Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

Parallel adaptive FETI-DP using lightweight asynchronous dynamic load balancing

Axel Klawonn[1,2] | Martin J. Kühn[3] | Oliver Rheinbach[4]

[1]Department of Mathematics and Computer Science, University of Cologne, Germany
[2]Center for Data and Simulation Science, University of Cologne, Germany
[3]Parallel Algorithms Team, CERFACS (Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique), France
[4]Institut für Numerische Mathematik und Optimierung, Technische Universität Bergakademie Freiberg, Germany

Correspondence
Martin J. Kühn, Parallel Algorithms Team, CERFACS (Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique). Email: martin.kuehn@cerfacs.fr

Present Address
42 Avenue Gaspard Coriolis, 31057 Toulouse Cedex 01, France

Summary

A parallel FETI-DP domain decomposition method using an adaptive coarse space is presented. The implementation builds on a recently introduced adaptive FETI-DP approach for elliptic problems in three dimensions and uses small, local eigenvalue problems for faces and, additionally, for a small number of edges. The condition number of the preconditioned operator then satisfies a bound which is independent of coefficient heterogeneities in the problem. The computational cost of the local eigenvalue problems is not negligible, and also a significant load imbalance can be introduced. As a remedy, certain eigenvalue problems are discarded by a theory-guided heuristic strategy, based on the diagonal entries of the stiffness matrices. Additionally, a lightweight pairwise dynamic load balancing strategy is implemented for the eigenvalue problems. The load balancing is supervised by an orchestrating rank using asynchronous point-to-point communication. The resulting method shows good weak and strong scalability up to thousands of cores while fast convergence is obtained even for heterogeneous problems.

KEYWORDS:
high performance computing, parallelization, domain decomposition methods, partial differential equations, highly heterogeneous, adaptive coarse spaces

1 | INTRODUCTION

In recent years, different approaches for the construction of adaptive coarse spaces for domain decomposition methods have been studied extensively; see, e.g., [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40].

However, parallel implementations of such adaptive coarse spaces [41,11,12,42,43,26,44,37,39] are not simple and are therefore still less common; see also the parallel implementation of the closely related ACMS special finite element method in [45]. Our work is mainly focussed on adaptive FETI-DP methods.

The ***F**inite **E**lement **T**earing and **I**nterconnecting* method was introduced in [46] and studied in early works in [47,48,49]. For condition number estimates, see [50,51]. In [52,53,54], the FETI method was applied to time-dependent, plate, and shell problems. The two-level FETI method was introduced in [53]. Following the two-level FETI method, the FETI-DP method was first introduced in [55,56]. For condition number estimates, see [57,58,59]. FETI-DP and its variants are highly scalable domain decomposition methods; see, i.a., [60,61,62,63]. Different successful approaches to further extend the scalability of FETI-DP methods to large supercomputers have been proposed, e.g., in [64,62,65].

In this article, we present a parallel implementation of an adaptive FETI-DP algorithm. This method constructs an automatic coarse space from local eigenvalue problems and uses local eigenvectors as coarse constraints, implemented by a (generalized)

This is the accepted version of the following article: Axel Klawonn, Martin J. Kühn, and Oliver Rheinbach (2020) Parallel adaptive FETI-DP using lightweight asynchronous dynamic load balancing. Int J Numer Methods Eng. 2020; 121: 621– 643 which has been published in final form at https://onlinelibrary.wiley.com/doi/10.1002/nme.6237.

transformation of basis; see[25,38]. Originally, deflation or balancing was used, this is, however, more sensitive to an inexact solution of the coarse problem. Deflation is a viable option if an exact solution of the coarse problem is affordable.

Using the adaptive coarse space, the condition number of the preconditioned FETI-DP operator satisfies a bound which only depends on a user-given tolerance and geometric properties of the domain decomposition[25,38]. Note that, since the condition number is controlled a priori, the size of the coarse space is only given ex post. A comparison of adaptive algorithms should therefore take into account the size of the resulting coarse space. Based on the numerical comparison of coarse spaces (see the number of eigenvectors #EV in, e.g., Tables 8.4 and 8.6 in[21]), we decided to implement the algorithm presented here.

Adaptive coarse space enrichment for FETI-DP and BDDC methods was first proposed in[3] as a theory guided heuristic technique to improve robustness for difficult problems. This algorithm uses local eigenvalue problems for edges in two dimensions and for faces in three dimensions. A theoretical bound was then first given in[21] for the adaptive algorithm in two dimensions proposed in[3]. Also a comparison of different adaptive approaches combined with different scalings was given. In[25,38] the bound was generalized to three dimensions but using additional local eigenvalue problems for edges with multiplicity four or larger, i.e., in addition to the eigenvalue problems for (closed) faces. The number of edges with a multiplicity four or larger is typically small for meshes from mesh partitioners, i.e., only a small number of edge eigenvalue problems is needed.

2 | MODEL PROBLEM AND GEOMETRY

Let $\Omega \subset \mathbb{R}^3$ be a bounded polyhedral domain with Dirichlet boundary $\partial\Omega_D \subset \partial\Omega$ and Neumann boundary $\partial\Omega_N := \partial\Omega \setminus \partial\Omega_D$. We consider the weak formulation of compressible linear elasticity, i.e., we are interested in finding $u \in H^1_0(\Omega, \partial\Omega_D)^3$ such that

$$a(u,v) = F(v) \quad \forall v \in H^1_0(\Omega, \partial\Omega_D)^3,$$

where $a(u,v) := \int_\Omega 2\mu\varepsilon(u) : \varepsilon(v)\mathrm{d}x + \int_\Omega \lambda \mathrm{div}(u)\mathrm{div}(v)\mathrm{d}x$, $F(v) := \int_\Omega f \cdot v\mathrm{d}x + \int_{\partial\Omega_N} g \cdot v\mathrm{d}s$. Here, λ and μ are the Lamé constants, and $\varepsilon(u) : \varepsilon(v) = \mathrm{tr}(\varepsilon(u)^T\varepsilon(v))$ where $\varepsilon(u) = \frac{1}{2}(\nabla u + \nabla u^T)$. The Lamé constants can be calculated from *Young's modulus* $\mathrm{E} > 0$ and *Poisson's ratio* $\nu \in (0, \frac{1}{2})$ by $\lambda = \frac{\mathrm{E}\nu}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{\mathrm{E}}{2(1+\nu)}$. The functions $f : \Omega \to \mathbb{R}^3$ and $g : \partial\Omega_N \to \mathbb{R}^3$ denote volume and surface force, respectively. The domain Ω is decomposed into N nonoverlapping subdomains Ω_i, $i = 1, \dots, N$. The interface Γ is defined as the union of the interior subdomain boundaries, i.e., $\Gamma := \{x \in \overline{\Omega_i} \cap \overline{\Omega_j};\ i \neq j\}$. We use piecewise linear conforming finite elements. In three dimensions, the interface can be decomposed into vertices, edges, and faces; cf.[59]. Edges and faces are understood as open sets. A face between two arbitrary subdomains Ω_i and Ω_j will be denoted by $\mathcal{F}^{ij}$ while we denote edges between Ω_i, Ω_j, Ω_l and possibly more subdomains by $\mathcal{E}^{il}$. Vertices of Ω_i that belong to multiple subdomains are denoted by $\mathcal{V}^{ik}$.

Finally, we introduce the finite element spaces used for the FETI-DP and BDDC methods. For $i = 1, \dots, N$, by $W^h(\Omega_i)$ we denote the local finite element space on Ω_i. The local trace space $W_i := W^h(\Gamma_i)$ is defined on $\Gamma_i := \overline{\Omega_i} \cap \Gamma$. We also introduce the global product space $W := \Pi_{i=1}^N W_i$ and denote the space of functions that are continuous on the interface by $\widehat{W} \subset W$.

3 | FETI-DP AND GENERALIZED TRANSFORMATION-OF-BASIS APPROACH

3.1 | FETI-DP methods

In FETI-DP methods[55,56,66] the degrees of freedom are partitioned into interior, dual, and primal degrees of freedom, denoted by an index I, Δ', and Π', respectively. The local stiffness matrix $K^{(i)}$ and the local load vector $f^{(i)}$ corresponding to the subdomain Ω_i, $i = 1, \dots, N$, are written

$$K^{(i)} = \begin{bmatrix} K^{(i)}_{II} & K^{(i)T}_{\Delta' I} & K^{(i)T}_{\Pi' I} \\ K^{(i)}_{\Delta' I} & K^{(i)}_{\Delta'\Delta'} & K^{(i)T}_{\Pi'\Delta'} \\ K^{(i)}_{\Pi' I} & K^{(i)}_{\Pi'\Delta'} & K^{(i)}_{\Pi'\Pi'} \end{bmatrix}, f^{(i)} = \begin{bmatrix} f^{(i)}_I \\ f^{(i)}_{\Delta'} \\ f^{(i)}_{\Pi'} \end{bmatrix}.$$

Define $K^{(i)}_{BB}$ as the upper-left 2x2 block in $K^{(i)}$, moreover $K^{(i)}_{\Pi' B} := \begin{bmatrix} K^{(i)}_{\Pi' I} & K^{(i)}_{\Pi'\Delta'} \end{bmatrix}$, and $f^{(i)}_B := \begin{bmatrix} f^{(i)T}_I & f^{(i)T}_{\Delta'} \end{bmatrix}^T$. Next, define $K^{(i)}_{\Gamma\Gamma}$ as the lower right 2x2 block in $K^{(i)}$ and $K^{(i)T}_{\Gamma I} := \begin{bmatrix} K^{(i)T}_{\Delta' I} & K^{(i)T}_{\Pi' I} \end{bmatrix}$. Then, we define the block matrices $K_{II} := \mathrm{diag}_{i=1}^N K^{(i)}_{II}$, $K_{\Delta'\Delta'} := \mathrm{diag}_{i=1}^N K^{(i)}_{\Delta'\Delta'}$ and $K_{\Pi'\Pi'} := \mathrm{diag}_{i=1}^N K^{(i)}_{\Pi'\Pi'}$, as well as $K_{BB} := \mathrm{diag}_{i=1}^N K^{(i)}_{BB}$, $K_{\Gamma\Gamma} := \mathrm{diag}_{i=1}^N K^{(i)}_{\Gamma\Gamma}$, and $K_{\Gamma I}$ as the corresponding global off-diagonal block.

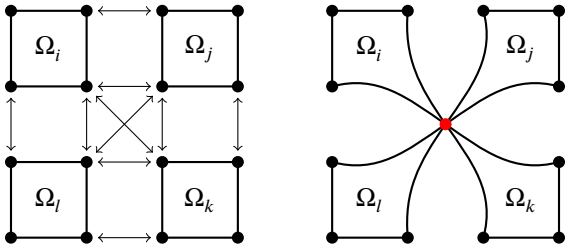


FIGURE 1 Cross sectional view of four subdomains sharing an edge. Arrows symbolize redundant Lagrange multipliers in FETI-DP (left). Using a transformation of the initial basis and partial subassembly in the red node, a primal constraint is now enforced between all four subdomains (right).

We also need assembly operators. The first, $R_{\Pi'}^T := \left[R_{\Pi'}^{(1)T}, \dots, R_{\Pi'}^{(N)T}\right]$, performs the assembly in the primal variables $u_{\Pi'}^{(i)}$ and is needed for FETI-DP and BDDC. The second, $R_{\Delta'}^T := \left[R_{\Delta'}^{(1)T}, \dots, R_{\Delta'}^{(N)T}\right]$, performs assembly in the dual variables $u_{\Delta'}^{(i)}$ and is only needed for BDDC.

For FETI-DP, in place of $R_{\Delta'}$, we need a jump operator $B_\Gamma = \left[B_\Gamma^{(1)}, \dots, B_\Gamma^{(N)}\right]$ which is built from values 0 and ± 1 such that $B_\Gamma u = 0$ holds for $u \in \widehat{W}$. The jump operator B on the whole domain is obtained by adding zero columns for interior varibales to B_Γ, i.e., $B = (0, B_\Gamma)$ for a corresponding ordering of the variables.

By assembly in the primal variables, we obtain

$$\widetilde{K}_{\Pi'\Pi'} = \sum_{i=1}^{N} R_{\Pi'}^{(i)T} K_{\Pi'\Pi'}^{(i)} R_{\Pi'}^{(i)}, \quad \widetilde{K}_{\Pi' B} = \left[R_{\Pi'}^{(1)T} K_{\Pi' B}^{(1)}, \dots, R_{\Pi'}^{(N)T} K_{\Pi' B}^{(N)}\right],$$

$$\widetilde{f} = \left[f_B^T, (\sum_{i=1}^{N} R_{\Pi'}^{(i)T} f_{\Pi'}^{(i)})^T\right]^T, \text{ and } \widetilde{S}_{\Pi'\Pi'} = \widetilde{K}_{\Pi'\Pi'} - \widetilde{K}_{\Pi' B} K_{BB}^{-1} \widetilde{K}_{\Pi' B}^T.$$

Then, the FETI-DP master system is given by

$$\begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi' B}^T & B_B^T \\ \widetilde{K}_{\Pi' B} & \widetilde{K}_{\Pi'\Pi'} & 0 \\ B_B & 0 & 0 \end{bmatrix} \begin{bmatrix} u_B \\ \widetilde{u}_{\Pi'} \\ \lambda \end{bmatrix} = \begin{bmatrix} \widetilde{f} \\ 0 \\ 0 \end{bmatrix}. \tag{1}$$

Here, B_B is obtained from B by removing the columns corresponding to Π'. It is has the form $B_B = \left[B_B^{(1)}, \dots, B_B^{(N)}\right]$. The (unpreconditioned) FETI-DP system

$$F\lambda = d \tag{2}$$

is obtained after elimination of $\widetilde{u}^T = [u_B^T, \ \widetilde{u}_{\Pi'}^T]^T$. The system eq. (2) is solved using the standard FETI-DP Dirichlet preconditioner.

For heterogeneous problems, the incorporation of a suitable scaling D in the preconditioner is an important ingredient of FETI-DP methods[66]. A suitable scaling is also important in adaptive methods since, otherwise, the adaptive coarse space may become rather large; see, e.g., [21, Tables 8.3] or [38, Table 5.6, 5.7, 5.8].

3.2 | The generalized transformation-of-basis approach for FETI-DP

We briefly review the operators needed to introduce the generalized transformation-of-basis approach; see also[67,38]. In section 4.4, we will explain in detail how this method is implemented in a parallel context; this has not been treated in our previous articles[67,38] which focus on general theory and on MATLAB results with some preliminary parallel insights.

Let us assume that a FETI-DP method given by

$$M^{-1}F\lambda = (B_{D,\Gamma}\widetilde{S}_{\Gamma\Gamma}B_{D,\Gamma}^T)\,(B_\Gamma \widetilde{S}_{\Gamma\Gamma}^{-1} B_\Gamma^T) = M^{-1}d \tag{3}$$

was set up with a coarse space Π' which ensures the invertibility of the local problems. The constraints incorporated into the coarse space are denoted a priori constraints, since they are chosen beforehand and therefore do not depend on, e.g., the scaling

or the coefficients in the underlying partial differential equation. Let us also assume that the scaling D is nonconstant on several faces and edges, which is generally the case if heterogeneous problems are considered.

In order to accelerate the convergence of the method, we now use (problem-dependent) additional constraints localized to single faces and edges. Note that, in case of a nonconstant scaling on a face or an edge denoted by $\mathcal{Z}$, an (a posteriori) constraint c on the face or the edge leads to nonzero values in the degrees of freedom chosen for partial subassembly; see [67,39]; the theoretical assumptions to prove the standard condition number bound for the standard transformation-of-basis approach are therefore violated since these proofs rely on zero values in the coarse-level components; see, e.g., [59, p. 1537] or [68, p. 266]. This problem is adressed by the generalized transformation-of-basis approach [67,38].

Note that, in contrast to the standard approach, we have to distinguish between a priori constraints in Π' and a posteriori constraints in Π when extending the coarse space Π' to $\widehat{\Pi} := \Pi' \cup \Pi$. By $T_{\mathcal{Z},\Pi}$, we denote the a posteriori constraint(s) associated with $\mathcal{Z}$. We then obtain the transformation matrix

$$T_{\mathcal{Z}} = \begin{pmatrix} T_{\mathcal{Z},\Pi} & T_{\mathcal{Z},\Delta} \end{pmatrix} \tag{4}$$

by requesting $T_{\mathcal{Z}}$ to be square and invertible. For simplicity, we here assume T to be orthogonal. Note that $T_{\mathcal{Z}}$ must be identical for all local subdomains sharing $\mathcal{Z}$. For any subdomain Ω_i, $i = 1, \dots, N$, the local transformation matrix $T_\Gamma^{(i)}$, $i = 1, \dots, N$, is a block diagonal matrix consisting of the blocks $T_{\mathcal{Z}}$ corresponding to the local faces and edges $\mathcal{Z}$ of Ω_i, $i = 1, \dots, N$ and the identity on the a priori primal variables. The global, transformed Schur complement writes $T_\Gamma \widetilde{S}_{\Gamma\Gamma} T_\Gamma$, where T_Γ contains the local components $T_\Gamma^{(i)}$.

In order to enforce the constraints, we introduce a partial subassembly operator R^T which assembles the additionally chosen primal variables. We also use the multiplicity-weighted assembly operator $R_\mu^T := (R^T R)^{-1} R^T$. Then, the transformed and assembled Schur complement is given by

$$\widehat{\widetilde{S}}_{\Gamma\Gamma} := R^T T_\Gamma^T \, \widetilde{S}_{\Gamma\Gamma} \, T_\Gamma R. \tag{5}$$

Taking the transformation and assembly into account, we obtain

$$\begin{aligned} \widehat{M}_T^{-1} \widehat{F} \; &: = (\widehat{B}_{D,\Gamma} \widehat{\widetilde{S}}_{\Gamma\Gamma} \widehat{B}_{D,\Gamma}^T) \, (\widehat{B}_\Gamma \widehat{\widetilde{S}}_{\Gamma\Gamma}^{-1} \widehat{B}_\Gamma^T) \\ &= \Big(B_{D,\Gamma} \underbrace{T_\Gamma R_\mu \widehat{\widetilde{S}}_{\Gamma\Gamma} R_\mu^T T_\Gamma^T}_{\text{replaces } \widetilde{S}_{\Gamma\Gamma}} B_{D,\Gamma}^T \Big) \Big(B_\Gamma \underbrace{T_\Gamma R \widehat{\widetilde{S}}_{\Gamma\Gamma}^{-1} R^T T_\Gamma^T}_{\text{replaces } \widetilde{S}_{\Gamma\Gamma}^{-1}} B_\Gamma^T \Big). \end{aligned} \tag{6}$$

representing the operator of the left hand side of the FETI-DP system using the generalized transformation-of-basis approach. Note that, in this approach, we have constraints (and Lagrange multipliers) corresponding to the adaptively chosen primal (i.e., a posteriori) constraints. These are redundant, resulting in additional zero eigenvalues of the FETI-DP operator. They are are, nevertheless, necessary to allow the implementation of the correct scaling, i.e., to allow an interaction of dual and adaptively chosen primal variables.

For a more detailed description, we refer to [67,39]. Note that (6) reduces to the standard approach as presented in [58,68,59,69,70,71] if more restrictive assumptions are made, e.g., if the scaling is constant on each face and each edge.

We then can show [67] that

$$\sigma(\widehat{M}_T^{-1} \widehat{F}) = \sigma(M_{PP}^{-1} F). \tag{7}$$

Here, $\sigma(\widehat{M}_T^{-1} \widehat{F})$ represents the spectrum of the preconditioned FETI-DP system matrix using the generalized transformation-of-basis approach and $\sigma(M_{PP}^{-1} F)$ is the spectrum using the deflation approach; see [67,39] for more details; see [38,39] for the application to adaptive FETI-DP.

4 | ADAPTIVE FETI-DP USING THE GENERALIZED TRANSFORMATION-OF-BASIS APPROACH

In this section, we briefly revisit our adaptive FETI-DP method (see [25,67,38]) and then show how our parallel implementation is realized using plain MPI and the PETSc and SLEPc high performance computing libraries.

4.1 | Local generalized eigenvalue problems for adaptive FETI-DP

This paper describes how to implement in parallel the adaptive approach described in[25]. Also see[25] for additional references.

Let us consider either a face $\mathcal{F}^{ij}$ between the subdomains Ω_i and Ω_j or an edge $\mathcal{E}^{ik}$ between the subdomains Ω_i and Ω_k, where Ω_i and Ω_k have no common face; cf. fig. 1 (left). As in[3] for faces and[25] for faces and edges, we define $B_{\mathcal{F}^{ij}} = \left(B^{(i)}_{\mathcal{F}^{ij}} \; B^{(j)}_{\mathcal{F}^{ij}}\right)$ on the closed face and $B_{\mathcal{E}^{ik}} = \left(B^{(i)}_{\mathcal{E}^{ik}} \; B^{(l)}_{\mathcal{E}^{ik}}\right)$ on the (closed) edge as the submatrix of $\left(B^{(i)} \; B^{(j)}\right)$ and $\left(B^{(i)} \; B^{(l)}\right)$, respectively, consisting of all the rows that contain exactly one $+1$ and one -1. The operators $B_{\mathcal{F}^{ij}}$ and $B_{\mathcal{E}^{ik}}$ provide the local jump on the closure of the face or edge, respectively. Since we assume all vertices to be primal, the operators on the edge can be defined on the interior of the edge only.

Analogously, $B_{D,\mathcal{F}^{ij}} = \left(B^{(i)}_{D,\mathcal{F}^{ij}} \; B^{(j)}_{D,\mathcal{F}^{ij}}\right)$ and $B_{D,\mathcal{E}^{ik}} = \left(B^{(i)}_{D,\mathcal{E}^{ik}} \; B^{(l)}_{D,\mathcal{E}^{ik}}\right)$ are the submatrices of $\left(B^{(i)}_D \; B^{(j)}_D\right)$ and $\left(B^{(i)}_D \; B^{(l)}_D\right)$, respectively. We also need

$$P_{D,\mathcal{F}^{ij}} := B^T_{D,\mathcal{F}^{ij}} B_{\mathcal{F}^{ij}}, \; P_{D,\mathcal{E}^{ik}} := B^T_{D,\mathcal{E}^{ik}} B_{\mathcal{E}^{ik}}, \text{ and } S_{is} := \begin{pmatrix} S^{(i)} & 0 \\ 0 & S^{(s)} \end{pmatrix}, \; s \in \{j,k\}.$$

We now establish and solve the following generalized eigenvalue problems that can be derived from the localization of the global P_D operator from the standard FETI-DP theory. Therefore, we also introduce the bilinear form $s_{is}(u_{is}, v_{is}) := (S_{is}u_{is}, v_{is})$ with the displacement variables $u_{is}, v_{is} \in W_i \times W_s$, $s \in \{j,k\}$. To describe face and edge eigenvalue problems together, we use $P_{D,\mathcal{Z}^{is}}$ as a generic representation for $P_{D,\mathcal{F}^{ij}}$ and $P_{D,\mathcal{E}^{ik}}$. The local generalized eigenvalue problem on either a face or an edge in variational form writes: Find $w_{is} \in (\ker S_{is})^\perp$ with $\mu_{is} > \text{TOL}$, such that

$$s_{is}(P_{D,\mathcal{Z}^{is}} v_{is}, P_{D,\mathcal{Z}^{is}} w_{is}) = \mu_{is} s_{is}(v_{is}, w_{is}) \quad \forall v_{is} \in (\ker S_{is})^\perp, \; s \in \{j,k\}; \tag{8}$$

cf.[25, Section 5] and[39, Section 5] or[3, Sections 3 and 4] for details.

We assume that we have now computed certain eigenvectors w^r_{is} for $\mu^r_{is} > \text{TOL}$, $s \in \{j,k\}$, $r = 1, 2, \ldots$, on the closure of the faces or edges. During the FETI-DP conjugate gradients iteration, we enforce constraints of the form

$$(w_{is}, P^T_{D,\mathcal{Z}^{is}} S_{is} P_{D,\mathcal{Z}^{is}} w^r_{is}) = 0 \quad \forall w_{is} \in W_i \times W_s, \; s \in \{j,k\} \tag{9}$$

by partial assembly after a transformation of basis on the faces and edges; see[25,38]

Eigenvectors on closed faces. We define $c^r_{ij} = P^T_{D,\mathcal{F}^{ij}} S_{ij} P_{D,\mathcal{F}^{ij}} w^r_{ij}$ and decompose it into edge components $c^r_{ij,\mathcal{E}_m}$, $m = 1, 2, \ldots$, and a component $c^r_{ij,\mathcal{F}}$ on the open face, all extended by zero to the closure (excluding the vertices) of the face; cf.[25,38,39].

Eigenvectors on (closed) edges. The edge eigenvalue problems are crucial to obtain a convergence bound for heterogeneous problems in three dimensions; see[25]. Additionally, to the constraints $c^r_{ij,\mathcal{E}_m}$, $m = 1, 2, \ldots$ from face eigenvalue problems, on edges we enforce the constraints $c^r_{ik} = P^T_{D,\mathcal{E}^{ik}} S_{ik} P_{D,\mathcal{E}^{ik}} w^r_{ik}$ which result from the edge eigenvalue problem between Ω_i and Ω_k.

4.2 | Condition number bound for adaptive FETI-DP

Let $N_\mathcal{F}$ denote the maximum number of faces of a subdomain, $N_\mathcal{E}$ the maximum number of edges of a subdomain, $M_\mathcal{E}$ the maximum multiplicity of an edge, and TOL a given tolerance for solving the local generalized eigenvalue problems. Furthermore, let all vertices be primal. Then, the condition number $\kappa(\widehat{M}^{-1}_T \widehat{F})$ of FETI-DP, where all face and edge eigenvalue problems (8) are solved and where the adaptive constraints (9) are enforced by the generalized transformation-of-basis approach, satisfies (see[38,39])

$$\kappa(\widehat{M}^{-1}_T \widehat{F}) \leq 4 \max\{N_\mathcal{F}, N_\mathcal{E} M_\mathcal{E}\}^2 \text{TOL}. \tag{10}$$

4.3 | Parallel implementation details of the local generalized eigenvalue problems

The FETI-DP algorithms are implemented in C/C++ using PETSc 3.8.0[72,73] and MPI. As a direct solver the PARDISO solver[74] from the Intel MKL is used. The local generalized eigenvalue problems are solved using the SLEPc library 3.8.0[75,76]. The adaptive software is implemented based on the parallel implementation of standard FETI-DP in[62,77,64].

The data and index sets used in the standard FETI-DP implementation are sufficient to construct the generalized eigenvalue problems in the adaptive version. To store the data of each eigenvalue problem, we define a data structure *EigenvalueProblem*. It then consists of an `std::vector` *subdomains*, which stores the two corresponding subdomains in the eigenvalue problem, an `std::vector` *subdomain_neighbors*, which stores additional neighbors for edge eigenvalue problems and is empty for face eigenvalue problems. We also need an `std::vector` *edges* that holds the edges in the eigenvalue problem in a local numbering

`struct` *EigenvalueProblem*

`int` *comm_tag*: individual communication tag

`std::vector` *subdomains*: pair of subdomains in the eigenvalue problem

`std::vector` *subdomain_neighbors*: other adjacent subdomains on the edge (empty for face eigenvalue problems)

`int` *face*: face index (in local list of faces; or -1 for edge eigenvalue problem)

`std::vector` *edges*: edges' indices (in local list of edges)

FIGURE 2 Data structure EigenvalueProblem which holds the elementary information of the eigenvalue problems; figure taken from[39].

and an integer corresponding to the face in the eigenvalue problem (or -1 if the eigenvalue problem is based on an edge); see fig. 2. In order to build these structures, for each face and each edge, we have to know the adjacent subdomain indices. For these latter sets, we make use of the categorization of nodes, which is already necessary in standard FETI-DP to set up the application of the jump operator B. In order to communicate the data, we have to create a consistent ordering of the eigenvalue problems and to set up an individual communication tag.

We now present details of the process to set up and solve the local generalized eigenvalue problems using PETSc, plain MPI, and SLEPc. Before the setup of the local generalized eigenvalue problems is executed, certain information has to be communicated between adjacent pairs of subdomains. The send and receive processes are executed with nonblocking point-to-point communication using `MPI_Isend` (and `MPI_Issend`) and `MPI_IRecv`.

Compared to[39], the send and receive process is split into two stages to avoid unnecessary communication. In the first step, eigenvalue problem information is collected and certain eigenvalue problems are then heuristically discarded as described in *Algorithm Ic*[25,39]; see also sections 4.6 and 5.3. The heuristics uses the diagonal entries of the local stiffness matrices of the interior of each face and each edge. If the ratio of the maximum and the minimum value is small (e.g., smaller than 24) and if the maximum diagonal entry is bounded by Ch_i, where C is smaller than the material discontinuity, then we tag the eigenvalue problem on the face or edge locally with zero. In the subsequent stage, these tags are sent by pointwise communication and collected on each rank. If, for some face or some edge, all collected local tags are zero, then the eigenvalue problem is discarded, otherwise it will be solved.

Then, in the second step, for each adjacent pair of subdomains $\{\Omega_i, \Omega_s\}$ where the eigenvalue problem was not discarded, data is exchanged. In a first, naive approach the data is always sent from the rank with the higher index to the rank with the lower index; cf.[39]. In a more sophisticated approach, the send and solution tasks can be (re)assigned statically and dynamically such that a better load balance is achieved; cf. section 5.2.

Integers and doubles are sent separately. In most cases, two subdomains only share one eigenvalue problem. However, it can occur that two subdomains share more than one edge eigenvalue problem. In these cases, one additional send and receive process is initiated per additional eigenvalue problem between these two subdomains. Otherwise, we cannot extract the correct subset of corresponding rows from the jump operator B.

For now, let us assume that the data is sent in the second stage from the rank of Ω_s to the rank of Ω_i as illustrated in fig. 3. If deluxe-scaling[78,34] is used, for any edge and any adjacent subdomain Ω_k, the Schur complement $S^{(k)}$ from the processes of Ω_k have also to be communicated to the process of Ω_i. This applies likewise to edges in face eigenvalue problems as to edges in edge eigenvalue problems.

As before, we use $P_{D,\mathcal{Z}^{is}}$ as a generic representation for the operator on the closure of the face, $P_{D,\mathcal{F}^{ij}}$, and on the edge, $P_{D,\mathcal{E}^{ik}}$. The SLEPc built-in Krylov-Schur eigensolver is then applied to the local generalized eigenvalue problem

$$Ax = \mu Cx, \tag{11}$$

where $A = \overline{\Pi}_{is}\Pi_{is}P_{D,\mathcal{Z}^{is}}^T S_{is} P_{D,\mathcal{Z}^{is}}\Pi_{is}\overline{\Pi}_{is}$ and $C = \overline{\Pi}_{is}(\Pi_{is}S_{is}\Pi_{is} + \sigma(I-\Pi_{is}))\overline{\Pi}_{is} + \sigma(I-\overline{\Pi}_{is})$; cf. (8). The operators Π_{is} and $\overline{\Pi}_{is}$ are projections such that the local eigenvectors are sought in $\left(\ker S_{is}\right)^{\perp}$; see, e.g.,[3,21,25,39], for more details.

The left hand side operator A is not formed explicitly, only the two local Schur complements are assembled. For each application of A, we need one matrix-vector multiplication with $S^{(i)}$ and $S^{(s)}$ each. Both projections, Π_{is} and $\overline{\Pi}_{is}$ are applied by several vector operations and we also just use two matrix-vector multiplications with the localized B_D- and B-operator.

- local and global numbering for all a priori primal variables Π' on $\partial\Omega_s$
- local (in the space of the degrees of freedom) and global (one-to-one from the corresp. degree of freedom to the space of Lagrange multipliers) for all a priori dual variables Δ' on $\partial\Omega_s$
- a mapping from the local edges to the primal indices on the closure of the edge
- the Schur complement $S^{(s)}$
- the degrees of freedom on $\partial\Omega_s \cap \partial\Omega_D$
  (to detect the common rigid body modes incl. possible hinge modes)
- the coordinates of the nodes on $\partial\Omega_s$
  (to detect the rigid body modes incl. possible hinge modes)
- the edge(s) and the possible face considered in the eigenvalue problem
  (one send per eigenvalue problem between two adjacent subdomains!)

FIGURE 3 Data sent from the rank of Ω_s to the rank of Ω_i in the second stage of point-to-point communication between the subdomains; figure taken from[39].

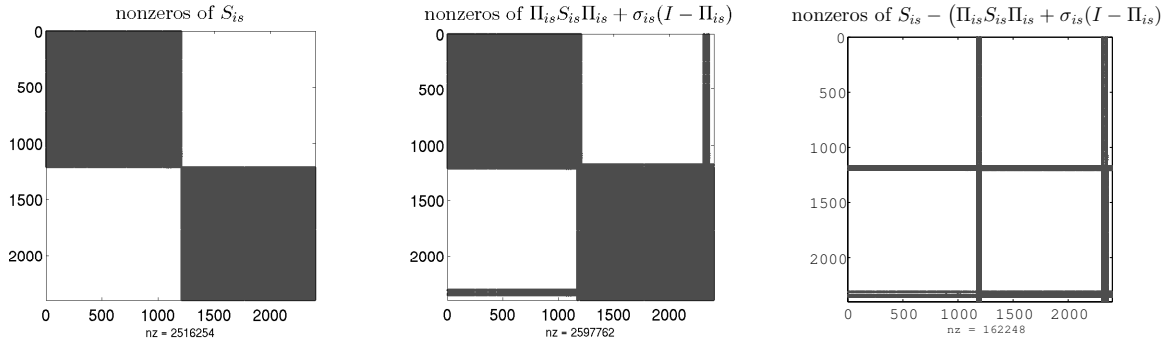


FIGURE 4 Representative nonzero pattern of the matrices S_{is} (left), $\Pi_{is}S_{is}\Pi_{is} + \sigma_{is}(I - \Pi_{is})$ (center), and $S_{is} - \left(\Pi_{is}S_{is}\Pi_{is} + \sigma_{is}(I - \Pi_{is})\right)$ (right). Plot for two randomly chosen subdomains Ω_i and Ω_s and for a composite material with an irregular decomposition of the unit cube, conforming $\mathcal{P}_2$ finite element discretization for $1/h = 6N^{1/3}$; cf.[79,39].

For the right hand side C, after successful reception of the data from Ω_s, we assemble the matrices

$$\widetilde{C} := \Pi_{is}S_{is}\Pi_{is} + \sigma_{is}(I - \Pi_{is}) \tag{12}$$

as a sparse sequential matrix since S_{is} and $\Pi_{is}S_{is}\Pi_{is} + \sigma_{is}(I - \Pi_{is})$ only differ in several rows and columns; see fig. 4. To assemble this matrix, we use the arrays of the dense Schur complements which are in column-major order and exploit the symmetry of the resulting matrix to set the column entries as row entries. If sufficient Dirichlet boundary conditions are present to prevent Ω_i or Ω_s from moving as a rigid body, the corresponding block in $\overline{\Pi}_{is}$ is empty. Otherwise, we make use of the fact that $I - \overline{\Pi}_{is}$ is an orthogonal projection onto the rigid body modes that are continuous on $W_i \times W_s$. Thus, the application of the right hand side operator of the generalized eigenvalue problem can be executed with just one matrix-vector multiplications with $\Pi_{is}S_{is}\Pi_{is} + \sigma_{is}(I - \Pi_{is})$ and several vector-vector or scalar-vector operations.

Inside the SLEPc `EPS` (Eigenvalue Problem Solver) object, the `ST` (Spectral Transformation) object handles the spectral transformations. Since we do not use any shift of the eigenvalues, the generalized problem is internally handled as $C^{-1}Ax = \mu x$ where the solution of the linear system defined by C is executed via the `KSP` object inside the spectral transformations object; see[76] and fig. 5. In our case, we set the preconditioner of the `KSP` to an LU decomposition of the (approximated) right hand side. If the two subdomains have sufficient Dirichlet boundary, the matrix $\Pi_{is}S_{is}\Pi_{is} + \sigma_{is}(I - \Pi_{is})$ is positive definite and an LU decomposition can be computed. If this is not the case, we conduct an LU factorization of $\widetilde{C}_\varepsilon := \left(\Pi_{is}S_{is}\Pi_{is} + \sigma_{is}(I - \Pi_{is})\right) + \varepsilon I$ with a standard choice of $\varepsilon = 1e - 4$ in order to prevent zero pivots, i.e., we have $\widetilde{C} = \left(\widetilde{C}_\varepsilon\right)_{|\varepsilon=0}$. The LU decomposition is performed inplace by Intel MKL PARDISO[74].

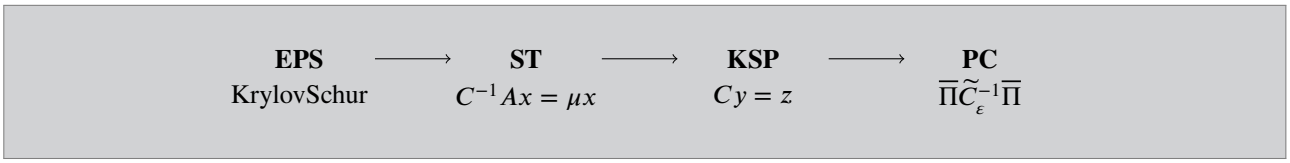


FIGURE 5 Chosen settings for the object structure in the SLEPc `EPS` solver. The `ST` object is used to build a basis for the Krylov decomposition. $\overline{\Pi}$ removes the common rigid body modes, $\widetilde{C}_{\varepsilon}^{-1}$ represents an LU-decomposition of $\widetilde{C}_{\varepsilon} = \left(\Pi_{is} S_{is} \Pi_{is} + \sigma_{is}(I - \Pi_{is})\right) + \varepsilon I$. The solution of the Ritz problem is performed by a direct solver and not illustrated, here.

Inside the Krylov-Schur method, a Krylov decomposition is established before applying the Rayleigh-Ritz procedure; see[80]. As[81] states, in the symmetric case, the Krylov-Schur method is equivalent to the *thick-restart* Lanczos method; see[82]. Consequently, in SLEPc, a Lanczos factorization is computed by using the left hand side A and the `ST`-owned `KSP` object for the right hand side C. The orthogonalization of the basis vectors is carried out by a (modified) Gram-Schmidt algorithm with respect to the inner product $\langle \cdot, C\cdot \rangle$; see[81]. After all eigenvalue problems are solved, we collect the number of constraints and their global indices by one call to `MPI_Allgather` and `MPI_Allgatherv` each. The more expensive communication of the constraints itself is performed by nonblocking point-to-point communication. The constraints are then orthogonalized edge by edge and face by face.

4.4 | A parallel implementation of adaptive FETI-DP using the generalized transformation-of-basis approach

For theoretical considerations of FETI-DP, it is often more convenient to consider the equations based on the Schur complements and the jump operator on the interface, i.e.,

$$F = B_\Gamma \widetilde{S}_{\Gamma\Gamma}^{-1} B_\Gamma^T; \tag{13}$$

cf. section 3. However, for the efficient parallel implementation of FETI-DP, the identity

$$F = B_{B'} K_{B'B'}^{-1} B_{B'}^T + B_{B'} K_{B'B'}^{-1} \widetilde{K}_{\Pi' B}^T \widetilde{S}_{\Pi'\Pi'}^{-1} \widetilde{K}_{\Pi' B'} K_{B'B'}^{-1} B_{B'}^T \tag{14}$$

is used; see, e.g.,[66]. For our adaptive method, the corresponding identity, which we will derive in the following, will be given in eq. (20).

Compared to section 3.2, we use a reordering of the transformation and assembly matrices and introduce additional identity matrices on, e.g., interior variables to fit the dimension. Hence, the equivalent definitions can be carried over into one another easily.

Let us recall the index sets $B' = (I, \Delta')$ and $\Delta' = (\Delta, \Pi)$, where Δ denotes the a posteriori dual and Π the a posteriori primal variables. Using the representation $B' = (I, \Delta')$, the system matrix of the FETI-DP master system (see (1)) writes

$$\begin{pmatrix} K_{II} & K_{\Delta' I}^T & \widetilde{K}_{\Pi' I}^T & 0 \\ K_{\Delta' I} & K_{\Delta'\Delta'} & \widetilde{K}_{\Pi'\Delta'}^T & B_{\Delta'}^T \\ \widetilde{K}_{\Pi' I} & \widetilde{K}_{\Pi'\Delta'} & \widetilde{K}_{\Pi'\Pi'} & 0 \\ 0 & B_{\Delta'} & 0 & 0 \end{pmatrix}. \tag{15}$$

Note that a further subdivision of $K_{\Delta'\Delta'}$ does not make much sense since the constraints in the transformation matrix are columns defined on $\Delta' = (\Delta, \Pi)$.

By incorporating the local components $T_{\mathcal{Z},\Delta}$ and $T_{\mathcal{Z},\Pi}$ for every face or edge $\mathcal{Z}$ into T_Δ and T_Π (cf. (4)), we obtain the global transformation matrix with an identity on the interior and a priori primal variables as

$$T = \begin{pmatrix} I_I & 0 & 0 & 0 \\ 0 & T_\Delta & T_\Pi & 0 \\ 0 & 0 & 0 & I_{\Pi'} \end{pmatrix}. \tag{16}$$

The global restriction operator R and the corresponding second level assembly operator R^T are given by

$$R = \begin{pmatrix} I_I & 0 & 0 & 0 \\ 0 & I_\Delta & 0 & 0 \\ 0 & 0 & R_\Pi & 0 \\ 0 & 0 & 0 & I_{\Pi'} \end{pmatrix}, \tag{17}$$

where R_Π^T assembles the a posteriori degrees of freedom. We also use the multiplicity-scaled variant of R_Π^T by $R_{\Pi,\mu}^T$. The operator R_μ is obtained from R by replacing R_Π with $R_{\Pi,\mu}$; cf. section 3.2.

We now want to derive the system matrix of the adaptive FETI-DP master system using the generalized transformation-of-basis approach. Therefore, the leading 3×3 block of (15) has to be transformed and assembled, i.e., $R^T T^T$ has to be applied from the left and TR has to be applied from the right. Correspondingly, we also have to adapt the application of the jump operator B to be in the correct basis; cf. the definition in (6). Note that for several submatrices, the multiplication with the submatrices of T and the application of R is trivial and has not be carried out.

With the transformation, the assembly, and the restriction applied to (15) as well as the adaptation of the jump operator, the system matrix of the adaptive FETI-DP master system is

$$\left(\begin{array}{cc|cc|c} K_{II} & K_{\Delta' I}^T T_\Delta & K_{\Delta' I}^T T_\Pi R_\Pi & \widetilde{K}_{\Pi' I}^T & 0 \\ T_\Delta^T K_{\Delta' I} & T_\Delta^T K_{\Delta'\Delta'} T_\Delta & T_\Delta^T K_{\Delta'\Delta'} T_\Pi R_\Pi & T_\Delta^T \widetilde{K}_{\Pi'\Delta'}^T & T_\Delta^T B_{\Delta'}^T \\ \hline R_\Pi^T T_\Pi^T K_{\Delta' I} & R_\Pi^T T_\Pi^T K_{\Delta'\Delta'} T_\Delta & R_\Pi^T T_\Pi^T K_{\Delta'\Delta'} T_\Pi R_\Pi & R_\Pi^T T_\Pi^T \widetilde{K}_{\Pi'\Delta'}^T & R_\Pi^T T_\Pi^T B_{\Delta'}^T \\ \widetilde{K}_{\Pi' I} & \widetilde{K}_{\Pi'\Delta'} T_\Delta & \widetilde{K}_{\Pi'\Delta'} T_\Pi R_\Pi & \widetilde{K}_{\Pi'\Pi'} & 0 \\ \hline 0 & B_{\Delta'} T_\Delta & B_{\Delta'} T_\Pi R_\Pi & 0 & 0 \end{array}\right) \tag{18}$$

Equivalently, we can write

$$\left(\begin{array}{c|c|c} \widehat{K}_{BB} & \widehat{\widetilde{K}}_{\hat{\Pi}B}^T & \widehat{B}_B^T \\ \hline \widehat{\widetilde{K}}_{\hat{\Pi}B} & \widehat{\widetilde{K}}_{\hat{\Pi}\hat{\Pi}} & \widehat{B}_{\hat{\Pi}}^T \\ \hline \widehat{B}_B & \widehat{B}_{\hat{\Pi}} & 0 \end{array}\right) \tag{19}$$

if grouping interior and a posteriori dual variables to the subindex $B := (I, \Delta)$ and a priori and a posteriori primal variables to the subindex $\hat{\Pi} := (\Pi, \Pi')$, i.e., merging the denoted submatrices inside the horizontal and vertical delimiters to the new notations. Note that $\widehat{B}_\Pi$ is only nonzero on the a posteriori primal degrees of freedom and enables the necessary interaction between these and the a posteriori dual variables as explained in detail in [67,38,39].

Thus, after solving the local generalized eigenvalue problems and distributing the computed constraints, the second step of the adaptive algorithm consists of establishing the new operators and matrices. The assembly in the primal variables is realized by setting up two `VecScatters`. The first scatter is necessary to assemble in all primal variables $\hat{\Pi}$ and, for the preconditioner, the second scatter is needed to assemble only the a posteriori primal variables Π; cf. the subsequent paragraphs for more details.

In a next step, the transformed matrices $T_\Delta^T K_{\Delta' I}$, $T_\Pi^T K_{\Delta' I}$, $T_\Delta^T K_{\Delta'\Delta'} T_\Delta$, $T_\Delta^T K_{\Delta'\Delta'} T_\Pi$, $T_\Pi^T K_{\Delta'\Delta'} T_\Pi$, $K_{\Pi'\Delta'} T_\Delta$, $K_{\Pi'\Delta'} T_\Pi$ are obtained block by block, where the blocks are local submatrices of $T^{(i)T} K^{(i)} T^{(i)}$. Here, $T^{(i)}$ is obtained by adding the identity on the interior variables to $T_\Gamma^{(i)}$, $i = 1, \ldots, N$. Note that $\widetilde{K}_{\Pi'\Delta'} T_\Delta$ and $\widetilde{K}_{\Pi'\Delta'} T_\Pi$ can be obtained from $K_{\Pi'\Delta'} T_\Delta$ and $K_{\Pi'\Delta'} T_\Pi$ since the order of the first level scatter (defined by $R_{\Pi'}$ in section 3) and the transformation T can be inversed. To be more precise, the first level scatter and the transformation act on disjoint variable sets.

By Gaussian elimination, we then obtain from (19)

$$\widehat{F} := \widehat{B}_B \widehat{K}_{BB}^{-1} \widehat{B}_B^T - \big(\widehat{B}_{\hat{\Pi}} - \widehat{B}_B \widehat{K}_{BB}^{-1} \widehat{\widetilde{K}}_{\hat{\Pi}B}^T\big) \widehat{\widetilde{S}}_{\hat{\Pi}\hat{\Pi}}^{-1} \big(\widehat{B}_{\hat{\Pi}}^T - \widehat{\widetilde{K}}_{\hat{\Pi}B} \widehat{K}_{BB}^{-1} \widehat{B}_B^T\big), \tag{20}$$

where $\widehat{\widetilde{S}}_{\hat{\Pi}\hat{\Pi}} := \widehat{\widetilde{K}}_{\hat{\Pi}\hat{\Pi}} - \widehat{\widetilde{K}}_{\hat{\Pi}B} \widehat{K}_{BB}^{-1} \widehat{\widetilde{K}}_{\hat{\Pi}B}^T$. In (20), the first summand $\widehat{F}$, i.e., $\widehat{B}_B \widehat{K}_{BB}^{-1} \widehat{B}_B^T$, remains perfectly parallelizable. Furthermore, as in standard FETI-DP, we need one coarse solve per iteration.

The scatter to realize the application of the jump operator does not need any new setup. Only the application has to be changed by applying the local (transposed) transformation matrices before and after the scatter, respectively. By using the scatter structure of $B_{\Delta'}$ and $B_{\Delta'}^T$ already established a priori, a posteriori dual and a posteriori primal variables are processed simultaneously. Thus, the application of $\widehat{F}$ is more involved. For instance, for the solution of the linear system

$$\widehat{K}_{BB} x = \widehat{B}_B^T \widehat{\lambda}$$

the values $\widehat{B}_B^T\widehat{\lambda} = T_\Delta^T B_{\Delta'}^T\widehat{\lambda}$ are extracted from $T_{\Delta'}^T B_{\Delta'}^T\widehat{\lambda}$ (note the additional prime!). The complementary (and assembled) part, i.e., $\widehat{B}_{\widehat{\Pi}}^T\widehat{\lambda} = R_\Pi^T T_\Pi^T B_{\Delta'}^T\widehat{\lambda}$ is added afterwards to the vector $-\widehat{\widetilde{K}}_{\widehat{\Pi}B}x$. The assembly in the a posteriori primal variables is naturally performed by the scatter operation. Then, the coarse solve can be executed. Before the application of $B_{\Delta'}$, the values in the a posteriori dual and a posteriori primal variables have to be collected.

Note that $\Delta' \cap \widehat{\Pi} = \Pi$. As a consequence, in the iterative scheme, we always work with two vectors $u_{\Delta'}$ and $u_{\widehat{\Pi}}$. The values in the a posteriori primal variables have to be transfered from one to the other, depending on the next matrix-vector multiplication or `KSPSolve` process to execute. This is different from the standard approach.

The Dirichlet preconditioner in the adaptive context writes

$$\widehat{M}_T^{-1} := B_{D,\Delta'} T_{\Delta'} R_{\Delta',\mu} \underbrace{R_{\Delta'}^T T_{\Delta'}^T S_{\Delta'\Delta'} T_{\Delta'} R_{\Delta'}}_{=\widehat{\widetilde{S}}_{\Delta'\Delta'}} R_{\Delta',\mu}^T T_{\Delta'}^T B_{D,\Delta'}^T, \tag{21}$$

where $T_{\Delta'} := (T_\Delta, T_\Pi)$ and $R_{\Delta'}$ and $R_{\Delta',\mu}$ assemble in the a posteriori primal variables (the latter with multiplicity scaling in Π) and do not change Δ variables. The preconditioner then is the sum of local operators with communication between neighboring subdomains via B, as before, and minor additional communication via the scatter $R_\Pi R_{\Pi,\mu}^T$ (inside RR_μ^T). This also differs from the standard approach. For the definition of $\widehat{M}_T^{-1}$, see also (6), and for the standard Dirichlet preconditioner M_D^{-1}, see (3). In $R_{\Pi,\mu}$, the scaling is multiplicity-scaling independent of the scaling chosen in B_D; cf. section 3.2.

4.5 | Computation of the solution in the displacement variables

Given the appropriately transformed right hand side $\widehat{f} = (\widehat{f}_B^T,\ \widehat{\widetilde{f}}_{\widehat{\Pi}}^T,\ 0)^T$, assembled in the primal variables, the master system of adaptive FETI-DP reads

$$\begin{pmatrix} \widehat{K}_{BB} & \widehat{\widetilde{K}}_{\widehat{\Pi}B}^T & \widehat{B}_B^T \\ \widehat{\widetilde{K}}_{\widehat{\Pi}B} & \widehat{\widetilde{K}}_{\widehat{\Pi}\widehat{\Pi}} & \widehat{B}_{\widehat{\Pi}}^T \\ \widehat{B}_B & \widehat{B}_{\widehat{\Pi}} & 0 \end{pmatrix} \begin{pmatrix} \widehat{u}_B \\ \widehat{\widetilde{u}}_{\widehat{\Pi}} \\ \widehat{\lambda} \end{pmatrix} = \begin{pmatrix} \widehat{f}_B \\ \widehat{\widetilde{f}}_{\widehat{\Pi}} \\ 0 \end{pmatrix}; \tag{22}$$

cf. (19). When computing the solution in the displacement variables, we have to keep in mind that $\widehat{B}_{\widehat{\Pi}} \neq 0$. In contrast to standard FETI-DP, the Gaussian elimination then yields

$$\widehat{\widetilde{u}}_{\widehat{\Pi}} = \widehat{\widetilde{S}}_{\widehat{\Pi}\widehat{\Pi}}^{-1}\big(\widehat{\widetilde{f}}_{\widehat{\Pi}} - \widehat{\widetilde{K}}_{\widehat{\Pi}B}\widehat{K}_{BB}^{-1}\widehat{f}_B - \widehat{B}_{\widehat{\Pi}}^T\widehat{\lambda} + \widehat{\widetilde{K}}_{\widehat{\Pi}B}\widehat{K}_{BB}^{-1}\widehat{B}_B^T\widehat{\lambda}\big), \tag{23}$$

$$\widehat{u}_B = \widehat{K}_{BB}^{-1}\big(\widehat{f}_B - \widehat{B}_B^T\widehat{\lambda} - \widehat{\widetilde{K}}_{\widehat{\Pi}B}^T\widehat{\widetilde{u}}_{\widehat{\Pi}}\big). \tag{24}$$

Thus, the term $-\widehat{\widetilde{S}}_{\widehat{\Pi}\widehat{\Pi}}^{-1}\widehat{B}_{\widehat{\Pi}}^T\widehat{\lambda}$ adds to the solution in the primal variables.

4.6 | Heuristic optimization

Considering the theory developed and used in [25,67,38], to obtain the condition number bound of section 4.2 eigenvalue problems have to be defined and solved on all edges and all faces of the partitioning. However, numerous numerical examples in [25,83,67,38,39] have shown that a heuristic approach to discard eigenvalue problems based on neighborhood considerations does not increase the observed condition numbers: In [25], we introduced *Algorithm Ic*, to discard edge eigenvalue problems and edge constraints from face eigenvalue problems if no large coefficients were detected in the neighborhood of the edge. The detection of the jumps was conducted by considering the coefficient function. As in [39], in this paper, we do not assume that the coefficient function is known explicitely. Instead the decision on whether to keep or discard edge eigenvalue problems is based on the entries of the stiffness matrix; see also the last paragraph in section 5.1. Additionally, we will extend the idea of discarding edge eigenvalue problems to face eigenvalue problems; see section 5.3. The modified *Algorithm Ic*, our new standard choice, will simply be denoted *adaptive FETI-DP* not introducing any new name.

5 | COMPUTATIONAL AND ALGORITHMIC IMPROVEMENTS COMPARED TO PREVIOUS WORKS

Some preliminary results using an early version of the parallel implementation of adaptive FETI-DP have already been presented in[38,39] without describing the implementation in detail. However, obvious and more elaborated improvements have been implemented only recently. In section 5.1, we will briefly present several computational and (minor) algorithmic improvements to speed up the algorithm and to make it, in our experience, more stable. In section 5.2, we will present our ideas on how to improve the load balance of the algorithm by using static and asynchronous dynamic processes to better distribute the local eigenvalue problems. Our ideas to discard certain edge eigenvalue problems (see[25]) are now also transfered to face eigenvalue problems; see section 5.3 for more details. Finally, we also present ideas on how to handle heterogeneous problems with continuous spectra without blowing up the adaptive coarse space; see section 5.4.

5.1 | Computational and minor algorithmic improvements

As already stated in[39], the application of the right hand side operator of the generalized eigenvalue problem can be executed with just one matrix-vector multiplication with $\widetilde{C} = \left(\Pi_{is} S_{is} \Pi_{is} + \sigma_{is}(I - \Pi_{is})\right)$ and several vector-vector or scalar-vector operations. In the previous implementation, however, two local applications were used if the two subdomains did not have sufficient Dirichlet boundary conditions. By improving the application, the computational work and time could be reduced.

Additionally, for the results presented in[38,39], a modified Gram-Schmidt algorithm with a very coarse convergence criterion was used. In rare cases, it was observed that this could lead to instabilities of the overall algorithm if demanding convergence criteria (e.g., a relative residual reduction of 1e-10) were chosen. Therefore, we replaced the Gram-Schmidt orthogonalization by two steps of a more stable Householder orthogonalization. For every face and every edge, generically denoted by $\mathcal{Z}$, the first step consists of a rather inaccurate orthogonalization of the constraints in $T_{\mathcal{Z},\Pi}$, i.e., by using a drop tolerance of 1e-1. This is done since the constraints are only needed approximately, as our results in[25,38] motivated. Thus, from a set of constraints which are very similar only a subset might be needed to obtain fast convergence and the coarse space size can be reduced. In a second step, we use a more accurate drop tolerance of 1e-9 to orthogonalize the identity against $T_{\mathcal{Z},\Pi}$ to obtain the orthogonal space represented by $T_{\mathcal{Z},\Delta}$ on every face and every edge $\mathcal{Z}$.

In[39], a large paragraph (i.e., Section 7.3.3) was devoted to the parallel implementation of our algorithm which discards eigenvalue problems based on the underlying coefficient or the entries of the local stiffness matrices. The idea was to reduce the communication time by only using information which was already present. However, it was found that the communication time of the information, which was avoided to communicate in[39], is negligible. Besides, the decision in[39] on how to discard eigenvalue problems was much more conservative since the corresponding bounds used scaling information whereas the entries of the stiffness matrices yield sharper bounds. In the consequence, the computational overhead of the adaptive method can be reduced since more eigenvalue problems can be discarded. Finally, for a diagonal entry $k_x^{(i)}$ of the stiffness matrix $K^{(i)}$, the bound $k_x^{(i)} \geq 1000h^3$ which was used to decide that a corresponding eigenvalue problem will be kept, has been replaced by the sharper bound $k_x^{(i)} \geq C(\mathrm{E}_1, \mathrm{E}_2, \nu_1, \nu_2)h$, where $C(\mathrm{E}_1, \mathrm{E}_2, \nu_1, \nu_2)$ is a constant depending on the coefficient discontinuity. For $\mathrm{E}_1 = 1$ and $\mathrm{E}_2 = 1e6$, it can be set to, e.g., 1000. Clearly, the corrected bound is more adequate for making decisions based on entries of the stiffness matrix.

5.2 | Lightweight dynamic load balancing

In[39], we have seen that the naive approach to always solve the eigenvalue problem related to the two subdomains Ω_i and Ω_j on the rank with the lower index, does not lead to a good load balance. We have therefore implemented two levels of load balancing. On both levels, the eigenvalue problems are solved either on the rank corresponding to Ω_i or corresponding to Ω_j, i.e., the eigenvalue problems are never migrated to a rank unrelated to Ω_i or Ω_j. This avoids additional overhead from communicating matrices.

We make use of an additional orchestrating rank which runs on a core, where no subdomains or eigenvalue problems are treated; it only serves to observe the solution process. The communication is lightweight as only a few integers are sent, which represents the solution status of the corresponding eigenvalue problem; no geometry information is sent. The send and receive processes are asynchronous.

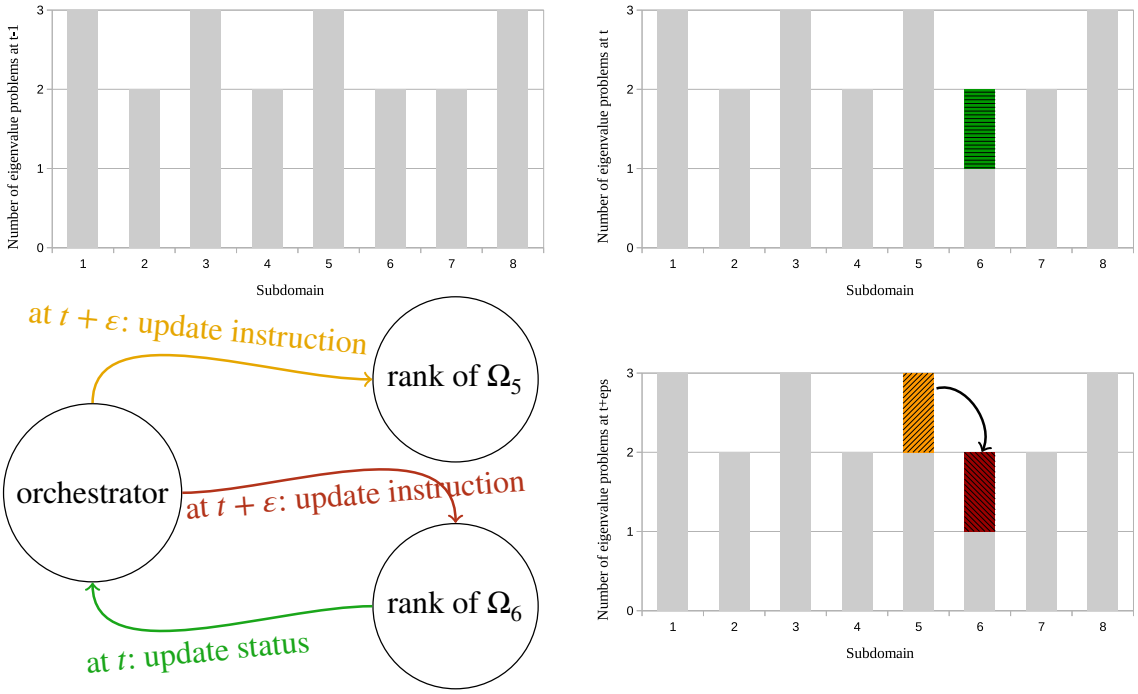


FIGURE 6 Minimal example of the asynchronous and dynamic load balancing. Number of eigenvalue problems per subdomain at computational time markers $t-1$, t, and $t+\varepsilon$ (top left and right, bottom right). The remaining eigenvalue problems are shown in blue, eigenvalue problems solved between two time markers are colored green, yellow beams represent eigenvalue problems that are turned over to another rank, such obtained eigenvalue problems are colored red. Communication between time markers t and $t+\varepsilon$ between the solution ranks of Ω_5 and Ω_6 and the orchestrator (bottom left).

The first, static level of load balancing consists of several steps. In a first step, the eigenvalue problems are assigned by a pseudo-random, deterministic pattern to the subdomains, either Ω_i or Ω_j, by considering the modulo-2-check of the sum $i+j$. Then, the number of eigenvalue problems per subdomain are collected once globally. We then iterate several times on the resulting vector to reassign the eigenvalue problems to either Ω_i and Ω_j. This, however, does not include a consideration of the size or difficulty of the single eigenvalue problems.

In order to take into account the different solution times needed for the eigenvalue problems which can differ in size and in difficulty, we make use of a dynamic, second level of load balancing. The orchestrating rank receives information on the completion of the local eigenvalue problems by asynchronous, nonblocking communication. If the orchestrator detects that certain ranks move forward faster in their process, it reassigns eigenvalue problems from ranks with large numbers of eigenvalue problems, to ranks with lower numbers of remaining eigenvalue problems; see fig. 6 for a representative action of the orchestrating rank in the solution process. In rare cases, an eigenvalue problem solution process may complete while it is migrated from one rank to the other by the orchestrator. In this case, the problem is solved redundantly, i.e., on both ranks. However, this does not appear often. Therefore, we still obtain a significant speedup of the overall algorithm. Numerical results documenting the speedup of our adaptive FETI-DP using load balancing are available in section 6.1. In this paper, this option is always used.

In [11], a different approach was taken. There, the eigenvalue problems are distributed to all ranks, and a rank can also handle eigenvalue problems not corresponding to a subdomain on the same rank. However, to avoid having to communicate the matrices, the iterates are communicated, i.e., the matrix-vector products are still carried out by the ranks of the corresponding subdomains. This is opposed to our approach. Moreover, in [11], synchronization takes place after each set of parallel eigenvalue problems, i.e., all ranks have to wait for the slowest eigenvalue problem. In our implementation, on the other hand, the barrier is encountered only after all eigenvalue problems have been computed.

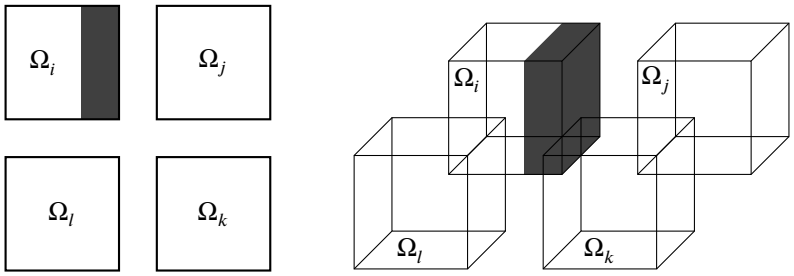


FIGURE 7 Minimal example of four subdomains sharing an edge where a face eigenvalue problem will be discarded although a large cofficient is on hand; cross section from above (left) and three-dimensional representation (right). A large coefficient (darkgray) exists in a part of Ω_i, leading to face eigenvalue problems between Ω_i and Ω_j and Ω_l, respectively. The face eigenvalue problem between Ω_i and Ω_k is discarded since large coefficients only appear on a zero set considering the two-dimensional Lebesque measure (i.e., we do not have jumps across the interior of the face).

5.3 | Discarding face eigenvalue problems

For adaptive FETI-DP, the idea to discard edge eigenvalue problems is now also transferred to face eigenvalue problems. Though, the face eigenvalue problems are defined on the closure of the face, only the entries of the stiffness matrix corresponding to the interior of the face are checked. The reasoning to do so is that a large coefficient that only intersects the border of the face necessarily intersects the interior of another face (there, the corresponding eigenvalue will be kept), otherwise the support of the stiff inclusion would be a zero set. Our numerical results support our assertion that those jumps only appearing on the border of the face are well handled by other face or edge eigenvalue problems. In fig. 7, we have given a minimal example.

5.4 | Flat spectrum heuristics

As we have already observed in [38, Fig. 3-4] and [39, Fig. 6.5 and 6.7], the spectrum of the local generalized eigenvalue problems does not necessarily have a gap. Then, if the user-defined a priori tolerance is too small, the coarse space will be unnecessarily large; if, on the other hand, the tolerance is too large, many problematic eigenmodes can remain in the solution space for our iterative solver. Consequently, there might be no optimal choice for an appropriate user-defined tolerance for the solution of the eigenvalue problems. In this case, we propose to stop growing the coarse space once the eigenvalues are below a certain tolerance TOL_H and, additionally, the spectrum is detected to become flat (see [38, Fig. 3-4]). In this case, increasing the coarse space further would not reduce the condition number significantly.

6 | NUMERICAL RESULTS

We present numerical results for our parallel implementation of adaptive FETI-DP applied to compressible linear elasticity. For simplicity, we always assume the parameters E and ν to be constant on each finite element.

We reduce the number of eigenvalue problems and constraints according to the strategy described in section 4.6. For a more detailed description, we refer to [25,39, Algorithm Ic] and section 5.3. We always use stiffness-scaling.

For all algorithms, the constraint vectors are orthogonalized blockwise (i.e., edge by edge and face by face) by a Householder orthogonalization; see section 5.1.

As in previous works, for short edges consisting of only one dual node, we convert the single dual node into a primal node, to make the corresponding edge eigenvalue problem superfluous.

In the experiments, regular as well as irregular decompositions are tested. The irregular decompositions are set up by the METIS graph partitioner [84] using the options `-ncommon=3` and `-contig` to avoid noncontiguous subdomains as well as additional hinge modes inside single subdomains. The regular decompositions are directly performed by our C/C++ software, the irregular decompositions are imported after being exported from our MATLAB software. In these cases, the corresponding total *time*, which is given in the tables, does not include the basic setup of the geometry.

The local generalized eigenvalue problems are set up and solved using PETSc[72,73] and SLEPc[75,76]; see section 4.3. We have chosen the modified Gram-Schmidt algorithm inside the Krylov-Schur method to orthogonalize the basis vectors. Due to highly ill-conditioned local right hand sides C in eq. (11), the local iterative solver of the preconditioned `KSP` object (see fig. 5) might not converge or may even break down if large coefficient jumps of 1e+6, irregular decompositions, the preset divergence tolerance, and only modest relative convergence tolerances for the `KSP` object are used. The breakdown mostly occurs due to large jumps in the residual. We have already documented a similar behavior for LOBPCG in[39, Chap. 6]. Note that the condition number of the local right hand sides C can exceed the condition number of the global system matrix by several orders of magnitude.

In[39], we provided several workarounds for the case that the iterative KSP solver breaks down. In the current implementation, however, we practically deactivate the breakdown test of the `KSP` object via `dtol` by setting it to, e.g., 1e12. We refer to[85, Fig. 5.1] and[39, Fig. 6.6], where large jumps in the residual of the first steps of the iterative solver have already been observed, even if an adequate preconditioner is used. We then only demand a relative residual reduction of a factor of 1e-2 for the `KSP` object. If the internal solver still breaks down, we directly make half of the face or edge primal. This rarely occurs. Note that the interal `KSP` solver is only used to build the Lanczos factorization. Bad approximations to eigenvectors of large eigenvalues, i.e., approximations that do not point into the direction of the eigenvector, are removed after the Ritz-values have been computed by a direct solver and when the Krylov decomposition is truncated; cf.[80,81]. The promising results of[25,38,39], where LOBPCG (see[86]) was used, lead us to a required relative residual reduction of a factor of 1e-5 or a maximum of 5 iterations for the `EPS` Krylov-Schur solver.

We use a Krylov-Schur eigensolver with a block size between 1 and 10. If the smallest eigenvalue in the computed block of (approximate) eigenvalues is still larger than our choice of TOL, we use the SLEPc functionality `EPSSetDeflationSpace` to compute another block of eigenvalues and -vectors in a deflated search space. As we have documented in[87], a single vector iteration (i.e. block size 1) and multiple restarts of the solver might be the fastest version, load balancing and scaling results are qualitatively comparably for the different block sizes.

The stopping criterion for PCG is a relative reduction of the preconditioned residual by a factor of 1e-8. The flat spectrum heuristics of section 5.4 is not used.

For our numerical experiments, we only use one tolerance TOL with $\text{TOL} = 50\log(H/h)$ for regular decompositions and $\text{TOL} = 50\log\left(N/n_i\right)^{1/3}$ for irregular decompositions. Here, n_i denotes the number of local nodes on Ω_i. The (lower) tolerance is therefore adapted to the estimate of edge terms in standard FETI-DP; see, e.g.,[66]. Similar adaptations of the tolerance were already used for another adaptive coarse space; see[35]. See also[37], for detailed study of the influences of the a priori tolerances on the coarse space dimension for another adaptive approach.

For all experiments with a regular decomposition, we enforce homogeneous Dirichlet boundary conditions on the complete boundary. For irregular decompositions and composite material A, we enforce homogeneous Dirichlet boundary conditions on the face with $x = 0$ and zero Neumann boundary conditions elsewhere. For the hemisphere considered in section 6.5, we enforce homogeneous Dirichlet boundary conditions on the upper part, satisfying $z = 0$, and, on the remaining part of the boundary, we enforce zero Neumann boundary conditions. We always apply the volume force $f := (0.1, 0.1, 0.1)^T$. Except for the hemisphere, we use a structured fine mesh consisting of cubes. The fine cubes are each decomposed into five (irregular decomp.) and six (reg. decomp.) tetrahedra, respectively.

We always consider a compressible material with $\nu = 0.3$ for the entire computational domain. All computations are conducted with one subdomain per core on the supercomputer magnitUDE at the Center for Computational Sciences and Simulation (CCSS) of the University of Duisburg-Essen. The supercomputer magnitUDE has 14 976 cores (Xeon E5-2650v4 12C 2.2GHz; 624 nodes with 24 cores each). All computing nodes hold, at least, 64 GB of main memory. Intel compilers v17.0.1 with the corresponding MKL are used.

In the tables, κ denotes the estimated condition number of the preconditioned FETI-DP operator. The condition number estimates are obtained from the Krylov scheme. By N we denote the number of subdomains. Additionally, we report the number of iterations of the PCG algorithm by *its*, by $|\Pi'|$ the size of the a priori and by $|\Pi|$ the size of the adaptive coarse space, respectively. All a posteriori constraints are implemented using the generalized transformation-of-basis approach. We also list the number of nonzeros in the final coarse matrix as *nnz (coarse)*. For the regular decompositions, we report H/h. For irregular decompositions, we only list $1/h$ and N in order to measure the mean size of the local problems. We also report the number of the global degrees of freedom as *#dofs* and the total *time* needed by the algorithm. A more detailed breakdown of the timings can be found in some corresponding diagrams.

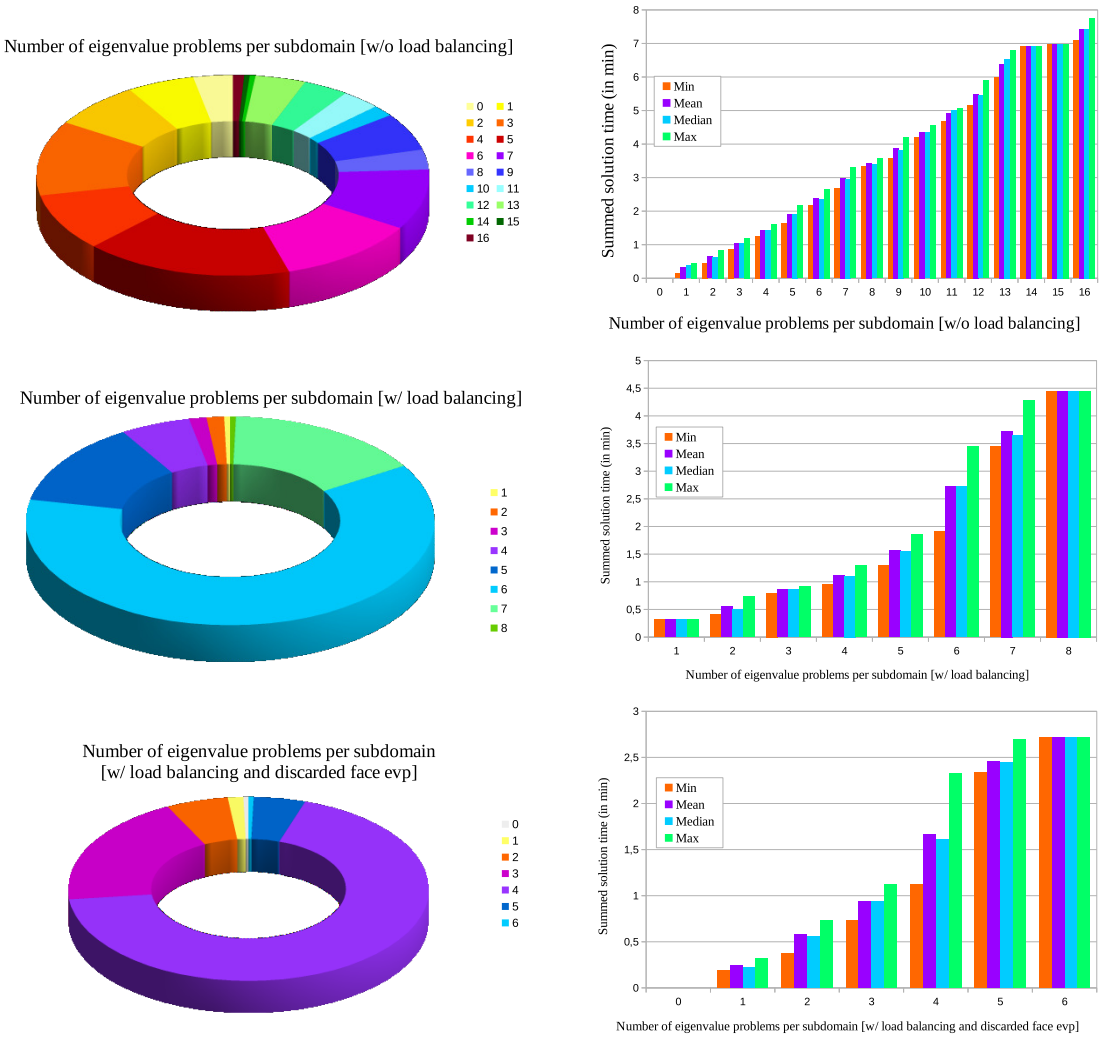


FIGURE 8 Number of subdomains (left) and summed solution time (right) per number of local eigenvalue problems on one subdomain for adaptive FETI-DP with 216 subdomains on 216 (top) and 217 (2nd to top and 2nd to bottom) cores, respectively. Composite material A and an irregular decomposition of the unit cube; cf. fig. 10. **Top:** without load balancing; **center:** with dynamic load balancing; **bottom:** with load balancing *and* discarding certain face eigenvalue problems.

6.1 | Performance of adaptive FETI-DP using load balancing

In this section, we consider a composite material denoted *composite material A*; see fig. 10. For this material and N subdomains, $N^{2/3}$ many beams with $\mathrm{E}_2 = 1e+6$ that run in a straight line from the face with $x = 0$ to the face with $x = 1$. The beams cover 1/9th of the cross-section of the material and are surrounded by a soft matrix material with $\mathrm{E}_1 = 1$. Here, we only consider $N = 216$ subdomains and focus on the effect of static and asynchronous dynamic load balancing; see section 5.2. For a detailed weak scaling analysis of adaptive FETI-DP with load balancing (and discarded face eigenvalue problems), i.e., our newly proposed algorithm, see section 6.3.

In figs. 8 and 9. we recognize that the maximum number of eigenvalue problems per subdomain can be reduced from 16 to 8. We also see that about 60% of the subdomains have exactly six eigenvalue problems to consider. The load balancing of the eigenvalue problems can thus reduce the maximum time spent in the solution of the eigenvalue problems by a factor of about 1.75 (from 465s to 266s), the computation of the eigenvectors can be reduced and the setup of the local preconditioners can be reduced by a factor of about 2 (i.e., from 81s to 43s and from 30s to 16s). As already observed in [39], the solution of the eigenvalue

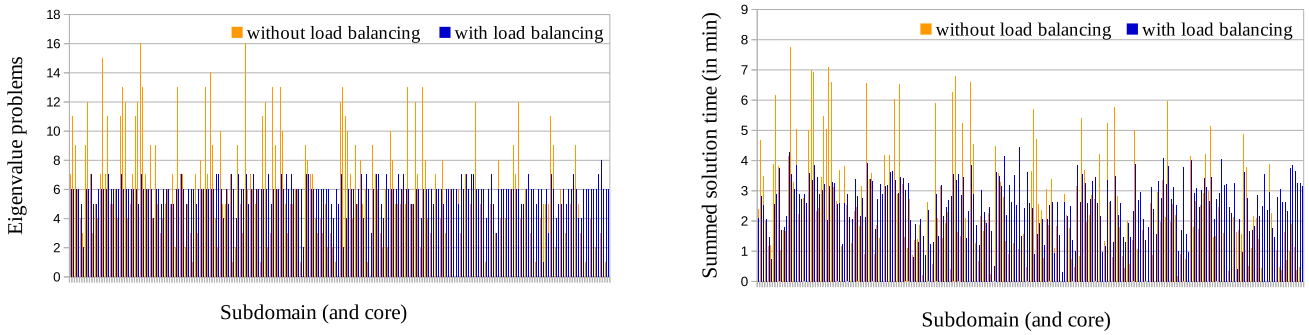


FIGURE 9 Total number of eigenvalue problems (left) and eigenvalue problem solution time (right) per core without and with load balancing. Composite material A and an irregular decomposition of the unit cube; cf. fig. 10.

problems is the most costly part of our algorithm. Consequently, also the total computation time can be reduced by a third (from 630s to 430s).

6.2 | Performance of adaptive FETI-DP discarding certain face eigenvalue problems

In this section, we consider a composite material denoted *composite material A* as in section 6.1. Here, we only consider $N = 216$ subdomains and focus on the effect of discarding certain face eigenvalue problems; see section 5.3. For a detailed weak scaling analysis of adaptive FETI-DP with discarded face eigenvalue problems (and load balancing), i.e., our new proposed algorithm, see section 6.3.

In fig. 8, we now see that the maximum number of eigenvalue problems per subdomains can again be reduced. Now from 8 to 6. Additionally, we also see that about 70% of the subdomains only have exactly four eigenvalue problems to consider (before: 60% with 6). The time spent in the solution phase of the local eigenvalue problems can be further reduced from 266s to 163s. Also the time to compute the eigenvectors from the Ritz-vectors can be reduced from 43s to 23s and the setup of the local preconditioners is reduced from 16s to 11s. In total, the runtime can be reduced from 430s to 251s, i.e., by a factor of about 1.7.

6.3 | Weak parallel scalability for adaptive FETI-DP on irregular decompositions

Again, we consider our parallel implementation of adaptive FETI-DP with two levels of load balancing (static and asynchronous dynamic) and the elimination of certain edge as well as face eigenvalue problems; see section 5.2 and section 5.3. As before, we consider composite material A.

In table 1, we present the results of weak parallel scalability from $N = 216$ to $N = 1000$ subdomains with $N + 1$ cores. fig. 10 gives detailed information on the weak scaling for the most expensive stages of the adaptive algorithm. The adaptive reassembling and the conjugate gradients do not scale perfectly since the size of the coarse space becomes quite large in absolute numbers. However, in relative numbers only about 1.4% of the degrees of freedom are coarse degrees of freedom. For 1000 subdomains, 98 of 102 seconds spent in the conjugate gradients scheme result from the forward backward solve of the coarse problem. The application of the adaptive preconditioner needs less than 1 second. This confirms that the additional nearest neighbor communication in the adaptive preconditioner (see section 4.4) is negligible in the total runtime. Eventually, note that results for 216 and 512 subdomains were already presented in[39], however, using the multiple improvements mentioned in section 5, the runs have been sped up by a factor of about 4.6.

6.4 | Weak parallel scalability for adaptive FETI-DP on regular decompositions

In the previous section, we have presented weak scaling results for our parallel implementation and irregular decompositions of the computational domain. Since our a priori coarse space is already quite large for irregular decompositions (which limits our simulations with exact coarse solves), we now also consider regular decompositions. The material considered here is again a composite material where material discontinuities appear on edges and faces; see fig. 11. Here, $N^{2/3}$ beams run from the face

composite material A – irregular partitioning – $1/h = 6N^{1/3}$

N (#cores-1)	κ	*its*	$\lvert\Pi'\rvert$	$\lvert\Pi\rvert$	*nnz (coarse)*	*#dofs*	*time*	*eff.*
216	54.82	60	9483	4134	7.38e+6	1.02e+6	252s	**100%**
512	60.58	65	24705	10345	2.05e+7	2.41e+6	339s	74%
1000	72.63	73	49122	15010	3.62e+7	4.67e+6	571s	44%

TABLE 1 Weak parallel scalability of adaptive FETI-DP with one subdomain per core plus one load balancing core, stiffness-scaling, the generalized transformation-of-basis approach, and block size 10 for the eigensolver. Compressible linear elasticity of composite material A with $E_1 = 1$ and $N^{2/3}$ beams with $E_2 = 1e+6$ on the unit cube; $\nu = 0.3$ for the whole domain; conforming $\mathcal{P}_2$ discretization with $1/h = 6N^{1/3}$ and irregular partitioning of the domain; see fig. 10. Coarse spaces for $\text{TOL} = 50 \log\left(\sqrt[3]{N/n_i}\right)$ for each generalized eigenvalue problem. N denotes the number of subdomains, κ the condition number estimates from the PCG iteration, *its* the number of iterations until convergence, $\lvert\Pi'\rvert$ the size of the a priori, and $\lvert\Pi\rvert$ the size of the adaptive coarse space. The number of nonzeros in the coarse matrix is given by *nnz (coarse)*, the global number of degrees of freedom is *#dofs*, the total runtime in minutes is denoted by *time*, and the efficiency is given by *eff.*

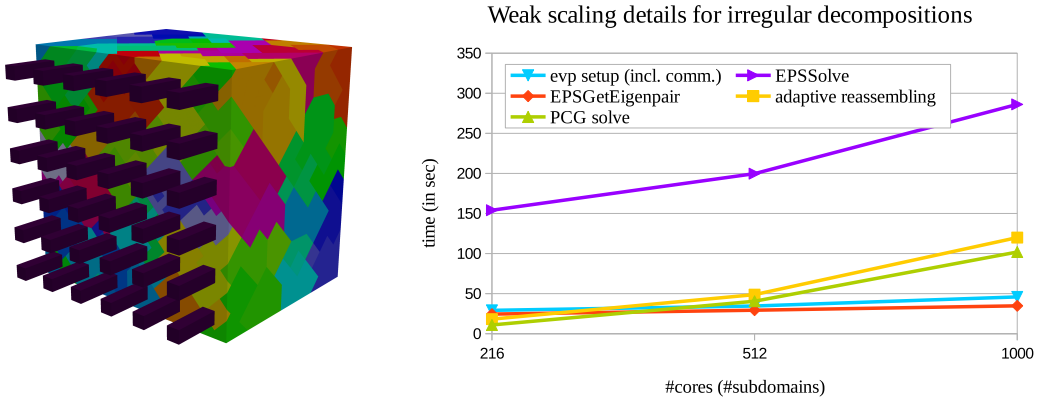


FIGURE 10 Composite material A on the unit cube for 216 subdomains: 36 beams of a stiff material with $E_2 = 1e+6$, $\nu = 0.3$, shown in dark purple for $x > \frac{4}{5}$, surrounded by a soft matrix material with $E_1 = 1$, $\nu = 0.3$ not shown; Irregular decomposition using METIS[84] for 216 subdomains; subdomains shown in different colors for $x < \frac{4}{5}$ (left). Weak scaling details for adaptive FETI-DP with one subdomain per core, stiffness-scaling, the generalized transformation-of-basis approach, and block size 10 for the eigensolver (right); plot of the of the weak scaling of the total runtime as well as of the most expensive code parts; the parts `EPSSolve` and `EPSGetEigenpair` refer to the multiple calls of the corresponding SLEPc functions (timed in total per subdomain).

with $x = 0$ to the face with $x = 1$ but are chopped each time they cross the interface. We refer to this material as *composite material B*.

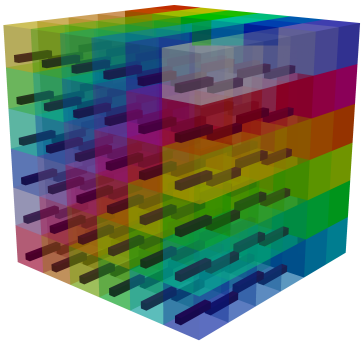
As before, we run our parallel implementation of adaptive FETI-DP with two levels of load balancing (static and asynchronous dynamic) and the elimination of certain edge as well as face eigenvalue problems; see section 5.2 and section 5.3.

For the regular decomposition, we present weak scaling results from 216 to 8000 subdomains; see table 2. As before, the conjugate gradients scheme is expected to be limited in the scaling since exact coarse solves are used. However, all other essential code parts scale well; see fig. 11. In total, we still have an efficiency of 51% when scaling from 216 to 8000 subdomains.

composite material B – regular partitioning – $1/h = 6N^{1/3}$

N (#cores-1)	κ	*its*	$\lvert\Pi'\rvert$	$\lvert\Pi\rvert$	*nnz (coarse)*	*#dofs*	*time*	*eff.*
216	14.02	35	375	1677	2.04e+5	1.17e+6	209s	**100%**
512	20.12	39	1029	4175	5.43e+5	2.74e+6	274s	76%
1000	24.60	43	2187	8551	1.18e+6	5.31e+6	266s	78%
1728	27.91	47	3993	15512	2.25e+6	9.15e+6	283s	74%
2744	30.22	50	6591	25545	3.82e+6	1.45e+7	320s	65%
4096	32.90	52	10125	39173	5.98e+6	2.16e+7	381s	55%
8000	39.39	59	20577	79097	1.24e+7	4.20e+7	409s	51%

TABLE 2 Weak parallel scalability of adaptive FETI-DP with one subdomain per core plus one load balancing core; using stiffness-scaling, the generalized transformation-of-basis approach, and block size 10 for the eigensolver. Compressible linear elasticity of composite material B with $E_1 = 1$ and $N^{2/3}$ beams with $E_2 = 1e+6$ on the unit cube; $\nu = 0.3$ for the whole domain; conforming P_2 discretization with $1/h = 6N^{1/3}$ and regular partitioning of the domain; see fig. 11. Coarse spaces for TOL $= 50 \log\left(\sqrt[3]{N/n_i}\right)$ for each generalized eigenvalue problem. Same notation as in table 1.



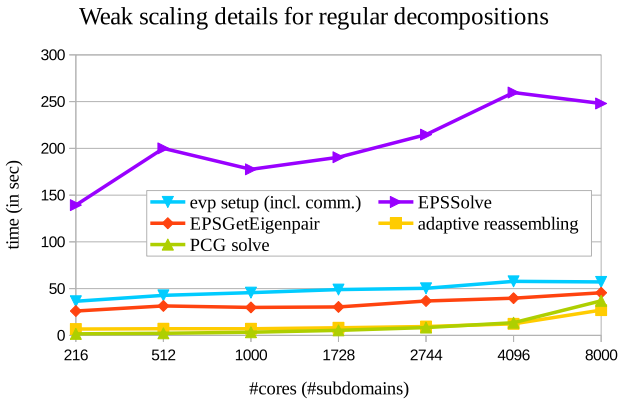


FIGURE 11 Composite material B on the unit cube for 216 subdomains: 36 chopped beams of a stiff material with $E_2 = 1e+6$ are surrounded by a soft matrix material with $E_1 = 1$. Regular decomposition for 216 subdomains; high coefficients are shown in dark purple; subdomains shown in different colors (left). Compressible linear elasticity with $\nu = 0.3$ for the whole domain. Weak scaling details for adaptive FETI-DP with one subdomain per core plus one load balancing core, stiffness-scaling, and the generalized transformation-of-basis approach (right); plot of the of the weak scaling of the total runtime as well as of the most expensive code parts; the parts `EPSSolve` and `EPSGetEigenpair` refer to the multiple calls of the corresponding SLEPc functions (timed in total per subdomain).

6.5 | Strong parallel scalability for adaptive FETI-DP on irregular decompositions

We now consider strong scaling using a decomposition from mesh partitioners. The strong scaling is conducted with one subdomain per core. An additional core is devoted to the lightweight load balancing of the eigenvalue problems.

Our example is motivated by[70]. We consider the hemisphere $\Omega := \{(x, y, z) \in \mathbb{R}^3 \ : \ 0.8 < \|(x, y, z)^T\|_2 < 1, \ z < 0\}$ with five alternating layers of stiff and soft materials; see fig. 12 (left). The geometry and the semi-structured surface mesh are created by means of SALOME v8.3.0[88] and NETGEN 1D-2D[89]. The surface mesh is denoted semi-structured since the discretization parameter h for the unstructured mesh is only in the narrow interval $[0.01, 0.018]$. The volumetric mesh then is created from the surface mesh by the Gmsh mesh generator[90] obtaining 2.6 million degrees of freedom and 580 thousand tetrahedral elements.

layered hemisphere – irregular partitioning – 2.58e+6 *dofs*

N (#cores-1)	κ	*its*	$\lvert\Pi'\rvert$	$\lvert\Pi\rvert$	*nnz (coarse)*	*time*	*eff.*
72	109.21	89	1248	4733	2.97e+6	1925s	**100%**
144	80.62	80	3003	9392	6.46e+6	890s	108%
288	77.20	81	9843	13704	1.33e+7	436s	110%
576	72.02	74	24033	16251	2.17e+7	224s	107%

TABLE 3 Strong parallel scalability of adaptive FETI-DP with one subdomain per core plus one load balancing core, stiffness-scaling, the generalized transformation-of-basis approach, and block size 10 for the eigensolver. Compressible linear elasticity of a layered hemisphere with $E_1 = 1$ and thin (but jagged) layers with $E_2 = 1e+6$; $\nu = 0.3$ for the whole domain; conforming $\mathcal{P}_2$ discretization with 2.58e+6 degrees of freedom and an irregular partitioning of the domain. Coarse spaces for TOL $= 50 \log\left(\sqrt[3]{N/n_i}\right)$ for each generalized eigenvalue problem. Same notation as in Table 1.

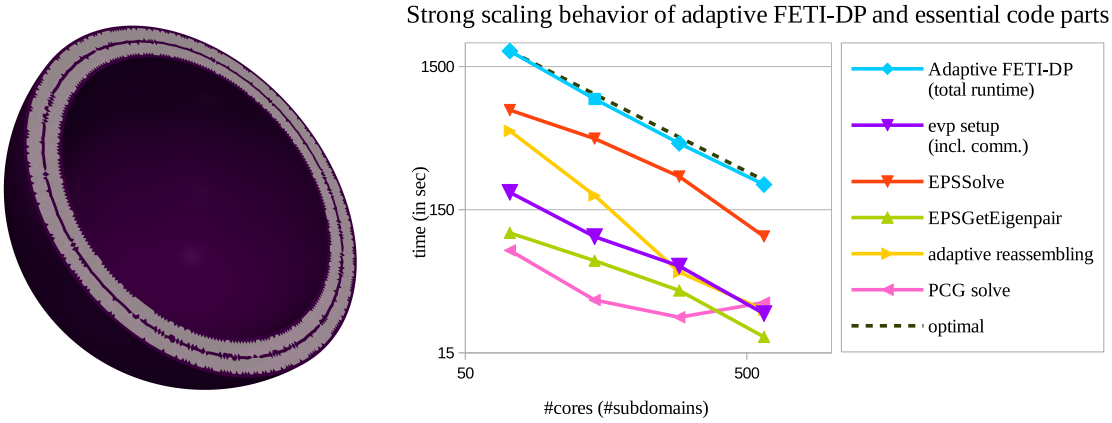


FIGURE 12 Hemisphere with thin layers of a stiff material ($E_2 = 1e+6$, $\nu = 0.3$) at the inner and outer surfaces and in a layer inside the hemisphere (shown in dark purple). The soft matrix material ($E_1 = 1$, $\nu = 0.3$) is shown in light, half-transparent gray (left); approx. 2.58 million degrees of freedom. The interface between the materials is jagged. Strong scaling details for adaptive FETI-DP with one subdomain per core plus one load balancing core, stiffness-scaling, the generalized transformation-of-basis approach, and block size 1 for the eigensolver (right); the total runtime is shown as well as expensive code parts; `EPSSolve` and `EPSGetEigenpair` refer to the multiple calls of the corresponding SLEPc functions (timed in total per subdomain).

For each tetrahedron, we compute its mass center c_T and set the coefficient on the element to $E_2 = 1e+6$ if $c_T > 0.98$, $c_T < 0.82$, or $c_T \in (0.89, 0.91)$. Thus, we obtain thin layers of a stiff material at the boundary and inside the geometry. By construction, the layers are not smooth; see fig. 12 (left).

In Table 3, we have an optimal scaling behavior from 72 to 576 cores. Except for the conjugate gradients scheme, all essential code parts scale very well. We clearly benefit from the superlinear complexity of the eigensolver and the reduced size of the local problems. In contrast to our previous results in[39], the load balancing avoids a drop in efficiency when going from 144 to 288 subdomains.

Note that for $N \in \{72, 144\}$ subdomains the memory usage is dominated by the local generalized eigenvalue problems, i.e., for 72 subdomains and cores, the peak memory consumption is approximately 200 GB per node (8.3 GB per core). For 144 subdomains and cores, the peak memory consumption is about 125 GB per node (5.2 GB per core). For $N = 576$, the memory usage is dominated by the coarse problem, which is already very large, a priori.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the computing time granted by the Center for Computational Sciences and Simulation (CCSS) of the University of Duisburg-Essen and provided on the supercomputer magnitUDE (DFG grants INST 20876/209-1 FUGG, INST 20876/243-1 FUGG) at the Zentrum für Informations- und Mediendienste (ZIM).

The corresponding author also gratefully acknowledges the funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 824158.

References

1. Bjørstad P, Koster J, Krzyżanowski P. Domain Decomposition Solvers for Large Scale Industrial Finite Element Problems. In: Sørevik T, Manne F, Gebremedhin AH, Moe R., eds. *Applied Parallel Computing. New Paradigms for HPC in Industry and Academia*. 1947 of *Lecture Notes in Comput. Sci.* Berlin: Springer. 2001 (pp. 373–383).

2. Bjørstad P, Krzyżanowski P. A Flexible 2-Level Neumann-Neumann method for structural analysis problems. In: Wyrzykowski R, Dongarra J, Paprzycki M, Waśniewski J., eds. *Parallel Processing and Applied Mathematics*. 2328 of *Lecture Notes in Comput. Sci.* Berlin: Springer. 2002 (pp. 387–394).

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

4. Sousedík B. *Comparison of some domain decomposition methods*. PhD thesis. Czech Technical University in Prague, Prague; 2008.

5. Galvis J, Efendiev Y. Domain decomposition preconditioners for multiscale flows in high-contrast media. *Multiscale Model. Simul.* 2010; 8(4): 1461–1483.

6. Galvis J, Efendiev Y. Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces. *Multiscale Model. Simul.* 2010; 8(5): 1621–1644.

7. Nataf F, Xiang H, Dolean V. A two level domain decomposition preconditioner based on local Dirichlet-to-Neumann maps. *C. R. Math. Acad. Sci. Paris* 2010; 348(21-22): 1163–1167.

8. Nataf F, Xiang H, Dolean V, Spillane N. A coarse space construction based on local Dirichlet-to-Neumann maps. *SIAM J. Sci. Comput.* 2011; 33(4): 1623–1642.

9. Dolean V, Nataf F, Scheichl R, Spillane N. Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet-to-Neumann maps. *Comput. Methods Appl. Math.* 2012; 12(4): 391–414.

10. Mandel J, Sousedík B, Šístek J. Adaptive BDDC in three dimensions. *Math. Comput. Simulation* 2012; 82(10): 1812–1831.

11. Šístek J, Mandel J, Sousedík B. Some practical aspects of parallel adaptive BDDC method. In: Acad. Sci. Czech Repub. Inst. Math., Prague. 2012 (pp. 253–266).

12. Sousedík B, Šístek J, Mandel J. Adaptive-multilevel BDDC and its parallel implementation. *Computing. Archives for Scientific Computing* 2013; 95(12): 1087–1119.

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

14. Spillane N, Dolean V, Hauret P, Nataf F, Pechstein C, Scheichl R. Achieving robustness through coarse space enrichment in the two level Schwarz framework. In: Erhel J, Gander MJ, Halpern L, Pichot G, Sassi T, Widlund O., eds. *Domain decomposition methods in Science and Engineering XXI*. 98 of *Lecture Notes in Computational Science and Engineering*. Springer, Cham. 2014 (pp. 447–455).

15. Spillane N, Rixen DJ. Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms. *Internat. J. Numer. Methods Engrg.* 2013; 95(11): 953–990.

16. Dohrmann C, Pechstein C. In C. Pechstein, Modern domain decomposition solvers - BDDC, deluxe scaling, and an algebraic approach. *Slides to a talk at NuMa Seminar, JKU Linz, December 10th, 2013, http://people.ricam.oeaw.ac.at/c.pechstein/pechstein-bddc2013.pdf* 2013.

17. Spillane N. *Méthodes de décomposition de domaine robustes pour les problèmes symétriques définis positifs*. PhD thesis. École Doctorale Paris Centre, Paris; 2014.

18. Haferssas R, Jolivet P, Nataf F. A robust coarse space for optimized Schwarz methods: SORAS-GenEO-2. *C. R. Math. Acad. Sci. Paris* 2015; 353(10): 959–963.

19. Klawonn A, Lanser M, Radtke P, Rheinbach O. On an adaptive coarse space and on nonlinear domain decomposition. In: Erhel J, Gander MJ, Halpern L, Pichot G, Sassi T, Widlund O., eds. *Domain decomposition methods in Science and Engineering XXI*. 98 of *Lecture Notes in Computational Science and Engineering*. Springer, Cham. 2014 (pp. 71–83).

20. Klawonn A, Radtke P, Rheinbach O. FETI-DP methods with an adaptive coarse space. *SIAM J. Numer. Anal.* 2015; 53(1): 297–320.

21. Klawonn A, Radtke P, Rheinbach O. A comparison of adaptive coarse spaces for iterative substructuring in two dimensions. *Electron. Trans. Numer. Anal.* 2016; 45: 75–106.

22. Radtke P. *Adaptive coarse spaces for FETI-DP and BDDC methods*. PhD thesis. Universität zu Köln, Cologne; 2015.

23. Gosselet P, Rixen D, Roux FX, Spillane N. Simultaneous FETI and block FETI: robust domain decomposition with multiple search directions. *Internat. J. Numer. Methods Engrg.* 2015; 104(10): 905–927.

24. Kim HH, Chung ET. A BDDC algorithm with enriched coarse spaces for two-dimensional elliptic problems with oscillatory and high contrast coefficients. *Multiscale Model. Simul.* 2015; 13(2): 571–593.

25. Klawonn A, Kühn M, Rheinbach O. Adaptive coarse spaces for FETI-DP in three dimensions. *SIAM J. Sci. Comput.* 2016; 38(5): A2880-A2911.

26. Zampini S. PCBDDC: a class of robust dual-primal methods in PETSc. *SIAM J. Sci. Comput.* 2016; 38(5): S282–S306.

27. Gander MJ, Loneland A, Rahman T. Analysis of a New Harmonically Enriched Multiscale Coarse Space for Domain Decomposition Methods. tech. rep., ; : 2015. https://arxiv.org/abs/1512.05285.

28. Heinlein A, Klawonn A, Knepper J, Rheinbach O. Multiscale Coarse Spaces for Overlapping Schwarz Methods Based on the ACMS Space in 2D. *Electron. Trans. Numer. Anal.* 2018; 48: 156-182.

29. Agullo E, Giraud L, Poirel L. Robust coarse spaces for Abstract Schwarz preconditioners via generalized eigenproblems. tech. rep., ; : 2016. [Research Report] RR-8978, INRIA Bordeau. https://hal.inria.fr/hal-01399203.

30. Veiga B. dL, Pavarino LF, Scacchi S, Widlund OB, Zampini S. Adaptive selection of primal constraints for isogeometric BDDC deluxe preconditioners. *SIAM J. Sci. Comput.* 2017; 39(1): A281–A302.

31. Oh DS, Widlund OB, Zampini S, Dohrmann CR. BDDC Algorithms with deluxe scaling and adaptive selection of primal constraints for Raviart-Thomas vector fields. *Math. Comp.* 2018; 87(310): 659–692.

32. Calvo JG, Widlund OB. An Adaptive Choice of Primal Constraints for BDDC Domain Decomposition Algorithms. *Electron. Trans. Numer. Anal.* 2016; 45: 524-544.

33. Zampini S, Tu X. Multilevel balancing domain decomposition by constraints deluxe algorithms with adaptive coarse spaces for flow in porous media. *SIAM J. Sci. Comput.* 2017; 39(4): A1389–A1415.

34. Pechstein C, Dohrmann CR. A unified framework for adaptive BDDC. *Electron. Trans. Numer. Anal.* 2017; 46: 273–336.

35. Kim HH, Chung E, Wang J. BDDC and FETI-DP preconditioners with adaptive coarse spaces for three-dimensional elliptic problems with oscillatory and high contrast coefficients. *J. Comput. Phys.* 2017; 349: 191–214.

36. Eikeland E, Marcinkowski L, Rahman T. Overlapping Schwarz Methods with Adaptive Coarse Spaces for Multiscale Problems in 3D. tech. rep., ; : 2016. https://arxiv.org/abs/1611.00968.

37. Bovet C, Parret-Fréaud A, Spillane N, Gosselet P. Adaptive multipreconditioned FETI: Scalability results and robustness assessment. *Computers & Structures* 2017; 193: 1–20.

38. Klawonn A, Kühn M, Rheinbach O. Adaptive FETI-DP and BDDC methods with a Generalized Transformation of Basis For Heterogeneous Problems. *Electron. Trans. Numer. Anal.* 2018; 49: 1-27. Also Preprint 2017-04 at http://tu-freiberg.de/fakult1/forschung/preprints.

39. Kühn MJ. *Adaptive FETI-DP and BDDC methods for highly heterogeneous elliptic finite element problems in three dimensions*. PhD thesis. Universität zu Köln, Cologne; 2018.

40. Daas HA, Grigori L. A class of efficient locally constructed preconditioners based on coarse spaces. tech. rep., ; : 2018. RR-8978, INRIA Bordeau. hal.inria.fr/hal-01816513, accepted in SIAM Journal on Matrix Analysis and Applications.

41. Jolivet P, Dolean V, Hecht F, Nataf F, Prud'Homme C, Spillane N. High performance domain decomposition methods on massively parallel architectures with freefem++. *J. Numer. Math.* 2012; 20(3-4): 287–302.

42. Jolivet P. *Méthodes de décomposition de domaine. Application au calcul haute performance*. PhD thesis. Université de Grenoble, Grenoble; 2014.

43. Jolivet P, Hecht F, Nataf F, Prud'Homme C. Scalable domain decomposition preconditioners for heterogeneous elliptic problems. *Scientific Programming* 2014; 22(2): 157–171.

44. Haferssas R, Jolivet P, Nataf F. An additive Schwarz method type theory for Lions's algorithm and a symmetrized optimized restricted additive Schwarz method. *SIAM J. Sci. Comput.* 2017; 39(4): A1345–A1365.

45. Heinlein A, Hetmaniuk U, Klawonn A, Rheinbach O. The approximate component mode synthesis special finite element method in two dimensions: parallel implementation and numerical results. *J. Comput. Appl. Math.* 2015; 289: 116–133.

46. Farhat C, Roux FX. A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm. *Int. J. Numer. Meth. Engrg.* 1991; 32: 1205–1227.

47. Farhat C, Roux FX. Implicit parallel processing in structural mechanics. In: Oden JT. , ed. *Computational Mechanics Advances*. 2. North-Holland. 1994 (pp. 1–124).

48. Farhat C, Mandel J, Roux FX. Optimal convergence properties of the FETI domain decomposition method. *Computer Methods in Applied Mechanics and Engineering* 1994; 115: 365-385.

49. Klawonn A, Kühn M, Rheinbach O. Adaptive Coarse Spaces for FETI-DP in Three Dimensions with Applications to Heterogeneous Diffusion Problems. In: Accepted for publication in Lecture Notes in Computational Science and Engineering. Springer International Publishing. 2016. Proceedings of the 23rd International Conference on Domain Decomposition Methods in Science and Engineerung, Jeju Island, South Korea, July 5-10, 2015; 8 p.

50. Mandel J, Tezaur R. Convergence of a Substructuring Method with Lagrange Multipliers. *Numer. Math.* 1996; 73: 473–487.

51. Klawonn A, Widlund OB. FETI and Neumann-Neumann iterative substructuring methods: connections and new results. *Comm. Pure Appl. Math.* 2001; 54(1): 57–90.

52. Farhat C, Chen PS, 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.

53. Farhat C, Mandel J. The two-level FETI method for static and dynamic plate problems. I. An optimal iterative solver for biharmonic systems. *Comput. Methods Appl. Mech. Engrg.* 1998; 155(1-2): 129–151.

54. Farhat C, Chen PS, Mandel J, Roux FXX. The two-level FETI method. II. Extension to shell problems, parallel implementation and performance results. *Comput. Methods Appl. Mech. Engrg.* 1998; 155(1-2): 153–179.

55. Farhat C, Lesoinne M, Le Tallec P, Pierson K, Rixen D. FETI-DP: a dual-primal unified FETI method. I. A faster alternative to the two-level FETI method. *Internat. J. Numer. Methods Engrg.* 2001; 50(7): 1523–1544.

56. Farhat C, Lesoinne M, Pierson K. A scalable dual-primal domain decomposition method. *Numer. Linear Algebra Appl.* 2000; 7(7-8): 687–714. Preconditioning techniques for large sparse matrix problems in industrial applications (Minneapolis, MN, 1999).

57. Mandel J, Tezaur R. On the convergence of a dual-primal substructuring method. *Numer. Math.* 2001; 88(3): 543–558.

58. Klawonn A, Widlund OB, Dryja M. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM J. Numer. Anal.* 2002; 40(1): 159–179.

59. Klawonn A, Widlund OB. Dual-primal FETI methods for linear elasticity. *Comm. Pure Appl. Math.* 2006; 59(11): 1523–1572.

60. Bhardwaj M, Pierson KH, Reese G, et al. Salinas: A Scalable Software for High Performance Structural and Mechanics Simulation. In: IEEE. ; 2002; Baltimore, MD, USA: 1–19

61. Klawonn A, Rheinbach O. Highly scalable parallel domain decomposition methods with an application to biomechanics. *ZAMM Z. Angew. Math. Mech.* 2010; 90(1): 5–32.

62. Klawonn A, Lanser M, Rheinbach O. Toward extremely scalable nonlinear domain decomposition methods for elliptic partial differential equations. *SIAM J. Sci. Comput.* 2015; 37(6): C667–C696.

63. Říha L, Brzobohatý T, Markopoulos A, Meca O, Kozubek T. Massively Parallel Hybrid Total FETI (HTFETI) Solver. In: ACM; 2016.

64. Rheinbach O. *Parallel scalable iterative substructuring: Robust exact and inexact FETI-DP methods with applications to elasticity*. PhD thesis. Universität Duisburg-Essen, Essen; 2006.

65. Toivanen J, Avery P, Farhat C. A multilevel FETI-DP method and its performance for problems with billions of degrees of freedom. *International Journal for Numerical Methods in Engineering* 2018; 116(10-11): 661–682.

66. Toselli A, Widlund OB. *Domain Decomposition Methods - Algorithms and Theory*. 34 of *Springer Series in Computational Mathematics*. Berlin Heidelberg New York: Springer-Verlag . 2005.

67. Klawonn A, Kühn M, Rheinbach O. Coarse spaces for FETI-DP and BDDC methods for heterogeneous problems: Connections of Deflation and a Generalized Transformation-of-Basis approach. tech. rep., Technische Universität Bergakademie Freiberg, Fakultät für Mathematik und Informatik, Preprint 2017-01; Freiberg: 2017. http://tu-freiberg.de/fakult1/forschung/preprints; submitted for publication.

68. Li J, Widlund OB. FETI-DP, BDDC, and block Cholesky methods. *International Journal for Numerical Methods in Engineering* 2006; 66(2): 250–271.

69. Klawonn A, Rheinbach O. A parallel implementation of dual-primal FETI methods for three-dimensional linear elasticity using a transformation of basis. *SIAM J. Sci. Comput.* 2006; 28(5): 1886–1906.

70. Klawonn A, Rheinbach O. Robust FETI-DP methods for heterogeneous three dimensional elasticity problems. *Comput. Methods Appl. Mech. Engrg.* 2007; 196(8): 1400–1414.

71. Klawonn A, Rheinbach O. Deflation, projector preconditioning, and balancing in iterative substructuring methods: connections and new results. *SIAM J. Sci. Comput.* 2012; 34(1): A459–A484.

72. Balay S, Abhyankar S, Adams MF, et al. PETSc Users Manual. Tech. Rep. ANL-95/11 - Revision 3.8, Argonne National Laboratory; : 2017.

73. Balay S, Gropp WD, McInnes LC, Smith BF. Efficient Management of Parallelism in Object Oriented Numerical Software Libraries. In: Arge E, Bruaset AM, Langtangen HP., eds. *Modern Software Tools in Scientific Computing*Birkhäuser; 1997: 163–202.

74. Schenk O, Gärtner K, Fichtner W. Efficient sparse *LU* factorization with left-right looking strategy on shared memory multiprocessors. *BIT* 2000; 40(1): 158–176.

75. Hernández V, Román JE, Vidal V. SLEPc: A Scalable and Flexible Toolkit for the Solution of Eigenvalue Problems. *ACM Trans. Math. Software* 2005; 31(3): 351–362.

76. Román JE, Campos C, Romero E, Tomás A. SLEPc Users Manual. Tech. Rep. DSIC-II/24/02 - Revision 3.8, D. Sistemes Informàtics i Computació, Universitat Politècnica de València; : 2017.

77. Rheinbach O. Parallel iterative substructuring in structural mechanics. *Arch. Comput. Methods Eng.* 2009; 16(4): 425–463.

78. Dohrmann CR, Widlund OB. Some recent tools and a BDDC algorithm for 3D problems in $H(\mathrm{curl})$. In: Bank R, Holst M, Widlund O, Xu J. , eds. *Domain decomposition methods in Science and Engineering XX*. 91 of *Lecture Notes in Computational Science and Engineering*. Springer, Heidelberg. 2013 (pp. 15–25).

79. Klawonn A, Kühn M, Rheinbach O. Preconditioning of Iterative Eigenvalue Problem Solvers in Adaptive FETI-DP. In: Cham: Springer Publishing. 2018. Accepted for the proceedings of the Internat. Conference on DDM XXIV, Svalbard, Norway, Feb. 6-10, 2017.

80. Stewart GW. A Krylov-Schur algorithm for large eigenproblems. *SIAM J. Matrix Anal. Appl.* 2001/02; 23(3): 601–614.

81. Hernández V, Román JE, Tomás A, Vidal V. Krylov-Schur Methods in SLEPc. tech. rep., D. Sistemes Informàtics i Computació, Universitat Politècnica de València; : 2007. SLEPc Tech. Report STR-07, http://slepc.upv.es/documentation/reports/str7.pdf.

82. Wu K, Simon H. Thick-restart Lanczos method for large symmetric eigenvalue problems. *SIAM J. Matrix Anal. Appl.* 2000; 22(2): 602–616.

83. Klawonn A, Kühn M, Rheinbach O. Adaptive Coarse Spaces for FETI-DP in Three Dimensions with Applications to Heterogeneous Diffusion Problems. In: . 116 of *Lecture Notes in Computational Science and Engineering*. Cham: Springer International Publishing. 2017 (pp. 187–196).

84. Karypis G, Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* 1998; 20(1): 359–392.

85. Sousedík B. *Adaptive-Multilevel BDDC*. PhD thesis. University of Colorado Denver, Denver; 2010.

86. Knyazev AV. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method; MATLAB Implementation. https://www.mathworks.com/matlabcentral/fileexchange/48-lobpcg-m, accessed: 2015-12-09; .

87. Klawonn A, Kühn M, Rheinbach O. A closer look at local eigenvalue solvers for adaptive FETI-DP and BDDC. In: Springer-Verlag. 2018. Submitted to the proceedings of the International Conference on Domain Decomposition Methods XXV, St. John's, Canada, July 23-27, 2018. Also Technical Report CDS-2018-11, https://kups.ub.uni-koeln.de/9020/.

88. Ribes A, Caremoli C. Salome platform component model for numerical simulation. In: . 2. ; 2007: 553-564.

89. Schöberl J. NETGEN An advancing front 2D/3D-mesh generator based on abstract rules. *Computing and Visualization in Science* 1997; 1(1): 41–52.

90. Geuzaine C, Remacle JF. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Internat. J. Numer. Methods Engrg.* 2009; 79(11): 1309–1331.

How to cite this article: A. Klawonn, M. J. Kühn, and O. Rheinbach (2019), Parallel adaptive FETI-DP using lightweight asynchronous dynamic load balancing, ., ..