Simultaneous-FETI and related block strategies: robust domain decomposition methods for engineering problems.

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

Simultaneous-FETI and related block strategies: robust domain decomposition methods for engineering problems.

Pierre Gosselet *– Daniel Rixen †– François-Xavier Roux ‡– Nicole Spillane §

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 sharing 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 performances 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) [1] and its primal counterpart (the Balanced Domain Decomposition or BDD [2]) have been successfully applied to solve several challenging mechanical problems (e.g. [3, 4]). 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

*LMT-Cachan / ENS-Cachan, CNRS, Pres UniverSud Paris, 61 avenue du président Wilson, 94235 Cachan, France

†Technische Universität München, Faculty of Mechanics, Institute of Applied Mechanics, Boltzmannstr. 15, 85748 Garching, Germany.

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

§Pontificia Universidad Católica de Chile, Programa ingeniería matématica, Av. V. Mackenna 4860, Santiago, Chile.

Decomposition have been developed over the years to improve their robustness and efficiency: an overview can be found for instance in [5].

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). Several improvements have been proposed over the years to tackle heterogeneous problems in domain decomposition techniques [6, 7, 8] and to circumvent the poor convergence of the iterations on the interface problems due to jagged interfaces [9]. Although several variants can handle heterogeneities across the interface efficiently, the efficiency and robustness of the methods are very poor when several heterogeneities are found *along* the interface (as exhibited when decomposing a tire in slices so that the cables cross over the interface). A remedy could be to decompose these hard problems in such a way that each domain is nearly homogeneous [3] but this approach most often results in bad load balancing and in domains with bad aspect ratios, which also poses a challenge for solving the interface problem iteratively.

As was described in recent publications [10, 11, 12, 13, 14] for the overlapping Additive Schwarz method and [15, 16] 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 interfaces of each subdomain: the strategy is to generate these problematic modes and apply deflation strategies (or coarse grid 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 [12, 16, 17], exhibit remarkable robustness, bot 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. Unfortunately that robustness comes with a significant computational overhead related to finding the coarse space of 'bad' modes through eigenvalue problems.

In the present contribution, we discuss an idea that was originally proposed in [18] for two subdomains and consists in generating several search directions originating from the preconditioning and using them to solve the interface problem iteratively. A similar paradigm was later used in [19] to solve problems that include repeated components. Here the method will be generalized for an arbitrary number of domains: the Simultaneous-FETI (or S-FETI). Since this method can be seen as a special block iterative strategies, we will also discuss a second block approach in this paper where several search directions are built considering several right-hand sides of the problem.

In section 2 we give a short summary of the FETI method since it forms the basis of the proposed strategy. Then, in section 3, the S-FETI is explained. Another block approach is described in section 4. 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 [20, 21]) or to primal Schur complement methods such as BDD.

2 FETI in a nutshell

Here we will shortly summarize the basic FETI strategy that will be used throughout this paper (see for instance [1, 5] for further details). Let us consider the symmetric positive definite problem $\mathbf{Ku} = \mathbf{f}$ 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

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

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:

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

$\mathbf{S}^{(s)}$ 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, $\mathbf{R}^{(s)}$ is a basis of rigid body modes. We also write:

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

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

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

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 [22].

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:

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

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 [7] 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 [23, 24]. 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.*

Algorithm 1: FETI2 with full orthogonalization

$\mathbf{r}_0 = \mathbf{P}^T{\mathbf{P}_\mathbf{C}}^T(\mathbf{d}-\mathbf{F}\boldsymbol{\lambda}_0)$
$\mathbf{z}_0 = \tilde{\mathbf{S}}\mathbf{r}$, $\mathbf{w}_0 = \mathbf{P}\mathbf{z}_0$, $\hat{\boldsymbol{\lambda}}_0 = 0$, $i = 0$
while $\sqrt{\mathbf{r}_i^T\mathbf{z}_i} > \epsilon$ **do**
  $\mathbf{q}_i = {\mathbf{P}_\mathbf{C}}^T\mathbf{F}\mathbf{w}_i$
  $\delta_i = \mathbf{q}_i^T\mathbf{w}_i$
  $\gamma_i = \mathbf{r}_i^T\mathbf{z}_i$
  $\hat{\boldsymbol{\lambda}}_{i+1} = \hat{\boldsymbol{\lambda}}_i + (\gamma_i/\delta_i)\mathbf{w}_i$
  $\mathbf{r}_{i+1} = \mathbf{r}_i - (\gamma_i/\delta_i)\mathbf{P}^T\mathbf{q}_i$
  $\mathbf{z}_{i+1} = \tilde{\mathbf{S}}\mathbf{r}_{i+1}$
  $\mathbf{w}_{i+1} = \mathbf{P}\mathbf{z}_{i+1}$ 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
$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \tilde{\boldsymbol{\lambda}}_0 + \mathbf{P}_\mathbf{C}\hat{\boldsymbol{\lambda}}_i$

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 associated with the largest eigenvalues of the following generalized system:

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

Various techniques can be employed to approximate these eigenvectors, like recycling of nearby Krylov subspaces [25]. One important result is provided by [16] where it is proved that these eigenvectors always originate from local effects so that they can be generated by a family of local eigenvalue problems:

$$\mathbf{S}^{(s)}\mathbf{v}^{(s)} - \mu^{(s)}{\mathbf{B}^{(s)}}^T\tilde{\mathbf{S}}\mathbf{B}^{(s)}\mathbf{v}^{(s)} = 0 \tag{8}$$

Including the smallest frequency modes resulting from these eigenvalue problems in an auxiliary coarse grid led to the so-called FETI-Geneo [16]. Unfortunately, the solution to these eigenproblems followed by an augmented resolution incures a significant computational overhead. Yet fewer iterations are needed to converge when solving the interface problem and the method is numerically more stable.

In [17] an algorithm, named frugal-FETI, was proposed in order to capture during the classical resolution these penalizing local contributions. It is an interesting step in building a robust and efficient algorithm although it suffers from the fact that it needs to modify the coarse space every time new "bad" vectors are detected in the iteration.

The strategy proposed next, generalizing the multi-direction approach discussed in [18], 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) was introduced in [18] on a simple example with two subdomains. It exploits 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. Finally we comment on the cost of S-FETI.

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 minimisation process of the CG algorithm choose the best combination of local terms $\tilde{\mathbf{B}}^{(s)}\tilde{\mathbf{S}}^{(s)}\tilde{\mathbf{B}}^{(s)^T}$ (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 $\mathbf{Z} = \left(\dots, \tilde{\mathbf{B}}^{(s)}\tilde{\mathbf{S}}^{(s)}\tilde{\mathbf{B}}^{(s)^T}\mathbf{r}, \dots\right)$. The connexion between $\mathbf{Z}$ and the usual $\mathbf{z}$ is of course that $\mathbf{z} = \mathbf{Z}\mathbf{1}$ where $\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^N$ 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 [26]).

The iteration scheme is given in Algorithm 2. For sake of clarity we give the size of the different operators (for $i, j = 1, \dots, N$)

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

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

3.2 S-FETI as a multipreconditioned CG algorithm

The S-FETI algorithm is a multipreconditioned CG algorithm [27]. 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 $\mathbf{w} = \mathbf{W}\mathbf{1} \in \text{range}(\mathbf{W})$ 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

Algorithm 2: Simultaneous FETI

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

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

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

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

since, in practice, full orthogonalization is often used in classical FETI. The conjugation is crucial since it allows to prove the minimization property in the following theorem [27].

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*

$$\tilde{\boldsymbol{\lambda}}_i \in \bigoplus_{j=0}^{i-1} \operatorname{span}\left(\mathbf{W}_j\right), \tag{9}$$

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 [28]. 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 ($\mathbf{W}_i^\top\mathbf{F}\mathbf{W}_j = 0, \quad \forall\, i \neq j$). The main difference with the block-FETI algorithm introduced in the next section is that the minimization space cannot be written in terms of Krylov spaces. Indeed, at each iteration the approximate solution is updated in the direction given by the optimal linear combination of the local preconditioners but, since the coefficients in the linear combination change from one iteration to the next, there is no Krylov structure.

Remark 2. *Although our numerical results (Section 5) point to the fact that S-FETI performs perfectly well we mention the more recent multipreconditioned GMRES algorithm [29] 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*

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

N being the number of subdomains. In [30] multiply preconditioned GMRES is applied to the Additive Schwarz domain decomposition technique.

3.3 Connection to FETI Geneo

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 behaviour is comparable to that of the FETI Geneo algorithm [16] where a coarse space is constructed by solving generalized eigenvalue problems (8) in each subdomain that isolate 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.

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 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 requires the same cost for the preconditioning 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. Further one observes that

$$\begin{aligned}\mathbf{Q}_{i+1} = \mathbf{F}\mathbf{W}_{i+1} &= \mathbf{F}\mathbf{P}\mathbf{Z}_{i+1} - \sum_{j=0}^{i} \mathbf{Q}_j \mathbf{\Delta}_j^+ \mathbf{\Phi}_{i,j} \\ &= \left(\mathbf{F}\mathbf{Z}_{i+1} - \mathbf{F}\mathbf{A}\mathbf{G}(\mathbf{G}^T\mathbf{A}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{Z}_{i+1}\right) - \sum_{j=0}^{i} \mathbf{Q}_j \mathbf{\Delta}_j^+ \mathbf{\Phi}_{i,j}\end{aligned} \tag{10}$$

  where the product $\mathbf{F}\mathbf{A}\mathbf{G}$ 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 5.6.

4 Block FETI

The sole flaw of Simultaneous FETI is that it breaks the short recurence by requiring full orthogonalization. The block FETI technique is a classical block Conjugate Gradient (block CG) [31] where a block of right-hand sides is generated in order to activate the local effects. We first introduce the block FETI algorithm. Then we give the minimization property that is satisfied at each iteration and discuss its initialization as well as why it is expected that it converge fast (in a way comparable to S-FETI and 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 can rewrite the dual interface problem (3) by considering separately the contribution of each subdomain to the right-hand $\mathbf{d}$: the N-block of multiple right-hand sides is $\left(\dots, \mathbf{B}^{(s)}\mathbf{K}^{(s)^+}\mathbf{f}^{(s)}, \dots\right)$. Then, for an initial guess $\boldsymbol{\lambda}_{00}$, the initial residual for the classical and for the block FETI respectively write

$$\begin{aligned} \mathbf{r}_0 &= \mathbf{P}^T \left(\sum_s \mathbf{B}^{(s)}\mathbf{K}^{(s)^+} \left(\mathbf{f}^{(s)} - \mathbf{B}^{(s)^T}(\boldsymbol{\lambda}_0 + \mathbf{P}\boldsymbol{\lambda}_{00}) \right) \right) \\ \mathbf{R}_0 &= \mathbf{P}^T \left(\dots, \mathbf{B}^{(s)}\mathbf{K}^{(s)^+} \left(\mathbf{f}^{(s)} - \mathbf{B}^{(s)^T}(\boldsymbol{\lambda}_0 + \mathbf{P}\boldsymbol{\lambda}_{00}) \right), \dots \right). \end{aligned} \tag{11}$$

With this setup we solve the system $\mathbf{P}^T\mathbf{F}\mathbf{P}\tilde{\boldsymbol{\Lambda}} = \mathbf{R}_0$ using block CG. This is the block FETI algorithm presented in Algorithm 3. 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.

Since the block system is connected to the original system by the relations $\tilde{\boldsymbol{\lambda}} = \tilde{\boldsymbol{\Lambda}}\mathbf{1}$ and $\mathbf{r}_0 = \mathbf{R}_0\mathbf{1}$ (once more with $\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^N$) we can monitor the convergence of the original system within the block CG.

As in the S-FETI case, 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 all Γ_i and $\boldsymbol{\Phi}_{i,j}$ are in range$(\boldsymbol{\Delta}_i)$. The possibility that $\boldsymbol{\Delta}_i$ 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* this residual as is proposed in [32]. 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.

[1] Note that in the case of two subdomains it was shown in [18] that the cost of one S-FETI iteration is nearly equal to an iteration of the classical FETI.

Algorithm 3: Block FETI with full orthogonalization

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

$\mathbf{Z}_0 = \tilde{\mathbf{S}}\mathbf{R}_0$, $\mathbf{W}_0 = \mathbf{P}\mathbf{Z}_0$, $\tilde{\boldsymbol{\Lambda}}_0 = 0$, $i = 0$

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

$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \mathbf{P}\boldsymbol{\lambda}_{00} + \tilde{\boldsymbol{\Lambda}}\mathbf{1}$

4.2 Minimization property

The block FETI algorithm belongs to the class of block CG algorithms [31]. The following theorem gives the minimization property that is satisfied at each iteration.

Theorem 2. *The approximate solution 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*

$$\tilde{\boldsymbol{\lambda}}_i \in \bigoplus_{j=0}^{i-1} \text{span}\,(\mathbf{W}_j) = \bigoplus_{s=1}^{N} \mathcal{K}_i(\tilde{\mathbf{S}}, \mathbf{F}, \mathbf{R}_0^{(s)}), \tag{12}$$

where $\oplus$ indicates a direct sum, $\mathbf{R}_0^{(s)}$ is the s-th column in the initial residual and $\mathcal{K}_i(\tilde{\mathbf{S}}, \mathbf{F}, \mathbf{R}_0^{(s)})$ is the associated Krylov subspace at iteration i:

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

The proof of convergence for block CG can be written in a similar way to the usual proof of convergence for CG [28]. The two main properties of block CG (which condition the choice of $\boldsymbol{\Delta}_i$, $\boldsymbol{\Gamma}_i$ and $\boldsymbol{\Phi}_{i,j}$) 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 $\boldsymbol{\Phi}_{i,j} = 0$ for all $i < j$ which is the short recurrence property.

Remark 3. *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.*

4.3 Random initialization and connexion with Geneo

A particular initialization $\boldsymbol{\lambda}_{00}$ is required in cases where the local right hand sides $\mathbf{B}^{(s)}\mathbf{K}^{(s)^+}\left(\mathbf{f}^{(s)} - \mathbf{B}^{(s)^T}\boldsymbol{\lambda}_0\right)$ 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

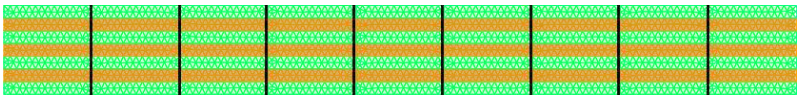


Figure 1: Heterogeneous beam, 3 stiff (orange, or dark) fibers are embedded in a soft (green) material, 9-subdomain band decomposition.

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 [33]. In particular we can count on a contribution to the initial residual from every vector that slows down convergence (according to the Geneo theory [16], see also section 2, it is a good strategy to look for them locally). Finally the theory of block CG [31] 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. In our case it is not only expected that the block CG algorithm find the isolated eigenvalues first (as any CG algorithm) but also that it find isolated eigenvalues corresponding to different subdomains simultaneously. For this reason convergence should be very fast and comparable both to S-FETI and FETI Geneo.

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 [12, 16]. Structures where heterogeneities are 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 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 3628 degrees of freedom of which 240 belong to the interface.

In table 1, we present the number of CG iterations needed to decrease the residual by a 10^6 factor, for Classical FETI, Simultaneous FETI and Block FETI, each of them equipped either with the cheaper ($\mathbf{P_I}$) or optimal projector ($\mathbf{P_{\tilde{S}}}$), for material contrasts ranging from 1 to 10^6.

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 7 times as many iterations than in the homogeneous case). On the other hand, Simultaneous and Block FETIs are much more stable, 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 [16] 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 [16] 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	8	16	29	44	57	63
# iterations FETI $\mathbf{P}_{\tilde{\mathbf{S}}}$	6	6	9	18	31	41	43
# iterations S-FETI $\mathbf{P_I}$	5	6	8	10	11	10	10
# iterations S-FETI $\mathbf{P}_{\tilde{\mathbf{S}}}$	5	6	8	9	9	9	8
# iterations B-FETI$\mathbf{P_I}$	5	6	7	8	9	9	9
# iterations B-FETI $\mathbf{P}_{\tilde{\mathbf{S}}}$	5	6	6	10	12	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

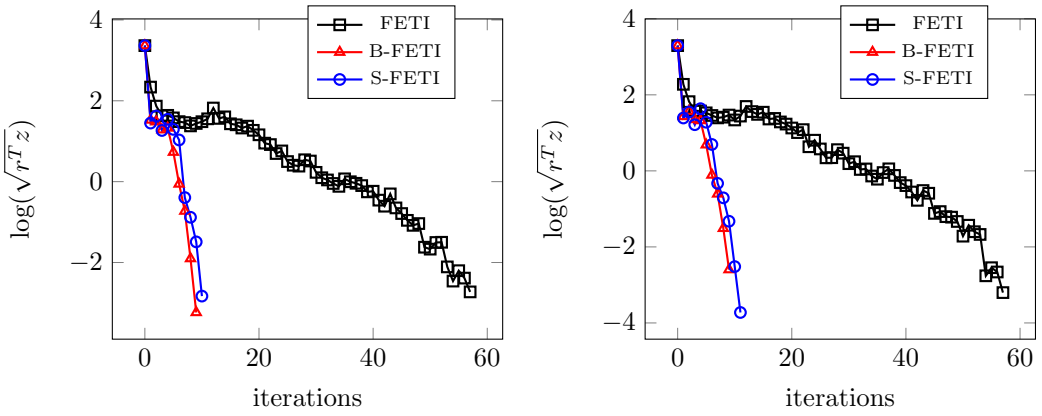
Of course the number of iterations is only one indication of how a method performs. Since this is the most costly part of the FETI algorithms we now compare the number of local solves. We use notation N for the number of subdomains and $\mathcal{N}$ for the largest number of neighbours of a subdomain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

Remark 4. *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 5.6 for FETI and S-FETI.*



(a) Coarse Mesh (b) Fine Mesh

Figure 2: 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

Figure 3: Homogeneous beam with irregular interfaces

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

5.3 Irregular interfaces

The shape of interfaces is known to have a strong influence on the convergence of the solver [9]: roughly, the straighter the better. The irregular decomposition of the beam shown in Figure 3 was obtained by an automatic graph partitioner (Metis [34]). 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.

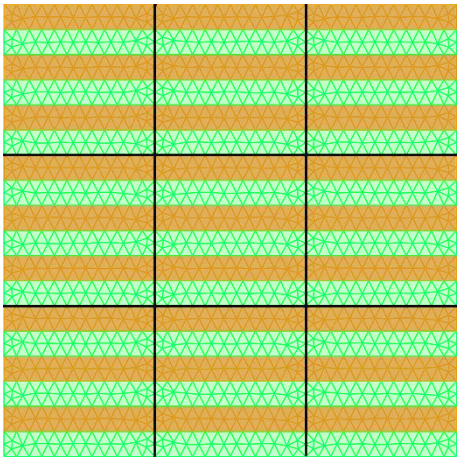
Decomposition	straight	irregular
# iterations FETI $\mathbf{P_I}$	6	12
# iterations FETI $\mathbf{P_{\tilde{S}}}$	6	11
# iterations S-FETI $\mathbf{P_I}$	5	8
# iterations B-FETI $\mathbf{P_I}$	5	8

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

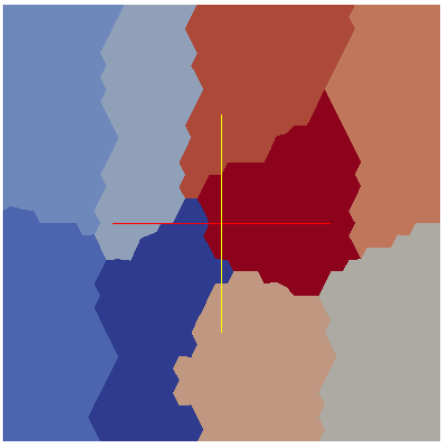
5.4 Decomposition with cross-points

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 [7].

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 decomposition 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	17	44	93
# iterations FETI $\mathbf{P_{\tilde{S}}}$	12	19	46	93
# iterations S-FETI $\mathbf{P_I}$	8	8	12	16
# iterations B-FETI $\mathbf{P_I}$	7	7	9	11

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

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 [35].

We consider the geometry of the beam, Figure 2, 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 [35] 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.

5.6 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 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 $\mathbf{P}\mathbf{Z}_i$ and these are not local. Fortunately, the FETI projection $\mathbf{P}$ only performs a low rank correction so $\mathbf{P}\mathbf{Z}_i$ 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:

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

As usual this guarantees that $\mathbf{G}^T\mathbf{P_I}\mathbf{Z}_i = 0$. 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 $\mathbf{P}\mathbf{Z}_i = \mathbf{Z}_i - \mathbf{G}\boldsymbol{\beta}_i$ 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 $\mathbf{F}\mathbf{W}_i$ As already explained in the last item of the discussion in Subsection 3.4, the costly part in computing $\mathbf{F}\mathbf{W}_i$ comes down to the computation of $\mathbf{F}\mathbf{P}\mathbf{Z}_i = \mathbf{F}\mathbf{Z}_i - \mathbf{F}\mathbf{G}\boldsymbol{\beta}_i$ and this can again be obtained with local low rank corrections of $\mathbf{F}\mathbf{Z}_i$ using $(\mathbf{F}\mathbf{G})$ that has been computed at the initialization phase of S-FETI.

Optimization of the orthogonalization procedure Once a set of vectors $\mathbf{P}\mathbf{Z}_i$ 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, $(\mathbf{P}\mathbf{Z}_i)^T\mathbf{F}\mathbf{P}\mathbf{Z}_i$ 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

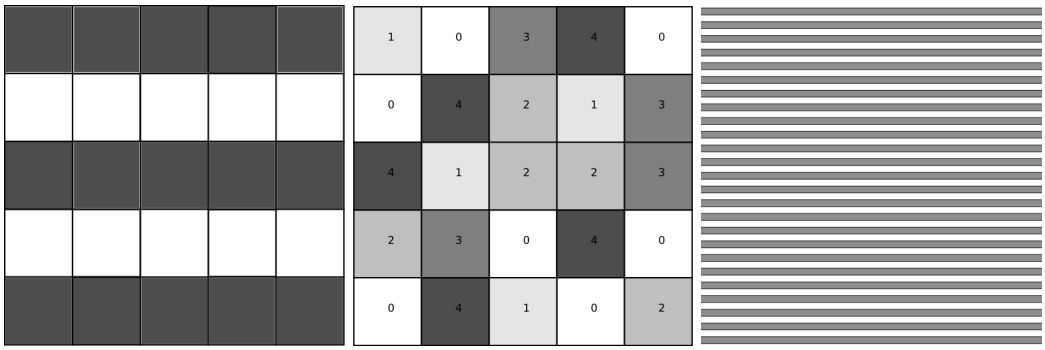


Figure 5: Three test cases on which we measure CPU times

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 three 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 5. We use either a regular checkerboard decomposition into 5×5 regular square subdomains each containing 80 000 degrees of freedom (the global problem is then 2M degrees of freedom large) or a decomposition into slices with 50 subdomains containing 200 000 degrees of freedom each (the global problem is then 10M degrees of freedom large). The subdomains are always homogeneous, but there exists a global heterogeneity pattern, with a maximal ratio of 10^4 in the Young modulus and constant Poisson ratio. This consists either of alternating layers of materials or of a random distribution where each subdomain is randomly affected a Young modulus of 1, 10, 10^2, 10^3 or 10^4. Note that this type of heterogeneous problems where the decomposition is chosen such that domains are homogeneous are usually well handled by the classical FETI methods [3] so these test case are not designed to be particularly favorable to our methods.

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 with a degree of parallelism of 4.

The FETI iterations are performed until both the relative error $\|\mathbf{u}_i - \mathbf{u}_{i-1}\|/\|\mathbf{u}_i\|$ and the relative residual $\|(\mathbf{Ku}_i - \mathbf{f})\|/\|\mathbf{f}\|$ for the global problem are smaller than 10^{-6}.

Decomp.	Hete.	Solver	#iterations	#search directions	Max #local resolutions	Time (s)
Checker	Layers	FETI	105	105	210	24
		S-FETI	39	975	273	24
Checker	Random	FETI	386	386	772	112
		S-FETI	102	2550	714	68
Slices	Layers	FETI	107	107	214	70
		S-FETI	2	100	10	3

Table 6: CPU performance of FETI and S-FETI for a 2D elasticity problem

Table 6 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 first case (checkerboard decomposition and alternating layers of material) S-FETI does not reduce the total time, but the effect of better efficiency with a block approach can already be seen since the global arithmetic complexity is higher but not the global time. Note also that the standard FETI requires as many Dirichlet and Neumann solutions whereas S-FETI requires much more Neumann than Dirichlet solutions, Neumann resolutions being slightly more expensive.

In the second case (checkerboard decomposition and random distribution of materials) which is more ill conditioned, the total number of solutions required by the S-FETI method is 10% smaller than with standard FETI, but the time is reduced by a factor of almost 2, thanks to better parallel efficiency, both of global MPI data transfers and at the local multi-core level.

In the last case (decomposition into slices and alternating layers of material), the S-FETI method performs extremely well. Note that the total number of search directions is almost the same for both methods but S-FETI requires much fewer local solutions. In this case the efficiency of the local solutions due to multiple forward-backward substitution is low because there are at most three local solutions to be performed at the same time.

To conclude, these first quantitative assessments show that S-FETI performs at least as well as the usual FETI method in terms of CPU time and in some cases significant performance improvements can be obtained (up to 23 times faster). Note that even better performance can be expected in 3D where the multiple forward-backward substitution is even more efficient.

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 and analyzed 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 summing them. The extra cost of the method is limited because sparsity can be exploited when searching 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] Farhat C, Roux FX. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering* 1991; **32**(6):1205, doi:10.1002/nme.1620320604.

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

[3] Bhardwaj M, Day D, Farhat C, Lesoinne M, Pierson K, Rixen D. 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* 2000; **47**(1-3):513–536.

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

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

[6] Farhat C, Rixen D. A new coarsening operator for the optimal preconditioning of the dual and primal domain decomposition methods: application to problems with severe coefficient jumps. *Proceedings of the Seventh Copper Mountain Conference on Multigrid Methods*, N Duane Melson SFM T A Manteuffel, DouglasM CC (eds.), 1995; 301–316.

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

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

[9] Klawonn A, Rheinbach O, Widlund OB. An analysis of a FETI–DP algorithm on irregular subdomains in the plane. *SIAM J. Numer. Anal.* 2008; **46**(5):2484–2504.

[10] Nataf F, Xiang H, Dolean V, Spillane N. A coarse space construction based on local Dirichlet-to-Neumann maps. *SIAM J. Sci. Comput.* 2011; **33**(4):1623–1642, doi:10.1137/100796376. URL http://dx.doi.org/10.1137/100796376.

[11] Dolean V, Nataf F, Scheichl R, Spillane N. Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet-to-Neumann maps. *Comput. Methods Appl. Math.* 2012; **12**(4):391–414, doi:10.2478/cmam-2012-0027. URL http://dx.doi.org/10.2478/cmam-2012-0027.

[12] Spillane N, Dolean V, Hauret P, Nataf F, Pechstein C, Scheichl R. Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. *Numer. Math.* 2014; **126**(4):741–770, doi:10.1007/s00211-013-0576-y.

[13] Galvis J, Efendiev Y. Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces. *Multiscale Model. Simul.* 2010; **8**(5):1621–1644, doi:10.1137/100790112. URL http://dx.doi.org/10.1137/100790112.

[14] Efendiev Y, Galvis J, Lazarov R, Willems J. Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms. *ESAIM Math. Model. Numer. Anal.* 2012; **46**(5):1175–1199, doi:10.1051/m2an/2011073. URL http://dx.doi.org/10.1051/m2an/2011073.

[15] Mandel J, Sousedík B. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Comput. Methods Appl. Mech. Engrg.* 2007; **196**(8):1389–1399, doi:10.1016/j.cma.2006.03.010. URL http://dx.doi.org/10.1016/j.cma.2006.03.010.

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

[17] Spillane N. 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.

[18] Rixen D. 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.

[19] Gosselet P, Rixen DJ, Rey C. A domain decomposition strategy to efficiently solve structures containing repeated patterns. *International Journal for Numerical Methods in Engineering* 2009; **78**(7):828–842, doi:10.1002/nme.2517.

[20] Farhat C, Pierson K, Lesoine M. 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* 2000; **184**(2-4):333–374, doi:10.1016/S0045-7825(99)00234-0.

[21] Farhat C, Lesoinne M, LeTallec P, Pierson K, Rixen D. 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* 2001; **50**(7):1523–1544.

[22] Rixen DJ, Farhat C, Tezaur R, Mandel J. 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* 1999; **46**(4):501–533, doi:10.1002/(SICI)1097-0207(19991010)46:4⟨501::AID-NME685⟩3.0.CO;2-7.

[23] Farhat C, Mandel J. 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* 1998; **155**:129–152.

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

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

[26] Rey V, Rey C, Gosselet P. 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* 2013; **270**(1):293–303.

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

[28] Saad Y. *Iterative methods for sparse linear systems*. Second edn., Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, 2003, doi:10.1137/1.9780898718003.

[29] Greif C, Rees T, Szyld DB. 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.

[30] Greif C, Rees T, Szyld DB. Additive Schwarz with variable weights. *Technical Report 12-11-30*, Department of Mathematics, Temple University Nov 2012.

[31] O'Leary DP. The block conjugate gradient algorithm and related methods. *Linear Algebra Appl.* 1980; **29**:293–322, doi:10.1016/0024-3795(80)90247-5.

[32] Nikishin AA, Yeremin AY. 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.* 1995; **16**(4):1135–1153, doi:10.1137/S0895479893247679.

[33] Brandt A, Brannick J, Kahl K, Livshits I. Bootstrap AMG. *SIAM J. Sci. Comput.* 2011; **33**(2):612–632, doi:10.1137/090752973.

[34] Karypis G, Kumar V. 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.

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