HAL open science

Simultaneous-FETI and Block-FETI: robust domain decomposition with multiple search directions.

Pierre Gosselet, Daniel Rixen, François-Xavier Roux, Nicole Spillane

To cite this version:

Pierre Gosselet, Daniel Rixen, François-Xavier Roux, Nicole Spillane. Simultaneous-FETI and Block-FETI: robust domain decomposition with multiple search directions.. International Journal for Numerical Methods in Engineering, 2015, 104 (10), pp.905-927. 10.1002/nme.4946 . hal-01056928v2

HAL Id: hal-01056928

https://hal.science/hal-01056928v2

Submitted on 3 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Simultaneous-FETI and Block-FETI: robust domain decomposition with multiple search directions.

Pierre Gosselet[1] – Daniel Rixen[2] – François-Xavier Roux [3] – Nicole Spillane [4]

[1] LMT-Cachan / ENS-Cachan, CNRS, U. Paris Saclay,
61 avenue du président Wilson, 94235 Cachan, France.

[2] Technische Universität München, Faculty of Mechanics,
Institute of Applied Mechanics, Boltzmannstr. 15, 85748 Garching, Germany.

[3] Laboratoire Jacques-Louis Lions, CNRS UMR 7598,
Université Pierre et Marie Curie, 75005 Paris, France.

[4] Center for Mathematical Modeling, Facultad de Ciencias Físicas y Matemáticas,
Beauchef 851, Universidad de Chile, Santiago, Chile.

March 1, 2015

Abstract

Domain Decomposition methods often exhibit very poor performance when applied to engineering problems with large heterogeneities. In particular for heterogeneities *along* domain interfaces the iterative techniques to solve the interface problem are lacking an efficient preconditioner. Recently a robust approach, named FETI-Geneo, was proposed where troublesome modes are precomputed and deflated from the interface problem. The cost of the FETI-Geneo is however high. We propose in this paper techniques that share similar ideas with FETI-Geneo but where no pre-processing is needed and that can be easily and efficiently implemented as an alternative to standard Domain Decomposition methods. In the block iterative approaches presented in this paper, the search space at every iteration on the interface problem contains as many directions as there are domains in the decomposition. Those search directions originate either from the domain-wise preconditioner (in the Simultaneous FETI method) or from the block structure of the right-hand side of the interface problem (Block FETI). We show on 2D structural examples that both methods are robust and provide good convergence in the presence of high heterogeneities, even when the interface is jagged or when the domains have a bad aspect ratio. The Simultaneous FETI was also efficiently implemented in an optimized parallel code and exhibited excellent performance compared to the regular FETI method.

Keywords: Domain decomposition; FETI; BDD; Block Krylov methods; multiple preconditioner; heterogeneity.

1 Introduction

Domain decomposition methods are mature solution techniques to enable computing the solution of large systems (typically arising from Finite Element models) on parallel computers. In particular the non-overlapping techniques such as the Finite Element Tearing and Interconnection (FETI) [17] and its primal counterpart (the Balanced Domain Decomposition or BDD [31]) have been successfully applied to solve several challenging mechanical problems (e.g. [2,3]). The fundamental idea behind these efficient parallel solvers consists in solving local problems related to each domain with techniques that perform well sequentially on one processor and applying iterative techniques to find the interface unknowns connecting domains together, namely the interface forces in the dual Schur complement approaches (such as FETI) or interface displacements in the primal Schur complement methods (such as BDD). Several variants of the primal and dual strategies in Domain Decomposition have been developed over the years to improve their robustness and efficiency: an overview can be found for instance in [20].

Nevertheless, for engineering problems where the structure is composed of parts with intricate shapes and/or made of materials with very different properties, domain decomposition techniques usually perform very poorly. A typical example one could mention is tires where the bulk is composed of soft rubber material in which different very stiff and slender components are embedded (steel cables and thin sheets of fiber reinforced composites). A remedy could be to decompose these hard problems in such a way that each domain is nearly homogeneous [2]. Indeed a number of contributions provide ways to solve problems with heterogeneities across the interfaces [7, 8, 10, 16, 29, 30, 32, 45] or inside the subdomains and not near their boundaries [19, 39, 40]. This approach often results in bad load balancing and in domains with bad aspect ratios, which also pose a challenge for solving the interface problem iteratively. In the case of the Darcy equation, the authors in [37, 38, 41] prove convergence results, even in some cases where the jumps in the coefficients are along the interfaces. Within a class of coefficient distributions, the theory, which is based on weighted

Poincare inequalities [42], allows to discriminate between jumps in the coefficients which will affect convergence and those that will not. A typical case which cannot be solved efficiently with the, by now *classical*, techniques is shown in our numerical section (Figure 1). A significant contribution in achieving robustness, with much ongoing research, is the deluxe scaling (see [1] and references therein) which proposes to use near optimal non diagonal scaling. Another strategy, which the present work builds on is presented next.

As was described in recent publications ([9, 11, 18, 34, 51] for the overlapping Additive Schwarz method and [27, 28, 33, 49, 52] for non-overlapping methods) the bad convergence of domain decomposition strategies in many challenging engineering problems can be traced back to the fact that important characteristics of the global problem cannot be approximated by the local information typically used to precondition the iterations on the interface problem. It was shown that the part of the problem that jeopardizes convergence can be revealed by eigenvalue problems on the interface of each subdomain: the strategy is to generate these problematic modes and apply deflation strategies (or coarse space approaches) to guarantee that the iterations are performed only on the part of the space that can be properly preconditioned. Those methods, given the generic name *Geneo* (Generalized Eigenvalues in the Overlaps) in [50–52], exhibit remarkable robustness, both in theory and practice, even for decompositions where the domains have bad aspect ratios and where large heterogeneities across and along the interface are present (see Section 2 for more detail). Unfortunately that robustness comes with a significant computational overhead related to finding the coarse space of 'bad' modes through eigenvalue problems.

The objective of the present contribution is to propose robust FETI type solvers that do not require a preprocessing step. We present two algorithms: the Simultaneous FETI (or S-FETI) algorithm and the Block FETI (or B-FETI) algorithm. The S-FETI algorithm was originally proposed in [44] for two subdomains and generalized in [47] to more subdomains. S-FETI consists in generating several search directions originating from the preconditioner and using them to solve the interface problem iteratively. A similar paradigm was later used in [23] to solve problems that include repeated components. The previous contributions [44, 47] had discussed the potential of the method on some geometries but showed bad computational efficiency due to a crude and naive implementation of the algorithm. One major contribution of this paper, in addition to placing S-FETI in the context of FETI with the Geneo coarse space, B-FETI and multi preconditioned conjugate gradient algorithms [24], is to propose an efficient organization of the algorithm such that it also becomes computational highly efficient. The B-FETI algorithm is a Block CG method [36]. The initialization procedure is new and allows to rewrite the FETI linear system in block form in such a way that we update the solution at the same time as we compute Krylov subspaces that are very closely related to the Lanczos spaces for the computation of the local eigenvectors responsible for slow convergence. For this reason, the fundamental advantage over the usual FETI algorithm is that we are now able to *catch* several isolated eigenvalues within one iteration. We propose a conjecture in equation (14) which attempts to draw a connection between the convergence of Block-FETI and the convergence of the (robust and scalable) FETI algorithm with the Geneo coarse space [52]. Numerical experiments will confirm that both the B-FETI and S-FETI algorithms have similar convergence behaviors that are also similar to the convergence of FETI with the Geneo coarse space.

The outline of the article is as follows. In Section 2 we give a short summary of the FETI method as well as the Geneo coarse space since they form the basis for the proposed strategy. Then, Sections 3 and 4 are dedicated to explaining S-FETI and B-FETI respectively. Finally, the methods are tested and evaluated in Section 5 for some simple but representative problems including some comparisons in CPU time between S-FETI and the classical FETI.

Note that, although only the FETI approaches are discussed in this paper, the ideas presented in this contribution can be extended in a straightforward manner to the other variants of FETI (such as the FETI-DP method [13, 15]) or to primal Schur complement methods such as BDD [31] and BDDC [6].

2 FETI in a nutshell

To begin, we shortly summarize the basic FETI strategy that will be used throughout this paper (see for instance [17,20] for further details). Let us consider the symmetric positive definite problem $\mathbf{K}\mathbf{u} = \mathbf{f}$ associated with the finite element approximation of a linear mechanical problem set on domain Ω. Assume a partitioning into N subdomains $\Omega^{(s)}$ conforming to the mesh such that the partitioned problem writes

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

where $\mathbf{t}^{(s)}$ are trace operators, $\mathbf{B}^{(s)}$ are signed Boolean assembly operators and $\boldsymbol{\lambda}$ is the set of Lagrange multipliers that connect subdomains.

We use the following classical notations:

$$\mathbf{S}^{(s)} = \mathbf{K}_{bb}^{(s)} - \mathbf{K}_{bi}^{(s)} {\mathbf{K}_{ii}^{(s)}}^{-1} \mathbf{K}_{ib}^{(s)}; \quad \mathbf{F}^{(s)} = \mathbf{t}^{(s)} {\mathbf{K}^{(s)}}^{+} {\mathbf{t}^{(s)}}^T; \quad \mathbf{R}^{(s)} = \ker(\mathbf{K}^{(s)})$$

where $\mathbf{S}^{(s)}$ is the local Schur complement (i stands for internal degrees of freedom and b for boundary degrees of

freedom), $\mathbf{F}^{(s)} = (\mathbf{S}^{(s)})^+$ is the local Dual Schur complement and $\mathbf{R}^{(s)}$ is a basis of rigid body modes. We also write:

$$\begin{aligned} \mathbf{e} &= -\left(\dots, \mathbf{f}^{(s)^T}\mathbf{R}^{(s)}, \dots\right)^T & \mathbf{G} &= \left(\dots, \mathbf{B}^{(s)}\mathbf{t}^{(s)}\mathbf{R}^{(s)}, \dots\right) \\ \mathbf{F} &= \sum \mathbf{B}^{(s)}\mathbf{F}^{(s)}\mathbf{B}^{(s)^T} & \mathbf{d} &= -\sum_s \mathbf{B}^{(s)}\mathbf{t}^{(s)}\mathbf{K}^{(s)^+}\mathbf{f}^{(s)} \end{aligned} \tag{2}$$

which leads to the classical FETI system:

$$\begin{pmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \alpha \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{e} \end{pmatrix}. \tag{3}$$

The constraint $\mathbf{G}^T\boldsymbol{\lambda}$ is handled by the introduction of the initial estimate and the projector

$$\begin{aligned} \boldsymbol{\lambda}_0 &= \mathbf{A}\mathbf{G}(\mathbf{G}^T\mathbf{A}\mathbf{G})^{-1}\mathbf{e} \\ \mathbf{P} &= \mathbf{I} - \mathbf{A}\mathbf{G}(\mathbf{G}^T\mathbf{A}\mathbf{G})^{-1}\mathbf{G}^T \end{aligned} \tag{4}$$

so that $\mathbf{G}^T\boldsymbol{\lambda}_0 = \mathbf{e}$ and $\mathbf{G}^T\mathbf{P} = \mathbf{0}$. Matrix $\mathbf{A}$ is a symmetric positive definite matrix and can be taken as being the preconditioner $\tilde{\mathbf{S}}$ (see below), identity or a scaling matrix [46].

The unknown $\boldsymbol{\lambda}$ is sought as $\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \mathbf{P}\tilde{\boldsymbol{\lambda}}$ where $\tilde{\boldsymbol{\lambda}}$ is a solution of:

$$\mathbf{P}^T\mathbf{F}\mathbf{P}\tilde{\boldsymbol{\lambda}} = \mathbf{P}^T\left(\mathbf{d} - \mathbf{F}\boldsymbol{\lambda}_0\right) = \mathbf{P}^T\left(\sum_s \mathbf{B}^{(s)}\mathbf{t}^{(s)}\mathbf{K}^{(s)^+}(\mathbf{f}^{(s)} - \mathbf{t}^{(s)^T}\mathbf{B}^{(s)^T}\boldsymbol{\lambda}_0)\right). \tag{5}$$

This system is solved by an iterative solver, the preconditioner $\tilde{\mathbf{S}}$ being

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

where $\tilde{\mathbf{B}}^{(s)}$ are scaled assembling operators such that $\sum_s \mathbf{B}^{(s)}\tilde{\mathbf{B}}^{(s)^T} = \mathbf{I}$, and $\tilde{\mathbf{S}}^{(s)}$ are the Schur complements $\mathbf{S}^{(s)}$ or an approximation thereof. The scaling used in $\tilde{\mathbf{B}}^{(s)}$ is typically chosen based on the diagonal coefficients of the local stiffness matrices on the interface, namely a so-called k-scaling or super-lumped scaling. As explained in [45] this scaling can be seen as choosing a mechanically consistent combination of the interface reaction forces arising from the Dirichlet problem in each subdomain.

It is by now quite standard to augment the resolution by an additional constraint of the form $\mathbf{C}^T\mathbf{r} = 0$ where matrix $\mathbf{C}$ is a basis of a well-chosen subspace of range($\mathbf{P}$) and $\mathbf{r}$ is the residual vector. This means that at every iteration the solution is required to solve the problem exactly in the subspace spanned by $\mathbf{C}$, or coarse space. When the constraint is implemented with a second level of initialization and projection, the resulting algorithm is referred to as FETI2 [12, 14]. It is summarized in Algorithm 1, where we introduced:

$$\begin{aligned} \tilde{\boldsymbol{\lambda}}_0 &= \mathbf{C}(\mathbf{C}^T\mathbf{F}\mathbf{C})^{-1}\mathbf{C}^T(\mathbf{d} - \mathbf{F}\boldsymbol{\lambda}_0) \\ \mathbf{P}_\mathbf{C} &= \mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{F}\mathbf{C})^{-1}\mathbf{C}^T\mathbf{F}. \end{aligned} \tag{6}$$

Remark 1. In Algorithm 1, a full orthogonalization is employed as it is almost required when solving real engineering problems. A classical Conjugate Gradient algorithm would correspond to $j = i$ (instead of $0 \leqslant j \leqslant i$) in the second last line of the loop.

Since the spectrum of the preconditioned FETI operator is bounded from below by 1 it is well known that matrix $\mathbf{C}$ should contain the eigenvectors $\mathbf{v}$ associated with the largest eigenvalues μ of the following generalized eigenvalue problem:

$$\mathbf{P}^T\left(\mathbf{F}\mathbf{v} - \mu\tilde{\mathbf{S}}^{-1}\mathbf{v}\right) = 0, \quad \mathbf{v} \in \text{range}(\mathbf{P}). \tag{7}$$

Various techniques can be employed to approximate these eigenvectors, like recycling of nearby Krylov subspaces [21]. One important result is provided by [52] where it is proved that the space spanned by the high frequency eigenvectors can be approximated by solving (in parallel) a family of generalized eigenvalue problems. More precisely, in [52] it is proposed to compute the eigenpairs $(\mu_k^{(s)}, \mathbf{v}_k^{(s)})$ indexed by k of

$$\mathbf{S}^{(s)}\mathbf{v}_k^{(s)} - \mu_k^{(s)}\mathbf{B}^{(s)^T}\tilde{\mathbf{S}}\mathbf{B}^{(s)}\mathbf{v}_k^{(s)} = 0 \text{ for each } s = 1, \dots, N, \tag{8}$$

and then define the columns of the constraint matrix $\mathbf{C}$ as $\{\mathbf{P}\tilde{\mathbf{S}}\mathbf{B}^{(s)^T}\mathbf{v}_k^{(s)};\ 0 < \mu_k^{(s)} < \tau;\ s = 1, \dots, N\}$ for some chosen threshold $\tau \geq 0$. This is the FETI-Geneo algorithm and it is guaranteed theoretically that the largest eigenvalue of the projected preconditioned operator is bounded by $\mathcal{N}\tau$ where $\mathcal{N}$ is the number of neighbors of a subdomain. In particular this estimate does not depend on the number of subdomains (scalability) or the difficulty of the problem (robustness). The only additional assumption is that the matrix $\mathbf{A}$ in (4) correspond to the choice $\mathbf{A} = \tilde{\mathbf{S}}$. Unfortunately, the solution to these eigenproblems followed by an augmented resolution incurs a significant computational overhead and the present article proposes a cheap, yet effective, alternative.

Algorithm 1: FETI2 with full orthogonalization

$\mathbf{r}_0 = \mathbf{P}^T {\mathbf{P_C}}^T(\mathbf{d} - \mathbf{F}\boldsymbol{\lambda}_0)$
$\mathbf{z}_0 = \tilde{\mathbf{S}}\mathbf{r}_0$, $\mathbf{w}_0 = \mathbf{P}\mathbf{z}_0$, $\hat{\boldsymbol{\lambda}}_0 = 0$, $i = 0$
while $\sqrt{\mathbf{r}_i^T \mathbf{z}_i} > \epsilon$ **do**
  $\mathbf{q}_i = {\mathbf{P_C}}^T \mathbf{F}\mathbf{w}_i$
  $\delta_i = \mathbf{q}_i^T \mathbf{w}_i$
  $\gamma_i = \mathbf{r}_i^T \mathbf{z}_i$
  $\hat{\boldsymbol{\lambda}}_{i+1} = \hat{\boldsymbol{\lambda}}_i + (\gamma_i/\delta_i)\mathbf{w}_i$
  $\mathbf{r}_{i+1} = \mathbf{r}_i - (\gamma_i/\delta_i)\mathbf{P}^T\mathbf{q}_i$
  $\mathbf{z}_{i+1} = \tilde{\mathbf{S}}\mathbf{r}_{i+1}$
  $\mathbf{w}_{i+1} = \mathbf{P}\mathbf{z}_{i+1}$ then for $0 \leqslant j \leqslant i$ $\begin{cases} \phi_{i,j} = \mathbf{q}_j^T \mathbf{w}_{i+1} \\ \mathbf{w}_{i+1} \leftarrow \mathbf{w}_{i+1} - (\phi_{i,j}/\delta_j)\mathbf{w}_j \end{cases}$
  $i \leftarrow i + 1$
end
$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \tilde{\boldsymbol{\lambda}}_0 + \mathbf{P_C}\hat{\boldsymbol{\lambda}}_i$

Remark 2. As already mentioned in the introduction there is in fact a variety of coarse spaces that are constructed by solving generalized eigenvalues problems [9, 11, 18, 27, 28, 33, 34, 49, 51]. More specifically the FETI-Geneo eigenvalue problem (8) is very strongly connected to the one that was previously proposed in [33, 49] for FETI-DP and BDDC. The difference is that, there, the eigenvalue problems are defined per interface between two subdomains whereas the Geneo problem involves a subdomain and all its neighbors (second term in (8)). Moreover the choice of eigenvalue problem relies on some heuristic assumptions. Also related to this family of methods, in [50] an algorithm (named frugal-FETI) was sketched in order to capture the local contributions that penalize convergence within the conjugate gradient algorithm.

The strategy proposed next, generalizing the multi-direction approach discussed in [44], can be considered as probably the simplest block procedure to build a multi-direction iteration and is thought to capture "on the fly" the bad modes typically constructed a priori in Geneo approaches.

3 Simultaneous FETI

The Simultaneous FETI (S-FETI) algorithm was introduced in [44] on a simple example with two subdomains. It enhances robustness by exploiting the additive structure of the preconditioner in order to generate as many search directions as there are subdomains at each step of the Conjugate Gradient (CG) algorithm. We first present the algorithm in a general case (Algorithm 2) as well as the ideas that led to it. Then we discuss its connections with two existing algorithms: multipreconditioned CG and FETI Geneo and why it is expected to be robust. Finally we comment on the cost of S-FETI and introduce an important trick (see equation (11)) to localize the application of the FETI operator which makes our algorithm a lot more competitive.

3.1 The S-FETI Algorithm

In classical FETI, the preconditioned residual writes $\mathbf{z} = \tilde{\mathbf{S}}\mathbf{r} = \sum_s \tilde{\mathbf{B}}^{(s)}\tilde{\mathbf{S}}^{(s)}{\tilde{\mathbf{B}}^{(s)}}^T\mathbf{r}$ and it is orthogonalized with respect to previous search directions to generate the new search direction. The idea underlying the S-FETI approach consists in letting the minimization process of the CG algorithm choose the best combination of local terms $\tilde{\mathbf{B}}^{(s)}\tilde{\mathbf{S}}^{(s)}{\tilde{\mathbf{B}}^{(s)}}^T\mathbf{r}$ (instead of simply adding them together to obtain $\mathbf{z}$) hence leading to an optimal, although more costly, choice. In other words, S-FETI, at a given iteration, uses each local term $\tilde{\mathbf{B}}^{(s)}\tilde{\mathbf{S}}^{(s)}{\tilde{\mathbf{B}}^{(s)}}^T\mathbf{r}$ as a search direction: the residual is minimized with respect to the subspace spanned by $\mathbf{Z} = \left(\ldots, \tilde{\mathbf{B}}^{(s)}\tilde{\mathbf{S}}^{(s)}{\tilde{\mathbf{B}}^{(s)}}^T\mathbf{r}, \ldots\right)$. The connection between $\mathbf{Z}$ and the usual $\mathbf{z}$ is of course that $\mathbf{z} = \mathbf{Z}\mathbf{1}$ where $\mathbf{1} = (1, \ldots, 1)^T \in \mathbb{R}^N$ which explains why we monitor convergence in a classical way ($\sqrt{\mathbf{r}^\top\mathbf{z}}$ gives a measure of the residual comparable to the discretization error [43]).

The iteration scheme is given in Algorithm 2. For sake of clarity we give the size of the different operators

$$\mathbf{r}_i,\ \tilde{\boldsymbol{\lambda}}_i,\ \boldsymbol{\lambda},\ \boldsymbol{\lambda}_0 \in \mathbb{R}^n; \quad \mathbf{Z}_i,\ \mathbf{W}_i,\ \mathbf{Q}_i \in \mathbb{R}^{n\times N}; \quad \boldsymbol{\Delta}_i,\ \boldsymbol{\Phi}_{i,j} \in \mathbb{R}^{N\times N}; \quad \boldsymbol{\gamma}_i \in \mathbb{R}^N \quad ;$$

where n is the number of unknowns and N is again the number of subdomains.

We point out that each iteration requires the inversion of the $N \times N$ matrix $\boldsymbol{\Delta}_i = \mathbf{W}_i^T\mathbf{F}\mathbf{W}_i$. Since $\mathbf{W}_i$ is the concatenation of localized contributions, it is reasonable to expect that $\boldsymbol{\Delta}_i$ be full-ranked. If it is not then $\boldsymbol{\Delta}_i$ is only positive semi definite and pseudo-inversion (denoted by $\boldsymbol{\Delta}_i^+$) is necessary. Another, equivalent, option would be to eliminate some directions in order to recover a full rank family of vectors in $\mathbf{W}_i$. Then the next approximate

Algorithm 2: Simultaneous FETI

$\mathbf{r}_0 = \mathbf{P}^T(\mathbf{d} - \mathbf{F}\boldsymbol{\lambda}_0)$

$\mathbf{Z}_0 = \left(\ldots, \tilde{\mathbf{B}}^{(s)}\tilde{\mathbf{S}}^{(s)}\tilde{\mathbf{B}}^{(s)^T}\mathbf{r}_0, \ldots\right)$, $\mathbf{W}_0 = \mathbf{P}\mathbf{Z}_0$, $\tilde{\boldsymbol{\lambda}}_0 = 0$, $i = 0$

while $\sqrt{\mathbf{r}^T\mathbf{Z}\mathbf{1}} > \epsilon$ **do**

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

  $\boldsymbol{\Delta}_i = \mathbf{Q}_i^T\mathbf{W}_i$

  $\boldsymbol{\gamma}_i = \mathbf{Z}_i^T\mathbf{r}_i$

  $\tilde{\boldsymbol{\lambda}}_{i+1} = \tilde{\boldsymbol{\lambda}}_i + \mathbf{W}_i\boldsymbol{\Delta}_i^+\boldsymbol{\gamma}_i$

  $\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{P}^T\mathbf{Q}_i\boldsymbol{\Delta}_i^+\boldsymbol{\gamma}_i$

  $\mathbf{Z}_{i+1} = \left(\ldots, \tilde{\mathbf{B}}^{(s)}\tilde{\mathbf{S}}^{(s)}\tilde{\mathbf{B}}^{(s)^T}\mathbf{r}_{i+1}, \ldots\right)$

  $\mathbf{W}_{i+1} = \mathbf{P}\mathbf{Z}_{i+1}$ then for $0 \leqslant j \leqslant i$ $\begin{cases} \boldsymbol{\Phi}_{i,j} = \mathbf{Q}_j^T\mathbf{W}_{i+1} \\ \mathbf{W}_{i+1} \leftarrow \mathbf{W}_{i+1} - \mathbf{W}_j\boldsymbol{\Delta}_j^+\boldsymbol{\Phi}_{i,j} \end{cases}$

  $i \leftarrow i + 1$

end

$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \tilde{\boldsymbol{\lambda}}_i$

solution would not change but fewer vectors would need to be saved for future orthogonalization. In any case the right-hand-side $\boldsymbol{\gamma}_i$ and range($\boldsymbol{\Phi}_{i,j}$) are both in range($\mathbf{W}_i^T$) = range($\boldsymbol{\Delta}_i$) so that the iteration is always well defined.

Remark 3. More precisely, we propose the following procedure to compute the (pseudo)-inverse of $\boldsymbol{\Delta}_i$ and simplify the subsequent orthogonalization steps. A rank-revealing Cholesky factorization (symmetric pivoting) is employed just after $\boldsymbol{\Delta}_i$ is computed:

$$\mathbf{N}\boldsymbol{\Delta}_i\mathbf{N}^T = \mathbf{L}\mathbf{L}^T, \qquad \text{with } \mathbf{L} = \begin{pmatrix} \tilde{\mathbf{L}} & 0 \\ \times & 0 \end{pmatrix} \qquad \begin{array}{l} \mathbf{N}: \text{ permutation matrix} \\ \tilde{\mathbf{L}}: \text{ lower triangular matrix} \end{array} \tag{9}$$

The search directions can then be $\mathbf{F}$–orthonormalized and redundant directions suppressed by setting:

$$\mathbf{W}_i \leftarrow \mathbf{W}_i\mathbf{N}^T\begin{pmatrix} \tilde{\mathbf{L}}^{-T} \\ 0 \end{pmatrix} \qquad \mathbf{Q}_i \leftarrow \mathbf{Q}_i\mathbf{N}^T\begin{pmatrix} \tilde{\mathbf{L}}^{-T} \\ 0 \end{pmatrix} \qquad \boldsymbol{\Delta}_i \leftarrow \mathbf{I}$$

In that case $\boldsymbol{\gamma}_i$ must be evaluated by the formula $\boldsymbol{\gamma}_i = \mathbf{W}_i^T\mathbf{r}_i$.

3.2 S-FETI as a multipreconditioned CG algorithm: minimization property

The S-FETI algorithm is a multipreconditioned CG algorithm [5]. Indeed, at each iteration, N preconditioners are applied to generate the N columns in $\mathbf{Z}_i$. Then, each of these vectors is projected and orthogonalized to give a search direction (these are the N columns in $\mathbf{W}_i$). Finally, $\tilde{\boldsymbol{\lambda}}_i$ is updated to $\tilde{\boldsymbol{\lambda}}_{i+1}$ by adding the linear combination $\mathbf{W}_i\boldsymbol{\Delta}_i^+\boldsymbol{\gamma}_i$ of these search directions that minimizes the error in the operator norm. Since the classical search direction $\mathbf{w} = \mathbf{W}\mathbf{1} \in$ range($\mathbf{W}$) the new approximation is obviously always better than what classical CG would have given at that iteration.

A negative point of the method is that the CG short recurrence is broken and full orthogonalization is required to fully benefit from the method. This is however only a theoretical drawback since, in practice, full orthogonalization is often used in classical FETI. The conjugacy ($\mathbf{F}$–orthogonality between search directions) is crucial since it ensures the following minimization property as proved in [5].

Theorem 1. *The approximate solution computed by the i-th iteration of S-FETI minimizes the error $\tilde{\boldsymbol{\lambda}}_i - P\tilde{\boldsymbol{\lambda}}$ in the $\mathbf{F}$-norm (induced by the FETI operator) over all possible*

$$\tilde{\boldsymbol{\lambda}}_i \in \bigoplus_{j=0}^{i-1} \operatorname{span}(\mathbf{W}_j), \tag{10}$$

where $\oplus$ indicates a direct sum and $\mathbf{W}_j$ is defined in algorithm 2.

The proof can be written in a similar way to the usual proofs for CG [48]. The two properties of multipreconditioned CG which condition the choice of $\boldsymbol{\Delta}_i$, $\boldsymbol{\gamma}_i$ and $\boldsymbol{\Phi}_{i,j}$ are this minimization result, and the $\mathbf{F}$ conjugacy of search directions ($\mathbf{W}_i^\top\mathbf{F}\mathbf{W}_j = 0, \quad \forall\, i \neq j$).

A particularity of multipreconditioned CG algorithms is that the minimization space is not a Krylov subspace. The reason is that at each iteration the approximate solution is updated in the direction given by the optimal linear

combination of all preconditioners but the coefficients (given by $\mathbf{\Delta}_i^+\boldsymbol{\gamma}_i \in \mathbb{R}^N$) in the linear combination change from one iteration to the next. For this reason we cannot justify a heuristic bound for the number of iterations as we will in (14) for block FETI. We will however observe in Section 5 that on all test cases both solvers behave in a similar way. This can be intuited by the fact that the search spaces are the same size ($N \times i$) and constructed with a combination of local and global components coming from the operator and the preconditioner. A strong connection between S-FETI and FETI-Geneo can also be drawn which justifies why we expect such good robustness.

3.3 Connection to FETI Geneo: why robustness is expected

As will become apparent in Section 5, the S-FETI algorithm is very efficient on hard problems for which the classical FETI typically requires many iterations. The convergence behavior is comparable to that of the FETI Geneo algorithm [52] where a coarse space is constructed by solving in each subdomain a generalized eigenvalue problem (8) that isolates the part of the solution on which the preconditioner is not sufficiently efficient for the iterative solver to perform well. More precisely, the matrices in the pencil of the Geneo eigenproblems are on one hand $\mathbf{B}^{(s)^T}\tilde{\mathbf{S}}\mathbf{B}^{(s)}$ and on the other $\mathbf{S}^{(s)}$. With words, the vectors that are detected are the ones for which the local restriction of the (assembled) preconditioner $\mathbf{B}^{(s)^T}\tilde{\mathbf{S}}\mathbf{B}^{(s)}$ is not a good approximation for the, non assembled, local component $\mathbf{S}^{(s)}$ of the FETI operator. In S-FETI the solution space results from successive applications of the local, non assembled, components $\tilde{\mathbf{B}}^{(s)}\mathbf{S}^{(s)}\tilde{\mathbf{B}}^{(s)^T}$ and the assembled $\mathbf{F}$ so the block of search directions spans a space where local effects are not gummed out. It thus bears similarities with the deflated space in which the Geneo iterations take place and for this reason convergence is expected to be very quick.

Remark 4. Although our numerical results (Section 5) point to the fact that S-FETI performs perfectly well we mention the more recent multipreconditioned GMRES algorithm [24] where, at the cost of saving more directions at each iteration, the error can be minimized over the larger subspace $\sum_{s=1}^N \mathcal{K}_i^N(\mathbf{F}, \mathbf{P}\tilde{\mathbf{B}}^{(s)}\mathbf{S}^{(s)}\mathbf{r}_0)$, with the multi-Krylov subspace defined by

$$\mathcal{K}_i^N(\mathbf{F}, \mathbf{x}) := \left\{ p(..., \mathbf{P}\tilde{\mathbf{B}}^{(s)}\mathbf{S}^{(s)}\tilde{\mathbf{B}}^{(s)^T}\mathbf{F}, ...)\,\mathbf{x};\ \begin{array}{c} p \text{ is a polynomial in N variables} \\ \text{of degree at most } i-1 \end{array} \right\},$$

N being the number of subdomains. In [25] multiply preconditioned GMRES is applied to the Additive Schwarz domain decomposition technique.

3.4 Cost of S-FETI

An iteration of S-FETI does not require too much extra computational cost compared to classical FETI (for the discussion regarding parallelism we assume there is a bijection between the N subdomains and processors), for the following reasons

- Exchanges are as frequent. Neighbor communications are identical although global reduction operations (due to scalar products) involve more data ($N \times N$ matrices $\mathbf{\Delta}$ and $\mathbf{\Phi}$, N vector $\boldsymbol{\gamma}$).
- Dense, but usually small $N \times N$ symmetric positive semi definite matrices $\mathbf{\Delta}$ need to be (pseudo)-inverted.
- Sequences of N-blocks of vectors $\mathbf{W}_i$ and $\mathbf{Q}_i$ need to be stored instead of sequences of vectors.
- The most costly parts of the FETI algorithm are the local Neumann and Dirichlet solves in each subdomain. Compared to a classical FETI iteration, an S-FETI iteration does not require more work in the preconditioning step but the operator $\mathbf{F}$ must now be applied to each of the N columns in $\mathbf{W}_i$. Nevertheless the computation of $\mathbf{F}\mathbf{W}_i$ can be performed efficiently. First one has to remember that block operations are often proportionally much less expensive than single vector operations because the computation time is driven by the memory access. Moreover it is possible to cleverly use the locality of data by noting that $\mathbf{Z}_{i+1}$ is a sparse matrix (it only gets values from its neighbors and itself) whereas $\mathbf{W}_{i+1}$ is not because of projection and orthogonalization. Furthermore one observes that

$$\begin{aligned} \mathbf{Q}_{i+1} = \mathbf{F}\mathbf{W}_{i+1} &= \mathbf{F}\mathbf{P}\mathbf{Z}_{i+1} - \sum_{j=0}^{i} \mathbf{Q}_j \mathbf{\Delta}_j^+ \mathbf{\Phi}_{i,j} \\ &= \left(\mathbf{F}\mathbf{Z}_{i+1} - \mathbf{F}\mathbf{A}\mathbf{G}(\mathbf{G}^T\mathbf{A}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{Z}_{i+1}\right) - \sum_{j=0}^{i} \mathbf{Q}_j \mathbf{\Delta}_j^+ \mathbf{\Phi}_{i,j} \end{aligned} \tag{11}$$

  where the product $\mathbf{FAG}$ is sparse (only neighbors of neighbors contribute to it) and can be computed once and for all during the initialization. This way only localized Neumann problems $\mathbf{F}\mathbf{Z}_{i+1}$ need to be solved and the computational efficiency of S-FETI is significantly improved.

In the end, the extra-cost per iteration is expected to remain very limited for a not too large number of subdomains N.[1] It should also be noted that part of the additional cost is alleviated by the fact that the multiple Neumann problems to be solved by each domain at an iteration can be solved simultaneously as a block. The extra costs have to be put in balance with our expectancy to divide the number of iterations to solve critical problems by a term of the order of N. More details on the practical implementation of S-FETI are given in Section 5.8.

4 Block FETI

We present here the block FETI (B-FETI) technique. It is closely related to Simultaneous FETI with the advantage of preserving the short recurrence property. Indeed it is a block Conjugate Gradient algorithm (block CG) [36] where an initial block of right-hand sides is generated in order to activate the local effects. This particular choice of initialization is new and we will justify it later in connection with the theory of the Geneo coarse space.

We first introduce the Block FETI algorithm. Then we give the minimization property that is satisfied at each iteration and discuss the choice of initialization and why it is expected that it leads to fast convergence (in a way comparable to FETI-Geneo). Finally we comment on the cost of block FETI.

4.1 The block FETI algorithm

The original block CG algorithm was designed to simultaneously solve the same problem for a number of different right hand sides. Here we solve for a unique right hand side but we reformulate the dual interface problem (3) in block form by considering separately the contribution of each subdomain to the right hand side $\mathbf{d}$:

$$\mathbf{d} = \left(\dots, \mathbf{B}^{(s)}\mathbf{t}^{(s)}\mathbf{K}^{(s)^+}\mathbf{f}^{(s)}, \dots\right)\mathbf{1} = \left(\dots, \mathbf{B}^{(s)}\mathbf{d}^{(s)}, \dots\right)\mathbf{1}; \qquad \mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^N.$$

Based on this fact, and for a given initial guess[2] $\boldsymbol{\lambda}_{00} \in \mathbb{R}^n$, we choose the initial block residual as

$$\mathbf{R}_0 := \mathbf{P}^T\left(\dots, \mathbf{B}^{(s)}(\mathbf{d}^{(s)} - \mathbf{F}^{(s)}\mathbf{B}^{(s)^T}(\boldsymbol{\lambda}_0 + \mathbf{P}\boldsymbol{\lambda}_{00})), \dots\right). \tag{12}$$

The block FETI algorithm presented in Algorithm 3 is the block CG algorithm applied to the multiple right hand side problem $\mathbf{P}^T\mathbf{F}\mathbf{P}\tilde{\boldsymbol{\Lambda}} = \mathbf{R}_0$. For sake of clarity we give the size of the different operators

$$\boldsymbol{\lambda},\, \boldsymbol{\lambda}_0,\, \boldsymbol{\lambda}_{00} \in \mathbb{R}^n; \quad \tilde{\boldsymbol{\Lambda}}_i, \mathbf{Z}_i,\, \mathbf{W}_i,\, \mathbf{Q}_i \in \mathbb{R}^{n\times N}; \quad \Gamma_i,\, \boldsymbol{\Delta}_i,\, \boldsymbol{\Phi}_{i,j} \in \mathbb{R}^{N\times N};$$

where again n is the number of unknowns and N is the number of subdomains. The block system is connected to the original system by the relation $\mathbf{r}_0 = \mathbf{R}_0\mathbf{1} = \mathbf{P}^T\left(\sum_s \mathbf{B}^{(s)}(\mathbf{d}^{(s)} - \mathbf{F}^{(s)}\mathbf{B}^{(s)^T}(\boldsymbol{\lambda}_0 + \mathbf{P}\boldsymbol{\lambda}_{00})\right)$ so we can monitor the convergence of the original system within block FETI and the final solution is computed as $\boldsymbol{\lambda}_0 + \mathbf{P}\boldsymbol{\lambda}_{00} + \tilde{\boldsymbol{\Lambda}}_i\mathbf{1}$.

Algorithm 3: Block FETI with full orthogonalization (written as Block CG)

$\mathbf{R}_0 = \mathbf{P}^T\left(\dots, \mathbf{B}^{(s)}(\mathbf{d}^{(s)} - \mathbf{F}^{(s)}\mathbf{B}^{(s)^T}(\boldsymbol{\lambda}_0 + \mathbf{P}\boldsymbol{\lambda}_{00})), \dots\right)$
$\mathbf{Z}_0 = \tilde{\mathbf{S}}\mathbf{R}_0$, $\mathbf{W}_0 = \mathbf{P}\mathbf{Z}_0$, $\tilde{\boldsymbol{\Lambda}}_0 = 0$, $i = 0$
while $\sqrt{\mathbf{1}^T\mathbf{R}_i^T\mathbf{Z}_i\mathbf{1}} > \epsilon$ **do**
  $\mathbf{Q}_i = \mathbf{F}\mathbf{W}_i$
  $\boldsymbol{\Delta}_i = \mathbf{Q}_i^T\mathbf{W}_i$
  $\Gamma_i = \mathbf{R}_i^T\mathbf{Z}_i$
  $\tilde{\boldsymbol{\Lambda}}_{i+1} = \tilde{\boldsymbol{\Lambda}}_i + \mathbf{W}_i\boldsymbol{\Delta}_i^+\Gamma_i$
  $\mathbf{R}_{i+1} = \mathbf{R}_i - \mathbf{P}^T\mathbf{Q}_i\boldsymbol{\Delta}_i^+\Gamma_i$
  $\mathbf{Z}_{i+1} = \tilde{\mathbf{S}}\mathbf{R}_{i+1}$
  $\mathbf{W}_{i+1} = \mathbf{P}\mathbf{Z}_{i+1}$ then for $0 \leqslant j \leqslant i$ $\begin{cases} \boldsymbol{\Phi}_{i,j} = \mathbf{Q}_j^T\mathbf{W}_{i+1} \\ \mathbf{W}_{i+1} \leftarrow \mathbf{W}_{i+1} - \mathbf{W}_j\boldsymbol{\Delta}_j^+\boldsymbol{\Phi}_{i,j} \end{cases}$
  $i \leftarrow i + 1$
end
$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \mathbf{P}\boldsymbol{\lambda}_{00} + \tilde{\boldsymbol{\Lambda}}_i\mathbf{1}$

As for S-FETI, in each iteration we invert an $N \times N$ matrix $\boldsymbol{\Delta}_i$. In the algorithm we use the notation $^+$ to refer to the pseudo inverse of $\boldsymbol{\Delta}_i$. Once more, these pseudo inversions are well defined since range(Γ_i) and range($\boldsymbol{\Phi}_{i,j}$) are

[1] Note that in the case of two subdomains it was shown in [44] that the cost of one S-FETI iteration is nearly equal to an iteration of the classical FETI.

[2] In Section 4.3, we propose to use a random initialization and explain why.

subsets of range($\mathbf{\Delta}_i$). The possibility that $\mathbf{\Delta}_i$ may be singular, or equivalently, that there be some linear dependencies between the residuals, is a well identified problem in block CG. It corresponds to the case where the problem has been solved on a linear combination of the initial residuals. The solution is to *deflate* the block residual as is proposed in [35]. This deflation step is crucial to the efficiency of block CG since, the presence of a linear dependency means that all the work done on one direction is not contributing to convergence anymore. Finally, we mention that the handling of $\mathbf{\Delta}_i$ can be done according to remark 3.

4.2 Minimization property and connection with Geneo

Thanks to the (new) choice of initialization in B-FETI, the algorithm satisfies a minimization property that justifies why we expect a convergence behaviour similar to FETI-Geneo. First we rewrite Algorithm 3 as Algorithm 4 where the approximate solution $\tilde{\boldsymbol{\lambda}}_i$ and residual $\mathbf{r}_i$ are updated directly and the block structure is only used to generate the search directions.

Algorithm 4: Block FETI with full orthogonalization (equivalent to Algorithm 3)

$\tilde{\boldsymbol{\lambda}}_0 = \boldsymbol{\lambda}_0 + \mathbf{P}\boldsymbol{\lambda}_{00}$, $\mathbf{r}_0 = \mathbf{P}^T\left(\mathbf{d} - \mathbf{F}\tilde{\boldsymbol{\lambda}}_0\right)$, $i = 0$
$\mathbf{Z}_0 = \tilde{\mathbf{S}}\left(\ldots, \mathbf{B}^{(s)}(\mathbf{d}^{(s)} - \mathbf{F}^{(s)}\mathbf{B}^{(s)^T}\tilde{\boldsymbol{\lambda}}_0, \ldots\right)$
$\mathbf{W}_0 = \mathbf{P}\mathbf{Z}_0$
while $\sqrt{\mathbf{r}_i^T\mathbf{Z}_i\mathbf{1}} > \epsilon$ **do**
  $\mathbf{Q}_i = \mathbf{F}\mathbf{W}_i$
  $\mathbf{\Delta}_i = \mathbf{Q}_i^T\mathbf{W}_i$
  $\gamma_i = \mathbf{r}_i^T\mathbf{Z}_i$
  $\tilde{\boldsymbol{\lambda}}_{i+1} = \tilde{\boldsymbol{\lambda}}_i + \mathbf{W}_i\mathbf{\Delta}_i^+\gamma_i$
  $\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{P}^T\mathbf{Q}_i\mathbf{\Delta}_i^+\gamma_i$
  $\mathbf{Z}_{i+1} = \tilde{\mathbf{S}}\mathbf{P}^T\mathbf{Q}_i$
  $\mathbf{W}_{i+1} = \mathbf{P}\mathbf{Z}_{i+1}$ then for $0 \leqslant j \leqslant i$ $\begin{cases} \mathbf{\Phi}_{i,j} = \mathbf{Q}_j^T\mathbf{W}_{i+1} \\ \mathbf{W}_{i+1} \leftarrow \mathbf{W}_{i+1} - \mathbf{W}_j\mathbf{\Delta}_j^+\mathbf{\Phi}_{i,j} \end{cases}$
  $i \leftarrow i + 1$
end
$\boldsymbol{\lambda} = \tilde{\boldsymbol{\lambda}}_i$

With this notation the following minimization property holds.

Theorem 2. *The approximate solution $\tilde{\boldsymbol{\lambda}}_i$ computed by the i-th iteration of block FETI minimizes the error $\tilde{\boldsymbol{\lambda}}_i - P\tilde{\boldsymbol{\lambda}}$ in the $\mathbf{F}$-norm (induced by the FETI operator) over all possible*

$$\tilde{\boldsymbol{\lambda}}_i \in \bigoplus_{s=1}^{N} \mathcal{K}_i(\tilde{\mathbf{S}}, \mathbf{F}, \mathbf{R}_0^{(s)}), \tag{13}$$

where $\oplus$ indicates a direct sum, $\mathbf{R}_0^{(s)} = \mathbf{B}^{(s)}(\mathbf{d}^{(s)} - \mathbf{F}^{(s)}\mathbf{B}^{(s)^T}(\boldsymbol{\lambda}_0 + \mathbf{P}\boldsymbol{\lambda}_{00}))$ and $\mathcal{K}_i(\tilde{\mathbf{S}}, \mathbf{F}, \mathbf{R}_0^{(s)})$ is the associated Krylov subspace at iteration i:

$$\mathcal{K}_i(\tilde{\mathbf{S}}, \mathbf{F}, \mathbf{R}_0^{(s)}) = \text{span}\{\tilde{\mathbf{S}}\mathbf{R}_0^{(s)}, \ldots, (\tilde{\mathbf{S}}\mathbf{F})^{i-1}\tilde{\mathbf{S}}\mathbf{R}_0^{(s)}\}.$$

The proof can be written in a similar way to the usual proof of convergence for block CG [36], itself similar to the proof of convergence for CG [48]. The two main properties of block FETI (which condition the choice of $\mathbf{\Delta}_i$, $\mathbf{\Phi}_{i,j}$ and $\mathbf{\Gamma}_i$ or γ_i as in any block CG algorithm) are this minimization result and $\mathbf{F}$ conjugacy of search directions ($\mathbf{W}_i^\top\mathbf{F}\mathbf{W}_j = 0, \quad \forall i \neq j$). Moreover, in exact arithmetic $\mathbf{\Phi}_{i,j} = 0$ for all $i < j$ which is the short recurrence property.

Remark 5. The usual block CG optimality property [36] is that the error is minimized columnwise: each of the errors corresponding to each of the right hand sides is minimized over the Krylov subspace generated by all N right hand sides. The variable block CG method [35] was already an application of block CG to linear systems with a unique right hand side with the objective of taking advantage of the enlarged Krylov subspace. There, in the initialization step, $N-1$ arbitrary vectors are added to the right hand side $\mathbf{b}$ and to the initial guess $\mathbf{x}_0$ in order to construct a block initial residual and so, part of the block CG minimization property is to ensure that the usual error $\mathbf{x} - \mathbf{x}_i$ is minimized over a subspace N times larger than the usual Krylov subspace for CG. This is also the case in Theorem 2 with the added advantage that the minimization spaces are designed to enhance convergence as explained further on.

An important difference with the S-FETI minimization result (Theorem 1) is that this time the space over which we minimize can be written as a sum of Krylov subspaces. The fact that each of these Krylov subspaces is originated

by a local vector is the reason why we make the following conjecture for the number n_{block} of iterations required to achieve convergence with block FETI:

$$n_{block} \leqslant n_{geneo} + \max(n_{lanczos}), \tag{14}$$

where n_{geneo} and $n_{lanczos}$ are respectively the number of iterations needed to achieve convergence with the Geneo algorithm and the worse number of iterations of the Arnoldi type eigensolver applied inside each subdomain to setup the Geneo coarse space. In turn this conjecture would guarantee that Block FETI is scalable (*i.e* that the convergence does not depend on the number of subdomains N): n_{geneo} does not depend on N [52] and $n_{lanczos}$ either because it refers to a computation inside one subdomain. This stems from the fact that we are simultaneously computing information comparable to the one that constitutes the Geneo coarse space and updating the approximate solution. The arguments to support this statement are the following:

1. from the Geneo theory [52] we know that the eigenvalues of the preconditioned operator that are responsible for the slow convergence of FETI can be identified locally (per subdomain),
2. the theory of block CG [36] states that the main advantage of this algorithm is that, instead of catching one bad eigenvalue per iteration, it can catch as many as there are blocks,
3. the Krylov subspaces that are computed throughout the block FETI iterations are exactly the spaces that are computed by an Arnoldi type algorithm initialized by $\mathbf{R}_0^{(s)}$ to find the largest eigenvalues of the preconditioned operator.

In fact we expect (14) to be a rather pessimistic estimate since we are in fact using more information than by solving first the generalized eigenvalue problems and then keeping only the eigenvectors: here instead we minimize over the full Lanczos subspaces. Another advantage is that we no longer need to specify a threshold for selecting eigenvectors since the process stops when convergence is achieved. We do point out that the subspaces do not correspond exactly to the Lanczos subspaces for computing the Geneo eigenvectors since there the operators are also local. An exact diagnostic of which of these two methods requires more operations will be the subject of future work. Nevertheless we give some insight on this question for a particular test case in Section 5.1.

4.3 Random initialization

A particular initialization $\boldsymbol{\lambda}_{00}$ is required in cases where the N local right hand sides $\mathbf{B}^{(s)}\mathbf{K}^{(s)^+}\left(\mathbf{f}^{(s)} - \mathbf{t}^{(s)^T}\mathbf{B}^{(s)^T}\boldsymbol{\lambda}_0\right)$ do not excite all subdomains. As can be seen from definitions (4) and (2), this can happen when not all domains are loaded and/or when not enough subdomains have rigid body modes. Therefore at initialization a random starting vector $\boldsymbol{\lambda}_{00}$ is generated for $\tilde{\boldsymbol{\lambda}}$ and, in our applications, it was scaled to represent 1% of the forces ($\mathbf{f}^{(s)}$). The reason to choose a random initialization is for all the columns in the initial residual to be linearly independent and also for all the vectors in the solution space to be represented. This is somewhat similar to the random initialization of the bootstrap adaptive multigrid algorithm [4].

4.4 Cost of Block-FETI

One iteration of Block-FETI is more expensive than Simultaneous-FETI. The principal extra cost is due to the fact that both $\mathbf{F}$ and $\tilde{\mathbf{S}}$ need to be applied to N-blocks of vectors. The advantage is that the full reorthogonalization is no more an obligation, although necessary for most practical applications.

5 Assessments

We first compare the different techniques for various academic problems known to trigger convergence difficulties with an `octave` implementation, then we present first results for S-FETI on a realistic `fortran-mpi` implementation which allows for time measurements.

5.1 High heterogeneity

These test cases are inspired by [51,52]. Structures where heterogeneities are not aligned with the interface are known to cause convergence difficulties since classical scaling strategies are inefficient and only dedicated (Geneo) coarse problems can restore fast convergence.

Figure 1 presents one typical case of the 2D representation of a horizontal beam clamped on its left side and submitted to given shear and traction on its right side; the ratio between the length and the thickness is 9. The beam is constituted by the stacking of 7 layers of linear elastic materials with the same Poisson coefficient and Young moduli alternating between two values E_{stiff} and E_{soft}. This design aims at representing a soft material reinforced by stiff fibers. A band domain decomposition of 9 square subdomains is employed, so that each subdomain has at most 2

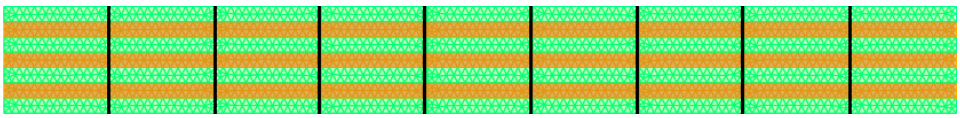


Figure 1: Heterogeneous beam, 3 stiff (orange, or dark) fibers are embedded in a soft (green) material, 9-subdomain band decomposition.

neighbours and the material is identical on either side of each interface. Each subdomain is meshed by 434 first-order triangular elements leading to a complete problem of 4 200 degrees of freedom of which 240 belong to the interface.

In table 1, we present the number of CG iterations needed for the classical FETI, Simultaneous FETI and Block FETI, each of them equipped either with the cheaper projector ($\mathbf{P}(\mathbf{A} = \mathbf{I})$) or the optimal one ($\mathbf{P}(\mathbf{A} = \tilde{\mathbf{S}})$), for material contrasts ranging from 1 to 10^6. Since the initial residual depends on the choice for $\boldsymbol{\lambda}_0$ and hence on the choice for $\mathbf{A}$ in the rigid body mode projector (see (4)), all iteration counts in table 1 have been obtained using a convergence criteria of 10^{-6} times the initial residual corresponding to $\mathbf{A} = \tilde{\mathbf{S}}$ in order to investigate only the convergence behavior independently of the initial estimate (the influence of the choice of $\mathbf{A}$ on the initial residual has been discussed in [22]).

We observe that even the optimized projector cannot prevent classical FETI from experiencing a degradation of its performance when heterogeneity increases (for the highest heterogeneity, FETI with $\mathbf{P}_{\tilde{\mathbf{S}}}$ needs 9 times as many iterations than in the homogeneous case). On the other hand, Simultaneous and Block FETI are much more robust, since at worse they need twice as many iterations as in the homogeneous case. Figure 2-a presents one classical evolution of the residual throughout the CG iterations, Figure 2-b was obtained with a refined mesh (7 times as many degrees of freedom), illustrating the independence of the performance with respect to the discretization.

At the bottom of table 1 we have also included the number of iterations needed for FETI Geneo (Algorithm 1 with the coarse space described at the end of Section 2) to converge. In the first case we have chosen the coarse space size to be 54: we select in each subdomain the six eigenvectors from (8) associated with the smallest non zero eigenvalues. It is known by experience [52] that this choice catches all the *bad* eigenvectors and indeed we observe that the number of iterations needed to converge is not influenced by the material heterogeneity. In the second case we have let the method select automatically the size of the coarse space by selecting all eigenvectors from (8) such that $0 < \mu^{(s)} < 0.15$ which guarantees [52] that the condition number of the preconditioned operator is below $3/0.15 = 20$. We observe that when the material is not very heterogeneous the method does not construct a coarse space and the number of iterations increases slightly from 6, when $E_{stiff} = E_{soft}$, to 9, when $E_{stiff} = 100\, E_{soft}$. After that the size of the coarse space increases with the heterogeneity (never exceeding 47) and the number of iterations remains below 14.

$\frac{E_{stiff}}{E_{soft}}$	1	10	100	10^3	10^4	10^5	10^6
# iterations FETI $\mathbf{P_I}$	6	9	18	34	51	63	67
# iterations FETI $\mathbf{P}_{\tilde{\mathbf{S}}}$	5	6	9	18	31	40	43
# iterations S-FETI $\mathbf{P_I}$	5	7	10	12	12	12	11
# iterations S-FETI $\mathbf{P}_{\tilde{\mathbf{S}}}$	5	6	8	9	10	9	9
# iterations B-FETI $\mathbf{P_I}$	5	7	9	10	11	11	11
# iterations B-FETI $\mathbf{P}_{\tilde{\mathbf{S}}}$	5	6	8	11	11	11	11
# iterations FETI Geneo $\mathbf{P}_{\tilde{\mathbf{S}}}$ (Fixed coarse space size:)	4 (54)	5 (54)	6 (54)	5 (54)	5 (54)	5 (54)	5 (54)
# iterations FETI Geneo $\mathbf{P}_{\tilde{\mathbf{S}}}$ ($0 < \mu^{(s)} < 0.15 \Rightarrow$ variable coarse space size:)	6 (0)	6 (0)	9 (0)	14 (13)	12 (32)	6 (44)	5 (47)

Table 1: Number of FETI iterations to decrease the initial residual by a 10^6 factor depending on the level of heterogeneity (Problem of figure 1)

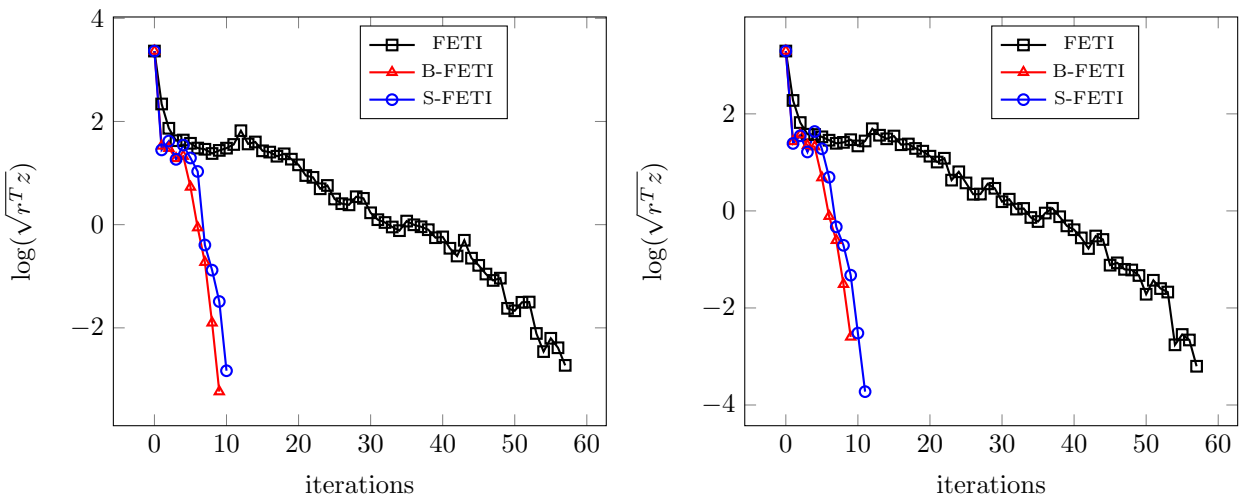
Of course the number of iterations is only one indication of how a method performs. Since this is the most costly part of the FETI algorithms we now compare the number of local solves. We use notation N for the number of subdomains and $\mathcal{N}$ for the largest number of neighbours of a subdomain including itself:

- Within the classical FETI algorithm two local solves (one Dirichlet and one Neumann) are performed per subdomain.
- Each iteration of S-FETI requires in each subdomain $\mathcal{N}$ Neumann solves (as explained in the last item in the discussion on the cost of S-FETI in Subsection 3.4) and 1 Dirichlet solve.
- Within Block FETI, once all blocks of vectors have complete fill-in, each iteration requires N Dirichlet solves and N Neumann solves per subdomain.

- Within FETI-2 with the Geneo coarse space no extra local solve is required in the iteration process but we need to consider the overhead cost of solving the Geneo eigenproblem. We assume that it is solved approximately by an, iterative, Lanczos method and that the computation of each eigenvector requires three Lanczos iterations and hence three applications of the operator ${\mathbf{S}^{(s)}}^{-1}{\mathbf{B}^{(s)}}^T\tilde{\mathbf{S}}\mathbf{B}^{(s)}\mathbf{v}^{(s)}$ which means 3 Neumann solves and $3\mathcal{N}$ Dirichlet solves. In order to build the coarse space we also need to apply the preconditioner to the eigenvector meaning $\mathcal{N}$ extra Dirichlet solves. In conclusion the cost of computing one vector for the coarse space is $3+4\mathcal{N}$.

In the test case at hand we have $N=9$ and $\mathcal{N}=3$ so each S-FETI iteration requires twice as many local solves as a classical FETI iteration, each B-FETI iteration requires 9 times as many local solves as a classical FETI iteration and FETI-Geneo iterations are as costly as classical FETI iterations but the computation of each eigenvectors requires approximately 15 local solves.
With this and the results from Table 1 we conclude that Block FETI increases the number of local solves while S-FETI and FETI-Geneo increase it for the easier problems and reduce it for the harder problems. The best improvements are observed with S-FETI.
In defense of the Block FETI method we recall that it does not require full reorthogonalization and that applying the same operator to N columns of a vector is a lot less expensive than applying the same operator N times. We also recall that FETI-Geneo is at this time the only algorithm for which convergence in a few iterations is guaranteed theoretically.

Remark 6. In this discussion we have left out two important parameters: the cost of orthogonalization (including for the coarse space) and the number of applications of the projector. In fact, as usual, the only way to truly compare the performance of our solver is to compare CPU times which we do in Subsection 5.8 for FETI and S-FETI.



(a) Coarse Mesh

(b) Fine Mesh

Figure 2: Convergence history: heterogeneous beam in flexion, 10^5-heterogeneity, $\mathbf{P_I}$

5.2 Bad aspect-ratio

Another cause for bad convergence is the bad aspect-ratio of subdomains. To activate this problem, we dilate the previous problem in the transverse directions: thickness now ranges between $1/5$ and 10 while the length of subdomains remains 1. The aspect ratio is defined as the thickness divided by the length. The connectivity of the mesh (and thus the number of nodes) is unchanged, elements are distorted when aspect ratio is far from 1.

Table 2 presents the number of CG iterations required to decrease the residual by a 10^6 factor. We observe that classical FETI converges more slowly when the interfaces are proportionally closer (aspect ratio >1): 3 times as many iterations for an aspect ratio of 5 and 5 times as many iterations when aspect ratio is 10. Simultaneous and Block FETI are in the worst case twice as slow.

Aspect ratio	1/5	1	5	10
# iterations FETI $\mathbf{P_I}$	5	6	17	29
# iterations S-FETI $\mathbf{P_I}$	5	5	9	11
# iterations B-FETI $\mathbf{P_I}$	5	5	8	10

Table 2: Number of FETI iterations to decrease the initial residual by a 10^6 factor depending on the aspect ratio (Problem of figure 1)

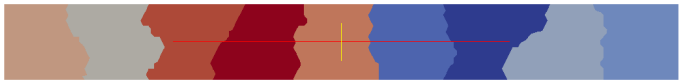


Figure 3: Homogeneous beam with irregular interfaces

5.3 Irregular interfaces

The shape of interfaces is known to have a strong influence on the convergence of the solver [30]: roughly, the straighter the better. The irregular decomposition of the beam shown in Figure 3 was obtained by an automatic graph partitioner (Metis [26]). Performances are presented in Table 3. We observe that the Simultaneous and Block FETI are less impacted by the irregularity of the interfaces than classical FETI. This will also be illustrated in the next example.

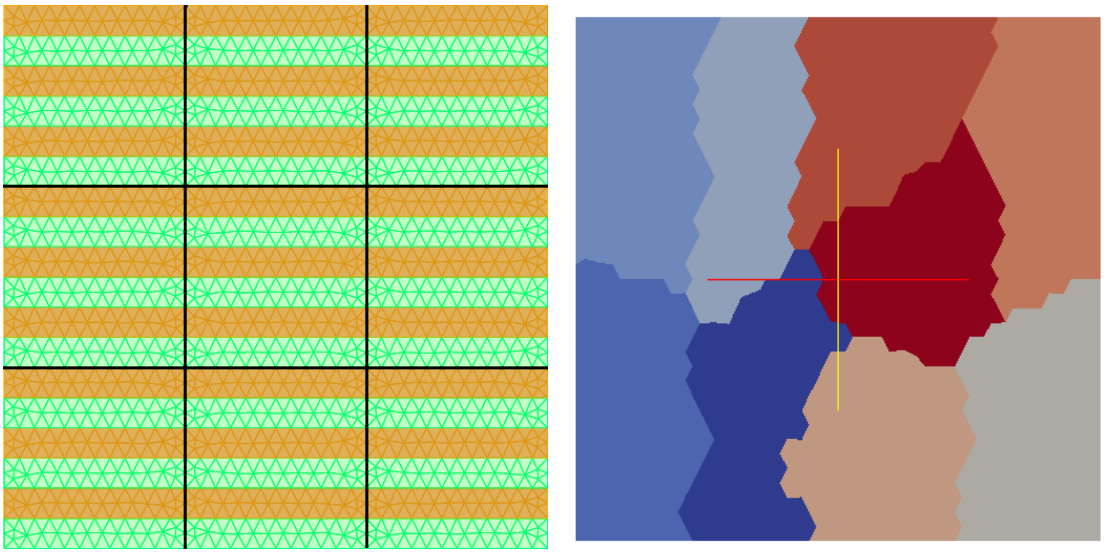
Decomposition	straight	irregular
# iterations FETI $\mathbf{P_I}$	6	18
# iterations FETI $\mathbf{P_{\tilde{S}}}$	5	17
# iterations S-FETI $\mathbf{P_I}$	5	9
# iterations S-FETI $\mathbf{P_{\tilde{S}}}$	5	9
# iterations B-FETI $\mathbf{P_I}$	5	10
# iterations B-FETI $\mathbf{P_{\tilde{S}}}$	5	10

Table 3: Number of FETI iterations to decrease the initial residual by a 10^6 factor depending on the decomposition (Figures 1 and 3)

5.4 Decomposition with cross-points

Cross-points, namely interface nodes that belong to more than two domains, often appear in decomposed problems and are known to sometimes be the cause of bad convergence in the case of heterogeneous structures [45].

Figure 4 presents the heterogeneous domain with two decompositions (regular and automatic). The square is clamped on its bottom side and submitted to traction and shear on its top side. The discretization leads to about 2 800 degrees of freedom; in the regular case the dimension of the interface is 300, in the automatic case it is 350. Table 4 summarizes the performance in the case of homogeneous or heterogeneous square (10^5 heterogeneity) with regular or automatic decompositions into 9 subdomains. As previously, we observe that the Simultaneous and Block FETI are much less influenced than the classical FETI by the irregular decomposition and the high heterogeneity in the presence of cross-points.



(a) Heterogeneous square (green = soft material, orange or dark = stiff material), regular decomposition (black lines)

(b) Automatic decomposition

Figure 4: Heterogeneous square

Heterogeneity ratio	1	1	10^5	10^5
Decomposition	regular	automatic	regular	automatic
# iterations FETI $\mathbf{P_I}$	12	19	42	94
# iterations FETI $\mathbf{P_{\tilde{S}}}$	12	21	48	93
# iterations S-FETI $\mathbf{P_I}$	8	11	10	19
# iterations S-FETI $\mathbf{P_{\tilde{S}}}$	8	12	11	19
# iterations B-FETI $\mathbf{P_I}$	7	9	8	13
# iterations B-FETI $\mathbf{P_{\tilde{S}}}$	7	9	9	13

Table 4: Number of FETI iterations to decrease the initial residual by a 10^6 factor for the square problem (Problem of figure 4)

5.5 Incompressibility

Incompressibility is also a known factor for convergence difficulties. A classical remedy is to add a coarse problem related to the conservation of the volume of the subdomains [7, 8, 19, 53, 54].

We consider the geometry of the beam, Figure 1, with homogeneous linear elastic material in plane strain. The bottom and top faces are clamped and a pressure is imposed on the left side whereas the right side is free. Table 5 gives the number of iterations to converge for various Poisson coefficients close to the incompressible limit $\nu \simeq 0.5$.

We observe how block strategies enable to limit the degradation of the convergence rate: when $(0.5 - \nu)$ goes from 10^{-5} to 10^{-6}, classical FETI needs twice as many iterations whereas block strategies only need 25% more iterations. Although this is interesting, it is clear that when the incompressibility limit is approached, an additional coarse grid as proposed in [53] is needed.

	$1/2 - \nu = 10^{-1}$	$1/2 - \nu = 10^{-5}$	$1/2 - \nu = 10^{-6}$
# iterations FETI	5	31	63
# iterations S-FETI	5	18	23
# iterations B-FETI	5	18	22

Table 5: Number of FETI iterations to decrease the initial residual by a 10^6 factor for the quasi-incompressible problem (Problem of figure 1)

5.6 Scalability results

We present scalability results for 2D problems. Beam-like structures similar to the ones in figure 1 and in figure 4-a are again considered. We make the number of domains vary considering a decomposition only in the horizontal direction for the problem of figure 1 and a regular decomposition in squares for the problem of figure 4. While changing the number of domains, we keep the discretization per domain constant, i.e. the size of the local problems in a domain are constant whereas the size of the overall problem is changing with the number of domains. The results are given in table 6 and 7 respectively. In the homogeneous case, the convergence of classical FETI is weakly impacted by the increase of the number of subdomains, whereas in the heterogeneous case, the number of iterations explodes when increasing N. S-FETI and B-FETI are much less sensitive with respect to the number of subdomains, which can be partially explained by the fact that for these methods adding subdomains also means adding search directions.

Nbr. of domains N		2	4	8	16	32
Homogeneous beam	FETI $\mathbf{P_I}$	5	6	6	6	6
	S-FETI $\mathbf{P_I}$	5	5	5	5	5
	B-FETI $\mathbf{P_I}$	5	5	5	5	5
Heterogeneous (10^5) beam	FETI $\mathbf{P_{\tilde{S}}}$	7	18	37	68	112
	S-FETI $\mathbf{P_{\tilde{S}}}$	5	8	9	10	10
	B-FETI $\mathbf{P_{\tilde{S}}}$	7	11	13	13	14
	S-FETI $\mathbf{P_I}$	7	10	12	13	13
	B-FETI $\mathbf{P_I}$	7	9	10	11	12

Table 6: Number of FETI iterations for various strip decompositions (problem of figure 1, constant subdomain size)

To investigate the convergence when the size of the local problems is changing, we consider the problem of figure 4-a with a decomposition in 3×3 domains and make the element size vary (hence also the size of the local problems). From table 8, it is observed that S-FETI and B-FETI are barely affected by the discretization size similarly to what is expected for the classical FETI.

		2×2 sd	3×3 sd	4×4 sd	5×5 sd
Homogeneous square	FETI $\mathbf{P_I}$	9	12	14	15
	S-FETI $\mathbf{P_I}$	8	8	9	9
	B-FETI $\mathbf{P_I}$	6	7	7	7
Heterogeneous (10^5) square	FETI $\mathbf{P_{\tilde{S}}}$	12	48	89	143
	S-FETI $\mathbf{P_{\tilde{S}}}$	7	11	13	14
	B-FETI $\mathbf{P_{\tilde{S}}}$	6	9	9	10
	S-FETI $\mathbf{P_I}$	7	10	12	13
	B-FETI $\mathbf{P_I}$	5	8	9	9

Table 7: Number of FETI iterations for various ruled decompositions of a square (problem of figure 4-a, constant subdomain size)

	# nodes	1531	4051	6919	10092	14273
Homogeneous square	FETI $\mathbf{P_I}$	12	13	14	15	15
	S-FETI $\mathbf{P_I}$	8	9	9	9	10
	B-FETI $\mathbf{P_I}$	7	7	8	8	8
Heterogeneous (10^5) square	FETI $\mathbf{P_{\tilde{S}}}$	48	42	46	46	50
	S-FETI $\mathbf{P_I}$	10	11	13	11	13
	B-FETI $\mathbf{P_I}$	8	8	9	9	9

Table 8: Number of FETI iterations for 3×3 decomposition of a square with varying discretization (problem of figure 4-a)

5.7 Inclusion problem

We consider the problem of inclusions near the boundary of the subdomains as represented in figure 5. Let L/H be the thickness of the layer between the boundary and the inclusion relative to the size of the subdomain. Table 9 gives the number of iterations for FETI, S-FETI and B-FETI in various geometric and material configurations. As explained in [19] (where a dedicated coarse space is proposed in the context of FETI-DP and BDDC to control the condition number) the soft inclusion case is problematic for classical domain decomposition methods, in particular when the inclusions are near the boundary.

As in previous experiments, S-FETI and B-FETI algorithms behave more robustly than classical FETI: for the thinest layer case, the worst heterogeneous case compared to the homogeneous case requires 2.5 times more iterations for classical FETI and only 1.3 for S-FETI and 1.37 for B-FETI.

L/H	1/8 (9 500 dofs)			1/12 (21 000 dofs)		
E_{inclu}/E_{bulk}	1	10^5	10^{-5}	1	10^5	10^{-5}
FETI	15	13	33	16	13	41
S-FETI	9	9	12	10	9	13
B-FETI	8	7	10	8	7	11

Table 9: Number of FETI iterations for 4×4 squares with thin layers (Fig. 5)

5.8 Optimized implementation and time measurement for S-FETI

The implementation of the S-FETI method can be optimized in several ways (refer to Algorithm 2):

Simultaneous forward-backward substitutions By definition, the s-th column of $\mathbf{Z}_i$ is non zero only on the interface of subdomain $\Omega^{(s)}$ and so, given a column in $\mathbf{Z}_i$, its product by $\mathbf{F}$ requires solving a Neumann problem only in the subdomain itself and its neighbours. Conversely, given one subdomain $\Omega^{(s)}$, the work load associated with the computation of $\mathbf{F}\mathbf{Z}_i$ is $\mathcal{N}$ Neumann solves $\mathbf{F}^{(s)}$ (once more $\mathcal{N}$ is the number of neighbours of a subdomain including itself). In particular this is much fewer than the rank of $\mathbf{Z}_i$ which is equal to the number of subdomains. These multiple local solutions can be computed much more efficiently on a multi-core machine as a single solution. Indeed, in both cases the number of memory accesses is almost equal to the number of non zero entries in the factorized matrix, whereas the arithmetic complexity is multiplied by the number of simultaneous right-hand-sides with the result that multiple forward-backward substitutions do not present the same memory bottleneck issue than a single one. For instance, for a finite element sparse matrix of dimension 200 000 on a 12-core Intel Nehalem processor, the time for a single forward-backward substitution on a single core is $0.7\,s$ with the Intel Pardiso solver whereas the time for 12 simultaneous forward-backward substitutions on 12 cores is only $1\,s$. For a single right-hand-side, multi-core parallelization only decreases the time by 30% at best. Finally, if several

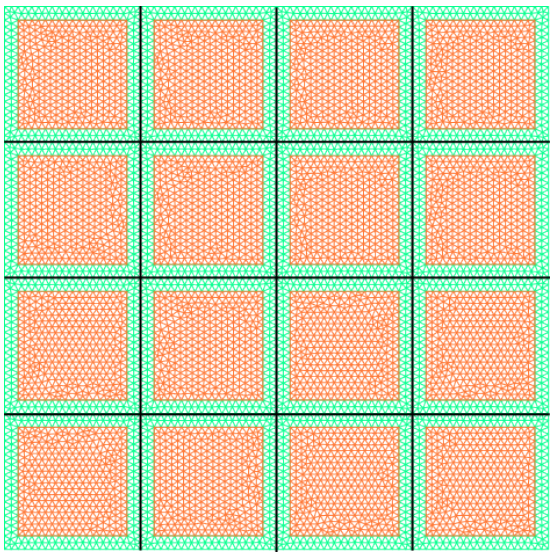


Figure 5: 4×4 square domain with inclusions (10^5 stiffness ratio).

single forward-substitutions for different matrices are performed at the same time, the memory bottleneck causes the time for each one to increase dramatically.

Parallel implementation of P with low rank corrections Within S-FETI, the columns of $\mathbf{W}_i$ are built from the projected vectors $\mathbf{PZ}_i$ and these are not local. Fortunately, the FETI projection $\mathbf{P}$ only performs a low rank correction so $\mathbf{PZ}_i$ can be computed at only a small extra cost. This cost is even further reduced with the simple $\mathbf{P_I}$ projector (*i.e.* when $\mathbf{A} = \mathbf{I}$, see section 2), which we have chosen to use in our tests. It operates as follows:

$$\mathbf{P_I Z}_i = \mathbf{Z}_i - \mathbf{G}\boldsymbol{\beta}_i \text{ where } \boldsymbol{\beta}_i \text{ solves } \mathbf{G}^T\mathbf{G}\boldsymbol{\beta}_i = \mathbf{G}^T\mathbf{Z}_i.$$

As usual this guarantees that $\mathbf{G}^T\mathbf{P_I Z}_i = 0$. Note that by construction $\mathbf{G}$ has the same sparse pattern as $\mathbf{Z}_i$ meaning that, given a column $\mathbf{Z}_i^{(s)}$ of $\mathbf{Z}_i$, $\mathbf{G}^T\mathbf{Z}_i^{(s)}$ is computed by applying only dot products by the columns of $\mathbf{G}$ corresponding to $\Omega^{(s)}$ and its neighbours. For this reason and because $(\mathbf{G}^T\mathbf{G})$ was factorized during the initialization phase of FETI, $\boldsymbol{\beta}_i$ can be computed in parallel. Each subdomain is in charge of computing one column $\boldsymbol{\beta}_i^{(s)}$ of $\boldsymbol{\beta}_i$ by solving, *via* a forward-backward substitution one system $(\mathbf{G}^T\mathbf{G})\boldsymbol{\beta}_i^{(s)} = \mathbf{G}^T\mathbf{Z}_i^{(s)}$. Of course, $\boldsymbol{\beta}_i$ is a dense matrix, whose number of rows is equal to the rank of $\mathbf{G}$ and number of columns is equal to the number of subdomains (also the rank of $\mathbf{Z}_i$), but nevertheless, once $\boldsymbol{\beta}_i$ has been computed, computing $\mathbf{PZ}_i = \mathbf{Z}_i - \mathbf{G}\boldsymbol{\beta}_i$ requires just a low rank correction of $\mathbf{Z}_i$ in each subdomain since only a few columns of $\mathbf{G}$ are non zero in each subdomain.

Preservation of locality in computing $\mathbf{FW}_i$ As already explained in the last item of the discussion in Subsection 3.4, the costly part in computing $\mathbf{FW}_i$ comes down to the computation of $\mathbf{FPZ}_i = \mathbf{FZ}_i - \mathbf{FG}\boldsymbol{\beta}_i$ and this can again be obtained with local low rank corrections of $\mathbf{FZ}_i$ using $(\mathbf{FG})$ that has been computed at the initialization phase of S-FETI.

Optimization of the orthogonalization procedure Once a set of vectors $\mathbf{PZ}_i$ has been computed, it must be $\mathbf{F}$-orthogonalized to compute the new set of search directions $\mathbf{W}_i$. Instead of using a modified Gram-Schmidt procedure that requires many MPI reductions of dimension 1, $(\mathbf{PZ}_i)^T\mathbf{FPZ}_i$ can be computed by computing the local contribution of each subdomain, using BLAS3 kernels, and only one MPI reduction to compute all the entries at once. Then we use a Choleski factorization of $\mathbf{PZ}_i^T\mathbf{FPZ}_i$ to compute the new set of $\mathbf{F}$-orthonormal search directions $\mathbf{W}_i$. The complete $\mathbf{F}$-orthogonalization with the previous search direction vectors can be performed by block as well, with the same kind of optimization using local BLAS3 kernels and reducing the number of reduction operations compared to the standard modified Gram-Schmidt method.

In order to evaluate the method, we consider several test cases of 2D linear elasticity on a square domain clamped on one side and with imposed displacements on the opposite side. They are presented in Figure 6. We use either a regular checkerboard decomposition into 10×10 regular square subdomains each containing 180 000 degrees of freedom (the global problem is then more than 17M degrees of freedom large once duplicated degrees of freedom have been removed) or a decomposition into slices (for two different orientations of the heterogenity bands, see figure 6) with 100 subdomains for the same global problems. The heterogeneous stiff stripes are either quasi-orthogonal to the interfaces, as in the figure at the center, or quasi-parallel to the interfaces as in the right. So, all the features that lead to poor convergence of the standard FETI method are present.

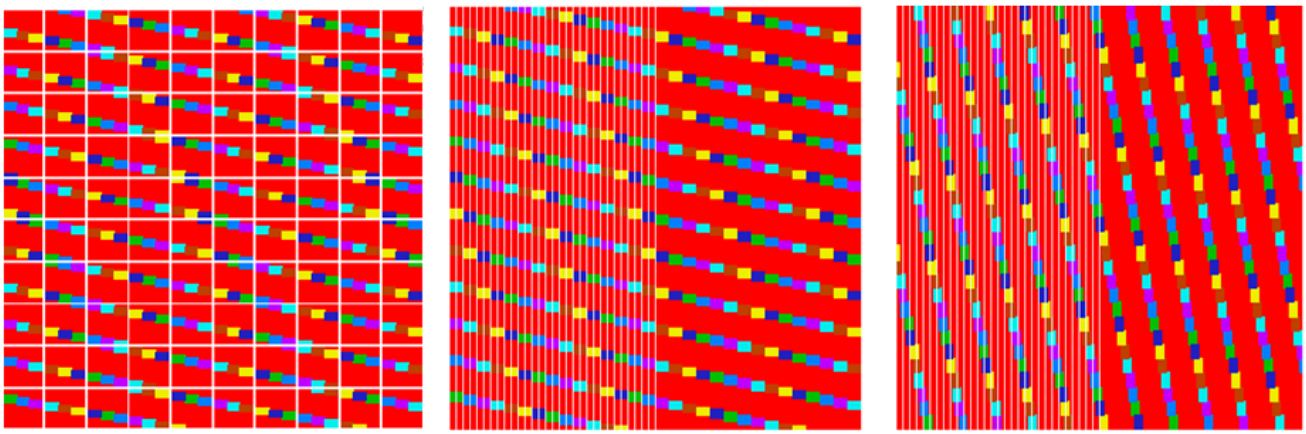


Figure 6: Three decompositions with heterogeneous materials

All subdomains are heterogeneous. The model features a composite material with a Young's modulus of 1 for the matrix which may be nearly incompressible (Poisson ratio 0.45 or 0.4999) containing rigid stripes made of various blocks randomly affected with a Young's modulus of 10, 10^2, 10^3 10^4 or 10^5. Since the stripes are not parallel to the borders of the square, each interface features simultaneously various degrees of heterogeneity along and across it. In the two cases of decomposition in slices, the aspect ratio of the subdomains is bad. The decomposition is the same in both cases, so all subdomains except the first are floating.

The tests are conducted on a cluster of 2.6 GHz 8-core Xeon processors connected by a gigabit ethernet network. Each subdomain is allocated to 4 cores (2 subdomains per processor). Intel fortran compiler and MKL-pardiso solver are used.

The FETI iterations are performed until both the relative error $\|\mathbf{u}_i - \mathbf{u}_{i-1}\|/\|\mathbf{u}_i\|$ and the relative residual $\|(\mathbf{K}\mathbf{u}_i - \mathbf{f})\|/\|\mathbf{f}\|$ for the global problem are smaller than 10^{-6}. For both FETI and S-FETI, the super-lumped scaling and rigid body projections are used.

Decomp.	Poisson	Solver	#iterations	#search directions	Max #local solutions	Time (s)
Checker	0.4999	FETI	233	233	466	991
		S-FETI	46	4600	276	320
Slices 1	0.4999	FETI	> 800	> 800	> 1600	> 7300
		S-FETI	152	15200	608	4653
Slices2	0.4999	FETI	> 800	> 800	> 1600	> 7300
		S-FETI	144	14400	576	4455
Slices 1	0.45	FETI	> 800	> 800	> 1600	> 7300
		S-FETI	48	4800	192	493
Slices2	0.45	FETI	409	409	818	1979
		S-FETI	36	3600	144	363

Table 10: CPU performance of FETI and S-FETI for 2D elasticity problems (figure 6)

Table 10 presents the results for the different test cases. For FETI and S-FETI, we compare the number of iterations needed to achieve convergence, the number of generated search directions (which is equal to the number of iterations for FETI and to the number of iterations multiplied by the number of subdomains for S-FETI), the maximal number of local (Dirichlet or Neumann) resolutions on the subdomains, and the wallclock time. In the cases where the number of FETI iterations is more than 800, the iterations were stopped before convergence was reached (the order of magnitude of the residual was still greater than 10^{-2}).

All these test cases show that the S-FETI method is more robust than the standard FETI method for this kind of highly heterogeneous problems. Even in the case of the checkerboard decomposition where the aspect ratio of the

subdomains is good, the S-FETI method is much faster than the FETI method. This case also clearly shows the benefit related to improved performance due to multiple forward-backward substitutions. The ratio between the numbers of local solutions for FETI and S-FETI is smaller than 2 whereas the ratio between the wall-clock times is greater than 3, although the reconjugation procedure is more arithmetically-intensive for S-FETI, since the total number of search directions is much larger.

6 Conclusion

A well-known problem when applying Domain Decomposition techniques to real engineering problems is the presence of heterogeneities along the interfaces. Classical preconditioning techniques perform very poorly in such cases. The FETI-Geneo was proposed lately to ensure robustness in those cases, by precomputing troublesome interface modes and including them in an auxiliary coarse-grid. That method however requires costly pre-processing.

In this paper we proposed two block strategies with the purpose of enriching the search space at every iteration on the interface problem. The methods were discussed and tested for the FETI method. Nevertheless applying the same concepts to other non-overlapping methods (e.g. BDD, FETIDP or BDDC) is straightforward.

The first method, Simultaneous-FETI (or S-FETI), exploits the additive structure of the preconditioner to generate a family of search directions and let the solver choose the best way to combine them instead of just computing their sum. The extra cost of the method is limited because sparsity can be exploited when searching for the solution in the multiple directions. One drawback of the S-FETI method is that it is no longer a genuine Conjugate Gradient method and full orthogonalisation of the search directions is necessary. This drawback is however a non-issue in practice since full orthogonalisation is required even for Conjugate Gradient.

The second method, Block-FETI, exploits the additive structure of the problem to generate a family of right-hand-sides to be solved by a block-conjugate gradient. Full reorthogonalization is no more mandatory, but this method requires many more local Dirichlet and Neumann solves than the classical FETI.

Both methods have very good computational properties since one iteration involves as many exchanges as classical FETI (but with larger amount of data) and they rely on block resolutions.

The assessments showed that both methods are much more robust than classical FETI for highly heterogeneous problems, for decompositions with jagged interfaces and for subdomains with bad aspect ratio. CPU time assessments of the S-FETI method showed that it can lead to very significant gains compared to standard FETI methods.

The methods need to be assessed on larger class of problems in particular in the cases where the decomposition involves a very large number of subdomains. It is expected that, since they imply highly optimized operations conducted on blocks of vectors, the proposed methods should be interesting default strategies for people concerned by the robustness of their Domain Decomposition solvers.

References

[1] L. Beirão da Veiga, L. F. Pavarino, S. Scacchi, O. B. Widlund, and S. Zampini. Isogeometric BDDC preconditioners with deluxe scaling. *SIAM J. Sci. Comput.*, 36(3):A1118–A1139, 2014.

[2] M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, and D. Rixen. Application of the FETI method to ASCI problems: Scalability results on a thousand-processor and discussion of highly heterogeneous problems. *International Journal for numerical methods in engineering*, 47(1-3):513–536, 2000.

[3] D. Brands, A. Klawonn, O. Rheinbach, and J. Schröder. Modelling and convergence in aterial wall simulations using a parallel FETI solution strategy. *Comput. Meth. Appl. Mech. Engrg.*, 11(5):569–583, October 2008.

[4] A. Brandt, J. Brannick, K. Kahl, and I. Livshits. Bootstrap AMG. *SIAM J. Sci. Comput.*, 33(2):612–632, 2011.

[5] Robert Bridson and Chen Greif. A multipreconditioned conjugate gradient algorithm. *SIAM J. Matrix Anal. Appl.*, 27(4):1056–1068 (electronic), 2006.

[6] C. R. Dohrmann. A preconditionner for substructuring based on constrained energy minimization. *SIAM Journal for Scientific Computing*, 25:246, 2003.

[7] Clark R. Dohrmann and Olof B. Widlund. An overlapping Schwarz algorithm for almost incompressible elasticity. *SIAM J. Numer. Anal.*, 47(4):2897–2923, 2009.

[8] Clark R. Dohrmann and Olof B. Widlund. Hybrid domain decomposition algorithms for compressible and almost incompressible elasticity. *Internat. J. Numer. Methods Engrg.*, 82(2):157–183, 2010.

[9] Victorita Dolean, Frédéric Nataf, Robert Scheichl, and Nicole Spillane. Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet-to-Neumann maps. *Comput. Methods Appl. Math.*, 12(4):391–414, 2012.

[10] Maksymilian Dryja, Marcus V. Sarkis, and Olof B. Widlund. Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions. *Numer. Math.*, 72(3):313–348, 1996.

[11] Yalchin Efendiev, Juan Galvis, Raytcho Lazarov, and Joerg Willems. Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms. *ESAIM Math. Model. Numer. Anal.*, 46(5):1175–1199, 2012.

[12] C. Farhat, P.-S. Chen, and F.-X. Roux. The two-level FETI method - part II: Extension to shell problems. parallel implementation and performance results. *Computer Methods in Applied Mechanics and Engineering*, 155:153–180, 1998.

[13] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method - part i: a faster alternative to the two-level FETI method. *International Journal for Numerical Methods in Engineering*, 50(7):1523–1544, 2001.

[14] C. Farhat and J. Mandel. The two-level FETI method for static and dynamic plate problems - part I: An optimal iterative solver for biharmonic systems. *Computer Methods in Applied Mechanics and Engineering*, 155:129–152, 1998.

[15] C. Farhat, K. Pierson, and M. Lesoine. The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems. *Computer Methods in Applied Mechanics and Engineering*, 184(2-4):333–374, 2000.

[16] C. Farhat and D. Rixen. A new coarsening operator for the optimal preconditioning of the dual and primal domain decomposition methods: application to problems with severe coefficient jumps. In S. F. McCormick N. Duane Melson, T. A. Manteuffel and C. C. DouglasM, editors, *Proceedings of the Seventh Copper Mountain Conference on Multigrid Methods*, pages 301–316, 1995.

[17] 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, 1991.

[18] Juan Galvis and Yalchin Efendiev. Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces. *Multiscale Model. Simul.*, 8(5):1621–1644, 2010.

[19] Sabrina Gippert, Axel Klawonn, and Oliver Rheinbach. Analysis of feti-dp and bddc for linear elasticity in 3d with almost incompressible components and varying coefficients inside subdomains. *SIAM Journal on Numerical Analysis*, 50(5):2208–2236, 2012.

[20] P. Gosselet and C. Rey. Non-overlapping domain decomposition methods in structural mechanics. *Archives of Computational Methods in Engineering*, 13(4):515–572, 2006.

[21] P. Gosselet, C. Rey, and J. Pebrel. Total and selective reuse of Krylov subspaces for the solution to a sequence of nonlinear structural problems. *International Journal for Numerical Methods in Engineering*, 94(1):60–83, 2013.

[22] P. Gosselet, C. Rey, and D. Rixen. On the initial estimate of interface forces in feti methods. *Computer Methods in Applied Mechanics and Engineering*, 192(25):2749–2764, 2003.

[23] P. Gosselet, D. J. Rixen, and C. Rey. A domain decomposition strategy to efficiently solve structures containing repeated patterns. *International Journal for Numerical Methods in Engineering*, 78(7):828–842, 2009.

[24] Chen Greif, Tyrone Rees, and Daniel B. Szyld. MPGMRES: a generalized minimum residual method with multiple preconditioners. Technical Report 11-12-23, Department of Mathematics, Temple University, 2011. Revised September 2012 and January 2014. Also available as Technical Report TR-2011-12, Department of Computer Science, University of British Columbia.

[25] Chen Greif, Tyrone Rees, and Daniel B. Szyld. Additive Schwarz with variable weights. Technical Report 12-11-30, Department of Mathematics, Temple University, November 2012.

[26] G. Karypis and V. Kumar. METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. Technical report, Department of Computer Science, University of Minnesota, 1998. http://glaros.dtc.umn.edu/gkhome/views/metis.

[27] A. Klawonn, P. Radtke, and O. Rheinbach. Adaptive coarse spaces for BDDC with a transformation of basis. In *Twenty Second International Conference on Domain Decomposition Methods*, 2014.

[28] A. Klawonn, P. Radtke, and O. Rheinbach. FETI-DP Methods with an Adaptive Coarse Space. *SIAM J. NUMER. ANAL.*, 53(1):297–320, 2015.

[29] Axel Klawonn and Oliver 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):1400–1414, 2007.

[30] Axel Klawonn, Oliver Rheinbach, and Olof B. Widlund. An analysis of a FETI–DP algorithm on irregular subdomains in the plane. *SIAM J. Numer. Anal.*, 46(5):2484–2504, 2008.

[31] Jan Mandel. Balancing domain decomposition. *Communications in Numerical Methods in Engineering*, 9(3):233, 1993.

[32] Jan Mandel and Marian Brezina. Balancing domain decomposition for problems with large jumps in coefficients. *Math. Comp.*, 65(216):1387–1401, 1996.

[33] Jan Mandel and Bedřich Sousedík. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Comput. Methods Appl. Mech. Engrg.*, 196(8):1389–1399, 2007.

[34] Frédéric Nataf, Hua Xiang, Victorita Dolean, and Nicole Spillane. A coarse space construction based on local Dirichlet-to-Neumann maps. *SIAM J. Sci. Comput.*, 33(4):1623–1642, 2011.

[35] A. A. Nikishin and A. Yu. Yeremin. Variable block CG algorithms for solving large sparse symmetric positive definite linear systems on parallel computers. I. General iterative scheme. *SIAM J. Matrix Anal. Appl.*, 16(4):1135–1153, 1995.

[36] Dianne P. O'Leary. The block conjugate gradient algorithm and related methods. *Linear Algebra Appl.*, 29:293–322, 1980.

[37] C. Pechstein, M. Sarkis, and R. Scheichl. New theoretical coefficient robustness results for FETI-DP. In *Domain Decomposition Methods in Science and Engineering XX*, Lecture Notes in Computational Science and Engineering. Springer-Verlag, 2011.

[38] Clemens Pechstein. *Finite and boundary element tearing and interconnecting solvers for multiscale problems*, volume 90 of *Lecture Notes in Computational Science and Engineering*. Springer, Heidelberg, 2013.

[39] Clemens Pechstein and Robert Scheichl. Analysis of FETI methods for multiscale PDEs. *Numer. Math.*, 111(2):293–333, 2008.

[40] Clemens Pechstein and Robert Scheichl. Scaling up through domain decomposition. *Appl. Anal.*, 88(10-11):1589–1608, 2009.

[41] Clemens Pechstein and Robert Scheichl. Analysis of FETI methods for multiscale PDEs. Part II: interface variation. *Numer. Math.*, 118(3):485–529, 2011.

[42] Clemens Pechstein and Robert Scheichl. Weighted Poincaré inequalities. *IMA J. Numer. Anal.*, 33(2):652–686, 2013.

[43] Valentine Rey, Christian Rey, and Pierre Gosselet. A strict error bound with separated contributions of the discretization and of the iterative solver in non-overlapping domain decomposition methods. *Computer Methods in Applied Mechanics and Engineering*, 270(1):293–303, 2013.

[44] D. Rixen. *Substructuring and Dual Methods in Structural Analysis*. PhD thesis, Université de Liège, Belgium, Collection des Publications de la Faculté des Sciences appliquées, n.175, 1997.

[45] D. Rixen and C. Farhat. A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *Internat. J. Num. Meth. Engin.*, 44(4):489–516, 1999.

[46] 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, 1999.

[47] D.J. Rixen. A domain decomposition interface solver with multiple direction of descent for heterogeneous problems. In *84th Annual Meeting of the International Association of Applied Mathematics and Mechanics, GAMM, Novi-Sad, Serbia (oral presentation only)*, 18-22 March 2013.

[48] Yousef Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2003.

[49] Bedřich Sousedík, Jakub Šístek, and Jan Mandel. Adaptive-Multilevel BDDC and its parallel implementation. *Computing*, 95(12):1087–1119, 2013.

[50] N. Spillane. *Robust domain decomposition methods for symmetric positive definite problems*. PhD thesis, Thèse de l'Ecole doctorale de Mathmatiques de Paris centre, Laboratoire Jacques Louis Lions, Université Pierre et Marie Curie, Paris, 2014.

[51] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl. Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. *Numer. Math.*, 126(4):741–770, 2014.

[52] Nicole Spillane and Daniel J. Rixen. Automatic spectral coarse spaces for robust FETI and BDD algorithms. *Internat. J. Num. Meth. Engin.*, 95(11):953–990, 2013.

[53] B. Vereecke, H. Bavestrello, and D. Dureisseix. An extension of the FETI domain decomposition method for incompressible and nearly incompressible problems. *Computer methods in applied mechanics and engineering*, 192(31):3409–3429, 2003.

[54] Attila Zsaki, Daniel J. Rixen, and Marius Paraschivoiu. A substructured based iterative inner solver coupled with Uzawa's algorithm for the Stokes problem. *Internat. J. Numerical Methods in Fluids*, 43:215–230, 2003.