is organized as follows: Section 2 gives a short presentation of Simultaneous FETI and its adaptive versions; then, academic validation and scalability results are provided in Section 3; Section 4 presents a realistic industrial simulation; finally, Section 5 concludes the paper.

2. Adaptive Multipreconditioned FETI: the algorithm

The algorithm that we study in this article is a combination of two things. First, the FETI domain decomposition method is applied to rewrite the original linear system as a linear system posed on the interfaces between subdomains that posesses an additive structure. Then the adaptive multipreconditioned conjugate gradient (AMPCG) is used as a linear solver. Although this algorithm is not limited to the context of structural mechanics, we restrain to this framework for simplicity.

2.1. The FETI linear system

Let us consider the linear system of equations of the form $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}$ arising from some finite element discretization of a linear mechanical problem set on a domain denoted by Ω. Vectors $\boldsymbol{u}$ and $\boldsymbol{f}$ are respectively the vector of displacements and the vector of generalized forces (as well as the forces this may include terms coming, for example, from the elimination of non-homogeneous Dirichlet boundary conditions). We assume that the stiffness matrix $\boldsymbol{K}$ is symmetric positive definite. In the FETI method [22], the global problem is not assembled. Instead, the computational domain Ω is partitioned into N non-overlapping subdomains Ω^s, $s = 1, \ldots, N$, and only local quantities are assembled: the matrix $\boldsymbol{K}^s$ and vector of generalized forces $\boldsymbol{f}^s$ of the same problem once restricted to each subdomain Ω^s. This way the problem can be reformulated as

$$\begin{aligned} \boldsymbol{K}^s\boldsymbol{u}^s &= \boldsymbol{f}^s + {\boldsymbol{t}^s}^\top {\boldsymbol{B}^s}^\top \boldsymbol{\lambda} \text{ for each } s = 1, \ldots, N, \\ \sum_s \boldsymbol{B}^s\boldsymbol{t}^s\boldsymbol{u}^s &= \boldsymbol{0} \end{aligned} \tag{1}$$

where $\boldsymbol{t}^s : \Omega^s \to \partial\Omega^s$ are trace operators and $\boldsymbol{B}^s : \partial\Omega^s \to \oplus_{1\leq i\neq j\leq N}(\partial\Omega^i \cap \partial\Omega^j)$ are signed Boolean assembly operators. The first equation ensures the equilibrium of subdomain Ω^s and the second is a constraint that ensures the continuity of the displacements on the interface. This is known as the compatibility condition and enforced by means of an extra unknown: the Lagrange multiplier $\boldsymbol{\lambda}$.

Next, the problem is restricted to the set of interfaces between subdomains. To this end, we introduce the local Schur complements :

$$\boldsymbol{S}^s = \boldsymbol{K}^s_{\Gamma\Gamma} - \boldsymbol{K}^s_{\Gamma I}{\boldsymbol{K}^s_{II}}^{-1}\boldsymbol{K}^s_{I\Gamma}; \qquad \begin{array}{l} I : \text{internal degrees of freedom} \\ \Gamma : \text{boundary degrees of freedom,} \end{array}$$

and their (pseudo) inverses ${\boldsymbol{S}^s}^\dagger$ that satisfy the identity: ${\boldsymbol{S}^s}^\dagger = \boldsymbol{t}^s{\boldsymbol{K}^s}^\dagger{\boldsymbol{t}^s}^\top$ (the reason pseudo-inversion $\cdot^\dagger$ is needed is to account for all cases where the kernel

of $\boldsymbol{K}^s$ is not restricted to zero). Let $\boldsymbol{R}^s$ be a matrix whose columns form a basis of $\mathrm{Ker}(\boldsymbol{K}^s)$ (in this case of the local rigid body modes) and define

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

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

The displacement unknown is then eliminated out of (1) to yield the FETI system: find $(\boldsymbol{\lambda}, \boldsymbol{\beta})$ such that

$$\begin{pmatrix} \boldsymbol{F} & \boldsymbol{G} \\ \boldsymbol{G}^\top & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \boldsymbol{d} \\ \boldsymbol{e} \end{pmatrix}. \tag{2}$$

The second equation, $\boldsymbol{G}^\top\boldsymbol{\lambda} = \boldsymbol{e}$, corresponds to the constraint that each subdomain must remain self-equilibrated. The connection with the original formulation is that $\boldsymbol{u}^s = \boldsymbol{K}^{s\dagger}(\boldsymbol{f}^s + \boldsymbol{t}^{s\top}\boldsymbol{B}^{s\top}\boldsymbol{\lambda}) + \boldsymbol{R}^s\boldsymbol{\beta}^s$ ($\boldsymbol{\beta}^s$ being the part of $\boldsymbol{\beta}$ corresponding to subdomain Ω^s). Instead of solving a saddle point system, an initialization–projection strategy is applied and $\boldsymbol{\lambda}$ is sought for as

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \boldsymbol{\Pi}\tilde{\boldsymbol{\lambda}}; \quad \boldsymbol{\lambda}_0 = \boldsymbol{A}\boldsymbol{G}(\boldsymbol{G}^\top\boldsymbol{A}\boldsymbol{G})^{-1}\boldsymbol{e}; \quad \boldsymbol{\Pi} = \boldsymbol{I} - \boldsymbol{A}\boldsymbol{G}(\boldsymbol{G}^\top\boldsymbol{A}\boldsymbol{G})^{-1}\boldsymbol{G}^\top$$

for a given symmetric positive definite matrix $\boldsymbol{A}$, most often chosen among the classical preconditioners $\tilde{\boldsymbol{S}}$ (see (4)) and the identity matrix $\boldsymbol{I}$ [23]. Note that $\boldsymbol{\Pi}$ is a projector that satisfies $\mathrm{Ker}(\boldsymbol{\Pi}^\top) = \mathrm{range}(\boldsymbol{G})$. Substituting this into (2), and pre-multiplying by $\boldsymbol{\Pi}^\top$ leads to the final linear system

$$\boldsymbol{\Pi}^\top\boldsymbol{F}\boldsymbol{\Pi}\tilde{\boldsymbol{\lambda}} = \boldsymbol{\Pi}^\top(\boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}_0), \tag{3}$$

that is solved by the conjugate gradient iterative solver. The preconditioner takes the form of a scaled sum of Schur complements (or approximations thereof):

$$\tilde{\boldsymbol{S}} = \sum_s \tilde{\boldsymbol{B}}^s \tilde{\boldsymbol{S}}^s \tilde{\boldsymbol{B}}^{s\top} \text{ with } \left| \begin{array}{l} \text{either } \tilde{\boldsymbol{S}}^s = \boldsymbol{S}^s : \text{ Dirichlet (or full) preconditioner,} \\ \text{or } \tilde{\boldsymbol{S}}^s = \boldsymbol{K}^s_{\Gamma\Gamma} : \text{ lumped preconditioner,} \\ \text{or } \tilde{\boldsymbol{S}}^s = \mathrm{diag}(\boldsymbol{K}^s_{\Gamma\Gamma}) : \text{ superlumped preconditioner,} \end{array} \right. \tag{4}$$

where operators $\tilde{\boldsymbol{B}}^s$ are scaled assembly operators such that:

$$\left(\sum_s \boldsymbol{B}^s \tilde{\boldsymbol{B}}^{s\top}\right) \boldsymbol{B}^j = \boldsymbol{B}^j \tag{5}$$

Two common choices (that we use in our numerical experiments) are multiplicity scaling and stiffness scaling (often called $\boldsymbol{K}$-scaling) [24]: the scaling is proportional either to the number of subdomains that the degree of freedom belongs to or to the diagonal of the local stiffness matrix $\boldsymbol{K}^s_{\Gamma\Gamma}$ on the boundary.

2.2. Adaptive Multipreconditioned Conjugate Gradient

Unfortunately there are many cases in which the FETI algorithm does not converge fast or does not converge at all (due to the accumulation of numerical errors over very many iterations) even with state of the art choices of the scaling operators. This is the case in particular for some heterogeneous material distributions and partitions into subdomains that cut through the heterogeneities [11]. The Simultaneous FETI algorithm [1] was proposed to solve even these hard test cases. It is a multipreconditioned conjugate gradient algorithm (MPCG) [18] where each term in the definition of the preconditioner $\tilde{\boldsymbol{S}}$ is considered to be a separate preconditioner. This way the search space is significantly enlarged and it was observed that convergence is greatly improved. The drawback is that an iteration of MPCG is more expensive than an iteration of the usual PCG and the extra work may not always be necessary. Recently, an improvement of MPCG, the Adaptive MPCG algorithm (AMPCG) [2] was introduced that chooses, at each iteration, how many search directions are used (between 1 and N, N being again the number of subdomains). The FETI linear system is ideal for the AMPCG framework since the main assumption is that a bound for the smallest eigenvalue of the globally preconditioned operator is known. This is the case here: all non zero eigenvalues of $\tilde{\boldsymbol{S}}\boldsymbol{\Pi}^\top \boldsymbol{F}\boldsymbol{\Pi}$ are larger than 1 [25, 26, 27].

Algorithm 1 describes AMPCG applied to FETI (equation (3) preconditioned by $\tilde{\boldsymbol{S}}$) using the notation introduced in the previous section as well as the following: $\boldsymbol{\lambda}_{00}$ is an initial guess, $\mathbf{1} = (\underbrace{1, 1, \ldots, 1}_{N \text{ times}})^\top$, $\epsilon > 0$ is the tolerance on the preconditioned residual, $\tau > 0$ is a threshold chosen by the user (more on this below), and the local components in the operator and the preconditioner are respectively denoted by

$$\boldsymbol{F}^s = \boldsymbol{B}^s \boldsymbol{S}^{s\dagger} \boldsymbol{B}^{s\top}, \text{ and } \tilde{\boldsymbol{S}}^s = \tilde{\boldsymbol{B}}^s \tilde{\boldsymbol{S}}^s \tilde{\boldsymbol{B}}^{s\top}$$

Let m_i denote the number of search directions selected at iteration i. There are in fact three variants in Algorithm 1 that differ in the preconditioning step (between lines 13 and 19):

- No τ-test: $m_i = N$ for all i. Each local contribution to the preconditioner serves as a search direction. This is the Simultaneous FETI algorithm [1], it is a non adaptive MPCG solver [18].
- Global τ-test: $m_i = 1$ or N. At each iteration, either all contributions are saved (as above) or they are summed into a global search direction which is what is done in the usual PCG algorithm. The test $t_i < \tau$ is called the global τ-test.
- Local τ-test: $1 \leq m_i \leq N$. This time, in each subdomain, the local τ-test $t_i^s < \tau$ determines whether to keep a separate search direction or not. If not all directions are selected, the global direction is also used as a search direction.

Algorithm 1: Adaptive Multipreconditioned FETI. The user inputs the initial guess $\boldsymbol{\lambda}_{00}$, the tolerance $\epsilon > 0$ and the threshold $\tau > 0$ in the τ-test.

$\boldsymbol{\lambda}_0 = \boldsymbol{AG}(\boldsymbol{G}^\top \boldsymbol{AG})^{-1}\boldsymbol{e} + \boldsymbol{\Pi}\boldsymbol{\lambda}_{00}$

$\boldsymbol{r}_0 = \boldsymbol{\Pi}^\top(\boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}_0)$

$\boldsymbol{Z}_0 = \left(\tilde{\boldsymbol{S}}^1\boldsymbol{r}_0 | \dots | \tilde{\boldsymbol{S}}^N\boldsymbol{r}_0\right)$

$\boldsymbol{W}_0 = \boldsymbol{\Pi}\boldsymbol{Z}_0$

$i = 0$

while $\sqrt{\boldsymbol{r}_i^\top \boldsymbol{Z}_i \mathbf{1}} > \epsilon$ **do**

  $\boldsymbol{Q}_i = \boldsymbol{F}\boldsymbol{W}_i$

  $\boldsymbol{\Delta}_i = \boldsymbol{Q}_i^\top \boldsymbol{W}_i$

  $\boldsymbol{\gamma}_i = \boldsymbol{Z}_i^\top \boldsymbol{r}_i$

  $\boldsymbol{\alpha}_i = \boldsymbol{\Delta}_i^\dagger \gamma_i$

  $\boldsymbol{\lambda}_{i+1} = \boldsymbol{\lambda}_i + \boldsymbol{W}_i\boldsymbol{\alpha}_i$

  $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \boldsymbol{\Pi}^\top \boldsymbol{Q}_i\boldsymbol{\alpha}_i$

  if *No τ-test* **then**

    $\boldsymbol{Z}_{i+1} = (\tilde{\boldsymbol{S}}^1\boldsymbol{r}_{i+1} | \dots | \tilde{\boldsymbol{S}}^N\boldsymbol{r}_{i+1})$ `/* Multi-Precondition */`

  else if *Global τ-test* **then**

    $$\boldsymbol{Z}_{i+1} = \begin{cases} (\tilde{\boldsymbol{S}}^1\boldsymbol{r}_{i+1} | \dots | \tilde{\boldsymbol{S}}^N\boldsymbol{r}_{i+1}) & \text{if} \quad t_i := \frac{\boldsymbol{\gamma}_i^\top \boldsymbol{\alpha}_i}{\boldsymbol{r}_{i+1}^\top \boldsymbol{z}_{i+1}} < \tau, \\ \tilde{\boldsymbol{S}}\boldsymbol{r}_{i+1} & \text{otherwise.} \end{cases}$$

  else if *Local τ-test* **then**

    $$t_i^s = \frac{(\boldsymbol{W}_i\boldsymbol{\alpha}_i)^\top \boldsymbol{F}^s(\boldsymbol{W}_i\boldsymbol{\alpha}_i)}{\boldsymbol{r}_{i+1}^\top \tilde{\boldsymbol{S}}^s \boldsymbol{r}_{i+1}}$$

    $\boldsymbol{Z}_{i+1} = \text{concatenate}(\{\tilde{\boldsymbol{S}}^s\boldsymbol{r}_{i+1}\,;\, t_i^s < \tau\}, \sum_{t_i^s > \tau} \tilde{\boldsymbol{S}}^s\boldsymbol{r}_{i+1})$

  $\boldsymbol{W}_{i+1} = \boldsymbol{\Pi}\boldsymbol{Z}_{i+1}$

  for $0 \le j \le i$ **do** $\begin{cases} \boldsymbol{\Phi}_{i,j} = \boldsymbol{Q}_j^\top \boldsymbol{W}_{i+1} \\ \boldsymbol{W}_{i+1} \leftarrow \boldsymbol{W}_{i+1} - \boldsymbol{W}_j \boldsymbol{\Delta}_j^\dagger \boldsymbol{\Phi}_{i,j} \end{cases}$

  $i \leftarrow i + 1$

Return $\boldsymbol{\lambda}_i$

For clarity we next state the sizes of the quantities in Algorithm 1 (denoting by n the size of the linear system):

$$\boldsymbol{r}_i,\, \boldsymbol{\lambda}_i,\, \boldsymbol{\lambda}_{00} \in \mathbb{R}^n; \quad \boldsymbol{Z}_i,\, \boldsymbol{W}_i,\, \boldsymbol{Q}_i \in \mathbb{R}^{n\times m_i}; \quad \boldsymbol{\Delta}_i \in \mathbb{R}^{m_i\times m_i};$$

$$\boldsymbol{\alpha}_i,\, \boldsymbol{\gamma}_i \in \mathbb{R}^{m_i}; \quad \boldsymbol{\Phi}_{i,j} \in \mathbb{R}^{m_j\times m_i}; \quad t_i,\, t_i^s \in \mathbb{R}.$$

The effective preconditioner changes at each iteration of AMPCG which makes full recurrence mandatory (see line 21 in Algorithm 1) in order to obtain $\boldsymbol{F}$-conjugate blocs of search directions: $\boldsymbol{W}_i^\top \boldsymbol{F}\boldsymbol{W}_j = \boldsymbol{0}$ for $i \neq j$. The fact that it is a linear combination of the same components does, however, ensure that a global minimization property holds [18]: the approximate solution $\boldsymbol{\lambda}_i$ at iteration i minimizes the $\boldsymbol{F}$ norm of the error $\boldsymbol{\lambda}_i - \boldsymbol{\lambda}$ over the minimization space: $\boldsymbol{\lambda}_0 + \sum_{j=0}^{i-1} \boldsymbol{W}_j$. This minimization space has higher dimension than the minimization space of PCG and this is the reason why the convergence of MPCG is so good. Of course, the enlargement induces an extra cost and the purpose of the τ-tests is precisely to detect automatically which iterations require it. In order to get the best balance between number of iterations and cost of each iteration, the choice of τ is crucial. In [2][Corollary 3.3] it is proposed to choose $\tau = \frac{1-\rho^2}{\rho^2}$ if the user would be satisfied by a relative reduction of the $\boldsymbol{F}$-norm of the error by a factor ρ at each iteration[1].

2.3. Optimized implementation

In order for the algorithm to be efficient, several points need to be addressed during implementation.

Low rank corrections to avoid many Neumann solves. Perhaps the most important modification is to not naively apply $\boldsymbol{F}$ to $\boldsymbol{W}_i$ as stated in line 7. Instead it was proposed in [1] to pre-compute and store the (sparse) product $\boldsymbol{F}\boldsymbol{A}\boldsymbol{G}$ and replace line 7 by the following computation:

$$\boldsymbol{Q}_i = \boldsymbol{F}\boldsymbol{Z}_i - (\boldsymbol{F}\boldsymbol{A}\boldsymbol{G})(\boldsymbol{G}^\top\boldsymbol{A}\boldsymbol{G})^{-1}\boldsymbol{G}^\top\boldsymbol{Z}_i - \sum_{j=0}^{i-1} \boldsymbol{Q}_j\boldsymbol{\Delta}_j^\dagger\boldsymbol{\Phi}_{i,j}. \tag{6}$$

This way $\boldsymbol{F}$ is applied to $\boldsymbol{Z}_i$ instead of $\boldsymbol{W}_i$: each component $\boldsymbol{F}^s$ need only be applied to the columns in $\boldsymbol{Z}_i$ corresponding to Ω^s and its neighbours. This represents a huge gain because there is a local solve in $\boldsymbol{F}^s$. What's more, since memory access is a significant part of the cost of a matrix vector product, multiplying by a reasonably sized block of vectors instead of a single vector is not much more expensive. Also, the term $(\boldsymbol{G}^\top\boldsymbol{A}\boldsymbol{G})^{-1}\boldsymbol{G}^\top\boldsymbol{Z}_i$ in (6) is not computed since it was already evaluated during the projection in line 20. Thus the low rank correction does not imply any additional global communication.

[1] In [2], the smallest eigenvalue $\boldsymbol{\lambda}_{min}$ appears as a factor in the denominator but here we used the fact that $\boldsymbol{\lambda}_{min} = 1$.

Factorization of $\boldsymbol{\Delta}_i$. In the update and orthogonalization steps the pseudo inverse $\boldsymbol{\Delta}_i^\dagger$ of $\boldsymbol{\Delta}_i$ must be applied. Note that these applications are well defined [1]. The reason the pseudo inverse is needed is to take care of cases where some search directions are linearly redundant. Here, we follow what was proposed in the initial article [28]: perform a Cholesky factorization of $\boldsymbol{\Delta}_i$ and use it to replace the block of search directions $\boldsymbol{W}_i$ by an $\boldsymbol{F}$-orthonormal family that spans the same space. This way $\boldsymbol{\Delta}_i$ can be replaced by identity in all evaluations. The use of a Crout factorization, which further improves accuracy since no square root is performed, and is required in some contexts, is exposed in the next subsection.

2.4. Extension to finite displacement problems

The MPFETI algorithm can be used as the linear solver for the tangent systems arising from the Newton-Raphson linearization scheme. In the case of finite displacement problems, a classical observation is the loss of definiteness of the Neumann operator (somehow infinitesimal rotations evolve into negative energy modes, see [28] for another illustration). Anyhow, because only a few eigenvalues are negative, reorthogonalized conjugate gradient can still be employed without a significant drop of performance [29].

In the case of MPCG, this leads to possibly non-positive definite symmetric matrices $\boldsymbol{\Delta}_i$ rendering the Cholesky factorization irrelevant. We propose instead to use a rank revealing Crout factorization to compute the pseudo-inverse of $\boldsymbol{\Delta}_i$. In that case, we are not able to $\boldsymbol{F}$-orthonormalize the columns of the blocks $(\boldsymbol{W}_i)$:

$$\boldsymbol{\Delta}_i = \boldsymbol{M}\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^\top\boldsymbol{M} \quad ; \quad \text{with } \boldsymbol{L} = \begin{pmatrix} \tilde{\boldsymbol{L}} & 0 \\ \times & \boldsymbol{I} \end{pmatrix} \quad ; \quad \boldsymbol{D} = \begin{pmatrix} \tilde{\boldsymbol{D}} & 0 \\ 0 & 0 \end{pmatrix} \tag{7}$$

where $\tilde{\boldsymbol{L}}$ is a unit lower triangular matrix and $\tilde{\boldsymbol{D}}$ is an invertible diagonal matrix (with positive and negative coefficients); $\boldsymbol{M}$ is a permutation matrix. The search directions are $\boldsymbol{F}$-orthogonalized and redundant direction are eliminated by setting:

$$\begin{aligned} \boldsymbol{W}_i &\leftarrow \boldsymbol{W}_i\boldsymbol{M}\begin{pmatrix} \tilde{\boldsymbol{L}}^{-\top} \\ 0 \end{pmatrix}; \quad \boldsymbol{\alpha}_i \leftarrow \tilde{\boldsymbol{D}}^{-1}\begin{pmatrix} \tilde{\boldsymbol{L}}^{-1} & 0 \end{pmatrix}\boldsymbol{M}\boldsymbol{\gamma}_i \\ \boldsymbol{Q}_i &\leftarrow \boldsymbol{Q}_i\boldsymbol{M}\begin{pmatrix} \tilde{\boldsymbol{L}}^{-\top} \\ 0 \end{pmatrix}; \quad \boldsymbol{\Delta}_i^\dagger \leftarrow \tilde{\boldsymbol{D}}^{-1} \end{aligned} \tag{8}$$

3. Numerical assessments

In this section, the classical FETI method is compared to the three solvers presented in Algorithm 1:

- MPFETI, is the optimized application of MPCG to FETI (see Algorithm 1 with no τ-test),

	Preconditioner		Projector	
Variant	**Operator**	**Scaling**	**Operator**	**Scaling**
a	Exact	Stiffness	Exact	Stiffness
b	Exact	Stiffness	Super lumped	Multiplicity
c	Lumped	Stiffness	Lumped	Stiffness
d	Lumped	Stiffness	Super lumped	Multiplicity

Table 1: Combinations of preconditioner and projector considered

- AMPFETIG, or Adaptive multipreconditioned FETI with a Global τ-test, is the application of the global version of Adaptive MPCG to FETI (see Algorithm 1 with the Global τ-test),
- AMPFETIL, or Adaptive multipreconditioned FETI with a Local τ-test, is the application of the local version of Adaptive MPCG to FETI (see Algorithm 1 with the Local τ-test).

Moreover, as stated in Section 2, the FETI method has the advantage of providing several choices for the preconditioner $\tilde{\boldsymbol{S}}$ and for the operator $\boldsymbol{A}$ used in the projector. Each of this choices is in turn defined by the choice of a scaling operator and an approximation of the Schur complement. We propose four different combinations of scaling and operator (see Tab. 1). This way we show that the adaptive multipreconditioning technique can be applied to the user's favorite FETI method and behaves as expected in all cases. Combinations (a) and (b) use the best available preconditioner and only differ in the choice of the projector. Likewise, the same lumped preconditioner is used for combinations (c) and (d) and the difference consists in the choice of the projector. The first two combinations are expected to converge faster than the others in terms of iterations. However, this statement is not so obvious in terms of computational time: even if the lumped preconditioner is numerically less efficient than the Dirichlet preconditioner, its low computational cost may lead to better performance.

Three specific timers have been defined:

- The *Orthog.* timer represents the time spent orthogonalizing the search directions, it corresponds to line 21 in Algorithm 1.
- The *Precond.* timer regroups the time spent computing the preconditioned residual and selecting search directions within the adaptive algorithms, it corresponds to lines 13 to 19 in Algorithm 1.
- The *Operator* timer is the time spent applying the FETI operator $\boldsymbol{F}$, it corresponds to line 7 in Algorithm 1 with the trick presented in equation (6) to make the application of the operator local.

The three aforementioned timers only consider subregions inside the while loop and the rest of the computational time is regrouped in the *Remaining* timer.

This timer consequently times the initialization of Algorithm 1 and operations in the while loop that are not already taken into account by the three previous timers. In particular, the factorization of operators $\boldsymbol{F}$, $\tilde{\boldsymbol{S}}$ and $\boldsymbol{A}$ and the computation of $\boldsymbol{FAG}$ needed to apply the low rank correction (see Section 2.3) are computed once during the initialization phase. Thus, the *Precond.* and *Operator* timers only catch forward and backward substitutions. Also, the time presented is the Wall time (not CPU time) so it is the actual time of the computation and takes into consideration the message passing latency and a perhaps perfectible synchronization of processes.

We first illustrate, in subsection 3.1, the robustness of the multipreconditioned approach for various academic problems known to trigger convergence difficulties of the classical FETI method. Then, we present, in subsection 3.2.1, the advantages of the adaptive approach. More precisely, we illustrate the advantage of local adaptivity with a specific test case, see subsection 3.2.2 whereas the question of the choice of the user threshold τ is raised in subsection 3.2.3. Finally, the weak scalability of the proposed methods is evaluated on problems with various values of material contrast (as expected), see subsection 3.3. These examples come in complement of the two-dimensional test cases already proposed in [1]. In particular, results involving regular interfaces have already been presented in [1].

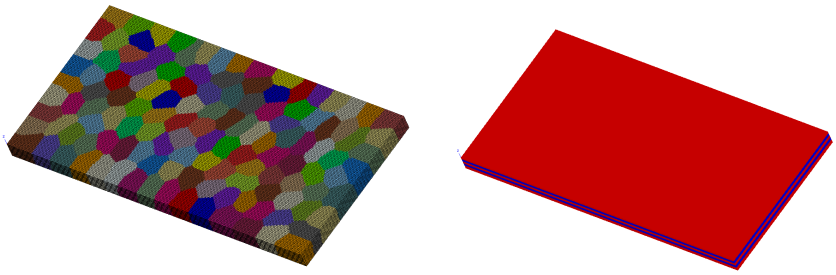
The solvers have been implemented in the finite element suite Z-Set 8.6 [30]. The tests presented in sections 3.1, 3.2 have been conducted on a cluster of 2.6 GHz 16-core Xeon processors connected by an InfiniBand Mellanox network, each subdomain being allocated to one core. The tests presented in sections 3.3 and 4 have been performed on the *Cobalt* supercomputer, which is a 1.299-petaflop Bull system at the french Computing Center for Research and Technology (CCRT, `http://www-ccrt.cea.fr`, ranked 75 in the last TOP500 list from November 2016) made of 1,422 Intel Broadwell nodes inter-connected with an Infiniband EDR network and providing a total of 34,816 available cores, and seven cores have been assigned to each subdomain. In all configuration, the Mumps solver [31] in association with the BLAS library provided by Intel MKL package has been used for local solves. Because of the high material contrast, the local rigid body modes of subdomains have been computed using the geometric-algebraic method [32]. For all computations, the partition of the domain has been obtained by the automatic graph partitioner Metis [33]. For all following computations, the solver is considered converged when the residual has decreased by a factor 10^6.

3.1. Robustness of MPFETI

A rather academic problem is used for this numerical study: a slender heterogeneous (3D) plate with aspect ratio $1\times20\times10$ discretized into $15\times250\times125$ regular twenty-node brick elements (c3d20). This way the problem size is $5,959,218$ degrees of freedom (*dofs*). This mesh is split into 127 domains (see Fig. 1a) composed on average of $55,000$ *dofs* and $3,800$ c3d20 elements. With this decomposition, there is only one subdomain in the thickness which leads to a reasonable number of neighbors per domain (between 2 and 8). It is a fact that

the multipreconditioned algorithms benefit from this low number of neighbors (see Section 2.3). Nevertheless, slender geometries are representative of many structural applications.

The plate is clamped on the left side and both pressure and shear are imposed on the opposite side. It consists of five thin linear elastic layers (see Fig. 1b). All layers share the same Poisson's coefficient 0.3 but the Young's modulus alternates between a stiff value E_1 and a soft value E_2. This case aims at representing a laminated composite material made out of a soft material reinforced by a stiffer one.



(a) Automatic decomposition

(b) Soft material is in red, stiff material in blue

Figure 1: Heterogeneous composite

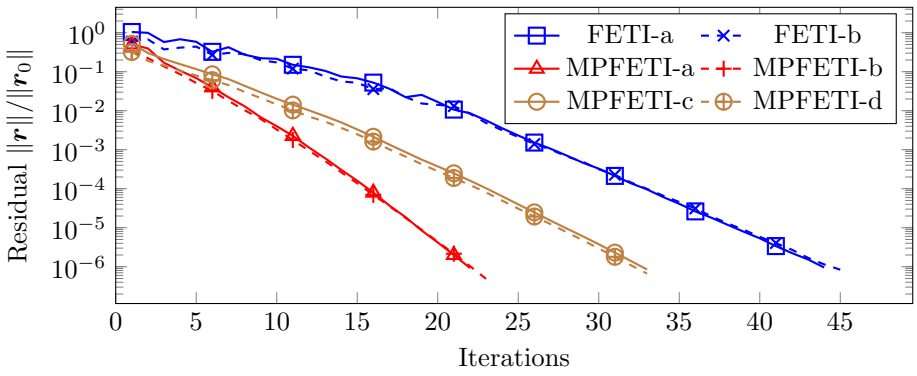
The results are summarized in Tab. 2. All four variants of the solver presented in Tab. 1 are considered with the MPFETI method. Combinations (c) and (d) are not considered for FETI since convergence was only achieved for a small material contrast. This alone is a first argument toward the fact that MPFETI improves the robustness of the FETI algorithms. Figures 2, 3, and 4 all share the same structure: the top graph presents the decrease of the residual and the bottom one displays the computational time for each combination.

The results obtained with the lowest material contrast ($E_1/E_2 = 10^1$), are shown in Fig. 2. In this case, even if the multipreconditioned solvers require approximately half as many iterations to converge as FETI, the computational time is more than double. The computational time repartition plotted in Fig. 2b clearly reveals that the performance of the FETI and MPFETI methods are not driven by the same computational tasks. For multipreconditioned approaches, the orthogonalization of search directions predominantly contributes to the computational time. This was expected since each iteration creates as many search directions as there are subdomains in the decomposition (127 in the present case), and a full modified Gram-Schmidt re-orthogonalization is performed. For the FETI method, the computational time inside the while loop is dominated by the forward and backward substitutions required by local solves. The use of a lumped preconditioner decreases the duration of the preconditioning step but it does not compensate for the additional iterations. For all MPFETI methods, the duration returned by the *Remaining* timer is longer than for the FETI

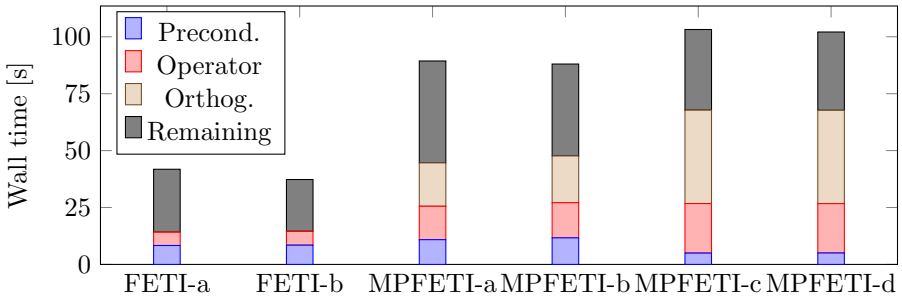
method even if the number of iterations is lower. This is mainly due to a longer initialization phase since extra work is needed to pre-compute $\boldsymbol{FAG}$. Finally, one can see that the choice of the projector has little effect on the convergence, however the superlumped option slightly decreases the computational time of all methods.

Figures 3 and 4 show results obtained with material contrasts of 10^5 and 10^6 respectively. Clearly, for all combinations of the preconditioner and the projector, multipreconditioned approaches are rather insensitive to the increase in the heterogeneity factor. Even for the highest material contrast and the cheapest preconditioner, the number of iterations is at most 42. The FETI method however is significantly penalized by this increase in heterogeneity. With the highest material contrast presented, only the FETI-a method is able to decrease the residual by a factor 10^6, the number of iterations needed is 766. The FETI-b combination, that uses a super lumped projector with a topological scaling, does not manage to decrease the residual by more than a factor 10^4. It however presented a faster convergence during the first iterations. We observed that this behavior is rather systematic, even for the multipreconditioned approaches. Indeed, for all cases, the super lumped projector combined with the multiplicity scaling exhibits a faster convergence during the first iterations. However, and unlike FETI-a, FETI-b does not benefit from an acceleration of the convergence (see Fig. 3a). The same remarks made for Fig. 2 about computational time are still relevant. Once more, for FETI the computational time mostly traces back to the local solves whereas the orthogonalization step is predominant for the multipreconditioned approaches. For a not too large heterogeneity factor of 10^3, computational times are of the same order of magnitude (see Table 1). With a material contrast of 10^5 and 10^6, the computational time of the multipreconditioned methods is more than three times shorter than for FETI. As seen before, the low computational cost of the lumped preconditioner hardly ever reduces the computational time of multipreconditioned approaches. It is barely visible in Figure 4, but this gain remains negligible and not systematic. As a conclusion MPFETI is robust. It is not optimal on the easier test cases where FETI already converges fast. This is expected and resolved by using the adaptive methods (as discussed at length in [2] on academic problems).

Since the lumped preconditioner hardly ever compensates its suboptimality, combinations (c) and (d) are not presented anymore in what follows.

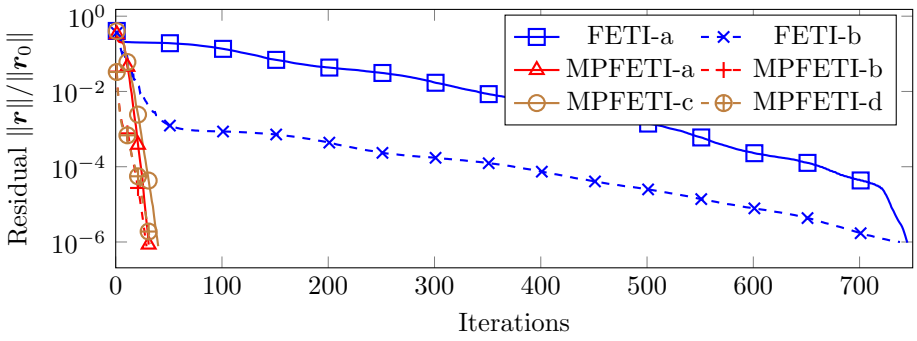


(a) Convergence history $E_1/E_2 = 10^1$

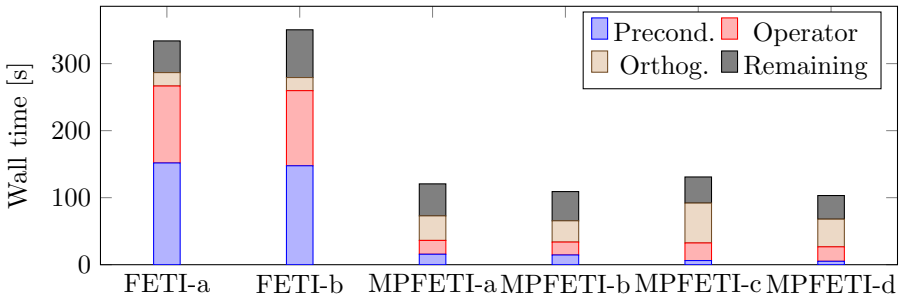


(b) Wall time repartition $E_1/E_2 = 10^1$

Figure 2: Heterogeneous composite: robustness w.r.t. heterogeneities ($E_1/E_2 = 10^1$).

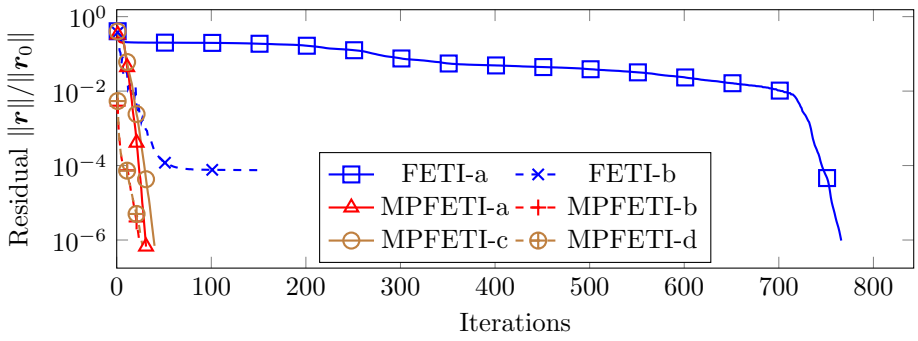


(a) Convergence history $E_1/E_2 = 10^5$

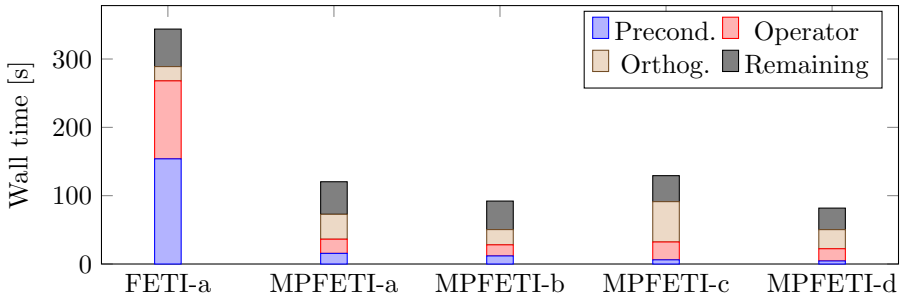


(b) Wall time repartition $E_1/E_2 = 10^5$

Figure 3: Heterogeneous composite: robustness w.r.t. heterogeneites ($E_1/E_2 = 10^5$).



(a) Convergence history $E_1/E_2 = 10^6$



(b) Wall time repartition $E_1/E_2 = 10^6$

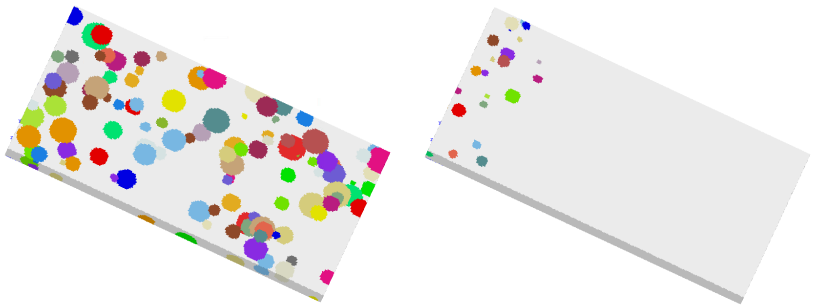
Figure 4: Heterogeneous composite: robustness w.r.t. heterogeneities ($E_1/E_2 = 10^6$). FETI-b is absent in the bottom plot since it failed to converge.

	Counters		Timers (s)			
$E_1/E_2 = 10^1$	# iter.	# search dir.	Total	Precond.	Operator	Orthog.
FETI-a	44	44	41.81	19.9%	13.9%	0.2%
FETI-b	45	45	37.3	22.7%	16.1%	0.2%
MPFETI-a	22	2794	89.37	12.2%	16.4%	21.3%
MPFETI-b	23	2921	88.03	13.3%	17.6%	23.3%
MPFETI-c	33	4191	103.2	4.8%	21.1%	39.8%
MPFETI-d	33	4191	102.1	4.9%	21.2%	40.2%
$E_1/E_2 = 10^3$	# iter.	# search dir.	Total	Precond.	Operator	Orthog.
FETI-a	177	177	99.0	36.1%	26.0%	1.2%
FETI-b	167	167	93.59	35.0%	25.6%	1.2%
MPFETI-a	32	4064	123.3	13.1%	17.1%	31.7%
MPFETI-b	33	4191	126.2	13.3%	17.7%	32.8%
MPFETI-c	42	5334	139.7	4.5%	19.4%	47.4%
MPFETI-d	40	5080	129.7	4.9%	20.4%	46.2%
$E_1/E_2 = 10^5$	# iter.	# search dir.	Total	Precond.	Operator	Orthog.
FETI-a	745	745	333.9	45.5%	34.4%	6.0%
FETI-b	738	738	350.5	42.1%	32.0%	5.6%
MPFETI-a	31	3937	120.6	13.0%	17.1%	30.4%
MPFETI-b	29	3683	109.1	13.3%	17.7%	29.1%
MPFETI-c	40	5080	130.9	4.7%	20.1%	45.5%
MPFETI-d	33	4191	103.2	4.9%	20.9%	40.0%
$E_1/E_2 = 10^6$	# iter.	# search dir.	Total	Precond.	Operator	Orthog.
FETI-a	766	766	343.8	44.8%	33.2%	6.1%
FETI-b	-	-	-	-	-	-
MPFETI-a	31	3937	120.4	12.9%	17.2%	30.3%
MPFETI-b	24	3048	92.11	12.8%	17.6%	24.2%
MPFETI-c	40	5080	129.3	4.7%	20.2%	45.7%
MPFETI-d	27	3429	81.81	5.5%	21.8%	34.2%

Table 2: Heterogeneous composite: summary of results

3.2. Robustness and efficiency of AMPFETI

The previous section has shown the robustness of MPFETI with respect to heterogeneous material. It has also emphasized the large computational effort involved in the orthogonalization of the search directions, which is likely to be the bottleneck of MPFETI in terms of scalability. It is therefore reasonable to wonder if all search directions are useful in the resolution, especially at the end of the computation when the isolated eigenvalues have been well approximated. Selecting only the useful search directions is the objective of the adaptive variants of MPFETI. In this section, results obtained with both global (AMPFETIG) and local (AMPFETIL) adaptive methods are presented. Letters a and b still refers to the combinations of preconditioner and projector as given by Table 1.



(a) Spread out inclusions (b) Gathered inclusions

Figure 5: Plate with heterogeneous inclusions

3.2.1. Randomly distributed inclusions

We consider the same geometry as previously but change the material distribution. This time, heterogeneous spherical inclusions are inserted inside a soft matrix (see Fig. 5). The location, size and Young's modulus of the inclusions are randomly selected using a uniform distribution. Such a test case generates various kind of heterogeneities in the context of domain decomposition: an inclusion may be contained within a domain, grossly match its interfaces, or cross them. The matrix behavior is linear elastic with a Poisson's coefficient of 0.45. All inclusions are linear elastic with a Poisson's coefficient of 0.3. The maximal value taken by the Young's modulus of the inclusions provides a material contrast $E_{inclusion}/E_{matrix} = 10^5$. The radius of the inclusions is bounded between 20% and 80% of the plate thickness.

Two different distributions of the inclusions are presented. In the first case (Fig. 5a), the 200 inclusions are spread out over the plate so that the computational complexity is evenly shared by all subdomains. The second distribution tries to gather the computational complexity: all 75 inclusions are located in a

small subregion of the plate (1/5 of the volume as shown in Fig. 5b). In this section, we solve both of these problems with the threshold τ set to 0.1.

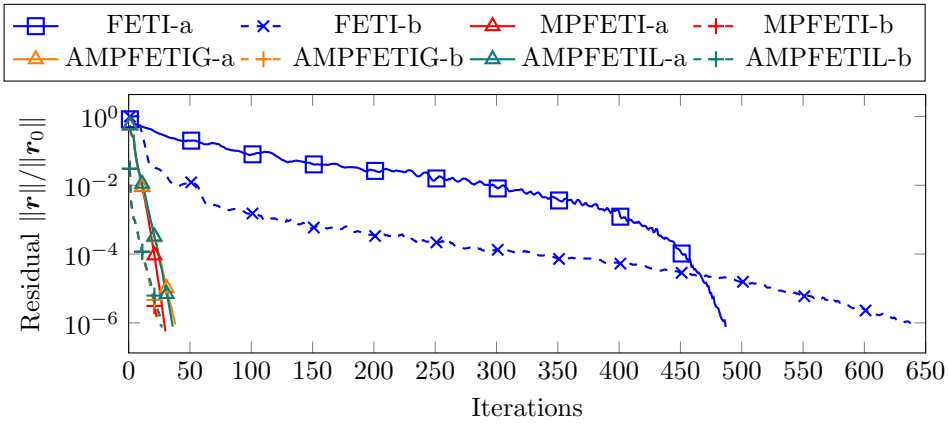
The results are summarized in Tab. 3. We observe that the AMPFETI solvers always converge faster than the MPFETI solvers which in turn always converge faster than the FETI solvers. The adaptive variants (AMPFETI) require a few more iterations to converge than the MPFETI, but their computational time is reduced. In detail, for the spread out inclusions (see Figure 6), the reduction of the computational time with AMPFETIG-a and AMPFETIL-a compared to MPFETI-a, is respectively 34.8% and 30.2%. This gain is mainly induced by a faster orthogonalization step: the *Orthog.* timer decreases by 76.9% and 78.9% for AMPFETIG-a and AMPFETIL-a (respectively) compared to MPFETI-a.

Although both the minimization space and number of iterations of AMPFETIL-a are smaller than with AMPFETIG-a, the computational time of AMPFETIL-a is slightly larger. This is due to the cost of evaluating the τ-test. Indeed, unlike the global τ-test, the cost of the evaluation of the local τ-test is not negligible : with the current implementation one global extra communication is needed for the local τ-test. This is accounted for in the *Precond.* timer. In the end, when the inclusions are spread out over the domain, the local τ-test does not provide a significant speedup compared to the global τ-test, which seems intuitive since the computation complexity is shared by all domains.

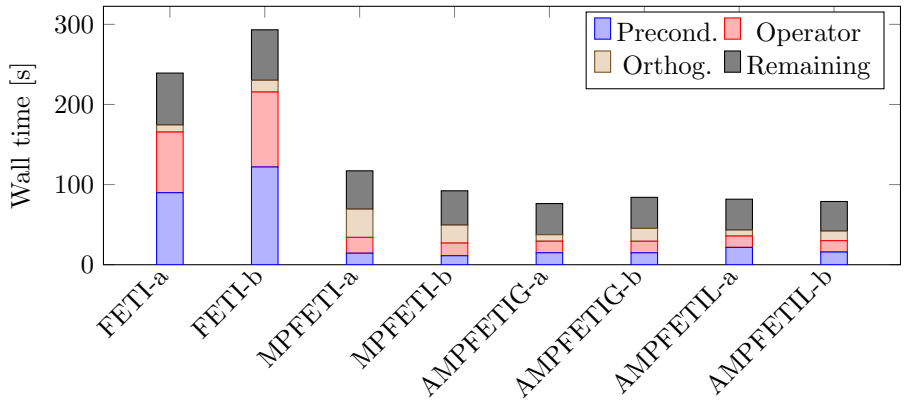
The results obtained when the inclusions are gathered are shown in Figure 7. This time, the FETI method experiences less difficulty to converge since the heterogeneity only concerns a subregion of the plate, and the speedup induced by multipreconditioning is smaller. However, the gain provided by the adaptive variants is worthwile, especially with the local τ-test. As early as the first iterations, the local τ-test only selects a few search directions (see Fig. 7c). The minimization space generated by AMPFETIL-a is six times smaller than the space generated by MPFETI-a. Consequently, the time spent in the orthogonalization of search directions is significantly reduced. It becomes comparable to the orthogonalization cost within the classical FETI method. Even though the global τ-test generates a larger minimization space, it converges more slowly than the local adaptive methods, both in terms of iterations and computational time. In this case, all search directions selected by the global τ-test do not provide valuable information. This seems consistent and directly related to the gathering of the inclusions: when the global τ-test is triggered, the search directions proposed by all subdomains are included, even those proposed by subdomains far away from the inclusions.

	Spread out inclusions ($\tau = 0.1$)					
	Counters		Timers (s)			
	# iter.	# search dir.	Total	Precond.	Operator	Orthog.
FETI-a	487	487	239.2	37.6%	31.6%	3.7%
FETI-b	638	638	293.2	41.6%	31.9%	5.0%
MPFETI-a	30	3810	117.2	12.3%	16.8%	30.0%
MPFETI-b	24	3048	92.27	12.1%	17.1%	24.3%
AMPFETIG-a	38	1550	76.33	19.6%	18.9%	10.6%
AMPFETIG-b	26	2420	84.09	17.7%	17.1%	19.0%
AMPFETIL-a	36	1219	81.79	26.5%	17.3%	9.1%
AMPFETIL-b	27	1948	78.91	20.1%	17.9%	15.1%
Gathered inclusions($\tau = 0.1$)						
	# iter.	# search dir.	Total	Precond.	Operator	Orthog.
FETI-a	181	181	126.2	32.4%	26.4%	1.2%
FETI-b	244	244	170.7	39.2%	30.8%	1.5%
MPFETI-a	24	3048	110.4	13.0%	18.6%	21.7%
MPFETI-b	19	2413	87.79	13.0%	19.2%	17.7%
AMPFETIG-a	28	1666	86.05	18.7%	18.8%	10.8%
AMPFETIG-b	19	2287	84.72	15.8%	18.8%	15.6%
AMPFETIL-a	27	486	73.3	25.6%	16.9%	3.3%
AMPFETIL-b	20	812	66.03	20.4%	17.1%	5.5%

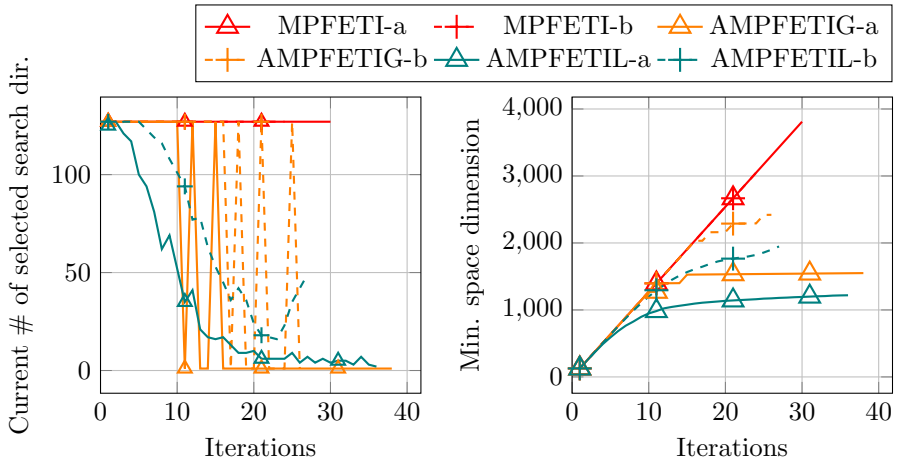
Table 3: Heterogeneous inclusions: summary of results. AMPFETI converges faster than MPFETI which in turn converges faster than FETI.



(a) Convergence history for spread out inclusions

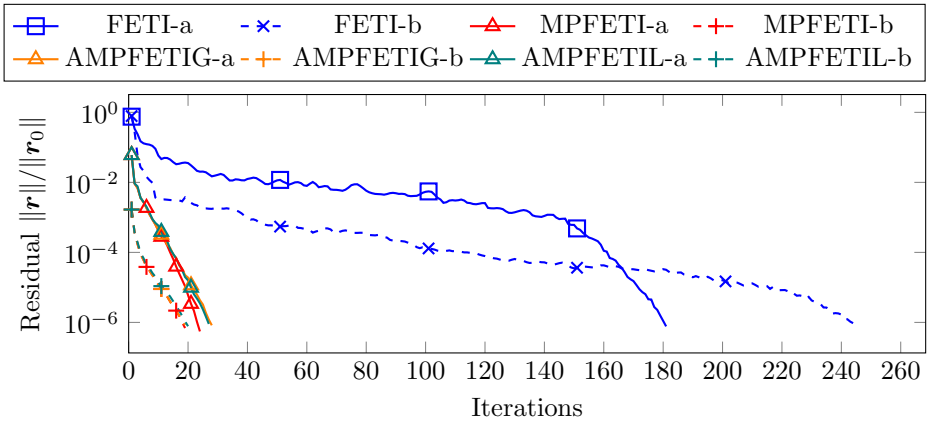


(b) Wall time repartition for spread out inclusions

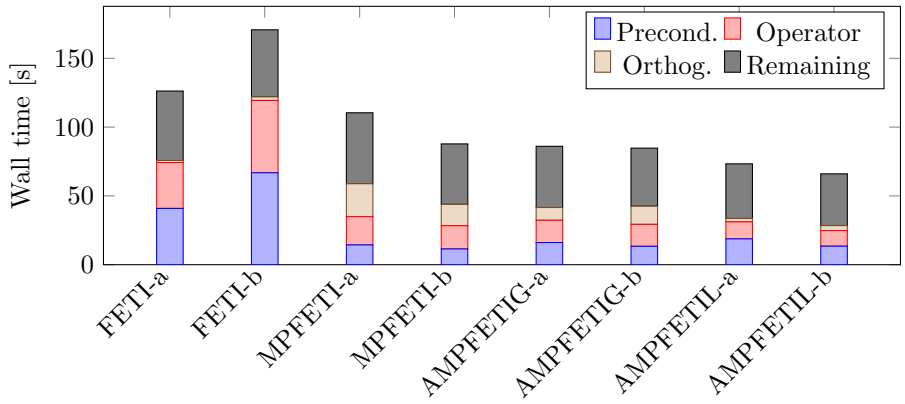


(c) Evolution of the size of minimization space for spread out inclusions

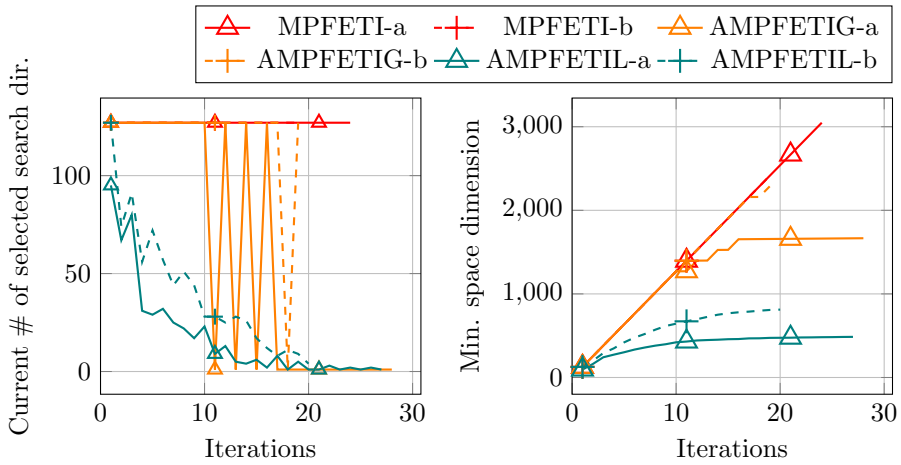
Figure 6: Convergence, time repartition and search space evolution for spread out inclusions



(a) Convergence history for gathered inclusions



(b) Wall time repartition for gathered inclusions

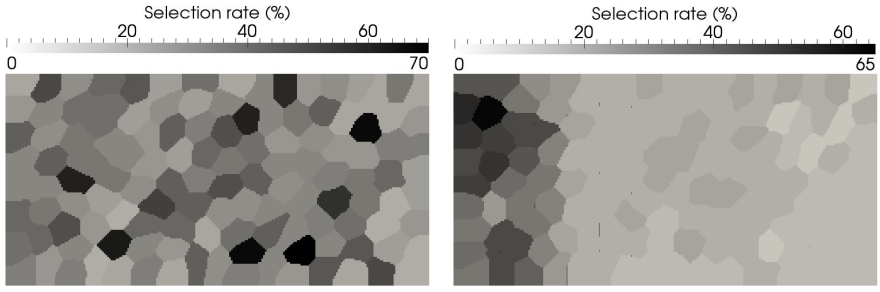


(c) Evolution of the size of the minimization space for gathered inclusions

Figure 7: Convergence and time repartition for gathered inclusions

3.2.2. *Gathering of the selected search directions when the inclusions are gathered*

For AMPFETIL (local τ-test), we define a selection rate as the ratio between the number of times the search direction proposed by a given subdomain is selected and the total number of iterations. Figure 8 shows the selection rate map obtained with AMPFETIL-a for the spread out inclusions and the gathered inclusions. When the inclusions are spread out over the volume (see Fig. 5a) the selection rate map does not exhibit any particular profile (Fig. 8a). However, when the inclusions are gathered (see Fig. 5b), Figure 8b distinctly reveals a higher selection rate for search directions provided by domains near the inclusions. The local τ-test detects on the fly the difficulties that slow down the convergence of FETI and triggers the multipreconditioning mechanism which accelerates convergence. This is the expected behaviour of the local τ-test and why it should prove very useful on industrial test cases where the complexity of the simulation is not evenly spread out over the domain.



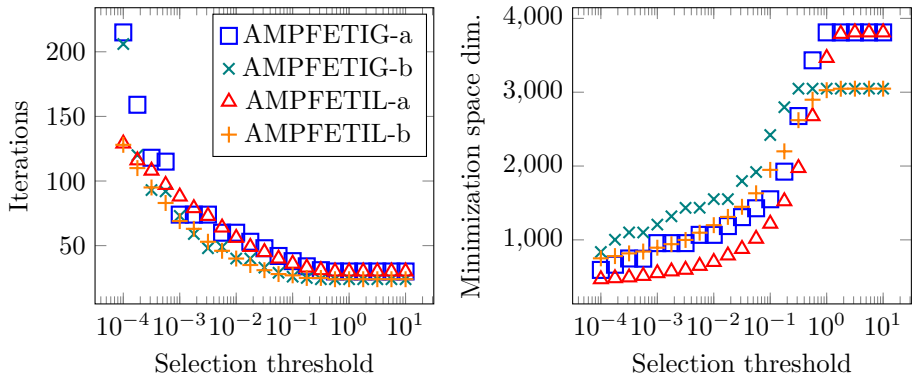
(a) Randomly distributed inclusions (b) Locally distributed inclusions

Figure 8: Selection rate of search directions provided by each domain (AMPFETIL-a)

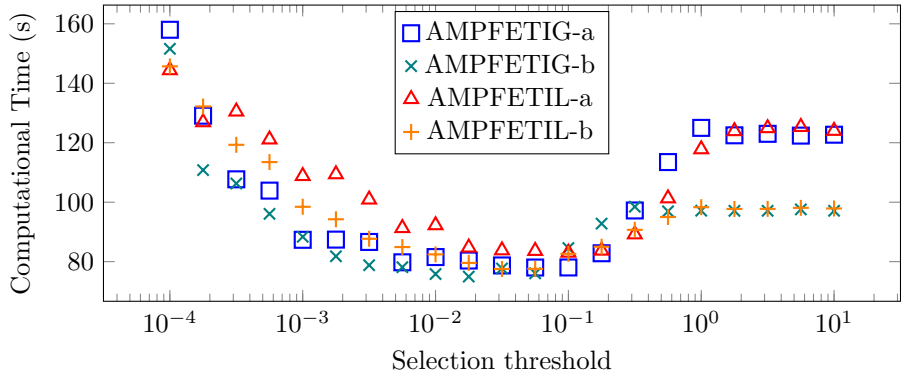
3.2.3. *Sensitivity of the adaptive process with respect to τ*

In the previous section, the (only) user parameter τ, which drives the adaptivity test was arbitrarily set to 0.1. Although τ can be connected to a targetted contraction factor (see Section 2.2 and [2]), it is an intricate task to estimate *a priori* an optimal value of τ, leading to the best computational time. It is indeed expected to depend on the test case (size and number of subdomains, size of the interfaces, heterogeneities, average number of neighbors...) and on the hardware characteristics (processors frequency, network bandwith and latency).

This section investigates the influence of τ on the performance of adaptive methods AMPFETIL and AMPFETIG. Both test cases with heterogeneous inclusions are solved for twenty one values of τ regularly spaced out (in logarithmic scale) over the interval $[10^{-4}, 10^{1}]$. The number of iteration, the size of the minimization space and the computational time for all values of τ are plotted in Figures 9 and 10. We recall that in the limit $\tau \to 0$, both AMPFETIL and AMPFETIG behave like the classical FETI, whereas for large values of τ all



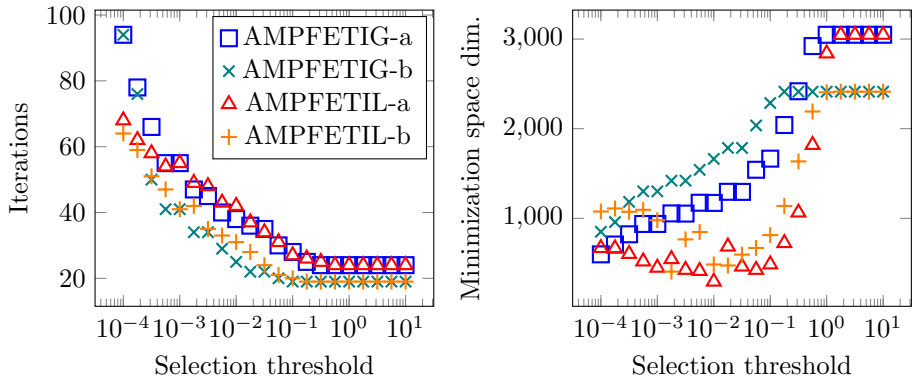
(a) Influence of τ on the number of iterations (b) Influence of τ on minimization space dim.



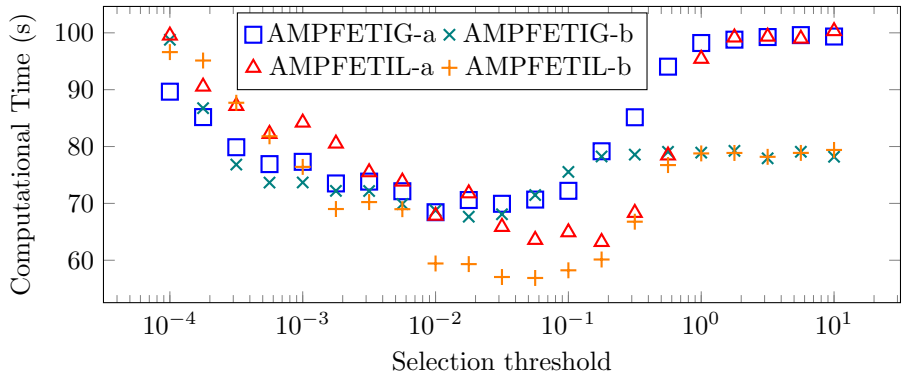
(c) Influence of τ on the computational time

Figure 9: Spread out inclusions: τ-sensitivity of the performance

candidate search directions are selected and the multipreconditioned iterations are recovered. As expected, for both test cases, the dimension of the minimization space becomes larger with the increase of τ whereas the number of iterations is reduced. In terms of computational time, there is an optimal value of τ which provides the best compromise between the reduction of the number of iterations and the cost induced by extra search directions. The good news is that the performance is not very sensitive to changes in τ since all values in the range $[10^{-2}, 10^{-1}]$ lead more or less to the same performance. This confirms the observations in [2]. The advantage of the local τ-test when the inclusions are gathered is again highlighted in Figure 10. Indeed, for a large range of τ, and despite a much smaller minimization space, the number of iterations of AMPFETIL-a (resp. AMPFETIL-b) and AMPFETIG-a (resp. AMPFETIG-b) are similar leading to a reduced computational time.



(a) Influence of τ on the number of iterations (b) Influence of τ on minimization space dim.

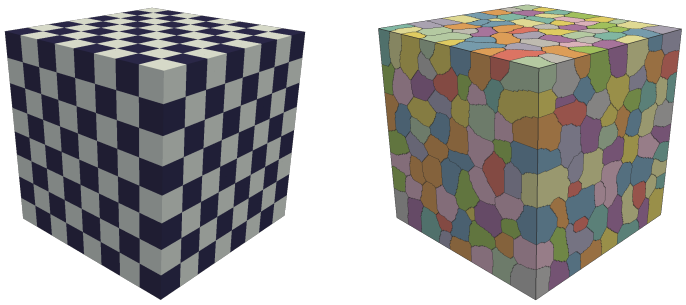


(c) Influence of τ on the computational time

Figure 10: Gathered inclusions: τ-sensitivity of the performance

3.3. Weak parallel scalability results

In this section, we study the weak parallel scalability of the previous algorithms up to 512 subdomains for both homogeneous and heterogeneous problems. For $N_c \in \{2, 3, 4, 5, 6, 7, 8\}$, we consider a set of three dimensional heterogeneous cube made of N_c identical cubic sub-structures (see Figure 11). Each sub-cube is discretized with the same ruled mesh made of 64,000 eight-node brick elements (c3d8), leading to a total number of approximately $N_c^3 \times 206,763$ *dofs* with a corresponding H/h ratio of 40. The cube is clamped on one side and subject to a prescribed unitary displacement in the three space directions on the opposite one, all other faces remaining traction-free. The material behavior is isotropic linear elastic, with a Poisson's coefficient of 0.3 and two values of Young's modulus assigned following a checkerboard pattern in order to obtain a coefficient jump E_1/E_2 between two adjacent sub-cubes. Finally, an unstructured decomposition in N_c^3 subdomains is obtained with a partitioning software which leads to interfaces no longer aligned with heterogeneities. Such a configuration is represented on Figure 11 for $N_c = 8$.



(a) Checkerboard cube (b) Automatic decomposition

Figure 11: Heterogeneous cube (configuration with 512 subdomains))

In the following, we compare FETI, AMPFETIL and AMPFETIG algorithms with three increasing values of ratio E_1/E_2 given by the set $\{1, 10^3, 10^6\}$. Let us note that MPFETI algorithm was excluded from this study for two reasons. First, the regular increase of the underlying minimization space becomes a serious drawback regarding memory footprint and time spent on the orthogonalization step when applied to large computations involving several hundred domains. Second, previous results from Section 3.2 show that AMPFETIL and AMPFETIG algorithms perform almost as well as MPFETI regarding the number of iterations needed to achieve convergence and better regarding total elapsed time, while keeping a significantly smaller minimization space. All algorithms are equipped with the best state of the art preconditioner and projector corresponding to the combination (a) of Table 1 and the convergence criterion ϵ is set to 10^{-6}. We also set the τ-test to 10^{-2} which, despite not being the optimal value, offers a good trade-off between the global performance of the algorithm and the size of the minimization space, as stated in Section 3.2.3. All

N_c	N	#DOFs total	#DOFs on interface	#cores
2	8	1,594,323	74,514	56
3	27	5,314,683	365,802	189
4	64	12,519,843	918,129	448
5	125	24,361,803	1,945,023	875
6	216	41,992,563	3,520,671	1,512
7	343	66,564,123	5,725,812	2,401
8	512	99,228,483	8,624,838	3,584

Table 4: Weak scalability: summary of configurations studied

the computations have been performed on the *Cobalt* supercomputer described at the beginning of the section. Table 4 summarizes the different configurations.

All the results for the three values of coefficient jumps are summarized in tables 5, 6 and 7. These results are also shown in Figures 12, 13 and 14.

First, we note that the variants of AMPFETI always converge in fewer iterations than the FETI method, which is in accordance with the results from the previous sections (see Figures 12b, 13b and 14b). Here, the most important fact for AMPFETIL and AMPFETIG is probably that the number of iterations to reach convergence is independent from the number of subdomains whatever the level of heterogeneity. Furthermore, it is only subject to a moderate increase when the ratio of heterogeneities increases (jumping to six orders of magnitude), which shows both the efficiency of the adaptive approach and also that the choice of 10^{-2} for the τ-test remains a good compromise in order to limit the dimension of the minimization space. On the other hand, again, we notice that the FETI method is significantly penalized by the increase of heterogeneities and also the number of subdomains for high value of the ratio E_1/E_2. More precisely, on the 512 subdomains configuration, FETI method needs 4 times more iterations to converge when E_1/E_2 is equal to 10^3 and more than 16 times for a ratio of 10^6. Again, it is still noticeable that AMPFETIG needs a little more iterations to converge than AMPFETIL, despite a larger minimization space, even on this particular problem where heterogeneities are spread over the whole space rather than gathered, which is in accordance with results of Section 3.2.

Figures 13a and 14a showing the global elapsed time of the solver corroborate the previous observations for heterogeneous problems. Indeed, both AMPFETIL and AMPFETIG exhibit fast solution, but AMPFETIG is always about 30% faster than AMPFETIL for the same reasons as stated in Section 3.2. Specifically, on the largest problem with 512 subdomains, the reduction of computational time compared to FETI is 45% (resp. 30%) for AMPFETIG (resp. AMPFETIL) when E_1/E_2 is set to 10^3, and 86% (resp. 81%) for a value of 10^6. As for the homogeneous configuration, from Figure 12a and Table 5, we may notice that AMPFETIG behaves almost the same way as FETI, whereas the computational time of AMPFETIL is larger, which is due to the extra cost of the local τ-test as well. As for the repartition of the computational time, the remarks already mentioned in previous sections are still relevant.

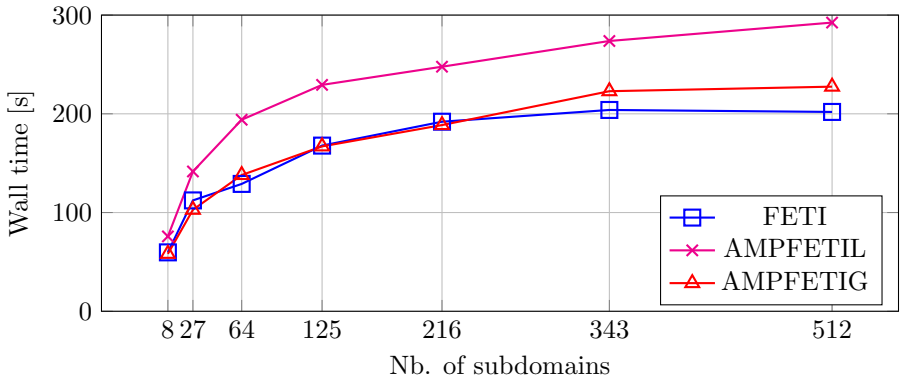
Finally, Figures 12c, 13c and 14c show the ratio between the size of the

		Counters		Timers (s)			
N	#DOFs ($\times 10^6$)	# iter.	# search dir.	Total	Precond.	Operator	Orthog.
FETI							
8	1.6	43	43	59.67	36.65%	25.10%	0.49%
27	5.3	79	79	112.30	47.19%	25.84%	0.71%
64	12.5	82	82	129.00	46.57%	24.41%	0.80%
125	24.4	106	106	167.80	48.35%	24.39%	1.18%
216	42.0	119	119	192.00	47.42%	24.25%	1.49%
343	66.6	130	130	203.90	47.76%	24.79%	1.81%
512	99.2	121	121	201.90	44.54%	23.64%	1.74%
AMPFETIL							
8	1.6	39	46	75.83	49.83%	18.53%	0.12%
27	5.3	56	97	141.60	59.00%	14.61%	0.27%
64	12.5	61	121	194.00	58.51%	11.82%	0.48%
125	24.4	65	231	229.30	54.80%	11.03%	0.98%
216	42.0	64	315	247.70	49.74%	10.48%	1.51%
343	66.6	67	424	273.76	49.49%	10.19%	2.23%
512	99.2	66	684	292.40	44.29%	9.98%	3.45%
AMPFETIG							
8	1.6	41	48	58.16	34.29%	25.85%	0.15%
27	5.3	63	115	102.80	42.78%	24.33%	0.37%
64	12.5	75	138	138.00	39.79%	22.52%	0.61%
125	24.4	76	324	167.00	37.06%	20.64%	1.21%
216	42.0	75	505	188.60	33.44%	19.41%	1.78%
343	66.6	87	771	222.89	31.90%	20.90%	2.79%
512	99.2	81	1103	227.50	28.77%	20.53%	3.81%

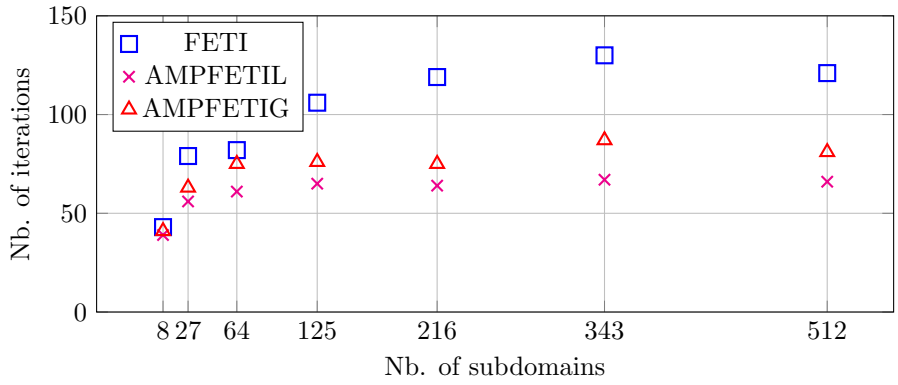
Table 5: Weak scalability ($E_1/E_2 = 1$): summary of results

minimization space produced and the number of subdomains N. We notice that (at least with this checkerboard test case and this value of τ), the size of the minimization spaces produced seems to be proportional to the number of subdomains. This ratio quickly stagnates with the increase of N. Furthermore, this ratio remains almost the same (around 10) when E_1/E_2 increase from 10^3 to 10^6, which illustrate the efficiency of the selection process of relevant search directions. We also notice that the minimization space of AMPFETI variants is 5 to 10 times larger than those of FETI, but that is the price to pay for the convergence properties of AMPFETI variants. Indeed, this factor tends to decrease as the heterogeneity is getting worse (and so does the performance of the FETI method).

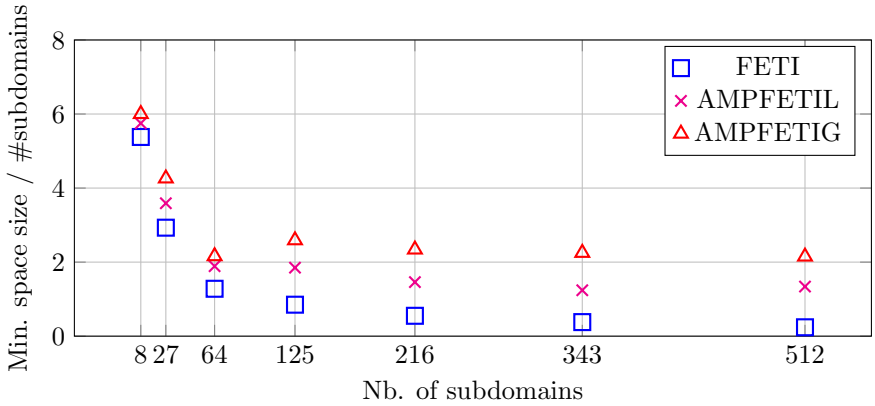
As a conclusion, AMPFETIL and AMPFETIG show very good scalability properties in both convergence and computational time, even on highly heterogeneous problem up to 512 subdomains on a 100 million *dofs* problem. Moreover, AMPFETIG shows almost the same computational time behavior as FETI on homogeneous problems. Finally, regarding time to solution, it seems that the use of AMPFETIL should be limited to problems with gathered heterogeneities, as stated in Section 3.2.2.



(a) Total wall time

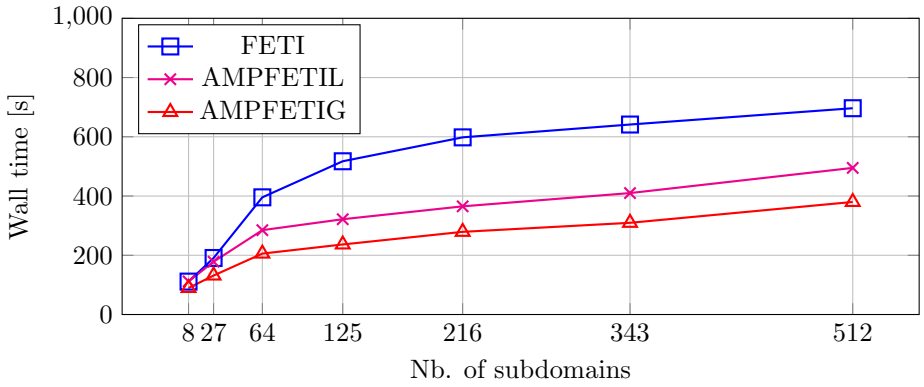


(b) Nb. of iterations vs. nb. of subdomains

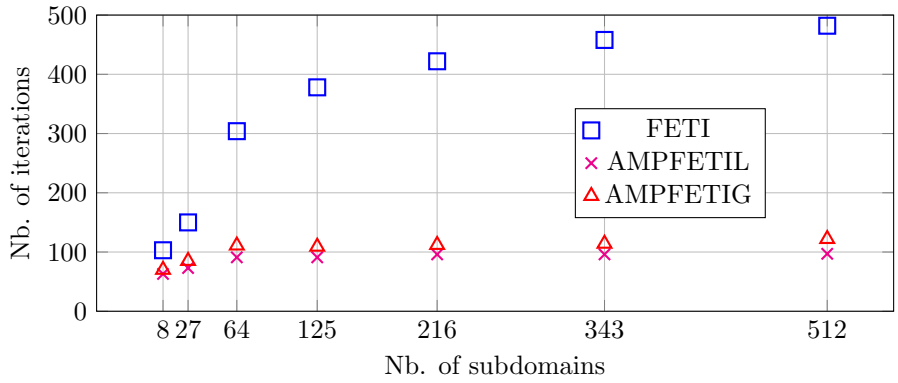


(c) Normalized minimization space size vs. nb. of subdomains

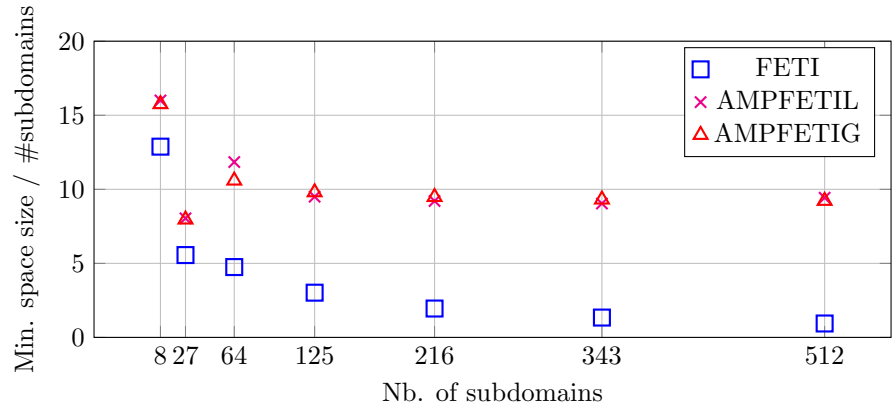
Figure 12: Weak scalability ($E_1/E_2 = 1$): time to solution, nb. of iterations and minimization space size



(a) Total wall time

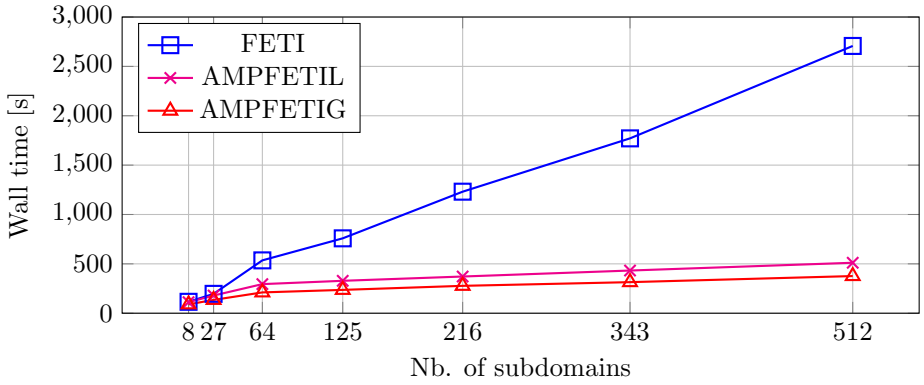


(b) Nb. of iterations vs. nb. of subdomains

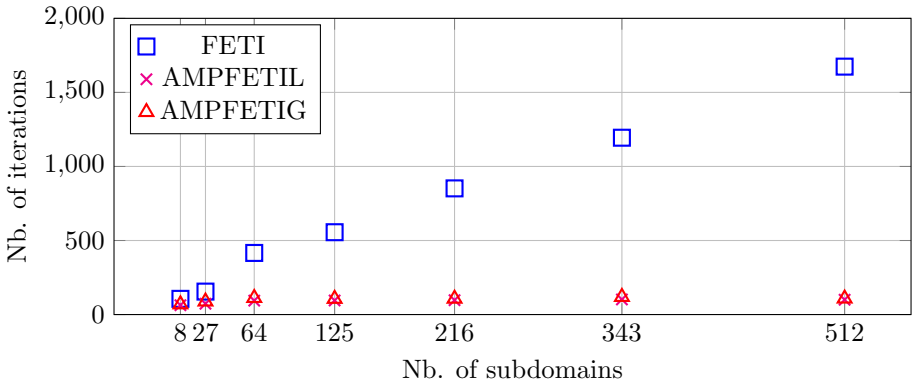


(c) Normalized minimization space size vs. nb. of subdomains

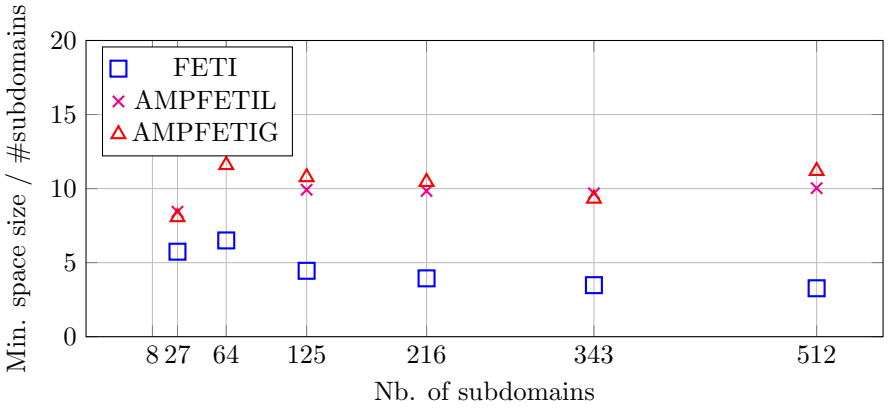
Figure 13: Weak scalability ($E_1/E_2 = 10^3$): time to solution, nb. of iterations and minimization space size



(a) Total wall time



(b) Nb. of iterations vs. nb. of subdomains



(c) Normalized minimization space size vs. nb. of subdomains

Figure 14: Weak scalability ($E_1/E_2 = 10^6$): time to solution, nb. of iterations and minimization space size

		Counters		Timers (s)			
N	#DOFs ($\times 10^6$)	# iter.	# search dir.	Total	Precond.	Operator	Orthog.
FETI							
8	1.6	103	103	111.10	46.49%	32.29%	0.88%
27	5.3	150	150	190.80	52.88%	28.98%	1.48%
64	12.5	304	304	395.66	56.40%	29.26%	3.33%
125	24.4	378	378	517.40	55.98%	27.97%	4.48%
216	42.0	422	422	598.10	53.80%	27.40%	5.43%
343	66.6	458	458	641.40	53.34%	27.51%	6.48%
512	99.2	482	482	696.50	51.16%	27.13%	7.08%
AMPFETIL							
8	1.6	63	128	111.80	54.01%	23.10%	0.21%
27	5.3	73	217	178.20	61.11%	16.23%	0.43%
64	12.5	91	758	284.80	59.59%	14.46%	1.54%
125	24.4	91	1187	321.70	54.97%	12.97%	2.60%
216	42.0	96	1989	365.20	50.66%	12.57%	4.62%
343	66.6	96	3102	410.00	47.46%	11.96%	7.46%
512	99.2	97	4829	495.00	38.41%	10.91%	15.19%
AMPFETIG							
8	1.6	70	126	89.53	42.32%	31.74%	0.27%
27	5.3	85	215	131.60	46.11%	27.45%	0.60%
64	12.5	111	678	205.69	45.33%	26.52%	1.96%
125	24.4	109	1225	236.50	41.04%	24.47%	3.22%
216	42.0	112	2047	279.10	37.42%	23.19%	5.45%
343	66.6	114	3192	309.60	33.66%	23.10%	8.61%
512	99.2	122	4721	380.00	28.66%	22.46%	12.39%

Table 6: Weak scalability ($E_1/E_2 = 10^3$): summary of results

		Counters		Timers (s)			
N	#DOFs ($\times 10^6$)	# iter.	# search dir.	Total	Precond.	Operator	Orthog.
FETI							
8	1.6	106	106	115.60	46.65%	32.61%	1.00%
27	5.3	155	155	196.10	53.22%	29.05%	1.53%
64	12.5	416	416	535.30	57.00%	29.63%	4.61%
125	24.4	556	556	758.50	56.13%	28.03%	6.60%
216	42.0	852	852	1231.00	52.76%	26.81%	10.60%
343	66.6	1194	1194	1771.00	50.33%	25.90%	15.55%
512	99.2	1674	1674	2707.00	45.72%	24.23%	21.13%
AMPFETIL							
8	1.6	63	132	111.90	53.87%	23.17%	0.21%
27	5.3	73	228	179.10	61.19%	16.22%	0.46%
64	12.5	94	838	294.70	59.59%	14.43%	1.65%
125	24.4	95	1240	328.30	56.08%	13.16%	2.69%
216	42.0	97	2125	371.70	50.42%	12.58%	4.89%
343	66.6	102	3319	432.30	47.84%	11.93%	7.99%
512	99.2	100	5134	511.10	38.48%	10.93%	16.24%
AMPFETIG							
8	1.6	69	132	89.58	42.21%	31.76%	0.27%
27	5.3	88	218	134.80	46.69%	27.74%	0.61%
64	12.5	112	742	210.80	45.30%	26.54%	2.18%
125	24.4	106	1346	236.30	40.69%	24.57%	3.41%
216	42.0	107	2257	277.30	36.45%	23.13%	5.81%
343	66.6	119	3197	314.70	33.52%	23.42%	8.62%
512	99.2	108	5729	376.30	26.96%	22.12%	13.98%

Table 7: Weak scalability ($E_1/E_2 = 10^6$): summary of results

4. Application to a real engineering problem : woven Organic Matrix Composite material (OMC)

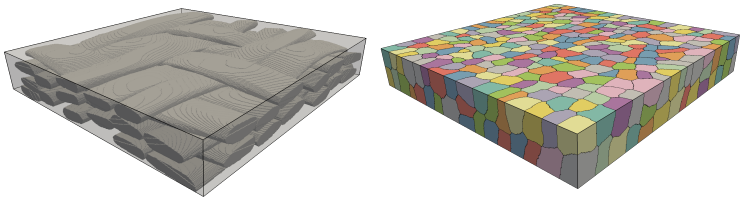
As an application to a real engineering problem, we choose the scope of architectured materials and more specifically of woven Organic Matrix Composites (OMC). This type of material is increasingly studied and used, *e.g.* in the automotive and aircraft industries, as it provides interesting properties in order to maximize strength while decreasing total mass. In the following, we consider a cuboid volume of composite material with an aspect ratio of $19 \times 19 \times 3$, made of a 26 yarn weave pattern inside a polymer matrix (Fig. 15). Typical use cases may be found in numerical homogenization procedures, where a Representative Elementary Volume (REV) of a given material is used to identify the equivalent macroscopic properties by means of the averaging of the mechanical response under specific load conditions (*e.g.* periodic, mixed, uniform) [34].

In order to accurately describe the architecture at the mesoscopic scale, given the complexity of the underlying weave, this type of problem may lead to a high number of *dof*: from several dozens of millions to several billions when taking into account fine morphologic details. Furthermore, as weave pattern complexity increases, its characteristic scale may be too high compared to that of the macroscopic structure such that no scale separation hypothesis can be used anymore. As a result, to get an accurate description of the medium, the mesoscopic model has to be used on the whole structure which, again, leads to a discretized problem with a high number of *dofs*.

The present weave pattern has been obtained by simulation [35] and a voxel-based finite element description is used to model the woven mesostructure [36]. The cuboid volume is discretized into a structured mesh of $30,223,800$ regular eight-node fully integrated hexahedral elements (c3d8), which leads to a global size of $91,995,540$ *dofs*. Depending on its spatial position, each element is either associated to one of the yarns, or to the matrix. All yarns share the same material behavior, chosen to be an orthotropic linear elastic law with a primary direction corresponding to the orientation of the yarn, whereas the behavior of the matrix is chosen to be isotropic linear elastic. The maximal ratio between all Young's moduli related to the yarns and the matrix is $6.9.10^3$. The cuboid is clamped on one side and subject to prescribed displacements corresponding to pure tension on the opposite side.

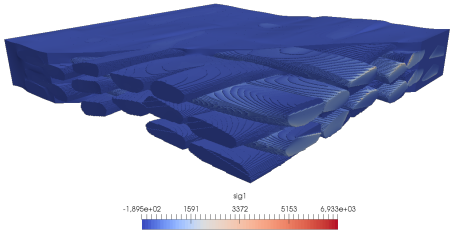
In the following, three configurations of splitting are considered, corresponding to $N = 175$, 250 and 500 subdomains (see Fig. 15 for $N = 500$), which lead to an average of about $555,000$, $390,000$ and $200,000$ *dofs* per subdomain, respectively. As in Section 3.3, both FETI, AMPFETIL and AMPFETIG algorithms, equipped with their best state of the art preconditioner and projector corresponding to variant (a) in table 1, are compared. Again, MPFETI has been excluded from this study for the same reasons as aforementioned in Section 3.3. All the computations have been performed on the *Cobalt* supercomputer (see Section 3.3) and the same values $\epsilon = 10^{-6}$ and $\tau = 10^{-2}$ are chosen for the convergence and search direction selection criteria as in Section 3.3.

The results, which are summarized in Table 8, are in accordance with those



(a) Woven microstructure

(b) Automatic decomposition (500 subdomains)



(c) Tensile principal stresses in the load direction

Figure 15: Representative elementary volume of woven composite

previously obtained.

Figure 16, which plots the global computational time of the three methods, clearly emphasizes the gain provided by the AMPFETI variants. Indeed, we may notice that the AMPFETIG method reduces the time to solution by a factor of more than two in comparison with FETI. Again, AMPFETIL shows slightly less efficiency compared to AMPFETIG, with a computational time about 20% larger. Besides, the time repartition of the different steps is uniformly decreasing as the number of subdomains increases, except regarding the orthogonalization step, which is logical since the size of the minimization space increases. Figure 17 shows the speedup of the three methods with respect to the 175 subdomain configuration, as defined by:

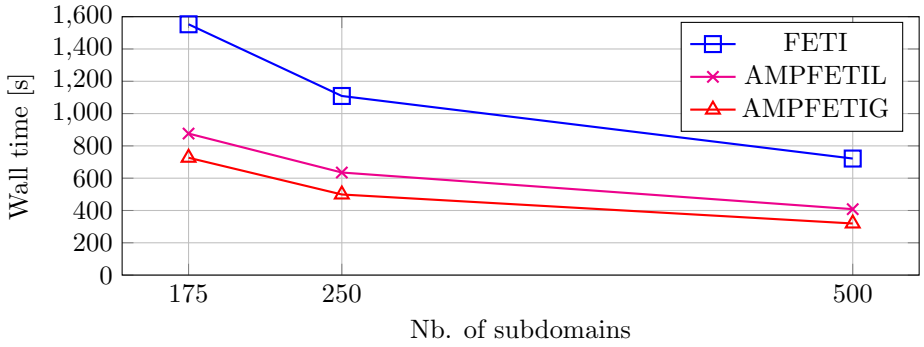
$$\text{Speedup}(N \text{ sd}) = \frac{\text{Wall time}(175 \text{ sd})}{\text{Wall time}(N \text{ sd})}$$

We note that all speedup from the three methods have nearly identical behavior, however, AMPFETIG has a slightly better one. Finally, we note from Table 8 that the size of the minimization space for AMPFETIL and AMPFETIG is about ten times the number of subdomains, whatever the splitting configuration, which corroborates the remark of Section 3.3.

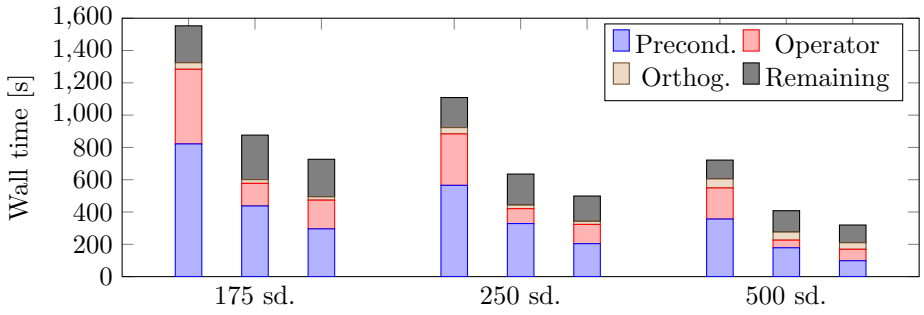
As a conclusion, AMPFETI variants have shown their ability to provide solution to this engineering problem twice as fast as the FETI method, exhibiting a numerical behavior consistent with what has been previously observed.

	Counters		Timers (s)			
$N = 175$	# iter.	# search dir.	Total	Precond.	Operator	Orthog.
FETI	371	371	1553.00	52.91%	29.77%	2.59%
AMPFETIL	103	1627	876.30	49.93%	15.97%	2.63%
AMPFETIG	120	1686	726.60	40.69%	24.49%	2.85%
$N = 250$	# iter.	# search dir.	Total	Precond.	Operator	Orthog.
FETI	405	405	1109.00	50.97%	28.71%	3.52%
AMPFETIL	103	2026	635.10	51.74%	14.52%	3.49%
AMPFETIG	121	2113	499.10	40.88%	23.87%	3.83%
$N = 500$	# iter.	# search dir.	Total	Precond.	Operator	Orthog.
FETI	544	544	721.60	49.40%	26.70%	7.75%
AMPFETIL	104	3907	408.20	43.73%	11.60%	12.25%
AMPFETIG	122	4613	319.52	30.75%	22.25%	12.64%

Table 8: Woven composite REV: summary of results



(a) Total wall time



(b) Wall time repartition for FETI (left), AMPFETIL (center) and AMPFETIG (right)

Figure 16: Woven composite REV: wall time vs. nb. of subdomains

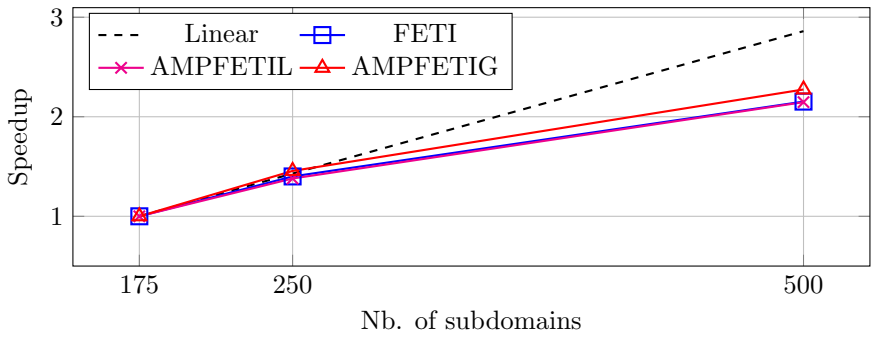


Figure 17: Woven composite REV: speedup vs. nb. of subdomains

5. Conclusion

In this article we have investigated the application to FETI of the adaptive multipreconditioned conjugate gradient algorithm [2] on problems of industrial complexity and on modern hardware. More precisely there are two adaptive variants: either with the Local or the Global τ-test. The Global test embeds a criterion which choses at each iteration whether a classical FETI iteration is sufficient to sufficiently reduce the error or if MPFETI would do significantly better. The Local test involves more computations but it allows to let only the meaningful subdomain individually contribute to the minimization space; which potentially generates a smaller search space.

Our numerical experiments show AMPFETI is indeed robust and gives interesting speedups compared to classical FETI. Tests went up to 500 subdomains, and more than 3500 cores, without any performance drop, letting us hope that the methods shall perform correctly on larger numbers of subdomains. We do note that, even if 500 subdomains is not the actual limit for our current code and hardware, it is clear that as such AMPFETI would not scale up to tens of thousand subdomains. We are currently investigating other adaptive strategies to further improve this bottleneck.

On most cases, the Global test proved to be the more efficient: even if it is less precise than the Local test, the absence of extra computation made it more efficient in term of computational time. The only situation where the Local test was significantly more interesting was cases where difficulties (heterogeneities) were concentrated in local areas such that the rest of the domain could be treated in a classical FETI way.

This article has confirmed the robustness and efficiency of AMPFETI compared to FETI and MPFETI and we believe that AMPFETI is a well suited parallel solver for the linear systems that arise from industrial structural dynamic simulations.

Acknowledgment

This work benefited from the support of the project SEMARFOR ANR-14-CE07-0037 of the French National Research Agency (ANR). The second author would also like to thanks Y. Wielhorski and J. Schneider from Safran Aircraft Engines for providing the engineering test case presented in Section 4.

References

[1] P. Gosselet, D. 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) (2015) 905–927, nme.4946. doi:10.1002/nme.4946.

[2] N. Spillane, An Adaptive Multipreconditioned Conjugate Gradient Algorithm, SIAM J. Sci. Comput. 38 (3) (2016) A1896–A1918. doi:10.1137/15M1028534.

[3] J. Mandel, Balancing domain decomposition, Communications in Numerical Methods in Engineering 9 (3) (1993) 233. doi:10.1002/cnm.1640090307.

[4] C. Farhat, F. X. Roux, Implicit parallel processing in structural mechanics, Computational Mechanics Advances 2 (1) (1994) 1–124, north-Holland. doi:10.1002/nme.1620371111.

[5] C. Farhat, M. Lesoinne, P. LeTallec, 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) (2001) 1523–1544. doi:10.1002/nme.76.

[6] C. R. Dohrmann, A preconditionner for substructuring based on constrained energy minimization, SIAM Journal for Scientific Computing 25 (2003) 246. doi:10.1137/s1064827502412887.

[7] A. Klawonn, M. Lanser, O. Rheinbach, Toward Extremely Scalable Nonlinear Domain Decomposition Methods for Elliptic Partial Differential Equations, SIAM Journal on Scientific Computing 37 (6) (2015) C667–C696. doi:10.1137/140997907.

[8] S. Badia, A. F. Martín, J. Principe, Multilevel Balancing Domain Decomposition at Extreme Scales, SIAM Journal on Scientific Computing 38 (1) (2016) C22–C52. doi:10.1137/15m1013511.

[9] A. Klawonn, O. Rheinbach, Robust FETI-DP methods for heterogeneous three dimensional elasticity problems, Computer Methods in Applied Mechanics and EngineerinComputer Methods in Applied Mechanics and Engineering 196 (8) (2007) 1400–1414. doi:10.1016/j.cma.2006.03.023.

[10] A. Klawonn, O. Rheinbach, O. B. Widlund, An analysis of a FETI–DP algorithm on irregular subdomains in the plane, SIAM J. Numer. Anal. 46 (5) (2008) 2484–2504. doi:10.1137/070688675.

[11] C. Pechstein, R. Scheichl, Analysis of FETI methods for multiscale PDEs. Part II: interface variation, Numer. Math. 118 (3) (2011) 485–529. doi:10.1007/s00211-011-0359-2.

[12] 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) (2007) 1389–1399. doi:10.1016/j.cma.2006.03.010.

[13] N. Spillane, D. J. Rixen, Automatic spectral coarse spaces for robust FETI and BDD algorithms, International Journal for Numerical Methods in Engineering 95 (11) (2013) 953–990. doi:10.1002/nme.4534.

[14] 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) (2014) 741–770. doi:10.1007/s00211-013-0576-y.

[15] B. Sousedík, J. Šístek, J. Mandel, Adaptive-Multilevel BDDC and its parallel implementation, Computing 95 (12) (2013) 1087–1119. doi:10.1007/s00607-013-0293-5.

[16] A. Klawonn, M. Kühn, O. Rheinbach, Adaptive coarse spaces for FETI-DP in three dimensions, SIAM J. Sci. Comput. 38 (5) (2016) A2880–A2911. doi:10.1137/15M1049610.
URL http://dx.doi.org/10.1137/15M1049610

[17] J. G. Calvi, O. B. Widlund, An adaptive choice of primal constraints for BDDC domain decomposition algorithms, Tech. Rep. TR2015-979, Courant Instute, New York University (2015). doi:10.1007/3-540-26662-3_5.

[18] R. Bridson, C. Greif, A multipreconditioned conjugate gradient algorithm, SIAM J. Matrix Anal. Appl. 27 (4) (2006) 1056–1068 (electronic). doi:10.1137/040620047.

[19] C. Greif, T. Rees, D. B. Szyld, MPGMRES: a generalized minimum residual method with multiple preconditioners, Tech. Rep. 11-12-23, Department of Mathematics, Temple University, revised September 2012 and January 2014. Also available as Technical Report TR-2011-12, Department of Computer Science, University of British Columbia (2011). doi:10.1007/s40324-016-0088-7.

[20] C. Greif, T. Rees, D. B. Szyld, Additive Schwarz with Variable Weights, Tech. Rep. 12-11-30, Department of Mathematics, Temple University (Nov. 2012). doi:10.1007/978-3-319-05789-7_75.

[21] N. Spillane, Algebraic Adaptive Multipreconditioning applied to Restricted Additive Schwarz, Accepted into the proceedings of the 23rd International Conference on Domain Decomposition Methods (2016).
URL https://hal.archives-ouvertes.fr/hal-01276446v2/document

[22] C. Farhat, F.-X. Roux, A method of finite element tearing and interconnecting and its parallel solution algorithm, International Journal for Numerical Methods in Engineering 32 (6) (1991) 1205. doi:10.1002/nme.1620320604.

[23] D. J. Rixen, C. Farhat, R. Tezaur, J. 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) (1999) 501–533. doi:10.1002/(SICI)1097-0207(19991010)46:4<501::AID-NME685>3.0.CO;2-7.

[24] D. J. Rixen, C. 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) (1999) 489–516. doi:10.1002/(SICI)1097-0207(19990210)44:4<489::AID-NME514>3.0.CO;2-Z.

[25] J. Mandel, R. Tezaur, Convergence Of A Substructuring Method With Lagrange Multipliers, Numerische Mathematik 73 (1994) 473–487. doi:10.1007/s002110050201.

[26] A. Klawonn, O. Widlund, FETI and Neumann-Neumann iterative substructuring methods: Connections and new results, Communications on Pure and Applied Mathematics 54 (1) (2001) 57–90. doi:10.1002/1097-0312(200101)54:1<57::AID-CPA3>3.0.CO;2-D.

[27] A. Toselli, O. Widlund, Domain decomposition methods—algorithms and theory, Vol. 34 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2005. doi:10.1007/b137868.

[28] P. Gosselet, C. Rey, 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) (2013) 60–83. doi:10.1002/nme.4441.

[29] C. Paige, Approximate solutions and eigenvalue bounds from Krylov subspaces, Num. Lin. Algebra Appl. 2 (2) (1995) 115–133. doi:10.1002/nla.1680020205.

[30] Transvalor S.A., Z-set 8.6 user manual (2015).
URL http://www.zset-software.com/

[31] P. Amestoy, I. Duff, J.-Y. L'Excellent, Multifrontal parallel distributed symmetric and unsymmetric solvers, Computer Methods in Applied

Mechanics and Engineering 184 (2–4) (2000) 501–520. doi:10.1016/S0045-7825(99)00242-X.

[32] C. Farhat, M. Géradin, On the general solution by a direct method of a large-scale singular system of linear equations: application to the analysis of floating structures, International Journal for Numerical Methods in Engineering 41 (4) (1998) 675–696. doi:10.1002/(SICI)1097-0207(19980228)41:4<675::AID-NME305>3.0.CO;2-8.

[33] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1) (1998) 359–392. doi:10.1137/S1064827595287997.

[34] J. Schneider, G. Hello, Z. Aboura, M. Benzeggagh, D. Marsal, A Meso-FE voxel model of an interlock woven composite, in: 17th International Conference on Composite Materials (ICCM-17), Edimburgh, Scotland, 2009.

[35] Y. Wielhorski, D. Durville, Finite element simulation of a 3d woven fabric: determination of the initial configuration and characterization of the mechanical behavior, in: 13th International conference on textile composites (TexComp-12), Rayleigh, USA, 2015.

[36] G. Hello, J. Schneider, Z. Aboura, Generation of Voxel-FE Models for Complex 3D Composite Architectures, in: 8th European Solid Mechanics Conference (ESMC-2012), Graz, Austria, 2012.