XL CILAMCE
IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING
NOVEMBER 11-14, 2019
Praiamar Natal Hotel & Convention
Natal, RN-BRAZIL

ON THE SOLUTION OF GENERALIZED EIGENVALUE PROBLEMS USING PROJECTED ARNOLDI AND DUAL DOMAIN DECOMPOSITION METHODS

Guilherme Jenovencio
Daniel J. Rixen
guilherme.jenovencio@tum.de
rixen@tum.de
TUM
Boltzmannstrasse 15, 85748 Garching, Germany

Abstract. Modal Analysis plays an important role in understanding the dynamics of structures. The mode shapes and natural frequencies are crucial structural properties that must the known to avoid catastrophic failures due to resonances. Moreover, eigenmodes can be used to reduce the number of degrees of freedom of a Dynamic System by using Reduced Order Models (ROM) such as Craig-Bampton, Dual-Craig-Bampton, Rubin, among others. Despite its importance, full eigenanalysis is rarely required, and in general, only the smallest eigenvalues are computed due to their relevance for practical problems. Therefore, Lanczos and implicit restarted Arnoldi algorithms are often used as eigensolvers due to their efficiency. The major cost of those algorithms lies in the multiple solutions of static-like problems, which can produce prohibitive computational cost when the discretized dynamic model has more than millions of unknowns. In this work, a nonoverlapping dual domain decomposition method, namely Finite Element Tearing and Interconnection, is combined with a modified version of the Arnoldi Algorithm to efficiently compute the eigenpairs of large scale structural problems.

Keywords: FETI, Projected Arnoldi, Constrained Eigenmodes, Krylov Method

1 Introduction

The core of linear dynamic systems lies in its free vibration properties, namely the eigen-frequencies and mode shapes. The eigen-modes are so important, that they are the basis for a variate of reduction techniques such as Craig-Bampton [1], Dual-Craig-Bampton [2], Rubin [3], among others. In general, only the smallest eigenvalues are computed due to their relevance for practical problems. Therefore, iterative methods such as Lanczos and implicit restarted Arnoldi algorithms are often used when large dynamic system are considered due to their computational efficiency, as shown in [4], [5],[6], [7], [8], [9].

The major cost of those algorithms lies in the inverse iteration, where the equilibrium solution between the internal elastic forces and inertia must be resolved. This procedure may be computationally prohibitive when the discretized model has more than millions of unknowns. In 1991 the Finite Element Tearing and Interconnection (FETI) was introduced by Farhat et al, [10] in order to solve structural static problems using parallel computers in an efficient and scalable manner. Since then, many authors have proposed different variants of the method to improve its converge properties and robustness. Some of the popular implementation variants are FETI-2 [11], FETI-DP [12], Total-FETI [13], FETI-Geneo [14], S-FETI [15].

The connection between free-vibration analysis and FETI was firstly exploited in [16] in a block Lanczos type of algorithm. However, the initial vector to start Lanczos procedure is not discussed in details. This seems to be an important aspect in the propagation of the error in higher frequency modes, due to the compatibility displacement requirements. Moreover, the aforementioned paper does not describe how the tolerance criteria in the FETI-PCPG solver affects the precision of the eigenpairs of the structural component. Another work concerning eigen-analyses and dual domain decomposition is presented in [17]. Although the Power eigensolver algorithm is described, no further discussion concerning the orthogonalization procedure of the constrained Krylov space is presented. Understanding how to build the orthogonal Krylov vectors is especially relevant when Lanczos and Arnoldi methods are used and the interface compatibility is not precisely fulfilled. In the present work, we discuss how to reformulate the inverse iteration in order to make the eigensolver less sensitive to the tolerances given to the iterative interface solver. Besides that, we perform strong scalability tests to prove that FETI-solvers can be efficiently used to solve generalized eigenvalue problems.

2 Modal Analysis of Connected Subdomains

In this section, the free vibration equation for one domain is reformulated in terms of multiple subdomains. This procedure introduces interface compatibility equations that must be satisfied together with the classical generalized eigenvalue problem. There are multiple ways to approach the constrained generalized eigenvalue problem however, we will focus on the global assembly, Dual and Dual-Projected formulations. Special treatment is given to the iterative dual formulations in order to solve robustly the modal analysis of connected substructures.

2.1 Equation of Motion of Subdomains

The linear dynamic equation of a global domain Ω decomposed in N subdomain $\Omega^{(i)}$ is governed by the local dynamic equations and the interface compatibility which in the discretized form are written

$$\begin{aligned} \mathbf{M}^{(i)}\ddot{\mathbf{u}}^{(i)} + \mathbf{K}^{(i)}\mathbf{u}^{(i)} = \mathbf{f}^{(i)} + \sum_{ij \in \mathcal{I}_\Gamma^i} \mathbf{B}^{(ij)T}\lambda^{(ij)} \qquad i \in \mathcal{I}_\Omega \qquad \text{and} \qquad (ij) \in \mathcal{I}_\Gamma^i \\ \mathbf{B}^{(ij)}\mathbf{u}^{(i)} + \mathbf{B}^{(ji)}\mathbf{u}^{(j)} = 0 \qquad ij \in \mathcal{I}_\Gamma \end{aligned} \tag{1}$$

where $\mathbf{K}^{(i)}$, $\mathbf{M}^{(i)}$, $\mathbf{f}^{(i)}$ and $\mathbf{u}^{(i)}$ represents the stiffness, mass, external force and displacement of the domain i respectively. The Lagrange multiplier $\lambda^{(ij)}$ represents the interface forces connecting subdomain i

and j. The matrix $\mathbf{B}^{(ij)}$ is a sign Boolean matrix which extracts the interface dofs from the local domain and has a minus sign for $j > i$. The $\mathcal{I}_\Omega = \{1, 2, ..., N\}$ is the index set of the subdomains, $\mathcal{I}_\Gamma$ is the index set of all interface pairs, and $\mathcal{I}_\Gamma^i$ is the index set of the neighbors of domain i. Defining the global disconnected displacement vector as the concatenation of local displacement $\mathbf{u}^{(i)}$, such that $\mathbf{u} = \bigcup_{\mathcal{I}_\Omega} \mathbf{u}^i$ and also the global interface Lagrange multipliers vector $\lambda = \bigcup_{\mathcal{I}_\Gamma} \lambda^{ij}$, one can write the block system of equations as

$$\begin{bmatrix} \mathbb{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}} \\ \ddot{\lambda} \end{bmatrix} + \begin{bmatrix} \mathbb{K} & \mathbb{B}^T \\ \mathbb{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix} \tag{2}$$

where $\mathbb{M}$ is the block diagonal matrix of $\mathbf{M}^{(i)}$, $\mathbb{K}$ is the block diagonal matrix of $\mathbf{K}^{(i)}$, and $\mathbb{B}$ is the global assembled Boolean matrix which acts in the global disconnected displacements and produces the interface gap vector that has the same size of the global Lagrange multiplier vector. Figure (1) shows a domain decomposed in subdomains and the $\mathbb{B}$ associated with it.

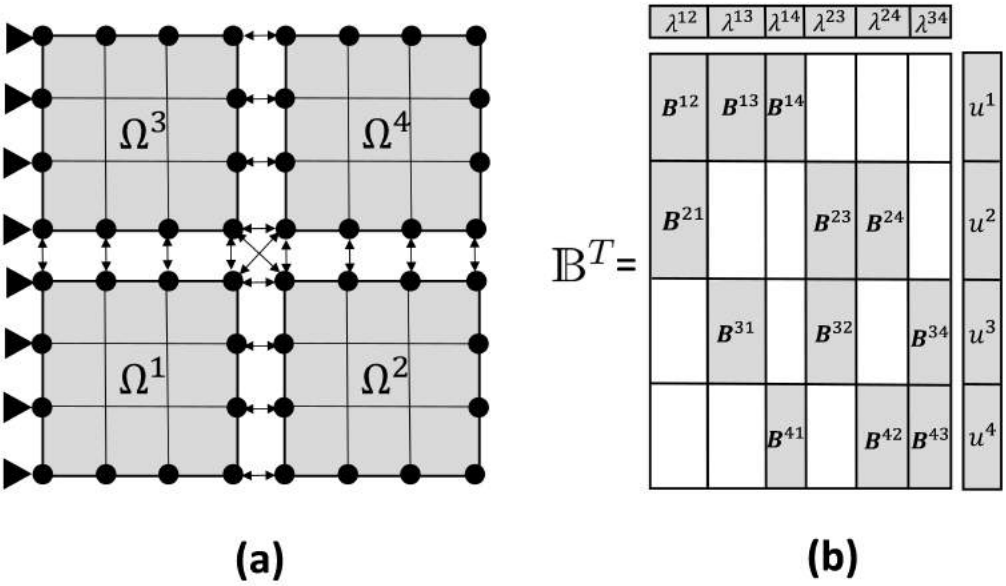

Figure 1. (a) Domain decomposed in 4 subdomains (b) Assembled global Boolean matrix.

2.2 Constrained Generalized Eigenvalue Problem

The free vibration analysis of eq.(2) assumes that both displacements and Lagrange multipliers are harmonic functions and also that no external force is applied into the system. This assumption leads to the constrained generalized eigenvalue problem

$$\begin{bmatrix} \mathbb{K} & \mathbb{B}^T \\ \mathbb{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}} \\ \tilde{\lambda} \end{bmatrix} = \omega^2 \begin{bmatrix} \mathbb{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}} \\ \tilde{\lambda} \end{bmatrix} \tag{3}$$

where ω is the circular frequency, and $[\tilde{u}, \tilde{\lambda}]$ is a eigenvector. The dimension of the above problem is $n_\lambda + n_{dofs}$, where n_λ is the number of interface constraints, and n_{dofs} is the total number of displacement degrees of freedom. As shown in the work developed by Cardona and Geradin in [18], the eigenpair solutions $(\omega_i, [\tilde{\mathbf{u}}_i, \tilde{\lambda}_i])$ of eq. (14) have the following form

$$\left(\omega_1^u, \begin{bmatrix} \tilde{\mathbf{u}}_1 \\ \tilde{\lambda}_1 \end{bmatrix}\right), ..., \left(\omega_{n_{dofs}}^{\mathbf{u}}, \begin{bmatrix} \tilde{\mathbf{u}}_{n_{dofs}} \\ \tilde{\lambda}_{n_{dofs}} \end{bmatrix}\right), \left(+\infty, \begin{bmatrix} 0 \\ \tilde{\lambda}_{n_{dofs}+1} \end{bmatrix}\right), ..., \left(+\infty, \begin{bmatrix} 0 \\ \tilde{\lambda}_{n_{dofs}+n_c} \end{bmatrix}\right) \tag{4}$$

In practical problems, the first eigenpairs are the most relevant, whereas higher frequency modes and the dual counterpart of the mode shapes $[\mathbf{0}, \tilde{\lambda}_i]^T$ can be discarded.

The eigenvalue problem presented in eq.(14) is generally solved by Krylov subspace iteration methods such as Lanczos and Arnoldi. Both algorithms produce a similar matrix of a dynamic system linear operator by means of the orthogonalization of the Krylov vectors generated by successively matrix multiplication. When the lowest eigenvalues are required to be computed, the inverse action of the linear operator is needed, what is known as the inverse iteration, which involves the solution of the linear system such

$$\begin{bmatrix} \mathbb{K} & \mathbb{B}^T \\ \mathbb{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_{n+1} \\ \tilde{\lambda}_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbb{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_n \\ \tilde{\lambda}_n \end{bmatrix} \tag{5}$$

The direct application of Krylov methods in the system (5) is generally not recommended due to the system singularity, and the possibility of finding spurious modes. In the next subsection we discuss how eq. (5) can be reformulated, in order to produce an stable and scalable eigensolver.

2.3 Global Assembly Formulation

It is possible to define new a set of basis vectors $\mathbb{L} = [l_1, l_2, ..., l_p]$ for eq. (1), such that they span the nullspace of $\mathbb{B}$ eliminating the need of the interface constraints. On the other hand, all the domain are coupled by this linear operator. In [19], Klerk et. al, called this approach the primal assembly formulation, once its operator reassembles the classic Finite Element assembly operator. However, in the presented case, the operator assembles subdomains instead of finite elements. Mathematically, one writes

$$\mathbf{u} = \mathbb{L}\mathbf{u}^p \tag{6}$$

where $\mathbb{L}$ is the primal assembly operator, that forms a bases of the nullspace of the constraint operator $\mathbb{B}$, mathematically

$$\mathbb{B}\mathbb{L} = \mathbf{0} \tag{7}$$

Figure (2) illustrates the connection between the linear constraint operator $\mathbb{B}$ and the basis of the primal assembly operator $\mathbb{L}$. It is possible to see that all bases vector of $\mathbb{L}$ lie in the constraint hyperplane $\mathbb{B}\mathbf{u} = \mathbf{0}$. Also it is shown the orthogonality between the $\mathbb{B}^T$ and $\mathbb{L}$.

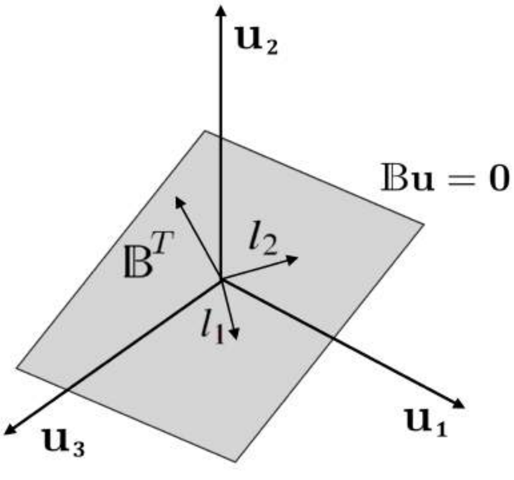

Figure 2. Constraint hyperplane $\mathbb{B}\mathbf{u}$ and the connection with the base vectors of $\mathbb{L}$.

Replacing definition (6) into eq. (14) and multiplying it by $\mathbb{L}^T$ results in the classical generalized eigenvalue problem without constraints

$$\mathbb{L}^T\mathbb{K}\mathbb{L}\mathbf{u}^p = \omega^2\mathbb{L}^T\mathbb{M}\mathbb{L}\mathbf{u}^p \tag{8}$$

The above formulation creates a positive-definite $\mathbb{L}^T\mathbb{K}\mathbb{L}$ which can be factorized and usually produces a stable inverse iteration in Krylov methods. On the other hand, the resulting system is not naturally parallel, which makes it difficult to achieve numerical scalability due to the strong coupling among the subdomains.

2.4 Projected Formulation

The global assembly formulation generates displacement vectors that always satisfy the constraint equation, but it does not explore fully the domain decomposition to accelerate the inverse iteration of the eigensolver. In order to solve the eigenvalue problem efficiently and profit from the independence of stiffness and mass matrices of the N subdomains, a constrained Krylov subspace might be built based on the interface constraint operator, see [20]. The Krylov matrix for the generalized eigenvalue problem without constraints is defined as:

$$\mathbf{V} = [\mathbf{b},\ \mathbf{D}\mathbf{b},\ \mathbf{D}^2\mathbf{b},\ \ldots,\ \mathbf{D}^{m-1}\mathbf{b}] \tag{9}$$

where $\mathbf{D} \equiv (\mathbf{K}-\sigma\mathbf{M})^{-1}\mathbf{M}$, which is the inverse shift operator such that $\mathbf{D} : \mathbb{R}^n \rightarrow \mathbb{R}^n$, and $\mathbf{b}$ is a initial vector, such that $\mathbf{b} \in \mathbb{R}^n$. When a homogeneus linear constraint $\mathbb{B}\mathbf{b} = \mathbf{0}$ is imposed, the constrained Krylov basis vectors $\mathbf{b}'$ must lie in the set of admissible vectors, mathematically $\nu = \{\mathbf{b}' \in \mathbb{R}^n \mid \mathbb{B}\mathbf{b}' = \mathbf{0}\}$. Therefore, the constrained Krylov matrix has the form

$$\mathbf{V}^c = [\mathbf{b}',\ (\mathbf{D}\mathbf{b}')',\ (\mathbf{D}^2\mathbf{b}')',\ \ldots,\ (\mathbf{D}^{m-1}\mathbf{b}')'] \tag{10}$$

Therefore, one can write a orthogonal projection operator $\mathbb{P}_{\mathbb{B}}$ into the $Null(\mathbb{B})$ such that $\mathbb{P}_{\mathbb{B}} : \mathbf{b} \rightarrow \mathbf{b}'$, where the constraint $\mathbb{B}(\mathbb{P}\mathbf{b}) = \mathbf{0}$ holds for every $\mathbf{b} \in \mathbb{R}^n$. The projection matrix is explicitly written as

$$\mathbb{P}_{\mathbb{B}} = \mathbb{I} - \mathbb{B}^T(\mathbb{B}\mathbb{B}^T)^{+}\mathbb{B} \tag{11}$$

where $^+$ represents the generalized inverse, due to $\mathbb{B}\mathbb{B}^T$ be generally singular and $\mathbb{I}$ is a block identity matrix. Defining the projected displacement as $\mathbf{u} = \mathbb{P}_{\mathbb{B}}\mathbf{u}^{proj}$ and replacing this definition in the dual inverse iteration in (5) by $\mathbb{P}_{\mathbb{B}}^T$

$$\mathbb{P}_{\mathbb{B}}^T\mathbb{K}\mathbb{P}_{\mathbb{B}}\mathbf{u}_{n+1}^{proj} = \mathbb{P}_{\mathbb{B}}\mathbb{M}\mathbb{P}_{\mathbb{B}}^T\mathbf{u}_n^{proj} \tag{12}$$

The inverse problem associated with the linear system above is singular but has a solution since the right-hand side is in the range of the operator $\mathbb{P}_{\mathbb{B}}^T\mathbb{K}\mathbb{P}_{\mathbb{B}}$. Besides that, the system is highly parallel when iterative methods such as Conjugate Gradient, BicGS, GMRES are used due to its block structure. The numerical and parallel scalability can be achieved by defining a proper preconditioner. The action of $\mathbb{B}$ only requires close neighbor communication, and the global coupling between subdomains is performed by the $(\mathbb{B}\mathbb{B}^T)^{T+}$ operator in expression (11). Its computation requires global communication and it is the factor which has the highest computational cost to assemble the projection.

If we define the scaled Boolean matrix $\tilde{\mathbb{B}} = \mathbb{S}\mathbb{B}$ with the following properties $\tilde{\mathbb{B}}\mathbb{B}^T = \mathbb{I}$, where $\mathbb{S}$ is a diagonal matrix whose elements are defined by the inverse of the number of shared subdomains for a given degree of freedom. Therefore $\mathbb{B}\mathbb{B}^T = \mathbb{S}^{-1}\mathbb{I}$ and the projection is given by

$$\mathbb{P}_{\mathbb{B}} = \mathbb{I} - \mathbb{B}^T\mathbb{S}\mathbb{B} \tag{13}$$

where the action of $\mathbb{P}_{\mathbb{B}}$ defined above may be easily implemented in parallel since it only requires communication among neighbors subdomains.

2.5 Dual-Projected Formulation

Whereas the dual inverse iteration (5) can produce spurious modes due to the tolerance error, ϵ, on the interface compatibility, $\mathbb{B}\mathbf{u} < \epsilon$, the projection iteration approach in eq. (12) fulfils this requirement by projecting the solution in the constraint hyperplane. On the other hand, the projection formulation needs to be solved by a iterative solver, which requires good preconditioner to achieve numerical scalability. Therefore we combine the right-hand side of the projected inverse version in eq. (12) with the left-hand side of the dual version in eq. (5) in order to have a robust and scalable formulation for the inverse iteration. Mathematically one writes

$$\begin{bmatrix} \mathbb{K} & \mathbb{B}^T \\ \mathbb{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_{n+1} \\ \tilde{\lambda}_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbb{P}_{\mathbb{B}}^T \mathbb{M} \mathbb{P}_{\mathbb{B}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_n \\ \tilde{\lambda}_n \end{bmatrix} \tag{14}$$

Therefore a projected Arnoldi algorithm can be used to build the constrained orthogonal Krylov bases vectors. This method is exploited in [21] for updating eigenvalues in nonlinear dynamic problems. Moreover in [22] the projected Arnordi is used together with PCG method to efficiently compute cyclic eigenmodes. The procedure is summarized in algorithm 1.

Algorithm 1 Projected Arnoldi Iteration

```
• Parameters: $\mathbb{K}, \mathbb{M}, \mathbb{B}, \mathbb{P}_{\mathbb{B}}$ and $n_{int}$.
Initialization
    Start with a projected arbitrary vector $\mathbf{u}^0 = \mathbb{P}_{\mathbb{B}}\mathbf{r}$ and $\beta_0 = 0$, where $\mathbf{r}$ is a random vector.
First step iteration
    Compute a projected unitary vector $\mathbf{v}_1 = \frac{\mathbb{P}_{\mathbb{B}}\mathbb{M}\mathbb{P}_{\mathbb{B}}\mathbf{u}^0}{||\mathbf{u}^0||_{\mathbb{P}_{\mathbb{B}}\mathbb{M}\mathbb{P}_{\mathbb{B}}}}$
    Solve the dual static-like problem using FETI-solver in alg. 2 : $\begin{bmatrix} \mathbb{K} & \mathbb{B}^T \\ \mathbb{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{0} \end{bmatrix}$
    Compute $\alpha_1 = \mathbf{u}_1^T \mathbf{v}_1$
    Compute a new orthogonal constrained Krylov vector $\mathbf{v}_2 = \mathbf{u}_1 - \alpha_1 \mathbf{v}_1$
For $j = 2, ..., n_{int}$
    compute $\beta_j = ||\mathbf{u}_{(j-1)}||_{\mathbb{P}_{\mathbb{B}}\mathbb{M}\mathbb{P}_{\mathbb{B}}}$
    if $\beta_j \neq 0$ then compute $\mathbf{v}^j = \mathbb{P}_{\mathbb{B}}\mathbb{M}\mathbb{P}_{\mathbb{B}}\mathbf{u}_{(j-1)}/\beta_j$
    Compute $\mathbf{u}_j$ by solving the dual static like problem for $\mathbf{v}^j$ using FETI-solver in alg. 2.
    Compute $\alpha_j = \mathbf{u}_j^T \mathbf{v}_j$
    Compute a new constrained Krylov vector $\mathbf{v}_{j+1} = \mathbf{u}_j - \alpha_j \mathbf{v}_j - \beta_j \mathbf{v}_{(j-1)}$
```

The bottleneck of alg. (1) is the solution of the dual linear system presented in (5). In the present work, the linear system is solved by a FETI algorithm which is described is alg. 2. Even though Dirichlet preconditioner is optimal as shown by Farhat in [23], the Lumped preconditioner is selected due to its low cost and good efficiency for the numerical examples described below.

3 Numerical Examples

The algorithms described in the previous sections were implemented in Python using Scipy [24], NumPy and mpi4py. Two numerical examples are considered to illustrate the impact of the different algorithms strategies in the computations of the eigenpairs. In both cases, structural steel constants are used ($E = 210GPa$, $\nu = 0.3$ and $= 7500Kg/m^3$) as the material property. In order to take advantage of the Scipy wrapper for the ARPACK [6], we construct a sparse linear operator for the inverse iteration for the three formulations: global assembly, Dual, and Projected-Dual. These linear operators are passed to the eigensolver method *eigsh* provided by the scipy linear algebra sparse module. The intention is to

Algorithm 2 FETI Algorithm

```
• Parameters: $\mathbb{K}, \mathbb{B}$, and $\mathbf{v}$.
    Initialization
    Compute the pseudoinverse and the Kernel of $\mathbb{K} \rightarrow \{\mathbb{K}^{+}, \mathbb{R}\}$
    Assemble $\mathbb{G} := -\mathbb{R}^T\mathbb{B}^T, \mathbf{d} := \mathbb{B}\mathbb{K}^{+}\mathbf{v}$, and $\mathbf{e} := -\mathbb{R}^T\mathbf{v}$
    Construct the parallel linear operators $\mathbb{F} := \mathbb{B}\mathbb{K}^{+}\mathbb{B}^T$ and $\mathbb{P}_{\mathbb{G}} := \mathbb{I} - \mathbb{G}^T(\mathbb{G}\mathbb{G}^T)^{-1}\mathbb{G}$
    Solving the interface force $\lambda_{im} \in Im\{\mathbb{G}\}$ and compute $\tilde{\mathbf{d}}$
    $\lambda_{im} = \mathbb{G}^T(\mathbb{G}\mathbb{G}^T)^{-1}\mathbf{e}$
    $\tilde{\mathbf{d}} = \mathbf{d} - \mathbb{F}\lambda_{im}$
    Solve Dual Interface problem by PCPG
    Initialize variables: $\mathbf{r}^0 = \tilde{\mathbf{d}}$, $\lambda_{ker} = \mathbf{0}, \beta^0 = 0, \mathbf{p}^0 = \mathbf{0}$
    Iterate $k = 1, 2, ...,$ while$||\mathbf{w}^k|| >$ tolerance
        Projection $\mathbf{w}^{k-1} = \mathbb{P}_{\mathbb{G}}\mathbf{r}^{k-1}$
        Precondition $\mathbf{z}^{k-1} = \tilde{\mathbb{F}}^-1\mathbf{w}^{k-1}$
        Projection $\mathbf{y}^{k-1} = \mathbb{P}_{\mathbb{G}}\mathbf{z}^{k-1}$
        $\beta^k = \mathbf{y}^{k-1T}\mathbf{w}^{k-1}/\mathbf{y}^{k-2T}\mathbf{w}^{k-2} \quad if \quad k > 1$
        $\mathbf{p}^k = \mathbf{y}^{k-1} + \beta^k\mathbf{p}^{k-1}$
        $\alpha^k = \mathbf{y}^{k-1T}\mathbf{w}^{k-1}/\mathbf{p}^{kT}\mathbb{F}\mathbf{p}^k$
        $\lambda_{ker}^k = \lambda_{ker}^{k-1} + \alpha^k\mathbf{p}^k$
        $\mathbf{r}^k = \mathbf{r}^k - \alpha^k\mathbf{p}^k\mathbb{F}\mathbf{p}^k$
    Assemble $\mathbf{u}$
        $\lambda = \lambda_{ker} + \lambda_{im}$
        $\tilde{\alpha} = (\mathbb{G}\mathbb{G}^T)^{-1}\mathbb{G}(\mathbf{d} - \mathbb{F}\lambda)$
        $\mathbf{u} = \mathbb{K}^{+}(\mathbf{v} - \mathbb{B}^T\lambda) + \mathbb{R}\tilde{\alpha}$
```

compare the results achieved by both dual strategies with the reference solution given by global assembly formulation, which is the most accurate and robust method. We vary the tolerances in the FETI solver to verify its impact in the computation of the eigenmodes. For both cases, 10 eigenpairs are computed where the eigensolver tolerance is set to 1.0^{-8} and maximum iterations n_{int} is set to 200.

The first example is a 2D plane-stress beam with width equals 100 m and height equals 10 m. The global domain is decomposed in 12 subdomains with 252 dofs each, and the local matrices are assembled using a AMFE, which is a Finite element framework written in Python. We solve the dual linear system with 3 different tolerances, 10^{-5},10^{-6}, and 10^{-7}, for the stop criteria on the interface compatibility defined in alg. 2. The first 6 mode shapes are illustrated in figure 3.

Table (1) shows the maximum, average, and minimum of the absolute value of the Pearson correlation between the first 10 eigenmodes of the reference solution versus the classical dual formulation, and versus the projected dual version.

Table 1. Pearson correlation between reference solution versus Dual and Dual-Projected formulation.

	Dual with Projection			Dual without Projection		
Tolerance	Max Corr.	Avg Corr.	Min Corr.	Max Corr.	Avg Corr	Min Corr
10^{-5}	0.99	0.42	0.01	0.99	0.47	0.02
10^{-6}	0.99	0.55	0.01	0.99	0.52	0.02
10^{-7}	0.99	0.97	0.90	0.99	0.88	0.46

Analyzing table (1), it is possible to verify that average and minimum values of the correlation are higher in the projected formulation which shows that this method is more accurate than the classical

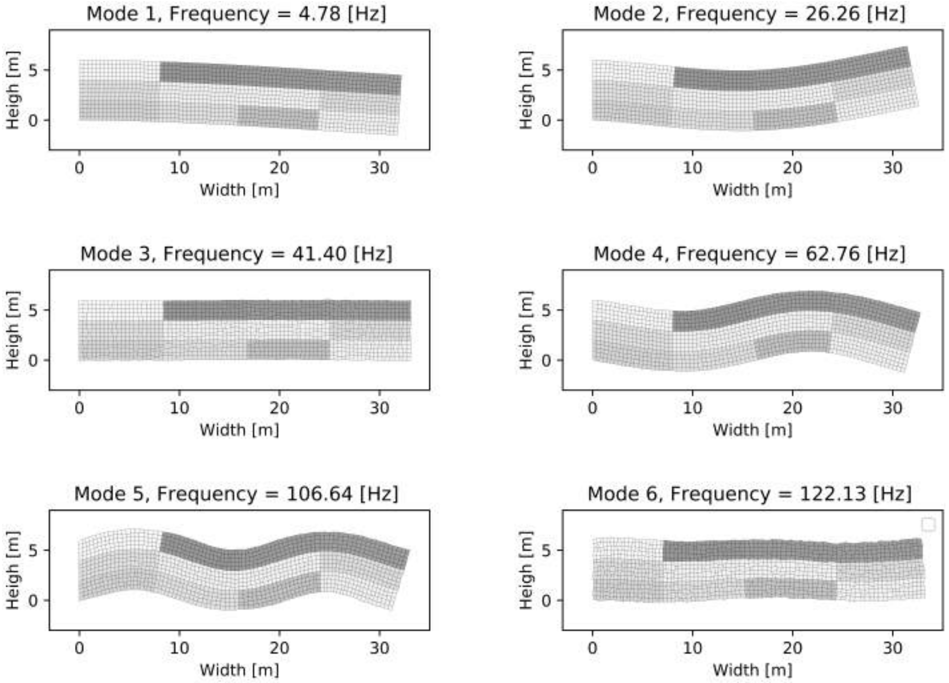

Figure 3. First 6 mode shapes solved by the Dual-Projected formulation using a tolerance equal 10^{-7}

version. Figure (4) illustrates the 8th computed mode shape for the three formulations where different tolerances are given to the iterative linear solver. It is possible to observe that the mode shape given by the projected formulation is more similar to the reference sector, when compared with the classical dual method.

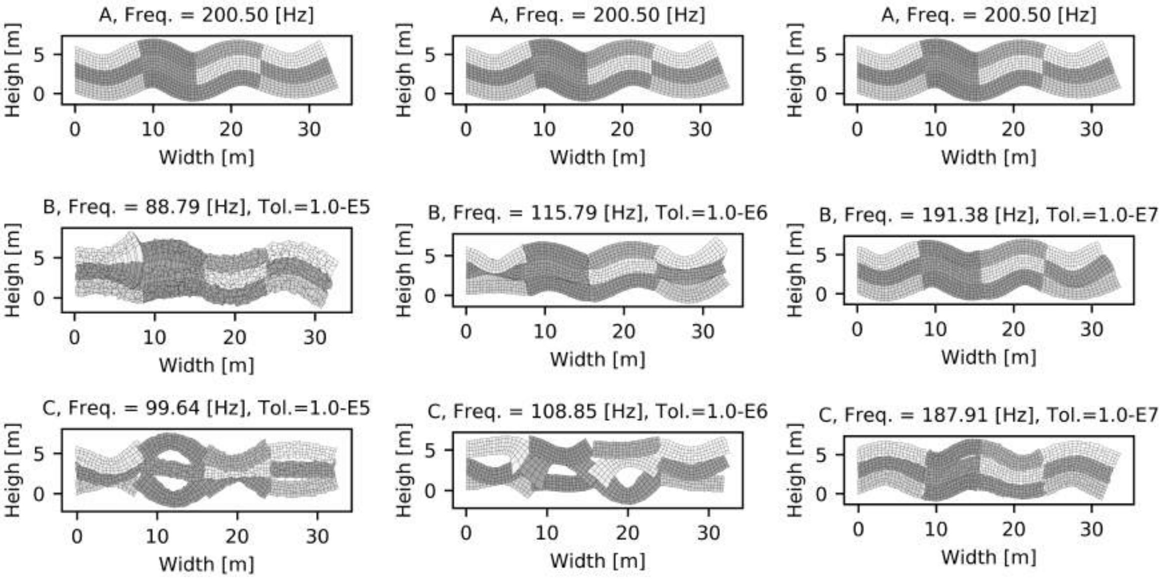

Figure 4. 8th mode shape for (A) Reference solution , (B) Projected-Dual, and (C) Dual using difference interface tolerances.

Looking fig. (4) we clearly notice that the projected formulation fulfills the compatibility constraint, once gaps cannot be seen between the interfaces of the subdomains. Moreover, it is possible to observe by the results in the table (1) that the dual formulation without the projection is more sensitivity to the solver tolerances.

The second numerical example is a 3D model of a simple bladed-disk made of steel with 400mm of diameter. The whole geometry is composed by 24 sector with 1536 dofs, which are considered as subdomains. We apply the same procedure described for the 2D case to compute the eigenpairs, however we solve the problem with 10^{-8}, 10^{-9}, and 10^{-10} as tolerances for the interface iterative solver. The small tolerances are necessary due to the smaller dimension and higher frequencies of the structure. Four difference eigenshapes are shown in Figure (5) for the tolerance equal 10^{-9}. Table 2 shows the correlation between the eigenmodes of the global assembly, classical FETI, and FETI with the projection.

Table 2. Pearson correlation between reference solution versus Dual and Dual-Projection formulation

	Dual with Projection			Dual without Projection		
Tolerance	Max Corr.	Avg Corr.	Min Corr.	Max Corr.	Avg Corr	Min Corr
10^{-8}	0.97	0.18	0.00	0.98	0.2	0.00
10^{-9}	0.99	0.51	0.01	0.99	0.52	0.01
10^{-10}	0.99	0.64	0.26	0.99	0.39	0.01

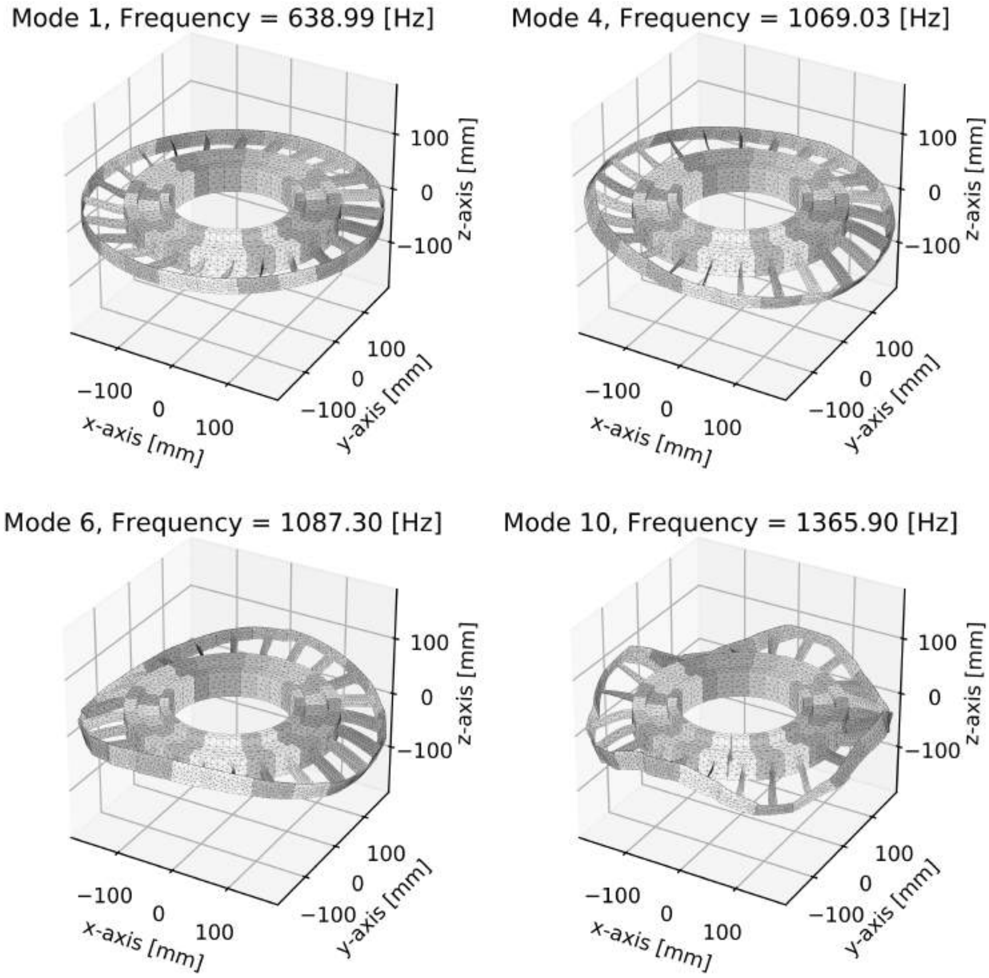
Mode 1, Frequency = 638.99 [Hz]
Mode 4, Frequency = 1069.03 [Hz]
Mode 6, Frequency = 1087.30 [Hz]
Mode 10, Frequency = 1365.90 [Hz]
100
0
−100
z-axis [mm]
x-axis [mm]
y-axis [mm]

Figure 5. 4 different mode shapes computed by the Dual-Projected formulation with tolerance equal 10^{-9}

For the second case, we observe the same behavior as in the 2D case. The projected formulation is converging quicker to the correct mode shapes whereas the standard formulation without projection is not improving even when small tolerances are used.

3.1 FETI Solver Parallel Scalability

To illustrate the parallel efficiency of the FETI algorithm implemented in Python we decomposed gradually a 2D rectangular body with width $=$ 800[mm] and height$=$ 200[mm] in identical squared subdomains with sizes 8x2,12x3,14x4,2x5. The element edge size of the global mesh is set equal 0.25mm creating a problem with 2,880,000 nodes and 5,760,000 displacement degrees of freedom without any domain decomposition. The body is fixed in all direction in the left side and it is loaded by a body force equal $10N$ in the negative Y-direction. The tolerance of the interface problem is set to 1.0^{-5}. The tests were executed on the Salomon supercomputer located in IT4Innovations in Czech Republic. In table 3, we report the computational time for preprocessing which involves the computation of pseudo-inverse, kernel and factorization of $\mathbb{G}\mathbb{G}^T$, which are performed in parallel. Besides that, the number of iterations, total solver time and iteration duration are presented.

Table 3. Strong Parallel Scalability of FETI-solver varying the number of partitions.

Number of Subdomains	16	36	64	100
Total time [s]	298.63	117.02	70.84	54.99
Number of iterations	30	38	41	42
PCPG time [s]	41.42	36.12	32.89	33.39
Preprocessing time [s]	201.33	55.03	23.95	12.04
Interface size	18,472	32,158	45,996	59,854
Kernel size	42	99	180	285
Displacement size	5,787,232	5,786,424	5,814,528	5,828,200
Efficiency [%]	100.0	113.4	105.4	86.9

As summarized in table 3, it is possible to verify a good strong parallel scalability of the implemented FETI algorithm up to 100 subdomains. The number of PCPG iterations increases slowly with the number of subdomains, whereas the factorization time is drastically reducing as presented in the preprocessing time row. Therefore, the use of FETI solver for the inverse iteration can reduce the overall computational time of the eigenpairs calculations in the presence of computational parallel resource.

4 Conclusion

We described how to use FETI solvers to speed-up the computation of large scale eigenvalue problem. Special attention was given for the interface tolerance and how it affects the mode shapes computation. We showed that the direct application of FETI solvers for the computation of generalized eigenvalue problem can produce modes that are not physical modes. Therefore, we introduced a projection formulation to enforce the compatibility after every inverse iteration and consequently stabilize the eigensolver. The projected formulation together with the classical FETI algorithm can be used safely and we showed its good parallel strong scalability up to 100 domains for a problem with originally 5,760,000 dofs.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 721865.

References

[1] Bampton, M. C. C. & Craig, Jr., R. R., 1968. Coupling of substructures for dynamic analyses. *AIAA Journal*, vol. 6, n. 7, pp. 1313–1319.

[2] Rixen, D. J., 2004. A dual craig–bampton method for dynamic substructuring. *Journal of Computational and applied mathematics*, vol. 168, n. 1-2, pp. 383–391.

[3] Rubin, S., 1975. Improved component-mode representation for structural dynamic analysis. *AIAA journal*, vol. 13, n. 8, pp. 995–1006.

[4] Grimes, R. G., Lewis, J. G., & Simon, H. D., 1994. A shifted block lanczos algorithm for solving sparse symmetric generalized eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, vol. 15, n. 1, pp. 228–272.

[5] Morgan, R. B. & Scott, D. S., 1993. Preconditioning the lanczos algorithm for sparse symmetric eigenvalue problems. *SIAM Journal on Scientific Computing*, vol. 14, n. 3, pp. 585–593.

[6] Lehoucq, R. B., Sorensen, D. C., & Yang, C., 1998. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, volume 6. Siam.

[7] Lehoucq, R., Sorensen, D., & Yang, C., 1997. Arpack users' guide: Solution of large scale eigenvalue problems with implicitly restarted arnoldi methods. *Software Environ. Tools*, vol. 6.

[8] Lehoucq, R. B., 2001. Implicitly restarted arnoldi methods and subspace iteration. *SIAM Journal on Matrix Analysis and Applications*, vol. 23, n. 2, pp. 551–562.

[9] Morgan, R. B., 2000. Implicitly restarted gmres and arnoldi methods for nonsymmetric systems of equations. *SIAM Journal on Matrix Analysis and Applications*, vol. 21, n. 4, pp. 1112–1135.

[10] Farhat, C. & Roux, F.-X., 1991. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, vol. 32, n. 6, pp. 1205–1227.

[11] Farhat, C. & Mandel, J., 1998. 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*, vol. 155, n. 1-2, pp. 129–151.

[12] Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., & Rixen, D., 2001. 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*, vol. 50, n. 7, pp. 1523–1544.

[13] Dostál, Z., Horák, D., & Kučera, R., 2006. Total feti—an easier implementable variant of the feti method for numerical solution of elliptic pde. *Communications in Numerical Methods in Engineering*, vol. 22, n. 12, pp. 1155–1162.

[14] Spillane, N., Dolean, V., Hauret, P., Nataf, F., & Rixen, D. J., 2013. Solving generalized eigenvalue problems on the interfaces to build a robust two-level feti method. *Comptes Rendus Mathematique*, vol. 351, n. 5-6, pp. 197–201.

[15] Gosselet, P., Rixen, D., Roux, F.-X., & Spillane, N., 2015. Simultaneous feti and block feti: Robust domain decomposition with multiple search directions. *International Journal for Numerical Methods in Engineering*, vol. 104, n. 10, pp. 905–927.

[16] Geradin, M., Coulon, D., & Delsemme, J.-P., 1997. Parallelization of the samcef finite element software through domain decomposition and feti algorithm. *The International Journal of Supercomputer Applications and High Performance Computing*, vol. 11, n. 4, pp. 286–298.

[17] Rixen, D. & Lohman, R., 2005. Efficient computation of eigenmodes of quasi-cyclic structures. In *Proceedings of the IMAC-XXIII conference and exposition on structural dynamics*, pp. 1–13.

[18] Cardona, A. & Geradin, M., 1989. Time integration of the equations of motion in mechanism analysis. *Computers & structures*, vol. 33, n. 3, pp. 801–820.

[19] Klerk, D. d., Rixen, D. J., & Voormeeren, S., 2008. General framework for dynamic substructuring: history, review and classification of techniques. *AIAA journal*, vol. 46, n. 5, pp. 1169–1181.

[20] Lehoucq, R. B. & Sorensen, D. C., 1996. Deflation techniques for an implicitly restarted arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications*, vol. 17, n. 4, pp. 789–821.

[21] Yang, D. & Ajjarapu, V., 2007. Critical eigenvalues tracing for power system analysis via continuation of invariant subspaces and projected arnoldi method. In *2007 IEEE Power Engineering Society General Meeting*, pp. 1–9. IEEE.

[22] Jenovencio, G. & Rixen, D., 2019. A dual formulation id cyclic symmetry: Application in free vibration analysis. proceedings of 7th eccomas thematic conference on computational methods in structural dynamics and earthquake engineering (compdyn 2019).

[23] Farhat, C., Mandel, J., & Roux, F. X., 1994. Optimal convergence properties of the feti domain decomposition method. *Computer methods in applied mechanics and engineering*, vol. 115, n. 3-4, pp. 365–385.

[24] Jones, E., Oliphant, T., Peterson, P., et al., 2001 – http://www.scipy.org/. SciPy: Open source scientific tools for Python.