



HAL
open science

Recycling of Solution Spaces in Multi-Preconditioned FETI Methods Applied to Structural Dynamics

Michael Leistner, Pierre Gosselet, Daniel Rixen

► **To cite this version:**

Michael Leistner, Pierre Gosselet, Daniel Rixen. Recycling of Solution Spaces in Multi-Preconditioned FETI Methods Applied to Structural Dynamics. 2018. hal-01687130v1

HAL Id: hal-01687130

<https://hal.science/hal-01687130v1>

Preprint submitted on 18 Jan 2018 (v1), last revised 13 Jun 2018 (v2)

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

Recycling of Solution Spaces in Multi-Preconditioned FETI Methods Applied to Structural Dynamics

Michael Leistner^{1,4}, Pierre Gosselet², Daniel J. Rixen¹

¹ Technical University of Munich, Chair of Applied Mechanics, Boltzmannstr. 15,
D-85748 Garching, Germany

² LMT-Cachan/ENS-Cachan, CNRS, Université Paris-Saclay, 61 avenue du président Wilson,
Cachan 94235, France

Abstract

This article presents a new method to recycle the solution space of an adaptive multi-preconditioned finite element tearing and interconnecting (AMP-FETI) algorithm in the case where the same operator is solved for multiple right-hand sides like in linear structural dynamics. It accelerates the computation from the second time step on by applying a coarse-space that is generated from Ritz approximations of local eigenproblems, using the solution space of the first time step. These eigenproblems are known to provide very efficient coarse-spaces but must usually be solved a priori at high computational cost. Their Ritz approximations are much smaller and less expensive to solve. Recycling methods based on Ritz approximations of global eigenproblems have been published for classical FETI algorithms but their efficient application to multi-preconditioned variants is not possible. The article also presents the application of a simpler recycling procedure, which reuses plain solution spaces, to AMP-FETI. Numerical results of the application of the presented methods to four test cases are shown. The new Ritz approximation method leads to coarse-spaces which turn out to be as efficient as those obtained from solving the unreduced eigenproblems. It is the most efficient recycling method currently available for multi-preconditioned dual domain decomposition techniques.

KEY WORDS: Feti; adaptive; multipreconditioned; recycling; geneo; Ritz vectors

1 Introduction

The finite element tearing and interconnecting algorithm, FETI for short, was originally published in [1]. Based on a non-overlapping decomposition of the domain into substructures, the problem is expressed only in terms of the Lagrange multipliers that connect the substructures. The resulting smaller set of equations is called interface problem. Solving it iteratively, e.g. by a conjugate gradient, results in a highly parallelizable algorithm which exhibits advantageous spectral properties. The singularity of the stiffness matrices of floating substructures in static problems introduces a type of coarse grid correction which implicates global propagation of error information and is thus responsible for the good scalability of the algorithm as shown in [2].

While the algorithm works well for homogeneous structures whose mesh is decomposed in substructures with smooth boundaries and balanced aspect ratios, several factors can cause bad conditioning of the linear problem to be solved. In the case of heterogeneities where the coefficients, i.e. stiffness, exhibit jumps from one substructure to another, a scaling method presented in [3] greatly improves the performance. However, slow convergence or even breakdown of the algorithm still results when the boundaries of such jumps in material coefficients cross the boundaries of substructures. As demonstrated in [4], this can sometimes be circumvented by choosing the decomposition aligned with the material distribution, though this solution might lead to very bad aspect ratios which again harm the performance.

A further enhancement of the dual substructuring methods was the introduction of arbitrarily chosen deflation spaces in [5] and [6], which were already known for conjugate gradient algorithms before [7]. The solution procedure then involves another coarse problem similar to those originating from rigid body modes in statics. The deflation space is usually called coarse space in the context of FETI algorithms. An a priori computed coarse space, based on the solution of local eigenproblems, was introduced in [8], called Generalized Eigenvalues in the Overlaps, abbreviated as GenEO. It identifies the so-called bad modes and effectively removes high, separated eigenvalues from the spectrum of the linear operator, so that robustness and fast convergence can be restored well even for strong, randomly distributed heterogeneities. Later, an alternative method, building on an idea originally proposed in [9], was published in [10]. Opposed to GenEO, no a-priori calculation is carried out. Instead, the standard conjugate gradient algorithm is extended to a multi-preconditioned conjugate gradient, exploiting the

⁴Correspondence to: Michael Leistner, m.leistner@tum.de

additive structure of the preconditioner. This algorithm, originally called simultaneous FETI or S-FETI, exhibits similar robustness and convergence as GenEO. In [11], it was found to belong to the class of multi-preconditioned Krylov solvers [12].

The FETI method was already applied to linear structural dynamics, e.g. in [13] and [14]. Different from static problems, no coarse grid correction is introduced automatically in the space of rigid body modes. Therefore, a coarse grid correction must be introduced artificially when applying the FETI method to dynamic problems in order to maintain scalability. A distinctive feature of linear structural dynamics is that the matrix, which describes the system of equations to be solved, remains constant while the right-hand side changes from one time step to the next. This suggests reusing the information gathered during a solution process, similar to what direct solvers can do by storing the factorization of the matrix. Of course, this is not possible for FETI solvers because they work iteratively and the global problem is never assembled but instead remains distributed over the individual substructures. The general case of solving a problem on the same structure for multiple right-hand sides has been addressed in [15] and [16] by exploiting the potential of a standard conjugate gradient algorithm to efficiently reuse previously built Krylov spaces. This recycling strategy of reusing the plain, unprocessed solution spaces was also applied in the case of changing operators when solving nonlinear problems in [17] and [18]. That certain eigenvectors of the system comprise an ideal coarse space for a conjugate gradient algorithm was already described in [7]. These techniques were refined in the context of domain decomposition in several publications like [19], [20] and [21] by approximating eigenvectors of the preconditioned operator inside the generated Krylov space, based on a Ritz ansatz similar to the Lanczos procedure.

Unfortunately, this global Ritz approximation breaks down if multi-preconditioning is employed because the actual preconditioner changes from one iteration to another and short recurrences of the classical conjugate gradient as well as important orthogonality relations do not hold any longer. The authors know no publications about the application of either established or newly developed recycling strategies to multi-preconditioned algorithms. However, as the multi-preconditioning is able to precisely consider local effects of substructure behavior instead of blurring them as it happens in the classical FETI methods, the generated solution spaces should be excellently suited to capture the local bad-modes, necessary to accelerate the iterative process. This article presents a method that allows accessing this potential by projecting the GenEO eigenproblems on the solution-space generated in the first time step, leading to much smaller eigenproblems that require a minimum of computational cost to be assembled and solved. From their solutions, a highly efficient coarse space is constructed. Finally, one more method is presented that applies the idea of reusing plain, unprocessed solution spaces as coarse space, like it was done for single-preconditioned procedures in [15, 16, 17, 18, 20, 21], to AMP-FETI.

The structure of the article is organized as follows. First, the discrete, linear dynamic equations and their decomposition in substructures as well as the applied time stepping scheme are introduced in section two. In section three, the basic framework of adaptive multi-preconditioning for FETI is explained. The methods to construct coarse spaces by recycling solution spaces are presented in section four, and their numerical results in section five. The conclusion, the references, and additional information on the algorithm and results in the appendix complete the article.

2 Dual Domain Decomposition for Linear Dynamics

The linear dynamics of a mechanical structure can be described by the system of equations

$$M\ddot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}(t).$$

with a mass matrix M and a stiffness matrix \mathbf{K} . The structure is loaded by an arbitrary time dependent force $\mathbf{f}(t)$. The values of the initial displacement $\mathbf{u}(0)$ and the initial velocity $\dot{\mathbf{u}}(0)$ are given. To simplify the formulas, the time dependency of $\ddot{\mathbf{u}}$, $\dot{\mathbf{u}}$, \mathbf{u} and \mathbf{f} is omitted from here on. The domain Ω is now decomposed into N_s non-overlapping subdomains Ω^s , $s = 1, \dots, N_s$, called substructures in the context of FETI. After decomposition, the local substructure equations and the compatibility constraints imposed on the accelerations write

$$\begin{aligned} M^s \ddot{\mathbf{u}}^s + \mathbf{K}^s \mathbf{u}^s + \mathbf{t}^{sT} \mathbf{B}^{sT} \boldsymbol{\lambda} &= \mathbf{f}^s \\ \sum_{s=1}^{N_s} \mathbf{B}^s \mathbf{t}^s \ddot{\mathbf{u}}^s &= \mathbf{0} \end{aligned}$$

The local substructure equations contain Lagrange multipliers $\boldsymbol{\lambda}$ which enforce the compatibility constraints, i.e. they make sure that the displacement field of connected substructures is the same on their

interface. The constraints are formulated with signed Boolean operators B^s , also called dual assembly operators, and trace operators t^s . The constraints are imposed on the accelerations to enable spectral stability even for time-integration schemes without numerical dissipation [22].

After discretization in time by applying the Newmark Beta scheme [23] and [24], the equations to be solved write

$$D^s \ddot{u}^{s,n+1} = g^{s,n+1} - t^{sT} B^{sT} \lambda \quad (1)$$

$$\sum_{s=1}^{N_s} B^s t^s \ddot{u}^s = 0 \quad (2)$$

where n denotes the current time step index and

$$D^s = M^s + h^2 \beta K^s$$

$$g^{s,n+1} = f^{s,n+1} - K^s \left[u^{s,n} + h \dot{u}^{s,n} + h^2 \left(\frac{1}{2} - \beta \right) \ddot{u}^{s,n} \right]$$

The matrix D^s is called stepping matrix. The parameters γ and β determine the Newmark Beta time stepping scheme. From the equations (1) and (2), the interface problem

$$F \lambda = d \quad (3)$$

can be built, where

$$F = \sum_{s=1}^{N_s} F^s = \sum_{s=1}^{N_s} B^s S^{s-1} B^{sT} \quad \text{and} \quad d = \sum_{s=1}^{N_s} B^s t^s D^{s-1} g^s$$

using the Schur complement S^s which is the stepping matrix D^s of substructure s condensed on its interface degrees of freedom. Note that D^s is always symmetric positive definite because of the presence of the mass matrix M^s . This is related to the fact that rigid body modes still possess kinetic energy. Consequently, the coarse space projection, which is mandatory in FETI for static problems to solve the problem in the nullspace of the operator, vanishes. However, an auxiliary coarse space projection, resulting in a FETI-2 algorithm, can still be applied [25]. We apply the well known Dirichlet preconditioner

$$H = \sum_{s=1}^{N_s} H^s = \sum_{s=1}^{N_s} \tilde{B}^s S^s \tilde{B}^{sT}$$

which was introduced for FETI in [25]. We further use scaled dual assembly operators

$$\tilde{B}^s = \beta^s B^s$$

with diagonal scaling matrices β^s . In what follows, the so-called stiffness or superlumped scaling is applied. Originally developed and published in [3] for static problems, this procedure defines a mechanically consistent preconditioner by scaling the gaps and the estimated correction forces not only based on the multiplicity, but on the stiffnesses of the interface degrees of freedom. It is adapted to dynamic problems here by using the coefficients of the stepping matrix D^s instead of the stiffness coefficients.

3 Adaptive Multi-Preconditioned FETI

This section shortly describes the deflated, adaptive multi-preconditioned FETI algorithm on which all presented methods in this paper are built on. In general, the algorithm is a preconditioned conjugate gradient that is deflated by a projection step. Because of the rapid loss of orthogonality in the presence of large, well-separated eigenvalues [26], full orthogonalization is carried out in every iteration as it was suggested for domain decomposition methods in [27]. Exploiting the additive structure of the preconditioner then leads to a multi-preconditioned conjugate gradient. The procedure as explained in this section is listed in Algorithm 1.

The adaptive multi-preconditioning is based on a classical FETI algorithm which solves the interface problem (3) by applying a preconditioned conjugate gradient procedure. With the residual in iteration i being defined as

$$\mathbf{r}_i = \mathbf{d} - \mathbf{F}\boldsymbol{\lambda}_i$$

the preconditioning step of a standard or single-preconditioned FETI algorithm computes the search direction \mathbf{z}_i as

$$\mathbf{z}_i = \mathbf{H}\mathbf{r}_i = \sum_{s=1}^{N_s} \mathbf{H}^s \mathbf{r}_i$$

by summing up individual contributions from every substructure. Based on the idea to compute the optimal step direction not only in the space of one but multiple search directions which was originally proposed in [9], the simultaneous FETI or S-FETI as published in [10] considers these individual contributions as independent search directions

$$\begin{aligned} \mathbf{z}_i^s &= \mathbf{H}^s \mathbf{r}_i \\ \mathbf{Z}_i &= \left[\mathbf{z}_i^1 \mid \mathbf{z}_i^2 \mid \dots \mid \mathbf{z}_i^{N_s} \right] \end{aligned}$$

They span a space inside which the energy of the solution is minimized. To keep the computational costs of this minimization step low, the algorithm has been enhanced by an adaptive selection criterion in [11] where it was also classified as a multi-preconditioned conjugate gradient. The adaptive selection is based on the τ -criterion, which selects only certain local contributions \mathbf{z}_i^s as basis vectors of the minimization space. First, the measure

$$\Xi_i^s = \frac{\boldsymbol{\alpha}_i^T \mathbf{W}_i^T \mathbf{F}^s \mathbf{W}_i \boldsymbol{\alpha}_i}{\mathbf{r}_{i+1}^T \mathbf{H}^s \mathbf{r}_{i+1}}$$

is defined for each substructure in each iteration. The columns of \mathbf{W}_i are the individual step directions and $\boldsymbol{\alpha}_i$ the corresponding step lengths used in iteration i to build the adaption of $\boldsymbol{\lambda}_i$. The measure Ξ_i^s then compares the energy $\boldsymbol{\alpha}_i^T \mathbf{W}_i^T \mathbf{F}^s \mathbf{W}_i \boldsymbol{\alpha}_i$ of the taken step in iteration i with the energy $\mathbf{r}_{i+1}^T \mathbf{H}^s \mathbf{r}_{i+1}$ of the resulting error, both localized to substructure s . Furthermore, a tolerance τ is defined by the user and the individual contributions \mathbf{z}_i^s of all those substructures s which fulfill the criterion

$$\Xi_i^s < \tau$$

are selected as individual basis vectors of the minimization space. This selection is based on a convergence estimate that makes an educated guess in which parts of the global domain convergence can be accelerated by the extra cost of solving the minimization problem. Several examples have been analyzed in [11] and [28] concerning the optimal choice of τ . A value of 0.1 has shown to be a reasonable choice that leads to robust behavior in the majority of cases.

Let $J = (j_1, j_2, \dots)$ be the numbers of the selected substructures for a specific iteration i . The remaining directions are summed up to a single direction so that the actual minimization space in iteration i is then spanned by the columns of

$$\mathbf{Z}_i = \left[\sum_{s \notin J} \mathbf{z}_i^s \mid \mathbf{z}_i^{j_1} \mid \mathbf{z}_i^{j_2} \mid \dots \right]$$

The procedure is furthermore augmented by a projection, also called deflation. This means effectively that in a certain, well-chosen subspace \mathbf{C} , the solution is calculated a priori, typically using a direct

solver like described in [29]. During the iterations, all columns of W_i must be made F -conjugate not only to themselves but also to the chosen coarse space C such that

$$C^T F W_i = \mathbf{0}$$

holds for all i . This can be realized by applying a projection

$$W_i \leftarrow P_C W_i$$

using the projector

$$P_C = I - C(C^T F C)^{-1} C^T F$$

Solving the coarse problem described by the operator $C^T F C$ introduces global communication within one iteration, while in all other operations only neighbors need to exchange information. Such kind of global communication, meaning propagation of information throughout the whole domain, is absolutely indispensable for this class of algorithms to be scalable. For that reason, almost any FETI method is equipped with a projection or some other type of coarse grid correction.

Algorithm 1: Deflated Adaptive Multi-Preconditioned FETI

Let the columns of C span the deflation space.

$$P_C = I - C(C^T F C)^{-1} C^T F$$

$$\lambda_0 = C(C^T F C)^{-1} C^T d$$

$$r_0 = d - F \lambda_0$$

for $s = 1, \dots, N_s$ **do**

$$\quad \lfloor z_0^s = H^s r_0$$

$$Z_0 = [z_0^1 | z_0^2 | \dots | z_0^{N_s}]$$

$$W_0 = P_C Z_0$$

$i = 0$

while $\sqrt{\|r_i^T H r_i\|} < \epsilon$ **do**

$$\quad Q_i = F W_i = F Z_i - F C(C^T F C)^{-1} C^T F Z_i - \sum_{j=0}^{i-1} Q_j \beta_{i,j}$$

$$\quad \Delta_i = Q_i^T W_i \quad \gamma_i = W_i^T r_i \quad \alpha_i = \Delta_i^+ \gamma_i$$

$$\quad \lambda_{i+1} = \lambda_i + W_i \alpha_i$$

$$\quad r_{i+1} = r_i - Q_i \alpha_i$$

$$Z_{sum} = 0 \quad Z_{i+1} = \emptyset$$

for $s = 1, \dots, N_s$ **do**

$$\quad \lfloor z_{i+1}^s = H^s r_{i+1}$$

$$\quad \Xi_i^s = \frac{(W_i \alpha_i)^T F^s W_i \alpha_i}{r_{i+1}^T H^s r_{i+1}}$$

if $\Xi_i^s < \tau$ **then**

$$\quad \lfloor Z_{i+1} \leftarrow [Z_{i+1} | z_{i+1}^s]$$

else

$$\quad \lfloor Z_{sum} \leftarrow Z_{sum} + z_{i+1}^s$$

$$Z_{i+1} \leftarrow [Z_{i+1} | Z_{sum}]$$

$i \leftarrow i + 1$

for $0 \leq j \leq i - 1$ **do**

$$\quad \lfloor \Phi_{i,j} = Q_j^T Z_i \quad \beta_{i,j} = \Delta_j^+ \Phi_{i,j}$$

$$\quad \lfloor W_i = P_C Z_i - \sum_{j=0}^{i-1} W_j \beta_{i,j}$$

$m = i$

What must be considered is the possible degeneration of the minimization space basis, i.e. a linear dependency of the chosen basis vectors. While such degeneration is very unlikely in statics or fully excited structures in dynamics, it happens usually in the first few time steps of a dynamic problem where the initial configuration of the structure is a state of equilibrium and external load is applied only on a small portion of the whole domain [30]. In this case, local parts of the residual and thus certain directions z^s can become exactly or numerically zero. It was demonstrated in [30] that the τ -criterion has the tendency to select these zero-directions.

To handle this problem, a pseudo inverse Δ^+ is introduced and built by removing linearly dependent directions from W_i and Q_i . As suggested in [10], this is done by applying a rank revealing LDL^T decomposition

$$\Delta_i = PLAL^T P^T \quad (4)$$

with a rectangular permutation matrix P , a full rank diagonal matrix A , and a lower triangular matrix L . The algorithm then proceeds with the substitutions

$$W_i \leftarrow W_i PL^{-T} \quad Q_i \leftarrow Q_i PL^{-T} \quad \Delta_i^+ = A^{-1} \quad \alpha_i = A^{-1}(PL^{-T})^T \gamma \quad (5)$$

The construction of Δ_i implies global communication and thereby allows for good scalability, just as a coarse grid correction by projection does.

4 Recycling of Solution Spaces

When applying a time stepping scheme to solve a problem of linear dynamics, the stepping matrices D^s remain constant over the time steps and so do the operator F and the preconditioner H . Thus, the same operator with changed right-hand side needs to be solved repeatedly. Several publications have addressed the need to recycle the gathered information from former solves to accelerate the future ones and appropriate procedures were proposed. They all have in common that they build up a coarse space which is subsequently removed from the iterative solution process, either by means of deflation through projection or by re-conjugation, usually depending on whether a basis orthonormal in the metric of the operator is available or not.

However, all known procedures of this kind were designed for single-preconditioned conjugate gradient methods. In this section, two techniques to realize solution space recycling for multi-preconditioning are introduced. In Section 4.3, the idea from single-preconditioning to reuse solution spaces without any additional processing is applied to multi-preconditioning. In Section 4.4, a new method that is based on a local Ritz approximation is presented.

4.1 Activation of the whole Structure in Time Step One

Before developing methods to reuse solution spaces, an important issue must be considered first. As already addressed before, the adaptive multi-preconditioned FETI algorithm is likely to produce search directions close to zero in the first time step because the deformation has not yet spread throughout the whole domain. This typically occurs in soft structures with localized loads and small time steps. In case of such degeneration of the search space, the performance of any method to reuse the solution space will drop. This happens, because it could not collect the necessary information about the inactive parts of the structure, while this information becomes mandatory in later time steps when those parts of the structure are activated.

A simple and obvious solution to this problem is to activate the recycling strategy at a later time step, but of course, this includes the significant drawback that several solutions must be computed without or at least with only a suboptimal coarse space. A different way to make the method robust against such difficulties is suggested here, independent of the mentioned influencing factors. Instead of using a zero vector as start value for λ , the vector λ_0 is constructed by putting loads with alternating directions on the interfaces so that also high frequencies are excited.

For example, if λ_0 contains the degrees of freedom of the interface nodes in sequential order, beginning with substructure one, it is set to

$$\lambda_0 = \lambda_{\text{activate}} = \begin{bmatrix} \lambda_{1,x}^1 \\ \lambda_{1,y}^1 \\ \lambda_{2,x}^1 \\ \lambda_{2,y}^1 \\ \lambda_{3,x}^1 \\ \vdots \end{bmatrix} = \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \\ +1 \\ \vdots \end{bmatrix} \quad (6)$$

In the case of a consecutive numbering of the interface nodes, this leads to an interface loading of substructure one as shown in Figure 1. A random initialization of λ , like it was applied in [10] for the

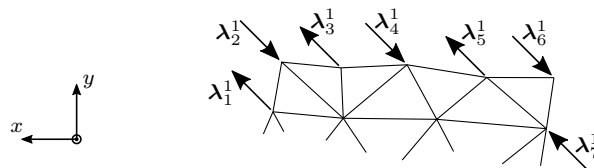


Figure 1. Initialization of λ

same reason, is also possible and leads to similar results but was not applied here in order to achieve maximum comparability and reproducibility.

Using such an initialization technique, the structure is already fully activated in the first time step. We observe a slight increase of iterations for the first time step in that case, but it is considered negligible compared to the advantage gained. What has become evident during the experiments, is that the magnitude η of that initial excitation, defined by the relation

$$\eta = \frac{\|\mathbf{F}\boldsymbol{\lambda}_0\|}{\|\mathbf{d}\|}$$

is crucial. If the magnitude is chosen too high, for example $\eta = 1$, the performance of the method drops. In all experiments, η is set to 0.05 leading to almost the same performance as if the recycled solution space had been taken from a later time step when the structure is fully activated naturally. This ratio of magnitudes can be achieved by first building a $\boldsymbol{\lambda}_0$ with the desired structure and arbitrary magnitude, e.g. filling it with +1 and -1 like in (6) and after computation of $\mathbf{F}\boldsymbol{\lambda}_0$, the result as well as $\boldsymbol{\lambda}_0$ are scaled such that $\eta = 0.05$ holds. This modified initialization phase is listed in detail in Algorithm 2 in Appendix A.1.

4.2 Desirable Coarse Spaces

The coarse space that is built within any recycling procedure should satisfy essentially two requirements. On the one hand, it should be as small as possible and on the other hand, chosen such that the convergence of the conjugate gradient, finding the solution in the remaining space, is as fast as possible. The convergence behavior of the preconditioned conjugate gradient algorithm is mainly dependent on the eigenspectrum of the preconditioned operator $\mathbf{H}\mathbf{F}$. This was thoroughly analyzed and demonstrated for the particular case of the dual Schur complement problem in [31]. If it does not converge quickly towards the solution, this is most likely due to the high part of the eigenspectrum, which typically consists of few high, well-separated eigenvalues. They often originate from local effects, which are accordingly called bad modes in this context. Once they have been captured during the solution process of the conjugate gradient, the rate of convergence increases as shown extensively in [32]. When the corresponding eigenvectors are removed from the solution process by deflation or re-conjugation, the conjugate gradient behaves as if they were not present in the spectrum. This can be interpreted by reducing the so-called effective condition number, and thereby accelerating convergence. Consequently, it would be desirable to find a coarse space that is as close as possible to the eigenvectors of the high, separated part of the spectrum.

4.3 Plain Reuse of Solution Spaces

First, the simplest approach is the plain reuse of the solution space built during the conjugate gradient iterations, that is to say, to employ that solution space completely or partially as coarse space without any further processing. This approach has been applied to classical single-preconditioned algorithms in [15, 16, 17, 18, 20, 21], where the solution space is a Krylov space, constructed by applying the preconditioned operator repeatedly. This suggests that the eigenvectors of few high, well-separated eigenvalues are captured well in the solution space. Thus, using the whole solution space of a former time step as coarse space seems a reasonable but possibly inefficient procedure.

The solution space of the multi-preconditioned methods is not a Krylov space anymore, and the preconditioner varies effectively from one iteration to another. However, the bad modes, as they originate from local effects, are likely to be captured well in the individual contributions \mathbf{W}_i . Furthermore, when the rank revealing LDL^T decomposition is applied as in Equations (4) and (5), all columns of all \mathbf{W}_i are each \mathbf{F} -conjugate to one another. Consequently, using them to build the coarse space, results in a diagonal coarse problem

$$\left(\mathbf{C}^T \mathbf{F} \mathbf{C}\right) \mathbf{x} = \mathbf{y} \quad \text{with} \quad \mathbf{C}^T \mathbf{F} \mathbf{C} = \text{diag}$$

that must be solved each time the projector \mathbf{P}_C is applied and thus reduces computational cost. This method of plain solution space reuse can be applied to multi-preconditioning by setting

$$\mathbf{C} \leftarrow [\mathbf{C} \mid \mathbf{W}_0 \mid \mathbf{W}_1 \mid \dots \mid \mathbf{W}_{m-1}]$$

at the end of a time step when the algorithm has converged after m iterations. If the target coarse space size is not reached in the first time step, the coarse space is extended further in the second time step.

The method has the potential to deliver more directions than a reasonable size of the coarse space would allow, even in the first time step. Therefore, the matrix \mathbf{C} is in fact chopped such that only the first

or left columns, corresponding to the earlier iterations, form the coarse space. This automatically implies a reasonable selection, as the well-separated extremal eigenvalues in the high part of the spectrum are likely to be captured in these earlier iterations. Also the sorting of the columns within the \mathbf{W}_i matrices might become important. They are sorted in descending order with respect to the corresponding pivot elements in the rank revealing LDL^T decomposition used to compute Δ^+ and α , and should thus be roughly sorted in the order of their importance for the solution.

4.4 Approximation of Local Eigenproblems

The unprocessed reuse of solution spaces is easy to implement and no computational extra cost is required to build the coarse space bases. However, the number of generated directions is large and the efficiency gain, that is the computational cost saved in relation to the additional cost caused by the growth of the coarse problem, might be low. In this section, a new method is presented. It is based on the idea to extract the essence out of the available spaces at the extra cost of approximating certain eigenproblems.

In [8], a local generalized eigenvalue problem called generalized eigenvalues in the overlaps, GenEO for short, was proposed. It is designed to compute the bad modes, slowing down the convergence. When its solutions are employed as coarse space, it effectively removes the high, extremal part of the spectrum of the deflated, preconditioned operator. For every substructure s , the generalized eigenproblem writes

$$\mathbf{S}^s \mathbf{y}^s = \Theta^s \mathbf{B}^{sT} \mathbf{H} \mathbf{B}^s \mathbf{y}^s \quad (7)$$

The coarse space is then chosen as

$$\mathbf{C} = [\mathbf{C}^1 | \mathbf{C}^2 | \dots | \mathbf{C}^{N_s}] \quad \text{where} \quad \mathbf{C}^s = [\mathbf{H} \mathbf{B}^s \mathbf{y}_1^s | \mathbf{H} \mathbf{B}^s \mathbf{y}_2^s | \dots | \mathbf{H} \mathbf{B}^s \mathbf{y}_{k^s}^s]$$

The contribution \mathbf{C}^s to the coarse space of a substructure s should in general be built from the directions \mathbf{y}_i^s with the k^s smallest eigenvalues Θ_i^s . An appropriate global threshold for Θ can be found by setting an explicit limit on the coarse space size. Another approach that is known to be very efficient is to search for a significant jump in the GenEO eigenvalues Θ_i^s of each substructure, when sorted in ascending order. The matrix \mathbf{C}^s of substructure s is then constructed by considering all eigenvalues before the jump. We make use of both methods here and indicate this by (j) for the jump-criterion and (n) for an explicitly set coarse space size limit.

As it is a generalized eigenvalue problem involving the Schur complements of the substructure itself and its neighbors, the computational cost of its solution is significant although easily parallelizable as proposed in [33]. The GenEO coarse space results in robust and fast convergence for various types of difficulties like arbitrary heterogeneities, jagged interfaces and bad aspect ratios, but it would be favorable to reduce the cost of its computation by making use of the information gathered once the linear system has been solved for the first time. Considering only the cost of computing the first time step, a multi-preconditioned FETI method is more efficient than computing an a priori coarse space as expensive as GenEO. However, it would be a waste of resources to ignore all the information gathered when proceeding with the next time steps.

4.4.1 Construction of a Local Ritz Ansatz

As remarked in [10], the multi-preconditioned FETI method bears a close resemblance to the GenEO deflated FETI method in the sense that both are able to detect local effects which are very important for the iterative solution. Where the standard preconditioner blurs the important local modes by simply summing up the substructure contributions, the multi-preconditioned FETI computes their optimal combination $\mathbf{W}_i \alpha_i$. Therefore, the space spanned by the solution increments $\mathbf{W}_i \alpha_i$ is likely to capture the local bad modes in some way.

On the one hand, the local eigenproblems of Equation (7) deliver the desired local bad modes but are large and costly to assemble. On the other hand, with the basis $[\dots \mathbf{W}_i \alpha_i \dots]$ a small space, able to represent the local bad modes within a certain accuracy, is available. The logical consequence is to construct a Ritz space for each substructure that provides test and trial functions for approximating the corresponding local eigenproblems of Equation (7). This results in a smaller eigenproblem, whose solutions are Ritz approximations of GenEO eigenvectors and -values.

While numerous variants are possible to construct such a Ritz space in which the GenEO eigenvectors are sought, it turned out that this is the most critical step. We construct the Ritz space for a specific

substructure s as

$$\mathbf{V}^s = (\mathbf{S}^s)^{-1} \mathbf{B}^{sT} \mathbf{V}_W^s \quad \text{where} \quad \mathbf{V}_W^s = [\mathbf{W}_0 \alpha_0 \mid \mathbf{W}_1 \alpha_1 \mid \dots \mid \mathbf{W}_{n^s-1} \alpha_{n^s-1}]$$

Because a conjugate gradient algorithm captures the outliers in the operator spectrum i.e. the bad modes first, only the solution space generated in the first n^s iterations is considered. The parameter n^s , determining the size of the Ritz space, is chosen individually for each substructure using a criterion explained later. The Ritz space size is limited by the relation $n^s \leq m$. The Ritz ansatz to approximate Equation (7) for substructure s then writes

$$\mathbf{V}^{sT} \mathbf{S}^s \mathbf{V}^s \mathbf{q}^s = \Theta^s \mathbf{V}^{sT} \mathbf{B}^{sT} \mathbf{H} \mathbf{B}^s \mathbf{V}^s \mathbf{q}^s$$

respectively

$$\mathbf{V}_W^{sT} \underbrace{\mathbf{B}^s \mathbf{S}^{s-1} \mathbf{S}^s \mathbf{S}^{s-1} \mathbf{B}^{sT}}_{=\mathbf{F}^s} \mathbf{V}_W^s \mathbf{q}^s = \Theta^s \mathbf{V}_W^{sT} \underbrace{\mathbf{B}^s \mathbf{S}^{s-1} \mathbf{B}^{sT}}_{=\mathbf{F}^s} \mathbf{H} \underbrace{\mathbf{B}^s \mathbf{S}^{s-1} \mathbf{B}^{sT}}_{=\mathbf{F}^s} \mathbf{V}_W^s \mathbf{q}^s$$

and can finally be expressed in the short form

$$\mathbf{V}_W^{sT} \mathbf{F}^s \mathbf{V}_W^s \mathbf{q}^s = \Theta^s \mathbf{V}_W^{sT} \mathbf{F}^s \mathbf{H} \mathbf{F}^s \mathbf{V}_W^s \mathbf{q}^s \quad (8)$$

Consequently, the coarse space is assembled as

$$\mathbf{C} = [\mathbf{C}^1 \mid \mathbf{C}^2 \mid \dots \mid \mathbf{C}^{N_s}]$$

where

$$\mathbf{C}^s = [\mathbf{H} \mathbf{F}^s \mathbf{V}_W^s \mathbf{q}_1^s \mid \mathbf{H} \mathbf{F}^s \mathbf{V}_W^s \mathbf{q}_2^s \mid \dots \mid \mathbf{H} \mathbf{F}^s \mathbf{V}_W^s \mathbf{q}_{k^s}^s]$$

This results in a coarse space size of

$$\text{rank}(\mathbf{C}) = \sum_{s=1}^{N_s} k^s$$

while $k^s \leq n^s$ must hold. In the Appendix A.2, a reorganization of the algorithm is presented that allows to assemble the reduced eigenproblem in Equation (8) without any further application of the local operators \mathbf{F}^s . Only a small number of applications of the local preconditioners \mathbf{H}^s between neighboring substructures and a few vector products remain to be computed. The results show that Ritz space sizes n^s using only a small fraction of the available m iterations already lead to very satisfying results. This is desirable as smaller Ritz spaces reduce the number of additionally required applications of \mathbf{H}^s . Furthermore, it is sufficient in linear structural dynamics to solve the reduced eigenproblems of Equation (8) of size n^s only once after the first time step to obtain a coarse space that works efficiently for the rest of the simulation. Therefore, the computational cost is considered negligible.

What remains to be determined is the number of iterations n^s that should be included in the Ritz space \mathbf{V}_W^s for a specific substructure, as well as the number of computed Ritz vectors k^s that should be selected from the solution of the reduced eigenproblem of a substructure s to be included into the coarse space.

4.4.2 Selection of Basis for Ritz Space

To determine the size of the Ritz space n^s , i.e. how many of the first iterations should be included, it is reasonable to consider the behavior of the τ -criterion. When it selects the local contribution of a substructure as individual basis vector for the minimization space, it means that the convergence was bad in this region because the conjugate gradient was supposedly busy catching high, well-separated eigenvalues. Consequently, it can be assumed that those search directions are very important to approximate the corresponding bad modes that should be included in the coarse space. Contrary to that, when the τ -criterion stops selecting individual contributions in later iterations, the conjugate gradient is finding parts of the solution that correspond to lower, clustered eigenvalues whose eigenvectors would create a very inefficient coarse space.

Altogether, this suggests to include iterations up to the point where the τ -criterion remains inactive for a certain substructure and its neighborhood. Precisely speaking, all those iterations are included into the Ritz space for a specific substructure until neither the substructures own, nor one of its neighbors search space contributions is selected individually by the τ -criterion.

4.4.3 Selection of Basis for Coarse Space

To choose for each substructure individually how many of the computed GenEO vector approximations $V_W^s q_i^s$ will be included in the coarse space, i.e. to determine k^s , the same methods as for GenEO can be employed. With the jump criterion, the Ritz approximation works completely autonomously. For better comparability with the results of the GenEO method, a fixed coarse space size equal to that of the GenEO method can be prescribed, selecting globally the lowest Ritz values. In everything what follows, the jump method is indicated by (j) and the fixed coarse grid size by (n).

5 Numerical Results

The previously described methods are applied to four different test cases. An overview of their setup and the summarized numerical results are presented in Appendix B in tabular form.

In this section, the first case is studied in detail. Figure 2 shows its mesh, decomposition, material distribution, and the loading over time. The material distribution is strongly heterogeneous. While the main part is a softer, less dense matrix, the thin horizontal fibers or stripes have a higher stiffness and density. The nodes on the left border of the structure are fixed. The structure is decomposed into 18 substructures using METIS. Some substructures exhibit bad aspect ratios as well as jagged interfaces, what makes the problem even harder. The loading is applied on a large part of the upper face so that a bending movement is induced. The load is increased and decreased linearly in the first half of the considered time span, beginning and ending at zero. At the given time step size of 10^{-3} s, the impulse is 10^{-2} s wide and thus approximately 32 times smaller than the period of the first eigenfrequency, which is 3.2×10^{-1} s long. We will identify this case as *Stripes/Metis/Bending*.

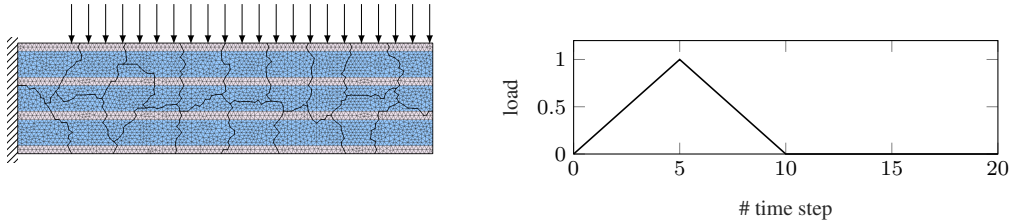


Figure 2. Problem setup for case Stripes/Metis/Bending. Left: Mesh, decomposition, material distribution, and applied load. Right: Scaling of the load over time.

Group	Short Name	A Priori Coarse Grid	Reuse Space Structure	Eigenproblem Size
reference algorithms	None	-	-	-
	GobalEV	eigenvectors of HF	-	# all interface dof
	GenEO	GenEO	-	# interface dof of s
recycling algorithms	PlainReuse	-	W_i	-
	RitzGenEO	-	$HF^s W_i \alpha_i q_k$	Ritz space size n^s

Table 1. Summary of algorithms.

All considered algorithms are summarized in Table 1, introducing short names like RitzGenEO which will be used in the legends of the plots. The table shows for each algorithm its short name, the type of the employed a priori coarse space, the basic structure of the space used for recycling purposes, and the size of possible eigenproblems that must be solved. All algorithms are based on the deflated adaptive multi-preconditioned FETI method as described in Algorithm 3 in which the adaptivity is configured by $\tau = 0.1$ without exception. The short names, which are derived from the source or type of the employed coarse space, refer to the methods described in Section 3 and Section 4 as follows.

None: Different from Algorithm 3, λ is initialized as zero in any case. No a priori coarse space or recycling technique is applied. It represents the simplest variant.

GobalEV: Different from Algorithm 3, λ is initialized as zero in any case. Uses an a priori computed coarse space, which is built from the eigenvectors associated with the highest eigenvalues of the

complete, assembled preconditioned conjugate gradient operator HF . This is, of course, impossible for larger problems and furthermore highly inefficient. It serves as a reference in terms of the best possible reduction of computational cost per coarse space basis vector.

GenEO(n/j): Different from Algorithm 3, λ is initialized as zero in any case. Uses an a priori computed coarse space which is built from GenEO modes, see Equation (7). Before the solution starts, the GenEO eigenproblem is solved for every substructure. The GenEO modes are selected by their GenEO values. For (j), this is done by the application of the jump criterion, for (n) the coarse space size is set manually and the globally lowest GenEO values are chosen.

PlainReuse: Reuse of individual contributions, see Section 4.3. All individual basis vectors of the minimization spaces from every iteration are reused. They are selected in the order of the iterations. The individual directions within one iteration are sorted in descending order of the pivot elements in the rank revealing LDL^T decomposition. For better comparability, the size of the coarse space is manually limited to be equal to the size of the GenEO coarse space.

RitzGenEO(n/j): A Ritz approximation of the GenEO modes is computed, see Section 4.4.1. The Ritz space basis is selected automatically by the method presented in Section 4.4.2. The approximated modes per substructure are selected, based on the corresponding Ritz values, as in GenEO either by the jump (j) criterion or a fixed coarse space size (n). In the case of (n), the size of the coarse space is chosen to be equal to the size of the GenEO coarse space to enable better comparability.

Both recycling variants are compared against the reference algorithms measuring the number of local solves and the effective eigenspectra after deflation. The left graph of Figure 3 shows the average number of local solves per substructure over time. That measure counts the number of all Dirichlet and all Neumann solves one substructure executes and depends on the number of iterations, the size of the minimization space and on how many neighbors one substructure has. The right graph of Figure 3 shows the size of the coarse grid in each time step. To allow a fair comparison by the number of local solves, a fixed number of basis vectors for the coarse space is prescribed. The target size of the coarse grid was set to the size which a GenEO(j) algorithm would have generated automatically. However, prescribing the same coarse space size for the GenEO(n) algorithm might lead to a slightly different selection of basis vectors among the substructures, because GenEO(j) uses individual thresholds for the GenEO values in each substructure whereas GenEO(n) uses a global one, equal for all substructures. Figure 4 shows the effective eigenspectrum, governing the convergence of the conjugate gradient algorithm. The lower and the more clustered the spectrum is, the faster the convergence is expected to be. The highest eigenvalue is equivalent to the condition number, as the spectrum of the operator in a preconditioned FETI algorithm is always bounded from below by 1 [10].

The PlainReuse method is able to reduce the number of local solves, but its performance is inferior to RitzGenEO. The high parts of the spectrum, shown in are clearly elevated in comparison to the references. The Ritz approximation of the GenEO modes in RitzGenEO shows a very good performance.

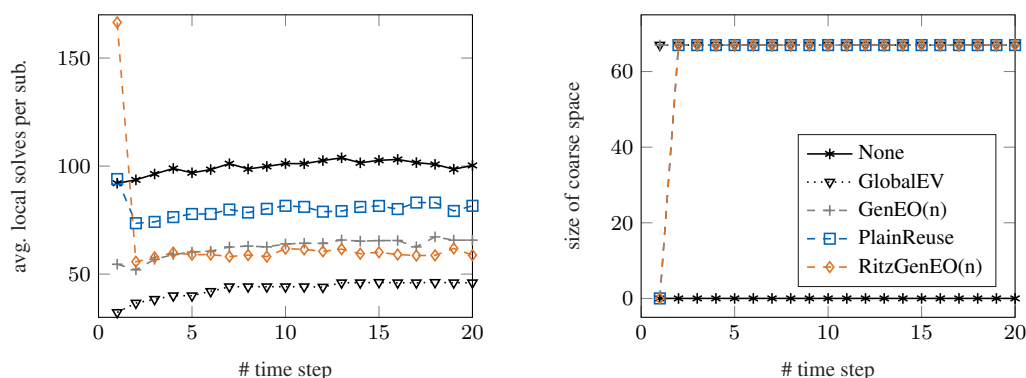


Figure 3. Performance of the recycling methods, measured by the average number of local solves per substructure at equal target coarse space sizes.

For later time steps, it is even slightly better than GenEO itself. An explanation could be the fact that different from the GenEO eigenproblems, the Ritz approximations are based on the right-hand side and thus take into account the specific excitation of modes by a particular loading. In the first time step, the numbers of local solves for the Ritz approximation-based methods are even higher than for the standard

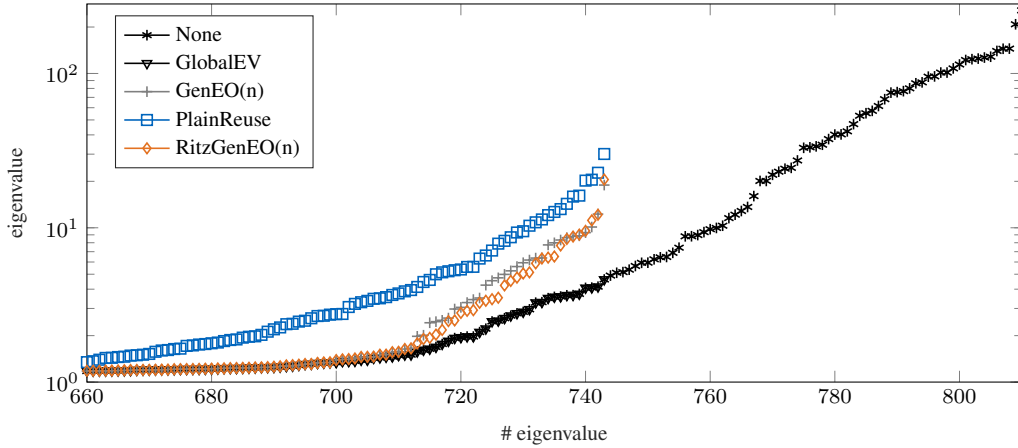


Figure 4. Spectrum of the projected preconditioned operator $HP_C^T F$. The ordinate shows the eigenvalues while the abscissa gives its number, sorted in ascending order. Zero eigenvalues, whether they are caused by cross points, i.e. redundant Lagrange multipliers, or deflation by projection, are ignored.

multi-preconditioned algorithm None. The reason is that the additional local solves, i.e. the application of the local preconditioners H^s between neighbors, needed to assemble the Ritz spaces as described in Appendix A.2, are included here. However, the cost of the GenEO coarse grid that must be computed a priori is not included in this graph. RitzGenEO performs clearly better than PlainReuse, which is confirmed by the accumulated numbers in Table 3 in Appendix B.

The eigenspectra shown in Figure 4 confirm the capability of the Ritz ansatz to generate highly efficient coarse spaces. On the one hand, the solution of RitzGenEO does not lead to a consistently good approximation of the spectrum produced by GenEO, but on the other hand, the high part of the spectrum of RitzGenEO is exceptionally low and clustered. About the first 710 eigenvalues coincide well with the original and with the GenEO deflated spectrum and what's more, the condition numbers, i.e. the highest eigenvalues, are almost the same.

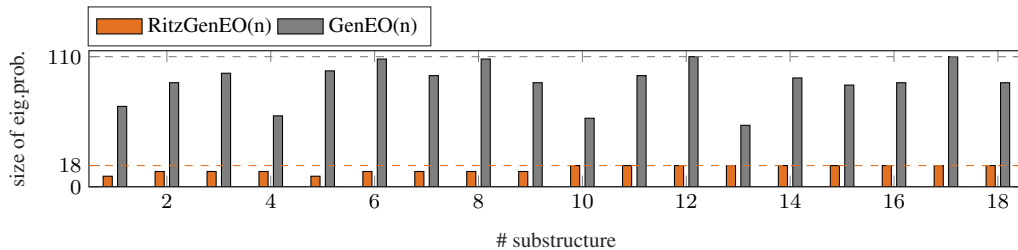


Figure 5. Size of the eigenproblem to be solved once for each substructure.

To illustrate the computational cost more in depth, Figure 5 shows the size of the eigenproblem that must be solved once for each substructure. For RitzGenEO, the size of an eigenproblem is equal to the number of solution increments that have been considered to build the specific Ritz space for this substructure.

This measure is much smaller for the Ritz approximations than for GenEO. Building the Ritz eigenproblems requires only a small number of vector exchanges between neighbors and vector products as well as some additional local solves, which are already incorporated in Figure 3. For GenEO, the full Schur complements of the iteration matrices must be exchanged between all neighbors and are then assembled to a significantly larger eigenproblem.

A comparison over a longer period of time can be found in Figure 9 in Appendix B.

5.1 Effect of Increasing the Ritz Space Size

This section illustrates one of the most significant findings of this research: A very rough and thus cheap approximation of the GenEO coarse space is sufficient to achieve a similar reduction of local solves. This is shown by comparing different Ritz space sizes because as a larger Ritz space results in a better,

but more expensive approximation. Figure 6 and Figure 7 show the performance by means of local

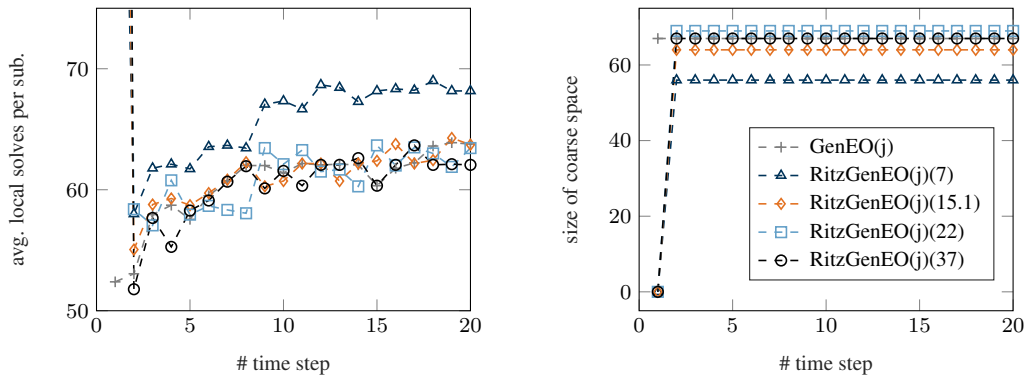


Figure 6. Performance of the Ritz approximation-based methods with different Ritz space sizes, compared to GenEO.

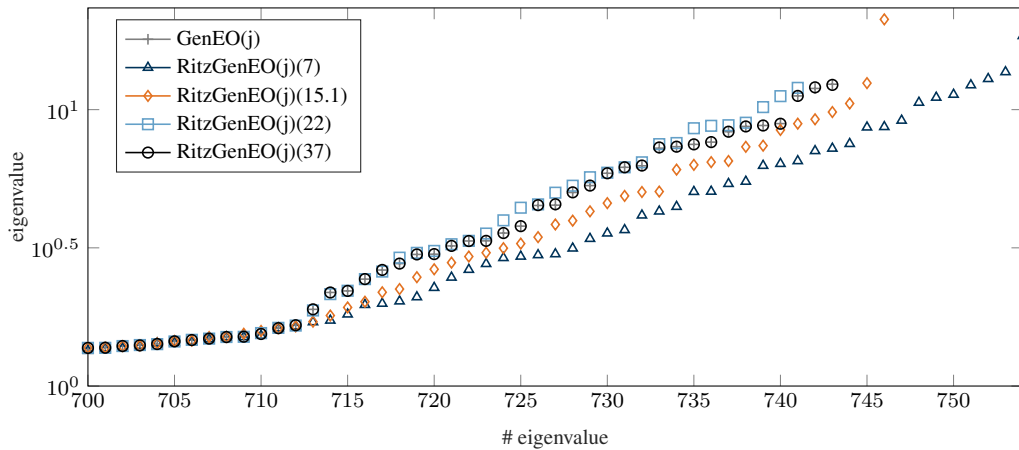


Figure 7. Effective spectra for the different Ritz space sizes, governing the convergence of the conjugate gradient algorithm. The ordinate shows the eigenvalues of the projected preconditioned operator $HP_C^T F$ while the abscissa gives its number, sorted in ascending order. Zero eigenvalues, whether they are caused by cross points, i.e. redundant Lagrange multipliers, or deflation by projection, are ignored.

solves and the effective operator spectra. The Ritz space size, averaged over all substructures as each substructure can use a different size, is given for each variant in the legend in brackets. The Ritz space size of 15.1 results from the automatic selection method in Section 4.4.2, the others were set manually.

The results confirm the assumption on which the Ritz space basis selection method in Section 4.4.2 was built: Not all available solution increments must be taken into account to build an efficient coarse space, but only those, where the τ -criterion identifies bad convergence and the minimization step computes a favorable combination of neighboring contributions. The automatic selection criterion leads to the small Ritz space size of 15.1 but already achieves the same performance as GenEO. Approximating the GenEO coarse space much more precisely, like a Ritz space size of 37 does, is not efficient.

5.2 Further Testcases

The methods are applied to three more test cases. A detailed overview of the setup of all four test cases as well as all results in tabular form are given in Appendix B. On average, the RitzGenEO algorithm saves 40% of the local solves after the first time step, when compared to the standard deflated AMP FETI.

For the homogeneous material distribution as well as for the case of a rectangular decomposition, the original GenEO coarse space gains advantage over its Ritz approximation, although the differences between the two algorithms remain small for all four cases.

6 Conclusions

This research presents recycling methods for the recently developed adaptive multi-preconditioned FETI (AMP-FETI). Their aim is to efficiently reuse information from prior solution processes, when the same linear operator is solved repeatedly. To the best of the authors's knowledge, this issue has not been addressed in any publication before. Two such methods are presented and assessed by application to four test cases. Both methods work by building a coarse space that is then employed to accelerate the AMP-FETI.

The first method, called PlainReuse here, is a straightforward application of an existing method for single-preconditioned FETI. It reuses the plain solution space without further processing as coarse space. It showed minor performance in the results. However, it is very easy to implement and can thus serve as a quickly set up acceleration technique in AMP-FETI for multiple right-hand side problems.

The second method, called RitzGenEO, approximates local eigenproblems (GenEO) that are known to provide highly efficient coarse spaces. This is done by applying a Ritz ansatz and consists of three steps. Build a Ritz space basis in which the approximation is sought, set up and solve the local eigenproblems and finally choose which of their solutions are used to construct the coarse space. The idea to compute a Ritz approximation of GenEO using a prior solution space of an AMP-FETI is completely novel. The specific choice of the Ritz space basis is crucial. This work presents not only a very effective basis but also a selection method to restrict the basis's size such that the highly efficient RitzGenEO method results. It achieves about the same performance as the original GenEO with just a rough and thus cheap approximation of it. For the solution of multiple right-hand side problems by a dual domain decomposition technique, RitzGenEO in combination with AMP is currently the most advanced and efficient way available.

In addition to the new RitzGenEO method, this work also gives valuable insight into the connection between AMP-FETI and GenEO. The results show that using a larger part of the solution space as Ritz space basis, results in a precise approximation of GenEO's behavior. This indicates that both identify the same bad modes using different mechanisms.

Future work needs to be done to further assess the abilities of these methods, especially applying them to larger, industrial examples and measuring their performance in terms of wall clock time.

A Modifications to Standard Adaptive Multi-Preconditioned FETI

The algorithm to which the recycling methods in this article have been applied to produce the results in Section 5 is a slightly extended and modified version of the deflated adaptive multi-preconditioned FETI procedure in Algorithm 1 as originally published in [11].

In this chapter, the modifications are explained in detail and the resulting procedure is listed in Algorithm 3.

A.1 Extended Initialization Phase

All recycling methods that are applied to AMP-FETI in this article reuse the solution space of the first time step to accelerate the solution of future time steps.

This results in two different initialization scenarios. In the first time step there is no coarse space projection yet but λ_0 is initialized as explained in Section 4.1 to activate the full structure. In all following time steps, a coarse space projection is present but no additional initialization of λ_0 is made.

As the τ -criterion involves information of the previous step direction, it is not available when the preconditioner is applied to the initial residual r_0 . In the standard AMP-FETI in Algorithm 1, this is handled by simply always applying full multi-preconditioning in the initialization phase. Because any chosen coarse space is expected to be efficient, it is considered inefficient to apply full multi-preconditioning in the initialization phase if a coarse space is present. Instead, standard single-preconditioning on r_0 is applied if the coarse space is not empty. This extended initialization phase is listed in Algorithm 2.

A.2 Assembly of the Reduced Eigenvalue Problem

In what follows, the assembly of the reduced eigenvalue problem Equation (8) is described, such that it requires minimal extra computational cost. The resulting procedure is listed in Algorithm 3. First, we localize the computation of Q_i as it must be done anyway in a parallel environment. Furthermore, the

Algorithm 2: Modified Initialization Phase

```

if  $C = \emptyset$  then
   $P_C = I$      $\lambda_0 = \lambda_{\text{activate}}$  (see Equation (6))
   $F_0 = F\lambda_0$      $\tilde{\eta} = \|F\lambda_0\| / \|d\|$      $\eta = 0.05$ 
   $\lambda_0 \leftarrow \lambda_0\eta/\tilde{\eta}$      $F_0 \leftarrow F_0\eta/\tilde{\eta}$ 
   $r_0 = d - F_0$ 
  for  $s = 1, \dots, N_s$  do
     $z_0^s = H^s r_0$ 
   $Z_0 = [z_0^1 | z_0^2 | \dots | z_0^{N_s}]$ 
else
   $P_C = I - C(C^T F C)^{-1} C^T F$ 
   $\lambda_0 = C(C^T F C)^{-1} C^T d$ 
   $r_0 = d - F\lambda_0$ 
   $Z_0 = \sum_{s=0}^{N_s} H^s r_0$ 

```

projection of the preconditioned residuals in Z_i by applying the projector P_C is done before. Both steps then write

$$Z_i \leftarrow P_C Z_i$$

$$Q_i^s = F^s W_i = F^s Z_i - \sum_{j=0}^{i-1} Q_j^s \beta_{i,j}$$

$$Q_i = \sum_{s=1}^{N_s} Q_i^s$$

Now $F^s V_W^s$ can be expressed as

$$F^s V_W^s = [Q_1^s \alpha_1 | Q_2^s \alpha_2 | \dots | Q_{n^s}^s \alpha_{n^s}]$$

It is important to note that the GenEO eigenvalue problem requires only the preconditioners H^s of adjacent substructures to be combined, i.e. those that share at least one degree of freedom on their interface. To express this topological restriction, the function

$$N(s) := \{s, \text{neighbors of } s\}$$

is introduced, providing the neighborhood of a given substructure s including itself. The last component missing to assemble the reduced eigenvalue problems and the coarse space then writes

$$H F^s V_W^s = \sum_{m \in N(s)} H^m F^m V_W^m$$

This means that assembling the reduced eigenvalue problem requires each substructure to solve additional Neumann problems by applying H^s to the basis vectors of its own Ritz space as well as on the basis vectors of its neighbors' Ritz spaces.

Algorithm 3: Deflated Adaptive Multi-Preconditioned FETI

See initialization phase in Algorithm 2

$$\mathbf{Z}_0 \leftarrow \mathbf{P}_C \mathbf{Z}_0$$

$$\mathbf{W}_0 = \mathbf{Z}_0$$

$$i = 0$$

while $\sqrt{\|\mathbf{r}_i^T \mathbf{H} \mathbf{r}_i\|} < \epsilon$ **do**

for $s = 1, \dots, N_s$ **do**

$$\quad \lfloor \mathbf{Q}_i^s = \mathbf{F}^s \mathbf{Z}_i - \sum_{j=0}^{i-1} \mathbf{Q}_j^s \beta_{i,j}$$

$$\mathbf{W}_i = \mathbf{Z}_i - \sum_{j=0}^{i-1} \mathbf{W}_j \beta_{i,j} \quad \Delta_i = (\sum_{s=1}^{N_s} \mathbf{Q}_i^s)^T \mathbf{W}_i \quad \gamma_i = \mathbf{W}_i^T \mathbf{r}_i$$

$$[\mathbf{P}, \mathbf{L}, \mathbf{A}] = \text{RankRevealingLDLT}(\Delta_i)$$

for $s = 1, \dots, N_s$ **do**

$$\quad \lfloor \mathbf{Q}_i^s \leftarrow \mathbf{Q}_i^s \mathbf{P} \mathbf{L}^{-T}$$

$$\mathbf{W}_i \leftarrow \mathbf{W}_i \mathbf{P} \mathbf{L}^{-T} \quad \Delta_i^+ = \mathbf{A}^{-1} \quad \alpha_i = \mathbf{A}^{-1} (\mathbf{P} \mathbf{L}^{-T})^T \gamma$$

$$\lambda_{i+1} = \lambda_i + \mathbf{W}_i \alpha_i$$

$$\mathbf{r}_{i+1} = \mathbf{r}_i - (\sum_{s=1}^{N_s} \mathbf{Q}_i^s) \alpha_i$$

$$\mathbf{Z}_{sum} = 0 \quad \mathbf{Z}_{i+1} = \emptyset$$

for $s = 1, \dots, N_s$ **do**

$$\quad \mathbf{z}_{i+1}^s = \mathbf{H}^s \mathbf{r}_{i+1}$$

$$\quad \Xi_i^s = \frac{(\mathbf{W}_i \alpha_i)^T \mathbf{F}^s \mathbf{W}_i \alpha_i}{\mathbf{r}_{i+1}^T \mathbf{H}^s \mathbf{r}_{i+1}}$$

if $\Xi_i^s < \tau$ **then**

$$\quad \quad \lfloor \mathbf{Z}_{i+1} \leftarrow [\mathbf{Z}_{i+1} \mid \mathbf{z}_{i+1}^s]$$

else

$$\quad \quad \lfloor \mathbf{Z}_{sum} \leftarrow \mathbf{Z}_{sum} + \mathbf{z}_{i+1}^s$$

$$\mathbf{Z}_{i+1} \leftarrow [\mathbf{Z}_{i+1} \mid \mathbf{Z}_{sum}]$$

$$i \leftarrow i + 1$$

for $0 \leq j \leq i - 1$ **do**

$$\quad \lfloor \Phi_{i,j} = (\sum_{s=1}^{N_s} \mathbf{Q}_j^s)^T \mathbf{Z}_i \quad \beta_{i,j} = \Delta_j^+ \Phi_{i,j}$$

$$\mathbf{Z}_i \leftarrow \mathbf{P}_C \mathbf{Z}_i$$

$$m = i$$

B Summary of Test Cases and Results

The problem settings for all four test cases are depicted in Figure 8, varying the direction and location of the load, the heterogeneity, and the decomposition of the mesh. The physical properties are summarized in Table 2. For all cases, the load is scaled over time according to Figure 2. Also constant among all cases is the AMP-criterion $\tau = 0.1$, the time step size $\Delta t = 10^{-3} s$ and the time stepping scheme configuration $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$. Table 2 shows as well the periods of the three lowest eigenmodes and the period of the highest eigenmode. While the lowest eigenmodes mainly result from the structure itself and the Dirichlet boundary conditions, the highest eigenmode is mainly determined by the element size. Similar to [14], the time step is selected such that the lowest structural modes are represented well.

The results are presented in Table 3, including the first case *Stripes/Metis/Bending* which was studied in detail before. The most important numbers, showing the performance without any coarse space, with a GenEO coarse space, and the Ritz approximation method, are emphasized in bold face.

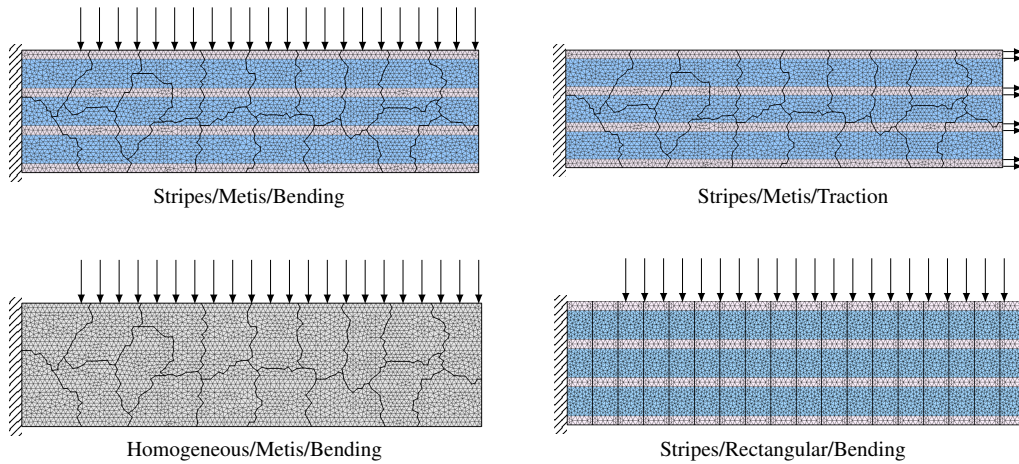


Figure 8. Problem setting of the four test cases, showing the mesh, its decomposition, the material distribution and the direction and location of applied loads.

Parameter		Case 1	Case 2	Case 3	Case 4
Material distribution	-	Stripes	Stripes	Homogeneous	Stripes
Decomposition	-	Metis	Metis	Metis	Rectangular
Type of loading	-	Bending	Traction	Bending	Bending
Youngs mod. fiber/matrix	$\frac{E_2}{E_1}$	10^4	10^4	1	10^4
Density fiber/matrix	$\frac{\rho_2}{\rho_1}$	10^1	10^1	1	10^1
Poisson ratio matrix	ν_1	0.49	0.49	0.3	0.49
Poisson ratio fiber	ν_2	0.3	0.3	0.3	0.3
Period of 1. eigenfreq.	T_1	$3.2 \times 10^{-1} s$	$3.2 \times 10^{-1} s$	$3.8 \times 10^{-2} s$	$3.2 \times 10^{-1} s$
Period of 2. eigenfreq.	T_2	$6.6 \times 10^{-2} s$	$6.6 \times 10^{-2} s$	$7.6 \times 10^{-3} s$	$6.7 \times 10^{-2} s$
Period of 3. eigenfreq.	T_3	$4.2 \times 10^{-2} s$	$4.2 \times 10^{-2} s$	$6.2 \times 10^{-3} s$	$4.2 \times 10^{-2} s$
Period of highest eig.f.	T_{\max}	$1.4 \times 10^{-5} s$	$1.4 \times 10^{-5} s$	$1.3 \times 10^{-5} s$	$1.5 \times 10^{-5} s$

Table 2. Configuration of the four test cases. They can be uniquely identified by the descriptions of material distribution, decomposition, and type of loading in the first three lines, e.g. read "Stripes/Metis/Bending".

In Figure 9, RitzGenEO and GenEO are compared over a longer period of time on the test case *Stripes/Metis/Bending*. As the graph shows, the performance of RitzGenEO does not deteriorate and stays slightly better than GenEO although the Ritz approximation was computed only once from the solution increments of the first time step. For all later time steps, the coarse space remains unchanged.

Stripes/Metis/Bending					
Measure	None	GenEO(j)	GenEO(n)	RitzGenEO(n)	PlainReuse
iterations	790	560	579	556	757
size of min. space	2300	766	773	832	960
avg. local solves per sub. in t_1	92	52	55	167	94
avg. local solves per sub. in $\sum_{i=2}^{20} t_i$	1901	1157	1193	1128	1509
size of coarse space in t_{20}	-	67	67	67	67
avg. size of eig.space. in t_1	-	88.1	88.1	15.1	-

Stripes/Metis/Traction					
Measure	None	GenEO(j)	GenEO(n)	RitzGenEO(n)	PlainReuse
iterations	756	602	613	602	787
size of min. space	2306	771	776	769	1055
avg. local solves per sub. in t_1	100	59	59	161	100
avg. local solves per sub. in $\sum_{i=2}^{20} t_i$	1840	1227	1249	1185	1580
size of coarse space in t_{20}	-	67	67	67	67
avg. size of eig.space. in t_1	-	88.1	88.1	12.6	-

Homogeneous/Metis/Bending					
Measure	None	GenEO(j)	GenEO(n)	RitzGenEO(n)	PlainReuse
iterations	720	460	331	365	573
size of min. space	1899	507	331	423	647
avg. local solves per sub. in t_1	86	45	32	142	89
avg. local solves per sub. in $\sum_{i=2}^{20} t_i$	1689	926	670	696	1115
size of coarse space in t_{20}	-	74	74	74	74
avg. size of eig.space. in t_1	-	88.1	88.1	11.1	-

Stripes/Rectangular/Bending					
Measure	None	GenEO(j)	GenEO(n)	RitzGenEO(n)	PlainReuse
iterations	928	394	733	785	908
size of min. space	3391	418	1560	1574	1891
avg. local solves per sub. in t_1	107	34	72	158	103
avg. local solves per sub. in $\sum_{i=2}^{20} t_i$	2164	797	1567	1632	1910
size of coarse space in t_{20}	-	132*	74	74	74
avg. size of eig.space. in t_1	-	113.3	113.3	19.0**	-

Table 3. Table of results for all test cases. The number of iterations and the size of min. space were accumulated over 20 time steps. The average number of local solves per substructure is given separately for the first time step and accumulated over the remaining time steps.

*: The coarse space size was limited to a lower number for the (n) variants because the jump criterion leads to a very large coarse space in this case.

** : This Ritz space size was limited manually because the automatic selection originally leads to a very large Ritz space in this case.

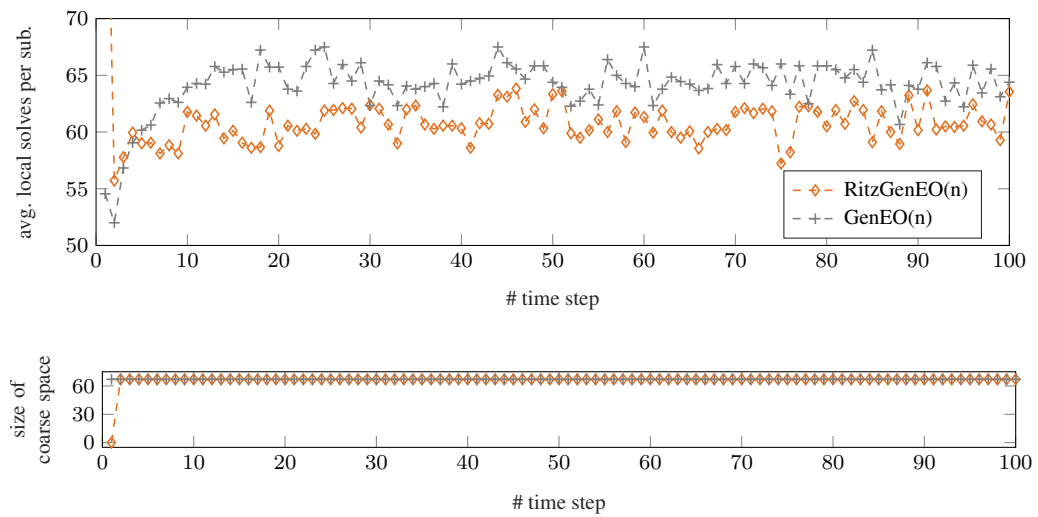


Figure 9. Comparison of performance for RitzGenEO and GenEO over a longer time span.

References

- [1] Farhat, C. and Roux, F.-X. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering* 1991; **32**(6), 1205–1227.
- [2] Farhat, C., Mandel, J., and Roux, F. X. Optimal convergence properties of the FETI domain decomposition method. *Computer Methods in Applied Mechanics and Engineering* 1994; **115**(3), 365–385.
- [3] Rixen, D. J. and Farhat, C. A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *International Journal for Numerical Methods in Engineering* 1999; **44**(4), 489–516.
- [4] Bhardwaj, M., Day, D., Farhat, C., et al. Application of the FETI method to ASCI problems - scalability results on 1000 processors and discussion of highly heterogeneous problems. *International Journal for Numerical Methods in Engineering* 2000; **47**(1-3), 513–535.
- [5] Farhat, C., Chen, P.-S., Mandel, J., and Roux, F. X. 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**(1), 153–179.
- [6] Farhat, C., Chen, P.-S., Risler, F., and Roux, F.-X. A unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers. *International Journal for Numerical Methods in Engineering* 1998; **42**(January 1997), 257–288.
- [7] Dostál, Z. Conjugate gradient method with preconditioning by projector. *International Journal of Computer Mathematics* 1988; **23**(3-4), 315–323.
- [8] Spillane, N. and Rixen, D. Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms. *International Journal for Numerical Methods in Engineering* 2013; **95**(11), 953–990.
- [9] Rixen, D. J. Substructuring and dual methods in structural analysis. PhD thesis. Universite de Liege, 1997.
- [10] Gosselet, P., Rixen, D., Roux, F.-X., and Spillane, N. Simultaneous FETI and block FETI: robust domain decomposition with multiple search directions. *International Journal for Numerical Methods in Engineering* 2015; **104**(10), 905–927.
- [11] Spillane, N. An adaptive multipreconditioned conjugate gradient algorithm. *SIAM Journal on Scientific Computing* 2016; **38**(3), A1896–A1918.
- [12] Bridson, R. and Greif, C. A multipreconditioned conjugate gradient algorithm. *SIAM Journal on Matrix Analysis and Applications* 2006; **27**(4), 1056–1068.
- [13] Farhat, C., Crivelli, L., and Roux, F.-X. A transient FETI methodology for large-scale parallel implicit computations in structural mechanics. *International Journal for Numerical Methods in Engineering* 1994; **37**(11), 1945–1975.
- [14] Farhat, C., Chen, P.-S., and Mandel, J. A scalable Lagrange multiplier based domain decomposition method for time-dependent problems. *International Journal for Numerical Methods in Engineering* 1995; **38**(22), 3831–3853.
- [15] Farhat, C., Crivelli, L., and Roux, F.-X. Extending substructure based iterative solvers to multiple load and repeated analyses. *Computer Methods in Applied Mechanics and Engineering* 1994; **117**(1), 195–209.
- [16] Farhat, C. and Chen, P.-S. Tailoring domain decomposition methods for efficient parallel coarse grid solution and for systems with many right hand sides. 1994.
- [17] Roux, F.-X. Parallel implementation of a domain decomposition method for non-linear elasticity problems. *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*. Ed. by Keyes D., Y. Saad, and Truhlar D.G. Society for Industrial and Applied Mathematics, 1995, 161–175.
- [18] Rey, C. and Léné, F. Reuse of Krylov spaces in the solution of large-scale nonlinear elasticity problems. *Ninth International Conference on Domain Decomposition Methods*. 1998, 465–471.
- [19] Risler, F. and Rey, C. On the reuse of ritz vectors for the solution to nonlinear elasticity problems by domain decomposition methods. 1998.
- [20] Gosselet, P. and Rey, C. On a selective reuse of Krylov subspaces in Newton-Krylov approaches for nonlinear elasticity. *Fourteenth international conference on domain decomposition methods* 2003, 419–426.
- [21] Gosselet, P., Rey, C., and Pebrel, J. Total and selective reuse of Krylov subspaces for the resolution of sequences of nonlinear structural problems. *International Journal for Numerical Methods in Engineering* 2013; **94**(1), 60–83. arXiv: 1010.1724.
- [22] Farhat, C., Crivelli, L., and Geradin, M. On the spectral stability of time integration algorithms for a class of constrained dynamics problems. *34th Structures, Structural Dynamics and Materials Conference*. Structures, Structural Dynamics, and Materials and Co-located Conferences. American Institute of Aeronautics and Astronautics, 1993, 80–97.

- [23] Newmark, N. M. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division* 1959; **85**(85), 67–94.
- [24] Geradin, M. and Rixen, D. J. *Mechanical vibrations: theory and application to structural dynamics*. 3rd. Wiley & Sons, 2015, 617.
- [25] Farhat, C. and Roux, F.-X. Implicit parallel processing in structural mechanics. *Computational Mechanics Advances* 1994; **2**(1), 1–124.
- [26] Paige, C. C. Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem. *Linear Algebra and its Applications* 1980; **34**, 235–258.
- [27] Roux, F.-X. Acceleration of the outer conjugate gradient by reorthogonalization for a domain decomposition method for structural analysis problems. *Proceedings of the 3rd International Conference on Supercomputing, ICS '89*. New York, NY, USA: ACM, 1989, 471–476.
- [28] Bovet, C., Parret-Fréaud, A., Spillane, N., and Gosselet, P. Adaptive multipreconditioned FETI: scalability results and robustness assessment. *Computers & Structures* 2017; **193**, 1–20.
- [29] Roux, F.-X. and Farhat, C. Parallel implementation of direct solution strategies for the coarse grid solvers in 2-level FETI method. *Contemporary Mathematics*. Vol. 218. 1998, 158–173.
- [30] Leistner, M., Rixen, D., and Gosselet, P. Application of multipreconditioned iterative algorithms in dual domain decomposition methods for structural dynamics. *To appear in: PAMM* 2017.
- [31] Roux, F.-X. Spectral analysis of the interface operators associated with the preconditioned saddle-point principle domain decomposition method. *Domain Decomposition Methods for Partial Differential Equations*. Ed. by D. Keyes, T. Chan, G. Meurant, J. Scroggs, and R. Voigt. Norfolk: Siam, 1992, 73–90.
- [32] Sluis, A. van der and Vorst, H. A. van der. The rate of convergence of conjugate gradients. *Numerische Mathematik* 1986; **48**(5), 543–560.
- [33] Jolivet, P., Dolean, V., Hecht, F., et al. High performance domain decomposition methods on massively parallel architectures with FreeFem++. *Journal of Numerical Mathematics* 2012; **20**(3-4), 287–302.