Fakultät für Maschinenwesen
Lehrstuhl für Angewandte Mechanik

Dual Domain Decomposition Methods in Structural Dynamics

Efficient Strategies for Finite Element Tearing and Interconnecting

Michael Christopher Dominik Leistner

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Michael W. Gee

Prüfer der Dissertation:

1. Prof. dr. ir. Daniel Rixen

2. Dr. habil. à diriger des recherches Pierre Gosselet

Die Dissertation wurde am 15.11.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 15.02.2019 angenommen.

Abstract

Parallel iterative domain decomposition methods are an essential tool to simulate structural mechanics because modern hardware takes its computing power almost exclusively from parallelization. These methods have undergone enormous development in the last three decades but still some important issues remain to be solved. This thesis focuses on the application of dual domain decomposition to problems of linear structural dynamics and the two main issues that appear in this context. First, the properties and the performance of known a priori coarse spaces are assessed for the first time as a function of material heterogeneity and the time step size. Second, strategies for the efficient repeated solution of the same operator with multiple right-hand sides, as it appears in linear dynamics, are investigated and developed. While direct solution methods can easily reuse their factorization of the operator, it is more difficult in iterative methods to efficiently reuse the gathered information of former solution steps. The ability of known recycling methods to construct tailor-made coarse spaces is assessed and improved. In particular, this work and the according preceding papers present the first highly efficient recycling strategies for the recently published adaptive multipreconditioning framework.

Zusammenfassung

Parallele iterative Gebietszerlegungsverfahren sind ein essenzielles Werkzeug zur Simulation von Strukturmechanik, denn moderne Hardware schöpft ihre Rechenleistung fast ausschließlich aus Parallelisierung. Diese Methoden haben in den letzten drei Jahrzehnten eine enorme Entwicklung durchlaufen, doch es gibt noch einige wichtige Probleme, die gelöst werden müssen. Diese Arbeit konzentriert sich auf die Anwendung von Gebietszerlegungsverfahren auf lineare Probleme der Strukturdynamik und die beiden Hauptprobleme die sich in diesem Zusammenhang ergeben. Dies sind erstens die Eigenschaften und die Leistungsfähigkeit von bekannten a priori Coarse Spaces, welche erstmalig in Abhängigkeit der Material-Heterogenität und der Zeitschrittgröße untersucht werden. Der zweite Punkt ist die Untersuchung und Entwicklung von Strategien für die effiziente mehrfache Lösung von konstanten Operatoren mit wechselnder rechter Seite. Während direkte Lösungsverfahren einfach die Faktorisierung des Operators wiederverwenden können, ist es in iterativen Verfahren schwieriger die gesammelten Informationen aus vorhergehenden Lösungsschritten effizient wiederzuverwenden. Die Fähigkeit bekannter Recyclingstrategien, maßgeschneiderte Coarse Spaces aufzubauen, wird untersucht und verbessert. Insbesondere werden in dieser Arbeit und in den entsprechenden vorangegangenen Publikationen die ersten hocheffizienten Recyclingstrategien für das kürzlich veröffentlichte multipreconditioning Verfahren präsentiert.

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Angewandte Mechanik der Technischen Universität München unter der Leitung meines Doktorvaters Prof. dr. ir. Daniel Rixen. An erster Stelle gilt ihm mein besonderer Dank für seine immer wohlwollende Unterstützung und für die stets exzellenten Rahmenbedingungen in fachlichen wie menschlichen Belangen. Sein mir entgegengebrachtes Vertrauen, die von ihm geförderten persönlichen Entwicklungsmöglichkeiten und Gestaltungsfreiräume sowie natürlich die steten fachlichen Diskussionen und Anregungen haben maßgeblich zum Gelingen meiner Arbeit beigetragen.

Ebenso danke ich Dr. habil. Pierre Gosselet für sein großes Interesse an meiner Arbeit und für seine Tätigkeit als Zweitprüfer. Prof. Dr.-Ing. Michael W. Gee danke ich für die Übernahme des Vorsitzes der Prüfungskommission.

Das Gelingen meiner Arbeit und dass ich die Zeit am Lehrstuhl in besonders positiver Erinnerung behalten werde, ist auch maßgeblich den vielen aktuellen und ehemaligen Mitarbeitern des Lehrstuhls, mit denen ich dort zusammenarbeiten durfte, zu verdanken. Viele wertvolle fachliche Diskussionen, die gemeinsame Teilnahme an Tagungen, die vielen gemeinsamen sportlichen Aktivitäten, sowie das stets freundschaftliche und von Hilfsbereitschaft geprägte Klima haben sehr dazu beigetragen. Ganz besonderer Dank gilt meinem langjährigen Kollegen Oliver Hofmann, dessen einzigartig motivierende, positive Persönlichkeit ich sehr schätzen gelernt habe und der mich in zahlreichen Situationen in besonderer Weise unterstützt hat.

Schließlich möchte ich meiner gesamten Familie und ganz besonders meinen Eltern für die stets umfassende und bedingungslose Unterstützung während meines bisherigen Werdegangs danken, der ohne diese Unterstützung nicht möglich gewesen wäre.

München, im März 2019 *Michael Leistner*

Contents

Nomenclature

Acronyms

FEM Finite element method

dof Degree of freedom

BDD Balancing domain decomposition

FETI Finite element tearing and interconnecting

AMP-FETI Adaptive multipreconditioned FETI

FETI-DP FETI dual-primal

GenEO Generalized eigenvalues in the overlaps

TRKS Total reuse of Krylov subspaces

SRKS Selective reuse of Krylov subspaces

CG Conjugate gradient

PPCG Projected preconditioned conjugate gradient

GMRES Generalized minimal residual

LMS Linear multistep

GSSSS Generalized single-step single-solve

RK Runge-Kutta

Gmsh A finite element mesh generator software

METIS A mesh partitioning algorithm

Continuum Mechanics

$\overline{\boldsymbol{F}}$ Deformation gradient

$\overline{\boldsymbol{S}}$ Second Piola-Kirchhoff stresses

$\overline{\boldsymbol{E}}$ Green-Lagrange strain

$\overline{W}$ Strain energy

$\overline{N}$ Normal vector of the surface

$\overline{\boldsymbol{u}}$ Exact continuous displacements

$\ddot{\overline{\boldsymbol{u}}}$ Exact continuous accelerations

$\overline{\boldsymbol{u}}_h$ Approximated continuous displacements

$\ddot{\overline{\boldsymbol{u}}}_h$ Approximated continuous accelerations

$\overline{\boldsymbol{v}}$ Continuous virtual displacements

ρ_0 Density in initial configuration

ρ Density

ν Poisson ratio

E Young's modulus

$\overline{\boldsymbol{b}}$ Body forces

$\overline{\boldsymbol{t}}$ Traction forces

Ω Computational domain in inital configuration

Γ Boundary of the computational domain

Γ_N Neumann boundary

Γ_D Dirichlet boundary

$\boldsymbol{N}$ Shape functions

$\boldsymbol{v}$ Nodal virtual displacements

Non-overlapping Domain Decomposition

$\star_I$ Restriction to the interface part of the boundary

$\star_r$ Restriction to the non-interface part of the boundary

$\hat{\boldsymbol{f}}_{\mathbf{ext}}$ Equivalent condensed external force

$\star^{+}$ Pseudoinverse

$\star^{\perp}$ Orthogonal complement

Elasticity and Motion

$\boldsymbol{M}$ Mass matrix

$\boldsymbol{K}$ Stiffness matrix

$\boldsymbol{S}_K$ Schur complement of the stiffness matrix on the interface

$\boldsymbol{f}_{\mathbf{ext}}$ Nodal external force

$\boldsymbol{f}_{\mathbf{int}}$ Nodal internal force

$\boldsymbol{u}$ Nodal displacements

$\dot{\boldsymbol{u}}$ Nodal velocities

$\ddot{\boldsymbol{u}}$ Nodal accelerations

Time Integration and Dynamic Problems

t Time

Δt Time step size

T_i Period of the structure's i-th eigenmode

γ_N Time integration parameter

β_N Time integration parameter

α_m Time integration parameter

α_f Time integration parameter

$\boldsymbol{D}$ Stepping matrix, i.e., operator of the time-discretized equations of motion

$\boldsymbol{S}_D$ Schur complement of the stepping matrix on the interface

$\boldsymbol{g}$ Right-hand side of the time-discretized equations of motion

$\star^{n}$ Time step index

Dual Interface Problem

N_s Number of substructures

N_f Number of floating substructures

N_λ Size of the interface problem

$N_{\mathbf{rbm}}$ Global number of rigid body modes

N_C Number of coarse space basis vectors

μ Multiplicity of a node

$\boldsymbol{R}$ Basis of the core of the stiffness matrix

$\boldsymbol{F}$ Operator of the interface problem

$\boldsymbol{H}$ Preconditioner of the interface problem

$\boldsymbol{d}$ Right-hand side of the interface problem

$\boldsymbol{e}$ Right-hand side of the self-equilibrium condition / error of the CG solution

$\boldsymbol{\lambda}$ Lagrange multipliers or interface forces

$\boldsymbol{\lambda}_G$ Lagrange multiplier part fulfilling the natural coarse problem condition

$\boldsymbol{\lambda}_C$ Lagrange multiplier part fulfilling the auxiliary coarse problem condition

$\hat{\boldsymbol{\lambda}}$ Lagrange multiplier part from conjugate gradient iterations

Λ Set of Lagrange multipliers

$\boldsymbol{\alpha}_R$ Rigid body mode amplitudes

$\boldsymbol{A}$ Boolean primal assembly matrix of the non-condensed system

$\boldsymbol{E}$ Boolean dual assembly matrix of the non-condensed system

$\boldsymbol{t}$ Trace operator

$\boldsymbol{L}$ Boolean primal assembly matrix of the condensed system

$\boldsymbol{B}$ Boolean dual assembly matrix of the condensed system

$\widetilde{\boldsymbol{B}}$ Scaled boolean dual assembly matrix

$\widetilde{\boldsymbol{\beta}}$ Scaling coefficients

$\boldsymbol{G}$ Natural coarse space basis

$\boldsymbol{C}$ Auxiliary coarse space basis

$\boldsymbol{P}_G$ Natural coarse space projector

$\boldsymbol{P}_C$ Auxiliary coarse space projector

$\star^{(s)}$ Substructure index

Conjugate Gradient

κ Condition number of a matrix

β Orthogonalization coefficient

m Number of CG iterations

$\epsilon_{r,\mathbf{rel}}$ Relative convergence criterion of CG

$\epsilon_{r,\mathbf{abs}}$ Absolute convergence criterion of CG

Coarse Spaces and Recycling Algorithms

$\boldsymbol{y}$ GenEO mode / eigenvector of the preconditioned operator $\boldsymbol{HF}$

θ GenEO value

K GenEO value threshold

Θ Eigenvalue of the preconditioned operator $\boldsymbol{HF}$

$\widetilde{\Theta}$ Approximated eigenvalue of the preconditioned operator $\boldsymbol{HF}$

$\boldsymbol{V}_z$ Ritz space for SRKS

Ψ Principal angle between subspaces

ϵ_Θ Convergence criterion for Ritz values in SRKS

ϵ_κ Target condition number in SRKS

Adaptive Multipreconditioning

Ξ Local convergence measure

τ Local convergence measure threshold

$\widetilde{\theta}$ Approximated GenEO value

$\boldsymbol{V}$ Ritz space for RitzGenEO and RitzDirect

$\boldsymbol{V}_W$ Auxiliary space of solution increments $\Delta\boldsymbol{\lambda}_i$ for Ritz space construction in RitzGenEO and RitzDirect

$\mathbf{J}$ Set of selected substructures

$\epsilon_{r,\mathbf{sub}}$ Relative localized error energy threshold

ϵ_{Δ} Pivot threshold for rank revealing LDLT

$e^{(s)}_{r,i+1}$ Relative localized error energy

$E^{(s)}_{p,i}$ Localized energy of the final step direction

$E^{(s)}_{r,i+1}$ Localized energy of the error

Chapter 1

Introduction

The possibility to precisely predict the behavior of structures by computer simulation, whether they are of very large scale like the wing of an airplane or of much smaller scale like the timing chain of a combustion engine, has revolutionized almost every single field of engineering. It allows to analyze the stability and the operating performance of a specific design of a mechanical part, long before the first prototype has been built. Even structural optimization, which had never been possible to discover by experiment, becomes a widely automated task. The according mathematical methods are not limited to the operating behavior of mechanical parts like in aerospace or automotive structures, but also include complex physical processes like plastic deformation and thermodynamic effects, for example, the welding of structural steel. Today, most product development processes are characterized by a strong interaction of simulation and experiment.

A common method to create mathematical models that can be solved by computer algorithms is to discretize the computational domain, for example, using the finite element method. With increasing demands on accuracy, the discretizations become ever finer, allowing for a more accurate resolution of the physical processes. However, this leads to computational problems with a huge number of unknowns, which in turn requires accordingly large resources of memory and computational power.

Unfortunately, the clock rate of processor cores as well as the memory that is usually addressed by one core did not increase anymore since more than ten years. This is related to physical limits concerning the scale of transistors, thermal issues and energy consumption. Even the effective performance, which is still increasing due to more complex instruction sets and internal parallelization, is almost stagnating as figure 1.1 shows. At the same time, the price for processor cores and memory has dropped significantly. As a result, modern computers, whether smartphone, workstation or supercomputer, take their power from multiple cores. To make use of these new hardware architectures, the numerical methods used to solve linear or nonlinear discretized mathematical problems have to exhibit a high degree of parallelism. That is to say, their routines must divide the computational task in as many as possible independent tasks with as little as possible increase of the overall computational complexity. Furthermore, the necessary amount of communication between the cores, i.e., between the independent tasks, should be very low and should rise only weakly when the number of parallel tasks is increased.

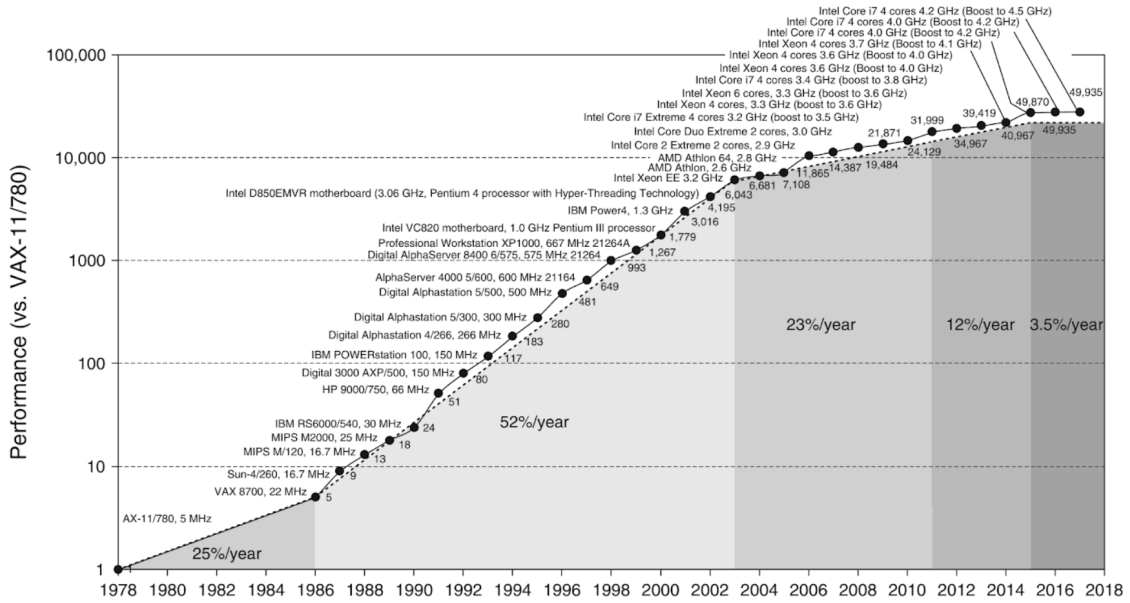


Figure 1.1. Growth in processor performance over 40 years relative to the VAX 11/780 as measured by the SPEC integer benchmark. For more information, see [1]. (Source: [1])

In the field of structural mechanics, domain decomposition methods have been developed since several decades to cope with these requirements. Most of them rely on two basic principles. First, they divide the computational domain into several subdomains so that each of them can be handled by a separate computing core. Second, they search for a solution iteratively. That means they start with a guess for the solution, and repeatedly compute the deformation of the subdomains. After each computation, neighboring domains exchange information in their common region to check if their solutions match. Based in the error, the solution is improved step by step until a sufficient accuracy is attained. Because the underlying mathematical algorithms guarantee convergence toward the exact solution, iterative methods that terminate at a certain accuracy can turn this very little and usually irrelevant deficiency of exactness into significant savings in memory and computation time. However, it is impossible to design domain decomposition methods in which the individual domains, i.e., the computing cores, do not have to communicate with each other.

1.1 Problem Statement

In this thesis, the focus is set on the finite element tearing and interconnecting (FETI) method, one of the most successful domain decomposition methods. In this chapter, no references are given. This is done in the following chapters when the according theory is introduced.

It is based on a non-overlapping decomposition of the domain, which is why the subdomains are usually called substructures in this context. They are connected by

a global set of interface forces, which are called dual unknowns and chosen such that they automatically fulfill the action-reaction principle. The interface forces are improved iteratively, based on the error of the displacement, called primal unknowns, on the substructure interfaces. Mathematically, all the iterative procedures discussed in this thesis are equivalent to a classical or multipreconditioned conjugate gradient (CG) algorithm.

The following key features of the basic method have helped to make it one of the most popular and most powerful methods for parallelizing the computation of large problems in structural mechanics. First, the vast majority of computations is local to a substructure and thus trivial to parallelize. Second, most communication is also local, i.e., only occurs between neighboring substructures. These two important properties also hold for the usual preconditioning methods. Third, by solving a problem of flexibility instead of stiffness, the formulated problem exhibits very advantageous spectral properties that make the CG algorithm find the most important physical modes first. This leads to fast convergence and physically accurate approximations after few iterations. Fourth, the mathematical structure allows to introduce a type of coarse space correction, called coarse space here, which implicates global propagation of error information and thereby leads to good scalability. Based on this foundation, a whole family of dual domain decomposition methods has been established over almost three decades.

In the past, several types of particularly hard problems for this class of dual domain decomposition methods have been investigated and numerous enhancements have been proposed and assessed. The original, basic algorithm works well for homogeneous structures whose mesh is decomposed in substructures with smooth boundaries and balanced aspect ratios. Less optimal conditions can cause slow convergence or breakdown. The hard problems are essentially heterogeneous material distributions, jagged interfaces, and bad aspect ratios. Heterogeneous material distributions occur, for example, in rubber tires that include steel parts. They are most difficult to solve if the material borders do not align with the substructure interfaces but cross them. Jagged interfaces happen especially when structures are decomposed into substructures after they have been meshed. In this case it is often difficult or impossible to find smooth paths along the mesh. Bad aspect ratios mean that the dimensions of a substructure in two different spatial directions have a ratio far from one. To cope with these problems, numerous modifications of the basic preconditioners, mostly scaling procedures, have been proposed. A second family of enhancements are the deflation methods, which build a coarse space that aims to relieve the iterative algorithm by computing the solution within a particularly difficult subspace with a direct solution method. The most recent enhancement is multipreconditioning, which exploits the additive structure of the preconditioner to compute multiple search directions in each iteration. However, most of these strategies have been tailored for the solution of static problems and have also only been applied to such in the majority of cases.

Regarding problems of linear structural dynamics, many important questions have not been investigated thoroughly enough yet. The effects of mass and time on the one hand and the solution of many problems with identical operators but changing right-hand sides on the other hand pose a special class of problems that require

separate consideration. Instead of a pure stiffness, the operator in linear dynamics is a combination of stiffness and mass. Their weights are primarily governed by the time step size, which is why the analyzed time scale is of major importance and can change the systems characteristics substantially. While the special case of linear dynamics seems to reduce the complexity, the fact that the operator remains identical over all time steps calls for a highly efficient reuse of information from one time step to the next. For these multiple right-hand side problems, direct solution methods simply compute a complete factorization of the operator and can subsequently solve all time steps very efficiently and fast. Instead, iterative methods do not offer such a convenient method and require a different approach to solve multiple right-hand side problems efficiently.

In the context of the classical single-preconditioned variants of FETI, a thorough analysis of the requirements on coarse spaces when solving dynamic problems is necessary. Some publications applied the rigid body mode coarse space and also considered heterogeneity, but widely varied scales of time have not been investigated. The more sophisticated and much more capable generalized eigenvalues in the overlaps (GenEO) coarse space has not been considered. It is unclear for which type of dynamic problems coarse spaces are necessary at all, and if they are, how they should be constructed. While basic recycling strategies have been applied to multiple right-hand side problems, their behavior and the according requirements as a function of the parameters heterogeneity and time step size have not been investigated. Similarly unknown is the assessment of more sophisticated recycling strategies like global eigenvector approximations against this background.

The recently published adaptive multipreconditioned FETI (AMP-FETI) has been applied to various linear and nonlinear problems of structural mechanics, while dynamic problems have not been considered yet. When the multipreconditioning framework is applied to solve problems of structural dynamics, new issues like search space degeneration appear. On the one hand these issues allow for improvements in efficiency, on the other hand their specific handling is critical for the successful application of recycling strategies. Because of the repeated solution of the same linear system, such recycling of former solution spaces is one of the most important ingredients for the efficient solution of problems in linear structural dynamics. The most simple recycling strategy known for the classical single-preconditioned framework can be transferred quite intuitively to multipreconditioning, but the resulting performance has not been assessed so far. What's more, it is not possible to apply the more powerful recycling strategies of single-preconditioning to multipreconditioning. Therefore, new methods that are specifically tailored for the new state of the art are absolutely essential to extend its scope to problems of linear structural dynamics.

1.2 Objective and Outline

This thesis discusses the application of the classical and the multipreconditioned FETI method to problems of linear structural dynamics. Non-linear problems are not considered. After a short introduction of the motivation to consider dual domain decomposition methods, an overview of the state of the art is given and the prevailing problems within the context of linear structural dynamics are highlighted. In general, this thesis presents the important aspects of the method that should be considered for the efficient solution of this class of problems and investigates some critical issues that have not been considered so far. In particular, the influence of the time scale, suitable coarse spaces, and recycling strategies are discussed and new methods are proposed to improve the state of the art.

Chapter 2 gives a thorough overview of the important aspects when applying FETI to dynamics. At first, non-overlapping domain decomposition and the concept of primal and dual interfaces, which constitute the basis of everything that follows, are introduced in detail. Next, importance is attached to the iterative solution process because of its particular significance to understand how coarse spaces and solution space recycling work. The physical view of the iterative improvement of the solution is connected to the mathematical framework of conjugate gradients with a natural coarse space projection. Furthermore, the augmentation by auxiliary deflation spaces is illuminated in detail. Important basic enhancements like preconditioners and scaling techniques are introduced, and the most important mathematical framework to describe and predict the convergence by spectral properties is explained.

Chapter 3 adds the surrounding framework of time integration schemes to apply dual domain decomposition methods to linear structural dynamics. It also explains the adaption of preconditioners and scaling procedures, which have been originally designed to solve static problems, to dynamics. In addition, special issues in dynamics like the computation of the initial acceleration and the stability of integration schemes for constrained mechanical systems are considered. Finally, a summary of the fundamental concepts that can be used to improve the convergence and to lower computational cost and time are presented. The short overview also highlights which of them are investigated in more detail in the following chapters.

Chapter 4 provides a thorough analysis of the requirements on coarse spaces and investigates suitable methods to construct them with particular attention to the changing conditions in different scales of time. To introduce the setting, the spectral properties of the dynamic interface problem are investigated as a function of the time step size and heterogeneity. Furthermore, the relation to the equivalent static problem is illustrated. The requirements on coarse spaces depending on the time scale and heterogeneity are analyzed by visualizing the local GenEO modes of a specific substructure. Especially the rigid body mode coarse space and several variants of its application are compared to the GenEO coarse space in order to identify for which types of problems it is able to accelerate the convergence and why it does not work well in other cases. Recycling strategies are introduced as a further method to construct efficient deflation spaces. They allow to reuse the information gathered during

the solution of former time steps to accelerate the solution of future ones. This is crucial to obtain an efficient algorithm because otherwise, the same linear system is solved over and over again without reusing any of the important properties of the system that had already been discovered. The a priori coarse spaces are subsequently compared to those constructed by recycling strategies. In particular, their ability to adapt the chosen subspace and its size to the requirements of the problem, which strongly change for different scales of time and heterogeneities, is investigated. To improve this adaptivity for a wide range of time scales and for the specific needs of linear dynamics, a new selection criterion for the global Ritz approximation strategy is proposed.

Chapter 5 focuses on multipreconditioning and its application to linear dynamics. It starts with the introduction of the theory of adaptive multipreconditioning. To prepare for the analysis and the development of recycling strategies, which is the main focus of the chapter and completely novel in the framework of multipreconditioning, the issue of search space degeneration is investigated. The effect is explained and illustrated by numerical examples. Subsequently, two methods to handle it are proposed. At first, a new method to improve the efficiency of multipreconditioning in case of degenerated search spaces is proposed, which can be used if no recycling strategy should be applied. Next, the important requirement of recycling strategies to have a fully populated search space available from the beginning is introduced. To meet this requirement, a new activation method, which is a slight modification of an existing one, is proposed and assessed numerically. The most important part of the chapter is the introduction and the numerical assessment of recycling strategies for multipreconditioning. For the most basic recycling strategy of single-preconditioning, the plain reuse of solution spaces, the analogous approach for multipreconditioning is introduced. Because the more sophisticated and more powerful recycling strategies of single-preconditioning cannot be transferred to multipreconditioning, a completely novel and comprehensive recycling strategy is presented. It is based on an approximation of the local GenEO eigenproblems. Beside its abilities to significantly improve the rate of convergence very efficiently, it also allows to gain insight into the connection between AMP-FETI and the GenEO coarse space. The recycling strategies are assessed numerically for different types of decomposition, heterogeneity and loading.

Chapter 6 summarizes the results and most important findings of this thesis and outlines possible future directions of research.

1.3 Scientific Contributions

This thesis and the corresponding publications of the author, on which this thesis is partially based, provide scientific contributions on four main areas, which are listed in the following.

A priori coarse spaces for single-preconditioned FETI for dynamic problems. For the first time, this thesis applies the GenEO coarse space to dynamic problems and analyses the performance of GenEO for this class of problems. In particular a visualization and spectral analysis of the GenEO coarse space as a function of the time scale and heterogeneity gives valuable insight into the general requirements on coarse spaces for dynamic problems. The analogous analysis of the behavior and performance of the rigid body mode coarse space results in a unique identification of time scales in which this coarse space can be applied efficiently to homogeneous and heterogeneous problems and when its application should not be considered. A direct comparison between the GenEO coarse space and the rigid body mode coarse space through the relevant time scales explains the underlying effects.

Recycling strategies in single-preconditioned FETI for dynamic problems. This thesis provides a review of the relevant recycling methods. In particular, their ability to deliver tailor-made coarse spaces for heterogeneities and different time scales is investigated for the first time. This includes a decision on the coarse space size, up to the point of a fully automated decision that no coarse space is needed at all. To meet these demands better, a new selection criterion for the global Ritz approximation recycling strategy is proposed that improves robustness and efficiency. An analysis of principal angles between subspaces visualizes the practical limitations of these recycling strategies.

Search space degeneration in multipreconditioned FETI for dynamic problems. This thesis, together with the preceding publication [2] of the author, provides a first dedicated analysis of the effect of search space degeneration in multipreconditioned FETI and explains possible causes. For the case that no recycling strategy should be applied, the proposal of a new criterion to handle the effect prior to the already known, but more complicated mechanisms, allows to improve the efficiency in case of search space degeneration. An analysis of the effect and the behavior of the new criterion is provided and its influence on recycling strategies is explained. Because of the importance of fully populated search spaces for recycling methods, a method that is specifically tailored to these demands is proposed and applied for all following investigations.

Recycling strategies in multipreconditioned FETI for dynamic problems. The author's publications [3, 4] and this thesis address for the first time the urgent demand for recycling strategies for the multipreconditioning framework. A basic,

widely known recycling strategy for single-preconditioning is adapted to multipreconditioning and the new, comprehensive, and competitive recycling strategy *RitzGenEO* is presented. It was first published in [3] and is tailored to multipreconditioning, including an intelligent criterion to preselect the numerous search directions. This thesis and the preceding publication [4] propose furthermore a simplified variant of the *RitzGenEO* recycling strategy, called *RitzDirect*, which provides new insight into the connection between multipreconditioning and GenEO.

1.4 Linear Static Elasticity

The problems of structural mechanics, which shall be solved here, are originally formulated as initial boundary value problems, or more precisely, as second-order differential equations that are continuous in space and time. To make them accessible to a numerical solution in general and to domain decomposition methods in particular, they must be discretized in space and time. In the following, the way in which the discretized problems considered in this thesis were derived is outlined shortly. For a detailed description of the underlying theory, see [5–8]. Because linear static problems are sufficient to illustrate the most basic and important aspects of finite element tearing and interconnecting, the dynamic and the nonlinear parts are dropped as soon as possible. In this section, an overline is used to denote some continuous quantities like the vector field $\overline{\boldsymbol{u}}$ describing the displacement, because the same letters are needed later for other purposes.

For a given displacement $\overline{\boldsymbol{u}}$ the deformation gradient is defined as

$$\overline{\boldsymbol{F}} = \mathrm{Grad}\overline{\boldsymbol{u}} + \boldsymbol{I}\,,$$

which is used to compute the Green-Lagrange strain

$$\overline{\boldsymbol{E}} = \frac{1}{2}\left(\overline{\boldsymbol{F}}^T\overline{\boldsymbol{F}} - \boldsymbol{I}\right).$$

For small displacements, the Green-Lagrange strain converges to the infinitesimal or engineering strain, which is thus used in linear computations. The constitutive equation relates stresses with strains and defines the second Piola-Kirchhoff stress $\overline{\boldsymbol{S}}$ as

$$\overline{\boldsymbol{S}} = \frac{\partial\overline{W}}{\partial\overline{\boldsymbol{E}}}\,,$$

which is the derivative of the strain energy $\overline{W}$ with respect to the Green-Lagrange strain. The definition of the functional relation between the strain energy and the Green-Lagrange strain implies the choice of a particular constitutive law, for example, St. Venant or Mooney Rivlin, which describes the material's behavior by a set of parameters. For linear problems, St. Venant material is always used. With these quantities, the balance equation that has to be fulfilled at every point in space and time, expressed with respect to the initial configuration, reads

$$\mathrm{Div}\left(\overline{\boldsymbol{F}}\,\overline{\boldsymbol{S}}\right) + \rho_0\overline{\boldsymbol{b}} - \rho_0\ddot{\overline{\boldsymbol{u}}} = 0\,.$$

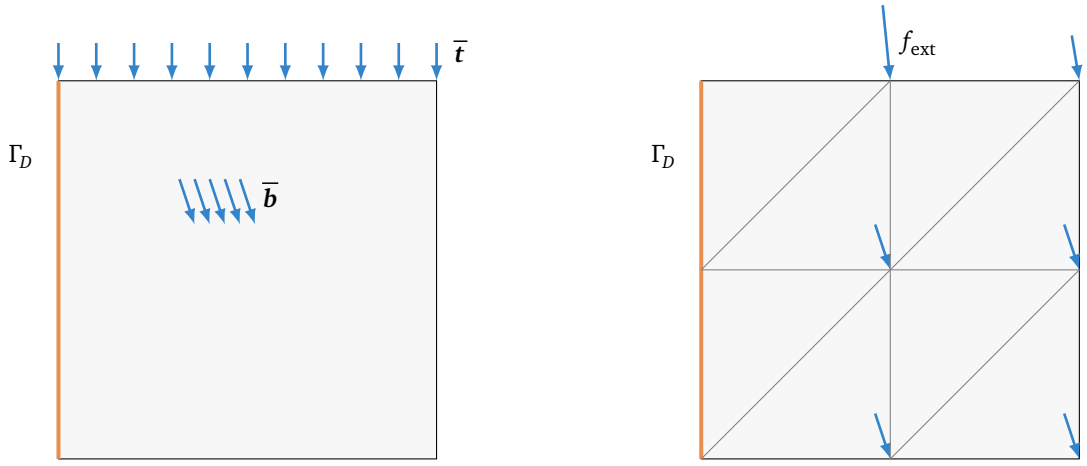


Figure 1.2. Left: Continuous structure in initial (undeformed) configuration, with body load $\bar{\boldsymbol{b}}$, traction forces $\bar{\boldsymbol{t}}$ and Dirichlet boundary Γ_D. Right: Discretized structure in initial (undeformed) configuration, with nodes, elements and the nodal external force $\boldsymbol{f}_{\text{ext}}$.

In the balance equation, $\bar{\boldsymbol{b}}$ denotes the body load, e.g., gravitational force, and ρ_0 the mass density in initial configuration. Figure 1.2 illustrates these quantities for an exemplary continuous structure. The boundary Γ of the structure is divided in a Dirichlet part Γ_D and a Neumann part Γ_N. Along those, the structure is subjected to the boundary conditions

$$\bar{\boldsymbol{u}}|_{\Gamma_D} = \bar{\boldsymbol{u}}_D \qquad \left(\bar{\boldsymbol{F}}\,\bar{\boldsymbol{S}}\,\bar{\boldsymbol{N}}\right)|_{\Gamma_N} = \bar{\boldsymbol{t}}\,,$$

using the surface normal in initial configuration $\bar{\boldsymbol{N}}$. For the rest of the section, the inertia term is dropped such that the static problem remains.

Trial and test functions are introduced. The trial functions

$$\bar{\boldsymbol{u}}_h \in \mathscr{U} \qquad \mathscr{U} = \left\{\bar{\boldsymbol{u}} \in H^1(\Omega),\ \bar{\boldsymbol{u}}|_{\Gamma_D} = \bar{\boldsymbol{u}}_D\right\}$$

approximate the exact solution and fulfill the Dirichlet boundary conditions. The test functions

$$\bar{\boldsymbol{v}} \in \mathscr{U}^0 \qquad \mathscr{U}^0 = \left\{\bar{\boldsymbol{v}} \in H^1(\Omega),\ \bar{\boldsymbol{v}}|_{\Gamma_D} = \boldsymbol{0}\right\}$$

vanish on the Dirichlet boundary and together they yield the weak form

$$G(\bar{\boldsymbol{u}}_h, \bar{\boldsymbol{v}}) = \int_\Omega \bar{\boldsymbol{S}}(\bar{\boldsymbol{u}}) : \delta\bar{\boldsymbol{E}}(\bar{\boldsymbol{u}}, \bar{\boldsymbol{v}})\,\mathrm{d}\Omega - \int_\Omega \bar{\boldsymbol{b}} \cdot \bar{\boldsymbol{v}}\,\mathrm{d}\Omega - \int_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \bar{\boldsymbol{v}}\,\mathrm{d}\Gamma = 0\,.$$

Here, $\delta\bar{\boldsymbol{E}}(\bar{\boldsymbol{u}}, \bar{\boldsymbol{v}})$ denotes the variation of the Green-Lagrange strain according to the test displacement $\bar{\boldsymbol{v}}$ and can also be expressed by a directional derivative as

$$\delta\bar{\boldsymbol{E}}(\bar{\boldsymbol{u}}, \bar{\boldsymbol{v}}) = \frac{\mathrm{d}}{\mathrm{d}\mu}\bar{\boldsymbol{E}}(\bar{\boldsymbol{u}} + \mu\bar{\boldsymbol{v}})\bigg|_{\mu=0}\,.$$

Discretization is obtained by meshing the domain and assembling the test and trial functions from shape functions $\boldsymbol{N}$ such that

$$\overline{\boldsymbol{u}}_h = \boldsymbol{N}\boldsymbol{u} \qquad \overline{\boldsymbol{v}} = \boldsymbol{N}\boldsymbol{v}\,.$$

The nodal internal reaction force $\boldsymbol{f}_{\text{int}}$ of the structure due to its displacement and the nodal external force $\boldsymbol{f}_{\text{ext}}$ are defined by

$$\boldsymbol{v}^T \boldsymbol{f}_{\text{int}}(\boldsymbol{u}) = \int_\Omega \overline{\boldsymbol{S}}(\boldsymbol{N}\boldsymbol{u}) : \delta\overline{\boldsymbol{E}}(\boldsymbol{N}\boldsymbol{u}, \boldsymbol{N}\boldsymbol{v})\,\mathrm{d}\Omega$$

$$\boldsymbol{v}^T \boldsymbol{f}_{\text{ext}} = \boldsymbol{v}^T \int_\Omega \boldsymbol{N}^T \overline{\boldsymbol{b}}\,\mathrm{d}\Omega + \boldsymbol{v}^T \int_{\Gamma_N} \boldsymbol{N}^T \overline{\boldsymbol{t}}\,\mathrm{d}\Gamma$$

resulting in the short form

$$\boldsymbol{v}^T \boldsymbol{f}_{\text{int}}(\boldsymbol{u}) - \boldsymbol{v}^T \boldsymbol{f}_{\text{ext}} = \boldsymbol{0}\,.$$

If the equation is restricted to degrees of freedom (dof) that are not subjected to Dirichlet boundary conditions, the test functions can be dropped and the discretized, nonlinear static equation

$$\boldsymbol{f}_{\text{int}}(\boldsymbol{u}) - \boldsymbol{f}_{\text{ext}} = \boldsymbol{0}$$

remains. The linearization of the weak form

$$\mathrm{D}G(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}) \cdot \Delta\overline{\boldsymbol{u}} = \boldsymbol{v}^T \mathrm{D}\boldsymbol{f}_{\text{int}}(\boldsymbol{u})\Delta\boldsymbol{u} = \boldsymbol{v}^T \boldsymbol{K}\Delta\boldsymbol{u}$$

leads to tangential matrix of the internal force $\boldsymbol{K}$, also called stiffness matrix. Again, the test functions may be dropped if only unconstrained dof are considered. This finally yields the linear system of equations

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}_{\text{ext}} \tag{1.1}$$

that has to be solved if a linear approximation $\boldsymbol{N}\boldsymbol{u}$ for the displacement field of the static problem is demanded. In the following, the nodal displacements $\boldsymbol{u}$ are simply called displacements.

Chapter 2

Finite Element Tearing and Interconnecting

Iterative domain decomposition methods date back at least to the Schwarz algorithm proposed by Hermann Schwarz in the year of 1869. Variants of it like the additive Schwarz method, the multiplicative Schwarz method or the alternating Schwarz method are still studied today and provide a prolific framework for developing concepts and enhancements of domain decomposition algorithms. Today, they aim at the parallelization and thus at improving the efficiency of the solution of boundary value problems for partial differential equations. The fundamental idea is to split the whole spatial domain into several subdomains like illustrated in the upper part of figure 2.1. While every part of the original domain Ω is assigned to at least one subdomain such that

$$\Omega = \Omega^{(1)} \cup \Omega^{(2)},$$

this mapping is not injective, i.e., the subdomains overlap in some region such that

$$\Omega^{(1)} \cap \Omega^{(2)} \neq \emptyset.$$

The boundary value problem is then solved on each individual subdomain and the solutions are improved iteratively until they match in the overlapping region.

The 1963 published paper [9] is often considered as the origin of substructuring, a different approach to domain decomposition. It is characterized by a non-overlapping decomposition of the domain like depicted in the lower part of figure 2.1. From the beginning, the single compartments were called substructures instead of subdomains. While the decomposed problem was originally solved by direct methods, the foundations of a new type of non-overlapping algorithms were developed in the 1980s, e.g., in [10–13]. They employ in particular the CG method [14] to solve the substructured problem, which leads to highly efficient iterative procedures.

Non-overlapping decompositions result in a new formulation in which the problem is restricted to the common interface that the substructures share. The substructures are thereby seen as kind of superelements that are described by Schur complements of their original, full operators. An additional condition appears in non-overlapping decompositions, the conservation of fluxes, which is the same as the

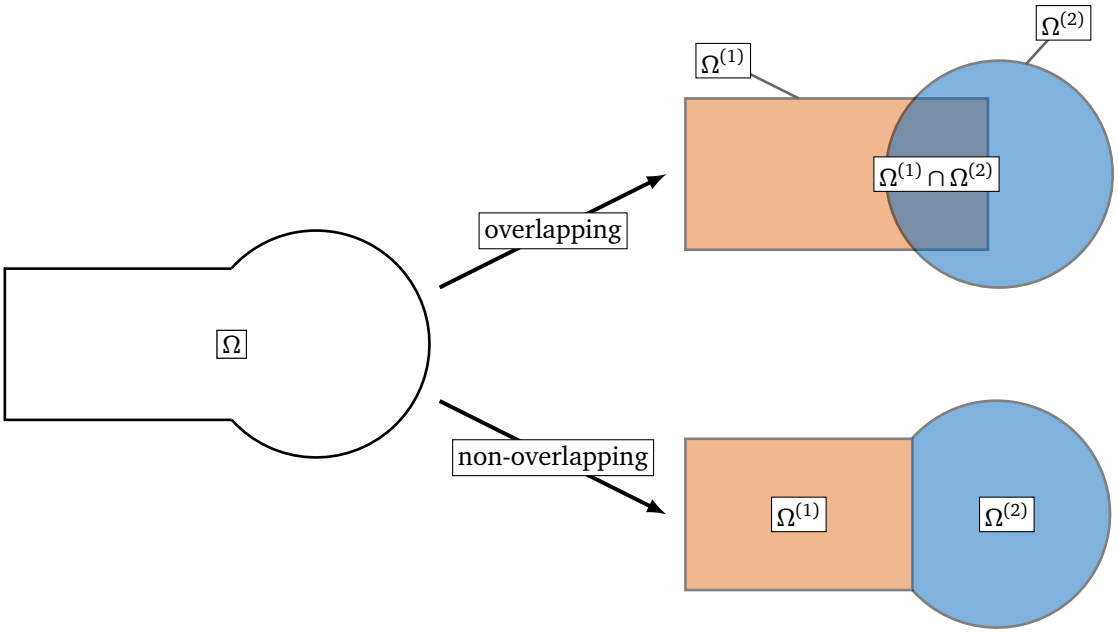


Figure 2.1. Overlapping and non-overlapping domain decomposition.

action-reaction principle in structural mechanics. The corresponding solution algorithm then searches iteratively for the displacement field on the interface, such that the connection forces will respect this additional transmission condition. If a solution of acceptable accuracy is found, the displacement field on the interior part of a subdomain can be recovered by solving a local problem.

Since that time, these two different basic approaches were classified and distinguished into overlapping Schwarz-type methods and non-overlapping Schur complement-type methods. The Schur complement-type methodology was further elaborated in terms of preconditioning and even already coarse spaces in a series of papers [15–18] before it was equipped with a rigid body mode coarse space for a specific preconditioner and denominated as balancing domain decomposition (BDD) in [19], which is the most widely used form today.

FETI was introduced 1991 in [20–22] and was soon identified as the dual counterpart of the BDD algorithm. Just as BDD, it operates on the interface using the Schur complements of the non-overlapping substructures. However, it formulates the problem not in terms of the original solution field, e.g., displacement, but in terms of the corresponding interface flux, e.g., connection forces. Different from BDD, the flux is chosen and adapted during the iterations such that it always fulfills the transmission conditions, and the solution is found when the original solution fields, e.g., displacement, of the substructures match on the interface. Accordingly, the original way of solving Schur complement-type decompositions like BDD is usually referred to as the primal approach to distinguish it from the FETI-like approach, the dual one. A further approach in which dual and primal assembly is mixed was published in [23] and is called FETI dual-primal (FETI-DP).

Today, a wide range of domain decomposition algorithms exists and they can still be classified in the two main groups. These are the overlapping Schwarz-type methods on the one hand, and non-overlapping Schur complement-type methods on the other hand as depicted in figure 2.1. Despite their apparent differences, both types were found to be equivalent in certain cases for specific preconditioners [24]. Profound surveys of domain decomposition methods, overlapping and non-overlapping, can be found in [25–29] and especially for non-overlapping methods in [30].

In this chapter, the non-overlapping domain decomposition of structural mechanics problems, discretized by the finite element method (FEM), is reviewed. The FETI family of methods, which is studied throughout this work, belongs to the non-overlapping classification and is presented subsequently in detail. At first, the dual mathematical formulation of the decomposed problem, on which FETI methods are based, is derived and illuminated. It is put in contrast to the older primal formulation, which gives rise to another large family of non-overlapping domain decomposition methods. After that, the iterative solution of the dual formulation is discussed. More precisely, only symmetric problems that are solvable by a CG method, are considered. Next, commonly used enhancements to accelerate the basic iterative solution algorithm are introduced. These are preconditioners, the according scaling techniques, and deflation spaces, which are also known as auxiliary coarse spaces. The chapter is closed by an analysis of the spectral properties of the presented method and by the statements about its convergence that can be made on this basis.

2.1 Non-overlapping Domain Decomposition

As it was already described and illustrated for the special case of two substructures, non-overlapping domain decomposition means in general that the spatial domain Ω, whose boundary is denoted by Γ, is decomposed into N_s non-overlapping subdomains $\Omega^{(s)}$, which are mostly referred to as substructures. Their union

$$\Omega = \bigcup_{s=1}^{N_s} \Omega^{(s)}$$

results again in the original domain. This process can be thought of as defining the future interfaces between the substructures as connected lines along which the structure is then sliced apart, resulting in the individual substructures. This global interface is denoted Γ_I and illustrated on the left of figure 2.2, which shows the example structure that is used throughout this section. Each substructure s touches a specific part of the global interface. This part is called the substructure's local interface $\Gamma_I^{(s)}$ and is a part of the substructure's entire boundary $\Gamma^{(s)}$. The local interfaces, illustrated on the right of figure 2.2 are defined by

$$\Gamma_I^{(s)} = \Gamma^{(s)} \setminus \Gamma \,.$$

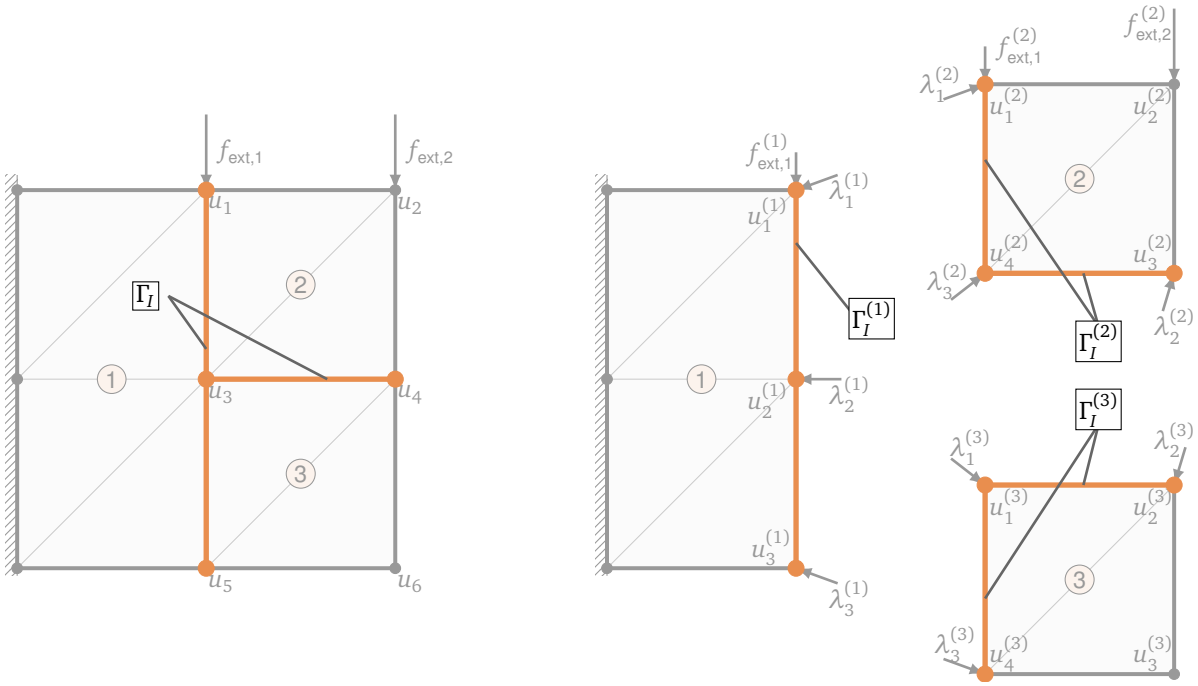


Figure 2.2. Left: The global interface Γ_I along which the global structure is sliced into non-overlapping substructures. Right: The torn apart substructures and their local interfaces $\Gamma^{(s)}$ along which they are connected to each other.

The part of the interface that is common to two substructures, i.e., the interface between two neighboring substructures, is defined by

$$\Gamma_I^{(s,r)} = \Gamma_I^{(s)} \cap \Gamma_I^{(r)} .$$

Furthermore, Dirichlet and Neumann boundaries can be determined for each substructure by

$$\Gamma_D^{(s)} = \Gamma^{(s)} \cap \Gamma_D$$
$$\Gamma_N^{(s)} = \Gamma^{(s)} \setminus \Gamma_D^{(s)} .$$

In general, it is possible to decompose the domain first and mesh the substructures afterwards independently, which results in non-conforming interfaces of the substructures. This means, that the nodes of two connected substructures do not necessarily coincide on their interface $\Gamma_I^{(s,r)}$. In this work, only conforming meshes are considered. Therefore, the decomposition or the substructuring can be done after the spatial domain was meshed and interfaces are then defined along the edges of the mesh.

2.1.1 Local Problems

The local interface $\Gamma_I^{(s)}$ of a substructure defines a set of dof, which is made up of the dof of all nodes located on the interface, also called the local interface space. This

set is denoted as

$$\boldsymbol{u}_I^{(s)} = \boldsymbol{u}^{(s)}|_{\Gamma_I^{(s)}},$$

which is the restriction of $\boldsymbol{u}^{(s)}$ to the interface $\Gamma_I^{(s)}$. The illustration on the right of figure 2.3 shows the set of local interface dof and their relation with the full set of local dof for the decomposed example structure. A local restriction or trace operator $\boldsymbol{t}^{(s)}$ can be defined, which realizes the restriction operation

$$\boldsymbol{u}_I^{(s)} = \boldsymbol{t}^{(s)}\boldsymbol{u}^{(s)}. \tag{2.1}$$

These local trace operators read in the case of the example structure

$$\boldsymbol{t}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{t}^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{t}^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

By cutting mechanical systems apart into substructures, internal forces become external forces that act on the newly created substructure interfaces. If the local forces $\boldsymbol{f}_{\text{ext}}^{(s)}$ are defined to represent only the original, global external force $\boldsymbol{f}_{\text{ext}}$, an additional term has to be added on the right-hand side of the local balance equation such that it reads for the substructure s

$$\boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)} = \boldsymbol{f}_{\text{ext}}^{(s)} - \boldsymbol{t}^{(s)^T}\boldsymbol{\lambda}^{(s)}.$$

Here, the restriction operator $\boldsymbol{t}^{(s)}$ is used reversely by transposition as expansion operator, which maps a quantity that is defined solely on the interface, like the connection forces, to the full local space of dof of a substructure. As the connection forces can only act on the interface, they are defined on the local interface space as $\boldsymbol{\lambda}^{(s)}$ and are expanded to the complete local space. In the resulting force vector, the forces on non-interface dof are set to zero by the operator $\boldsymbol{t}^{(s)^T}$.

Assembly of the local problems. Similar to the assembly process in the finite element method, strictly Boolean assembly matrices $\boldsymbol{A}^{(s)}$ can be defined that realize a mapping from local to global dof. In transposed form $\boldsymbol{A}^{(s)^T}$, they realize a reverse mapping from global to local. This way, local force contributions $\boldsymbol{f}^{(s)}$ can be assembled to a global force $\boldsymbol{f}$ by

$$\boldsymbol{f} = \sum_{s=1}^{N_s} \boldsymbol{A}^{(s)}\boldsymbol{f}^{(s)}. \tag{2.2}$$

Reversely, local displacements $\boldsymbol{u}^{(s)}$ can be obtained by the restriction of the global displacement $\boldsymbol{u}$ by

$$\boldsymbol{u}^{(s)} = \boldsymbol{A}^{(s)^T}\boldsymbol{u}. \tag{2.3}$$

The assembly process that is illustrated in equation (2.2) can be applied to the local balance equation. This leads to the assembly of the left-hand side and the right-hand side, which represent the internal and the external force, respectively. If additionally

the local displacements are extracted from the global displacement by the transposed assembly matrices, like it was done in equation (2.3), the assembly of the balance equations reads

$$\sum_{s=1}^{N_s} \boldsymbol{A}^{(s)} \boldsymbol{K}^{(s)} \boldsymbol{A}^{(s)^T} \boldsymbol{u} = \sum_{s=1}^{N_s} \boldsymbol{A}^{(s)} \boldsymbol{f}_{\text{ext}}^{(s)} - \sum_{s=1}^{N_s} \boldsymbol{A}^{(s)} \boldsymbol{t}^{(s)^T} \boldsymbol{\lambda}^{(s)} .$$

If the system of local balance equations shall represent the original global system, the global stiffness is identified as

$$\boldsymbol{K} = \sum_{s=1}^{N_s} \boldsymbol{A}^{(s)} \boldsymbol{K}^{(s)} \boldsymbol{A}^{(s)^T} .$$

In this manner, the substructures can also be considered as a kind of superelements. A block matrix notation that defines block matrices of diagonal, vertical, and horizontal concatenation is used in the following and is introduced here using the examples of the local stiffnesses $\boldsymbol{K}^{(s)}$, the assembly matrices $\boldsymbol{A}^{(s)}$, and the local displacements $\boldsymbol{u}^{(s)}$. Their block versions are denoted by

$$\boldsymbol{K}^{\boxtimes} = \begin{bmatrix} \boldsymbol{K}^{(1)} & & & \\ & \boldsymbol{K}^{(2)} & & \\ & & \ddots & \\ & & & \boldsymbol{K}^{(N_s)} \end{bmatrix} \qquad \boldsymbol{u}^{\boxplus} = \begin{bmatrix} \boldsymbol{u}^{(1)} \\ \boldsymbol{u}^{(2)} \\ \vdots \\ \boldsymbol{u}^{(N_s)} \end{bmatrix}$$

$$\boldsymbol{A}^{\boxminus} = \begin{bmatrix} \boldsymbol{A}^{(1)} & \boldsymbol{A}^{(2)} & \cdots & \boldsymbol{A}^{(N_s)} \end{bmatrix},$$

which allow for instance to imply the summation over all substructures by matrix multiplication like in

$$\boldsymbol{K} = \boldsymbol{A}^{\boxminus} \, \boldsymbol{K}^{\boxtimes} \, \boldsymbol{A}^{\boxminus T} .$$

Action-reaction principle. According to equation (2.2), the assembly of the right-hand sides of the local balance equations must yield the global external force

$$\boldsymbol{f}_{\text{ext}} = \sum_{s=1}^{N_s} \boldsymbol{A}^{(s)} \boldsymbol{f}_{\text{ext}}^{(s)} - \sum_{s=1}^{N_s} \boldsymbol{A}^{(s)} \boldsymbol{t}^{(s)^T} \boldsymbol{\lambda}^{(s)} . \tag{2.4}$$

Simultaneously, the local contributions $\boldsymbol{f}_{\text{ext}}^{(s)}$ are defined to represent exclusively the original global force $\boldsymbol{f}_{\text{ext}}$, i.e.,

$$\boldsymbol{f}_{\text{ext}} = \sum_{s=1}^{N_s} \boldsymbol{A}^{(s)} \boldsymbol{f}_{\text{ext}}^{(s)} , \tag{2.5}$$

and must not include any of the connection forces acting between the substructures. Those are represented by $\boldsymbol{\lambda}^{(s)}$ and by combining equation (2.4) and equation (2.5), the condition

$$\sum_{s=1}^{N_s} \boldsymbol{A}^{(s)} \boldsymbol{t}^{(s)^T} \boldsymbol{\lambda}^{(s)} = \boldsymbol{0} \tag{2.6}$$

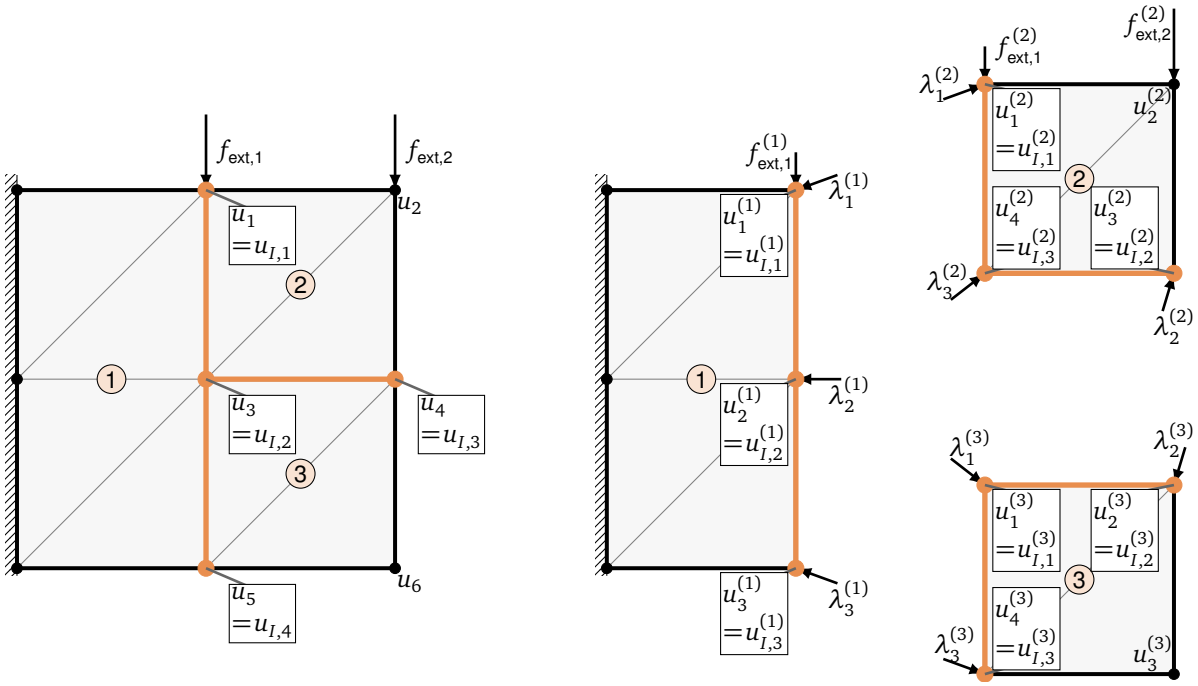


Figure 2.3. Left: The global interface space, established by the subset of dof that belong to nodes situated on the global interface. Right: The local interface spaces, established by the subsets of dof that belong to nodes situated on the local interfaces.

for the connection forces $\boldsymbol{\lambda}^{(s)}$ is obtained, which is exactly the action-reaction principle in mechanics. More general, this condition ensures the conservation of fluxes, i.e., the surface normal derivatives of the solution on the interface.

Continuity of the displacement field. A global displacement $\boldsymbol{u}$ that fulfills equation (2.3) for each substructure can only exist if all pairs of neighboring substructures s and r exhibit identical solutions on their common interface such that

$$\boldsymbol{u}^{(s)}|_{\Gamma^{(s,r)}} = \boldsymbol{u}^{(r)}|_{\Gamma^{(s,r)}}$$

This condition ensures the continuity of the solution across substructure interfaces, which is the continuity of displacement in the case of structural mechanics. For the discretized problem, this means that local dof that correspond to the same global dof must be equal, i.e., for the example in figure 2.3

$$\begin{aligned} u^{(1)}_1 &= u^{(2)}_1 \qquad u^{(1)}_2 = u^{(2)}_4 \qquad u^{(1)}_2 = u^{(3)}_1 \\ u^{(2)}_4 &= u^{(3)}_1 \qquad u^{(1)}_3 = u^{(3)}_4 \qquad u^{(2)}_3 = u^{(3)}_2 . \end{aligned} \tag{2.7}$$

These conditions can be expressed by defining signed Boolean operators $\boldsymbol{E}^{(s)}$, i.e., composed by 1, −1, and 0, for each substructure. For the example in figure 2.3, they

read

$$\boldsymbol{E}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{E}^{(2)} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \boldsymbol{E}^{(3)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 \end{bmatrix}. \quad (2.8)$$

The conditions of equation (2.7) can then be expressed as

$$\sum_{s=1}^{N_s} \boldsymbol{E}^{(s)} \boldsymbol{u}^{(s)} = \boldsymbol{0}. \quad (2.9)$$

A more general definition of the operators $\boldsymbol{E}^{(s)}$ can be given by using the block notations $\boldsymbol{E}^{\boxminus}$ and $\boldsymbol{u}^{\boxplus}$. The condition

$$\boldsymbol{E}^{\boxminus}\, \boldsymbol{u}^{\boxplus} = \boldsymbol{0} \quad (2.10)$$

ensures displacement continuity if $\boldsymbol{E}^{\boxminus}$ fulfills

$$\operatorname{range}\left(\boldsymbol{E}^{\boxminus\, T}\right) = \ker\left(\boldsymbol{A}^{\boxminus}\right). \quad (2.11)$$

In other words, any set of $\boldsymbol{E}^{(s)}$ that fulfills equation (2.11) is suitable to formulate the displacement continuity condition as written in equation (2.9) or in block formulation in equation (2.10). For example, from the signed Boolean operators $\boldsymbol{E}^{(s)}$ that were defined in equation (2.8), a new set of suitable $\boldsymbol{E}^{(s)*}$ could be created by $\boldsymbol{E}^{(s)*} = \boldsymbol{Q}\boldsymbol{E}^{(s)}$, where $\boldsymbol{Q}$ is any symmetric positive definite matrix. Every row of $\boldsymbol{E}^{(s)*}$ is then a linear combination of the rows of $\boldsymbol{E}^{(s)}$, and $\boldsymbol{E}^{(s)*}$ are not Boolean anymore. However, a natural choice to begin with is to connect all possible pairs of spatially coincident nodes, like it was done in equation (2.7), which leads to signed Boolean operators $\boldsymbol{E}^{(s)}$ like in equation (2.8). With either choice for $\boldsymbol{E}^{(s)}$, exact displacement continuity is enforced at the nodes. This is not necessarily the case for other parts of the interface, i.e., along the common edge of two connected elements. Displacement continuity is fulfilled exactly along the whole interface if the same element formulations and the same shape functions are used on all connected elements.

A local formulation of the global problem. The decomposition of the global structure into N_s non-overlapping substructures results in the following system of equations that is equivalent to the original global problem $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}_{\text{ext}}$.

$$\boldsymbol{K}^{(s)} \boldsymbol{u}^{(s)} = \boldsymbol{f}_{\text{ext}}^{(s)} - \boldsymbol{t}^{(s)^T} \boldsymbol{\lambda}^{(s)} \qquad s = 1, \ldots, N_s \quad (2.12)$$

$$\sum_{s=1}^{N_s} \boldsymbol{A}^{(s)} \boldsymbol{t}^{(s)^T} \boldsymbol{\lambda}^{(s)} = \boldsymbol{0} \quad (2.13)$$

$$\sum_{s=1}^{N_s} \boldsymbol{E}^{(s)} \boldsymbol{u}^{(s)} = \boldsymbol{0} \quad (2.14)$$

The system of equations comprises the balance equations (2.12), the action-reaction principle in equation (2.13), and the displacement continuity in equation (2.14).

2.1.2 Condensation on the Interface

Just like it was done for the local interfaces, a global interface space is established by the dof of the global system that belong to nodes on the global interface Γ_I. The illustration on the left of figure 2.3 shows the set of global interface dof and their relation with the full set of global dof for the example structure. Global interface dof are denoted by the subscript I, i.e., the corresponding displacements by $\boldsymbol{u}_I$. The remaining dof are denoted by the subscript r. Again, this gives rise to a restriction operator $\boldsymbol{t}$, enabling the mapping operation

$$\boldsymbol{u}_I = \boldsymbol{t}\,\boldsymbol{u}$$

Condensation of the global problem. A linear problem like $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}_{\text{ext}}$ can be condensed on an arbitrary subset of dof. That is to say, one can define an operator that establishes an input-output relation, e.g., between forces and displacements, solely on these dof. First, the original problem is split into coupled equations by splitting $\boldsymbol{K}$ into four parts. For example, $\boldsymbol{K}_{rI}$ is the restriction of $\boldsymbol{K}$ to rows corresponding to r dof and columns corresponding to I dof. The resulting equations, equivalent to $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}_{\text{ext}}$, then read

$$\begin{aligned}\boldsymbol{K}_{rr}\boldsymbol{u}_r + \boldsymbol{K}_{rI}\boldsymbol{u}_I &= \boldsymbol{f}_{\text{ext},r}\\ \boldsymbol{K}_{Ir}\boldsymbol{u}_r + \boldsymbol{K}_{II}\boldsymbol{u}_I &= \boldsymbol{f}_{\text{ext},I}\,.\end{aligned}$$

They enable to eliminate the $\boldsymbol{u}_r$ displacements, which results in the equation

$$\left(\boldsymbol{K}_{II} - \boldsymbol{K}_{Ir}\boldsymbol{K}_{rr}^{-1}\boldsymbol{K}_{rI}\right)\boldsymbol{u}_I = \boldsymbol{f}_{\text{ext},I} - \boldsymbol{K}_{Ir}\boldsymbol{K}_{rr}^{-1}\boldsymbol{f}_{\text{ext},r}\,.$$

The operator on the left-hand side is then denoted by

$$\boldsymbol{S}_K = \boldsymbol{K}_{II} - \boldsymbol{K}_{Ir}\boldsymbol{K}_{rr}^{-1}\boldsymbol{K}_{rI}$$

and represents the condensation of the global stiffness $\boldsymbol{K}$ on the global interface space I. It is mostly called the Schur complement of $\boldsymbol{K}$.

The right-hand side is denoted by

$$\hat{\boldsymbol{f}}_{\text{ext},I} = \boldsymbol{f}_{\text{ext},I} - \boldsymbol{K}_{Ir}\boldsymbol{K}_{rr}^{-1}\boldsymbol{f}_{\text{ext},r}$$

and is called the external force condensed on the interface. It represents a force on the interface that is, regarding the displacements on the interface, equivalent to the original external force that acts on arbitrary dof, including those in the r-set.

After all, the global problem can be expressed only in terms of global interface dof, reading

$$\boldsymbol{S}_K\boldsymbol{u}_I = \hat{\boldsymbol{f}}_{\text{ext},I} \tag{2.15}$$

Condensation of the local problems. As it was done for the global problem, the local problems are also split into interface dof, denoted by the subscript I, and the remaining dof, denoted by the subscript r. In the context of a substructure, I always refers to the *local* interface part and r always refers to the non-interface part of that substructure. A further differentiation like denoting by I the global interface and by $I^{(s)}$ the local interface of substructure s is omitted in favor of better readability. The mapping operation from the full local substructure space to the local interface space I was already defined by the trace or restriction operator $\boldsymbol{t}^{(s)}$ in equation (2.1). Especially in the case of the local substructure problems, it is noteworthy to mention that the r-set can include inner dof as well as boundary dof of a substructure because not the entire boundary of a substructure is also necessarily an interface. The location of the local interfaces and the location and numbering of the local interface dof is shown on the right of figure 2.3.

The condensation of the local stiffness $\boldsymbol{K}^{(s)}$ on the local interface is defined analogously to the global Schur complement as

$$\boldsymbol{S}_K^{(s)} = \boldsymbol{K}_{II}^{(s)} - \boldsymbol{K}_{Ir}^{(s)} \boldsymbol{K}_{rr}^{(s)^{-1}} \boldsymbol{K}_{rI}^{(s)} ,$$

and the condensation of the external force is defined as

$$\hat{\boldsymbol{f}}_{\text{ext},I}^{(s)} = \boldsymbol{f}_{\text{ext},I}^{(s)} - \boldsymbol{K}_{Ir}^{(s)} \boldsymbol{K}_{rr}^{(s)^{-1}} \boldsymbol{f}_{\text{ext},r}^{(s)} .$$

They allow to formulate the local problems solely on their interface. The condensed local problem reads

$$\boldsymbol{S}_K^{(s)} \boldsymbol{u}_I^{(s)} = \hat{\boldsymbol{f}}_{\text{ext},I}^{(s)} - \boldsymbol{\lambda}^{(s)} .$$

In these condensed local problems, the connection forces $\boldsymbol{\lambda}^{(s)}$ no longer need to be expanded by the transposed restriction operators $\boldsymbol{t}^{(s)^T}$ as they are already defined on the local interface space I, which is also illustrated in figure 2.3.

Assembly of the condensed local problems. Analogously to the assembly operators $\boldsymbol{A}^{(s)}$ for the non-condensed system, the condensed local problems can be assembled to the global condensed problem. These condensed assembly operators are defined by

$$\boldsymbol{L}^{(s)} = \boldsymbol{t}\boldsymbol{A}^{(s)} \boldsymbol{t}^{(s)^T} .$$

The condensed global left-hand side and right-hand side are then recovered by assembly as

$$\boldsymbol{S}_K = \sum_{s=1}^{N_s} \boldsymbol{L}^{(s)} \boldsymbol{S}_K^{(s)} \boldsymbol{L}^{(s)^T} = \boldsymbol{L}^{\boxminus} \, \boldsymbol{S}_K^{\boxtimes} \, \boldsymbol{L}^{\boxminus^T} \qquad \hat{\boldsymbol{f}}_{\text{ext},I} = \sum_{s=1}^{N_s} \boldsymbol{L}^{(s)} \hat{\boldsymbol{f}}_{\text{ext},I}^{(s)} = \boldsymbol{L}^{\boxminus} \, \hat{\boldsymbol{f}}_{\text{ext},I}^{\boxplus} .$$

Action-reaction principle. From the action-reaction principle condition in equation (2.13), the condensed form of the condition directly follows as

$$\sum_{s=1}^{N_s} \boldsymbol{L}^{(s)} \boldsymbol{\lambda}^{(s)} = \boldsymbol{0} . \tag{2.16}$$

It enforces the action-reaction principle of the connection forces $\boldsymbol{\lambda}^{(s)}$ without going the indirect way of expanding them to the full space of substructure dof.

Continuity of the displacement field. Similar like the action-reaction principle only includes quantities on the interface, only the displacements of interface dof have to be taken into account in the conditions, that ensure the continuity of displacement. In other words, in

$$\boldsymbol{B}^{(s)} = \boldsymbol{E}^{(s)} \boldsymbol{t}^{(s)^T}$$

the zero columns are removed from the $\boldsymbol{E}^{(s)}$ matrices by multiplication with $\boldsymbol{t}^{(s)^T}$. The resulting $\boldsymbol{B}^{(s)}$ matrices can be used subsequently to formulate the displacement continuity condition solely on the space of the interface. The condition then reads

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}_I^{(s)} = \boldsymbol{0}\,. \tag{2.17}$$

The same relation that was set up between $\boldsymbol{A}^{(s)}$ and $\boldsymbol{E}^{(s)}$ also holds for $\boldsymbol{L}^{(s)}$ and $\boldsymbol{B}^{(s)}$, i.e., any matrix $\boldsymbol{B}^{\boxminus}$ fulfilling

$$\operatorname{range}\left(\boldsymbol{B}^{\boxminus\,T}\right) = \ker\left(\boldsymbol{L}^{\boxminus}\right) \tag{2.18}$$

such that

$$\boldsymbol{L}^{\boxminus}\, \boldsymbol{B}^{\boxminus\,T} = \sum_{s=1}^{N_s} \boldsymbol{L}^{(s)} \boldsymbol{B}^{(s)^T} = \boldsymbol{0}$$

holds can be used to formulate the continuity of the displacement. The $\boldsymbol{B}^{(s)}$ matrices that connect each pair of coinciding interface nodes in the example of figure 2.3 read

$$\boldsymbol{B}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{B}^{(2)} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \boldsymbol{B}^{(3)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

The compatibility condition from equation (2.17) reads in this case

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}_I^{(s)} = \begin{bmatrix} u_{I,1}^{(1)} - u_{I,1}^{(2)} \\ u_{I,2}^{(1)} - u_{I,3}^{(2)} \\ u_{I,2}^{(1)} - u_{I,1}^{(3)} \\ u_{I,3}^{(2)} - u_{I,1}^{(3)} \\ u_{I,3}^{(1)} - u_{I,3}^{(3)} \\ u_{I,2}^{(2)} - u_{I,2}^{(3)} \end{bmatrix} = \boldsymbol{0}\,.$$

A condensed local formulation of the global problem. The former results can be summarized to the system of equations

$$\boldsymbol{S}_K^{(s)} \boldsymbol{u}_I^{(s)} = \hat{\boldsymbol{f}}_{\text{ext},I}^{(s)} - \boldsymbol{\lambda}^{(s)} \qquad s = 1, \ldots, N_s \tag{2.19}$$

$$\sum_{s=1}^{N_s} \boldsymbol{L}^{(s)} \boldsymbol{\lambda}^{(s)} = \boldsymbol{0} \tag{2.20}$$

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}_I^{(s)} = \boldsymbol{0} \tag{2.21}$$

This system of equations with its two conditions is equivalent to the original global problem $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}_{\text{ext}}$ and operates solely on the interface dof. The local balance equations (2.19) are furthermore completely uncoupled and the coupling is established exclusively by the conditions of equation (2.20) and equation (2.21).

Equations (2.19) to (2.21) are the point of departure for almost any non-overlapping domain decomposition method. Most of these methods follow one of two approaches, which are are presented in the following subsections. Either they construct the primal interface problem or the dual interface problem. Only the dual approach is considered as object of research in this work. Nevertheless, a short description of the primal one is given first since it is very helpful to understand the terminology and the advantages of the dual approach.

2.1.3 Primal Interface Problem

The fundamental idea in the concept of primal Schur complement-type domain decomposition methods is to operate only on a unique global vector of the interface displacement $\boldsymbol{u}_I$ and derive all local interface displacements $\boldsymbol{u}_I^{(s)}$ from it. This is done using the assembly matrices $\boldsymbol{L}^{(s)}$ of the condensed local problems as mapping from the global interface to the local interfaces. For a given global interface displacement $\boldsymbol{u}_I$ local interface displacements $\boldsymbol{u}_I^{(s)}$ are computed as

$$\boldsymbol{u}_I^{(s)} = \boldsymbol{L}^{(s)^T} \boldsymbol{u}_I \tag{2.22}$$

In this case, the continuity of displacement $\boldsymbol{B}^{\boxminus}\, \boldsymbol{u}_I^{\boxplus} = \boldsymbol{0}$ is always fulfilled because of $\boldsymbol{B}^{\boxminus}\, \boldsymbol{L}^{\boxminus^T} = \boldsymbol{0}$ and the condition is eliminated. The resulting system of equations reads

$$\boldsymbol{S}_K^{(s)} \boldsymbol{L}^{(s)^T} \boldsymbol{u}_I = \hat{\boldsymbol{f}}_{\text{ext},I}^{(s)} - \boldsymbol{\lambda}^{(s)} \qquad s = 1, \ldots, N_s \tag{2.23}$$

$$\sum_{s=1}^{N_s} \boldsymbol{L}^{(s)} \boldsymbol{\lambda}^{(s)} = \boldsymbol{0} \tag{2.24}$$

Solving the local balance equations (2.23) for the connection forces $\boldsymbol{\lambda}^{(s)}$ and injecting the results in the remaining action-reaction principle condition of equation (2.24) yields

$$\underbrace{\sum_{s=1}^{N_s} \boldsymbol{L}^{(s)} \hat{\boldsymbol{f}}_{\text{ext},I}^{(s)}}_{\hat{\boldsymbol{f}}_{\text{ext},I}} - \underbrace{\sum_{s=1}^{N_s} \boldsymbol{L}^{(s)} \boldsymbol{S}_K^{(s)} \boldsymbol{L}^{(s)^T}}_{\boldsymbol{S}_K} \boldsymbol{u}_I = \boldsymbol{0}$$

and in short

$$\boldsymbol{S}_K \boldsymbol{u}_I = \hat{\boldsymbol{f}}_{\text{ext},I} \tag{2.25}$$

These two versions of the problem, i.e., the substructured one of equations (2.23) and (2.24) and the assembled one of equation (2.25), are called *primal interface problem*. Equation (2.25) is identified to be identical to the already discussed condensed global problem of equation (2.15). In this context, the $\boldsymbol{L}^{(s)}$ matrices are denominated *primal assembly matrices*.

Iterative primal Schur complement domain decomposition methods search for an interface displacement $\boldsymbol{u}_I$ for which the primal interface problem equation (2.25), which is actually the action-reaction principle condition, is fulfilled. On the left of figure 2.4, the primal interface problem is depicted. Taking up the illustrations in [30], it emphasizes that the displacement $\boldsymbol{u}_I$ represents the interface through which the substructures are connected. Accordingly, the continuity of displacement always holds during the iterations in primal Schur complement domain decomposition methods, while the action-reaction principle condition does not until convergence. Instead, the error of the action-reaction principle condition is the residual that is minimized throughout the iterations.

From here on, the space of $\boldsymbol{u}_I$ as well as $\boldsymbol{u}_I$ itself is referred to as *primal interface*.

2.1.4 Dual Interface Problem

The fundamental idea in the concept of dual Schur complement-type domain decomposition methods is to operate only on a unique global vector of connection forces $\boldsymbol{\lambda}$, which is then mapped to the local connection forces $\boldsymbol{\lambda}^{(s)}$ by

$$\boldsymbol{\lambda}^{(s)} = \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda} \qquad \text{or equivalently} \qquad \boldsymbol{\lambda}^{\square} = \boldsymbol{B}^{\boxminus^T} \boldsymbol{\lambda}\,. \tag{2.26}$$

Each element in $\boldsymbol{\lambda}$ corresponds to one line in $\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)} = \boldsymbol{0}$. In other words, for each pair of dof of coinciding nodes that are connected by a condition, one unique connection force is defined. By equation (2.26), this connection force is put with different signs on the two connected dof, such that the action-reaction principle is respected by design.

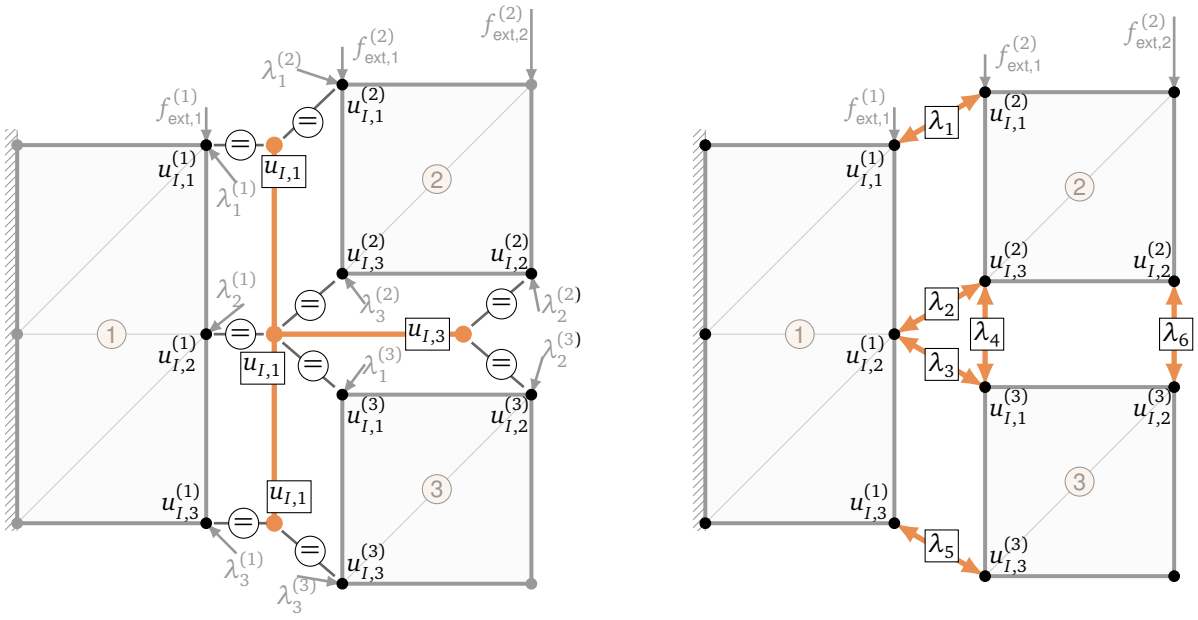


Figure 2.4. Left: The primal interface. Local interface displacements are derived from a unique global interface displacement and thus fulfill the continuity condition by definition. Right: The dual interface. Local connection forces are derived from a unique global connection force and thus fulfill the action-reaction principle by definition.

More general, by selecting

$$\boldsymbol{\lambda}^{\square} \in \operatorname{range}\left(\boldsymbol{B}^{\boxminus T}\right)$$

the action-reaction principle condition $\boldsymbol{L}^{\boxminus}\,\boldsymbol{\lambda}^{\square} = \boldsymbol{0}$ will always be fulfilled because of $\boldsymbol{L}^{\boxminus}\,\boldsymbol{B}^{\boxminus T} = \boldsymbol{0}$. Analogously to the elimination of the displacement continuity condition in the primal interface problem by introducing the *primal interface* $\boldsymbol{u}_I$, it is the action-reaction principle condition that is eliminated in the dual interface problem by introducing the *dual interface* $\boldsymbol{\lambda}$. An illustration of the dual interface for the example structure is depicted on the right of figure 2.4. According to the fact that all possible pairs of spatially coincident nodes were connected in the continuity of displacement conditions, all those possible pairs are also connected by a connection force. In the case of more than two spatially coinciding nodes, like at the crosspoint of the example structure where all three substructures touch, redundancy can be introduced in the continuity of displacement conditions as well as in the connection forces. These redundancies are discussed in section 2.1.5.

The resulting system of equations reads

$$\boldsymbol{S}_K^{(s)}\boldsymbol{u}_I^{(s)} = \hat{\boldsymbol{f}}_{\text{ext},I}^{(s)} - \boldsymbol{B}^{(s)^T}\boldsymbol{\lambda} \qquad s = 1,\dots,N_s \tag{2.27}$$

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)}\boldsymbol{u}_I^{(s)} = \boldsymbol{0} \tag{2.28}$$

The vector of connection forces $\boldsymbol{\lambda}$ can as well be interpreted as vector of Lagrange multipliers that enforce the compatibility conditions, i.e., they make sure that the displacement field of connected substructures is continuous across their interface.

Floating substructures. An important aspect that arises especially in dual methods[1] is the fact that substructures possess zero-energy modes if they lack of enough Dirichlet boundary conditions. In static problems of structural mechanics, these are rigid body modes and the affected substructures are said to be floating. In this case, their stiffness $\boldsymbol{K}^{(s)}$ as well as their condensed stiffness $\boldsymbol{S}_K^{(s)}$ are singular and they feature a core or null space denoted by

$$\boldsymbol{R}^{(s)} = \ker\left(\boldsymbol{K}^{(s)}\right) \qquad \text{and} \qquad \boldsymbol{R}_I^{(s)} = \ker\left(\boldsymbol{S}_K^{(s)}\right) = \boldsymbol{t}^{(s)}\boldsymbol{R}^{(s)}\,.$$

The number of rigid body modes per substructure and the total number of rigid body modes is denoted by

$$n_{\text{rbm}}^{(s)} = \operatorname{rank}\left(\boldsymbol{R}^{(s)}\right) \qquad \text{and} \qquad N_{\text{rbm}} = \sum_{s=1}^{N_s} n_{\text{rbm}}^{(s)}$$

respectively. The implications of this are two-fold.

On the on hand, when searching for connection forces $\boldsymbol{\lambda}$ that solve equations (2.27) and (2.28), the right-hand side of equation (2.27) must lie inside the range of the corresponding local operator, i.e.,

$$\left(\hat{\boldsymbol{f}}_{\text{ext},I}^{(s)} - \boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}\right) \in \operatorname{range}\left(\boldsymbol{S}_K^{(s)}\right). \tag{2.29}$$

According to the fundamental theorem of linear algebra [31]

$$\operatorname{range}\left(\boldsymbol{S}_K^{(s)}\right) = \ker\left(\boldsymbol{S}_K^{(s)^T}\right)^{\perp} = \ker\left(\boldsymbol{S}_K^{(s)}\right)^{\perp}$$

and thus equation (2.29) can be expressed as the condition

$$\boldsymbol{R}_I^{(s)^T}\left(\hat{\boldsymbol{f}}_{\text{ext},I}^{(s)} - \boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}\right) = \boldsymbol{0}\,. \tag{2.30}$$

In structural mechanics, this condition corresponds to the requirement of floating substructures to be self-equilibrated, at least in the direction of their zero-energy modes. Therefore, equation (2.30) is referred to as self-equilibrium condition.

[1] It is also relevant in primal methods but not before certain preconditioners are applied.

On the other hand, when solving the local balance equations (2.27) for $\boldsymbol{u}_I^{(s)}$ to evaluate the dual interface error of equation (2.28), a generalized inverse of $\boldsymbol{S}_K^{(s)}$ or $\boldsymbol{K}^{(s)}$ must be introduced. These generalized inverses are denoted by $\boldsymbol{S}_K^{(s)^+}$ and $\boldsymbol{K}^{(s)^+}$, for which

$$\boldsymbol{S}_K^{(s)^+} = \boldsymbol{t}^{(s)} \boldsymbol{K}^{(s)^+} \boldsymbol{t}^{(s)^T}$$

holds. However, this also means that the right-hand side, i.e., the force acting on the substructure, only determines a solution $\boldsymbol{u}_I^{(s)}$ up to an additive element from $\ker\left(\boldsymbol{S}_K^{(s)}\right)$. Accordingly, the full solution reads

$$\boldsymbol{u}_I^{(s)} = \boldsymbol{S}_K^{(s)^+} \left(\hat{\boldsymbol{f}}_{\text{ext},I}^{(s)} - \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda} \right) + \boldsymbol{R}_I^{(s)} \boldsymbol{\alpha}_R^{(s)} \tag{2.31}$$

where $\boldsymbol{\alpha}_R^{(s)}$ are the amplitudes of the zero-energy modes, which are represented by the columns of $\boldsymbol{R}_I^{(s)}$. It remains to be noted that indeed the generalized inverse is not uniquely defined, but the complete solution for $\boldsymbol{u}_I^{(s)}$ is independent from a specific choice. This holds for all dual methods. The practical computation of the null space and solving singular problems of this type is discussed in [32] and more generally in [33].

Dual assembly. In the *primal* as well as in the *dual* interface problem, it is the remaining interface condition that is evaluated. In the *primal* interface problem, this is the action-reaction principle, which is evaluated by solving the balance equations for the connection forces $\boldsymbol{\lambda}^{(s)}$. In the *dual* interface problem, this is the continuity of displacement, which is evaluated by solving the balance equations for the interface displacements $\boldsymbol{u}_I^{(s)}$. As a consequence, dual assembly is attained by injecting the solved $\boldsymbol{u}_I^{(s)}$ of equation (2.31) into the continuity of displacement condition of equation (2.28). The result reads

$$\sum_{s=1}^{N_s} \left(\boldsymbol{B}^{(s)} \boldsymbol{S}_K^{(s)^+} \hat{\boldsymbol{f}}_{\text{ext},I}^{(s)} \right) - \left(\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{S}_K^{(s)^+} \boldsymbol{B}^{(s)^T} \right) \boldsymbol{\lambda} + \sum_{s=1}^{N_s} \left(\boldsymbol{B}^{(s)} \boldsymbol{R}_I^{(s)} \boldsymbol{\alpha}_R^{(s)} \right) = \boldsymbol{0} . \tag{2.32}$$

Similar to the primal interface problem where the local Schur complements were assembled to the global Schur complement by the primal assembly matrices $\boldsymbol{L}^{(s)}$, the generalized inverses of the local Schur complements are assembled to a global quantity by the $\boldsymbol{B}^{(s)}$ matrices. Accordingly, the $\boldsymbol{B}^{(s)}$ matrices are also called *dual assembly matrices*. This global quantity is denoted by

$$\boldsymbol{F} = \sum_{s=1}^{N_s} \boldsymbol{F}^{(s)} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{S}_K^{(s)^+} \boldsymbol{B}^{(s)^T} = \boldsymbol{B}^{\boxminus} \, \boldsymbol{S}_K^{\boxtimes^+} \, \boldsymbol{B}^{\boxminus^T} . \tag{2.33}$$

Like $\boldsymbol{S}_K$ is a stiffness defined on the primal interface $\boldsymbol{u}_I$, the operator $\boldsymbol{F}$ is a flexibility defined on the dual interface $\boldsymbol{\lambda}$. In the context of dual Schur complement-type methods, i.e., FETI, $\boldsymbol{F}$ will often be called *interface operator*. In the same way, the term from external forces is also assembled to a global quantity denoted by

$$\boldsymbol{d} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{S}_K^{(s)^+} \hat{\boldsymbol{f}}_{\text{ext},I}^{(s)} = \boldsymbol{B}^{\boxminus} \, \boldsymbol{S}_K^{\boxtimes^+} \, \hat{\boldsymbol{f}}_{\text{ext},I}^{\boxplus} \tag{2.34}$$

To abbreviate the term that originates from the zero-energy modes and to write the self-equilibrium condition in block form, the quantities

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{B}^{(1)}\boldsymbol{R}_I^{(1)} & \boldsymbol{B}^{(2)}\boldsymbol{R}_I^{(2)} & \cdots & \boldsymbol{B}^{(N_s)}\boldsymbol{R}_I^{(N_s)} \end{bmatrix} \qquad \boldsymbol{\alpha}_R = \begin{bmatrix} \boldsymbol{\alpha}_R^{(1)} \\ \boldsymbol{\alpha}_R^{(2)} \\ \vdots \\ \boldsymbol{\alpha}_R^{(N_s)} \end{bmatrix} \qquad \boldsymbol{e} = \begin{bmatrix} \boldsymbol{R}_I^{(1)^T}\hat{\boldsymbol{f}}_{\text{ext},I}^{(1)} \\ \boldsymbol{R}_I^{(2)^T}\hat{\boldsymbol{f}}_{\text{ext},I}^{(2)} \\ \vdots \\ \boldsymbol{R}_I^{(N_s)^T}\hat{\boldsymbol{f}}_{\text{ext},I}^{(N_s)} \end{bmatrix}$$

are introduced. The columns of $\boldsymbol{G}$ can be interpreted as a mapping of the zero-energy modes onto the dual interface. The vector $\boldsymbol{e}$ reflects the non-equilibrated part of the external forces. With these abbreviations, the *dual interface problem* finally reads

$$\boldsymbol{F}\boldsymbol{\lambda} + \boldsymbol{G}\boldsymbol{\alpha}_R = \boldsymbol{d} \tag{2.35}$$
$$\boldsymbol{G}^T\boldsymbol{\lambda} = \boldsymbol{e} \tag{2.36}$$

Equation (2.35) is the short form of equation (2.32), using the introduced global abbreviations. Different from the primal interface problem, it still requires the self-equilibrium condition of equation (2.30), which is now expressed as the single equation (2.36). It is often expressed as one equation in the form

$$\begin{bmatrix} \boldsymbol{F} & \boldsymbol{G} \\ \boldsymbol{G}^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha}_R \end{bmatrix} = \begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{e} \end{bmatrix}. \tag{2.37}$$

This form is known as saddle-point problem and emphasizes that the unknowns are the connection forces $\boldsymbol{\lambda}$ and the zero-energy mode amplitudes $\boldsymbol{\alpha}_R$.

Dual interface problem using non-condensed quantities. The dual interface problem can also be expressed using the non-condensed quantities. The local balance equations and the continuity of displacement condition then read

$$\boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)} = \boldsymbol{f}_{\text{ext}}^{(s)} - \boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda} \qquad s = 1, \ldots, N_s \tag{2.38}$$
$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)}\boldsymbol{t}^{(s)}\boldsymbol{u}^{(s)} = \boldsymbol{0} \tag{2.39}$$

and the self-equilibrium condition

$$\boldsymbol{R}^{(s)^T}\left(\boldsymbol{f}_{\text{ext}}^{(s)} - \boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}\right) = \boldsymbol{0}. \tag{2.40}$$

The interface operator expressed by non-condensed quantities reads

$$\boldsymbol{F} = \sum_{s=1}^{N_s} \boldsymbol{F}^{(s)} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)}\boldsymbol{t}^{(s)}\boldsymbol{K}^{(s)^+}\boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T} = \boldsymbol{B}^{\boxminus}\ \boldsymbol{t}^{\boxtimes}\ \boldsymbol{K}^{\boxtimes^+}\ \boldsymbol{t}^{\boxtimes^T}\ \boldsymbol{B}^{\boxminus^T} \tag{2.41}$$

and the corresponding right-hand side

$$\boldsymbol{d} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{t}^{(s)} \boldsymbol{K}^{(s)^+} \boldsymbol{f}_{\text{ext}}^{(s)} = \boldsymbol{B}^{\boxminus} \; \boldsymbol{t}^{\boxtimes} \; \boldsymbol{K}^{\boxtimes^+} \; \hat{\boldsymbol{f}}_{\text{ext}}^{\boxplus} \; . \tag{2.42}$$

Analogously, the restricted zero-energy modes and the non-equilibrated part of the external forces read

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{B}^{(1)} \boldsymbol{t}^{(1)} \boldsymbol{R}^{(1)} & \boldsymbol{B}^{(2)} \boldsymbol{t}^{(2)} \boldsymbol{R}^{(2)} & \cdots & \boldsymbol{B}^{(N_s)} \boldsymbol{t}^{(N_s)} \boldsymbol{R}^{(N_s)} \end{bmatrix} \qquad \boldsymbol{e} = \begin{bmatrix} \boldsymbol{R}^{(1)^T} \boldsymbol{f}_{\text{ext}}^{(1)} \\ \boldsymbol{R}^{(2)^T} \boldsymbol{f}_{\text{ext}}^{(2)} \\ \vdots \\ \boldsymbol{R}^{(N_s)^T} \boldsymbol{f}_{\text{ext}}^{(N_s)} \end{bmatrix} .$$

The amplitudes $\boldsymbol{\alpha}_R^{(s)}$ of the zero-energy modes and their block representation are not affected.

2.1.5 Redundant Lagrange Multipliers

Already equation (2.18) as definition for possible dual connectivity matrices $\boldsymbol{B}^{\boxminus}$ implies that those are not unique. For example, they do not necessarily have to be Boolean. Furthermore, equation (2.18) allows redundant constraints, i.e., that $\boldsymbol{B}^{\boxminus^T}$ does not exhibit full column rank.

Redundancy of the Lagrange multipliers is closely related to the occurrence of crosspoints. While more elaborate descriptions are used in the literature where this aspect is of primary interest, it is sufficient in this work to define crosspoints as nodes of the original global system that are shared by more than two substructures. In the presence of crosspoints, redundancy is introduced automatically if each pair of coinciding interface nodes is connected via a Lagrange multiplier and with it by a constraint.

For example, as illustrated in figure 2.5, two Lagrange multipliers are sufficient to connect three substructures. However, connecting each pair creates a third Lagrange multiplier and thereby a redundant one. A further example is given in figure 2.6 where the situation is illustrated for the case of four substructures. They are connected sufficiently by three Lagrange multipliers, while the full set includes three more, such that in total six Lagrange multipliers are generated.

For the understanding of mechanically consistent preconditioners and for analyzing the spectral properties of the interface operator, it is important to know the number of redundant Lagrange multipliers. To this end, the multiplicity of a node is defined as the number of substructures that share this node. According to this, all nodes depicted in figure 2.5 hold a multiplicity of three and all depicted nodes in figure 2.6 hold a multiplicity of four. The smallest possible multiplicity of an interface node is thus two and there is no upper limit in theory. Let the multiplicity of a global

Figure 2.5. Non-redundant and redundant Lagrange multipliers at a crosspoint with a multiplicity of three.

node i be denoted by μ_i and let the number of Lagrange multipliers that are generated to connect the substructure nodes that emerged from global node i, if the full, redundant set is used, by $N_{\lambda,i,\max}$. Then, $N_{\lambda,i,\max}$ is computed by

$$N_{\lambda,i,\max} = \sum_{k=1}^{\mu_i - 1} k \,,$$

corresponding to the illustrations on the *right* of figure 2.5 ($N_{\lambda,i,\max} = 3$) and figure 2.6 ($N_{\lambda,i,\max} = 6$). On the contrary, the number of Lagrange multipliers that is needed when a minimal, non-redundant set should be used, is determined by

$$N_{\lambda,i,\min} = \mu_i - 1 \,,$$

corresponding to the illustrations on the *left* of figure 2.5 ($N_{\lambda,i,\min} = 2$) and figure 2.6 ($N_{\lambda,i,\min} = 3$). By applying the formula for a finite arithmetic series, the number of additional redundant Lagrange multipliers for the global node i can be computed by

$$N_{\lambda,i,\text{redundant}} = N_{\lambda,i,\max} - N_{\lambda,i,\min} = \sum_{k=1}^{\mu_i - 2} k = \frac{(\mu_i - 2)(\mu_i - 2 + 1)}{2} \,.$$

Consequently, the number of redundant Lagrange multipliers per global node i and the total number are defined as

$$N_{\lambda,i,\text{redundant}} = \frac{(\mu_i - 2)(\mu_i - 1)}{2} \qquad \text{and} \qquad N_{\lambda,\text{redundant}} = \sum_i N_{\lambda,i,\text{redundant}} \tag{2.43}$$

respectively. The formula shows that for the simplest case of an interface node $\mu_i = 2$, no redundant Lagrange multipliers are created. Therefore, only interface nodes with $\mu_i > 2$ must be taken into account here.

Consequences for the solution of the interface problem. Using a redundant set of Lagrange multipliers causes the interface operator $\boldsymbol{F}$ to become semi-positive definite instead of positive-definite. Recalling its definition by block matrices in equation (2.33) reveals that $\boldsymbol{F}$ inherits the core of $\boldsymbol{B}^{\boxminus T}$ and thus

$$\ker(\boldsymbol{F}) = \ker\left(\boldsymbol{B}^{\boxminus T}\right)$$

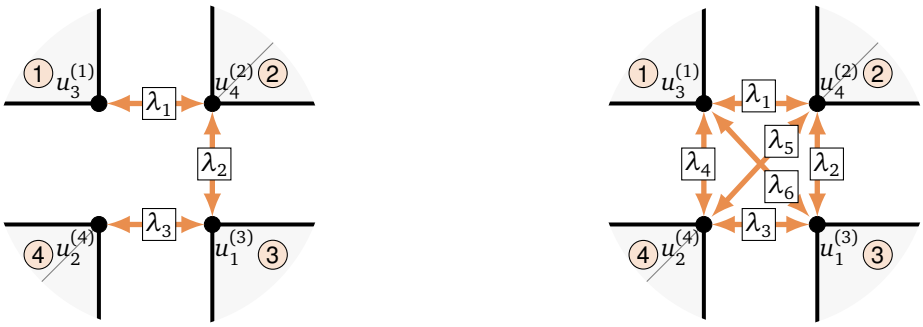


Figure 2.6. Non-redundant and redundant Lagrange multipliers at a crosspoint with a multiplicity of four.

The dimension of the kernel of the interface operator $\boldsymbol{F}$ can be computed by summing equation (2.43) over all nodes i on the global interface Γ_I, which yields

$$\operatorname{rank}(\ker(\boldsymbol{F})) = \sum_i \frac{(\mu_i - 2)(\mu_i - 1)}{2}.$$

The interface problem is still solvable although multiple solutions exist because the range of the right-hand side $\boldsymbol{d}$ and the range of the interface operator $\boldsymbol{F}$ are both in $\operatorname{range}\left(\boldsymbol{B}^{\boxminus}\right)$. Choosing the solution from $\operatorname{range}\left(\boldsymbol{B}^{\boxminus}\right)$ leads to a unique solution because of the fundamental theorem of linear algebra [31]

$$\operatorname{range}\left(\boldsymbol{B}^{\boxminus}\right) = \ker\left(\boldsymbol{B}^{\boxminus T}\right)^{\perp}.$$

In either case, the resulting originally sought displacement field, the corresponding strain, and the corresponding stress are all unique and independent of the specific choice of $\boldsymbol{\lambda}$ [34].

Necessity of redundant constraints. While finite element tearing and interconnecting was originally introduced without redundant Lagrange multipliers in [20], they were soon recognized as useful or even necessary for the performance of the algorithm. In [35] it is claimed and justified that redundant constraints have a positive impact on the convergence of the iterative procedure as they establish better communication between substructures. Furthermore, the efficiency of the Dirichlet preconditioner is shown to be reliant on a full set of inter-subdomain constraints. In [36] then, mechanically consistent preconditioners, which are accounted for in section 2.3 are built upon this foundation.

2.2 Iterative Solution

After the non-overlapping decomposition and dual formulation of the original global problem, an important part of the FETI method is the iterative solution of the interface problem. In fact, only an iterative solution procedure enables a large number

of operations to be efficiently performed in parallel. The short form of the interface problem, consisting of equation (2.35) and equation (2.36), is used throughout this section to describe the iterative numerical solution algorithm because this renders possible to clearly see important numerical properties of the method and improves the readability of the algorithms.

It should always be kept in mind that these global quantities, like the interface operator $\boldsymbol{F}$, are never assembled explicitly in the implementation of these computations. Instead, most of the operations can obviously be carried out by independent calculations within each substructure and require only a small amount of communication, mostly limited to neighboring substructures. In this context, one substructure can usually be regarded as one independent process, probably handled by one processor in a large network of connected processors, like in modern high performance computing hardware. This parallel structure becomes clear when expressing the global quantities by local quantities, e.g., the interface operator $\boldsymbol{F}$ as its sum of local contributions. For example, the application of the operator to a vector, which is the most important operation of most iterative algorithms, reads

$$\boldsymbol{F}\boldsymbol{\lambda} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{S}_K^{(s)^+} \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda}\,.$$

Assuming that every substructure already knows $\boldsymbol{\lambda}$ on its local interface, the single contributions require to solve completely independent Neumann problems on each substructure, i.e., on each processor. Subsequently, the results must be exchanged between every pair of neighbors s and r on their common interface $\Gamma^{(s,r)}$. Finally, every substructure can compute the solution of $\boldsymbol{F}\boldsymbol{\lambda}$ on its interface independently.

While many also widely used iterative solvers, for example, the generalized minimal residual (GMRES) algorithm, can be employed, the best choice is always the conjugate gradient method if the problem allows its application. That is most generally the case when $\boldsymbol{F}$ is symmetric. For non-symmetric problems, which can occur in fracture mechanics or in rotational dynamics with gyroscopic effects, the conjugate gradient method cannot be used. Throughout this work, only symmetric problems are solved and thus no other iterative algorithm is employed.

Next, this section motivates the concept of conjugate directions before introducing the method of conjugate gradients. Subsequently, the preconditioned version of the method, adapted to the constrained interface problem of FETI by projections, is presented.

2.2.1 Conjugate Directions

To introduce the concept of conjugate directions, a fundamental basis of the CG algorithm, an unconstrained system of linear equations, e.g., the FETI system when no floating substructures are present, is considered. When an initial value $\boldsymbol{\lambda}_0$ has

been set as first estimate for $\boldsymbol{\lambda}$, the remaining problem reads

$$\boldsymbol{F}\hat{\boldsymbol{\lambda}} = \boldsymbol{r}_0 \qquad \text{with} \qquad \boldsymbol{r}_0 = \boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}_0 \qquad \text{and} \qquad \boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \hat{\boldsymbol{\lambda}}\,.$$

Considering a symmetric positive definite (SPD) matrix $\boldsymbol{F}$, or a symmetric semi-positive definite matrix as long as the right-hand side is in the operators range, this problem is equivalent to solving the optimization problem

$$\min_{\hat{\boldsymbol{\lambda}}} \left[f(\hat{\boldsymbol{\lambda}}) = \frac{1}{2}\hat{\boldsymbol{\lambda}}^T \boldsymbol{F}\hat{\boldsymbol{\lambda}} - \boldsymbol{r}_0^T\hat{\boldsymbol{\lambda}} \right]. \tag{2.44}$$

In case $\boldsymbol{F}$ is a diagonal matrix

$$\boldsymbol{F} = \operatorname{diag}\left(\Theta_1, \Theta_2, \ldots, \Theta_{N_\lambda}\right)$$

the function $f(\hat{\boldsymbol{\lambda}})$ would not contain any terms where different elements $\hat{\lambda}_1$, $\hat{\lambda}_2$, ... of $\hat{\boldsymbol{\lambda}}$ are mixed, i.e.,

$$f(\hat{\boldsymbol{\lambda}}) = \frac{1}{2}\left(\hat{\lambda}_1^2\Theta_1 + \hat{\lambda}_2^2\Theta_2 + \cdots + \hat{\lambda}_{N_\lambda}^2\Theta_{N_\lambda}\right) - \hat{\lambda}_1 r_1 - \hat{\lambda}_2 r_2 - \cdots - \hat{\lambda}_{N_\lambda} r_{N_\lambda}\,.$$

Finding the solution $\hat{\boldsymbol{\lambda}}$ would be easy in this case because setting every partial derivation of $f(\hat{\boldsymbol{\lambda}})$ with respect to $\hat{\lambda}_i$ to zero immediately delivers the solution

$$\hat{\lambda}_i = \frac{r_i}{\Theta_i}\,.$$

In the general case of course, $\boldsymbol{F}$ is not diagonal, and every possible mixed term in the form of $\hat{\lambda}_i\hat{\lambda}_j$ with $i \neq j$ can appear. The partial derivatives will then form a coupled system of equations and a simple solution like above is not possible. The first idea behind the method of CG is now to compose an approximate solution $\hat{\boldsymbol{\lambda}}_i$ from the linear combination

$$\hat{\boldsymbol{\lambda}}_i = \sum_{k=0}^{i-1} \alpha_k \boldsymbol{w}_k \qquad i = 1, \ldots, N_\lambda$$

using i $\boldsymbol{F}$-conjugated vectors $\boldsymbol{w}_k$, and to find the optimal coefficients α_k. Optimal refers in this case to minimizing $f(\hat{\boldsymbol{\lambda}}_i)$ according to equation (2.44) if $i < N_\lambda$ vectors are employed because in general, the exact solution can only be found if $i = N_\lambda$ linearly independent vectors are used.

$\boldsymbol{F}$-conjugated means that they are orthogonal with respect to the inner product that is induced by the matrix $\boldsymbol{F}$ such that

$$\boldsymbol{w}_i^T \boldsymbol{F}\boldsymbol{w}_j = 0 \quad \forall \quad i \neq j$$

holds. Because of the pairwise $\boldsymbol{F}$-conjugacy of the $\boldsymbol{w}_i$ vectors, the function $f(\hat{\boldsymbol{\lambda}})$ now shrinks down to

$$f(\alpha_0, \alpha_1, \ldots, \alpha_{n-1}) = \frac{1}{2}\sum_{i=0}^{n-1} \alpha_i^2 \boldsymbol{w}_i^T \boldsymbol{F}\boldsymbol{w}_i - \sum_{i=0}^{n-1} \alpha_i \boldsymbol{r}_0^T \boldsymbol{w}_i\,.$$

Partial derivation with respect to α_i now immediately delivers the solution

$$\frac{\partial}{\partial \alpha_i} f(\alpha_i) = \alpha_i \boldsymbol{w}_i^T \boldsymbol{F} \boldsymbol{w}_i - \boldsymbol{r}_0^T \boldsymbol{w}_i = 0 \qquad \text{and thus} \qquad \alpha_i = \frac{\boldsymbol{r}_0^T \boldsymbol{w}_i}{\boldsymbol{w}_i^T \boldsymbol{F} \boldsymbol{w}_i}.$$

This result means that for a given set of $\boldsymbol{F}$-conjugate directions, their optimal coefficients to minimize the function f can be computed independently from each other. Even when they are added subsequently, like in an iterative method that improves the solution with every added $\boldsymbol{w}_i$, they do not interfere with each other. A conjugate basis of the space spanned by an arbitrary set of vectors can be computed by the Gram-Schmidt orthogonalization procedure [37] and will simply be referred to as orthogonalization in the following.

2.2.2 Conjugate Gradients

It is not specified yet how to compute the directions $\boldsymbol{w}_i$, called search directions. An arbitrary choice of conjugate search directions will not show good convergence, i.e., decreasing value of the functional f when adding the directions step by step. A suitable choice, leading to the method of conjugate gradients, is to build the search direction subsequently from the residual

$$\boldsymbol{r}_i = \boldsymbol{r}_0 - \boldsymbol{F}\hat{\boldsymbol{\lambda}}_i = \boldsymbol{r}_{i-1} - \alpha_{i-1}\boldsymbol{F}\boldsymbol{w}_{i-1} \tag{2.45}$$

of the current solution that is composed of the already used search directions. The next search direction $\boldsymbol{w}_i$ can be then be obtained by making $\boldsymbol{r}_i$ through an orthogonalization $\boldsymbol{F}$-conjugate against all former search directions. Because of the $\boldsymbol{F}$-conjugacy, adding a new search direction does not change the optimal choice of the coefficients for the old search directions already used. This trick ensures that the conjugate gradient algorithm will find the exact solution after at most N_λ iterations using N_λ search directions.

If the problem is well conditioned, the use of the residual to build the next search direction by orthogonalization leads to suitable convergence, i.e., the solution will improve with every iteration. The residual always equals the direction of steepest descent or the negative gradient of $f(\hat{\boldsymbol{\lambda}})$. While the solution in iteration i is composed from the first i search directions, the error is of course composed of the remaining $N_\lambda - i$ unused search directions. Denoting the first search direction with $\boldsymbol{w}_0$ the error writes

$$\boldsymbol{e}_i = \hat{\boldsymbol{\lambda}}_i - \hat{\boldsymbol{\lambda}} = -\sum_{j=i}^{N_\lambda - 1} \alpha_j \boldsymbol{w}_j .$$

As a result, the error $\boldsymbol{e}_i$ is $\boldsymbol{F}$-conjugated to every search direction $\boldsymbol{w}_j$ that has been used before so that

$$\boldsymbol{w}_j^T \boldsymbol{F} \boldsymbol{e}_i = 0 \quad j = 0, \ldots, i-1$$

holds. Because

$$\boldsymbol{F}\boldsymbol{e}_i = \boldsymbol{F}(\hat{\boldsymbol{\lambda}}_i - \hat{\boldsymbol{\lambda}}) = \boldsymbol{F}\hat{\boldsymbol{\lambda}}_i - \boldsymbol{r}_0 - \boldsymbol{F}\hat{\boldsymbol{\lambda}} + \boldsymbol{r}_0 = -\boldsymbol{r}_i$$

the residual is always orthogonal to the previously used search directions and because the previous directions were constructed by the previous residuals, every residual is also orthogonal to all the previous residuals, i.e.,

$$\boldsymbol{r}_j^T \boldsymbol{r}_i = 0 \quad \forall \quad j < i\,. \tag{2.46}$$

Recalling how $\boldsymbol{r}_j$ is built from $\boldsymbol{r}_{j-1}$ in equation (2.45) and the search direction $\boldsymbol{w}_{j-1}$, this relation can also be expressed as

$$\left(\boldsymbol{r}_{j-1} - \alpha_{j-1}\boldsymbol{F}\boldsymbol{w}_{j-1}\right)^T \boldsymbol{r}_i = 0 \quad \forall \quad j < i$$

and leads by using again equation (2.46) to

$$\boldsymbol{w}_{j-1}^T \boldsymbol{F}\boldsymbol{r}_i = 0 \quad \forall \quad j < i \qquad \text{or equivalently} \qquad \boldsymbol{w}_j^T \boldsymbol{F}\boldsymbol{r}_i = 0 \quad \forall \quad j < i-1\,.$$

As a consequence, a new search direction $\boldsymbol{w}_i$ that is built from the current residual $\boldsymbol{r}_i$ is already $\boldsymbol{F}$-conjugate to the former search directions $\boldsymbol{w}_j$ with $j = 0, \ldots, i-2$ and must thus only be orthogonalized against the latest search direction $\boldsymbol{w}_{i-1}$. This is also called the short recurrence relation of the conjugate gradient algorithm. Unfortunately, in many practical applications, numerical errors make it necessary to orthogonalize a new search direction against all former search directions.

2.2.3 Projected Preconditioned Conjugate Gradient

The basic FETI procedure is basically a CG algorithm that solves the interface problem of equation (2.37). In the presence of floating substructures, this interface problem is a constrained problem as it must respect the self-equilibrium of floating substructures at all times. This section illustrates the basic steps of the *projected* CG. It enables to restrict the iterative search for a solution to the subspace of self-equilibrated substructures by projections. Furthermore, a preconditioner is introduced, finally leading to the projected preconditioned conjugate gradient (PPCG) method that was originally proposed as solution algorithm for FETI in [20]. For a static problem, the following steps are run through.

1. Choose a start solution: choose any initial $\boldsymbol{\lambda}$ that fulfills the self-equilibrium condition
2. Compute the projected residual: it corresponds to the gaps between substructure interfaces
   - Calculate the deformation of every substructure according to $\boldsymbol{f}_{\text{ext}}$ and $\boldsymbol{\lambda}$
   - Calculate the resulting gaps between all pairs of connected interface nodes
   - Choose the rigid body mode amplitudes for floating substructures such that the gaps are minimized

3. Check for convergence: if the gaps undercut a specific tolerance, exit
4. Generate a new search direction: estimate a correction of connection forces
   - Preconditioning: estimate an adaption of connection forces needed to close the gaps
   - Projection: remove parts from the search directions that would hurt the self-equilibration of floating substructures
   - Orthogonalization: make the new search direction $\boldsymbol{F}$-conjugate to the previous ones
5. Adapt the solution: compute the optimal step length and adjust $\boldsymbol{\lambda}$
6. Return to step 2

Step 1: find an admissible start vector

In the first step, one needs to find a start vector that satisfies the self equilibrium condition. It is denoted $\boldsymbol{\lambda}_G$ from here on. Every vector or search direction fulfilling or not violating the self-equilibrium condition is called admissible. Recalling that requirements are imposed on $\boldsymbol{\lambda}$ only within the space spanned by $\boldsymbol{G}$, these requirements can be satisfied by choosing $\boldsymbol{\lambda}$ from the space spanned by $\boldsymbol{G}$. More generally, any other space, for example, the space spanned by a symmetric positive definite matrix $\boldsymbol{Q}$, could be used such that

$$\boldsymbol{\lambda}_G = \boldsymbol{Q}\boldsymbol{G}\boldsymbol{\mu}.$$

it is only required that

$$\operatorname{rank}\left(\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{G}\right) = \operatorname{rank}\left(\boldsymbol{G}^T\boldsymbol{G}\right) \tag{2.47}$$

holds. Injecting this Ansatz into the self-equilibrium condition $\boldsymbol{G}^T\boldsymbol{\lambda}_G = \boldsymbol{e}$ yields

$$\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{G}\boldsymbol{\mu} = \boldsymbol{e} \quad \text{and thus} \quad \boldsymbol{\mu} = (\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{G})^{-1}\boldsymbol{e}\,.$$

This finally leads to the admissible starting vector

$$\boldsymbol{\lambda}_0 = \boldsymbol{\lambda}_G = \boldsymbol{Q}\boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{G})^{-1}\boldsymbol{e}\,. \tag{2.48}$$

If the global structure is not floating, $\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{G}$ and $\boldsymbol{G}^T\boldsymbol{G}$ have full rank. In case of a floating global structure, $\boldsymbol{G}^T\boldsymbol{G}$ as well as the global stiffness matrix $\boldsymbol{K}$ are singular. However, if equation (2.47) holds and the global problem in equation (1.1) is well posed, i.e., $\boldsymbol{f}_{\text{ext}}$ is in the range of $\boldsymbol{K}$, which is the case if $\boldsymbol{f}_{\text{ext}}$ is self-equilibrated in the directions of global rigid body modes, the FETI method can still be used. [38]

Step 2: calculate gaps

These steps require solving local Neumann problems to calculate the deformation of each substructure. They further require communication between neighbors to exchange the resulting displacements. The computation of the displacements of the

substructures includes choosing the amplitudes of their rigid body modes. The residual considering only the displacements by deformation, computed by applying the pseudoinverse, is denoted as

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

while the residual also considering rigid body modes is denoted as

$$\boldsymbol{r}_i = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)}\boldsymbol{t}^{(s)}\boldsymbol{u}_i^{(s)} = \boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}_i - \boldsymbol{G}\boldsymbol{\alpha}_{R,i} = \tilde{\boldsymbol{r}}_i - \boldsymbol{G}\boldsymbol{\alpha}_{R,i} .$$

The first residual, corresponding to the initial admissible starting vector $\boldsymbol{\lambda}_0$ is thus indexed as $\boldsymbol{r}_0$. What can be expected from choosing $\boldsymbol{\alpha}_R$ and what is as well necessary to find the correct solution is to eliminate the error when projected onto the space of rigid body modes. More general, an oblique projection

$$\boldsymbol{G}^T\boldsymbol{Q}(\tilde{\boldsymbol{r}}_i - \boldsymbol{G}\boldsymbol{\alpha}_{R,i}) = \boldsymbol{0} \tag{2.49}$$

can be used. This is equivalent to using a particular norm that is defined by the symmetric positive matrix $\boldsymbol{Q}$ and yields the solution

$$\boldsymbol{\alpha}_{R,i} = (\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{G})^{-1}\boldsymbol{G}^T\boldsymbol{Q}\tilde{\boldsymbol{r}}_i . \tag{2.50}$$

As long as the solution is not converged, the rigid body modes are not of explicit interest. However, they appear in the resulting residual $\boldsymbol{r}$. Injecting the formula for the rigid body mode amplitudes $\boldsymbol{\alpha}_R$ of equation (2.50), the residual can be expressed as

$$\boldsymbol{r}_i = \tilde{\boldsymbol{r}}_i - \boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{G})^{-1}\boldsymbol{G}^T\boldsymbol{Q}\tilde{\boldsymbol{r}}_i .$$

By defining the projector

$$\boldsymbol{P}_G = \boldsymbol{I} - \boldsymbol{Q}\boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{G})^{-1}\boldsymbol{G}^T ,$$

this operation can be shortly written as

$$\boldsymbol{r}_i = \boldsymbol{P}_G^T\tilde{\boldsymbol{r}}_i . \tag{2.51}$$

The conclusion of this step is that the residual $\boldsymbol{r}$ with optimally chosen rigid body mode amplitudes, i.e., the best residual for a given $\boldsymbol{\lambda}$, is found by projecting the residual $\tilde{\boldsymbol{r}}$, which can include arbitrary rigid body mode components. Hence, the projection of the residual by the projector $\boldsymbol{P}_G^T$ implicates the calculation of optimal rigid body mode amplitudes. In this context, optimal refers to the criterion of equation (2.49). This step is coupling all floating substructures because a rigid body displacement of one substructure inherently also affects the residual, i.e., the gaps, of its neighbors along the common interface.

The problem of the type $(\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{G})\boldsymbol{x} = \boldsymbol{y}$, which must be solved during every application of $\boldsymbol{P}_G$ or $\boldsymbol{P}_G^T$, is thus called natural coarse problem and the space spanned by the columns of $\boldsymbol{G}$ is called the natural coarse space.

Step 3: check for convergence

After the current residual is computed, it must be checked for convergence, i.e., if the quality of the solution is already satisfying. In general, the residual could be measured in any suitable norm. The norm induced by the preconditioner $\boldsymbol{H}$ is most common and a good estimator for the global residual. Two different types of residuals are important for the numerical experiments in this work. The first one is based on the absolute measure of the residual, such that the convergence criterion is expressed by the inequality

$$r_{\mathrm{abs},i} < \epsilon_{r,\mathrm{abs}} \qquad r_{\mathrm{abs},i} = \|\boldsymbol{r}_i\|_{\boldsymbol{H}} \,.$$

Usually, the absolute scale of the residual is unknown. In this case, it is more reasonable from a numerical point of view to expect a decrease of the residual by a certain number of orders of magnitude. In this case, the inequality that expresses the convergence criterion reads

$$r_{\mathrm{rel},i} < \epsilon_{r,\mathrm{rel}} \qquad r_{\mathrm{rel},i} = \frac{\|\boldsymbol{r}_i\|_{\boldsymbol{H}}}{\|\boldsymbol{r}_0\|_{\boldsymbol{H}}} \,.$$

For example, setting a relative tolerance of $\epsilon_{r,\mathrm{rel}} = 10^{-8}$ asks the residual to decrease by 8 orders of magnitude during the iterations. In the following algorithms, the placeholders $r_{*,i}$ and $\epsilon_{r,*}$ are used.

Step 4: estimate a correction for connection forces

The probably most important step regarding the performance of an iterative solution algorithm is the preconditioning. Basically, the preconditioner tries to estimate the solution error by means of the residual. If this estimate is chosen well, an iterative algorithm is able to converge in a number of iterations that is much lower than the size of the solved problem. As a consequence, it can be much more efficient than a direct solution algorithm, provided that the application of the preconditioner is not too costly. In the case of FETI, the preconditioning step estimates the necessary correction of the connection forces based on the current gap between the interfaces of the substructures. In the PPCG algorithm, the preconditioning step must be followed by a projection to make sure that the self-equilibrium condition is not violated. Afterwards, the final search direction can be created by orthogonalization.

Preconditioning. Instead of directly employing the residual as search direction, one tries to estimate the adaption that is necessary to make the residual vanish. Physically, in terms of solving a problem of linear elasticity, this is the adaption of the interface forces, necessary to close the gaps between the substructures. It usually involves solving local Dirichlet problems when applying a preconditioner $\boldsymbol{H}$, which should be an approximation of the inverse of the operator $\boldsymbol{F}$ such that

$$\boldsymbol{H} \approx \boldsymbol{F}^{-1} \,.$$

Its application to the residual then delivers an estimate of the direction in which the exact solution can be found. This direction is denoted by

$$\boldsymbol{z}_i = \boldsymbol{H}\boldsymbol{r}_i\,,$$

which will also be called preconditioned residual from now on.

Projection. Directly using the preconditioned residual as search direction, for example, $\Delta\boldsymbol{\lambda}_{i+1} = \alpha_i \boldsymbol{z}_i$, to adapt the solution for $\boldsymbol{\lambda}$ would corrupt the self-equilibration of free floating substructures such that

$$\boldsymbol{G}^T(\boldsymbol{\lambda}_G + \Delta\boldsymbol{\lambda}_{i+1}) \neq \boldsymbol{e}\,.$$

This problem is handled by another projection that makes sure that

$$\boldsymbol{G}^T \Delta\boldsymbol{\lambda}_{i+1} = \boldsymbol{0} \tag{2.52}$$

holds and thus the condition will not be hurt. In particular, $\boldsymbol{z}$ needs to be split into $\hat{\boldsymbol{z}}$ and $\boldsymbol{w}$ such that

$$\boldsymbol{z} = \hat{\boldsymbol{z}} + \boldsymbol{w} \tag{2.53}$$

where

$$\boldsymbol{G}^T \boldsymbol{w} = \boldsymbol{0}$$

This decomposition is not unique. Similar to the computation of an admissible start vector in equation (2.48) and the oblique projection of the residual in equation (2.51), we use the matrix $\boldsymbol{Q}$ to construct the basis $\boldsymbol{QG}$ for $\hat{\boldsymbol{z}}$. For reasons of symmetry, which become more obvious in the following, it is necessary to use the same matrix $\boldsymbol{Q}$ here as for the projection of the residual in equation (2.51). Multiplying equation (2.53) with $\boldsymbol{G}^T$ and injecting the Ansatz $\hat{\boldsymbol{z}} = \boldsymbol{QG}\boldsymbol{\mu}$ yields

$$\boldsymbol{G}^T\hat{\boldsymbol{z}} = \boldsymbol{G}^T\boldsymbol{QG}\boldsymbol{\mu} \quad \text{and} \quad \boldsymbol{\mu} = (\boldsymbol{G}^T\boldsymbol{QG})^{-1}\boldsymbol{G}^T\hat{\boldsymbol{z}}\,.$$

This results in the splitting

$$\hat{\boldsymbol{z}} = \boldsymbol{QG}(\boldsymbol{G}^T\boldsymbol{QG})^{-1}\boldsymbol{G}^T\boldsymbol{z} \qquad \text{and} \qquad \boldsymbol{w} = \boldsymbol{z} - \hat{\boldsymbol{z}} = \left(\boldsymbol{I} - \boldsymbol{QG}(\boldsymbol{G}^T\boldsymbol{QG})^{-1}\boldsymbol{G}^T\right)\boldsymbol{z}\,.$$

The projector $\boldsymbol{P}_G$ used before can be identified here again and an admissible search direction is obtained by the projection step

$$\boldsymbol{w}_i = \boldsymbol{P}_G\boldsymbol{z}_i\,.$$

The same projector $\boldsymbol{P}_G$ that was found to realize the projection of the residual, i.e., choosing amplitudes of rigid body modes, is also suitable to project a new search direction onto the space of admissible directions. Physically, this is the step of self-equilibrating the connection force vector $\boldsymbol{z}$, which is a coupled problem because adjusting the force on one substructure inherently also changes the force and thereby the equilibrium of its neighbor due to the action-reaction principle. As the first residual, corresponding to the starting vector $\boldsymbol{\lambda}_0$, is index as $\boldsymbol{r}_0$, also the corresponding first generated search direction is indexed as $\boldsymbol{w}_0$.

Orthogonalization. A simple Gram-Schmidt procedure using the norm induced by the operator $\boldsymbol{F}$ to measure angles between vectors enables $\boldsymbol{F}$-conjugacy. To this end, the orthogonalization step coefficients

$$\beta_{i,j} = \frac{\boldsymbol{w}_j^T \boldsymbol{F} \boldsymbol{w}_i}{\boldsymbol{w}_j^T \boldsymbol{F} \boldsymbol{w}_j}$$

are defined, such that the orthogonalization of $\boldsymbol{w}_i$ against all former search directions $\boldsymbol{w}_0$ to $\boldsymbol{w}_{i-1}$ is achieved by

$$\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i - \sum_{j=0}^{i-1} \boldsymbol{w}_j \beta_{i,j} .$$

As indicated before, due to the short recurrence in the CG method, a reorthogonalization of $\boldsymbol{w}_i$ against $\boldsymbol{w}_{i-1}$ would be sufficient in exact arithmetics, but due to numerical errors, a full orthogonalization, then sometimes also called reorthogonalization, is necessary.

Step 5: adaption of the interface forces

After the projection of the residual, the preconditioning step and another projection step followed by orthogonalization, a new admissible conjugate search direction is ready to be used. The only step that remains to be done is the computation of a suitable step length. Ideally, the adaption $\Delta\boldsymbol{\lambda}_{i+1}$ should solve

$$\boldsymbol{F}\Delta\boldsymbol{\lambda}_{i+1} = \boldsymbol{r}_i \qquad \text{with} \qquad \Delta\boldsymbol{\lambda}_{i+1} = \alpha_i \boldsymbol{w}_i$$

to find the exact solution. Of course, this is usually not possible. Therefore, the problem is again reformulated as the optimization problem

$$\min_{\alpha_i} \left[f(\alpha_i) = \frac{1}{2}(\alpha_i \boldsymbol{w}_i)^T \boldsymbol{F}(\alpha_i \boldsymbol{w}_i) - \boldsymbol{r}_i^T \alpha_i \boldsymbol{w}_i \right],$$

which yields as optimal solution for the step length α_i

$$\frac{\partial}{\partial \alpha_i} f(\alpha_i) = \alpha_i \boldsymbol{w}_i^T \boldsymbol{F} \boldsymbol{w}_i - \boldsymbol{r}_i^T \boldsymbol{w}_i = 0 \qquad \Rightarrow \qquad \alpha_i = \frac{\boldsymbol{r}_i^T \boldsymbol{w}_i}{\boldsymbol{w}_i^T \boldsymbol{F} \boldsymbol{w}_i} .$$

The connection forces $\boldsymbol{\lambda}$ are subsequently adapted to

$$\boldsymbol{\lambda}_{i+1} = \boldsymbol{\lambda}_i + \alpha_i \boldsymbol{w}_i$$

At this point, one more iteration is completed. For example if $i = 2$, $\boldsymbol{\lambda}_3$ is the approximate solution of the interface problem after 3 iterations, consisting of the starting vector $\boldsymbol{\lambda}_0$ and the three search directions $\boldsymbol{w}_0$, $\boldsymbol{w}_1$, and $\boldsymbol{w}_2$, which have been computed and added to improve the accuracy.

The complete procedure is listed in Algorithm 1. Some formulas are slightly adapted so that no identical computations, e.g., the application of the operator or the preconditioner, are unnecessarily executed twice.

Algorithm 1: FETI-1

$\boldsymbol{\lambda}_G = \boldsymbol{Q}\boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{G})^{-1}\boldsymbol{e}$
$\hat{\boldsymbol{\lambda}}_0 = \boldsymbol{0}$
$\boldsymbol{r}_0 = \boldsymbol{P}_G^T(\boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}_G)$
$\boldsymbol{z}_0 = \boldsymbol{H}\boldsymbol{r}_0$
$i = 0$
while $r_{*,i} > \epsilon_{r,*}$ **do**
  for $0 \leq j \leq i-1$ **do**
    $\phi_{i,j} = \boldsymbol{q}_j^T\boldsymbol{z}_i \qquad \beta_{i,j} = \frac{\phi_{i,j}}{\Delta_j}$
  $\boldsymbol{w}_i = \boldsymbol{P}_G\boldsymbol{z}_i - \sum_{j=0}^{i-1}\boldsymbol{w}_j\beta_{i,j}$
  $\boldsymbol{q}_i = \boldsymbol{F}\boldsymbol{w}_i$
  $\Delta_i = \boldsymbol{q}_i^T\boldsymbol{w}_i \qquad \gamma_i = \boldsymbol{w}_i^T\boldsymbol{r}_i \qquad \alpha_i = \frac{\gamma_i}{\Delta_i}$
  $\hat{\boldsymbol{\lambda}}_{i+1} = \hat{\boldsymbol{\lambda}}_i + \boldsymbol{w}_i\alpha_i$
  $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \boldsymbol{P}_G^T\boldsymbol{q}_i\alpha_i$
  $\boldsymbol{z}_{i+1} = \boldsymbol{H}\boldsymbol{r}_{i+1}$
  $i \leftarrow i+1$
$m = i$
$\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_G + \hat{\boldsymbol{\lambda}}_i$

2.3 Preconditioning

The preconditioning is probably the most crucial step in an iterative algorithm regarding its performance, especially if the conditioning of the problem is very bad.

2.3.1 Basic Preconditioners

In this context, one of the most important properties a preconditioner must possess, is of course that its application can be computed in parallel. This implies that it will most likely be constructed from local contributions from the individual substructures, similar to the operator $\boldsymbol{F}$. The first preconditioner for FETI that was introduced and employed in the original publication [20] is the so-called lumped preconditioner. It is computed as

$$\boldsymbol{H}_L = \sum_{s=1}^{N_s}\boldsymbol{H}_L^{(s)} = \sum_{s=1}^{N_s}\boldsymbol{B}^{(s)}\boldsymbol{t}^{(s)}\boldsymbol{K}^{(s)}\boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T}$$

and physically implies the solution of a Dirichlet problem in each substructure in which all non-interface dof are fixed. A variant of this preconditioner is called superlumped. It is obtained by replacing the stiffness matrix $\boldsymbol{K}^{(s)}$ by its diagonal.

Later, the so-called Dirichlet preconditioner was proposed in [34] and proven to be optimal in [39], which means that the effective condition number of the PPCG algorithm grows only slowly when decreasing the mesh size and that it is independent of the number of subdomains. The Dirichlet preconditioner is computed as

$$\boldsymbol{H}_D = \sum_{s=1}^{N_s} \boldsymbol{H}_D^{(s)} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{S}_K^{(s)} \boldsymbol{B}^{(s)^T} .$$

Like the lumped preconditioner, it requires a Dirichlet problem to be solved on each subdomain. However, the full Schur complement is used here. As a result, the exact reaction force of the structure for a prescribed interface displacement is returned. It was also shown in [39] that the Dirichlet preconditioner is not always the computationally most efficient choice, which is related to its higher computational cost when compared to the lumped version.

Both preconditioners are symmetric. Nevertheless, also non-symmetric preconditioners can be employed as long as full orthogonalization against all former search directions is applied [40]. In algorithms throughout this work, the subscripts L and D to distinguish the different preconditioners are omitted unless a specific choice should be emphasized.

2.3.2 Scaling

All preconditioners in the current state of the art build on the two basic versions presented above. The main improvements that have been achieved are scaling procedures that allow to properly handle crosspoints and certain types of heterogeneities. In principle, these techniques scale the residual, i.e., the gaps, as well as the corresponding force resulting from the local Dirichlet problem. The scaling factors are specific to the substructures and their properties like their stiffness or node multiplicities. This can be expressed as replacing the dual assembly matrices $\boldsymbol{B}^{(s)}$ by *scaled* dual assembly matrices

$$\widetilde{\boldsymbol{B}}^{(s)} = \widetilde{\boldsymbol{\beta}}^{(s)} \boldsymbol{B}^{(s)}$$

where $\widetilde{\boldsymbol{\beta}}^{(s)}$ are diagonal scaling matrices. They have the dimension $N_\lambda \times N_\lambda$ and can assign a scaling factor to each element of a vector on the dual interface space, for example, to each connection force in $\boldsymbol{\lambda}$ or to each gap in $\boldsymbol{r}$. However, they are specific to each substructure, and $\widetilde{\boldsymbol{\beta}}^{(s)}$ only scales connection forces or gaps on the interface of s. Considering that the substructure s has at least one neighbor on its interface, it becomes apparent that the same element in $\boldsymbol{r}$ may be scaled by different factors in the two associated substructures.

Usually, all scaling procedures, even if motivated purely physically, lead to a partition of unity expressed by

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \widetilde{\boldsymbol{B}}^{(s)^T} = \boldsymbol{I} . \tag{2.54}$$

The scaled versions of the preconditioners are not denoted by separate symbols and read

$$\boldsymbol{H}_L = \sum_{s=1}^{N_s} \boldsymbol{H}^{(s)} = \sum_{s=1}^{N_s} \widetilde{\boldsymbol{B}}^{(s)} \boldsymbol{t}^{(s)} \boldsymbol{K}^{(s)} \boldsymbol{t}^{(s)^T} \widetilde{\boldsymbol{B}}^{(s)^T} = \sum_{s=1}^{N_s} \widetilde{\boldsymbol{\beta}}^{(s)} \boldsymbol{B}^{(s)} \boldsymbol{t}^{(s)} \boldsymbol{K}^{(s)} \boldsymbol{t}^{(s)^T} \boldsymbol{B}^{(s)^T} \widetilde{\boldsymbol{\beta}}^{(s)}$$

$$\boldsymbol{H}_D = \sum_{s=1}^{N_s} \boldsymbol{H}^{(s)} = \sum_{s=1}^{N_s} \widetilde{\boldsymbol{B}}^{(s)} \boldsymbol{S}_K^{(s)} \widetilde{\boldsymbol{B}}^{(s)^T} = \sum_{s=1}^{N_s} \widetilde{\boldsymbol{\beta}}^{(s)} \boldsymbol{B}^{(s)} \boldsymbol{S}_K^{(s)} \boldsymbol{B}^{(s)^T} \widetilde{\boldsymbol{\beta}}^{(s)} .$$

A first energy based attempt to deal with jumps in the coefficients of the partial differential equation was published in [41]. The concept of a node's multiplicity and an according scaling that respects equation (2.54) was developed and justified in [35]. Let the elements of the diagonal scaling matrices and the Lagrange multipliers be defined by

$$\widetilde{\boldsymbol{\beta}}^{(s)} = \begin{bmatrix} \beta_1^{(s)} & & & \\ & \beta_2^{(s)} & & \\ & & \ddots & \\ & & & \beta_{N_\lambda}^{(s)} \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{N_\lambda} \end{bmatrix}$$

and let $\Lambda^{(s)}$ be the set of Lagrange multipliers that act on substructure s. Then

$$\beta_i^{(s)} = 0 \qquad \forall \quad i \in \left\{ i \in \{1, \dots, N_\lambda\} \mid \lambda_i \notin \Lambda^{(s)} \right\},$$

which is to say that all diagonal elements of $\widetilde{\boldsymbol{\beta}}^{(s)}$ that do not correspond to interface dof on s are zero. In multiplicity scaling, the scaling coefficients are set to

$$\beta_i^{(s)} = \frac{1}{\mu_i^{(s)}}$$

where $\mu_i^{(s)}$ is the multiplicity of the node corresponding to λ_i. In case of heterogeneities, for example, when the stiffness exhibits a jump across the interface, multiplicity scaling is not sufficient anymore. The so-called stiffness or superlumped scaling was originally developed and published in [36] for static problems. It defines a mechanically consistent preconditioner by scaling the gaps and the estimated correction forces not only based on the multiplicity but on the stiffnesses of the interface dof. Recalling that every Lagrange multiplier acts on two distinct substructures, let λ_i be the Lagrange multiplier that connects the local dof n_s in substructure s and the local dof n_r in substructure r. The corresponding global dof of the undecomposed structure is denoted by n and it is shared by the substructures $S^n = \{s, r, \dots\}$. The non-zero diagonal elements in the scaling matrix of substructure s then read

$$\beta_i^{(s)} = K_{n_r,n_r}^{(r)} \left(\sum_{j \in S^n} K_{n_j,n_j}^{(j)} \right)^{-1} \qquad \forall \quad i \in \left\{ i \in \{1, \dots, N_\lambda\} \mid \lambda_i \in \Lambda^{(s)} \right\}$$

where $K_{n_r,n_r}^{(r)}$ refers to the entry of the matrix $\boldsymbol{K}^{(r)}$ in the n_r-th row and the n_r-th column, i.e. to its n_r-th diagonal entry. Explained in words, the scaling coefficient of

a Lagrange multiplier in substructure s is the stiffness of the interface dof in substructure r it connects to, divided by the sum of the stiffnesses of all dof that have emerged from the corresponding original, single dof of the undecomposed structure.

In a homogeneous structure, taking into account that the preconditioned residual is globally scaled by a step length anyway, it is equivalent to multiplicity scaling. The exact definition of the multiplicity scaling coefficients is recovered when all stiffnesses are set to one such that

$$\beta_i^{(s)} = 1 \left(\sum_{j \in S^n} 1 \right)^{-1} \qquad \forall \quad i \in \left\{ i \in \{1, \ldots, N_\lambda\} \mid \lambda_i \in \Lambda^{(s)} \right\}.$$

In this work, stiffness scaling is applied to static problems if nothing different is stated explicitly. To dynamic problems, the analogous superlumped scaling procedure described in section 3.3.1 is applied.

2.4 Deflation by Auxiliary Coarse Spaces

Already before the FETI algorithm was published, [42–44] studied deflation of the CG algorithm by projection independently from each other. It was basically identified as a method to accelerate the convergence of the standard CG method, based on auxiliary subspaces. In [44], it was already proven that the effective condition number of the deflated operator has an equal or lower condition number than the original operator, regardless of the deflation space.

In [34], the first published application to FETI can be found where the approach was described as a splitting of the substructure solution into “nice” and “troublesome” components. The technique was further used in [45] to substitute the vanishing natural coarse space in the case of dynamics. In [46] and [47], a thorough analysis was dedicated exclusively to this technique and FETI algorithms augmented this way were denominated as two-level FETI or FETI-2.

The procedure can be derived from the idea to enforce the auxiliary condition

$$\boldsymbol{C}^T \boldsymbol{r} = \boldsymbol{C}^T \left(\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{u}_I^{(s)} \right) = \boldsymbol{C}^T (\boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}) = \boldsymbol{0} \tag{2.55}$$

to be fulfilled throughout all iterations. In the standard CG algorithm, this condition is not fulfilled before convergence. The auxiliary condition equation (2.55) states that the residual should vanish within the subspace spanned by the columns of $\boldsymbol{C}$. This subspace is denominated *deflation space* or *auxiliary coarse space* or, if confusions with the natural coarse space are impossible, just *coarse space*. To realize this, the solution is calculated within the auxiliary coarse space prior to the iterative solution process, e.g., by a direct method. During the iterative solution process, a projection will prevent changes to the solution within the auxiliary coarse space.

To make sure that one can operate on the connection forces $\boldsymbol{\lambda}$ inside the auxiliary coarse space, it must not violate the self-equilibrium condition, which is ensured if

$$\boldsymbol{G}^T\boldsymbol{C} = \boldsymbol{0}$$

holds. This can easily be achieved, e.g., by using the natural coarse space projector $\boldsymbol{P}_G$ to make all basis vectors of the auxiliary coarse space self-equilibrated. This projection step must be carried out only in the beginning and can be used to self-equilibrate an arbitrarily chosen coarse space without changing its global effect. It is executed by setting

$$\boldsymbol{C} \leftarrow \boldsymbol{P}_G\boldsymbol{C}\,.$$

An admissible start solution for the Lagrange multipliers, denoted by $\boldsymbol{\lambda}_C$, ensures that $\boldsymbol{\lambda}_G + \boldsymbol{\lambda}_C$ fulfills the self-equilibrium condition and the auxiliary constraint of equation (2.55). It can be computed from the Ansatz

$$\boldsymbol{C}^T\boldsymbol{F}\boldsymbol{\lambda}_C = \boldsymbol{C}^T(\boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}_G) \qquad \text{where} \quad \boldsymbol{\lambda}_C = \boldsymbol{C}\boldsymbol{\mu}$$

and finally reads

$$\boldsymbol{\lambda}_C = \boldsymbol{C}\left(\boldsymbol{C}^T\boldsymbol{F}\boldsymbol{C}\right)^{-1}\boldsymbol{C}^T(\boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}_G)\,.$$

In order to prevent that the adaptions that are made to $\boldsymbol{\lambda}$ during the iterations lead to a violation of equation (2.55), the search directions $\boldsymbol{w}_i$ must be made $\boldsymbol{F}$-conjugate to the auxiliary coarse space, i.e.,

$$\boldsymbol{C}^T\boldsymbol{F}\boldsymbol{w}_i = \boldsymbol{0}$$

must hold for all i. The corresponding correction of $\boldsymbol{w}_i$ reads

$$\boldsymbol{w}_i = \widetilde{\boldsymbol{w}}_i - \boldsymbol{C}\boldsymbol{\mu} \qquad \text{such that} \qquad \boldsymbol{C}^T\boldsymbol{F}\,(\widetilde{\boldsymbol{w}}_i - \boldsymbol{C}\boldsymbol{\mu}) = \boldsymbol{0}$$

where $\widetilde{\boldsymbol{w}}_i$ denotes the original uncorrected search direction. The new admissible search direction $\boldsymbol{w}_i$ is finally computed as

$$\boldsymbol{w}_i = \widetilde{\boldsymbol{w}}_i - \boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{F}\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{F}\widetilde{\boldsymbol{w}}_i\,.$$

This step is usually written as the projection

$$\boldsymbol{w}_i = \boldsymbol{P}_C\widetilde{\boldsymbol{w}}_i \qquad \text{or} \qquad \boldsymbol{w}_i = \boldsymbol{P}_C\boldsymbol{P}_G\boldsymbol{z}_i \tag{2.56}$$

using the projector

$$\boldsymbol{P}_C = \boldsymbol{I} - \boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{F}\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{F}\,.$$

Each time the projector $\boldsymbol{P}_C$ is applied, a problem of the type $(\boldsymbol{C}^T\boldsymbol{F}\boldsymbol{C})\boldsymbol{x} = \boldsymbol{y}$ must be solved. While in all non-projection steps, only neighbors need to exchange information, this projection step introduces global communication within one iteration, similar to the natural coarse space projection. This kind of global communication propagates information throughout the whole domain. It is usually very important for domain decomposition methods to be scalable and almost any FETI method is

Algorithm 2: FETI-2

$\boldsymbol{\lambda}_G = \boldsymbol{Q}\boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{G})^{-1}\boldsymbol{e}$
$\boldsymbol{\lambda}_C = \boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{F}\boldsymbol{C})^{-1}\boldsymbol{C}^T(\boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}_G)$
$\hat{\boldsymbol{\lambda}}_0 = \boldsymbol{0}$
$\boldsymbol{r}_0 = \boldsymbol{P}_G^T(\boldsymbol{d} - \boldsymbol{F}(\boldsymbol{\lambda}_G + \boldsymbol{\lambda}_C))$
$\boldsymbol{z}_0 = \boldsymbol{H}\boldsymbol{r}_0$
$i = 0$
while $r_{*,i} > \epsilon_{r,*}$ **do**
  for $0 \leq j \leq i-1$ **do**
    $\phi_{i,j} = \boldsymbol{q}_j^T\boldsymbol{z}_i \qquad \beta_{i,j} = \frac{\phi_{i,j}}{\Delta_j}$
  $\boldsymbol{w}_i = \boldsymbol{P}_C\boldsymbol{P}_G\boldsymbol{z}_i - \sum_{j=0}^{i-1}\boldsymbol{w}_j\beta_{i,j}$
  $\boldsymbol{q}_i = \boldsymbol{F}\boldsymbol{w}_i$
  $\Delta_i = \boldsymbol{q}_i^T\boldsymbol{w}_i \qquad \gamma_i = \boldsymbol{w}_i^T\boldsymbol{r}_i \qquad \alpha_i = \frac{\gamma_i}{\Delta_i}$
  $\hat{\boldsymbol{\lambda}}_{i+1} = \hat{\boldsymbol{\lambda}}_i + \boldsymbol{w}_i\alpha_i$
  $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \boldsymbol{P}_G^T\boldsymbol{q}_i\alpha_i$
  $\boldsymbol{z}_{i+1} = \boldsymbol{H}\boldsymbol{r}_{i+1}$
  $i \leftarrow i+1$
$m = i$
$\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_G + \boldsymbol{\lambda}_C + \hat{\boldsymbol{\lambda}}_i$

equipped with a projection or some other type of coarse space correction. For simple static problems, the natural coarse space is often sufficient. However, as it is illustrated in the following, dynamic or more ill-conditioned problems require auxiliary coarse spaces.

The resulting iterative algorithm is listed in Algorithm 2. It should be noted that there, the projection is applied before the orthogonalization step, i.e., to $\boldsymbol{z}_i$, which is advantageous in multipreconditioning later.

To solve the auxiliary or coarse problem, either iterative solvers as proposed and analyzed in [47] or direct solvers, discussed and compared in [40, 48], can be used. According to [40], solving the problem by direct solvers is more efficient when the number of subdomains becomes large.

Reconjugation and projection. In several cases, the columns of the matrix $\boldsymbol{C}$ are either naturally already $\boldsymbol{F}$-conjugate or have been orthogonalized in such a way explicitly. In this case, the coarse problem $(\boldsymbol{C}^T\boldsymbol{F}\boldsymbol{C})^{-1}$ becomes diagonal and the calculation of the initial value can be rewritten as

$$\boldsymbol{\lambda}_C = \sum_{j=1}^{N_c}\boldsymbol{c}_j\frac{\boldsymbol{c}_j^T(\boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda})}{\boldsymbol{c}_j^T\boldsymbol{F}\boldsymbol{c}_j} \qquad \text{with} \quad \boldsymbol{C} = \left[\boldsymbol{c}_1 \,\middle|\, \boldsymbol{c}_2 \,\middle|\, \ldots \,\middle|\, \boldsymbol{c}_{N_c}\right]$$

and the projection can be expressed as re-conjugation

$$\boldsymbol{P}_C \boldsymbol{w}_i = \widetilde{\boldsymbol{w}}_i - \sum_{j=1}^{N_c} \boldsymbol{c}_j \frac{\boldsymbol{c}_j^T \widetilde{\boldsymbol{w}}_i}{\boldsymbol{c}_j^T \boldsymbol{F} \boldsymbol{c}_j}. \tag{2.57}$$

In exact arithmetic, both methods lead to equal results, regardless of how the basis vectors $\boldsymbol{c}_j$ are chosen. For both, the solution is computed a priori in the coarse space and the CG algorithm searches for the solution only in the remaining space. The latter is realized by making all search directions $\boldsymbol{F}$-conjugate to the auxiliary coarse space, no matter if this is done by a projection like in equation (2.56) or independent re-conjugation operations for each coarse space basis vector like in equation (2.57). Therefore, deflation by projection and re-conjugation against a set of conjugate directions is mathematically equivalent as long as the $\boldsymbol{c}_i$ basis vectors span the same space. Of course, a projection step with non-conjugate vectors involves a higher computational cost.

Deflation and preconditioning. Deflation is often formulated in terms of a preconditioner. Indeed, a deflated preconditioner

$$\boldsymbol{H}_{C,G} = \boldsymbol{P}_C \boldsymbol{P}_G \boldsymbol{H} \boldsymbol{P}_G^T \boldsymbol{P}_C^T$$

can be defined, which allows to apply a standard preconditioned CG procedure to the system $\boldsymbol{F}\boldsymbol{\lambda} = \boldsymbol{d}$. However, it was noted in [49] that it is reasonable to differ between the application of a classical preconditioner $\boldsymbol{H}$ and a deflation by the projector $\boldsymbol{P}_C$. Both techniques aim to lower the condition number by changing the spectrum of the operator, but only deflation makes the operator become singular, i.e. changes the spectrum by moving parts of it into the kernel. If the operator was already singular before, the size of its kernel is increased by the size of the coarse space.

2.5 Convergence and spectral properties.

Beside other considerations like the necessary amount of communication between substructures and the locality of operations, the performance of the standard or the deflated FETI method can be analyzed by evaluating the computational cost and the rate of convergence of the underlying CG algorithm. At first, this section introduces the spectral properties of the CG algorithm in general and then the special case of its application to the interface problem of FETI.

2.5.1 General Spectral Properties of the Conjugate Gradient Algorithm

The CG algorithm is a Krylov space based method, because the space inside which the solution is sought is a Krylov space, built up during the iterations. For a matrix $\boldsymbol{F}$

and a vector $\boldsymbol{d}$, the Krylov space $\mathscr{K}_j(\boldsymbol{F}, \boldsymbol{d})$ is defined by

$$\mathscr{K}_j(\boldsymbol{F}, \boldsymbol{d}) = \operatorname{span}\left\{\boldsymbol{d}, \boldsymbol{F}\boldsymbol{d}, \boldsymbol{F}^2\boldsymbol{d}, \ldots, \boldsymbol{F}^{j-1}\boldsymbol{d}\right\} .$$

It is identical to the space spanned by the first j search directions of a standard CG algorithm with start value $\mathbf{0}$, applied to solve

$$\boldsymbol{F}\boldsymbol{\lambda} = \boldsymbol{d} .$$

In each iteration, the operator $\boldsymbol{F}$ is applied once more to build the next residual and thus the next search direction. From the definition of the Krylov space, it becomes obvious that in the first directions created, the eigenvectors of the largest eigenvalues will dominate. For better readability, the eigenvectors of the smallest or highest eigenvalues are denominated from here the smallest eigenvectors or largest eigenvectors, respectively. While in the Krylov space as defined above, the domination of the largest eigenvectors will increase for further basis vectors, this is intentionally not the case for the CG algorithm because of the orthogonalization.

Once an eigenvector can be represented by the search directions up to a certain precision, the corresponding eigenvalue is said to be captured. All following search directions are $\boldsymbol{F}$-conjugate to the eigenvector, that is to say, it does not reappear. Consequently, the highest eigenvalues are captured first, and the part of the solution that is composed of the smallest eigenvectors will likely be discovered last, in the final iterations before convergence. This behavior, driven by the distribution of eigenvalues, implies in particular three important consequences.

First, this is the reason why a rapid loss of orthogonality occurs in the presence of large, well-separated eigenvalues as analyzed in [50]. Small numerical errors in the orthogonalization grow exponentially, and the high eigenvectors captured by early search directions come back into later search directions when orthogonalization is carried out only against the most recent one. As such a distribution of eigenvalues is almost always present in larger realistic models of physical structures, full orthogonalization is carried out in every iteration in all algorithms considered in this work. This was also suggested for domain decomposition methods in [51]. Of course, only eigenvectors triggered by the right-hand side $\boldsymbol{d}$ will occur, but the right-hand sides $\boldsymbol{d}$ of the interface problems of structural mechanics usually contain significant parts of all higher eigenvectors of $\boldsymbol{F}$.

Second, when an eigenvalue was captured with a certain precision and $\boldsymbol{F}$-conjugacy of search directions is successfully maintained, the CG algorithm behaves as if it was not present in the spectrum. This becomes clear when recalling the equivalence between deflation and reconjugation. It does not make a difference if the CG algorithm keeps iterating, or if the search directions built so far are used as coarse space basis vectors within a newly started deflated CG algorithm on the same problem.

Finally, in the direction of nearly equal or close eigenvalues, which are then said to be clustered, an accurate solution is found after very few iterations. In fact, if the operator has only n distinct eigenvalues, a standard CG algorithm finds the exact solution in at most n iterations, presuming exact arithmetic [52]. In contrast, the worst case is an evenly distributed spectrum. In this case, for an arbitrary right-hand

side, almost as many iterations as dof are required to achieve a reasonable accuracy. Summarized, the rate of convergence is not only dependent on the ratio between the largest and the smallest eigenvalue, but also heavily on the distribution of all eigenvalues in between. Several specific types of distributions and their influence on the real rate of convergence were analyzed in [53].

Nevertheless, the smallest and the largest eigenvalue of $\boldsymbol{F}$ can be used to formulate a worst case scenario of the real convergence of a CG algorithm, employed to solve the example linear system of equations $\boldsymbol{F}\boldsymbol{\lambda} = \boldsymbol{d}$. Let the eigenvalues of $\boldsymbol{F}$ be denoted by $\Theta_{\min}(\boldsymbol{F}), \ldots, \Theta_i(\boldsymbol{F}), \ldots, \Theta_{\max}(\boldsymbol{F})$ when sorted in ascending order. The condition number of $\boldsymbol{F}$ is defined as

$$\kappa(\boldsymbol{F}) = \frac{\Theta_{\max}(\boldsymbol{F})}{\Theta_{\min}(\boldsymbol{F})}$$

and the error at start and the error after iteration i as

$$\boldsymbol{e}_0 = \hat{\boldsymbol{\lambda}}_0 - \hat{\boldsymbol{\lambda}} \qquad \text{and} \qquad \boldsymbol{e}_i = \hat{\boldsymbol{\lambda}}_i - \hat{\boldsymbol{\lambda}},$$

respectively. Then, the upper bound for the error at iteration i is described by

$$\|\boldsymbol{e}_i\|_{\boldsymbol{F}} \leq \|\boldsymbol{e}_0\|_{\boldsymbol{F}}\, 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i, \tag{2.58}$$

i.e., at iteration i, the error has decreased by a factor of at least $2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i$. [37]

2.5.2 Spectral properties of the FETI operator.

The unpreconditioned FETI operator $\boldsymbol{F}$ and the preconditioned FETI operator $\boldsymbol{HF}$ exhibit a number of beneficial properties for the application of the CG algorithm. In particular, the preconditioning leads to a bounded and clustered spectrum at the lower end, and the property of the CG algorithm to capture the highest eigenvalues first is exploited optimally by the dual formulation, based on flexibility operators instead of stiffnesses.

Spectral Bounds of the FETI Operator

The undeflated standard FETI procedure as listed in algorithm 1 to find $\hat{\boldsymbol{\lambda}}$ is mathematically equivalent to applying a standard CG algorithm to

$$\boldsymbol{P}_G\boldsymbol{H}\boldsymbol{P}_G^T\boldsymbol{F}\hat{\boldsymbol{\lambda}} = \boldsymbol{P}_G\boldsymbol{H}\boldsymbol{P}_G^T(\boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}_G). \tag{2.59}$$

It becomes obvious from equation (2.59) that the spectral properties of the preconditioned FETI operator $\boldsymbol{P}_G\boldsymbol{H}\boldsymbol{P}_G^T\boldsymbol{F}$ define the convergence of the preconditioned FETI method.

For the spectrum of this operator, useful bounds have been proven for the case of the Dirichlet preconditioner $\boldsymbol{H} = \boldsymbol{H}_D$. They depend on the mesh and the decomposition

into substructures. The smallest and largest eigenvalues of $\boldsymbol{P}_G\boldsymbol{H}\boldsymbol{P}_G^T\boldsymbol{F}$ are bounded such that

$$1 \leq \mathrm{eig}(\boldsymbol{P}_G\boldsymbol{H}\boldsymbol{P}_G^T\boldsymbol{F}) \leq C\left(1+\log\left(\frac{H}{h}\right)\right)^2 \tag{2.60}$$

under the requirement that $\boldsymbol{Q}$ is chosen as the preconditioner $\boldsymbol{H}$. A first similar proof was published in [54], and it was improved in [55] in the way that the exponent could be lowered from 3 to 2. In the definition of the upper bound of the spectrum in equation (2.60), h is a measure for the typical size of one element and H measures the typical size, e.g., the diameter, of one substructure. Furthermore, C is a constant that is independent of h and H. On the one hand, it follows that the eigenspectrum of the preconditioned operator is bounded from below by one. On the other hand, the upper bound proved numerical scalability with respect to the mesh size h and the substructure size H. Iterative domain decomposition algorithms that satisfy this property are said to be optimal, which is why the Dirichlet preconditioner is also called an optimal preconditioner for FETI.

Typical Eigenvalue Distribution

An important aspect of the FETI method and its success are its specific spectral properties in terms of the eigenvalue distribution and the correspondence of the high and low eigenvalues to certain physical modes. In a finite element discretized problem of structural dynamics, the eigenvectors that correspond to the smallest eigenvalues are usually the most important ones, because they correspond to global, physical modes. In contrast, the highest eigenvalues correspond to very local, unphysical modes. The finer the mesh, the more numerous and higher these eigenfrequencies of the upper part of the spectrum become. Unfortunately, as described above, iterative algorithms that are based on building Krylov spaces capture the highest and thereby the most unimportant modes first, which leads to poor convergence.

The dual FETI operator is instead built from the inverses of the substructure stiffnesses. As a result, its highest frequencies correspond to slow physical modes and its smallest frequencies to local, unphysical modes of the mesh. When the mesh is refined, the slow physical modes consolidate in the upper part of the spectrum and the ever growing number of mesh modes accumulates to zero. As it was described above, preconditioning the FETI operator shifts up the lower part of the spectrum such that it is bounded from below by one. The result is that the solution in the high physical modes is captured first with fast convergence during the iterations, undisturbed from the numerous mesh modes. Consequently, a good approximation of the global shape is found in very few iterations. This difference in the typical distribution of eigenvalues between the primal and the dual Schur complement method and the advantageous properties of the dual one concerning the convergence of a CG algorithm were analyzed in [56].

When solving problems of structural mechanics, it is typical for the FETI algorithm that a few high, well-separated eigenvalues are captured by almost one iteration per eigenvalue. Subsequently, due to the massive clustering of the smallest eigenvalues

at and above one, these are captured all together in the very last, few iterations. As extensively studied in [57], this leads to an increase of the convergence rate, also called superconvergent behavior, in which the number of iterations until convergence can be much less then predicted by the classical estimate of equation (2.58).

Active Spectrum

For a very large, real engineering problem, it may be impossible to compute and analyze its spectrum. In this case, the active spectrum, i.e., the part of the spectrum that needed to be captured to reach convergence, can be recovered after the iterative solution. This is discussed in Chapter 4. However, in case the complete spectrum should be computed by eigenanalysis for small academic examples, it is also necessary to distinguish between relevant and irrelevant parts of the spectrum. Not relevant for the active spectrum are zero eigenvalues that emanate from the well-chosen core of the possibly deflated or preconditioned operator. The zero eigenvalues can be ambiguous due to numerical precision and in order to handle them appropriately, at least their number should be known in advance. Subsequently, the corresponding smallest eigenvalues should be discarded. Of course, a check for plausibility must be implemented. The dimension of the core of a deflated, possibly preconditioned interface operator, i.e., of $\boldsymbol{P}_C\boldsymbol{P}_G\boldsymbol{H}\boldsymbol{P}_G^T\boldsymbol{P}_C^T\boldsymbol{F}$, is equal to the exact number of zero eigenvalues and is computed as

$$\operatorname{rank}\left(\ker\left(\boldsymbol{P}_C\boldsymbol{P}_G\boldsymbol{H}\boldsymbol{P}_G^T\boldsymbol{P}_C^T\boldsymbol{F}\right)\right) = N_{\lambda,\text{redundant}} + N_{\text{rbm}} + N_c$$

where $N_{\lambda,\text{redundant}}$ is the total number of redundant Lagrange multipliers defined in equation (2.43), N_{rbm} the total number of rigid body modes of floating substructures, and N_c the size of the auxiliary coarse space.

Ill-Conditioning

The high part of the spectrum of $\boldsymbol{F}$ is obviously not composed directly by mode shapes of the global structure. Instead, it is a combination of the smooth physical modes of the substructures. This circumstance, introduced by the decomposition of the domain, can be the origin of severe ill-conditioning. If a single substructure exhibits smooth modes, i.e., modes with large displacements at very little force, which are at the same time almost non-existent in the global structure, these modes can appear as very high eigenvalues at the upper end of the spectrum of $\boldsymbol{F}$. Because they emanate solely from the decomposition and do not represent mode shapes of the global structure, they severely impair the rate of convergence. In the context of FETI, they are called *bad modes*.

From a numerical point of view, it happens due to many high, often well-separated eigenvalues that the CG algorithms has to capture. From a physical point of view, it is the discrepancy in behavior between local substructures and the assembled, global structure. This type of ill-conditioning occurs most regularly when the coefficients of the underlying partial differential equation, for example, the stiffness, exhibit strong

jumps along interfaces. While jumps across interfaces can be handled well by the scaling procedures described in section 2.3.2, jumps along the interface require additional attention and are more difficult to address.

2.5.3 Desireable Coarse Spaces

An auxiliary coarse space should satisfy essentially two requirements. On the one hand, it should be as small as possible and on the other hand, chosen such that the convergence of the CG algorithm, which shall find the solution in the remaining space, is as fast as possible.

Already in [44], as one of the first publications considering a deflation of the CG algorithm, certain eigenvectors were identified to compose an ideal coarse space. Because the problems addressed in [44] were based on stiffness operators, the smallest eigenvalues have been considered for an efficient coarse space. In dual domain decomposition, because of the flexibility based operator, the higher eigenvalues reflect smooth, physical modes and possibly originate from local bad modes, while the smallest ones are clustered towards one and correspond to less important mesh modes. Thus, ill-conditioning can only result from the upper part of the spectrum, and the ideal coarse space for FETI is built from the highest eigenvectors of $\boldsymbol{F}$. When they are removed from the solution process by deflation or re-conjugation, the CG algorithm behaves as if they were not present in the spectrum and the effective condition number is lowered significantly. The iterative process starts immediately to capture the strongly clustered smaller eigenvalues with fast convergence. Of course, computing the high eigenvectors or bad modes of $\boldsymbol{F}$ directly is not possible in practical applications. It is thus the aim to construct auxiliary coarse spaces that approximate the bad modes with the highest possible accuracy at the lowest possible cost.

2.6 Generalized Eigenvalues in the Overlaps

An explicit method to construct an auxiliary coarse space for FETI, designed to discover the bad modes for a given decomposition and a given preconditioner, was proposed in [58]. It was originally developed within the overlapping Schwarz framework and is based on solving local generalized eigenvalue problems. It is denominated GenEO. When its solutions are employed as coarse space, it effectively removes the high extremal part of the spectrum in the deflated, preconditioned operator.

For every substructure s, the generalized eigenproblem reads

$$\boldsymbol{S}_K^{(s)}\boldsymbol{y}^{(s)} = \theta^{(s)}\boldsymbol{B}^{(s)^T}\boldsymbol{H}\boldsymbol{B}^{(s)}\boldsymbol{y}^{(s)} \tag{2.61}$$

where $\theta^{(s)}$ is called a GenEO eigenvalue and $\boldsymbol{y}^{(s)}$ a GenEO eigenvector. The solutions $(\theta_1^{(s)}, \boldsymbol{y}_1^{(s)}), (\theta_2^{(s)}, \boldsymbol{y}_2^{(s)}), \ldots$ are sorted in ascending order of the GenEO eigenvalues $\theta_i^{(s)}$.

The coarse space is then chosen as

$$\boldsymbol{C} = \left[\boldsymbol{C}^{(1)} \middle| \boldsymbol{C}^{(2)} \middle| \ldots \middle| \boldsymbol{C}^{(N_s)}\right] \qquad \boldsymbol{C}^{(s)} = \left[\boldsymbol{H}\boldsymbol{B}^{(s)}\boldsymbol{y}_1^{(s)} \middle| \boldsymbol{H}\boldsymbol{B}^{(s)}\boldsymbol{y}_2^{(s)} \middle| \ldots \middle| \boldsymbol{H}\boldsymbol{B}^{(s)}\boldsymbol{y}_{k^s}^{(s)}\right].$$

An important property is its locality. That means that on the one hand, each substructure s must solve its own eigenproblem of equation (2.61), which is a completely local problem once the substructure has the matrix ${\boldsymbol{B}^{(s)}}^T\boldsymbol{H}\boldsymbol{B}^{(s)}$ available. On the other hand, though the global preconditioner in present here, the computation of the matrix ${\boldsymbol{B}^{(s)}}^T\boldsymbol{H}\boldsymbol{B}^{(s)}$ is still limited to neighbors. That becomes obvious when expressing the preconditioner as sum of local contributions. The right-hand side of the GenEO eigenproblem for substructure r then reads in case of the Dirichlet preconditioner

$${\boldsymbol{B}^{(r)}}^T\boldsymbol{H}\boldsymbol{B}^{(r)} = \sum_{s=1}^{N_s} {\boldsymbol{B}^{(r)}}^T\boldsymbol{B}^{(s)}\boldsymbol{S}_K^{(s)}{\boldsymbol{B}^{(s)}}^T\boldsymbol{B}^{(r)}.$$

All terms for which s is not a neighbor of r vanish because ${\boldsymbol{B}^{(s)}}^T\boldsymbol{B}^{(r)} = \boldsymbol{0}$ in case that the substructures s and r do not share any interface node. As a result, only neighbors need to communicate, for example, by exchanging their local operators $\boldsymbol{S}_K^s$ to prepare solving the local eigenproblems.

An important issue to be addressed is which GenEO vectors $\boldsymbol{y}_i^{(s)}$ should be considered to construct the contribution of substructure s to the coarse space. First of all, the GenEO vectors $\boldsymbol{y}_i^{(s)}$ with the smallest GenEO values form together an approximation of the eigenvectors of $\boldsymbol{F}$ with the highest eigenvalues and should thus be used to form the coarse space. However, for static problems, the core of $\boldsymbol{S}_K^{(s)}$ is found as solutions $\boldsymbol{y}^{(s)}$ with zero GenEO eigenvalues $\theta^{(s)}$, in case of floating substructures. These zero energy modes may not be included in the auxiliary coarse space, as they were already considered in the natural coarse space by the projector $\boldsymbol{P}_G$. It remains to determine the number of basis vectors $k^{(s)}$ that should be contributed by a substructure s.

An appropriate global threshold for θ can be found by setting an explicit limit on the size of the auxiliary coarse space. Another approach that is known to be very efficient is to search for a significant jump in the GenEO eigenvalues $\theta_i^{(s)}$ of each substructure when sorted in ascending order. The matrix $\boldsymbol{C}^{(s)}$ of substructure s is then constructed by considering all eigenvalues before the jump.

Chapter 3

Application to Dynamic Elasticity

This section covers all important aspects that should be considered when applying the method of finite element tearing and interconnecting to solve linear problems of structural dynamics. First, the corresponding equations of motion are presented and their decomposition in non-overlapping substructures, according to the theory of dual domain decomposition. The requirements on a reasonable step-by-step time integration method are explained and several possible approaches are presented. Furthermore, the specific requirements of the constrained system of substructure equations, which must be solved each time step, are introduced. After the decomposed linear system to be solved has been derived, the corresponding interface problem is constructed and the common techniques of preconditioning and scaling are transferred from the static case. Further issues like the computation of the initial acceleration and the iterative solution within the FETI framework are covered. Also the specific spectral properties of the interface problem in the dynamic case are highlighted. The chapter is closed by an overview of the basic principles that could be used to accelerate the iterative solution procedure.

FETI was already applied earlier to structural dynamics, e.g., in [59], where it was employed to solve the linear systems occurring during Newton-Raphson iterations, solving a non-linear problem. No coarse space correction is introduced automatically in the space of rigid body modes as it happens in static problems. Accordingly, no coarse space technique was applied in [59].

The introduction of an auxiliary or artificial coarse space correction when applying the FETI method to dynamic problems in order to maintain scalability was proposed in [45]. While the rigid body modes possess energy and thus do not cause any singularities in dynamics anymore, they can still be computed and used as coarse space. The corresponding results in [45] showed reasonable improvements in scalability and performance.

3.1 Linear Dynamic Elasticity

An inertia term, considering the acceleration $\ddot{\overline{\boldsymbol{u}}}$, is added to the weak form of the equilibrium in initial configuration, which reads

$$G(\overline{\boldsymbol{u}},\overline{\boldsymbol{v}}) = \int_\Omega \overline{\boldsymbol{S}}(\overline{\boldsymbol{u}}) : \delta\overline{\boldsymbol{E}}(\overline{\boldsymbol{u}},\overline{\boldsymbol{v}})\,\mathrm{d}\Omega - \int_\Omega \overline{\boldsymbol{b}}\cdot\overline{\boldsymbol{v}}\,\mathrm{d}\Omega + \int_\Omega \rho_0\ddot{\overline{\boldsymbol{u}}}\cdot\overline{\boldsymbol{v}}\,\mathrm{d}\Omega - \int_{\Gamma_N} \overline{\boldsymbol{t}}\cdot\overline{\boldsymbol{v}}\,\mathrm{d}\Gamma$$

$$= 0\,.$$

For the definition of the variation of the Green-Lagrange strain $\delta\overline{\boldsymbol{E}}(\overline{\boldsymbol{u}},\overline{\boldsymbol{v}})$, see section 1.4. Similar to the stiffness matrix in section 1.4, the mass matrix is defined by

$$\boldsymbol{v}^T\boldsymbol{M}\ddot{\boldsymbol{u}} = \boldsymbol{v}^T\int_\Omega \rho_0\boldsymbol{N}^T\boldsymbol{N}\,\mathrm{d}\Omega\,\ddot{\boldsymbol{u}}\,.$$

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

$$\boldsymbol{M}\ddot{\boldsymbol{u}}(t) + \boldsymbol{K}\boldsymbol{u}(t) = \boldsymbol{f}_{\text{ext}}(t)\,. \tag{3.1}$$

The structure is loaded by an arbitrary time dependent force $\boldsymbol{f}_{\text{ext}}(t)$. The values of the initial displacement $\boldsymbol{u}(0)$ and the initial velocity $\dot{\boldsymbol{u}}(0)$ are given. To simplify the formulas, the time dependency of $\ddot{\boldsymbol{u}}$, $\boldsymbol{u}$ and $\boldsymbol{f}_{\text{ext}}$ is omitted from here on. After decomposition, the local substructure equations read

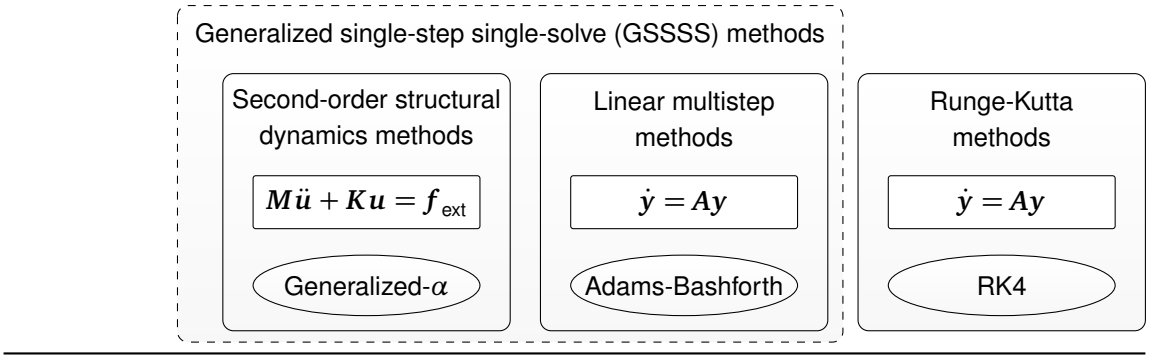
$$\boldsymbol{M}^{(s)}\ddot{\boldsymbol{u}}^{(s)} + \boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)} + \boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda} = \boldsymbol{f}_{\text{ext}}^{(s)}\,.$$

Naturally, the compatibility constraints are imposed on the displacements. However, according to [60], it is equivalent to impose the constraints on the velocities or accelerations. The possible constraints then write

$$\sum_{s=1}^{N_s}\boldsymbol{B}^{(s)}\boldsymbol{t}^{(s)}\boldsymbol{u}^{(s)} = \boldsymbol{0} \qquad \sum_{s=1}^{N_s}\boldsymbol{B}^{(s)}\boldsymbol{t}^{(s)}\dot{\boldsymbol{u}}^{(s)} = \boldsymbol{0} \qquad \sum_{s=1}^{N_s}\boldsymbol{B}^{(s)}\boldsymbol{t}^{(s)}\ddot{\boldsymbol{u}}^{(s)} = \boldsymbol{0}\,.$$

3.2 Time Integration of Constrained Dynamical Problems

To compute the behavior of the structure over time, it is necessary to discretize the differential equation of motion in time. This is usually achieved by applying a step-by-step integration method, which allows to compute the solutions for the discretized points in time subsequently one after the other. In this section, an overview over the relevant methods is given and unconditionally stable, implicit one-step algorithms are justified as a reasonable choice. Furthermore, the specific requirements in the case of finite element tearing and interconnecting as a constrained problem are addressed.



Generalized-α	✔ (native)	✔(three-step)	✘
Newmark-β	✔ (native)	✔ (two-step)	✘
trapezoidal rule	✔	✔ (one-step)	✔
forward Euler	✘	✔ (one-step)	✔
backward Euler	✘	✔ (one-step)	✔

Figure 3.1. Three boxes at the top of the figure: Selected main classifications of time integration methods, the differential equation they can be applied to, and an exemplary, representative method for which the class is well known. Table below: Further examples of the discussed, most relevant and most widely known time integration methods and into which classifications they can be casted equivalently.

3.2.1 Overview and Classification

When looking for a time integration method, one can choose from an extensive variety of algorithms that exhibit highly different properties regarding stability, accuracy, computational complexity, and for which type of differential equation they fit best. It is thereby essential to know the specific needs of the type of problem considered here, which is mainly described as second-order differential equation with additional constraints. This section describes the physically motivated requirements on the accuracy of the solution and how the demand to invest the least possible computational effort then indicates the optimal properties of a well suited method.

Figure 3.1 presents an overview of the most important classes of step-by-step integration methods, namely linear multistep (LMS) methods, generalized single-step single-solve (GSSSS) methods and Runge-Kutta (RK) methods. Furthermore, it names a representative example for each class like the widely used fourth-order Runge-Kutta scheme, mostly known as RK4. The vast majority of integration schemes, especially the most simple ones like the trapezoidal rule, fit into either one or both of the traditional LMS or RK classes.

Recently, another unified framework of time integration algorithms was proposed in [61, 62], the GSSSS algorithms. They encompass the complete range of LMS methods as well as the one-step methods that are specifically designed for the solution of second-order differential equations of structural dynamics. The authors prove that practically all known single-step methods for second-order structural dynamics, like those of the generalized-α family, are equivalent to LMS methods. This is why funda-

mental findings on LMS methods, like the Dahlquist barriers, are valid for the single-step representations of the methods as well. The lower part of figure 3.1 lists some of the most relevant algorithms as well as those employed in this work and shows how they fit into the three main classes of step-by-step time integration methods. For example, the generalized-α family can be written as a three-step LMS method [63] and the Newmark family, itself a special case of generalized-α, as a two-step LMS method [64–66].

3.2.2 Requirements

The basic requirements on a time integration method for dynamic mechanical systems have been listed [67, 68] and were rendered to the needs and the context of this work in the following. Every method to be considered should exhibit or provide

1. Unconditional stability
2. Second-order accuracy
3. Solution of only one implicit system of the size of the mass matrix per time step
4. (User-controllable numerical dissipation in the high frequency regime without affecting lower modes too strongly)

Unconditional stability

The first important distinction to be made is if wave propagation effects or just the global structural response are of interest. In the majority of problems and those considered in this work, the global structural response is of interest. In these cases, usually only the lowest modes are excited in the physical system and contribute to the total structural response. Accordingly, those low modes are the only ones of interest in the FEM model and should be reproduced with an appropriate accuracy. [7]

The accuracy of low mode representation can be measured well by period elongation and amplitude decay and is mainly dependent on the size of the time step. For example, the trapezoidal rule leads to approximately 3% period elongation and no amplitude decay for a mode with period T when the time step is chosen as $\Delta t = \frac{T}{10}$. All modes with longer periods are represented more accurately. The period elongation improves to about 1% for $\Delta t = \frac{T}{15}$. [69]

This approach for determining a reasonable time step size has been used in [45], where a time step size of $\frac{T_3}{15}$ was used to analyze a diffraction grating system. In this example, T_3 denotes the period of the third lowest mode. Consequently, the first three modes of the structure were represented at an accuracy of at least 1%. In [59] and [70], time steps of $\frac{T_6}{10}$ and $\frac{T_6}{15}$, respectively, have been used to compute the transient dynamics of plates discretized by shell elements using implicit integration schemes.

The stable regime of conditionally stable algorithms is confined by a time step size that is inversely proportional to the highest frequency ω_{max} of the FEM model. It can be estimated from above by the highest eigenfrequency of an isolated element [33] and thus increases for a finer mesh, while the low modes and with them the desired time step for solution accuracy do not change. If, for example, the central difference scheme is used, the time step size must satisfy the relation

$$\Delta t \leq \frac{2}{\omega_{max}},$$

which is known as the Courant-Friedrichs-Lewy condition [71]. It relates the wave propagation speed to the mesh size. As a consequence, conditionally stable algorithms often require time steps that are many orders of magnitude smaller than actually needed for the desired solution accuracy. Therefore, for the considered type of problem, an unconditionally stable algorithm is recommended as the most effective and can save a substantial amount of computational cost. [7, 67, 68]

Implicit methods and second-order accuracy

According to [72], no explicit LMS method exists that is unconditionally stable. Furthermore, the order of accuracy of implicit and unconditionally stable LMS methods cannot exceed 2. These statements are also known as the second Dahlquist barrier. While this result confines the possible accuracy of these methods from above, it is widely known by experience that second-order accurate methods exhibit strong superiority to first-order accurate methods [68]. It follows that second-order accuracy is a reasonable property to aim for.

Numerical damping

The spatial discretization in FEM models introduces artificial modal components, so-called spurious high frequency oscillations. They can affect the total response, especially in problems where non-linear material behavior is considered. Numerical dissipation is used to damp out these components of the response, where asymptotic annihilation refers to numerical dissipation that eliminates high-frequency modes after one time step. Of course, the numerical dissipation should influence the low-mode response as little as possible. In many applications, numerical damping is not necessary and in this case, a wider range of time integration methods becomes available. [66–68]

Review of the three main classifications

RK methods fail to fulfill the third requirement stated earlier, as they require the solution of more than one implicit system or of larger systems than the mass matrix. This is due to the multi-stage character of RK methods and due to the fact that any differential equation must be transformed into a first-order system, also known as

state-space form. This leads to the storage of at least twice as many variables, for example, when employing the well-known fourth-order Runge-Kutta (RK4) scheme [8]. Of course, equation (3.1) can be transformed into the state-space form, yet the RK and LMS methods are not able to exploit all the advantages of the specific form of the second-order differential equations [33].

Instead, another family of integration schemes is directly applicable to second-order differential equations. The Newmark method was the first basic procedure of this kind and was originally introduced in [73]. While there exist other step-by-step time integration methods that were specifically designed to handle second-order differential equations describing structural dynamics problems, the generalized-α family is one of the most interesting. It includes besides Newmark the most widely known and used methods. Generalized-α fulfills all of the requirements stated at the beginning of this section, including numerical damping up to asymptotic annihilation that barely affects low frequencies.

The original Newmark family itself is of minor interest, as it loses second-order accuracy as soon as numerical damping is introduced [68] and the numerical damping significantly affects the low modes [74]. If numerical damping is not necessary, the trapezoidal rule, of course a special case of the Newmark family, offers second-order accuracy with the lowest truncation error of all unconditionally stable LMS methods and no amplitude decay independently of the time step size [7, 68, 72]. Thus, without the necessity of numerical damping, it is best to go with the trapezoidal rule. Only in case that numerical damping is required, generalized-α must be used. However, problem specific tuning is often necessary for efficient numerical damping, which is why it is reasonable to stick with the trapezoidal rule unless this effort shall be made.

3.2.3 Generalized-α

The generalized-α was originally published in [74] and is basically a combination of the HHT-α method [67] and the WBZ-α method [75], both building upon the foundation of Newmark's integration rules.

Let $t^1, \ldots, t^n, \ldots$ be equidistant discrete points in time with a time step size defined as

$$\Delta t = t^{n+1} - t^n$$

where n denotes the current time step index. The quantities $\ddot{\boldsymbol{u}}^{(s),n}$, $\dot{\boldsymbol{u}}^{(s),n}$, and $\boldsymbol{u}^{(s),n}$ are approximations of $\ddot{\boldsymbol{u}}^{(s)}(t^n)$, $\dot{\boldsymbol{u}}^{(s)}(t^n)$, and $\boldsymbol{u}^{(s)}(t^n)$, respectively. The basic idea of the so-called α-methods is that not only the differential equation at one specific point in time is evaluated, but a linear combination of it at the current time n and the next time step $n+1$. The weights for inertia terms are denoted by α_m and can be different from the weights of the remaining terms denoted by α_f. To this end, the

abbreviations

$$\begin{aligned}
\ddot{\boldsymbol{u}}^{(s),n+1-\alpha_m} &= (1-\alpha_m)\ddot{\boldsymbol{u}}^{(s),n+1} + \alpha_m \ddot{\boldsymbol{u}}^{(s),n} \\
\boldsymbol{u}^{(s),n+1-\alpha_f} &= (1-\alpha_f)\boldsymbol{u}^{(s),n+1} + \alpha_f \boldsymbol{u}^{(s),n} \\
\boldsymbol{\lambda}^{n+1-\alpha_f} &= (1-\alpha_f)\boldsymbol{\lambda}^{n+1} + \alpha_f \boldsymbol{\lambda}^{n} \\
\boldsymbol{f}_{\text{ext}}^{(s),n+1-\alpha_f} &= (1-\alpha_f)\boldsymbol{f}_{\text{ext}}^{(s),n+1} + \alpha_f \boldsymbol{f}_{\text{ext}}^{(s),n}
\end{aligned} \tag{3.2}$$

are introduced. The resulting modified balance equation then reads

$$\boldsymbol{M}^{(s)}\ddot{\boldsymbol{u}}^{(s),n+1-\alpha_m} + \boldsymbol{K}^{(s)}\boldsymbol{u}^{(s),n+1-\alpha_f} + \boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}^{n+1-\alpha_f} = \boldsymbol{f}_{\text{ext}}^{(s),n+1-\alpha_f}. \tag{3.3}$$

The classical Newmark scheme is employed as implicit integration rule to compute displacement and velocity increments from given accelerations of the next time step. They provide

$$\begin{aligned}
\dot{\boldsymbol{u}}^{(s),n+1} &= \dot{\boldsymbol{u}}^{(s),n} + \Delta t\left((1-\gamma_N)\ddot{\boldsymbol{u}}^{(s),n} + \gamma_N \ddot{\boldsymbol{u}}^{(s),n+1}\right) \\
\boldsymbol{u}^{(s),n+1} &= \boldsymbol{u}^{(s),n} + \Delta t\dot{\boldsymbol{u}}^{(s),n} + \Delta t^2\left(\left(\frac{1}{2}-\beta_N\right)\ddot{\boldsymbol{u}}^{(s),n} + \beta_N \ddot{\boldsymbol{u}}^{(s),n+1}\right).
\end{aligned} \tag{3.4}$$

Second-order accuracy and unconditional stability. As it was shown in [74], the generalized-α is second-order accurate for a structural differential equation without constraints if

$$\gamma_N - 0.5 - (\alpha_f - \alpha_m) = 0$$

holds and it is unconditionally stable for

$$\alpha_m \leq \alpha_f \leq \frac{1}{2} \qquad \beta_N \geq \frac{1}{4} + \frac{1}{2}(\alpha_f - \alpha_m). \tag{3.5}$$

Numerical dissipation. In [74], the authors of generalized-α further describe the optimal choice of the parameters. First, to maximize the high-frequency damping and minimize the low-frequency damping,

$$\beta_N - \frac{1}{4}(1+\alpha_f-\alpha_m)^2 = 0$$

should hold. Second, the user can set the spectral radius of the algorithm at infinity $\rho_\infty \in [0,1]$ and compute the weights subsequently from

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1} \qquad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1}.$$

A spectral radius of $\rho_\infty = 0$ leads to maximal high-frequency damping, i.e., the case of asymptotic annihilation. At the other end, $\rho_\infty = 1$ means that no numerical dissipation is present, that is to say, the algorithm is undamped. If no numerical damping should be used, the trapezoidal rule can provide the smallest truncation error and should be considered as simple alternative.

Iteration or stepping matrix. To receive the final, linear, constrained system of equations that needs to be solved for the accelerations of time t_{n+1}, the interpolated quantities of equation (3.2) and the Newmark integration rules in equation (3.4) are inserted into the modified balance equation (3.3). This yields the effective structural equation

$$\begin{aligned}
&\boldsymbol{M}^{(s)}\left((1-\alpha_m)\ddot{\boldsymbol{u}}^{(s),n+1}+\alpha_m\ddot{\boldsymbol{u}}^{(s),n}\right)\\
&+\boldsymbol{K}^{(s)}\left((1-\alpha_f)\left(\boldsymbol{u}^{(s),n}+\Delta t\dot{\boldsymbol{u}}^{(s),n}+\Delta t^2\left((0.5-\beta_N)\ddot{\boldsymbol{u}}^{(s),n}+\beta_N\ddot{\boldsymbol{u}}^{(s),n+1}\right)\right)+\alpha_f\boldsymbol{u}^{(s),n}\right)\\
&\quad=(1-\alpha_f)\boldsymbol{f}_{\text{ext}}^{(s),n+1}+\alpha_f\boldsymbol{f}_{\text{ext}}^{(s),n}-\boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}^{n+1-\alpha_f}
\end{aligned}$$

which can be rearranged as

$$\begin{aligned}
&\left(\boldsymbol{M}^{(s)}(1-\alpha_m)+\boldsymbol{K}^{(s)}(1-\alpha_f)\Delta t^2\beta_N\right)\ddot{\boldsymbol{u}}^{(s),n+1}\\
&\quad=(1-\alpha_f)\boldsymbol{f}_{\text{ext}}^{(s),n+1}+\alpha_f\boldsymbol{f}_{\text{ext}}^{(s),n}-\boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}^{n+1-\alpha_f}-\boldsymbol{M}^{(s)}\alpha_m\ddot{\boldsymbol{u}}^{(s),n}\\
&\quad-\boldsymbol{K}^{(s)}(1-\alpha_f)\left(\boldsymbol{u}^{(s),n}+\Delta t\dot{\boldsymbol{u}}^{(s),n}+\Delta t^2(0.5-\beta_N)\ddot{\boldsymbol{u}}^{(s),n}+\alpha_f\boldsymbol{u}^{(s),n}\right)
\end{aligned}$$

so that the acceleration terms are summarized on the left-hand side, and the right-hand side is a known force vector if the Lagrange multipliers are given. To obtain a shorter notation, abbreviations for the left- and right-hand side are introduced. The matrix

$$\boldsymbol{D}_{\text{g}\alpha}^{(s)}=(1-\alpha_m)\boldsymbol{M}^{(s)}+(1-\alpha_f)\Delta t^2\beta_N\boldsymbol{K}^{(s)} \tag{3.6}$$

is called iteration matrix [33] or, to distinguish it from the iterative process of the CG algorithm, stepping matrix. For the right-hand side, the generalized force vector

$$\begin{aligned}
\boldsymbol{g}_{\text{g}\alpha}^{(s),n+1}&=(1-\alpha_f)\boldsymbol{f}_{\text{ext}}^{(s),n+1}+\alpha_f\boldsymbol{f}_{\text{ext}}^{(s),n}-\boldsymbol{M}^{(s)}\alpha_m\ddot{\boldsymbol{u}}^{(s),n}\\
&\quad-\boldsymbol{K}^{(s)}(1-\alpha_f)\left(\boldsymbol{u}^{(s),n}+\Delta t\dot{\boldsymbol{u}}^{(s),n}+\Delta t^2(0.5-\beta_N)\ddot{\boldsymbol{u}}^{(s),n}+\alpha_f\boldsymbol{u}^{(s),n}\right)
\end{aligned} \tag{3.7}$$

is defined. Injecting them into the rearranged effective structural equation results in the short notation

$$\boldsymbol{D}_{\text{g}\alpha}^{(s)}\ddot{\boldsymbol{u}}^{(s),n+1}=\boldsymbol{g}_{\text{g}\alpha}^{(s),n+1}-\boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}^{n+1-\alpha_f}\,.$$

Newmark. For $\alpha_m=\alpha_f=0$, the Newmark method is obtained for which the stepping matrix $\boldsymbol{D}$ and the generalized force vector $\boldsymbol{g}$ simplify to

$$\boldsymbol{D}_{\text{N}}^{(s)}=\boldsymbol{M}^{(s)}+\Delta t^2\beta_N\boldsymbol{K}^{(s)} \tag{3.8}$$

$$\boldsymbol{g}_{\text{N}}^{(s),n+1}=\boldsymbol{f}_{\text{ext}}^{(s),n+1}-\boldsymbol{K}^{(s)}\left[\boldsymbol{u}^{(s),n}+\Delta t\dot{\boldsymbol{u}}^{(s),n}+\Delta t^2(0.5-\beta_N)\ddot{\boldsymbol{u}}^{(s),n}\right]. \tag{3.9}$$

The short form of the structural equation reads in this case

$$\boldsymbol{D}_{\text{N}}^{(s)}\ddot{\boldsymbol{u}}^{(s),n+1}=\boldsymbol{g}_{\text{N}}^{(s),n+1}-\boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}^{n+1}\,.$$

Trapezoidal rule. The Newmark method without numerical damping is the same as the so-called implicit trapezoidal rule, also known as constant average acceleration method or Crank-Nicolson method. Furthermore, the implicit trapezoidal rule is equivalent to the midpoint rule for the case of linear dynamic problems and is simply denoted as trapezoidal rule in the following. It is obtained by setting

$$\beta_N = \frac{1}{4} \qquad \gamma_N = \frac{1}{2}$$

and results in the most simple variant for the iteration matrix and the generalized force vector

$$\boldsymbol{D}_{\mathrm{tr}}^{(s)} = \boldsymbol{M}^{(s)} + \Delta t^2 \frac{1}{4} \boldsymbol{K}^{(s)} \tag{3.10}$$

$$\boldsymbol{g}_{\mathrm{tr}}^{(s),n+1} = \boldsymbol{f}_{\mathrm{ext}}^{(s),n+1} - \boldsymbol{K}^{(s)} \left[\boldsymbol{u}^{(s),n} + \Delta t \dot{\boldsymbol{u}}^{(s),n} + \Delta t^2 \frac{1}{4} \ddot{\boldsymbol{u}}^{(s),n} \right]. \tag{3.11}$$

The short form for the structural equation for the trapezoidal rule reads

$$\boldsymbol{D}_{\mathrm{tr}}^{(s)} \ddot{\boldsymbol{u}}^{(s),n+1} = \boldsymbol{g}_{\mathrm{tr}}^{(s),n+1} - \boldsymbol{t}^{(s)^T} \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda}^{n+1} .$$

Generalized substructure equations. In the following, the subscripts gα, N and tr are omitted for the simplicity of notation. The general substructure equations used from here on read

$$\boldsymbol{D}^{(s)} \ddot{\boldsymbol{u}}^{(s),n+1} = \boldsymbol{g}^{(s),n+1} - \boldsymbol{t}^{(s)^T} \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda} \tag{3.12}$$

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{t}^{(s)} \ddot{\boldsymbol{u}}^{(s),n+1} = \boldsymbol{0} \tag{3.13}$$

In this notation, it is not specified anymore to which point in time $\boldsymbol{\lambda}$ corresponds. This could be effectively t_{n+1} in case of a Newmark time integration scheme or the trapezoidal rule, i.e., the same point in time at which the accelerations are computed. In the case of generalized-α, it would correspond to $t_{n+1-\alpha_f}$, referring to a point in time between t_n and t_{n+1}.

Limitations of generalized-α methods. Despite the broad success of the family of generalized-α methods in the field of computational structural dynamics, its limitations are also worth mentioning. As the main drawback, the whole family of methods was found to inherently exhibit velocity overshoot as well as first-order accuracy in accelerations. But due to the results in [61], it is impossible for methods within the GSSSS framework and thus within the LMS framework to possess L-stability, i.e., unconditional stability and asymptotic annihilation of the high-frequency response and not to suffer from velocity overshoot at the same time. In other words, either one or the other always occurs. In [62], a method without velocity overshoot is presented, which is accordingly not L-stable. For further information about overshooting, see [76] and for further attempts to develop methods which do not suffer from these drawbacks [77, 78].

3.2.4 Constrained dynamics

While the methods recommended so far are all unconditionally stable for unconstrained systems, i.e., the global, undecomposed system of equation (3.1), it is shown in [60] that for a system with constraints on the displacements, instability is introduced. This instability is caused by infinite double eigenfrequencies, which can be interpreted as modes associated with the Lagrange multipliers due to their lack of mass [35].

This weak instability is transferred to the accelerations, disabling the unconditional stability of Newmark family methods like the trapezoidal rule, independent of the time step value Δt. Adding numerical dissipation in the Newmark framework, restores stability but its second-order accuracy is then lost.

Three ways were presented in [60] to retain unconditional stability for the decomposed, constrained system of substructure equations.

Numerical damping. As presented earlier, the generalized-α method allows for numerical high-frequency dissipation and is proven in [60] to be stable as well as second-order accurate for the decomposed system with constraints on the displacements. This only holds if numerical dissipation is present and thus $\alpha_m = \alpha_f$ is not longer allowed. Consequently, the requirement for unconditional stability in equation (3.5) must be adapted to

$$\alpha_m < \alpha_f \leq \frac{1}{2} \qquad \beta_N \geq \frac{1}{4} + \frac{1}{2}(\alpha_f - \alpha_m).$$

Constraints on accelerations. If the constraints are imposed on the accelerations, zero eigenfrequencies are introduced but can be filtered out if the constraints on accelerations are chosen such that displacement compatibility follows. As it was analyzed in [79], the interface problem derived in this chapter always satisfies this requirement. It is proven in [60] that in this case, every time integration scheme that is unconditionally stable and second-order accurate for the undecomposed, global system, holds these properties for the decomposed, (acceleration-)constrained system as well, even without numerical dissipation.

Bypassing the constraint dynamics. A third method, based on bypassing the constraint dynamics by introducing momentum variables is presented in [60].

In this work, only the second approach, i.e., formulating the constraints on the acceleration level, is considered as it allows for the largest variety of time integration schemes and is the least complicated to implement.

Further investigations on the stability and convergence of the generalized-α scheme were carried out in [63] for nonlinear dynamic problems. The accuracy for constrained mechanical systems with non-constant mass matrix are analyzed in [80] for a variant of the algorithm.

3.3 Construction of the Interface Problem and Iterative Solution

Like it was done for static problems, the interface problem is built by evaluation of the interface compatibility condition of equation (3.13). To do so, the substructure balance equation (3.12) is solved for the accelerations

$$\ddot{\boldsymbol{u}}^{(s),n+1} = \boldsymbol{D}^{(s)^{-1}} \left(\boldsymbol{g}^{(s),n+1} - \boldsymbol{t}^{(s)^T} \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda} \right)$$

and inserted into the compatibility constraints of equation (3.13). This yields the dynamic interface problem

$$\boldsymbol{F}\boldsymbol{\lambda} = \boldsymbol{d}^{n+1} . \tag{3.14}$$

In this case, the dynamic interface operator $\boldsymbol{F}$ and the right-hand side $\boldsymbol{d}$ are defined by

$$\boldsymbol{F} = \sum_{s=1}^{N_s} \boldsymbol{F}^{(s)} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{S}_D^{(s)^{-1}} \boldsymbol{B}^{(s)^T} \qquad \text{and} \qquad \boldsymbol{d}^{n+1} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{t}^{(s)} \boldsymbol{D}^{(s)^{-1}} \boldsymbol{g}^{(s),n+1} .$$

The interface operator is built using the Schur complement $\boldsymbol{S}_D^{(s)}$, which is the stepping matrix $\boldsymbol{D}^{(s)}$ of substructure s condensed on its interface dof. It is defined by

$$\boldsymbol{S}_D^{(s)} = \boldsymbol{D}_{II}^{(s)} - \boldsymbol{D}_{Ir}^{(s)} \boldsymbol{D}_{rr}^{(s)^{-1}} \boldsymbol{D}_{rI}^{(s)} \qquad \text{and} \qquad \boldsymbol{S}_D^{(s)^{-1}} = \boldsymbol{t}^{(s)} \boldsymbol{D}^{(s)^{-1}} \boldsymbol{t}^{(s)^T} .$$

Different from the stiffness matrices in static problems, the stepping matrices $\boldsymbol{D}^{(s)}$ are always symmetric positive definite because of the presence of the mass matrix $\boldsymbol{M}^{(s)}$. From a physical point of view, rigid body modes still exist but possess kinetic energy in the case of dynamics.

Consequently, the matrix $\boldsymbol{G}^{(s)}$ that consists of the zero energy modes, i.e., rigid body modes in statics, is empty. As a result, the self-equilibrium condition and the natural coarse space projection by the projector $\boldsymbol{P}_G$ vanish. However, an auxiliary coarse space projection that results in a FETI-2 algorithm can still be applied.

FETI interface problems of this type were first constructed in [59] and later in [40, 45, 70] with auxiliary coarse spaces. Further improvements on auxiliary coarse space for dynamics were proposed in [81] and an adapted auxiliary coarse space for contact problems was introduced in [82].

3.3.1 Preconditioning and Scaling

Already in the first approach to employ the FETI algorithm for dynamic problems in [59], the lumped and the Dirichlet preconditioner were adapted to the dynamic interface problem by replacing the condensed stiffness by the condensed stepping

matrix. The scaled lumped preconditioner for the dynamic interface problem of equation (3.14) is then defined as

$$\boldsymbol{H}_L = \sum_{s=1}^{N_s} \boldsymbol{H}_L^{(s)} = \sum_{s=1}^{N_s} \widetilde{\boldsymbol{B}}^{(s)} \boldsymbol{t}^{(s)} \boldsymbol{D}^{(s)} \boldsymbol{t}^{(s)^T} \widetilde{\boldsymbol{B}}^{(s)^T} = \sum_{s=1}^{N_s} \widetilde{\boldsymbol{B}}^{(s)} \boldsymbol{D}_{II}^{(s)} \widetilde{\boldsymbol{B}}^{(s)^T}$$

and the scaled Dirichlet preconditioner as

$$\boldsymbol{H}_D = \sum_{s=1}^{N_s} \boldsymbol{H}^s = \sum_{s=1}^{N_s} \widetilde{\boldsymbol{B}}^{(s)} \boldsymbol{S}_D^{(s)} \widetilde{\boldsymbol{B}}^{(s)^T} .$$

Also the scaling coefficients have to be adapted in dynamic problems. The scaled dual assembly matrices are still defined by

$$\widetilde{\boldsymbol{B}}^{(s)} = \boldsymbol{\beta}^{(s)} \boldsymbol{B}^{(s)} .$$

In [36], a mechanical consistent scaling was proposed for static problems. The method scales the gaps and the estimated correction forces based on their stiffness. As proposed in [81], the diagonal scaling matrices $\boldsymbol{\beta}^{(s)}$ for dynamic problems have to be computed using the coefficients of the stepping matrix and not only the stiffness coefficients anymore. While nothing changes for multiplicity scaling, the diagonal elements $\beta_1^{(s)}, \beta_2^{(s)} \dots \beta_{N_\lambda}^{(s)}$ of the superlumped scaling matrices $\boldsymbol{\beta}^{(s)}$ for the preconditioners of the dynamic interface problem are defined by

$$\beta_i^{(s)} = 0 \qquad \forall \quad i \in \left\{ i \in \{1, \dots, N_\lambda\} \mid \lambda_i \notin \Lambda^{(s)} \right\}$$

and

$$\beta_i^{(s)} = D_{n_r,n_r}^{(r)} \left(\sum_{j \in S^n} D_{n_j,n_j}^{(j)} \right)^{-1} \qquad \forall \quad i \in \left\{ i \in \{1, \dots, N_\lambda\} \mid \lambda_i \in \Lambda^{(s)} \right\} .$$

Similar to the static problem, λ_i is the Lagrange multiplier that connects the local dof n_s in substructure s and the local dof n_r in substructure r. The scalar $D_{n_r,n_r}^{(r)}$ refers to the entry of the matrix $\boldsymbol{D}^{(r)}$ in the n_r-th row and the n_r-th column, i.e., to its n_r-th diagonal entry. Furthermore, $\Lambda^{(s)}$ is the set of Lagrange multipliers that act on substructure s and $S^n = \{s, r, \dots\}$ is the set of substructures that share the corresponding global dof n of the undecomposed structure, which then becomes after decomposition the local dof n_r in substructure r, and so on.

3.3.2 Initial Acceleration

To start the time stepping scheme, an initial iteration must be available. Its computation is slightly different from the subsequent time steps, but the FETI solver can still be applied. For a given initial displacement $\boldsymbol{u}^{(s),0}$, the local balance equation for substructure s reads

$$\boldsymbol{M}^{(s)} \ddot{\boldsymbol{u}}^{(s),0} = \boldsymbol{f}_{\text{ext}}^{(s),0} - \boldsymbol{K}^{(s)} \boldsymbol{u}^{(s),0} - \boldsymbol{t}^{(s)^T} \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda}^0$$

and can be solved for the initial accelerations, yielding

$$\ddot{\boldsymbol{u}}^{(s),0} = \boldsymbol{M}^{(s)^{-1}} \left(\boldsymbol{f}_{\text{ext}}^{(s),0} - \boldsymbol{K}^{(s)} \boldsymbol{u}^{(s),0} - \boldsymbol{t}^{(s)^T} \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda}^0 \right). \tag{3.15}$$

These accelerations must fulfill the interface compatibility constraint, which reads after injection of the initial acceleration

$$\sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{t}^{(s)} \boldsymbol{M}^{(s)^{-1}} \boldsymbol{t}^{(s)^T} \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda}^0 = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{t}^{(s)} \boldsymbol{M}^{(s)^{-1}} \left(\boldsymbol{f}_{\text{ext}}^{(s),0} - \boldsymbol{K}^{(s)} \boldsymbol{u}^{(s),0} \right).$$

This is called initial interface problem and reads in short

$$\boldsymbol{F}^0 \boldsymbol{\lambda}^0 = \boldsymbol{d}^0$$

with

$$\boldsymbol{F}^0 = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{t}^{(s)} \boldsymbol{M}^{(s)^{-1}} \boldsymbol{t}^{(s)^T} \boldsymbol{B}^{(s)^T}$$

$$\boldsymbol{d}^0 = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{t}^{(s)} \boldsymbol{M}^{(s)^{-1}} \left(\boldsymbol{f}_{\text{ext}}^{(s),0} - \boldsymbol{K}^{(s)} \boldsymbol{u}^{(s),0} \right).$$

If a diagonal, i.e., lumped, mass matrix is used, the system is trivial to solve. In this work, a consistent and thus non-diagonal mass matrix is always used. Consequently, the interface problem must be solved iteratively by employing the standard FETI procedure. After the solution for $\boldsymbol{\lambda}^0$ is found, each substructure can compute its initial accelerations from equation (3.15).

There exist some alternative procedures to avoid the solution of this system. One possibility is to simply set the accelerations to zero, also called truncation. This was analyzed for mechanical systems in [83] and found to lead to a loss of accuracy. More precisely, the presented algorithms of second-order degrade to first-order accuracy. According to [35], zero initial acceleration introduces parasitic force and energy, which is reduced for smaller time steps but remains nonetheless when neither numerical nor physical damping is present. Further alternatives are presented in [35]. One approximates the mass matrix by its diagonal and another neglects interface coupling, i.e., sets the interface forces to zero. However, this introduces an error in the interface compatibility condition and thus leads to permanent drift over time if the constraints are set on the accelerations.

3.3.3 Spectral Properties

Recalling the definition of the stepping matrix for generalized-α in equation (3.6) or Newmark-β in equation (3.8), which reads

$$\boldsymbol{D}_{\text{N}}^{(s)} = \boldsymbol{M}^{(s)} + \Delta t^2 \beta_N \boldsymbol{K}^{(s)},$$

one observes a strong dependency of its composition on the time step size Δt. The following implications of small and large time steps, in particular a strong dependency of the conditioning on the size of Δt, are described, e.g., in [59, 82].

For small time steps, the influence of the stiffness matrix becomes small, and the stepping matrix inherits the properties of the mass matrix with only small perturbations from the stiffness. In this case, $\boldsymbol{D}^{(s)}$ is also said to be dominated by the mass matrix, which is usually diagonally dominant [35]. The mass matrix is very well conditioned and so is the resulting stepping matrix.

If instead, the time step size is large, the stepping matrix is dominated by the stiffness matrix with only small perturbations from the mass matrix. In this case, $\boldsymbol{D}^{(s)}$ is likely to be much more ill-conditioned. Furthermore, if there are floating subdomains, eigenvalues close to zero occur, originating from the kernel of the stiffness matrix that represents the physical rigid body modes.

The eigenvalue distribution of the dynamic interface operator was already discussed in the first application of FETI to dynamic problems in [59]. For a refined mesh, the authors observed for the dynamic case the same effect that results in superconvergent behavior in the static case. Small eigenvalues of $\boldsymbol{F}$ accumulate to zero or in case of the preconditioned operator to one, while large eigenvalues that correspond to smooth physical modes stabilize. Consequently, a superconvergent behavior according to [57] is observed for the dynamic case.

Regarding the numerical scalability of the method, a weak dependence of the condition number on the mesh size h was observed in [59] for the Dirichlet preconditioner if crosspoints, i.e., interface nodes with a multiplicity greater than two, are present. In absence of crosspoints, the condition number was observed to be independent from the mesh size h.

3.3.4 Iterative Solution

The time stepping algorithm with the inner loop of FETI iterations is listed in algorithm 3. It computes the displacements of a transient problem for the discrete points in time t_n with $n = 1, \ldots, n_{\text{end}}$. Because the discussed time stepping schemes only vary in the definition of $\boldsymbol{D}$ and $\boldsymbol{g}$, the algorithm is valid for all of them. The computation of the initial acceleration is not listed explicitly, as only $\boldsymbol{F}$ and $\boldsymbol{d}^{n+1}$ must be replaced by $\boldsymbol{F}^0$ and $\boldsymbol{d}^0$, respectively.

While the interface operator $\boldsymbol{F}$ remains constant, the right-hand side $\boldsymbol{d}$ changes for every time step. After convergence of the FETI algorithm, the accelerations are recovered and the Newmark integration rules are applied to compute the corresponding velocities and displacements.

Algorithm 3: FETI-2 for dynamic problems

Given: $\boldsymbol{u}^{(s),0}$, $\dot{\boldsymbol{u}}^{(s),0}$, $\ddot{\boldsymbol{u}}^{(s),0}$

for $n = 0 \dots (n_{end} - 1)$ **do**

$\boldsymbol{d}^{n+1} = \sum_{s=1}^{N_s} \boldsymbol{B}^{(s)} \boldsymbol{t}^{(s)} \boldsymbol{D}^{(s)^{-1}} \boldsymbol{g}^{(s),n+1}$

$\hat{\boldsymbol{\lambda}}_0 = \boldsymbol{0} \qquad \boldsymbol{\lambda}_C = \boldsymbol{C}(\boldsymbol{C}^T \boldsymbol{F} \boldsymbol{C})^{-1} \boldsymbol{C}^T \boldsymbol{d}^{n+1}$

$\boldsymbol{r}_0 = \boldsymbol{d}^{n+1} - \boldsymbol{F} \boldsymbol{\lambda}_C$

$\boldsymbol{z}_0 = \boldsymbol{H} \boldsymbol{r}_0$

$i = 0$

while $r_{*,i} > \epsilon_{r,*}$ **do**

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

$\phi_{i,j} = \boldsymbol{q}_j^T \boldsymbol{z}_i \qquad \beta_{i,j} = \frac{\phi_{i,j}}{\Delta_j}$

$\boldsymbol{w}_i = \boldsymbol{P}_C \boldsymbol{z}_i - \sum_{j=0}^{i-1} \boldsymbol{w}_j \beta_{i,j}$

$\boldsymbol{q}_i = \boldsymbol{F} \boldsymbol{w}_i$

$\Delta_i = \boldsymbol{q}_i^T \boldsymbol{w}_i \qquad \gamma_i = \boldsymbol{w}_i^T \boldsymbol{r}_i \qquad \alpha_i = \frac{\gamma_i}{\Delta_i}$

$\hat{\boldsymbol{\lambda}}_{i+1} = \hat{\boldsymbol{\lambda}}_i + \boldsymbol{w}_i \alpha_i$

$\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \boldsymbol{q}_i \alpha_i$

$\boldsymbol{z}_{i+1} = \boldsymbol{H} \boldsymbol{r}_{i+1}$

$i \leftarrow i + 1$

$m_{n+1} = i$

$\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_C + \hat{\boldsymbol{\lambda}}_i$

$\ddot{\boldsymbol{u}}^{(s),n+1} = \boldsymbol{D}^{(s)^{-1}} \left(\boldsymbol{g}^{(s),n+1} - \boldsymbol{t}^{(s)^T} \boldsymbol{B}^{(s)^T} \boldsymbol{\lambda}_i \right)$

$\dot{\boldsymbol{u}}^{(s),n+1} = \dot{\boldsymbol{u}}^{(s),n} + \Delta t \left((1 - \gamma_N) \ddot{\boldsymbol{u}}^{(s),n} + \gamma_N \ddot{\boldsymbol{u}}^{(s),n+1} \right)$

$\boldsymbol{u}^{(s),n+1} = \boldsymbol{u}^{(s),n} + \Delta t \dot{\boldsymbol{u}}^{(s),n} + \Delta t^2 \left(\left(\frac{1}{2} - \beta_N \right) \ddot{\boldsymbol{u}}^{(s),n} + \beta_N \ddot{\boldsymbol{u}}^{(s),n+1} \right)$

3.4 Fundamental Concepts for Improving Convergence

The main question addressed in this work is how the iterative procedure can be accelerated by enhancing the algorithm. In particular, the specific requirements of problems of structural dynamics are investigated. This section presents an overview of the most important fundamental concepts to improve convergence. Figure 3.2 summarizes them and groups them into different classes. Furthermore, this section differentiates between concepts that are not considered in this thesis, concepts that are employed as state of the art, and concepts that are extended by newly proposed methods. Where possible, the corresponding chapters and sections are referred.

An important requirement to obtain fast convergence of an efficient iterative algo-

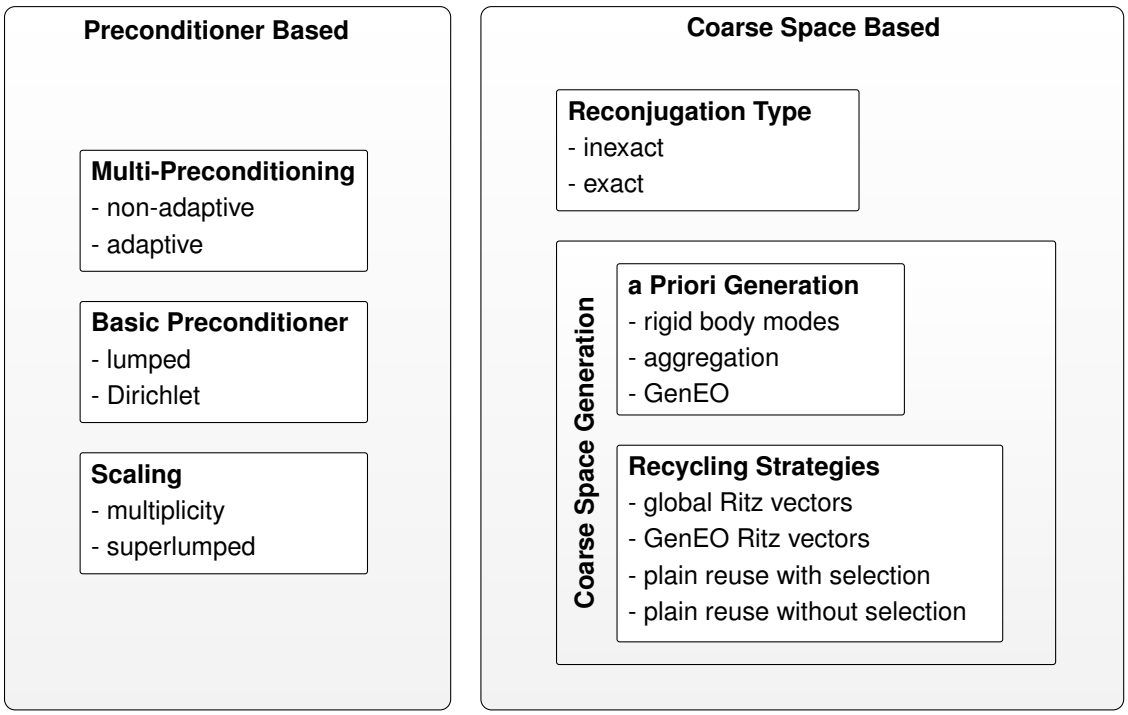


Figure 3.2. Basic techniques to accelerate the iterative procedure and non-exhaustive lists of examples.

rithm that is based on the method of conjugate gradients is to ensure a beneficial spectrum of the deflated, preconditioned operator. The according relations between the interface operator spectrum and the convergence were described in section 2.5. This leads to a fast rate of convergence and an accurate solution is found in only few iterations. However, it must always be considered how much additional computational cost and how much additional communication is introduced in each iteration. Both should be kept as low as possible.

First of all, methods based on preconditioning and methods based on auxiliary coarse spaces are differentiated. Although auxiliary coarse space projections can formally be integrated completely in the preconditioner, they work in fundamentally different ways as it was remarked in [49]. An auxiliary coarse space projection makes the effective operator singular and removes parts of the spectrum. Accordingly, an additional coarse problem must be solved in each iteration to obtain and maintain the solution within the coarse space, which is not considered anymore by the iterative process. In contrast, a classical preconditioner does not make the effective operator singular. Instead it approximates the inverse of the original operator. The aim is to change the spectrum in a beneficial way without removing parts, i.e., without shifting eigenvalues to zero. In FETI, the requirement for a viable preconditioner is of course that it is constructed from local contributions with only local communication, which allows for an efficient parallel application. The deflation based methods are further differentiated between those who construct the coarse space independently of a former solution process, called *a priori* spaces, and those who construct the coarse space from the solution space of a former time step, called *recycled* spaces.

3.4.1 Preconditioner Based Methods

Concerning the basic preconditioners, i.e., a way to approximate the inverse of $\boldsymbol{F}$ by local contributions, only the lumped and the Dirichlet preconditioner that were introduced in section 2.3.1 are known in the literature. While the Dirichlet preconditioner is optimal in terms of numerical scalability, the lumped preconditioner can be even more efficient, but this depends on the problem to be solved [34, 70, 84].

Independently from the chosen basic preconditioner, a scaling procedure can be employed by using scaled dual assembly matrices. In this work, multiplicity and superlumped scaling as introduced in section 2.3.2 are considered. In the presence of crosspoints, at least multiplicity scaling is mandatory. Because scaling adds only negligible computational cost to the algorithm, it should always be used. Superlumped scaling can improve the rate of convergence significantly if there are jumps in the coefficients across the substructure interfaces.

A more recent advancement in preconditioning of non-overlapping dual domain decomposition methods is multipreconditioning. It is discussed in detail in chapter 5 and exploits in principal the structure of the preconditioner as sum of local contributions. Each local contribution is viewed as individual preconditioner such that instead of one, multiple search directions become available in each iteration.

This work only makes use of the state of the art concerning basic preconditioners and scaling techniques. In any case, the superlumped scaling is used. The standard preconditioning is sometimes called single-preconditioning to differentiate it from multipreconditioning. In chapter 4, single-preconditioned methods are applied to dynamic problems with heterogeneous and homogeneous material distributions. In particular, the behavior of a priori coarse spaces and recycling strategies in different scales of time is considered. Chapter 5 introduces multipreconditioning, highlights some special issues when used for the solution of dynamic problems, and proposes and investigates new recycling strategies applied to problems with heterogeneous and homogeneous material distributions.

3.4.2 Deflation by a Priori Spaces

An auxiliary coarse space as introduced in section 2.4 has been employed to accelerate the solution of problems in structural dynamics for the first time in [45] where the rigid body mode coarse space was used. It was refined in [81] and further modified for dynamic contact problems in [82]. The GenEO coarse space that was introduced in section 2.6 has been first been applied to problems of structural dynamics in the author's publication [3].

Rigid Body Modes

Already in [59], the standard FETI algorithm was found to lack the important numerical H scalability. That is to say, when the number of subdomains was increased by decreasing the subdomain size, the condition number was observed to increase significantly. This was explained in [59] by the fact that the subdomain bandwidth does not decrease as fast as the subdomain size, and it was proposed to allocate more than one processor to a given subdomain, i.e., to use a kind of hybrid parallelization to cope with this problem. A better solution was provided in [45]. It was suggested to employ the rigid body modes of floating substructures as auxiliary coarse space in order to establish more global communication during the iterations. Several experiments were conducted in [45] that indicated that the rigid body modes are viable to restore numerical H scalability. However, these experiments did neither consider heterogeneities, nor different time scales. The applicability to dynamic problems and the performance of the rigid body mode coarse space, in particular as a function of heterogeneity and time scale, is investigated in chapter 4.

The GenEO Coarse Space

The GenEO coarse space has been introduced for static problems in section 2.6. It was shown to construct very efficient coarse spaces for diverse static problems, e.g., in [58, 85], but was never studied when applied to dynamic problems. Its application to linear dynamic problems, its relation to rigid body modes, and in particular its behavior as a function of the time scale is investigated in detail in chapter 4. Moreover, it is compared to recycling strategies.

Aggregation

A further well known method to construct a priori coarse spaces is aggregation. It means to enforce exact compatibility of certain individual interface nodes. Aggregation was the coarse space for which FETI-2 was initially invented in [46] and [47]. It was designed especially for fourth-order shell problems, which required to glue the corner dof permanently together throughout the iterative process. Later, this technique was replaced by FETI-DP, which was published in [23]. It ensured the compatibility on the corner dof not by an auxiliary coarse space like introduced in chapter 2 but by splitting the dof into dual and primal ones. However, this also leads to the solution of a coarse problem in each iteration. The primal dof are never connected by Lagrange multipliers and respect the compatibility by design, just like it is done in the primal Schur complement method. Neither aggregation as auxiliary coarse space nor the improved variant of FETI-DP are considered in this work. FETI-DP is able to exploit its strengths if either fourth-order problems should be solved or if zero energy modes would require a natural rigid body mode coarse space otherwise. In this case, FETI-DP was found to be superior compared to FETI-2. However, arbitrary and abstract coarse spaces are necessary in dynamic problems. Furthermore, only second-order problems are considered in this work. As a consequence, a limitation to FETI-2 methods is reasonable.

3.4.3 Deflation by Recycled Spaces

A distinctive feature of linear structural dynamics is that the matrix that describes the system of equations to be solved remains constant. Only the right-hand side changes from one time step to the next. This suggests reusing the information gathered during a solution process, similar to what direct solvers can do by storing the factorization of the matrix. Of course, this is not possible for FETI solvers because they work iteratively and the global problem is never assembled but instead remains distributed over the individual substructures. The general case of solving a problem on the same structure for multiple right-hand sides has been addressed in [86] and [87] by exploiting the potential of a standard CG algorithm to efficiently reuse previously built Krylov spaces. This recycling strategy of reusing the plain, unprocessed solution spaces was also applied in the case of changing operators when solving nonlinear problems in [88] and [89]. These techniques were refined in the context of domain decomposition in several publications, mostly [90], [91], and [92], by approximating eigenvectors of the preconditioned operator inside the generated Krylov space, based on a Ritz ansatz similar to the Lanczos procedure. Unfortunately, the Ritz approximation based recycling strategies can not be transferred to multipreconditioned FETI. Moreover, recycling has not been applied to multipreconditioned FETI at all before the publications [3, 4] of the author and this work.

Recycling for Single-Preconditioning

These recycling strategies combined with single-preconditioned FETI are applied to problems of linear dynamics in chapter 4. In particular, their capabilities to construct specific and efficient coarse spaces for heterogeneous dynamic problems at different time scales are assessed. They are further compared to the a priori coarse spaces.

Recycling for Multipreconditioning

In chapter 5, the most simple strategy of plain reuse is transferred to multipreconditioning. The chapter proposes furthermore two completely new and comprehensive recycling strategies that are based on a Ritz approximation of the GenEO eigenmodes. The strategies are assessed and compared for homogeneous and heterogeneous problems as well as for different mesh decompositions.

Reconjugation Types

Figure 3.2 lists two types of reconjugation. This is to express that mainly two different ways that are not mathematically equivalent exist to use or reuse a solution subspace for deflation. In linear problems, deflation by a projector and reconjugation as part of the orthogonalization step in the CG algorithm are equivalent. This was noted earlier in [40] and demonstrated in section 2.4. Because this results in exact $\boldsymbol{F}$-conjugacy between the newly created search directions and the reused solution

space, called coarse space, augmentation space or the like, it is distinguished here as a reconjugation of the type *exact*. Another possibility arises if the operator does not remain constant, for example, in nonlinear problems. To attain exact reconjugation in this case, either the basis vectors of the reused solution space must be reorthogonalized with respect to the new interface operator $\boldsymbol{F}^n$ of the next step, or a full non-diagonal projector must be used. A different strategy with the aim to save computational cost has been proposed in some very similar ways in [88, 89, 93]. They all have in common that they use the reused directions as they are, i.e., conjugate with respect to the old interface operator $\boldsymbol{F}^{n-1}$. Furthermore, they also reuse the already computed applications $\boldsymbol{F}^{n-1}\boldsymbol{w}_i^{n-1}$ of the old interface operator to the reused search directions. The remaining process works like a reorthogonalization. However, it is an *inexact* reorthogonalization because of the simplifications stated above. That is why it may only be applied as a preconditioner and it is not possible to predict the quality of the procedure.

Chapter 4

Deflation Strategies on Different Scales of Time

The aim of this chapter is to analyze the properties of the FETI interface problem on different scales of time and to assess deflation strategies, i.e., methods to construct coarse spaces, with regard to their ability to increase the efficiency of the iterative solution algorithm. To this end, mainly spectral properties are studied, as they essentially determine convergence rate and robustness.

The first section introduces the major effects that occur when the time step size is varied across a large range. In particular, it studies the role of rigid body modes and furthermore the influence of the Dirichlet and the lumped preconditioner on the discussed phenomena. Rigid body modes still exist in problems of structural dynamics but, different from the static case, possess kinetic energy and therefore play a different role. A final discussion highlights the impact of these influencing factors on the conditioning of the interface problem and how this affects the convergence behavior of the conjugate gradient algorithm.

The use of a priori coarse spaces, in particular of those based on rigid body modes, is be discussed in section two. Mode shape visualizations and principal angles between

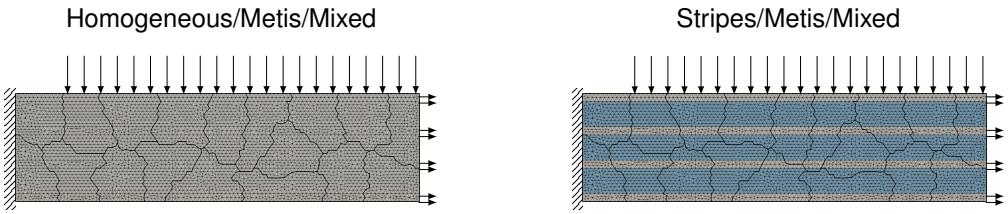


Figure 4.1. Structural problem setup. The homogeneous structure is shown on the left, the heterogeneous structure on the right. The heterogeneous structure is composed of a soft matrix (blue) between small fibers (gray), which consist of the same stiff material as the homogeneous structure. The nodes on the left egde are fixed by Dirichlet conditions. The nodes on the upper edge are subjected to a bending load and those parts of the right edge that belong to stiff fibers in the heterogeneous structure are subjected to a traction load. Because of the two different types of load, these variants are also identified by the keyword *Mixed*.

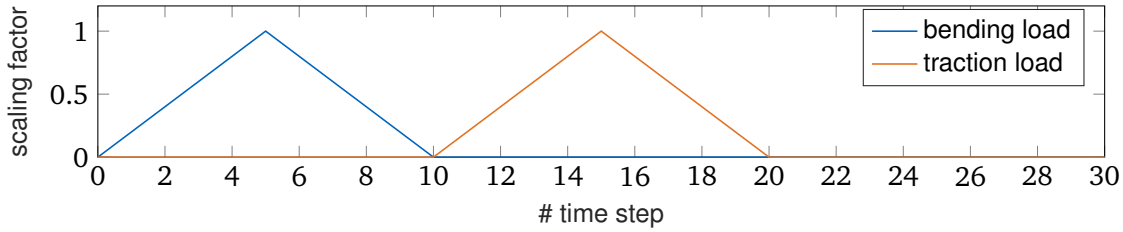


Figure 4.2. Scaling factors of the bending and the traction load over time. First, the bending load linearly increases from zero to its maximum and then decreases back to zero again. Subsequently, the traction load is scaled alike.

subspaces illustrate their relation to GenEO modes, which are used as a kind of reference. To find a recommendation which technique should be applied according to the problem setting, their spectral efficiency and their ability to reduce computational cost are investigated.

The third section mainly addresses recycling methods, which build coarse spaces from former solution spaces, and studies their fitness to increase the solver's efficiency for homogeneous problems as well as for heterogeneous ones and independently from the time step size. The main aspect considered is the ability to construct coarse spaces that are appropriately adapted to the problem in terms of the spanned subspace and their size. For the selective reuse of Krylov subspaces (SRKS) algorithm, which is based on Ritz approximations of global eigenvectors, the optimal tuning of its parameters and an alternative selection criterion are investigated. While the GenEO method is used in section two more as a reference for the rigid body mode based coarse spaces, it is included here to the full extent of its capabilities.

Throughout the whole chapter, homogeneous and heterogeneous problems are considered to illustrate the effects of material heterogeneity. The two example structures used are shown in figure 4.1. They correspond to the example structures 1 and 4 in appendix A. Both are two-dimensional, linear finite element problems and use the same geometry, mesh, and decomposition. The mesh was generated by Gmsh [94] and decomposed into 18 substructures by the built-in METIS [95] implementation. In the heterogeneous structure on the right of figure 4.1, the four gray colored thin fibers have the same stiff material properties as the identically colored homogeneous structure on the left. The space between the fibers is filled by a much softer matrix, which is colored blue. In table A.1 in appendix A, the stiffness ratio is given.

The nodes on the left edges are fixed by Dirichlet boundary conditions, and both structures are subjected to two different kinds of load. First, a bending load on the top edge linearly increases until time step number 5 and then linearly decreases to zero again. The same linear scaling is subsequently applied to the traction load, which acts on the nodes on the right edge. However, in both examples, the traction load is only applied to nodes which belong to stiff fibers in the heterogeneous example. The scaling of both loads over time is illustrated in figure 4.2.

Table A.1 in appendix A summarizes the most important physical properties. This also includes the periods of the first three eigenfrequencies, which are the determin-

ing measures when choosing large time step sizes that are appropriate to simulate slow, global oscillations. Furthermore, the period of the highest eigenfrequency is included, which is highly dependent on the mesh size and gives an upper bound for the time step size when analyzing wave propagation phenomena.

4.1 Spectral Properties of the Interface Problem

This section illustrates the change of the interface problem's spectral properties in case of material heterogeneities and for different scales of time. Special attention is given to the composition of the spectrum, to the role of rigid body modes, and to the convergence behavior of the conjugate gradient algorithm for time steps of practically relevant size.

4.1.1 Relation to the Equivalent Static Problem

There exists an intuitive relation between the static and the dynamic interface problem. It is illustrated in the following for the case of the trapezoidal rule. When the time step size increases to infinity, the dynamic interface operator, scaled by $\Delta t^2/4$, becomes

$$\begin{aligned}\lim_{\Delta t\to\infty}\frac{\Delta t^2}{4}\boldsymbol{F}_{\text{dynamic}} &= \lim_{\Delta t\to\infty}\frac{\Delta t^2}{4}\sum_{s=1}^{N_s}\boldsymbol{B}^{(s)}\boldsymbol{t}^{(s)}\boldsymbol{D}^{(s)^{-1}}\boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T}\\ &= \lim_{\Delta t\to\infty}\frac{\Delta t^2}{4}\sum_{s=1}^{N_s}\boldsymbol{B}^{(s)}\boldsymbol{t}^{(s)}\left(\boldsymbol{M}^{(s)}+\frac{\Delta t^2}{4}\boldsymbol{K}^{(s)}\right)^{-1}\boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T} \qquad (4.1)\\ &= \lim_{\Delta t\to\infty}\frac{\Delta t^2}{4}\frac{4}{\Delta t^2}\sum_{s=1}^{N_s}\boldsymbol{B}^{(s)}\boldsymbol{t}^{(s)}\boldsymbol{K}^{(s)^{-1}}\boldsymbol{t}^{(s)^T}\boldsymbol{B}^{(s)^T}=\boldsymbol{F}_{\text{static}}\,.\end{aligned}$$

At least, this relation holds in the case where the inverse of each stiffness matrix $\boldsymbol{K}^{(s)}$ exists. Note that in equation (4.1), the interface operators for a dynamic and a static problem are exceptionally distinguished by a subscript. The subscript $\star_{\text{dynamic}}$ is omitted in the remaining part of this chapter.

If floating substructures exist and thus pseudo inverses of the affected stiffness matrices have to be used, an important effect becomes visible in the spectrum of the interface operator $\boldsymbol{F}$. When Δt becomes infinitely large, the scaled iteration matrix

$$\frac{4}{\Delta t^2}\boldsymbol{D}^{(s)}=\frac{4}{\Delta t^2}\boldsymbol{M}^{(s)}+\boldsymbol{K}^{(s)} \qquad (4.2)$$

of a floating substructure returns infinitely small forces for rigid body accelerations. In other words, for a floating substructure s in a two-dimensional problem, three eigenvalues of $\boldsymbol{D}^{(s)}$ tend to zero. Conversely, $(\Delta t^2/4)\boldsymbol{D}^{(s)^{-1}}$ exhibits three eigenvalues

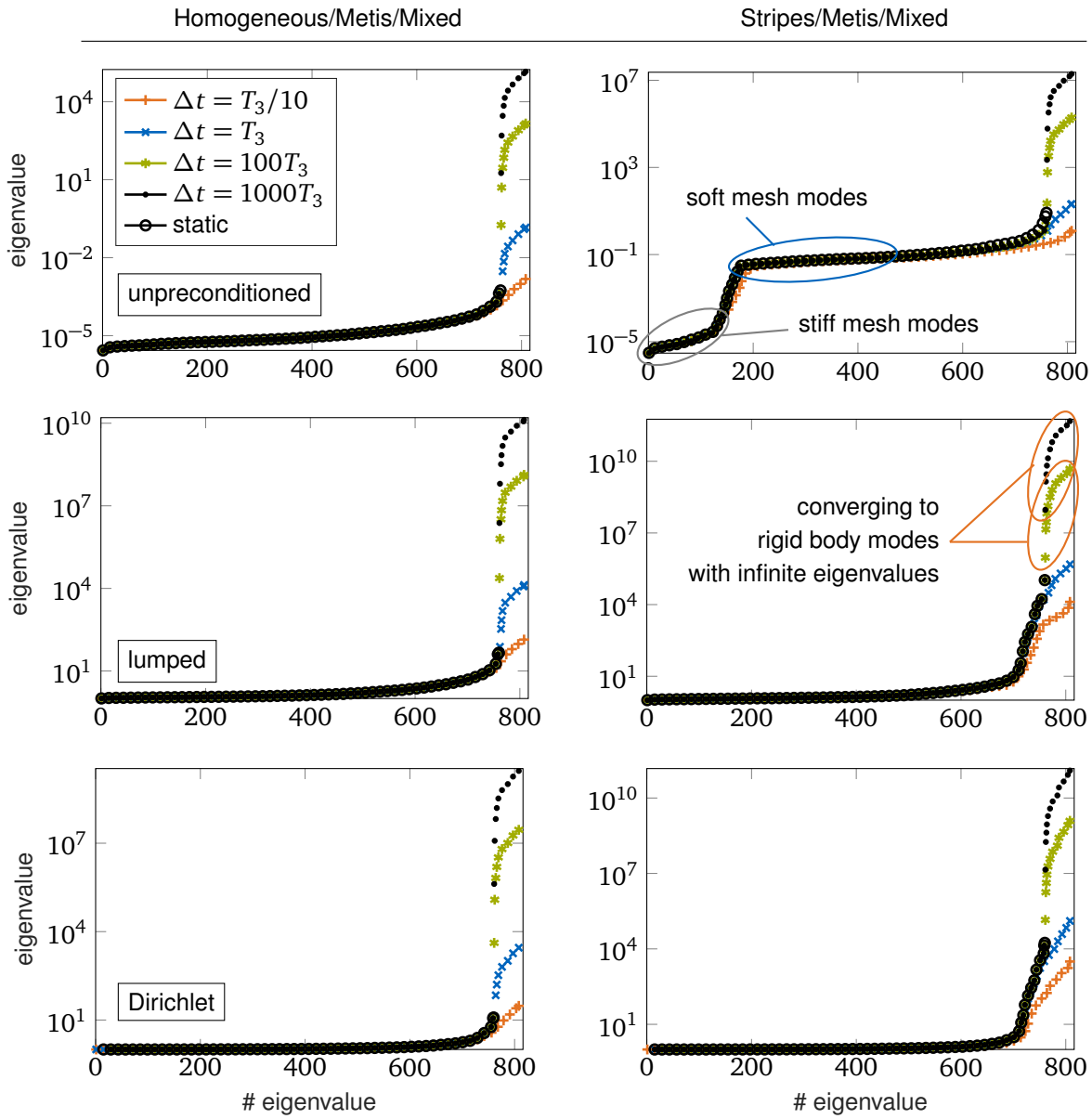


Figure 4.3. Spectra of the unpreconditioned and preconditioned interface operators for both examples at four different, very large time steps. All spectra of dynamic, unpreconditioned interface operators are scaled by the factor $\Delta t^2/4$ to show how they converge to the spectrum of the static interface operator for very large time steps. The spectra also show the diverging eigenvalues, which emerge from rigid body modes. They also occur in the preconditioned spectra. Furthermore, mesh modes of stiff and soft parts are clearly distinguishable in the heterogeneous example.

diverging to infinity. At the same time, the other eigenvalues of $(\Delta t^2/4)\boldsymbol{D}^{(s)^{-1}}$ stabilize and converge towards the eigenvalues of $\boldsymbol{K}^{(s)^+}$. For a total number of N_f floating substructures, these three eigenvalues that diverge to infinity cause $3N_f$ eigenvalues in the interface operator $\boldsymbol{F}$, which also diverge to infinity.

In figure 4.3, the eigenspectra for several interface operators of both examples at different time step sizes are shown. For a clearer visualization, the number of data

points has been reduced by deleting extremely close data points, which would not have been distinguishable at the printed resolution. The left column shows the homogeneous problem, whereas the right column shows the heterogeneous one. In each plot, the spectra at four different time steps are shown. The second smallest time step $\Delta t = T_3$ can be seen as the largest reasonable time step for any practical investigation because it is close to $T_1/10$ and therefore only suitable to resolve the lowest structural mode. The larger time steps $100T_3$ and $1000T_3$ only illustrate the effects of very large time steps by exaggeration.

Unpreconditioned operators. The first row of plots in figure 4.3 shows the eigenvalue distributions of the unpreconditioned interface operators $\boldsymbol{F}$. To illustrate how the dynamic interface problem converges to its static equivalent, the eigenvalues of the dynamic, *unpreconditioned* interface problem are scaled by the factor $\frac{\Delta t^2}{4}$, according to the discussion above. As predicted, the largest 48 eigenvalues, which is the number of rigid body modes of the 16 floating substructures in both examples, diverge to infinity as the time step becomes very large. At the same time, the remaining spectrum aligns with that of the static interface problem. This is most clearly visible for the heterogeneous case. The static spectrum has 48 eigenvalues less than the dynamic spectra, despite the fact that both interface problems are of the same size. This is due to the necessary natural coarse space projection. In fact, the static spectrum shown is the spectrum of $\boldsymbol{P}_G{}^T \boldsymbol{F} \boldsymbol{P}_G$. This projection by $\boldsymbol{P}_G$ generates a zero eigenvalue for each rigid body mode of each substructure. Zero eigenvalues are generally omitted in plots of interface operator's spectra and thus they are shifted to the left and look like possessing fewer eigenvalues.

Mesh Modes. The heterogeneous case allows a clear distinction between stiff and soft mesh modes, i.e., modes that represent a very localized deformation of only a few elements. Due to the narrow fibers and the relatively large matrix area in between, the heterogeneous example exhibits considerably less stiff mesh modes with very small eigenvalues than the homogeneous example. Instead, the spectrum of the heterogeneous example shows a large number of softer mesh modes. In this context, it must be noted that the interface operator is a flexibility operator and therefore the highest frequencies of substructure operators $\boldsymbol{M}^{(s)}$ and $\boldsymbol{K}^{(s)}$ transform into the smallest frequencies of $\boldsymbol{F}$.

Preconditioned operators. In the second and third row of plots in figure 4.3, the spectra of the preconditioned interface operators $\boldsymbol{HF}$ are shown, for the Dirichlet and for the lumped preconditioner. Because the preconditioners already include the scaling by the factor $\frac{\Delta t^2}{4}$ by design, these spectra are not scaled artificially. Both preconditioners fail to capture the rising eigenvalues that emerge from rigid body modes of floating substructures. The mesh modes are handled very well by shifting them closely to 1, which is characteristic of the preconditioners used in dual domain decomposition. Consequently, the corresponding solution subspaces require only modest effort in the iterative solution algorithm to be found. Those higher eigenvalues in the spectra which do not result from rigid body modes correspond to lower, structural

modes. They are not captured very well by the preconditioners and pose the most severe problems to iterative solution algorithm as they describe more global and thus coupled effects among the substructures.

4.1.2 Conditioning and Convergence on Relevant Time Scales

As explained before in the context of time stepping schemes, a tenth of the period of a structural mode is a good rule of thumb for the maximal time step size that still allows to resolve the mode's oscillations at a reasonable accuracy. Based on that, the following four time step sizes have been chosen for all investigations in the remaining chapter, in order to cover the complete range of relevant time scales for implicit time stepping schemes:

$\Delta t = T_3$. It is close to $\Delta t = T_1/10$, which would allow to investigate at least the lowest structural mode and is thus considered as an upper limit for reasonable time step sizes.

$\Delta t = T_3/10$. This time step size allows an investigation of the lowest three structural modes. It is large but highly relevant in practice.

$\Delta t = T_3/100$. It is an intermediate step size between large and small time step sizes and allows to observe a fair number of structural modes.

$\Delta t = T_3/1000$. For both examples, this time step size is close to the one prescribed by Courant's condition for explicit time stepping schemes. It is thus considered as lower limit for reasonable implicit time step sizes.

As an introductory example, the courses of the residuals during the solution of time step number 22 are shown in figure 4.4. They illustrate the behavior of the yet undeflated iterative solution algorithm for both examples on different time scales, and for both preconditioners.

For very large time steps, the graphs show a distinct superlinear convergence, also called superconvergence, and a saw-toothed course of the residual. In particular in the case of the homogeneous example and the time step size $\Delta t = T_3$, a very strong plateau for about 48 iterations can be observed. This is also the number of rigid body modes, which lead in the case of $\Delta t = T_3$ to very large eigenvalues of the interface operator, as shown later in figure 4.5. Once these very low-energy modes have been captured, convergence is extremely fast for the remaining problem. They can be captured well by a rigid body mode coarse space, as it is shown later in figure 4.12.

For very small time steps, convergence is very fast from the beginning. The time steps in between include a transition zone in which the convergence does not exhibit the strong plateau and the saw-tooth shape of the largest time step anymore but is still slow and more bumpy than for the smallest time step. For the homogeneous problem, this transition zone can be identified already round $T_3/10$ and for the heterogeneous problem at smaller time steps around $T_3/100$.

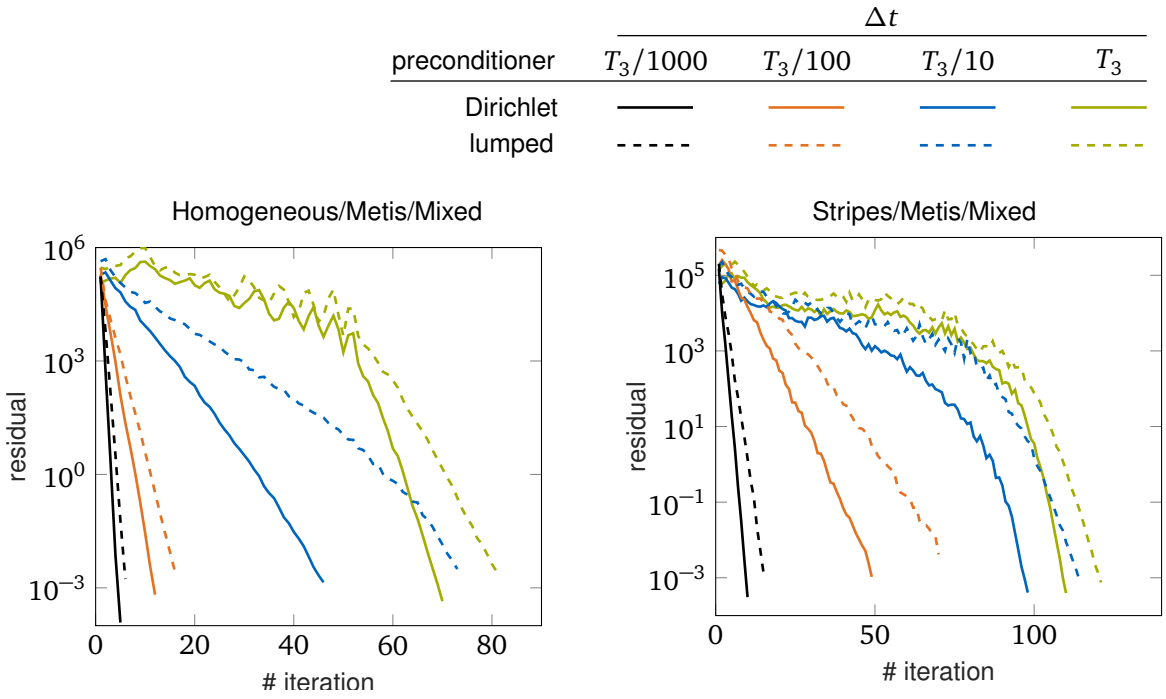


Figure 4.4. Progress of the residual $\sqrt{r^T z}$ during the solution of the 22nd time step for the homogeneous and the heterogeneous example. The lumped and the Dirichlet preconditioner as well as all four time step sizes are considered. Smaller time step sizes result in a significant acceleration of the convergence rate.

While the Dirichlet preconditioner leads to clearly better convergence, the lumped preconditioner may be much cheaper in computational cost, but performance comparisons between the basic preconditioners are not considered further in this work. Instead, it is clearly observable that the qualitative behavior of the solution algorithms is the same for both of them. This includes plateaus or saw-tooth shapes in the residual courses as well as slow convergence in general. Therefore, all following investigations are restricted to the Dirichlet preconditioner. Figure 4.5 shows the corresponding eigenspectra and demonstrates how well they allow to predict convergence behavior almost independently from the actual right-hand side, i.e., the applied load. The upper two plots in figure 4.5 show the full spectra, and it is clearly observable that the lower parts of the spectra are almost identical. Furthermore, these lower parts are not important for the iterative solution algorithm because the eigenvalues are very close to 1. Therefore, from here on, only the high parts of the spectra are shown in detail like in two lower plots in figure 4.5.

At the time step sizes considered here, the eigenvalues caused by rigid body modes do not clearly stand out anymore. Nevertheless, the time step sizes with very slow convergence show well-separated eigenvalues far above 100, while those in the transition zone are limited to about 50 and the quickly converging ones to less the 2.

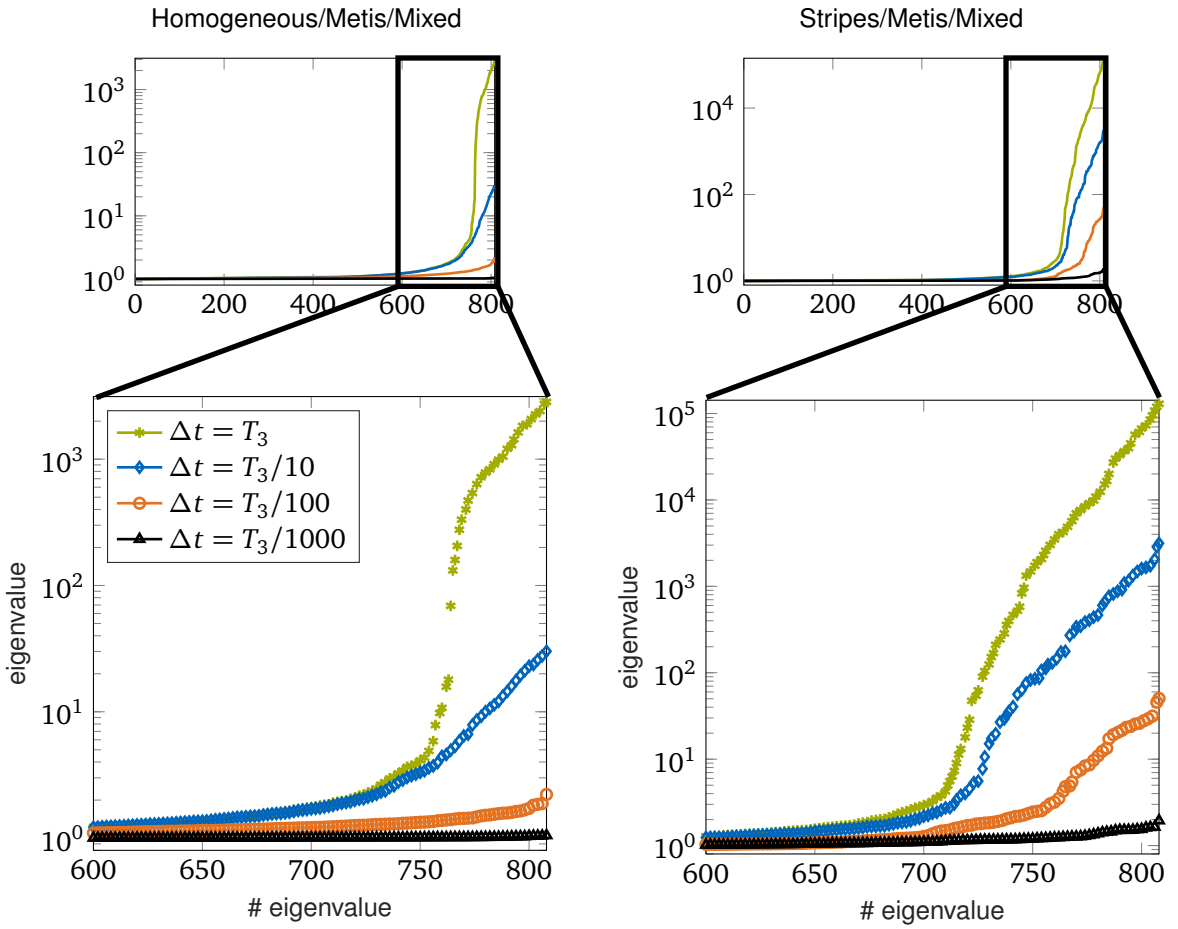


Figure 4.5. Eigenvalue distributions of the preconditioned interface operators HF for the homogeneous and the heterogeneous example and for all four time step sizes. The eigenvalues are sorted in ascending order, and only the Dirichlet preconditioner is considered. Because the spectrum is bounded from below by 1, only eigenvalues significantly larger than 1 are relevant for the convergence of the iterative algorithm. Therefore, this part is shown in detail.

4.2 A Priori Coarse Spaces

In this section, mainly rigid body mode based coarse spaces are discussed and assessed. They are called a priori coarse spaces because they have to be calculated before the solution of the first linear system. Furthermore, the GenEO coarse space for dynamic problems is presented and used as a reference for the rigid body mode based coarse spaces. For reasons of comparability and competitiveness, the performance of the GenEO coarse space when used to the full extent of its capabilities is assessed together with recycling strategies in the next section.

4.2.1 Rigid Body Modes

It was first proposed in [45] to employ the rigid body modes as an auxiliary coarse space when solving problems of structural dynamics with FETI. This is achieved by setting the coarse space $\boldsymbol{C}$, which is used to build the auxiliary coarse space projector $\boldsymbol{P}_C$, to

$$\boldsymbol{C} = \boldsymbol{G} = \begin{bmatrix} \boldsymbol{B}^{(1)}\boldsymbol{R}_I^{(1)} & \boldsymbol{B}^{(2)}\boldsymbol{R}_I^{(2)} & \cdots & \boldsymbol{B}^{(N_s)}\boldsymbol{R}_I^{(N_s)} \end{bmatrix}.$$

In [45], this approach was only applied to homogeneous problems and a large time step size that is suitable to capture the first few lowest structural modes. It was identified as efficient coarse space that is capable to restore convergence and scalability. A variant of this method is to also include the rigid body modes of the interfaces of fixed substructures. However, in the case of using rigid body modes of all substructures, it must be noted that the whole coarse space becomes rank deficient, similar to the natural coarse space for a globally floating structure.

In [34], a modified natural coarse spaced for static problems was proposed. Instead of $\boldsymbol{G}$, the more general space $\boldsymbol{QG}$ was used to build the natural coarse space projector. When the additional matrix $\boldsymbol{Q}$ is identical to the preconditioner, the performance improves especially for heterogeneous problems. This result was confirmed in [84], where another choice for $\boldsymbol{Q}$ was proposed, namely the so-called superlumped preconditioner defined by

$$\boldsymbol{Q} = \boldsymbol{H}_{SL} = \sum_{s=1}^{N_s} \widetilde{\boldsymbol{B}}^{(s)} \boldsymbol{t}^{(s)} \operatorname{diag}\left(\boldsymbol{K}^{(s)}\right) \boldsymbol{t}^{(s)^T} \widetilde{\boldsymbol{B}}^{(s)^T}.$$

It is similar to the lumped preconditioner, which uses the full stiffness matrix is instead of only the diagonal.

This technique was first applied to dynamic problems in [81], where these kinds of preconditioned rigid body modes were employed to solve homogeneous and heterogeneous problems for a large time step size. Similar to the findings in [45], the standard rigid body mode coarse space, i.e., $\boldsymbol{Q} = \boldsymbol{I}$, was found to be efficient for homogeneous dynamic problems. For heterogeneous dynamic problems, significant improvements were shown for the choice $\boldsymbol{Q} = \boldsymbol{H}_{SL}$.

In this section, the performance of these coarse spaces on different time scales is assessed. Furthermore, they are compared to the GenEO modes, which, different from the rigid body modes, depend on the time step size. In addition to the cases $\boldsymbol{Q} = \boldsymbol{I}$ and $\boldsymbol{Q} = \boldsymbol{H}_{SL}$, the cases $\boldsymbol{Q} = \boldsymbol{H}_L$ and $\boldsymbol{Q} = \boldsymbol{H}_D$ are also taken into consideration.

4.2.2 Generalized Eigenvalues in the Overlaps

The GenEO coarse space is of particular interest for this analysis of rigid body mode based coarse spaces because on the one hand, it also delivers local substructure

modes and on the other hand, those modes are already known to set up a well suited coarse space. It solely relies on the Schur complement of local iteration matrix $\boldsymbol{S}_D^{(s)}$ and the preconditioner $\boldsymbol{H}$, which both include the full information about the contribution of the mass matrix as a function of the time step size. Therefore, the GenEO coarse space is expected to take the time step size correctly into account.

To compute the GenEO coarse space, the same generalized eigenvalue problems as for static problems are solved, replacing the Schur complement of the stiffness matrix by the Schur complement of the iteration matrix $\boldsymbol{D}^{(s)}$. The eigenproblem for the substructures $s = 1 \dots N_s$ then reads

$$\boldsymbol{S}_D^{(s)} \boldsymbol{y}^{(s)} = \theta^{(s)} \boldsymbol{B}^{(s)^T} \boldsymbol{H} \boldsymbol{B}^{(s)} \boldsymbol{y}^{(s)} . \tag{4.3}$$

One first important observation is that the presence of the mass matrix in $\boldsymbol{S}_D^{(s)}$ prevents zero eigenvalues for $\boldsymbol{y}^{(s)}$ describing rigid body modes. In other words, different from statics, the lowest GenEO modes of floating substructures are not identical to rigid body modes anymore. However, one can expect the first three GenEO modes of floating substructures to converge to rigid body modes for very large time step sizes. This leads to the following consideration: if the GenEO modes are especially well suited to build a coarse space, rigid body modes should be a good choice for large time steps. For an decreasing time step size then, the evolution of the first three GenEO modes should illustrate the necessary change in the coarse space and reveal when and why a rigid body mode based coarse space suits well or not.

Figure 4.6 shows the GenEO values for the heterogeneous problem for all four time step sizes and for three substructures. In the top row, the GenEO values for one of the two fixed substructures are shown, and in the two rows below, the GenEO values for two floating substructures are shown. The GenEO values for the homogeneous problem can be found in appendix A in figure A.2.

The results shown in figure 4.6 illustrate the following findings. First, for very large time steps, the three smallest GenEO values of floating substructures diverge to zero as expected. However, this is more obvious for the homogeneous case shown in appendix A in figure A.2. Second, it is not optimal to choose always the smallest three GenEO modes of floating substructures. For very large time steps, more than three modes might be important, and for small time steps, also non floating substructures become important. Third, the jump criterion is not suitable to identify the optimal number of GenEO modes per substructure. It is very difficult to identify a unique, significant jump. Furthermore, also for the smallest time step sizes, significant jumps can be identified. It is illustrated in more detail later that problems with very small time step sizes do not require any coarse space at all due to their good conditioning. As a consequence, the jump criterion should not be used exclusively.

The threshold criterion was applied in figure 4.7 to select the same total number of 48 GenEO modes. Instead of selecting 3 modes of each floating substructure, those 48 modes with the lowest GenEO values among all substructures were chosen. Figure 4.7 shows the resulting number of selected modes per substructure for the homogeneous and the heterogeneous example for all four time step sizes. The selected number of GenEO modes for the fixed substructures, which have the number $s = 10$ and $s = 13$, are highlighted.

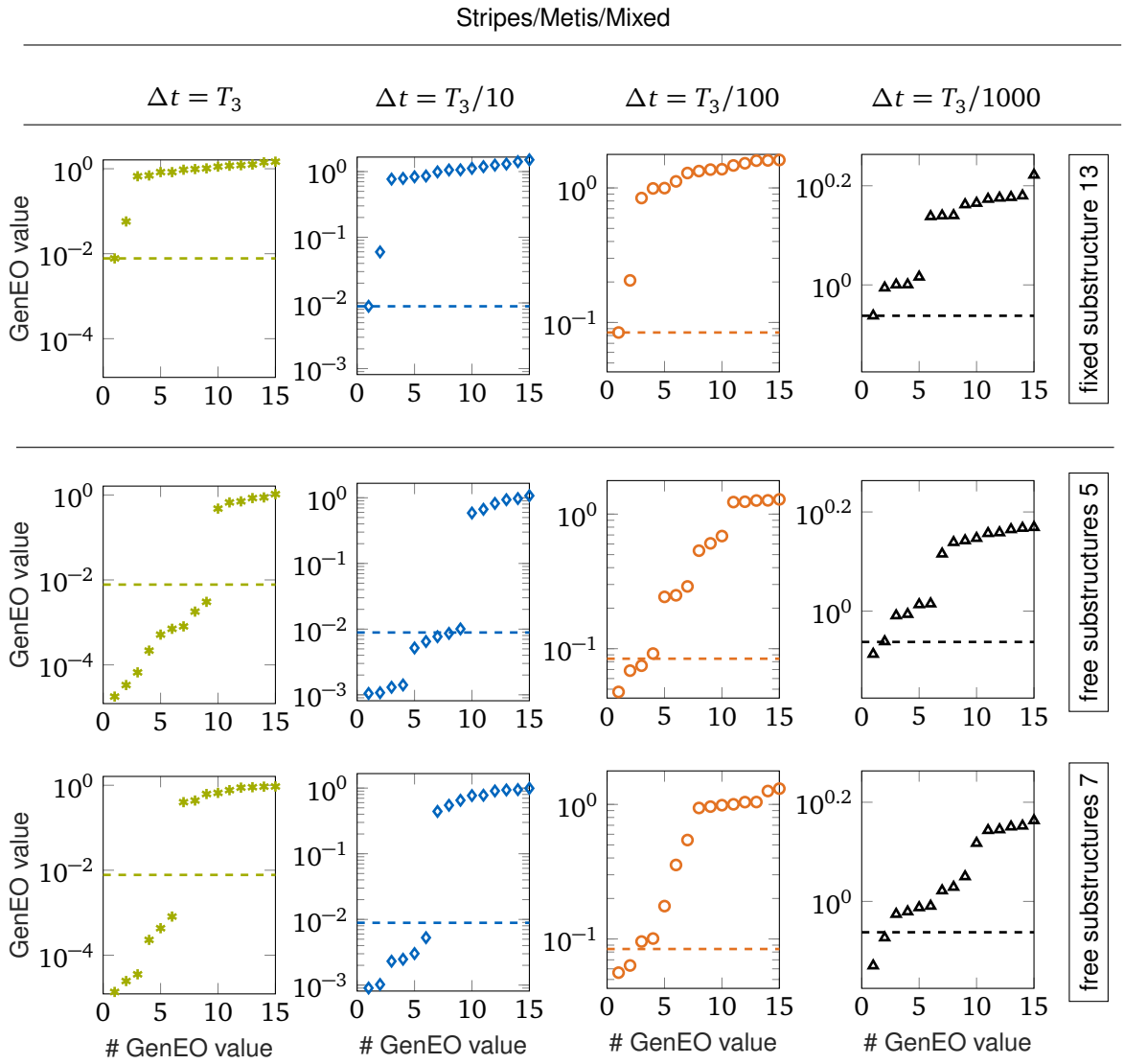


Figure 4.6. GenEO values of three substructures in the heterogeneous example. For each time step size, the smallest GenEO value of the fixed substructure is indicated by a dashed line. For large time steps, the free substructures have much smaller and thus more important GenEO values than the fixed substructure. For smaller time steps, the distributions of free and fixed substructures become more similar, and the overall level of the GenEO values rises.

The distributions show that for the largest time step $\Delta t = T_3$, exactly three modes are chosen for each floating substructure of the homogeneous problem. This happens because of the dominating role of the stiffness matrix, i.e., the proximity to the static problem, and it indicates that rigid body modes might be a well suitable choice. In contrast, for the heterogeneous problem, this is not even true for the largest time step. For smaller time steps, the distributions begin to change and the fixed substructures are selected as well. From this it follows for the GenEO coarse space that, in the general case, neither three local contributions per substructure nor a restriction to floating substructures is optimal within the assessed practically relevant range of time step sizes.

Figure 4.7. Number of selected GenEO modes per substructure $k^{(s)}$ for different time step sizes. The 48 GenEO modes that have the lowest GenEO values among all substructures are selected. For large time steps, three modes are considered for most substructures and only the fixed substructures are less important. For smaller time steps, this rule of thumb no longer holds and local effects lead to distributions which depend on the mesh and its decomposition.

In the remaining part of this section, the first three GenEO modes of each floating substructure are used as reference to analyze the rigid body mode coarse spaces. As it was shown above, it is necessary to apply the threshold criterion without a restriction to 48 modes to enable the full capabilities of the GenEO method. Such a sound application of it is presented in the subsequent section 4.3.

4.2.3 Local Mode Shapes and Subspace Angles

As a next step, the rigid body modes of each floating substructure are directly compared to the first three GenEO modes of each substructure. In particular, the evolution of the GenEO modes from very large to very small time steps is studied.

The local rigid body modes, which are used to build the different variants of rigid body mode coarse spaces, are always the unprocessed rigid body modes $\boldsymbol{y}^{(s)}_{\mathrm{rbm},i}$ defined by

$$\boldsymbol{R}^{(s)}_I = \left[\boldsymbol{y}^{(s)}_{\mathrm{rbm},1}, \boldsymbol{y}^{(s)}_{\mathrm{rbm},2}, \boldsymbol{y}^{(s)}_{\mathrm{rbm},3}\right], \qquad \boldsymbol{R}^{(s)}_I = \ker\left(\boldsymbol{S}^{(s)}_K\right).$$

Only the computation of the final contribution $\boldsymbol{C}^{(s)}_{\mathrm{rbm},i}$ to the coarse space $\boldsymbol{C}$ of a local mode $\boldsymbol{y}^{(s)}_{\mathrm{rbm},i}$ may include the application of a preconditioning step as it reads

$$\boldsymbol{C}^{(s)}_{\mathrm{rbm},i} = \boldsymbol{H}_* \boldsymbol{B}^{(s)} \boldsymbol{y}^{(s)}_{\mathrm{rbm},i}.$$

The placeholder $\boldsymbol{H}_*$ is replaced either by $\boldsymbol{H}_D$, $\boldsymbol{H}_L$, $\boldsymbol{H}_{SL}$, or $\boldsymbol{I}$. It should be noted that as soon as $\boldsymbol{y}^{(s)}_{\mathrm{rbm},i}$ is transformed into the global interface space by multiplication with $\boldsymbol{B}^{(s)}$, its interpretation as a local deformation of one substructure is lost.

This entails two major conclusions. First, for analyzing local mode shapes of substructures, only the unprocessed rigid body modes $\boldsymbol{y}^{(s)}_{\mathrm{rbm},i}$ are available, and they are identical for all variants of rigid body mode coarse spaces. Second, they are independent of the time step size. In contrast, the GenEO modes $\boldsymbol{y}^{(s)}_{\mathrm{GenEO},i}$ are defined as the solutions of

$$\boldsymbol{S}^{(s)}_D \boldsymbol{y}^{(s)}_{\mathrm{GenEO}} = \theta^{(s)} \boldsymbol{B}^{(s)^T} \boldsymbol{H} \boldsymbol{B}^{(s)} \boldsymbol{y}^{(s)}_{\mathrm{GenEO}}$$

and change with the time step size. The computation of the final contribution $\boldsymbol{C}_{\mathrm{GenEO},i}$ of a local mode $\boldsymbol{y}^{(s)}_{\mathrm{GenEO},i}$ reads

$$\boldsymbol{C}^{(s)}_{\mathrm{GenEO},i} = \boldsymbol{H}_D \boldsymbol{B}^{(s)} \boldsymbol{y}^{(s)}_{\mathrm{GenEO},i},$$

which is identical to the computation in rigid body mode coarse spaces. The only difference to the processing in rigid body mode coarse spaces is that the GenEO coarse space always uses the Dirichlet preconditioner $\boldsymbol{H}_D$. For this reason, a direct comparison between $\boldsymbol{y}^{(s)}_{\mathrm{rbm},i}$ and $\boldsymbol{y}^{(s)}_{\mathrm{GenEO},i}$ should lead to meaningful conclusions. Of course, the GenEO modes deliver true accelerations, while the rigid body modes have the unit of a displacement. Nevertheless, when $\boldsymbol{y}^{(s)}_{\mathrm{rbm},i}$ is considered as an acceleration, it describes a rigid body motion as well.

The shapes of these local modes are visualized in figure 4.8 for the homogeneous example and in figure 4.9 for the heterogeneous example. In both cases, the same floating substructure is considered. As these mode shapes effectively depict displacements, while $\boldsymbol{y}^{(s)}_{\mathrm{GenEO},i}$ is an acceleration, a short explanation is required. From the accelerations, displacements can be computed, e.g., by a Newmark integration scheme. With a certain initial acceleration $\ddot{\boldsymbol{u}}^{(s),0}$, zero initial displacement, and zero initial velocity, one obtains the relation

$$\boldsymbol{u}^{(s),1} = \frac{1}{2}\Delta t^2 \ddot{\boldsymbol{u}}^{(s),0}$$

Consequently, plotting the accelerations as if they were displacements, is equivalent to computing the displacements after one fictitious time step, starting with zero displacement and zero velocity. This method is thus considered suitable to visualize the shapes of the local GenEO modes $\boldsymbol{y}^{(s)}_{\mathrm{GenEO},i}$.

The $\boldsymbol{y}^{(s)}_{\text{GenEO},i}$ are only defined on the interface dof of a substructure, e.g., for $\boldsymbol{u}^{(s)}_I$. Therefore, the plotted displacements (originally accelerations) of interface dof I and internal dof r were computed by

$$\begin{aligned}\boldsymbol{u}^{(s)}_{I,\text{GenEO},i} &= \boldsymbol{y}^{(s)}_{\text{GenEO},i}\,,\\ \boldsymbol{u}^{(s)}_{r,\text{GenEO},i} &= -\boldsymbol{D}^{(s)^{-1}}_{rr}\boldsymbol{D}^{(s)}_{rI}\boldsymbol{y}^{(s)}_{\text{GenEO},i}\,.\end{aligned}$$

Another important aspect for the interpretation of the homogeneous mode shapes in figure 4.8 is the coloring. To illustrate whether the modes are dominated more by global or more by local effects, the stresses corresponding to the displacements have been computed. The displacements used to calculate the stresses have been normalized such that the largest magnitude of a nodal displacement is equal to 1 within each mode. Because this scaling works well for computing comparable stress distributions but not for plotting a shape, the displacements used to plot the shape have been scaled differently than the displacements used to compute the stresses. In other words, the visualized mode shapes correspond to the plotted stresses not in a quantitative, but only in a qualitative way. To apply a common color scale for all investigated time step sizes a logarithmic scaling was required, which is indicated by a color bar in the upper left corner. The rigid body modes are a by-product of the factorization of the stiffness matrix. For this reason, they are not split into purely translational and rotational modes. More important, their coloring represents zero stresses throughout the whole substructure.

Three different regimes can be identified for the homogeneous example in figure 4.8. In the first regime, which is represented by time step size $\Delta t = T_3$, the GenEO modes look very similar to rigid body modes and the coloring confirms almost no deformation. In the second regime, which is represented by $\Delta t = T_3/10$, the modes have apparently changed as they clearly show deformation now. However, an important property is their homogeneous coloring, which indicates that the deformation is equally distributed among the whole substructure. The next smaller time step $\Delta t = T_3/100$ in the bottom row represents the third regime. It is classified by very localized deformations that mainly affect only few elements near the interface, while the major part of the structure is undeformed.

These three regimes can be identified similarly for the heterogeneous example, but they are not separated as sharply as for the homogeneous one. The first and the second regime are shown in figure 4.9. The first regime, where the GenEO modes closely resemble rigid body modes shrinks down to almost only the first GenEO mode of $\Delta t = T_3$ in the second row. Nevertheless, the second and the third mode still include a fair amount of rigid body motion. The second regime is distributed over $\Delta t = T_3/10$ and $\Delta t = T_3/100$. It is characterized by deformations that include less rigid body motion but are still distributed over a significant part of the substructure and are not limited to a few single elements. The third regime of highly localized effects can be observed for the heterogeneous problem only for the smallest time step size $\Delta t = T_3/1000$, which is illustrated for the same substructure in the upper part of figure 4.10. The deformations are highlighted by circles and mainly comprise only a single node. To investigate the dependence of this effect on the mesh size, the same global physical structure was remeshed with elements that are approximately three

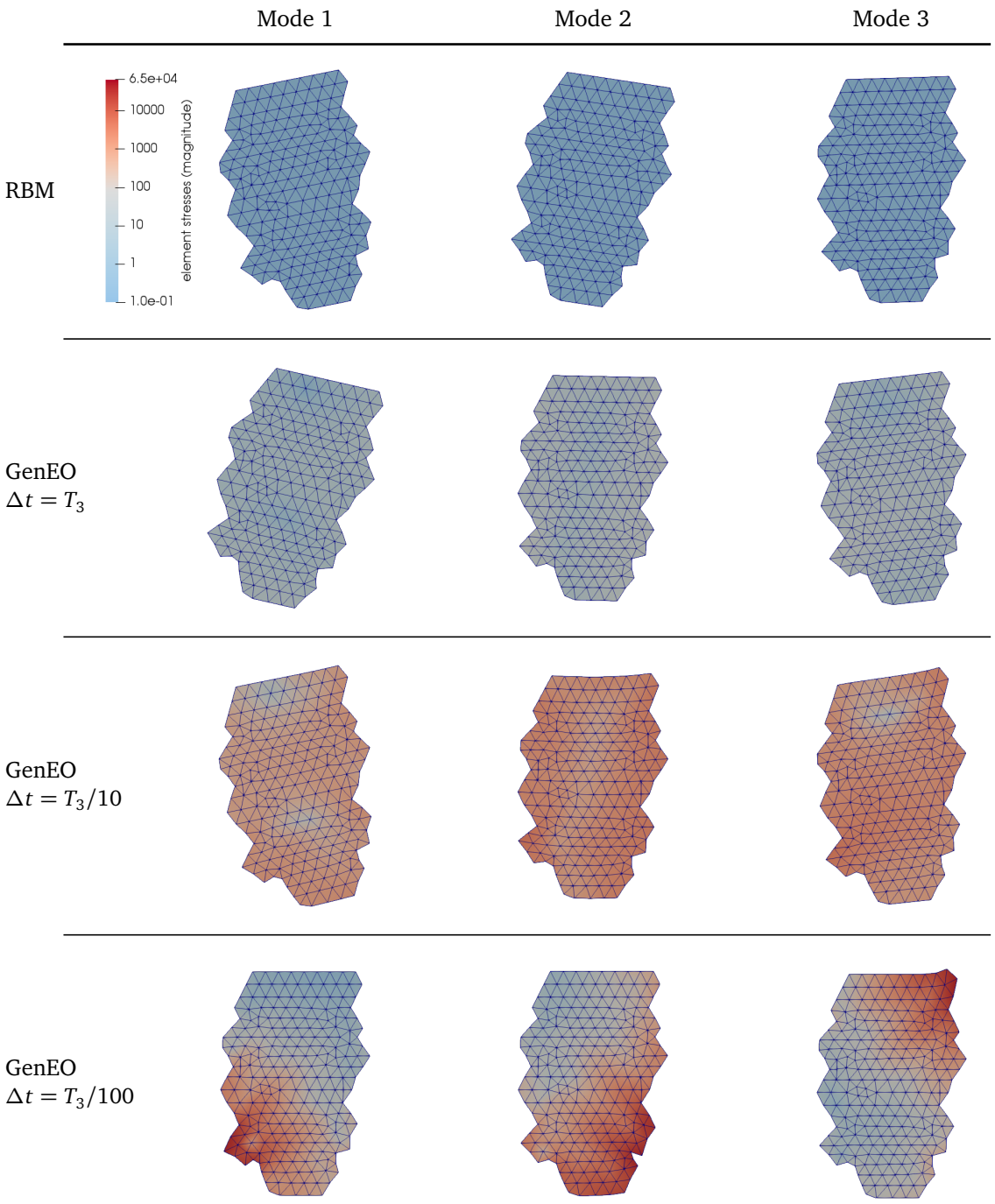


Figure 4.8. Visualization of the rigid body modes and the GenEO modes as displacements for the homogeneous example. The coloring represents the corresponding stresses and visualizes if the deformation is distributed uniformly or if very localized effects prevail. The displacement used to compute the stresses are scaled differently than the plotted displacement.

times smaller in size. The resulting structure is illustrated as example structure 2 in appendix A. For this finer mesh, the GenEO modes have been computed as well for the time step size $\Delta t = T_3/1000$. The period T_3 in this case was recomputed for the finer mesh, resulting in a slightly larger time step due to the stiffening of the coarse

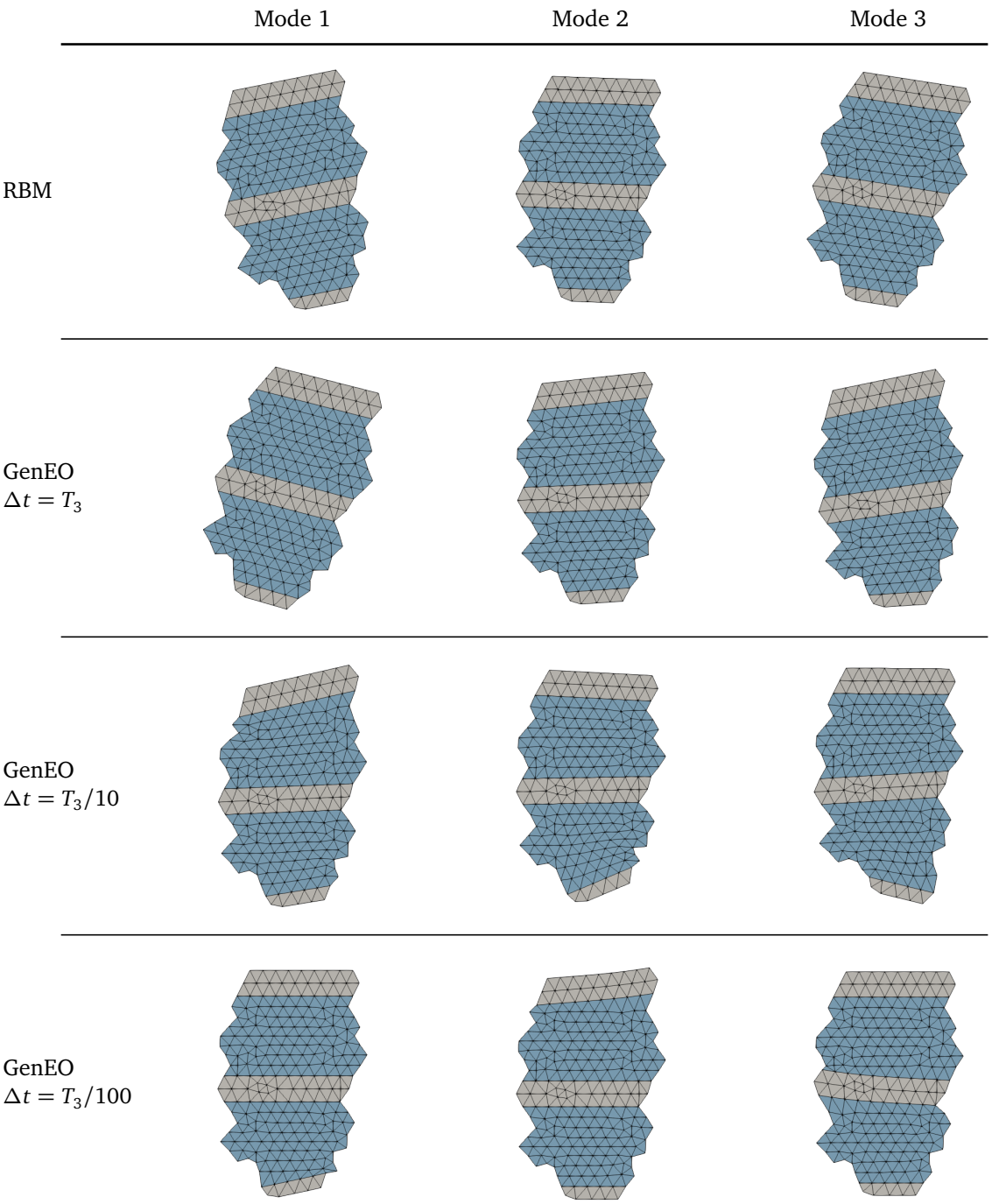


Figure 4.9. Visualization of the rigid body modes and the GenEO modes as displacements for the heterogeneous example. In favor of indicating the different materials by different colors, a coloring based on stresses was omitted.

mesh. The lower part of figure 4.10 shows the GenEO modes for one floating substructure, which exhibit the same highly local mesh modes like the coarser standard example. This confirms that these extremely localized effects are not mainly induced by the mesh size, but represent a change in important solution spaces when the time step size varies.

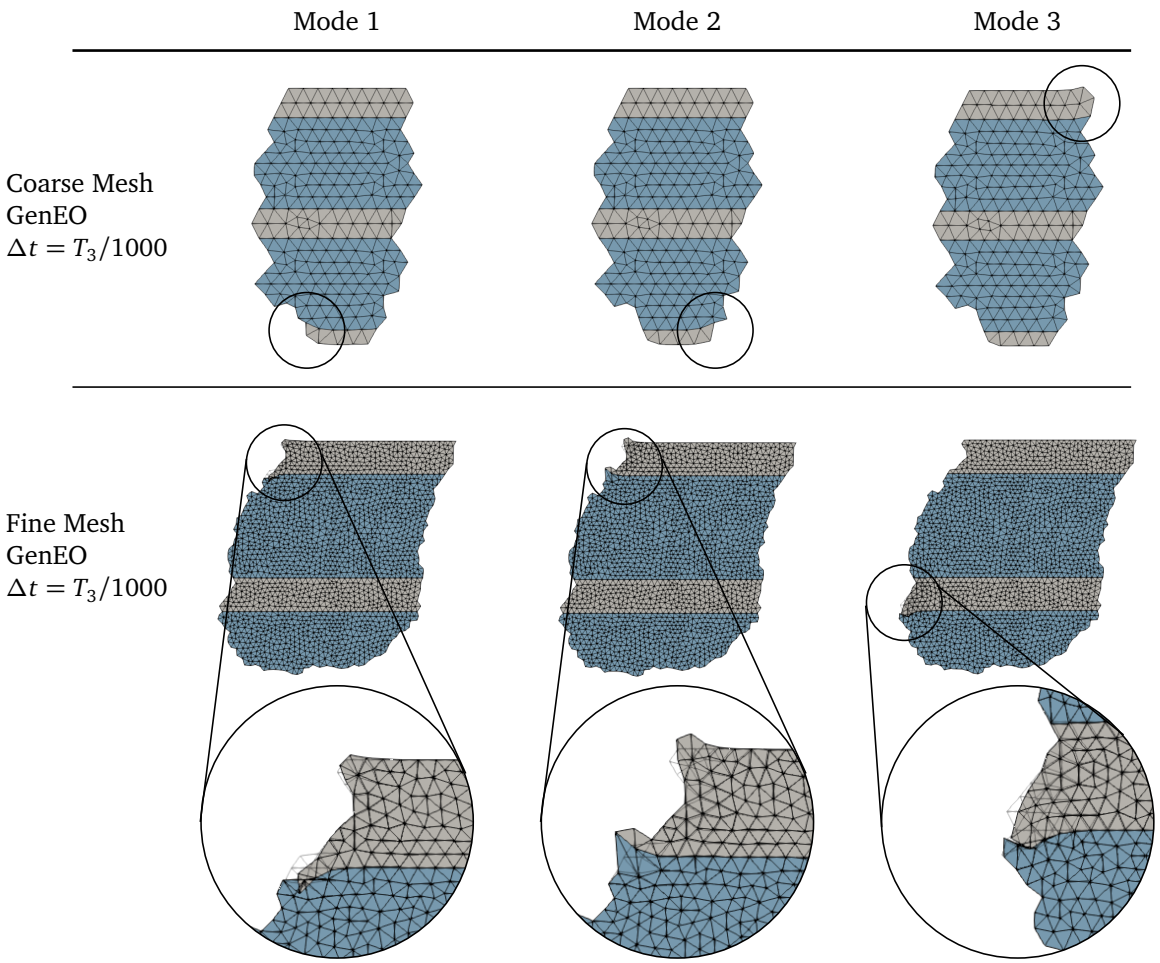


Figure 4.10. Visualization of the GenEO modes for very small time steps for the heterogeneous example. In the lower part, the GenEO modes for the same $\Delta t / T_3$ ratio, and for the same physical structure with a significantly smaller mesh are shown.

The observations made in the analysis of the local mode shapes allow a first prediction about the requirements on efficient coarse spaces in structural dynamics, which can be summarized as follows:

1. As expected, the GenEO modes resemble rigid body modes for very large time steps, called first regime in the discussion above. This indicates that for very large time steps, rigid body modes would be well suited to build a coarse space from.

2. For middle sized time steps, effects that are still distributed very uniformly throughout the substructure are responsible for the high part of the spectrum of the preconditioned interface operator. This range is called the second regime. They cause a strong coupling between the substructures and can presumably be handled efficiently by a coarse space. However, the rigid body modes might not fit optimally because the GenEO modes already show a strong overall deformation.

Algorithm 4: Computation of principal angles between subspaces

Given: $\boldsymbol{A}$, $\boldsymbol{B}$

Compute orthonormal bases of $\boldsymbol{A}$ and $\boldsymbol{B}$
$\boldsymbol{A} = \boldsymbol{A}_Q \boldsymbol{A}_R, \quad \boldsymbol{B} = \boldsymbol{B}_Q \boldsymbol{B}_R \quad \text{with} \quad \boldsymbol{A}_Q^T \boldsymbol{A}_Q = \boldsymbol{I}, \quad \boldsymbol{B}_Q^T \boldsymbol{B}_Q = \boldsymbol{I}$

Compute a singular value decomposition
$\boldsymbol{A}_Q^T \boldsymbol{B}_Q = \boldsymbol{U} \mathrm{diag}(\sigma_1, \ldots, \sigma_q) \boldsymbol{V}^T$

$$\boldsymbol{S} = \begin{cases} \boldsymbol{B}_Q - \boldsymbol{A}_Q (\boldsymbol{A}_Q^T \boldsymbol{B}_Q) & \text{if} \quad \mathrm{rank}\left(\boldsymbol{A}_Q\right) \geq \mathrm{rank}\left(\boldsymbol{B}_Q\right), \\ \boldsymbol{A}_Q - \boldsymbol{B}_Q (\boldsymbol{B}_Q^T \boldsymbol{A}_Q) & \text{otherwise} \end{cases}$$

Compute a singular value decomposition
$\boldsymbol{S} = \boldsymbol{U} \mathrm{diag}(\mu_1, \ldots, \mu_q) \boldsymbol{V}^T$

Compute principal angles
$$\Psi_i = \begin{cases} \mathrm{acos}\,(\sigma_i) & \text{if} \quad \sigma_i^2 < 1/2, \\ \mathrm{asin}\,(\mu_i) & \text{if} \quad \mu_i^2 \leq 1/2 \end{cases}$$

3. For even smaller time steps towards Courant's condition, where at some point implicit time stepping schemes can be replaced by cheaper explicit schemes, the effects that are identified as the most important ones begin to be extremely localized. They can thus be assumed to cause a minimum of coupling between the substructures and are already known to be handled well by the preconditioned iterative solution algorithm. Under these conditions, a coarse space might be unnecessary and inefficient, regardless of its chosen subspace.

The direct comparison between GenEO modes and rigid body modes is concluded by a more objective and mathematical measure than visually comparing the appearance of mode shapes. In fact, the GenEO modes $\boldsymbol{y}_{\mathrm{GenEO},i}^{(s)}$ and the rigid body modes $\boldsymbol{y}_{\mathrm{rbm},i}^{(s)}$, where $i = 1, 2, 3$, span each a 3-dimensional subspace of the complete interface space I of a substructure s. A convenient method to compare the proximity of subspaces inside a common space is the method of subspace angles. Profound introductions can be found in [96, 97], and the specific method used here to compute the so-called principal angles between two subspaces was published in [98].

The exact procedure is listed in algorithm 4. It computes the principal angles Ψ_i between two subspaces $\mathscr{A}$ and $\mathscr{B}$, which are spanned by the columns of $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. The meaning of principal angles can be illustrated most clearly by the recursive description given in [96]. It first defines the smallest principle angle by

$$\cos(\Psi_1) = \max_{\boldsymbol{a} \in \mathscr{A}} \max_{\boldsymbol{b} \in \mathscr{B}} \boldsymbol{a}^T \boldsymbol{b}\,, \quad \|\boldsymbol{a}\| = 1, \|\boldsymbol{b}\| = 1\,.$$

If the smallest principal angle has been found for the so-called principal vectors $\boldsymbol{a} = \boldsymbol{a}_1$ and $\boldsymbol{b} = \boldsymbol{b}_1$, the second smallest principal angle Ψ_2 between $\mathscr{A}$ and $\mathscr{B}$ is defined as the smallest angle between the orthogonal complement of $\mathscr{A}$ with respect to $\boldsymbol{a}_1$, and the orthogonal complement of $\mathscr{B}$ with respect to $\boldsymbol{b}_1$, and so on.

It follows from this definition that $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{m \times k}$ and furthermore that for

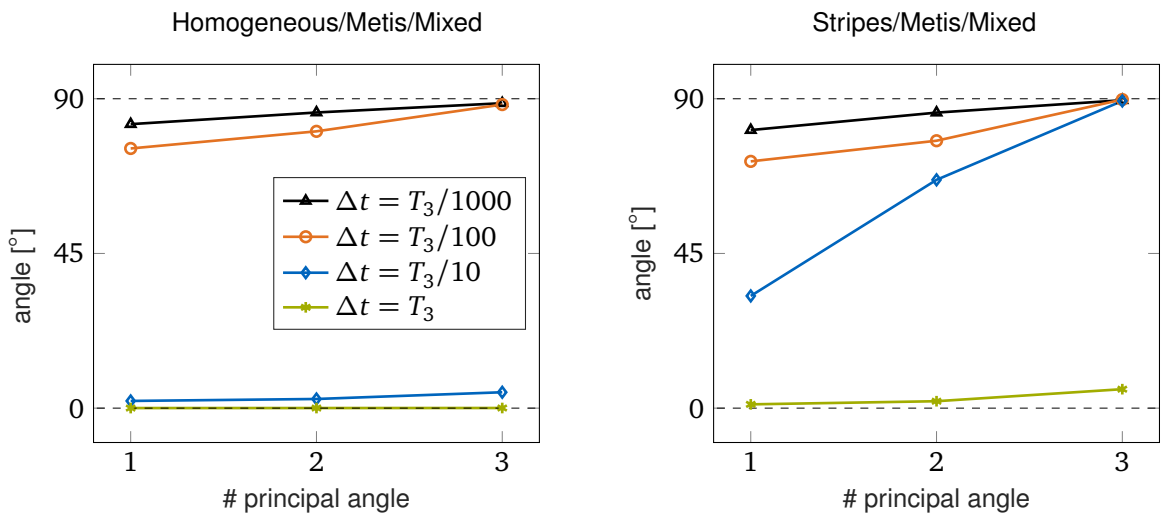


Figure 4.11. Principal angles or subspace angles between the subspaces spanned by the rigid body modes, which are constant over all time step sizes, and the GenEO modes, which are dependent on the time step size. Small principle angles indicate overlapping parts of the subspaces, large principle angles close to 90° indicate rather orthogonal parts.

$n < k$, exactly n principal angles exist. Each of the very small principle angles that are close to 0° represents a direction that is almost covered by both subspaces, i.e., where the subspaces overlap. Each of the very large principle angles close to 90° represents a direction that is covered by one subspace but is almost orthogonal to the other subspace, i.e., where the subspaces are orthogonal.

The resulting principal angles between GenEO modes and rigid body modes are shown in figure 4.11 for the homogeneous and the heterogeneous example for all four considered time step sizes. Like in the visualization of the mode shapes shown earlier, the rigid body modes do not depend on the time step size. Thus, each set of three principal angles relates the same set of rigid body modes $\boldsymbol{y}^{(s)}_{\text{rbm},i}$ against the GenEO modes $\boldsymbol{y}^{(s)}_{\text{GenEO},i}$ of the specific time step size.

For the homogeneous example, a very good agreement with the local mode shape visualizations can be observed. For the largest time step, the GenEO modes are found to be very close to the rigid body modes. For $\Delta t = T_3/10$, the subspace angles reveal what is difficult to see in the mode shape visualizations. The visualization of the GenEO modes already showed overall deformation, but the low principal angles clearly reveal that the rigid body motion still dominates. The large principle angles close to 90° confirm the observation of very local effects for $\Delta t = T_3/100$, which have almost no relation to rigid body modes anymore.

Like in the local mode shape visualizations, the results of the heterogeneous example are not separated as sharply as those of the homogeneous one. Already for the largest time step size $\Delta t = T_3$, at least one principal angle shows that the GenEO modes are not aligned well with rigid body modes. Except for one principal angle, the GenEO modes show no close relation to the rigid body modes anymore for larger time step

sizes.

The following conclusions about the efficiency of rigid body mode based coarse spaces can be drawn from the preceding observations: for the homogeneous example, one would expect that rigid body modes work as well as GenEO modes for the two largest time step sizes $\Delta t = T_3$ and $\Delta t = T_3/10$. For the heterogeneous example, this is expected only for the largest time step size $\Delta t = T_3$. In both cases, the results suggest a bad efficiency of rigid body mode based coarse spaces for smaller time steps.

4.2.4 Spectral Efficiency

It was already justified theoretically and by experiment that the eigenvalue distribution of the operator which is effectively solved by the conjugate gradient algorithm almost exclusively predicts the rate of convergence. In case of the preconditioned interface problem of FETI, deflated by an auxiliary coarse space, this effective operator was identified as $\boldsymbol{P}_C\boldsymbol{H}\boldsymbol{P}_C^T\boldsymbol{F}$. Consequently, the spectra of this operator are now shown for the different variants of rigid body mode coarse spaces and compared to those of the GenEO mode coarse space. Furthermore, the spectra if the undeflated preconditioned operators are included for comparison. From this point on, the different rigid body mode coarse spaces lead to different results because the full coarse space $\boldsymbol{C}$, including the different choices for $\boldsymbol{Q}$, becomes relevant.

Figure 4.12 shows the spectra of the most relevant algorithms for the homogeneous example. The results of all considered algorithms for the homogeneous example can be found in appendix A in figure A.3 and figure A.4. The algorithms considered in figure 4.12 are all based on the standard FETI algorithm using the Dirichlet preconditioner. They are named and distinguished by the employed coarse space as listed in the following.

none No coarse space is employed.

GenEO The GenEO coarse space using the 48 GenEO modes with the smallest GenEO values is used. Note that this means that not necessarily three modes per substructure are used, as shown in figure 4.7. The results of a GenEO coarse space that uses exactly three modes per floating substructure is included in the results in the appendix.

RBM The rigid body modes are directly used as coarse space, i.e., $\boldsymbol{C} = \boldsymbol{Q}\boldsymbol{G}$ with $\boldsymbol{Q} = \boldsymbol{I}$

GEV The 48 eigenvectors of the preconditioned interface operator $\boldsymbol{H}\boldsymbol{F}$ that exhibit the largest eigenvalues are employed as coarse space. This is considered as the optimal coarse space in case that a size of 48 is prescribed. As this spectrum is exactly congruent with the spectrum of *none*, except for less eigenvalues, it is not plotted. Instead, only the corresponding condition numbers are shown.

Figure 4.12 also includes a table with the effective condition numbers of the resulting

Homogeneous/Metis/Mixed

$T_3/\Delta t$	none (——)	GenEO (——)	RBM (- - - -)	GEV
1	2.85×10^3	1.14×10^1	1.21×10^1	1.07×10^1
10	3.02×10^1	5.37×10^0	7.61×10^0	4.44×10^0
100	2.21×10^0	1.58×10^0	2.03×10^0	1.38×10^0
1000	1.05×10^0	1.03×10^0	1.04×10^0	1.01×10^0

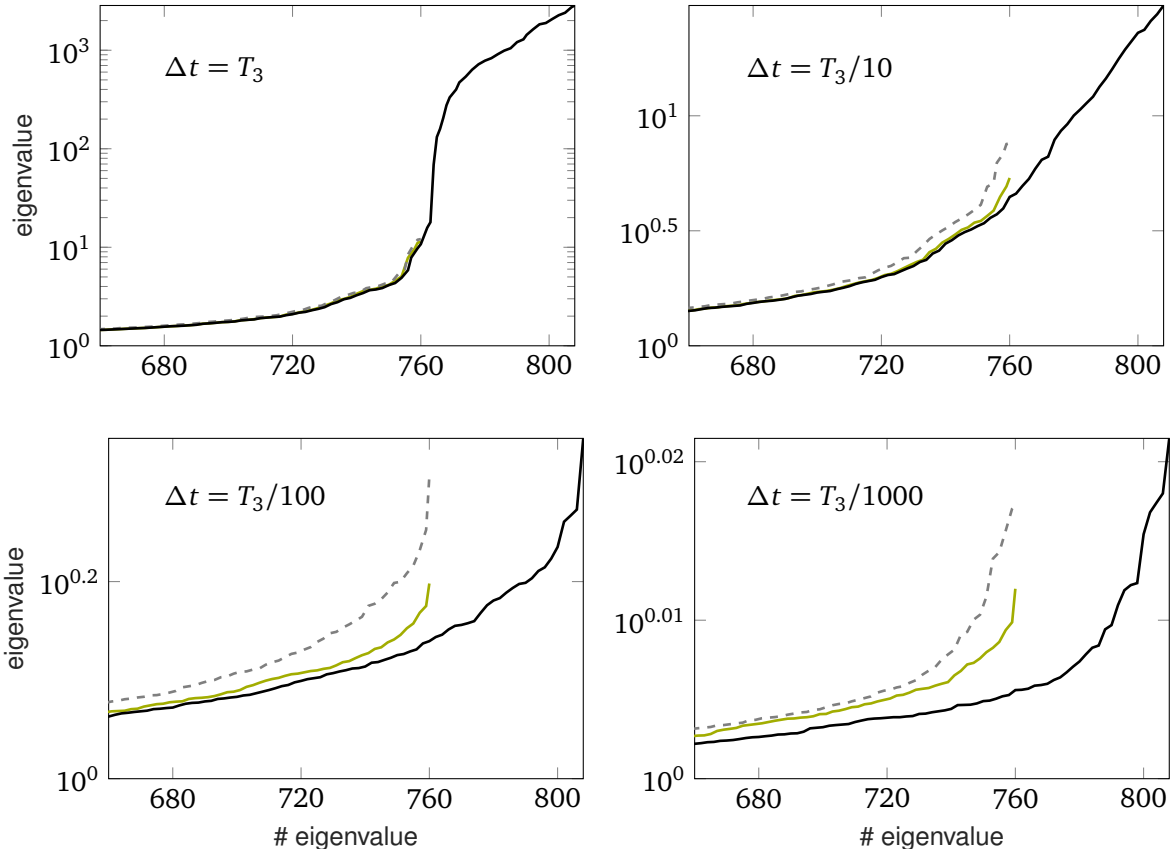


Figure 4.12. Spectra of the deflated operator $\boldsymbol{P}_C\boldsymbol{H}\boldsymbol{P}_C^T\boldsymbol{F}$ for the homogeneous example. Different variants of coarse spaces are compared against the undeflated variant for all four time step sizes. In the upper part, a table including the plot legend and the resulting effective condition numbers is shown.

deflated operators. Because the spectra are bounded from below by one, the effective condition numbers are practically identical to the largest eigenvalue.

In very good agreement with the prior results, the rigid body mode coarse space *RBM* manages effectively to improve the spectrum for the two largest time steps $\Delta t = T_3$ and $\Delta t = T_3/10$. The other rigid body mode coarse space variants have been omitted here because they are more costly and show almost identical or worse performance compared to *RBM*. For time step sizes around $\Delta t = T_3/10$, similar results were found in [45, 81]. For the two smaller time steps $\Delta t = T_3/100$ and $\Delta t = T_3/1000$, *RBM*

Stripes/Metis/Mixed

$T_3/\Delta t$	none (——)	GenEO (——)	RBM (- - -)	HDRBM (——)	HSLRBM (——)	GEV
1	1.30×10^5	6.23×10^3	2.11×10^4	1.10×10^4	2.11×10^4	3.49×10^3
10	3.15×10^3	3.38×10^2	2.57×10^3	1.18×10^3	1.94×10^3	1.37×10^2
100	5.05×10^1	1.06×10^1	5.00×10^1	2.58×10^1	3.50×10^1	3.18×10^0
1000	1.95×10^0	1.49×10^0	1.94×10^0	1.78×10^0	1.81×10^0	1.24×10^0

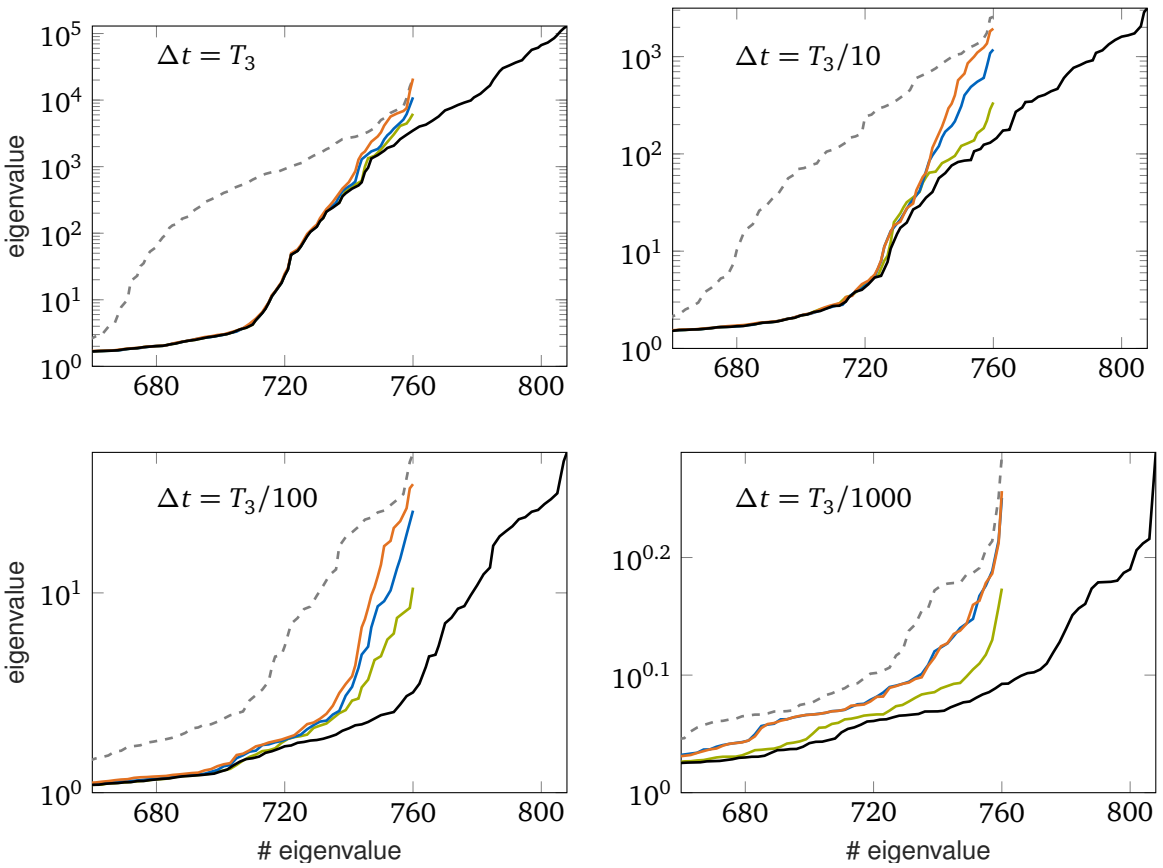


Figure 4.13. Spectra of the deflated operator $\boldsymbol{P}_C\boldsymbol{H}\boldsymbol{P}_C^T\boldsymbol{F}$ for the heterogeneous example. Different variants of coarse spaces are compared against the undeflated variant for all four time step sizes. In the upper part, a table including the plot legend and the resulting effective condition numbers is shown.

clearly declines in comparison to *GenEO*. In fact, its spectra even resemble the undeflated ones and appear to be merely shifted to the left. This means that mainly small eigenvalues around 1 on the lower end of the spectrum have been removed by the projection, which does not affect the rate of convergence at all. Interestingly, also *GenEO* is not able to effectively capture the largest 48 eigenvalues. However, most important for the two smaller time steps is that the condition numbers of the undeflated operator are already lower than 2, which indicates that a coarse space is not efficient or necessary at all.

Figure 4.13 shows the results for the heterogeneous example, which are more differentiated than in the homogeneous case. Again, only the most relevant coarse space variants are shown, which are the variants already shown in the homogeneous case and additionally the following ones.

HSLRBM The rigid body modes are preconditioned by the superlumped preconditioner, i.e., $\boldsymbol{C} = \boldsymbol{QG}$ with $\boldsymbol{Q} = \boldsymbol{H}_{SL}$.

HDRBM The rigid body modes are preconditioned by the Dirichlet preconditioner, i.e., $\boldsymbol{C} = \boldsymbol{QG}$ with $\boldsymbol{Q} = \boldsymbol{H}_{D}$.

The results of all considered algorithms for the heterogeneous example can be found in appendix A in figure A.5 and figure A.6. First of all, *RBM* performs very bad, even for the largest time step for which the subspace angles predicted at least a moderate agreement between the rigid body modes and the GenEO modes. This can be explained by the different methods to construct the coarse space. For *GenEO*, a coarse space contribution is computed by $\boldsymbol{C}^{(s)}_{\text{GenEO},i} = \boldsymbol{H}_D \boldsymbol{B}^{(s)} \boldsymbol{y}^{(s)}_{\text{GenEO},i}$, while for *RBM*, it is computed by $\boldsymbol{C}^{(s)}_{\text{rbm},i} = \boldsymbol{B}^{(s)} \boldsymbol{y}^{(s)}_{\text{rbm},i}$. Consequently, different coarse spaces result even in the case of perfect congruence between the GenEO modes and the rigid body modes. Instead, the coarse space of *HDRBM* is constructed as $\boldsymbol{C}^{(s)}_{\text{rbm},i} = \boldsymbol{H}_D \boldsymbol{B}^{(s)} \boldsymbol{y}^{(s)}_{\text{rbm},i}$ and should thus show the expected similarity.

In agreement with these considerations and the subspace angles investigated before, *GenEO* and *HDRBM* show the expected alignment for the largest time step size $\Delta t = T_3$ in figure 4.13. For $\Delta t = T_3/10$ and $\Delta t = T3/100$, the performance of all rigid body based variants declines significantly. Similar to the homogeneous example, the spectral efficiency of the coarse spaces in terms of catching the highest 48 eigenvalues of the undeflated, preconditioned operator is bad but at the same time assumed to be irrelevant because the undeflated spectrum is already very well conditioned.

4.2.5 Computational Efficiency

To complete the assessment of the efficiency of the considered a priori, rigid body mode based coarse spaces, 30 time steps are computed for both example problems in order to measure computational cost.

Measuring computational cost of a parallel domain decomposition method is a particularly difficult task. Usually, one is mainly interested in the wall clock time the computation takes to complete as well as in the memory requirement. To generate meaningful measurements of these quantities, a high-performance code in which all operations are implemented in the most efficient and thus fastest way available is necessary. This includes in particular the communication among the computing nodes of a distributed parallel computer. If one aspect, be it of computational or communicational character, is implemented badly, an algorithm relying on this specific operation might deliver worse results than another actually superior algorithm

that does not depend on the badly implemented operation. Such kind of high-performance code whose efficiency throughout all implemented operations has been proved has not been available for this work. Furthermore, the aim of this work is to assess the algorithms in terms of purely algorithmic and mathematical measures and not in terms of computer science. Consequently, measurements of wall clock time or processor time have not been taken.

One important numerical measure to quantify the computational cost of a classical FETI algorithm is the number of iterations needed to solve the linear system of each time step. Of course, the cost of one iteration for a specific system depends on the preconditioner used. In the results discussed here, only the Dirichlet preconditioner was applied and therefore the cost of the preconditioner is always the same. The main difference between the algorithms assessed in this chapter is their coarse space. In particular, the *none* algorithm is not using any coarse space, while all others employ a coarse space of size 48. Finally, this leads to the question how much cost is generated by the auxiliary coarse space projections in relation to the cost saved by requiring less iterations than without a coarse space.

Thorough measurements of the computational cost of deflated and non-deflated FETI algorithms were conducted in [92]. After all, it was found that 4 to 7 coarse space basis vectors are as costly as one iteration. In the following, the estimation that 4 coarse space basis vectors generate the same computational cost as one additional iteration is adopted. Consequently, the measure *total saving* $\overline{m}_s$ is defined as

$$\overline{m}_s = m_{none} - m - \frac{N_C}{4}, \tag{4.4}$$

where m and N_C are the number of iterations and the size of the coarse space, respectively, of the variant to be assessed. The number of iterations, which are necessary to solve the same linear system without a coarse space, is denoted by m_{none}. The total saving $\overline{m}_s$ quantifies the saved computational costs in terms of saved iterations while considering the additional cost of the auxiliary coarse space projection.

Figure 4.14 shows the according results for the homogeneous example as well as for the heterogeneous one. In both cases, the efficiency of all rigid body mode based coarse spaces deteriorates for smaller time step sizes because of the improved conditioning of the undeflated operator. For middle sized time steps, where the important solution spaces change from rigid body motion to effects that are dominated more by deformation, *GenEO* shows the expected advantage. It is recalled that the a priori costs to compute the GenEO modes is not included here. For small time steps, *GenEO* shows even negative efficiency and thus additional computational cost as well. The main reason for this is that *GenEO* is basically used improperly here because regardless of the operator spectra, the same number of modes, i.e., the same coarse space size, was employed. In this subsection, this was done to allow a comparison between the time dependent GenEO modes and the rigid body modes. The GenEO method with a variable coarse space is considered in section 4.3.

A particularly interesting case is the heterogeneous example at the time step size $\Delta t = T_3/100$. In the first time steps, the rigid body mode coarse spaces show a negative efficiency and even *GenEO* starts at zero. During the next 10 time steps,

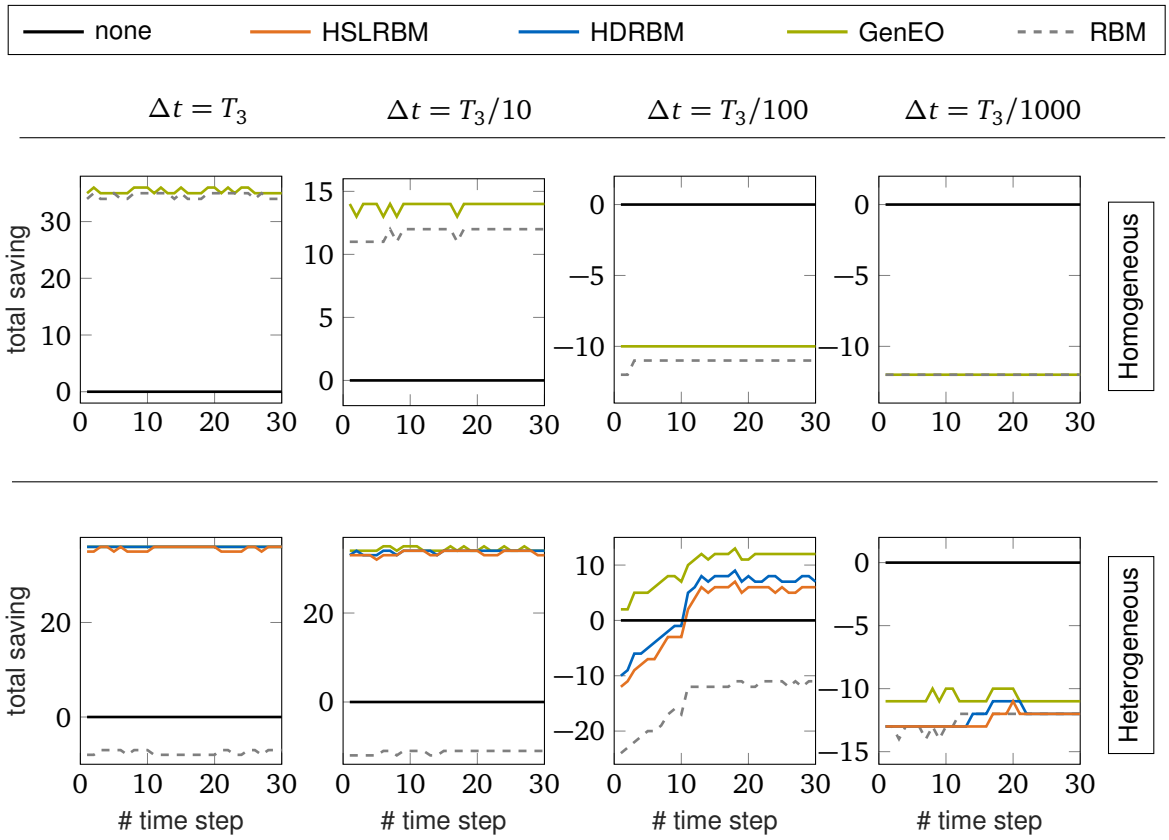


Figure 4.14. Total saving based on the saved iterations and the additional cost of the auxiliary coarse space projection. For the homogeneous case, the coarse spaces HSLRBM and HDRBM are not shown.

the efficiency improves and finally *HSLRBM* and *HDRBM* show a clearly positive efficiency, i.e., they save computational cost. This effect is assumed to originate from a lack of physical information propagation. That is to say, in the first few time steps, the acceleration caused by the loading on the upper edge has not yet reached all parts of the structure. Therefore, in these first time steps, the problem is comparatively easy to solve because no coupling and thus no coarse space is required. Later, when the whole structure is accelerated, the time step is still sufficiently large to require global coupling through a coarse space.

4.2.6 Conclusions

Within the scope of rigid body mode based coarse spaces, the following conclusions can be drawn. For homogeneous problems, the use of unpreconditioned rigid body modes, called *RBM* in the preceding investigations, is the best choice. For heterogeneous problems, only the preconditioned variants could effectively save computational cost. Because the lumped preconditioner in *HLRBM* did not show significant improvements over the superlumped one in *HSLRBM*, and because the Dirichlet preconditioner in *HDRBM* generates high additional cost before the iterations start, only

HSLRBM is recommended. Similar results were found in [45, 81], but only a single time step size around $T_3/10$ was considered.

In the preceding part of this chapter, insight was gained in the dependence of these effects on the time scale, which have been defined relative to the physical properties of the structure. With the help of GenEO modes, it could be shown that the important effects that are responsible for the high end of the operator spectrum change from global to very local effects, passing through the following three differentiable regimes.

1. At very large time steps, the most important effects are mainly comprised of rigid body motion, which is why rigid body mode based coarse spaces suit well.
2. For a range of middle sized time steps, the important effects change to overall deformations of the substructures. This still requires some coupling by a coarse space. The rigid body modes are not optimal anymore.
3. Problems at even smaller time step sizes begin to be dominated by much more localized effects. In the course of this, coarse spaces become inefficient in general because coupling becomes less important and the undeflated operators are already very well conditioned.

The rigid body mode based coarse spaces inherently exhibit another severe limitation. There is no reasonable mechanism to adapt their coarse space sizes. At the same time, the preceding results already suggest that larger coarse spaces could further improve the efficiency at large time step sizes and that for middle and small time steps sizes, smaller coarse spaces would be necessary. Any practical a priori determinable criterion to decide as a function of the time step size and the heterogeneity pattern if a rigid body mode coarse space should by applied or not is missing.

4.3 Recycling Strategies

In this section, mainly recycling algorithms are assessed. In addition, the GenEO method is studied here because of its similar capabilities regarding an adaptive choice of the coarse space size. The aim is to study methods that are able to automatically construct coarse spaces that are specifically designed for a particular problem. This includes that the method should be able to automatically select a reasonable coarse space size up to the point where using no coarse space at all is the best available alternative.

When a time stepping scheme is applied to solve a problem of linear structural dynamics, the same linear system must be solved for each time step in order to compute the according solution. Only the right hand side usually changes with each time step. The iteration matrices $\boldsymbol{D}^{(s)}$ and thus the interface operator $\boldsymbol{F}$ as well as its preconditioner $\boldsymbol{H}$ remain constant throughout all computations. Recycling strategies aim

to reuse the information gathered in earlier time steps to accelerate the solution of future ones. They all have in common that they build a coarse space based on the search directions that have been generated by the CG algorithm before. The numerous techniques that have been suggested in the literature either use the plain (i.e., unprocessed) search directions as coarse space basis vectors or try first to extract the most important information from them. In this section, both types are presented, extended, and compared concerning their capabilities to fulfill the requirements that have been set above.

4.3.1 Plain Reuse

Building the coarse space $\boldsymbol{C}$ directly from the directions $\boldsymbol{w}$, which were used to adapt the solution $\boldsymbol{\lambda}$, is the most simple and the oldest recycling method. It was introduced for dual domain decomposition in [86] and further applied and investigated in [40, 87, 91, 92]. The standard variant for linear problems was denominated total reuse of Krylov subspaces (TRKS) in [92]. In [88, 89, 93], the plain reuse technique was extended to nonlinear problems, in which the interface operator changes slightly from one iteration to the next.

A straightforward implementation of the method is to set

$$\boldsymbol{C} \leftarrow \left[\boldsymbol{C} \,\middle|\, \boldsymbol{w}_0 \,\middle|\, \boldsymbol{w}_1 \,\middle|\, \ldots \,\middle|\, \boldsymbol{w}_{m-1}\right]$$

at the end of each time step when the algorithm has converged after m iterations. Due to the $\boldsymbol{F}$-conjugacy of all search directions $\boldsymbol{w}_i$, $i = 0, \ldots, m-1$ with respect to each other and with respect to the (also $\boldsymbol{F}$-conjugate) coarse space $\boldsymbol{C}$ that was used during the iterations, this results in a diagonal coarse problem

$$\left(\boldsymbol{C}^T \boldsymbol{F} \boldsymbol{C}\right) \boldsymbol{x} = \boldsymbol{y}\,, \qquad \boldsymbol{C}^T \boldsymbol{F} \boldsymbol{C} = \mathrm{diag}\,.$$

Because of this orthogonality (i.e., $\boldsymbol{F}$-conjugacy), the coarse space projection could also be integrated into the standard orthogonalization step of the conjugate gradient algorithm. In this case, the current search direction $\boldsymbol{w}_i$ is not only orthogonalized with respect to the search directions $\boldsymbol{w}_j$, $j = 0, \ldots, i-1$ of the current time step, but also against the search directions of former time steps. Both methods were shown to be mathematically equivalent in [40].

The method is expected to deliver reasonable results, because the search directions generated by the conjugate gradient algorithm must have been important to find the solution. Furthermore, considering that the solution space is a Krylov space and how it is constructed, it should especially capture the eigenvectors with large eigenvalues. These have already been identified as the optimal basis vectors to build an efficient coarse space. However, one particular problem is already predictable at this point. If the number of necessary iterations does not drop to zero quickly, the coarse space grows larger and larger because a well-founded criterion that could decide when to stop adding search directions to the coarse space is not available.

4.3.2 Global Ritz Approximation

The similarity of the single-preconditioned CG algorithm to the Lanczos algorithm, which is used to compute eigenvalues and -vectors iteratively, led to the idea of building the coarse space from so-called Ritz vectors. The Ritz vectors approximate the eigenvectors of the preconditioned operator $\boldsymbol{HF}$ within the solution space spanned by the search directions. This strategy has been presented and analyzed in several publications [90–92]. Like the Lanczos algorithm, the efficiency of the procedure is based on two main principles.

First, a basis of the Krylov space that is orthonormal with respect to the inverse of the preconditioner $\boldsymbol{H}^{-1}$ can be built from the preconditioned residuals $\boldsymbol{z}_i$. With the columns of $\boldsymbol{V}_z$ being the basis vectors, this can be expressed as

$$\boldsymbol{I} = \boldsymbol{V}_z^T \boldsymbol{H}^{-1} \boldsymbol{V}_z\,, \qquad \boldsymbol{V}_z = \left[\dots \left| (-1)^i \frac{\boldsymbol{z}_i}{\sqrt{\boldsymbol{r}_i^T \boldsymbol{z}_i}} \right| \dots \right]$$

Furthermore, due to the short recurrence relation of the CG algorithm, the projection of the unpreconditioned operator $\boldsymbol{F}$ onto this subspace results in the tridiagonal matrix

$$\boldsymbol{H}_m = \boldsymbol{V}_z^T \boldsymbol{F} \boldsymbol{V}_z = \begin{bmatrix} \delta_0 & \eta_1 & 0 & \dots & 0 \\ \eta_1 & \delta_1 & \eta_2 & \ddots & \vdots \\ 0 & \eta_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \eta_{m-1} \\ 0 & \dots & 0 & \eta_{m-1} & \delta_{m-1} \end{bmatrix} \tag{4.5}$$

whose entries are even scalar functions of the scalar CG algorithm step lengths α_i and β_i, readily available from the preceding solution procedure. They are given by

$$\delta_0 = \frac{1}{\alpha_0}\,, \qquad \delta_i = \frac{1}{\alpha_i} - \frac{\beta_{i,i-1}}{\alpha_{i-1}}\,, \qquad \eta_i = -\frac{\sqrt{\beta_{i,i-1}}}{\alpha_{i-1}}\,, \qquad i = 1, \dots, m-1$$

It is remarked here that $\boldsymbol{H}$ denotes the preconditioner, and $\boldsymbol{H}_m$ denotes the tridiagonal matrix defined in equation (4.5). Furthermore, according to the algorithm listings like algorithm 3, m is the number of iterations after which the CG loop was terminated.

Second, the repeated application of the preconditioned operator in such Krylov space based methods is likely to capture the desired eigenvectors with extremal eigenvalues in the high part of the spectrum well. These properties allow to approximate the desired eigenvalue problem

$$\boldsymbol{HFy} = \Theta \boldsymbol{y} \qquad \text{or} \qquad \boldsymbol{Fy} = \Theta \boldsymbol{H}^{-1} \boldsymbol{y}$$

by using a Ritz ansatz, which means in general that the solution is sought in some subspace while the projection of the error on this subspace must vanish. The projected eigenvalue problem reads

$$\boldsymbol{H}_m \boldsymbol{q} = \widetilde{\Theta} \boldsymbol{q}$$

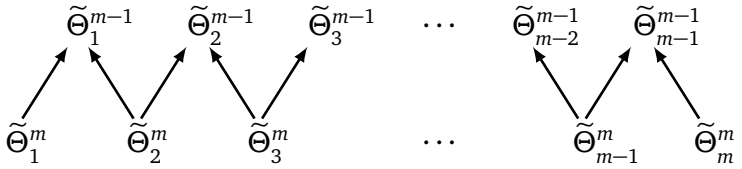


Figure 4.15. Visualization of the Ritz values after $m-1$ and after m iterations. Arrows indicate against which Ritz values of the $m-1$ set the Ritz values of the m set are tested for convergence.

in which $\widetilde{\Theta}$ are the Ritz values that approximate the eigenvalues Θ of $\boldsymbol{HF}$ and the projection of the corresponding solutions back on the full space $\boldsymbol{V}_z\boldsymbol{q}$ are the Ritz vectors. For such tridiagonal matrices, very efficient eigensolvers are available. Furthermore, the system is rather small because its size is equal to the number of iterations.

It is well known from the earlier works [99–101] on CG and Lanczos algorithms in general that the Ritz values converge rapidly to extremal eigenvalues in the high part of the spectrum when the size of the Ritz space $\boldsymbol{V}_z$ is increased and that the corresponding Ritz vectors then converge to the corresponding eigenvectors as well. The Ritz values were used in [56] to analyze and compare the convergence of the dual and the primal Schur complement method, and the Ritz vectors were identified as very efficient coarse space basis vectors in a series of publications [90, 102, 103]. In [91, 92], the method was further studied, extended, and denominated SRKS.

It is important to note that using all available Ritz vectors would be equivalent to the TRKS because only the space that is spanned by the columns of $\boldsymbol{C}$ is relevant and not the basis itself. Consequently, a profound choice should be made. The authors in [92] suggest to compute the convergence of the Ritz values, i.e., to check if the change of a Ritz value from one iteration to another is small enough to assume that it has converged to an eigenvalue of the original preconditioned operator $\boldsymbol{HF}$. However, this convergence measure is not unique because the number of Ritz values increases in every iteration by one. Let $\widetilde{\Theta}_1^m > \widetilde{\Theta}_2^m > \cdots > \widetilde{\Theta}_m^m$ be the eigenvalues of $\boldsymbol{H}_m$, i.e., the Ritz values after m iterations, and $\widetilde{\Theta}_1^{m-1} > \widetilde{\Theta}_2^{m-1} > \cdots > \widetilde{\Theta}_{m-1}^{m-1}$ the eigenvalues of $\boldsymbol{H}_{m-1}$. To check the convergence of $\widetilde{\Theta}_2^m$, the difference $\widetilde{\Theta}_2^m - \widetilde{\Theta}_2^{m-1}$ but also the difference $\widetilde{\Theta}_2^m - \widetilde{\Theta}_1^{m-1}$ must be considered. This relation is illustrated by figure 4.15. The procedure applied in [92] to compute the coarse space in the case of the SRKS algorithm is listed in algorithm 5. It is executed in each time step after the conjugate gradient algorithm has converged and extends the coarse space by the Ritz vectors that correspond to converged Ritz values. To decide if a Ritz value is converged or not, a convergence criterion is defined by ϵ_Θ. In [92], the two values $\epsilon_\Theta = 10^{-6}$ and $\epsilon_\Theta = 10^{-14}$ have been considered.

The technique of choosing converged Ritz values is mainly based on the fact that only high, well-separated eigenvalues of $\boldsymbol{HF}$ are captured precisely by the conjugate gradient algorithm and thus lead to converged Ritz values. At the same time, this also indicates that the corresponding eigenvector was important for finding the solution and consequently, that it is worth a place in the coarse space. Furthermore, it is well known that in problems of structural dynamics, the complete higher part of the spectrum is usually excited and thus important for the solution. However, the algorithm does not verify if a converged Ritz value is large enough to be able to

Algorithm 5: Construction of the coarse space for SRKS

Given: $\boldsymbol{C}$, $\boldsymbol{H}_m$, $\boldsymbol{H}_{m-1}$, ϵ_Θ

Compute the eigendecompositions of $\boldsymbol{H}_m$ and $\boldsymbol{H}_{m-1}$

$$\boldsymbol{H}_m \boldsymbol{q}_i^m = \widetilde{\Theta}_i^m \boldsymbol{q}_i^m, \qquad i = 1,\dots,m, \qquad \widetilde{\Theta}_1^m > \widetilde{\Theta}_2^m > \cdots > \widetilde{\Theta}_m^m$$

$$\boldsymbol{H}_{m-1} \boldsymbol{q}_i^{m-1} = \widetilde{\Theta}_i^{m-1} \boldsymbol{q}_i^{m-1}, \quad i = 1,\dots,m-1, \qquad \widetilde{\Theta}_1^{m-1} > \widetilde{\Theta}_2^{m-1} > \cdots > \widetilde{\Theta}_{m-1}^{m-1}$$

Compute convergence measures $\Delta\widetilde{\Theta}_{i,\mathrm{rel}}$

$$\Delta\widetilde{\Theta}_{1,\mathrm{rel}} = \left| \frac{\widetilde{\Theta}_1^m - \widetilde{\Theta}_1^{m-1}}{\widetilde{\Theta}_1^m} \right|, \quad \Delta\widetilde{\Theta}_{m,\mathrm{rel}} = \left| \frac{\widetilde{\Theta}_m^m - \widetilde{\Theta}_{m-1}^{m-1}}{\widetilde{\Theta}_m^m} \right|$$

$$\Delta\widetilde{\Theta}_{i,\mathrm{rel}} = \min\left(\left| \frac{\widetilde{\Theta}_i^m - \widetilde{\Theta}_{i-1}^{m-1}}{\widetilde{\Theta}_i^m} \right|, \left| \frac{\widetilde{\Theta}_i^m - \widetilde{\Theta}_i^{m-1}}{\widetilde{\Theta}_i^m} \right| \right)$$

Construct the coarse space

for $1 \le i \le m$ **do**

$$\boldsymbol{C} \leftarrow \left[\boldsymbol{C}, \frac{\boldsymbol{V}_z \boldsymbol{q}_i^m}{\sqrt{|\widetilde{\Theta}_i^m|}} \right] \quad \text{if} \quad \Delta\widetilde{\Theta}_{i,\mathrm{rel}} < \epsilon_\Theta$$

significantly affect convergence. Finally, it must be noted that it is not necessary to build the coarse space basis from precise approximations of high eigenvectors of $\boldsymbol{HF}$. Instead, it is sufficient and mathematically equivalent if a coarse space can represent the according set of high eigenvectors. In other words, the specific basis chosen to span the coarse space is not relevant for its efficiency in terms of reducing iterations.

These considerations lead to an alternative criterion for choosing the Ritz values whose Ritz vectors should be included into the coarse space. The procedure to construct the coarse space by the new selection criterion is listed in algorithm 6. The aim of the new criterion, which is simpler and less costly to compute than the one of algorithm 5, is to lower the effective condition number of the deflated, preconditioned interface operator below the value of ϵ_κ. The parameter ϵ_κ is thus called target condition number. As long as the conjugate gradient algorithm discovers parts of the spectrum that have eigenvalues higher than ϵ_κ, they are removed in the subsequent time step. The first question arising for the target condition number criterion is if it leads to more coarse space basis vectors for capturing the same number of high eigenvectors when compared to the classical convergence criterion. This would result in less efficient coarse spaces by including less important subspaces, i.e., lower parts of the spectrum.

Algorithm 6: Construction of the coarse space for SRKSκ

Given: $\boldsymbol{C}$, $\boldsymbol{H}_m$, ϵ_κ

Compute the eigendecomposition of $\boldsymbol{H}_m$

$$\boldsymbol{H}_m \boldsymbol{q}_i^m = \widetilde{\Theta}_i^m \boldsymbol{q}_i^m, \quad i = 1, \ldots, m, \qquad \widetilde{\Theta}_1^m > \widetilde{\Theta}_2^m > \cdots > \widetilde{\Theta}_m^m$$

Construct the coarse space

for $1 \leq i \leq m$ **do**

$$\boldsymbol{C} \leftarrow \left[\boldsymbol{C}, \frac{\boldsymbol{V}_z \boldsymbol{q}_i^m}{\sqrt{|\widetilde{\Theta}_i^m|}} \right] \quad \text{if} \quad \widetilde{\Theta}_i^m > \epsilon_\kappa$$

4.3.3 Accuracy of Ritz Approximation

In this section, the ability of the SRKS algorithm, i.e., of a global Ritz approximation, to precisely capture a set of high eigenvectors of the preconditioned interface operator $\boldsymbol{HF}$ is studied. In particular, the convergence method of algorithm 5 using ϵ_Θ and the target condition number method of algorithm 6 using ϵ_κ are compared. Also the FETI method without a coarse space and the TRKS approach are considered. In [92], the choice $\epsilon_\Theta = 10^{-14}$ lead to a saturation of the coarse space, i.e., no further converged Ritz values were found. The choice $\epsilon_\Theta = 10^{-6}$ was instead found to be too large because the coarse space kept growing steadily and did not saturate. However, this result was found for nonlinear problems in which the operator always changes slightly. This could be an explanation why the SRKS algorithm was so prone to keep finding new converged Ritz values. In the case of linear dynamics, the SRKS coarse space is expected to saturate also for tolerances $\epsilon_\Theta >> 10^{-14}$.

For the according investigation, the heterogeneous example at the time step size $\Delta t = T_3/100$ was chosen as a practically relevant but also difficult enough problem. Figure 4.16 shows the results of computing 30 time steps. The algorithms considered are all based on the standard FETI algorithm using the Dirichlet preconditioner. They are named and distinguished by the employed recycling strategy as listed in the following.

none No coarse space is employed.

TRKS The coarse space is built up from all search directions the conjugate gradient algorithm generates.

SRKS10^{-4} The coarse space is built up from all Ritz vectors corresponding to converged Ritz values using the tolerance $\epsilon_\Theta = 10^{-4}$.

SRKS10^{-14} The coarse space is built up from all Ritz vectors corresponding to converged Ritz values using the tolerance $\epsilon_\Theta = 10^{-14}$.

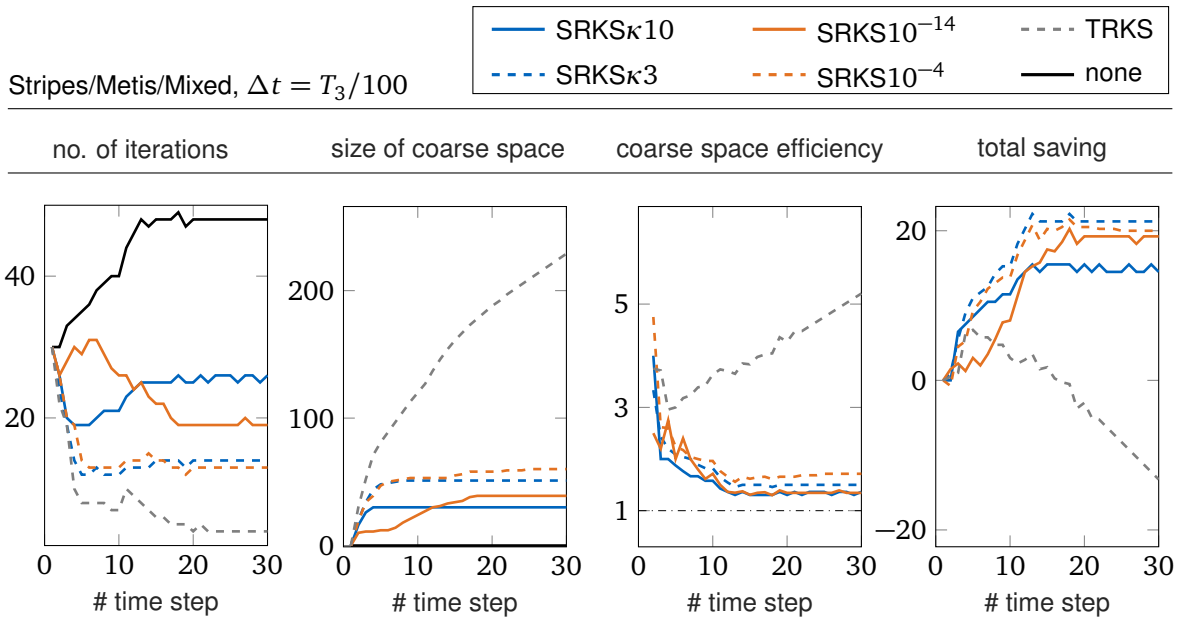


Figure 4.16. Results for the first 30 time steps of the heterogeneous example. Beginning from the left, the figure shows the number of iterations per time step and the number of coarse space basis vectors. Furthermore, it shows the coarse space efficiency as defined in equation (4.6) and the total saving as defined in equation (4.4).

SRKSκ3 The coarse space is built up from all Ritz vectors corresponding to Ritz values $\widetilde{\Theta}_i^m > \epsilon_\kappa = 3$.

SRKSκ10 The coarse space is built up from all Ritz vectors corresponding to Ritz values $\widetilde{\Theta}_i^m > \epsilon_\kappa = 10$.

A successful recycling strategy should build up the coarse space very fast and then stop to add further directions as soon as an efficient size has been reached. The TRKS coarse space keeps does not reach saturation. Instead, it keeps growing fast even though the efficiency drops. The reason is that the number of iterations does not drop to zero and thus more and more search directions are generated. All SRKS coarse spaces reach saturation. The buildup phase is also very short, only that of $SRKS10^{-14}$ is comparatively slow and delayed. This may be caused by the gradual activation of all substructures, which is also indicated by a growth phase in the number of iterations of *none*. It causes an only partial activation of the spectrum during the first time steps and as a consequence, only few Ritz values converge at very high precision. The other SRKS based variants are less sensitive to this effect.

The third plot in figure 4.16 shows the coarse space efficiency ξ_C, which is defined as

$$\xi_C = \frac{m_{none} - m}{N_C}, \tag{4.6}$$

i.e., the number of saved iterations per coarse space basis vector. In practice, the best possible coarse space efficiency that could be attained is $\xi_C = 1$, i.e., one saved

iteration per coarse space basis vector. This can be deducted from the influence of the highest eigenvalue in the spectrum. It is already known that the corresponding eigenvector is the most effective coarse space basis vector that could be chosen. For example, the linear system $\boldsymbol{F}\boldsymbol{\lambda} = \boldsymbol{d}$ is considered. To construct a worst case scenario, the highest eigenvalue of $\boldsymbol{F}$ is assumed to be arbitrarily large in comparison to its other eigenvalues. Under the assumption that the whole spectrum is excited by $\boldsymbol{d}$, the eigenvector corresponding to that highest eigenvalue arbitrarily dominates the second basis vector $\boldsymbol{Fd}$ of the Krylov space that is generated during the iterations. Consequently, the eigenvector is rapidly captured in the search space and removed from subsequent iterations by the orthogonalization step. This is mathematically equivalent to remove it right from the start by a coarse space projection.

The final performance in saving computational cost is illustrated in the rightmost plot that shows the total saving $\overline{m}_s$, which was introduced in equation (4.4). The increase of the coarse space size by setting $\epsilon_\Theta = 10^{-4}$ in comparison to the more conservative choice $\epsilon_\Theta = 10^{-14}$, which was suggested for nonlinear problems, significantly improves the performance. The newly proposed target condition number criterion with $\epsilon_\kappa = 3$ leads to very similar behavior and performance when compared to the convergence criterion with $\epsilon_\Theta = 10^{-4}$. A major difference between the two criteria is the much faster buildup of the coarse space by *SRKSκ10* or *SRKSκ3* in comparison to *SRKS*10^{-14}.

It is now investigated further, if the reason for the very similar results in performance is that the newly proposed target condition number criterion ϵ_κ is able to approximate the large global eigenvectors at the same precision as the classical ϵ_Θ criterion. For this purpose, the principal angles between the space spanned by the largest 150 eigenvectors of $\boldsymbol{HF}$ and the coarse space of the 30th time step have been computed. They are shown in figure 4.17 for the homogeneous example at $\Delta t = T_3/10$ and for the heterogeneous example at $\Delta t = T_3/100$. The two time step sizes were chosen for this study because they have been identified before as especially challenging. The high end of the effective spectra of the deflated operators is shown in the upper part, and the principal angles are shown the lower part of the figure. In the plots of the spectra, the resulting condition numbers, which are equivalent to the largest eigenvalue, are indicated by horizontal lines and given in nearby boxes. In the plots of the principal angles, the coarse space size is given in nearby boxes. Furthermore, the principal angles are plotted from right to left and have been aligned such that they are positioned just below the "removed" part of the corresponding spectrum above.

For the SRKS based variants, the results are very similar and show that a large part of the coarse space is used to precisely approximate a subspace of the largest 150 eigenvectors of $\boldsymbol{HF}$. Only very few principal angles are larger than 10^{-3}. From this it clearly follows that all SRKS variants chose a coarse space of similarly high efficiency, which agrees well with the plot of the coarse space efficiency ξ_C in figure 4.16. Furthermore, the condition number prescribed by the target condition number criterion, e.g., 10 in *SRKSκ*10, is attained quite precisely.

Even more revealing are the principal angles of the *TRKS*. It should be noted first that the coarse space sizes for *TRKS* (265 and 227) are much larger than the number of principal angles (150) shown in figure 4.17. This allows two conclusions. First, *TRKS*

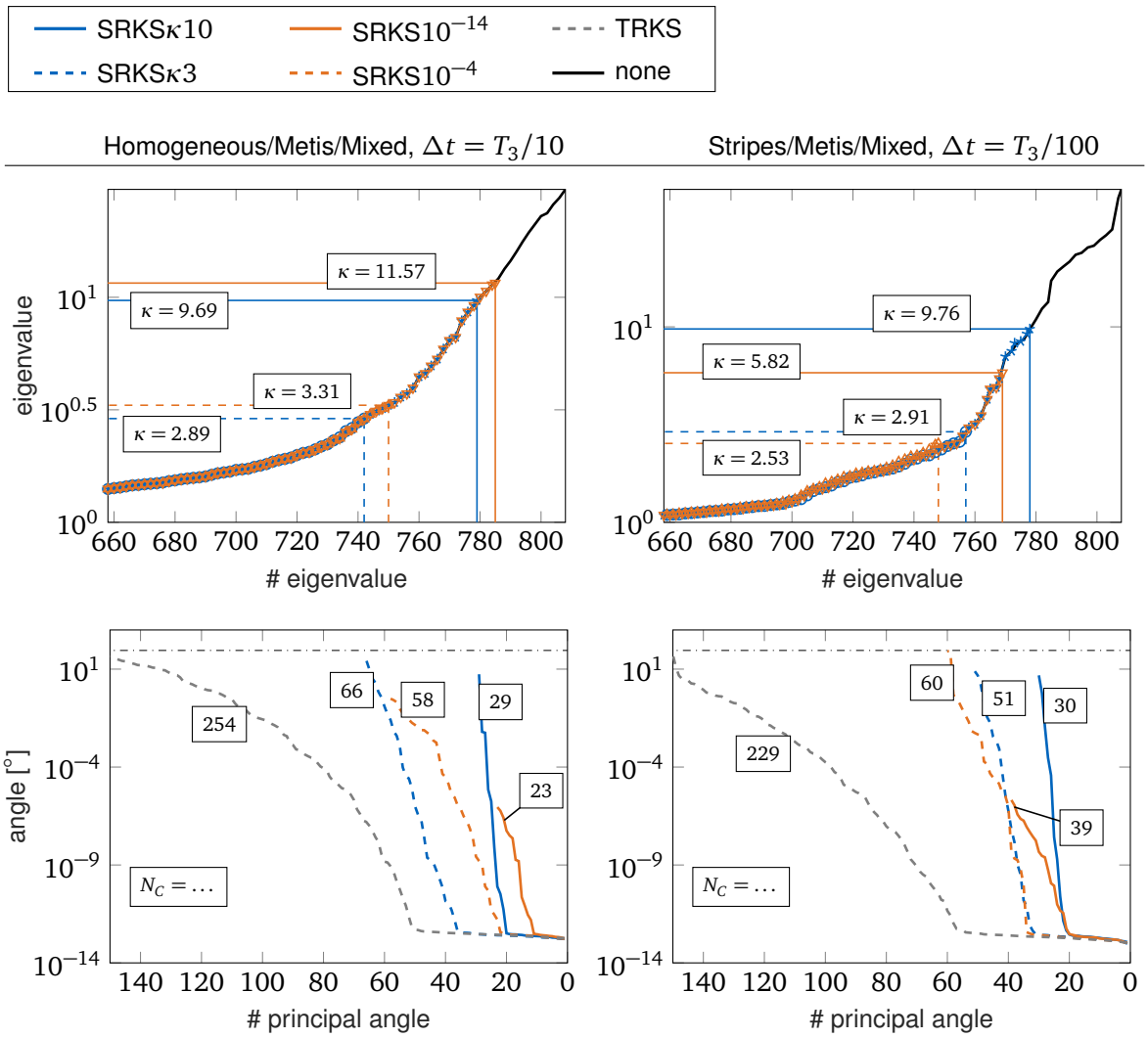


Figure 4.17. In the upper row, the high end of the eigenspectra of the deflated operators is plotted with boxes that indicate the condition number. The vertical lines indicate the number of non-zero eigenvalues. In the lower row, the principal angles between the space spanned by the largest 150 eigenvectors of $\boldsymbol{HF}$ and the coarse space are plotted from right to left to align them with the corresponding removed parts of the spectrum above.

is very inefficient because, for example, in the heterogeneous case, the large coarse space with 227 basis vectors only manages to capture about 60 directions within the space of the 150 largest eigenvectors at a high precision. Second and most important, the principal angles of *TRKS* should be recognized as an indicator of the best possible achievement regarding the precise approximation of large eigenvectors by a selective recycling algorithm. It is recalled that the *TRKS* uses the complete available solution space that has been produced over all 30 time steps computed so far. By selecting a subspace of this, regardless of the chosen basis, no recycling algorithm can produce principal angles below the grey dashed curve of *TRKS*.

This realization is especially interesting in combination with the quite sharp rise of the *TRKS*'s principal angles between #40 and #60. It means that it is impossible

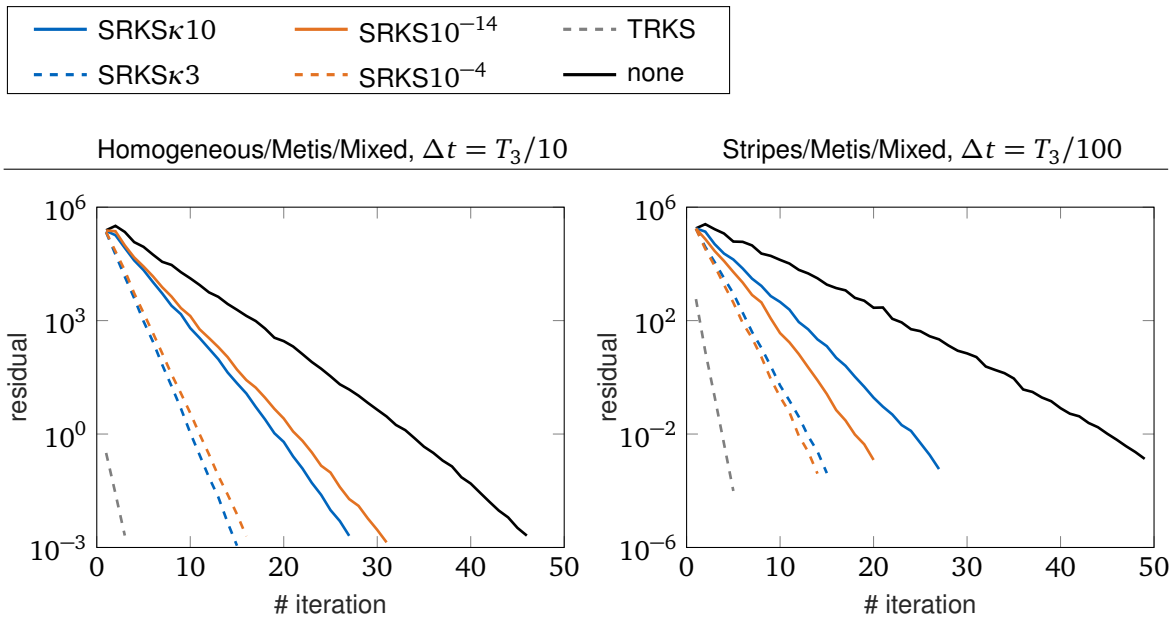


Figure 4.18. Progress of the residual $\sqrt{\boldsymbol{r}^T\boldsymbol{z}}$ during the solution of the 30th time step for the homogeneous and the heterogeneous example.

to approximate more than about 50 large eigenvectors at a very high precision by a selective recycling algorithm like the SRKS method; simply because the other part of the spectrum was not important enough for the solution. Consequently, $SRKS10^{-4}$ and $SRKS\kappa3$ can be identified to be very close to this optimal point of precisely approximating not more and not less than the most important subspace of the largest eigenvectors.

To conclude this study of comparison and parameter tuning of the SRKS variants, the course of the residuals over the iterations during the solution of the 30th time step are shown in figure 4.18. The courses for $\epsilon_\Theta = 10^{-14}$ and $\epsilon_\kappa = 10$ are still a little jagged and unsteady, which confirms that the choices $\epsilon_\Theta = 10^{-4}$ and $\epsilon_\kappa = 3$ are closer to the optimum. It should also be noted that despite of the applied coarse space, all SRKS variants start with a residual that is not significantly smaller than in the case of no coarse space. This makes clear that these coarse spaces do not work by already finding a good approximation of the solution itself within the coarse space, but simply remove those specific subspaces that hurt the convergence of the conjugate gradient algorithm.

4.3.4 Appropriate Coarse Space Sizes

This section compares the performance of all coarse space methods that are basically able to automatically generate coarse spaces that are adapted to the requirements of the underlying problem. In particular, the appropriate selection of the coarse space size is investigated. The rigid body mode based coarse spaces are simple solutions for large time steps but the coarse space size is limited and fixed. As a consequence,

these methods cannot adapt to different needs. The GenEO method is expensive in the a priori computations, but with the jump or the threshold criterion, at least an appropriate coarse space size should result. The jump criterion is not considered here because it was already shown that at very small time steps, jumps are present although the problem is already very well conditioned. Instead, the threshold criterion as explained in [58] is applied. Let κ be the effective condition number of the deflated preconditioner interface operator $\boldsymbol{P}_C\boldsymbol{H}\boldsymbol{P}_C^T\boldsymbol{F}$ and $N_{\max}$ the maximal number of neighbors of a substructure, including itself. Then, if all GenEO modes corresponding to GenEO values $\theta^{(s)} < K^{(s)}$, $s = 1,\ldots,N_s$ are selected as coarse space basis vectors, the relation

$$\kappa \leq \max\left\{1, N_{\max} \max_{1\leq s\leq N_s}\left(\frac{1}{K^{(s)}}\right)\right\} \tag{4.7}$$

holds. In other words, if the threshold for choosing the GenEO modes of substructure s is set to $K^{(s)}$, it is guaranteed that the effective condition number of $\boldsymbol{P}_C\boldsymbol{H}\boldsymbol{P}_C^T\boldsymbol{F}$ is bounded by the limit of equation (4.7). Consequently, one can prescribe a target condition number κ by setting

$$K^{(s)} = \frac{N_{\max}}{\kappa}, \quad s = 1,\ldots,N_s$$

The only methods offering an adaptive coarse space size are thus GenEO with the threshold criterion and SRKS based variants. While the SRKS based variants have been introduced already before, the following GenEO based variants are added. They are both based on a deflated FETI algorithm using the Dirichlet preconditioner.

GenEO10 The coarse space is built from GenEO modes using the threshold criterion, which is set by the target condition number $\kappa = 10$.

GenEO60 The coarse space is built from GenEO modes using the threshold criterion, which is set by the target condition number $\kappa = 60$.

The different coarse spaces have been applied to solve 30 time steps of the homogeneous and the heterogeneous example at all four time step sizes. The number of iterations, the size of the coarse space, and the total saving are measured. Figure 4.19 shows the results for the homogeneous example, and figure 4.20 shows the results for the heterogeneous example.

Throughout all cases, the GenEO based variants work well in saving computational cost and generating appropriate coarse space sizes but their performance stays consistently below the recycling algorithms. Neither the additional a priori computational cost of the GenEO eigenproblems nor the cost of computing the Ritz approximations is included here but the GenEO eigenproblems are much more expensive.

The SRKS algorithms perform well, and through all test cases, the larger coarse spaces generated by $\epsilon_\Theta = 10^{-4}$ and $\epsilon_\kappa = 3$ are more efficient than the smaller ones generated by $\epsilon_\Theta = 10^{-14}$ and $\epsilon_\kappa = 10$. In summary, the newly proposed target condition criterion ϵ_κ is preferred over the classical convergence criterion ϵ_Θ for the following two reasons:

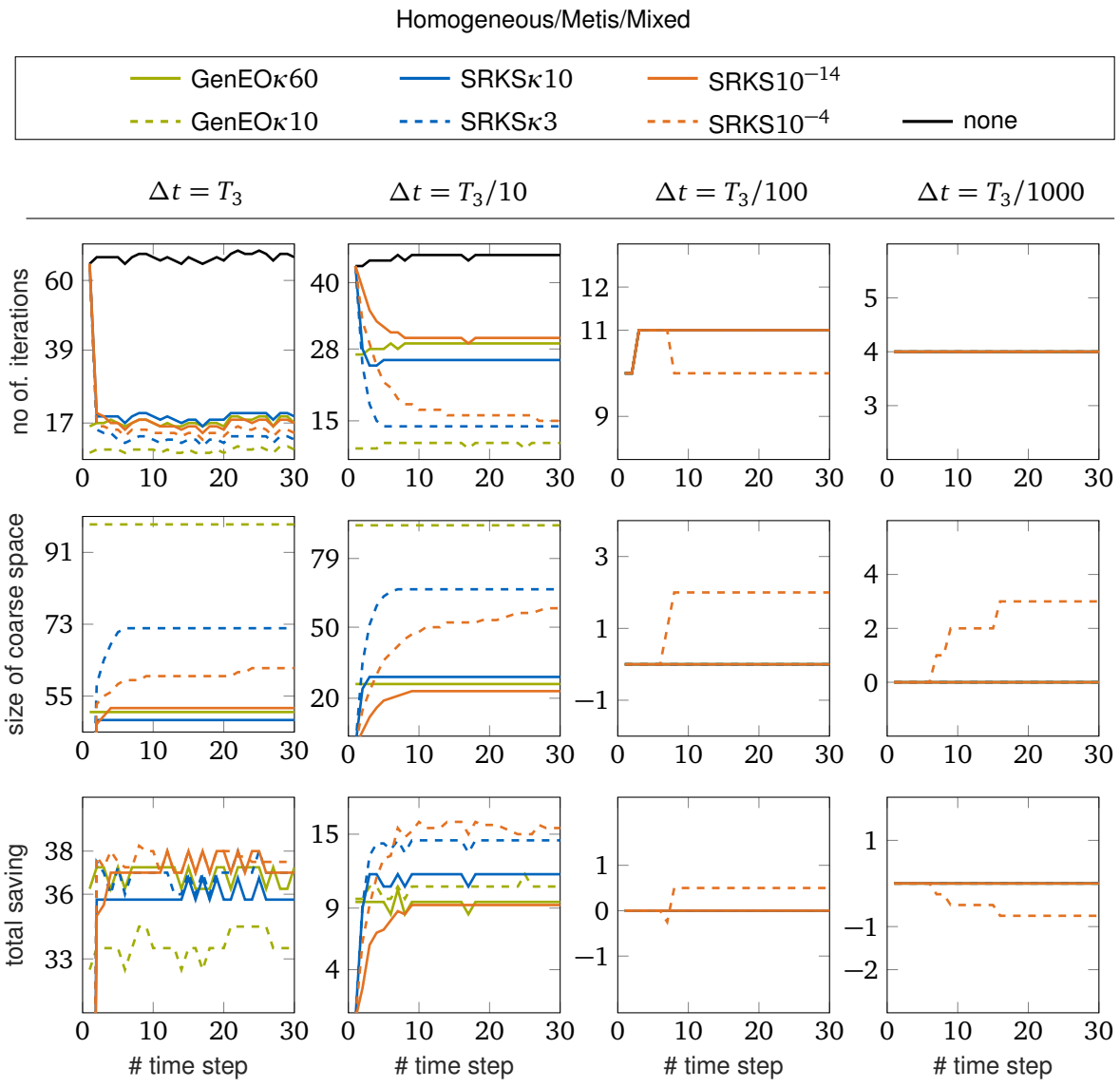


Figure 4.19. Number of iterations, size of the coarse space and total saving for 30 time steps of the homogeneous example at four different time step sizes. The GenEO method is least efficient and *SRKS*10^{-4} shows inefficiency for small time steps. *SRKS*$\kappa 3$ shows the best combination of consistently efficient coarse spaces.

1. Setting the tolerance ϵ_Θ as large as 10^{-4} is required to reach the most efficient coarse space size but leads to the problem that non-empty coarse spaces are applied to already very well-conditioned problems, e.g., at small time step sizes, even if this leads to negative efficiency. The target condition number criterion ϵ_κ does not suffer from this.

2. The target condition number criterion ϵ_κ builds up the coarse space significantly faster and does not require to solve a second eigendecomposition of H_{m-1}.

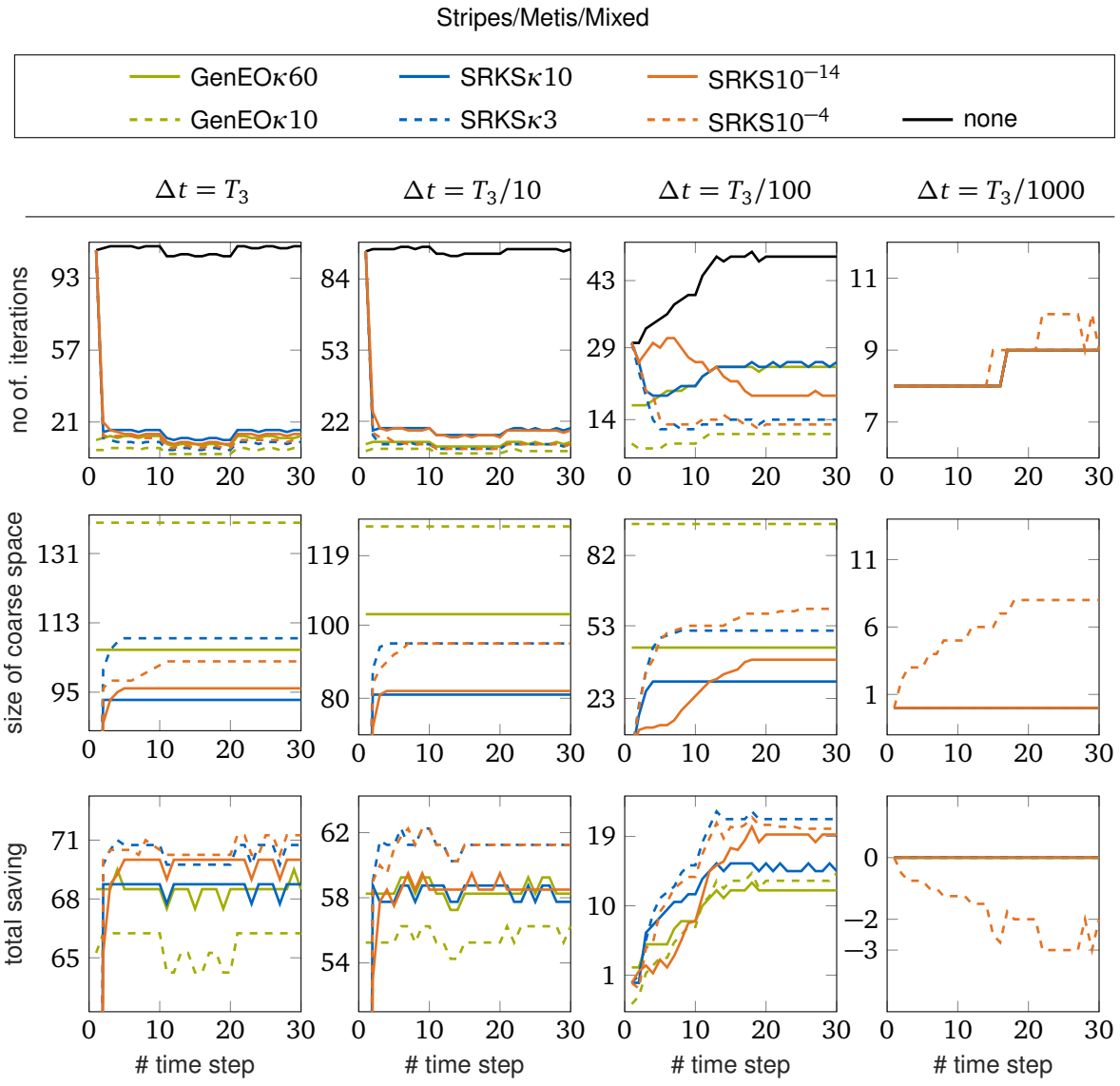


Figure 4.20. Number of iterations, size of the coarse space and total saving for 30 time steps of the heterogeneous example at four different time step sizes. The GenEO method is least efficient and *SRKS*10^{-4} shows inefficiency for small time steps. *SRKS*$\kappa 3$ shows the best combination of consistently efficient coarse spaces.

4.4 Conclusion

In this chapter, different deflation strategies, i.e., methods to generate coarse spaces, have been tested for their ability to speed up convergence and reduce computational cost. In particular, problems differing by material heterogeneity and time step size have been considered. A main aspect assessed was the capability of the methods to adapt to the specific requirements in different time scales, regarding construction and size of the coarse space.

Rigid body mode based coarse spaces have been assessed and shown to result in

efficiency gains at least for large time steps. It was found that they suffer from severe disadvantages. These are in particular their fixed coarse space size and bad efficiency for middle sized time steps. The reason was found to be a change of the subspaces which are most important for the solution and thus constitute the optimal coarse space. They change from global, rigid body mode dominated effects at large time steps to still global but deformation dominated effects at middle sized time steps until they end up at completely local effects, which cause less coupling and do not necessitate coarse spaces anymore. Although such effects have not been considered here, this is important results for questions of scalability. It indicates that for very small time steps, and then possibly for explicit integration schemes, the FETI method might numerically scale up to very high numbers of substructures without a coarse space.

The GenEO method, although it is very adaptable to changing problems and was used as reference for the analysis of rigid body mode coarse spaces, was found to be less efficient than recycling based methods.

The TRKS recycling method was found to produce inefficient coarse spaces, mainly due to its lack of a reasonable criterion to limit the coarse space size. It could be used as a temporary solution to quickly implement a recycling strategy. However, a limitation of the coarse space size must be included, e.g., reusing only the search directions generated during the solution of the first time step.

The SRKS methods showed a remarkable ability to precisely approximate the largest eigenvectors and thus to build highly efficient coarse spaces. On the one hand, the results of the TRKS illustrated that only a strictly limited subspace within the space of large eigenvectors can be extracted from the solution spaces. On the other hand, it was shown that the classically used strict accuracy criterion of $\epsilon_\Theta = 10^{-14}$ for detecting converged Ritz values is not able to capture this subspace to the full extent, while the much smaller choice $\epsilon_\Theta = 10^{-4}$ suffers from the problem of constructing unnecessary and inefficient coarse spaces in the case of well conditioned problems, e.g., at very small time step sizes.

To cure this problem, an alternative selection criterion for the SRKS method was proposed, which is based on choosing a target condition number. It allows to directly select the Ritz vectors based on their Ritz value and thus without solving a second tridiagonal eigenproblem for a convergence check. The spectra of the deflated operators as well as the principal angles between the coarse spaces and the high eigenvectors have shown that the target condition number criterion does not suffer from lower approximation accuracy. Instead, it allows to precisely attain a prescribed effective condition number and builds up the coarse space extremely fast. Furthermore, it has shown to build the most appropriately sized and most efficient coarse spaces through all relevant time scales for implicit integration schemes in structural dynamics. It is thus the recommended coarse space in the case of solving problems of linear structural dynamics with a classical deflated FETI method.

Chapter 5

Multipreconditioning

This chapter introduces adaptive multipreconditioning for problems of structural dynamics. After a theoretical introduction in the first section, the problem of degenerated search spaces is discussed, which is practically non-existent in static problems, but typically occurs in dynamic ones. These effects have been discussed earlier in the author's publications [2] and [3]. In section three, several recycling strategies that specifically address multipreconditioning are presented. These encompass a straightforward application of the plain reuse strategy to multipreconditioning, a more sophisticated procedure combining the GenEO eigenproblems with Ritz approximations, and a simplified variant of the latter method. These recycling strategies give valuable insight into the internal mechanisms of AMP-FETI and its connection to the auxiliary coarse space generated by GenEO. They have been published earlier in the author's publications [3] and [4]. The following sections are substantially based on these publications.

In the preceding chapters, all the discussed techniques to accelerate the solution process and to make it more efficient were applied to the classical form of the finite element tearing and interconnecting algorithm. Their similarity of the fundamental concept becomes clearly visible when one recalls that all these techniques, whether it may be the choice of the basic preconditioner, a scaling procedure, or deflation, can be expressed in the form of a classical CG algorithm. Despite the mentioned reasons why a distinction should be made between classical preconditioners and deflation techniques, all these enhancements of the classical FETI algorithm fit into the application of a simple preconditioning operator and thus into the standard concept of conjugate gradients as an iterative Krylov subspace method.

The first idea that outlined a concept beyond this mathematical framework was described in [35] for the special case of two substructures. It was based on the observation that when the procedure seeks for a correction of the interface forces $\boldsymbol{\lambda}$ within the preconditioning step, it asks both substructures for the necessary change in interface forces to fulfill a prescribed interface displacement. In the classical FETI algorithm, the answers of the two substructures are always averaged somehow, i.e., by the rules of a scaling procedure. This averaging is done in exactly the same way in each iteration. For this reason, the preconditioning operator is always the same. The fundamental idea was to consider both answers of the two substructures as independent search directions, i.e., to compute the optimal step lengths in both directions

simultaneously. Consequently, it was called FETI Simultaneous. This concept has been generalized to an arbitrary number of substructures in [104], and the implications of this new method called multipreconditioning are profound. The resulting procedure can be described as a CG algorithm whose preconditioner changes in each iteration. As a result, many of the mathematical relations of CG algorithm do not hold any longer. The solution space is not a classical Krylov space anymore, most orthogonality relations vanish and the typical short recurrence relation of the conjugate CG algorithm no longer exists. On the other hand, multipreconditioning, which was first described for the CG algorithm as such in [105], showed a massive improvement in performance. It was proven to solve the hardest problems for iterative domain decomposition techniques at a comparable efficiency and robustness as the GenEO coarse space.

When solving the dynamic interface problem of (3.14) by applying a preconditioned CG procedure, the residual

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

is computed at each iteration i. It is used in the preconditioning step of a standard or single-preconditioned FETI algorithm to compute the search direction

$$\boldsymbol{z}_i = \boldsymbol{H}\boldsymbol{r}_i = \sum_{s=1}^{N_s} \boldsymbol{H}^{(s)}\boldsymbol{r}_i$$

by summing up individual contributions from every substructure. See algorithm 3 to recall the notation. Multipreconditioning exploits the additive structure of the preconditioner to compute the optimal step direction not only in the space of one, but multiple search directions. These multiple search directions are the individual contributions of the substructures $\boldsymbol{z}_i^{(s)}$. They form the matrix $\boldsymbol{Z}_i$ column-wise, such that

$$\boldsymbol{z}_i^{(s)} = \boldsymbol{H}^{(s)}\boldsymbol{r}_i$$
$$\boldsymbol{Z}_i = \left[\boldsymbol{z}_i^{(1)} \,|\, \boldsymbol{z}_i^{(2)} \,|\, \ldots \,|\, \boldsymbol{z}_i^{(N_s)}\right].$$

Consequently, the orthogonalized and projected search direction $\boldsymbol{w}_i^{(s)}$ form the matrix $\boldsymbol{W}_i$ column-wise, and the formerly scalar search direction α_i becomes the vector $\boldsymbol{\alpha}_i$, which includes an individual step length for each individual search direction. The new residual is then formed by $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \boldsymbol{F}\boldsymbol{W}_i\boldsymbol{\alpha}_i$, and just like in a classical CG algorithm, the step length $\boldsymbol{\alpha}_i$ is computed by requiring the projection of the new residual $\boldsymbol{r}_{i+1}$ into the space of the search directions $\boldsymbol{w}_i^{(s)}$ to vanish. This minimization problem reads

$$\boldsymbol{W}_i^T\boldsymbol{r}_i = \boldsymbol{W}_i^T\boldsymbol{F}\boldsymbol{W}_i\boldsymbol{\alpha}_i$$

and yields for $\boldsymbol{\alpha}_i$ the computation

$$\boldsymbol{\alpha}_i = \boldsymbol{\Delta}_i^+\boldsymbol{W}_i^T\boldsymbol{r}_i \qquad \boldsymbol{\Delta}_i = \boldsymbol{W}_i^T\boldsymbol{F}\boldsymbol{W}_i\,.$$

Unfortunately, it is not guaranteed that the inverse of $\boldsymbol{\Delta}_i$ exists, which is why the pseudoinverse $\boldsymbol{\Delta}_i^+$ has to be used. This rank deficiency of $\boldsymbol{\Delta}_i$ occurs, when the individual search directions, i.e., the columns of $\boldsymbol{W}_i$, are linearly dependent. The causes and implications of this possible linear dependence of the individual search directions are investigated further in section 5.2.

5.1 Adaptive Multipreconditioning

To keep the computational costs of the minimization step low, the FETI Simultaneous method of [104] has been enhanced by an adaptive selection criterion in [106], where it was also classified as a multipreconditioned CG algorithm. The adaptive selection is based on the so-called τ-criterion. By defining the threshold τ, it selects only certain individual, local contributions $\boldsymbol{z}_i^{(s)}$ as basis vectors of the minimization space. First, the measure

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

is defined for each substructure in each iteration. The product of $\boldsymbol{W}_i$ and $\boldsymbol{\alpha}_i$ forms the adaption $\Delta\boldsymbol{\lambda}_i$ of the solution $\boldsymbol{\lambda}_i$ in iteration i. The measure $\Xi_i^{(s)}$ then compares the localized energy $\boldsymbol{\alpha}_i^T \boldsymbol{W}_i^T \boldsymbol{F}^{(s)} \boldsymbol{W}_i \boldsymbol{\alpha}_i$ of the taken step in iteration i with the localized energy $\boldsymbol{r}_{i+1}^T \boldsymbol{H}^{(s)} \boldsymbol{r}_{i+1}$ of the resulting error, both localized to substructure s. At last, all those individual contributions $\boldsymbol{z}_{i+1}^{(s)} = \boldsymbol{H}^{(s)} \boldsymbol{r}_{i+1}$ whose substructures s fulfill the criterion

$$\Xi_i^{(s)} < \tau$$

are selected as individual basis vectors of the minimization space. This selection is based on a convergence estimate that makes an educated guess in which parts of the global domain convergence can be accelerated by the extra cost of solving the minimization problem. Several examples have been analyzed in [106] and [107] concerning the optimal choice of τ. A value of 0.1 has shown to be a reasonable choice that leads to robust behavior in the majority of cases. Consequently, $\tau = 0.1$ holds throughout all investigations in this chapter.

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

$$\boldsymbol{Z}_i = \left[\sum_{s \notin J_i} \boldsymbol{z}_i^{(s)} \,\middle|\, \boldsymbol{z}_i^{(j_1)} \,\middle|\, \boldsymbol{z}_i^{(j_2)} \,\middle|\, \ldots \right] .$$

The resulting AMP-FETI algorithm, including a coarse space projection, is listed in algorithm 7.

One important aspect to consider is the application of the interface operator $\boldsymbol{F}$ to the search directions in $\boldsymbol{W}_i$ to obtain $\boldsymbol{Q}_i$. If projection or orthogonalization, which are doing the same in principal, would be applied before applying $\boldsymbol{F}$, this would mean that each substructure has to solve as many Neumann problems as there are search directions. In the extreme case of full multipreconditioning without adaptivity, each substructure would have to solve N_s Neumann problems, which obviously renders scalability impossible. Fortunately, small reformulations that were proposed in [104] and [106] allow to apply $\boldsymbol{F}$ before orthogonalization and projection, respectively.

Algorithm 7: Deflated adaptive multipreconditioned FETI

Let the columns of $\boldsymbol{C}$ span the deflation space.

$\boldsymbol{P}_C = \boldsymbol{I} - \boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{F}\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{F}$

$\hat{\boldsymbol{\lambda}}_0 = \boldsymbol{0} \qquad \boldsymbol{\lambda}_C = \boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{F}\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{d}$

$\boldsymbol{r}_0 = \boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}_C$

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

  $\boldsymbol{z}_0^s = \boldsymbol{H}^{(s)}\boldsymbol{r}_0$

$\boldsymbol{Z}_0 = \left[\boldsymbol{z_0}^{(1)} \,|\, \boldsymbol{z_0}^{(2)} \,|\, \ldots \,|\, \boldsymbol{z_0}^{(N_s)}\right]$

$i = 0$

while $r_{*,i} > \epsilon_{r,*}$ **do**

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

    $\boldsymbol{\Phi}_{i,j} = \left(\sum_{s=1}^{N_s} \boldsymbol{Q}_j^s\right)^T \boldsymbol{Z}_i \qquad \boldsymbol{\beta}_{i,j} = \boldsymbol{\Delta}_j^+ \boldsymbol{\Phi}_{i,j}$

  $\overline{\boldsymbol{Z}}_i = \left(\boldsymbol{C}^T\boldsymbol{F}\boldsymbol{C}\right)^{-1}(\boldsymbol{C}\boldsymbol{F})^T\boldsymbol{Z}_i \qquad \boldsymbol{Z}_{i,C} = \boldsymbol{C}\overline{\boldsymbol{Z}}_i \qquad \boldsymbol{W}_i = \boldsymbol{Z}_i - \boldsymbol{Z}_{i,C} - \sum_{j=0}^{i-1} \boldsymbol{W}_j\boldsymbol{\beta}_{i,j}$

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

    $\boldsymbol{Z}_{i,FC}^{(s)} = \left(\boldsymbol{F}^{(s)}\boldsymbol{C}\right)\overline{\boldsymbol{Z}}_i$

    $\boldsymbol{Q}_i^{(s)} = \boldsymbol{F}^{(s)}\boldsymbol{Z}_i - \boldsymbol{Z}_{i,FC}^{(s)} - \sum_{j=0}^{i-1} \boldsymbol{Q}_j^{(s)}\boldsymbol{\beta}_{i,j}$

  $\boldsymbol{\Delta}_i = \left(\sum_{s=1}^{N_s} \boldsymbol{Q}_i^s\right)^T \boldsymbol{W}_i \qquad \boldsymbol{\gamma}_i = \boldsymbol{W}_i^T\boldsymbol{r}_i \qquad \boldsymbol{\alpha}_i = \boldsymbol{\Delta}_i^+\boldsymbol{\gamma}_i$

  $\hat{\boldsymbol{\lambda}}_{i+1} = \hat{\boldsymbol{\lambda}}_i + \boldsymbol{W}_i\boldsymbol{\alpha}_i$

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

  $\boldsymbol{Z}_{sum} = 0 \qquad \boldsymbol{Z}_{i+1} = \emptyset$

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

    $\boldsymbol{z}_{i+1}^{(s)} = \boldsymbol{H}^{(s)}\boldsymbol{r}_{i+1}$

    $\Xi_i^{(s)} = \frac{(\boldsymbol{W}_i\boldsymbol{\alpha}_i)^T\boldsymbol{Q}_i^{(s)}\boldsymbol{\alpha}_i}{\boldsymbol{r}_{i+1}^T\boldsymbol{z}_{i+1}^{(s)}}$

    if $\Xi_i^{(s)} < \tau$ **then**

      $\boldsymbol{Z}_{i+1} \leftarrow \left[\boldsymbol{Z}_{i+1} \,|\, \boldsymbol{z}_{i+1}^{(s)}\right]$

    else

      $\boldsymbol{Z}_{sum} \leftarrow \boldsymbol{Z}_{sum} + \boldsymbol{z}_{i+1}^{(s)}$

  $\boldsymbol{Z}_{i+1} \leftarrow [\boldsymbol{Z}_{i+1} \,|\, \boldsymbol{Z}_{sum}]$

  $i \leftarrow i + 1$

$m = i$

$\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_C + \hat{\boldsymbol{\lambda}}_i$

Those reformulations are included in algorithm 7 and allow to apply $\boldsymbol{F}$ directly to the raw $\boldsymbol{Z}_i$, in which each column is local to a substructure and its neighbors. As a result, the number of Neumann problems each substructure has to solve does only depend on the number of neighbors one substructure has, but not on the global number of

substructures N_s.

5.2 Degeneration of the Search Space

In this section, causes, implications, and remedies of the search space degeneration in multipreconditioning are investigated and presented. Degeneration of the search space means that the columns of $\boldsymbol{W}_i$ are linearly dependent, i.e., $\boldsymbol{W}_i$ is rank deficient, which also implies rank deficiency of $\boldsymbol{\Delta}_i$. This can happen even if a selection of the direction of descend, e.g., by the τ-criterion, is performed. First, the general mathematical procedure to compute the step lengths in this case, a rank revealing LDLT decomposition as proposed in [104], is presented. Second, causes and implications of this effect are discussed. Finally, two completely different strategies to improve this part of multipreconditioned FETI are presented. While the first strategy simply allows to prefilter the individual contributions at low cost to relieve the more costly LDLT decomposition algorithm, the second strategy aims to restore a full, linearly independent search space, which is an important requirement for efficient recycling procedures.

5.2.1 Minimization by a Rank Revealing Matrix Factorization

The rank revealing LDLT decomposition is first and foremost a mathematical tool to guarantee a stable and robust iterative method, no matter how many linearly independent search directions can be produced by the preconditioning step. Therefore, it is always necessary and should never be replaced by the subsequently presented methods. In general, the decomposition identifies linearly dependent directions on the basis of a certain numerical tolerance. What's more, the procedure leads to an $\boldsymbol{F}$-orthogonal basis of the space spanned by the remaining directions. This simplifies the repeated solution of $\boldsymbol{\Delta}_i^+$, which is then a diagonal matrix.

A full LDLT factorization with symmetric pivoting, extended to the semidefinite case, as described in [37], computes the factorization

$$\boldsymbol{\Delta}_i = \widetilde{\boldsymbol{P}}\widetilde{\boldsymbol{L}}\boldsymbol{A}\widetilde{\boldsymbol{L}}^T\widetilde{\boldsymbol{P}}^T \tag{5.1}$$

with a square permutation matrix $\widetilde{\boldsymbol{P}}$, a diagonal matrix $\boldsymbol{A}$, and a lower triangular matrix $\widetilde{\boldsymbol{L}}$. More details can be found in [108]. For a given $\boldsymbol{\Delta}_i$ of dimension $n \times n$ with rank r the results have the dimensions

$$\widetilde{\boldsymbol{P}}:\ n \times n \qquad \widetilde{\boldsymbol{L}}:\ n \times r \qquad \boldsymbol{A}:\ r \times r$$

and can be written as

$$\widetilde{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{P} & * \end{bmatrix} \quad \text{with} \quad \boldsymbol{P}:\ n \times r \qquad \widetilde{\boldsymbol{L}} = \begin{bmatrix} \boldsymbol{L} \\ * \end{bmatrix} \quad \text{with} \quad \boldsymbol{L}:\ r \times r$$

The rank is determined to be r, when the $(r+1)$-th pivot element is smaller than ϵ_Δ multiplied by the largest pivot element. As a result of the pivoting strategy, the permutation matrix $\widetilde{\boldsymbol{P}}$ puts the linearly dependent directions at last. Ignoring the last $(n-r)$ columns of $\widetilde{\boldsymbol{P}}$ and the last $(n-r)$ rows of $\widetilde{\boldsymbol{L}}$ leads to the factorization of the symmetric positive matrix $\boldsymbol{P}^T\boldsymbol{W}^T\boldsymbol{F}\boldsymbol{W}\boldsymbol{P}$

$$\boldsymbol{P}^T\boldsymbol{W}^T\boldsymbol{F}\boldsymbol{W}\boldsymbol{P} = \boldsymbol{L}\boldsymbol{A}\boldsymbol{L}^T \qquad \text{or} \qquad \boldsymbol{L}^{-1}\boldsymbol{P}^T\boldsymbol{W}^T\boldsymbol{F}\boldsymbol{W}\boldsymbol{P}\boldsymbol{L}^{-T} = \boldsymbol{A}$$

In fact, the algorithm can be interrupted as soon as a pivot element that undercuts the tolerance, set by ϵ_Δ, has been found. A reasonable choice for ϵ_Δ is the machine precision, which is $2.3 \cdot 10^{-16}$ on the machine that has been used to generate the results presented in this chapter.

Algorithm 7 then proceeds with the substitutions

$$\boldsymbol{W}_i \leftarrow \boldsymbol{W}_i\boldsymbol{P}\boldsymbol{L}^{-T} \qquad \boldsymbol{Q}_i \leftarrow \boldsymbol{Q}_i\boldsymbol{P}\boldsymbol{L}^{-T} \qquad \boldsymbol{\Delta}_i^+ = \boldsymbol{A}^{-1} \qquad \boldsymbol{\alpha}_i = \boldsymbol{A}^{-1}(\boldsymbol{P}\boldsymbol{L}^{-T})^T\boldsymbol{\gamma}_i\,, \tag{5.2}$$

which leads to $\boldsymbol{F}$-conjugate columns of $\boldsymbol{W}_i$ without a linearly dependent part. An important aspect for the scalability of multipreconditioned FETI and its performance similar to coarse space methods is that the construction of $\boldsymbol{\Delta}_i$ implies global communication. On the one hand this makes the minimization step costly, but, on the other hand, allows for scalability and robustness.

5.2.2 Zero Contributions due to Inactive Parts

While linear dependency of the individual search directions is very unlikely in statics or fully excited structures in dynamics, it happens usually in the first few time steps of a dynamic problem where the initial configuration of the structure is a state of equilibrium. Unfortunately, most dynamic problems start in an initial state of equilibrium. In this case, it takes some time steps before acceleration and deformation have spread throughout the domain, such that all substructures are fully activated. This effect is particularly strong in soft structures with localized loads and small time steps. During the first time steps, when some substructures are still in equilibrium, local parts of the residual and thus certain directions $\boldsymbol{z}^{(s)}$ can become exactly or numerically zero. Furthermore, it was demonstrated in [2] that the τ-criterion has the tendency to select these zero-directions.

To investigate this effect, the example problem shown in figure 5.1 is considered. The mesh, the decomposition, and the material distribution matches the example *Stripes/Metis* as described in appendix A. The corresponding physical properties can be found in appendix A in table A.1. Here, a shock load is applied on a single element in the first time step, which has a size of $T_3/100$.

In this example, very localized effects dominate and thus lead to individual, local search directions $\boldsymbol{z}_i^{(s)}$ whose norms differ in value by several orders of magnitude.

The plot on the left of figure 5.2 shows the number of directions that have been removed by the rank revealing matrix factorization for different tolerances ϵ_Δ. The

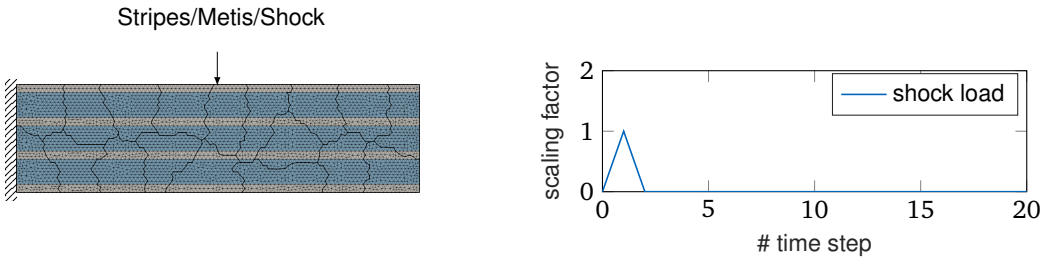


Figure 5.1. Structural problem setup. The heterogeneous structure is composed of a soft matrix (blue) between small fibers (gray). The nodes on the left egde are fixed by Dirichlet conditions. An element in the middle of the upper edge is subjected to a shock load.

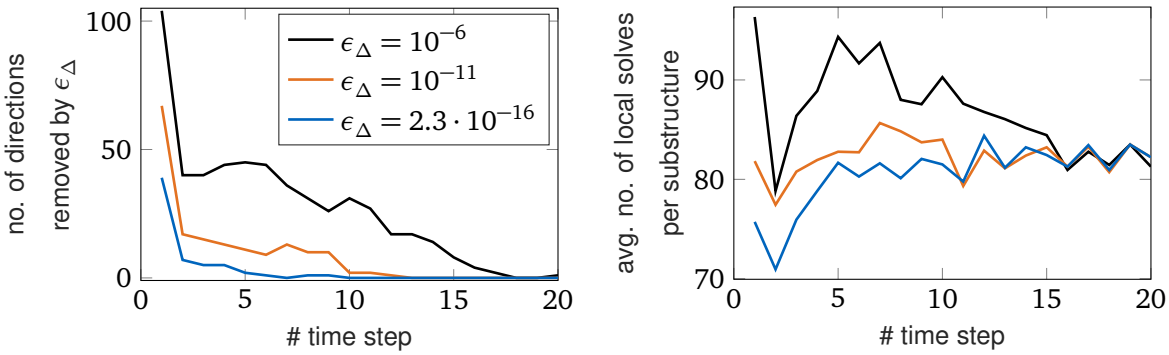


Figure 5.2. Number of directions removed by the rank revealing matrix factorization and the according average number of local solves per substructure at different values for the tolerance ϵ_Δ.

plot on the right of figure 5.2 shows the corresponding average number of local solves per substructure. Any Neumann or Dirichlet problem that a substructure has to solve is considered as one local solve. Different from classical single-preconditioned FETI algorithms, iterations are not a well-suited indicator anymore to measure the computational cost of the algorithm because the cost of each iteration strongly depends on the number of search directions and on further aspects like the number of neighbors each substructure has. The number of local solves is instead a much more meaningful measure of the computational cost.

First of all, figure 5.2 shows the intuitive result that for larger tolerances ϵ_Δ, more directions are dropped because they are identified to be linearly dependent. Furthermore, it also shows that the efficiency of AMP-FETI strongly depends on the tolerance. For larger tolerances, the number of local solves rises. Consequently, the tolerance should be set as low as possible, i.e., to machine precision.

It is recalled that the removal of directions by the ϵ_Δ tolerance implies that they have been selected by the τ-criterion in the first place. To investigate this further, some energy expressions are introduced. The contribution of the substructure s to the energy of the final step direction $\boldsymbol{W}_i\boldsymbol{\alpha}_i$ is denoted by $E_p^{(s)}$, and the contribution of the substructure s to the energy of the error is denoted by $E_r^{(s)}$. They allow to write the

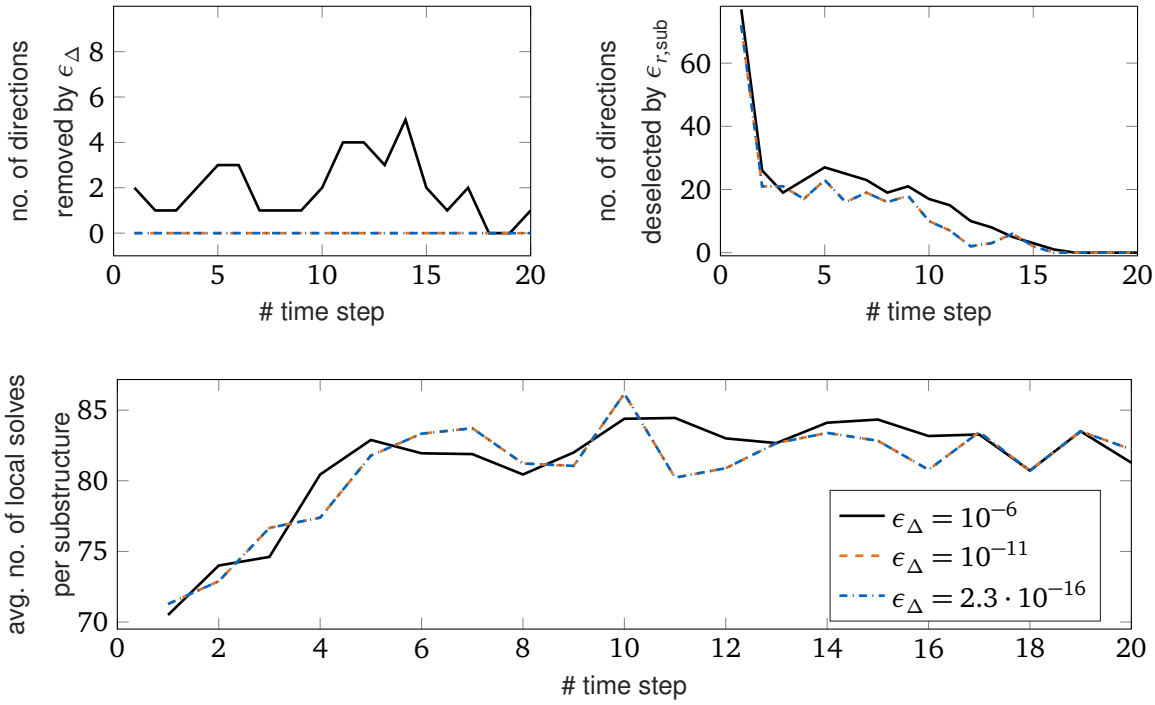


Figure 5.3. Results in case of applying the relative localized error criterion with $\epsilon_{r,\text{sub}} = 1 \cdot 10^{-6}$, which deselects certain directions prior to the rank revealing matrix factorization. In the upper part: Number of directions removed by the rank revealing matrix factorization and the number of directions removed beforehand by the relative localized error criterion. In the lower part: The according average number of local solves per substructure.

measure $\Xi^{(s)}$ of the τ-criterion in terms of localized energies. Recalling the definition of the measure $\Xi^{(s)}$, it reads for substructure s in iteration i

$$\Xi_i^{(s)} = \frac{E_{p,i}^{(s)}}{E_{r,i+1}^{(s)}} = \frac{\boldsymbol{\alpha}_i^T \boldsymbol{W}_i^T \boldsymbol{F}^{(s)} \boldsymbol{W}_i \boldsymbol{\alpha}_i}{\boldsymbol{r}_{i+1}^T \boldsymbol{H}^{(s)} \boldsymbol{r}_{i+1}}$$

As indicated before, the LDLT factorization is limited by machine precision. It is numerically impossible to simultaneously minimize the error in two directions with extremely different magnitudes. In addition, such a small search direction is an indicator that the error on which it is based is already very small. The magnitude of the residual, measured in the norm of the preconditioner, is usually used as measure for the termination criterion of the CG algorithm. Its square can be expressed as

$$\boldsymbol{r}_{i+1}^T \boldsymbol{H} \boldsymbol{r}_{i+1} = \sum_s \boldsymbol{r}_{i+1}^T \boldsymbol{H}^{(s)} \boldsymbol{r}_{i+1} = \sum_s E_{r,i+1}^{(s)}$$

and is thus a sum of local contributions of the individual substructures. If the contribution $E_r^{(s)}$ of a substructure s to the global error energy is negligibly small, it is reasonable to assume that this substructure's search direction is not important. Furthermore, this indicates that the corresponding local search direction is of extremely small magnitude and therefore numerically linearly dependent.

Based on these considerations, a new selection criterion for the individual contributions in multipreconditioned FETI, based on a new measure $e_r^{(s)}$, was introduced in

[2]. It is used in addition to the τ-criterion and aims to prevent linear dependency in most cases before the rank revealing LDLT matrix factorization is applied. The new measure $e_r^{(s)}$ can be understood as a relative localized error energy and is defined by

$$e_{r,i+1}^{(s)} = \frac{E_{r,i+1}^{(s)}}{\sum_s E_{r,i+1}^{(s)}}$$

The criterion selects only those directions of a substructure s that fulfill the criterion

$$e_{r,i+1}^{(s)} > \epsilon_{r,\text{sub}} \tag{5.3}$$

as significant for the global error. Directions not fulfilling this $\epsilon_{r,\text{sub}}$-criterion are summed up as if they had not been selected by the τ-criterion, that is to say, they are deselected.

This suggests two advantages compared to just letting the rank revealing LDLT factorization remove them. On the one hand, directions that are considered linearly dependent by the factorization are removed completely from the minimization space, while the $\epsilon_{r,\text{sub}}$-criterion keeps them within the summed direction. On the other hand, the cost of the factorization is reduced by decreasing the number of put in directions and thereby the size of $\mathbf{\Delta}$. Considering the communication cost of the rank revealing LDLT matrix factorization, which is not required by the $\epsilon_{r,\text{sub}}$-criterion, this results in improved performance.

The results of applying the criterion with $\epsilon_{r,\text{sub}} = 10^{-6}$ in combination with the tolerances ϵ_Δ as considered before are shown in figure 5.3. The effectiveness of the new $\epsilon_{r,\text{sub}}$-criterion can be observed in the upper-left plot, showing the number of directions removed by the rank revealing factorization. Except for the extremely slack tolerance of $\epsilon_\Delta = 10^{-6}$, no directions are removed anymore by the rank revealing LDLT factorization. The upper-right of figure 5.3 shows that the directions have been deselected by the $\epsilon_{r,\text{sub}}$-criterion prior to the factorization. As a result, the factorization works more efficiently because it only has to consider directions that are more likely to be actually important for the solution process.

The second observation is that for all tolerances ϵ_Δ, the same performance in terms of local solves, shown in the lower part of figure 5.3, is attained. The performance is furthermore comparable to that of the smallest tolerance ϵ_Δ in figure 5.2 without the new deselection criterion. This means that the whole algorithm has become much more robust against different tolerances ϵ_Δ, such that optimal performance is always reached while simultaneously lowering the cost of the rank revealing factorization.

5.2.3 Activation Strategies

In the preceding section, the mechanisms leading to degenerated search spaces have been explained and investigated. Furthermore, a criterion was presented that allows to prevent that degeneration of the search space at very low cost before the rank revealing factorization of $\mathbf{\Delta}_i$. This was done by moving directions of substructures

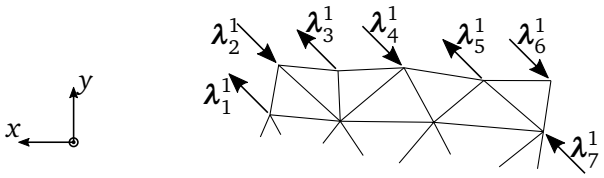


Figure 5.4. Initialization of $\boldsymbol{\lambda}$

with a very low relative localized error energy into the sum direction, where all other directions that have not been selected by the τ-criterion were already accumulated.

With regard to recycling algorithms, this is not an ideal scenario. Recycling algorithms rely on a fully populated search space that contains information about all parts of the structure. If the residual is almost zero on the interface of a specific substructure, no relevant information about this substructure can be extracted from the solution process. Consequently, if the coarse space generated from such a recycling process should later be used to solve a time step in which this substructure is active, the performance of this coarse space is very low. If a recycling procedure should be applied efficiently from the first time step on, it is necessary to ensure a rich solution space throughout the whole structural domain.

One solution would be to activate the recycling strategy at a later time step when the structure is fully excited. However, this is not maximally efficient and it is unclear how to decide in which time step the recycling strategy should be activated. A different, much more reliable procedure is an artificial activation technique, which was presented and applied in [3], is described in the following. The reason why the search space on a substructure is not populated is that no problem had to be solved, because the initial guess $\boldsymbol{\lambda}_0 = \mathbf{0}$ for the solution, i.e., the interface forces, is already

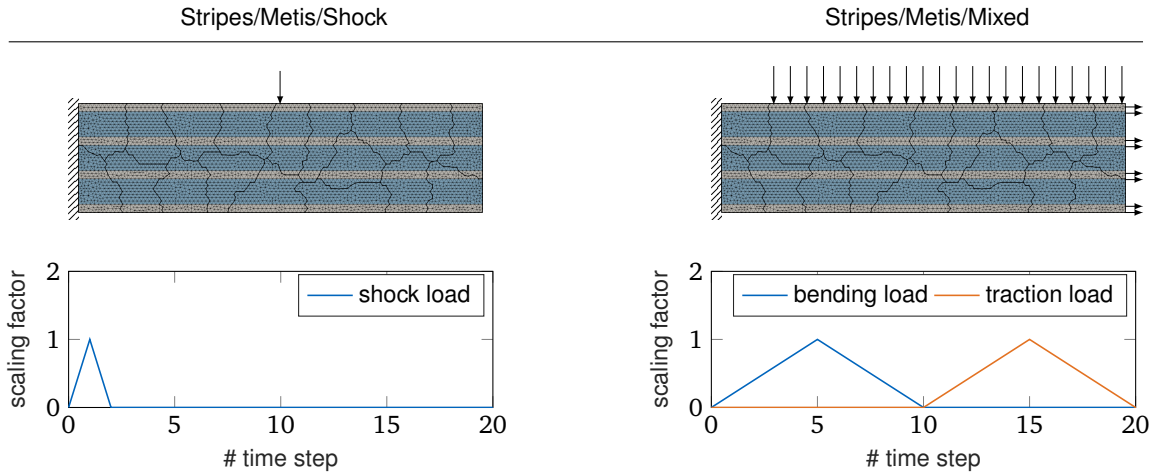


Figure 5.5. Structural problem setup. The heterogeneous structure is composed of a soft matrix (blue) between small fibers (gray). The nodes on the left egde are fixed by Dirichlet conditions. In the left example, an element in the middle of the upper edge is subjected to a shock load. In the right example, a bending load and a traction load are scaled over time.

correct. To artificially cause a full solution process that also involves an already equilibrated substructure, a simple way is to start with a random distribution of interface forces as initial guess. The solution algorithm has to find the real solution, which is zero, by a comprehensive iterative process. During this process, the important modes of the substructure are discovered, which leads to a fully populated search space. What's more, this procedure prevents any limitation of the search space to the modes activated by a very restricted excitation. A random activation procedure was applied already in [104] for the same reason. To achieve maximum comparability and reproducibility, a deterministic activation procedure is presented and applied in the following.

Instead of using a zero vector as start value for $\boldsymbol{\lambda}$, the vector $\boldsymbol{\lambda}_0$ is constructed by putting loads with alternating directions on the interfaces so that also high frequencies are excited. For example, if $\boldsymbol{\lambda}_0$ contains the dof of the interface nodes in sequential order, beginning with substructure one, it is set to

$$\boldsymbol{\lambda}_0 = \boldsymbol{\lambda}_{\text{activate}} = \begin{bmatrix} \lambda^1_{1,x} \\ \lambda^1_{1,y} \\ \lambda^1_{2,x} \\ \lambda^1_{2,y} \\ \lambda^1_{3,x} \\ \vdots \end{bmatrix} = \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \\ +1 \\ \vdots \end{bmatrix} . \tag{5.4}$$

In the case of a consecutive numbering of the interface nodes, this leads to an interface loading of substructure one as shown in figure 5.4. What has been identified as an important aspect, is the relative magnitude η of the initial excitation. It is defined by the relation

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

If the relative magnitude is chosen too high or two low, the performance of the method drops. A specific relative magnitude can be achieved by the following procedure. First $\boldsymbol{\lambda}_0$ with the desired structure of equation (5.4) and arbitrary relative magnitude is built. After computation of $\boldsymbol{F}\boldsymbol{\lambda}_0$, the result as well as $\boldsymbol{\lambda}_0$ are scaled such that η results to the desired value.

To investigate the effect of the proposed activation procedure and the influence of different relative magnitudes η, the two example problems shown in figure 5.5 are considered. In both cases, the mesh, the decomposition, and the material distribution match the example *Stripes/Metis* as described in appendix A. The according physical properties can be found in appendix A in table A.1. The time step size is set to $T_3/100$ for both examples. Accordingly, the shock load example is identical to the example of figure 5.1, which was considered before to analyze the $\epsilon_{r,\text{sub}}$-criterion.

The results are shown in figure 5.6 for both examples and for two different choices of the tolerance ϵ_Δ. The rougher tolerance of $\epsilon_\Delta = 1 \cdot 10^{-8}$ is considered in addition

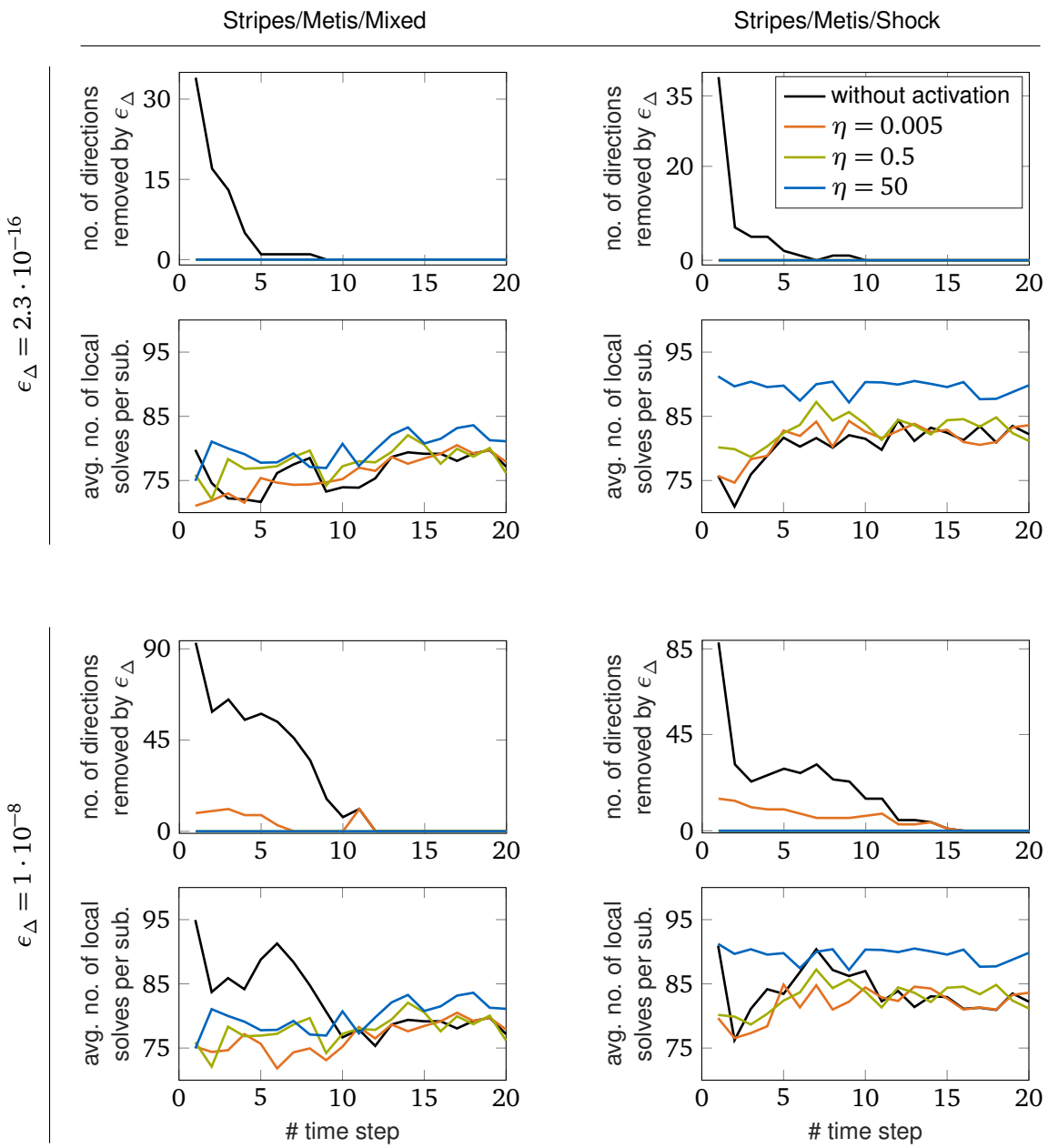


Figure 5.6. Results for for applying the activation strategy with different values for η.

to the standard choice, which is the machine precision, to get an impression of how close the search space is to being degenerated.

For $\epsilon_\Delta = 2.3 \cdot 10^{-16}$, all choices of η manage to produce a fully populated search space without any removed directions because of numerical linear dependency. For $\eta = 50$ a significant rise in local solves is caused in the shock example. A very small choice like $\eta = 0.005$ on the other hand, is not able to prevent search space degeneration at the more rough tolerance $\epsilon_\Delta = 1 \cdot 10^{-8}$. As a rule of thumb, a relative magnitude between 0.5 and 0.01 makes sure that the artificial excitation on the interface does not disturb the solution process of the actual external load by dominating it. At the

Algorithm 8: Modified initialization for AMP-FETI

if $\boldsymbol{C} = \emptyset$ **then**

$\boldsymbol{P}_C = \boldsymbol{I} \qquad \boldsymbol{\lambda}_0 = \boldsymbol{\lambda}_{\text{activate}}$ (see Equation (5.4))

$\boldsymbol{F}_0 = \boldsymbol{F}\boldsymbol{\lambda}_0 \qquad \eta = \|\boldsymbol{F}\boldsymbol{\lambda}_0\| \,/\, \|\boldsymbol{d}\| \qquad \eta_{\text{set}} = 0.05$

$\boldsymbol{\lambda}_0 \leftarrow \boldsymbol{\lambda}_0 \eta_{\text{set}}/\eta \qquad \boldsymbol{F}_0 \leftarrow \boldsymbol{F}_0 \eta_{\text{set}}/\eta$

$\boldsymbol{r}_0 = \boldsymbol{d} - \boldsymbol{F}_0$

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

$\boldsymbol{z}_0^{(s)} = \boldsymbol{H}^{(s)} \boldsymbol{r}_0$

$\boldsymbol{Z}_0 = \left[\boldsymbol{z}_0^{(1)} \,|\, \boldsymbol{z}_0^{(2)} \,|\, \ldots \,|\, \boldsymbol{z}_0^{(N_s)} \right]$

else

$\boldsymbol{P}_C = \boldsymbol{I} - \boldsymbol{C}(\boldsymbol{C}^T \boldsymbol{F} \boldsymbol{C})^{-1} \boldsymbol{C}^T \boldsymbol{F}$

$\boldsymbol{\lambda}_0 = \boldsymbol{C}(\boldsymbol{C}^T \boldsymbol{F} \boldsymbol{C})^{-1} \boldsymbol{C}^T \boldsymbol{d}$

$\boldsymbol{r}_0 = \boldsymbol{d} - \boldsymbol{F}\boldsymbol{\lambda}_0$

$\boldsymbol{Z}_0 = \sum_{s=0}^{N_s} \boldsymbol{H}^{(s)} \boldsymbol{r}_0$

same time it makes sure that it is not so much smaller than the actual external load that it becomes less important for the solution process. Of course, this is coupled to the numerical precision in general. The more accurate the interface problem is solved, the lower the choice for η can be.

This activation strategy is used in the remaining chapter to support all recycling algorithms that are presented and investigated in the following. Because all recycling strategies only recycle the solution space of the first time step, two different initialization scenarios are distinguished. In the first time step, when the coarse space is empty and a recycling strategy is be applied, $\boldsymbol{\lambda}_0$ is initialized as explained in this section to activate the full structure. When a coarse space is present in later time steps, no artificial initialization of $\boldsymbol{\lambda}_0$ is added anymore. Furthermore, because the application of the τ-criterion is not yet possible in the initialization phase, algorithm 7 suggests to consider all substructure contributions as individual search directions. In the following, this is only done if no coarse space is present. If $\boldsymbol{C}$ is not empty, the individual contributions are summed up just like in a classical conjugate gradient procedure because the coarse space is assumed to be efficient, so that a full multipreconditioning is not necessary. The resulting initialization phase, which replaces the initialization before the iterative loop in algorithm 7, is listed in algorithm 8.

5.3 Recycling Strategies

Recycling former solution spaces to accelerate the solution of future solves in iterative dual domain decomposition methods was addressed in numerous publications. They are all based on deflation, i.e., they construct a coarse space either directly from

the former search spaces or by a more complex recycling procedure, aiming to extract only the most important subspace. Some do not explicitly use a deflation projector, but incorporate the procedure into the orthogonalization step. This is always possible and only requires a basis of the coarse space that is orthogonal with respect to the interface operator $\boldsymbol{F}$. However, all these methods were designed for classical conjugate gradient procedures, i.e., single-preconditioning and Krylov subspaces. Such recycling strategies for single-preconditioning were discussed in chapter 4. The first recycling strategies that are applicable to multipreconditioning have been proposed by the author in [3] and [4].

5.3.1 Plain Reuse

The simplest approach is to reuse the plain, that is to say, unprocessed, solution space that has been build during the CG iterations. It can include a selection criterion to not reuse the complete former solution spaces, but only a part of them. In [86–89, 91, 92] as well as in section 4.3.1, this approach was applied to the classical single-preconditioning algorithm, which generates Krylov solution spaces by a repeated application of the preconditioned interface operator. It was shown in section 4.3.1, that this is an inefficient procedure for single-preconditioning and that at least a criterion to limit the coarse space size is necessary.

In this subsection the strategy of plain reuse is applied to multipreconditioning. The mechanisms of AMP-FETI suggest that plain reuse strategy could be much more efficient here because the individual contributions of the individual substructures are available. They are more likely to be able to capture the relevant local effects. As it was shown in section 4.3.1, a limitation of the coarse space size is necessary and is be applied here. While it is quite obvious that all individual contributions in $\boldsymbol{W}$ that were generated within one time step can represent the relevant local effects, the important question to be addressed is how much computational cost can be saved in relation to the (limited) size of the coarse space.

The simple method is implemented by setting

$$\boldsymbol{C} \leftarrow \left[\boldsymbol{C} \,\middle|\, \boldsymbol{W}_0 \,\middle|\, \boldsymbol{W}_1 \,\middle|\, \ldots \,\middle|\, \boldsymbol{W}_{m-1}\right]$$

at the end of a time step when the conjugate gradient algorithm has converged after m iterations. Because the method has the potential to produce already in the first time step much more directions than an efficient coarse space size would allow, a target coarse space size is set, and the matrix $\boldsymbol{C}$ is chopped such that only the first, i.e., left, columns remain. This automatically implies a reasonable selection, because the kept directions correspond to earlier iterations and are thus more likely to capture the well-separated extremal eigenvalues in the high part of the spectrum. Such advantageous sorting even applies to the directions within one iteration, because they are sorted within $\boldsymbol{W}_i$ in the order of the size of the pivot elements in the rank revealing LDLT factorization. This order reflects their importance for the solution, and thus also their relevance to discover the detrimental local effects.

5.3.2 Local Ritz Approximation

While the plain reuse strategy presented above is easy to implement and does not require any extra cost to build the coarse space, a very large number of directions is generated and the efficiency of this coarse space, i.e., the computational cost saved in relation to the coarse space size, might be comparably low. For improved efficiency, a method to extract the essence out of the available spaces is necessary, just like the Ritz approximation of global eigenvectors of the preconditioned interface operator, which was considered in section 4.3.2. Unfortunately, this strategy is not applicable to multipreconditioning, because the important orthogonality relations as well as the short recurrence of the classical conjugate gradient, on which this method is based, are no longer valid. This happens, because the effective preconditioner changes from one iteration to the next, even without the adaptivity by the τ-criterion. At the same time, AMP-FETI is known for precisely considering the detrimental local effects of substructure behavior because it does not blur the individual contributions by simply adding them to a single search direction. Consequently, the search spaces generated by AMP-FETI should be well suited to extract highly efficient coarse spaces.

This section presents a method, which was first presented in the author's publication [3], that applies such an extraction method to AMP-FETI. The basic idea is to solve the GenEO eigenproblems in a strongly reduced form, such that they require a minimum of computational cost to be assembled and solved. To construct the reduced eigenproblems, the original GenEO eigenproblems are projected onto a specific part of the search space.

Although the GenEO eigenproblems can be easily parallelized as proposed in [109], they involve the Schur complements of the substructure itself and its neighbors and result in a dense generalized eigenproblem of the size of the substructures interface. The resulting coarse space was shown to result in fast convergence for the hardest problems for dual domain decomposition algorithms, including material heterogeneity, jagged interfaces and bad aspect ratios. However, it would be favorable to reduce the computational cost by making use of the information that was gathered during the solution of the first time step.

Construction of a Local Ritz Space for GenEO

As remarked earlier, the main strength of the multipreconditioned FETI method lies in its ability to compute the optimal combination $\boldsymbol{W}_i\boldsymbol{\alpha}_i$ of the substructure contributions, instead of blurring the local modes by simply summing them up. These solution increments $\boldsymbol{W}_i\boldsymbol{\alpha}_i$ are thus the point of departure to build a Ritz space that is able to represent the local bad modes. The basis vectors of this Ritz space are then applied, analogously to the global approximation in section 4.3.2, as test and trial functions for approximating the corresponding local GenEO eigenproblems. As a result, a much smaller eigenproblem has to be solved, which is likely to deliver similar results as the unreduced and much larger original problem.

Numerous variants are possible to construct a Ritz space from the solution increments

$\boldsymbol{W}_i\boldsymbol{\alpha}_i$ and it is a very critical step of the method. First of all, it is recalled that the GenEO modes of equation (4.3) are interface displacements. Because the solution increments are global interface forces for the whole domain, they can not be used directly as basis to approximate local interface displacements of a single substructure. Instead, they are first mapped by $\boldsymbol{B}^{(s)^T}$ to local interface forces on a substructure s. Subsequently, the inverse of the substructure's Schur complement $(\boldsymbol{S}_D^{(s)})^{-1}$ is applied, to compute the corresponding interface displacements of the substructure. The result is then an adequate basis to approximate the substructures GenEO modes.

According to this mapping process, the Ritz space for a specific substructure s is constructed as

$$\boldsymbol{V}^{(s)} = (\boldsymbol{S}_D^{(s)})^{-1}\boldsymbol{B}^{(s)^T}\boldsymbol{V}_W^{(s)} \qquad \text{where} \qquad \boldsymbol{V}_W^{(s)} = \left[\boldsymbol{W}_0\boldsymbol{\alpha}_0 \,\middle|\, \boldsymbol{W}_1\boldsymbol{\alpha}_1 \,\middle|\, \dots \,\middle|\, \boldsymbol{W}_{n^s-1}\boldsymbol{\alpha}_{n^s-1}\right]$$

Using all available solution increments $\boldsymbol{W}_i\boldsymbol{\alpha}_i$ of the first time step to build the Ritz space would result in a smaller eigenproblem in most cases. However, an intelligent selection can further improve the efficiency, as is shown in the results later. Based on the property of the CG algorithm to capture the largest, well-separated eigenvalues and with them the bad modes first, the most reasonable selection is to consider only the first n^s iterations to build the Ritz space. This parameter is chosen individually for each substructure and directly controls the size of the Ritz space, which is limited by the relation $n^s \leq m$. The specific value of n^s is set by a criterion that is explained later.

First, the GenEO eigenproblem and the according coarse space construction is recalled, which read

$$\begin{aligned}
&\boldsymbol{S}_D^{(s)}\boldsymbol{y}^{(s)} = \theta^{(s)}\boldsymbol{B}^{(s)^T}\boldsymbol{H}\boldsymbol{B}^{(s)}\boldsymbol{y}^{(s)} \\
&\boldsymbol{C}^{(s)} = \left[\boldsymbol{H}\boldsymbol{B}^{(s)}\boldsymbol{y}_1^{(s)} \,\middle|\, \boldsymbol{H}\boldsymbol{B}^{(s)}\boldsymbol{y}_2^{(s)} \,\middle|\, \dots \,\middle|\, \boldsymbol{H}\boldsymbol{B}^{(s)}\boldsymbol{y}_{k^s}^{(s)}\right] \\
&\boldsymbol{C} = \left[\boldsymbol{C}^{(1)} \,\middle|\, \boldsymbol{C}^{(2)} \,\middle|\, \dots \,\middle|\, \boldsymbol{C}^{(N_s)}\right],
\end{aligned} \tag{5.5}$$

where k^s is the number of kept modes from substructure s. Accordingly, the reduced eigenproblem, obtained by projection on the space spanned by $\boldsymbol{V}^{(s)}$, reads

$$\boldsymbol{V}^{(s)^T}\boldsymbol{S}_D^{(s)}\boldsymbol{V}^{(s)}\boldsymbol{q}^{(s)} = \widetilde{\theta}^{(s)}\boldsymbol{V}^{(s)^T}\boldsymbol{B}^{(s)^T}\boldsymbol{H}\boldsymbol{B}^{(s)}\boldsymbol{V}^{(s)}\boldsymbol{q}^{(s)}$$

and in a more extended form

$$\begin{aligned}
&\boldsymbol{V}_W^{(s)^T}\underbrace{\boldsymbol{B}^{(s)}\boldsymbol{S}_D^{(s)^{-1}}\boldsymbol{S}_D^{(s)}\boldsymbol{S}_D^{(s)^{-1}}\boldsymbol{B}^{(s)^T}}_{=\boldsymbol{F}^{(s)}}\boldsymbol{V}_W^{(s)}\boldsymbol{q}^{(s)} = \\
&\quad\widetilde{\theta}^{(s)}\boldsymbol{V}_W^{(s)^T}\underbrace{\boldsymbol{B}^{(s)}\boldsymbol{S}_D^{(s)^{-1}}\boldsymbol{B}^{(s)^T}}_{=\boldsymbol{F}^{(s)}}\boldsymbol{H}\underbrace{\boldsymbol{B}^{(s)}\boldsymbol{S}_D^{(s)^{-1}}\boldsymbol{B}^{(s)^T}}_{=\boldsymbol{F}^{(s)}}\boldsymbol{V}_W^{(s)}\boldsymbol{q}^{(s)}.
\end{aligned}$$

Using the localized interface operators $\boldsymbol{F}^{(s)}$, the reduced eigenproblem can be written in the form

$$\boldsymbol{V}_W^{(s)^T}\boldsymbol{F}^{(s)}\boldsymbol{V}_W^{(s)}\boldsymbol{q}^{(s)} = \widetilde{\theta}^{(s)}\boldsymbol{V}_W^{(s)^T}\boldsymbol{F}^{(s)}\boldsymbol{H}\boldsymbol{F}^{(s)}\boldsymbol{V}_W^{(s)}\boldsymbol{q}^{(s)}. \tag{5.6}$$

According to the original procedure of GenEO, the coarse space is assembled as

$$\boldsymbol{C}^{(s)} = \left[\boldsymbol{H}\boldsymbol{F}^{(s)}\boldsymbol{V}_W^{(s)}\boldsymbol{q}_1^{(s)} \,\middle|\, \boldsymbol{H}\boldsymbol{F}^{(s)}\boldsymbol{V}_W^{(s)}\boldsymbol{q}_2^{(s)} \,\middle|\, \dots \,\middle|\, \boldsymbol{H}\boldsymbol{F}^{(s)}\boldsymbol{V}_W^{(s)}\boldsymbol{q}_{k^s}^{(s)} \right]$$

$$\boldsymbol{C} = \left[\boldsymbol{C}^1 \,\middle|\, \boldsymbol{C}^2 \,\middle|\, \dots \,\middle|\, \boldsymbol{C}^{N_s} \right].$$

The resulting coarse space has a size of

$$\operatorname{rank}(\boldsymbol{C}) = \sum_{s=1}^{N_s} k^s$$

and is limited by the relation $k^s \leq n^s$. For $k^s = n^s$, all computed solutions of the reduced eigenproblems would be included in the coarse space. In this case, it would be unnecessary to solve the reduced eigenproblem at all because the solutions then exactly span the Ritz space, i.e., it would be equivalent to apply the Ritz space as coarse space directly.

Assembly of the Reduced Eigenproblem

While the sizes of the reduced eigenproblems and thus their solution is negligibly small, the assembly of equation (5.6) must be implemented carefully because the solution of a few local Dirichlet problems and some communication between neighboring substructures inevitably remain. The solution of any additional local Neumann problem, i.e., the application of a local operator $\boldsymbol{F}^{(s)}$, can be completely avoided: based on $\boldsymbol{Q}_i^{(s)}\boldsymbol{\alpha}_i$ being readily available from the deflated AMP-FETI procedure listed in algorithm 7, $\boldsymbol{F}^{(s)}\boldsymbol{V}_W^{(s)}$ can be expressed as

$$\boldsymbol{F}^{(s)}\boldsymbol{V}_W^{(s)} = \left[\boldsymbol{Q}_1^{(s)}\boldsymbol{\alpha}_1 \,\middle|\, \boldsymbol{Q}_2^{(s)}\boldsymbol{\alpha}_2 \,\middle|\, \dots \,\middle|\, \boldsymbol{Q}_{n^s}^{(s)}\boldsymbol{\alpha}_{n^s} \right]$$

It must further be noted that the original as well as the reduced GenEO eigenproblem for one substructure only requires the application of the preconditioners $\boldsymbol{H}^{(s)}$ of the substructure itself and its neighbors. Two substructures are called neighbors, if they share at least one dof on their interfaces. This topological restriction is expressed by the function

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

The application of the preconditioner in the reduced eigenproblems of equation (5.6) then reads

$$\boldsymbol{H}\boldsymbol{F}^{(s)}\boldsymbol{V}_W^{(s)} = \sum_{j \in N(s)} \boldsymbol{H}^{(j)}\boldsymbol{F}^{(s)}\boldsymbol{V}_W^{(s)}$$

and consequently requires each substructure to solve additional Dirichlet problems by applying $\boldsymbol{H}^{(s)}$ only to the basis vectors of its own Ritz space as well as to the basis vectors of its neighbors' Ritz spaces.

A Selection Criterion for the Basis of the Ritz Space

An important task that remains, is the choice of n^s, i.e., to choose of how many iterations the solution increments should be included as basis vectors. To select a reasonable number of iterations for building the Ritz space, the behavior of the τ-criterion delivers valuable insight. As long as it selects the local contribution of a specific substructure or its neighbors to be included as a separate direction in the minimization space, it means that the multipreconditioned CG algorithm was probably busy catching high, well-separated eigenvalues whose eigenvectors are very active on the interface of the observed substructure. Such individual contribution should be taken into the Ritz space because they are assumed to be important for the representation of the bad modes. If, on the contrary, the τ-criterion has consistently stopped to select the individual contributions of a substructure and its neighborhood, these contributions are very likely to be not important anymore for representing bad modes. Instead, these are contributions at the end of the iterative process that correspond to very low eigenvalues of the interface flexibility operator, describing mesh modes of high physical frequencies.

The criterion presented and used here to choose a value n^s for each substructure, consequently includes all iterations into the Ritz space up to the point, where the τ-criterion remains consistently inactive for the considered substructure and its neighborhood. In other words, it includes all iterations until no search space contribution of the substructure itself and no search space contribution of one of its neighbors is selected anymore for the rest of the iterations.

A Selection Criterion for the Basis of the Coarse Space

As noted before, it is a waste of computational resources to solve a reduced eigenproblem and subsequently include all computed GenEO vector approximations $\boldsymbol{V}_W^{(s)}\boldsymbol{q}_i^{(s)}$ into the coarse space. The solution only makes sense, if just a small number compared to the Ritz space size is taken into the coarse space. Therefore, a criterion to determine a reasonable k^s is necessary. This problem is identical to the choice of how many GenEO eigenvectors should be taken into the coarse space. Accordingly, the same methods and criteria can be applied here. When the jump criterion or the threshold criterion is combined with the method to choose a value for n^s, the presented local Ritz approximation method can choose a reasonable coarse space completely autonomously. In the following, either the jump criterion or a fixed coarse space size is applied. The fixed coarse space size is used to ensure a maximum of comparability to the results of the GenEO method and its use is indicated in the legends by (n). The use of the jump criterion is indicated in the legends by (j).

5.3.3 Computational and Spectral Efficiency

This section shows the results of applying the two recycling strategy of plain reuse (section 5.3.1) and the recycling strategy of local Ritz approximations (section 5.3.2).

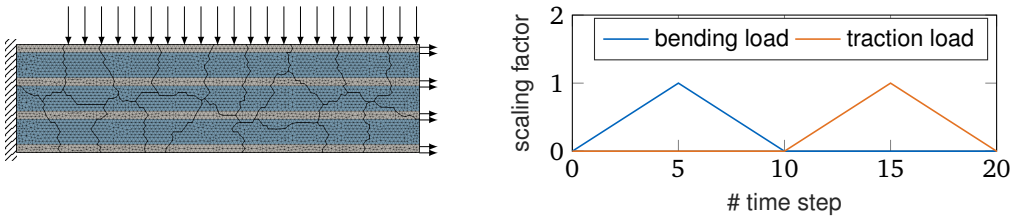


Figure 5.7. Structural problem setup. Left: Mesh, decomposition, material distribution, and applied load. Right: Scaling of the load over time.

Group	**Short Name**	A Priori Coarse Grid	Reuse Space Structure	Eigenproblem Size
reference algorithms	None	-	-	-
	GobalEV	eigenvectors of $\boldsymbol{HF}$	-	# all interface dof
	GenEO	GenEO	-	# interface dof of s
recycling algorithms	PlainReuse	-	$\boldsymbol{W}_i$	-
	RitzGenEO	-	$\boldsymbol{HF}^{(s)}\boldsymbol{W}_i\boldsymbol{\alpha}_i\boldsymbol{q}_k$	Ritz space size n^s

Table 5.1. Summary of algorithms.

The strategies are applied during the solution of the example problem shown in figure 5.7. The mesh, the decomposition, and the material distribution match the example *Stripes/Metis* as described in appendix A. The according physical properties can be found in appendix A in table A.1. The time step size is set to $T_3/10$. Table 5.1 summarizes the investigated algorithms. Beside the recycling strategies, three reference algorithms are considered for comparison. For all algorithms, short names like RitzGenEO are introduced and used in the legends of the plots. Furthermore, the type of the employed a priori coarse space, the basic structure of the recycled space, and the size of the eigenproblems that must be solved are listed for each algorithm in table 5.1. The short names refer to the presented methods as follows.

None: Implements algorithm 7 and $\boldsymbol{\lambda}$ is initialized as zero for each time step. No a priori coarse space or recycling technique is applied. It represents the simplest variant.

GobalEV: Implements algorithm 7 and $\boldsymbol{\lambda}$ is initialized as zero for each time step. Uses an a priori computed coarse space that is built from the eigenvectors associated with the highest eigenvalues of the complete, assembled preconditioned CG operator $\boldsymbol{HF}$. This is, of course, impossible for larger problems and furthermore highly inefficient. It serves as a reference in terms of the best possible reduction of computational cost per coarse space basis vector.

GenEO(n/j): Implements algorithm 7 and $\boldsymbol{\lambda}$ is initialized as zero for each time step. Uses an a priori computed coarse space that is built from GenEO modes, see equation (5.5). Before the solution starts, the GenEO eigenproblem is solved for every substructure. The GenEO modes are selected by their GenEO values.

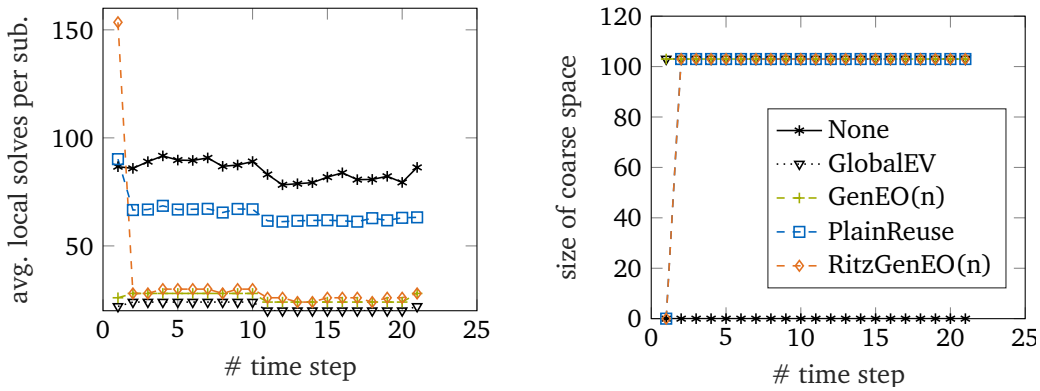


Figure 5.8. Performance of the recycling methods, measured by the average number of local solves per substructure at equal target coarse space sizes.

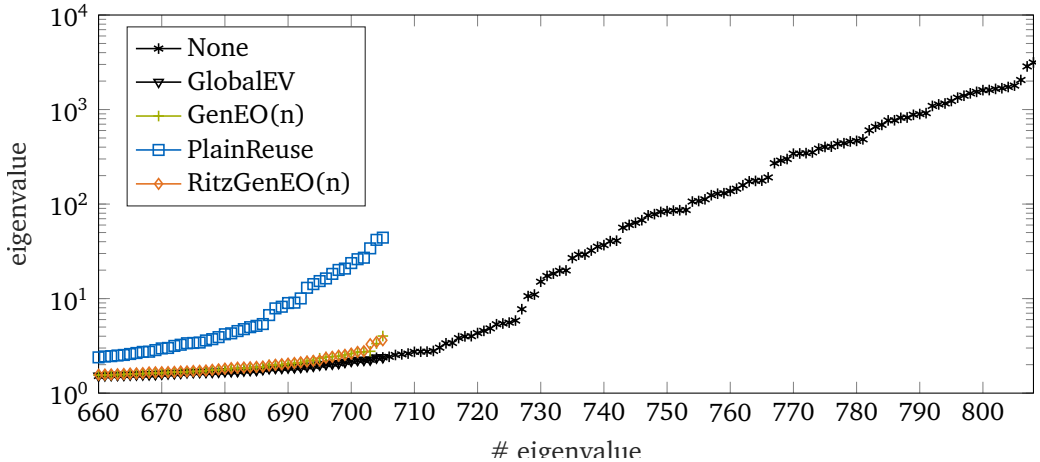


Figure 5.9. Spectrum of the projected preconditioned operator $\boldsymbol{H}\boldsymbol{P}_C^T\boldsymbol{F}$. The ordinate shows the eigenvalues while the abscissa gives its number, sorted in ascending order. Zero eigenvalues, whether they are caused by cross points, i.e., redundant Lagrange multipliers, or deflation by projection, are ignored.

For (j), this is done by the application of the jump criterion, for (n) the coarse space size is set manually and the globally lowest GenEO values are chosen.

PlainReuse: Reuse of individual contributions, see section 5.3.1. All individual basis vectors of the minimization spaces from every iteration are reused. They are selected in the order of the iterations. The individual directions within one iteration are sorted in descending order of the pivot elements in the rank revealing LDLT decomposition. For better comparability, the size of the coarse space is manually limited to be equal to the size of the GenEO coarse space.

RitzGenEO(n/j): A Ritz approximation of the GenEO modes is computed, see section 5.3.2. The Ritz space basis is selected automatically by the method presented in section 5.3.2. The approximated modes per substructure are selected, based on the corresponding Ritz values, as in GenEO either by the jump (j) criterion or a fixed coarse space size (n). In the case of (n), the size of the coarse space is chosen to be equal to the size of the GenEO coarse space to enable better comparability.

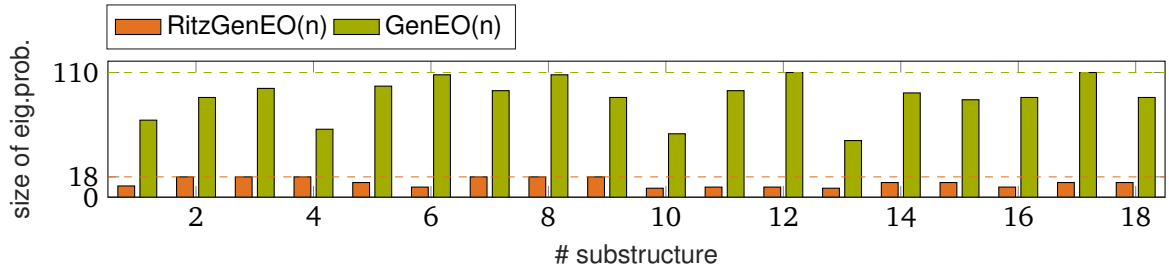


Figure 5.10. Size of the eigenproblem to be solved once for each substructure.

For both recycling strategies, the number of local solves and the size of the coarse space is measured and plotted in figure 5.8. The left of figure 5.8 shows the average number of local solves per substructure per time step. The number of local solves one substructure needs to execute depends on many factors like the number of iterations, the size of the minimization space and on how many neighbors one substructure has. Because all variants employ identically sized coarse spaces from the second time step on, their performance can be compared in terms of local solves. The coarse space size has been manually set to the size that was produced automatically by a GenEO(j) algorithm. The resulting effective eigenspectrum of the preconditioned and projected interface operator is shown in figure 5.9.

The results clearly show, similar to the investigations in chapter 4, that the plain reuse of solution spaces in PlainReuse is not very efficient. Instead, RitzGenEO using a cheap Ritz approximation of the GenEO modes shows a remarkable performance, which is very close to the original GenEO method. In [3], a different example was used to compare these algorithms and RitzGenEO was even able to save more local solves than GenEO. This was explained by the fact that while the original GenEO coarse space is completely independent from the external load, the Ritz space used for approximation is directly derived from the response of the structure to the specific load. The reason for the very high number of local solves in the first time step for RitzGenEO is that the additional local solves, more precisely, the application of the local preconditioners $\boldsymbol{H}^{(s)}$ between neighbors, needed to assemble the Ritz spaces, are included here. In contrast, the a priori cost to compute the GenEO modes is not included in the measurements shown.

The eigenspectra shown in figure 5.9 confirm the close approximation of the original GenEO modes by the Ritz recycling strategy. In particular, the spectra show, how close both methods are to the optimum of removing only the high part of the spectrum. This being considered, the plain reuse strategy clearly delivers better results than the plain reuse strategy investigated for single preconditioning in chapter 4. Of course, this is due to the fact that the individual contributions that were used as coarse space here allow a much better representation of the bad modes than the final step directions of a single preconditioned algorithm.

To illustrate the computational cost more in depth, figure 5.10 shows the size of the eigenproblem that must be solved once for each substructure. For RitzGenEO, the size of an eigenproblem is equal to the number of solution increments that have been

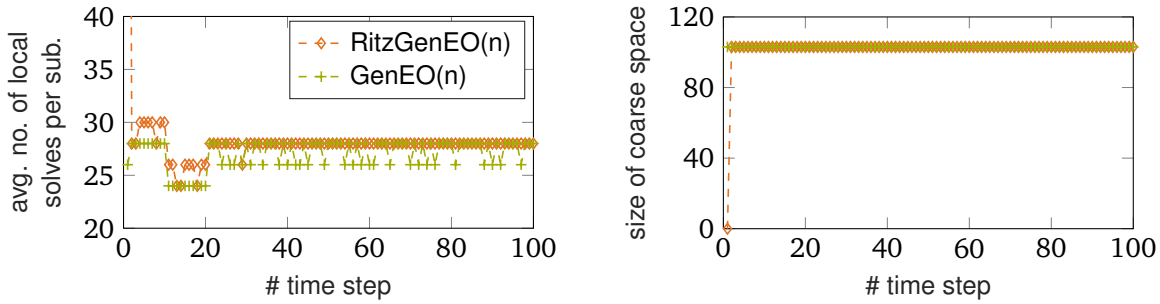


Figure 5.11. Comparison of performance for RitzGenEO and GenEO over a longer time span.

considered to build the specific Ritz space for this substructure. This measure is much smaller for the Ritz approximations than for GenEO. Building the Ritz reduced eigenproblems requires only a small number of vector exchanges between neighbors and vector products as well as some additional local solves, which are already incorporated in figure 5.8. For GenEO, the full Schur complements of the iteration matrices must be exchanged between all neighbors and are then assembled to a significantly larger eigenproblem.

In figure 5.11, RitzGenEO and GenEO are compared over a longer period of time on

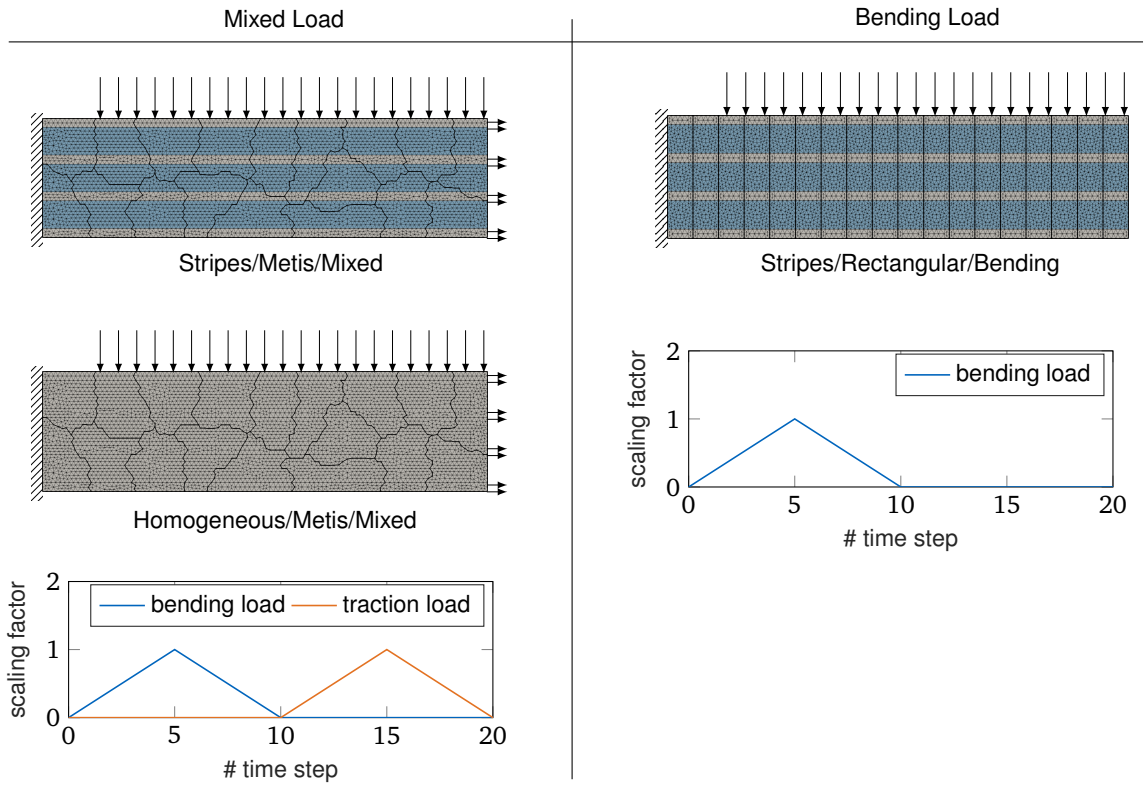


Figure 5.12. Problem setting of the three test cases, showing the mesh, its decomposition, and the material distribution. Furthermore, the direction, the location, and the scaling over time of applied the loads applied load is shown.

Stripes/Metis/Mixed					
Measure	None	GenEO(j)	GenEO(n)	RitzGenEO(n)	PlainReuse
iterations	597	252	252	283	640
size of min. space	2869	252	252	400	864
avg. local solves per sub. in t_1	87	26	26	154	90
avg. local solves per sub. in $\sum_{i=2}^{20} t_i$	**1694**	520	**520**	**546**	1284
size of coarse space in t_{20}	-	103	103	103	103
avg. size of eig.space. in t_1	-	87.9	87.9	13.1	-

Homogeneous/Metis/Mixed					
Measure	None	GenEO(j)	GenEO(n)	RitzGenEO(n)	PlainReuse
iterations	686	316	192	242	305
size of min. space	1379	316	192	275	338
avg. local solves per sub. in t_1	75	32	20	147	78
avg. local solves per sub. in $\sum_{i=2}^{20} t_i$	**1506**	640	**404**	**454**	580
size of coarse space in t_{20}	-	114	114	114	114
avg. size of eig.space. in t_1	-	87.9	87.9	14.6	-

Stripes/Rectangular/Bending					
Measure	None	GenEO(j)	GenEO(n)	RitzGenEO(n)	PlainReuse
iterations	605	590	590	609	722
size of min. space	4384	1061	999	1127	1597
avg. local solves per sub. in t_1	82	60	61	129	82
avg. local solves per sub. in $\sum_{i=2}^{20} t_i$	**1635**	1228	**1219**	**1248**	1520
size of coarse space in t_{20}	-	140	140	140	140
avg. size of eig.space. in t_1	-	113.3	113.3	16.2	-

Table 5.2. Table of results for all three test cases. The number of iterations and the size of minimization space have been accumulated over 20 time steps. The average number of local solves per substructure is given separately for the first time step and accumulated over the remaining time steps.

the test case *Stripes/Metis/Mixed*. As the graph shows, the performance of RitzGenEO does not deteriorate and remains almost identical to GenEO although the Ritz approximation was computed only once from the solution increments of the first time step. For all later time steps, the coarse space remains unchanged.

The methods are applied to two more test cases. The problem settings for all three test cases including the scaling of the load over time are shown in figure 5.12. The physical properties can be found in appendix A, table A.1, where in particular the periods of the three lowest eigenmodes and the period of the highest eigenmode are listed. The time step size is set to $T_3/10$ in all cases where T_3 always refers to the

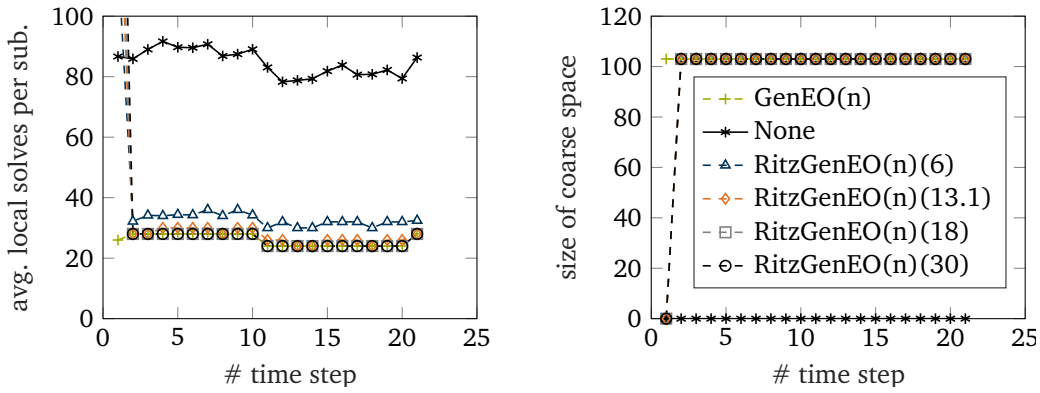


Figure 5.13. Performance of the Ritz approximation based methods with different Ritz space sizes, compared to GenEO for the case *Stripes/Metis/Mixed*.

actual structure.

The results are listed in table 5.2, including the first case *Stripes/Metis/Mixed* which was studied in detail before. The most important numbers, showing the performance without any coarse space, with a GenEO coarse space, and the Ritz approximation method, are emphasized in bold face.In all cases, the GenEO coarse space has a slight advantage over its Ritz approximation, although the differences between the two algorithms remain small. The overall performance of GenEO and RitzGenEO is very good for the METIS decomposition and much worse for the rectangular decomposition. The reason is that the rectangular decomposition is particularly unfavorable. Obviously, the rectangular subdomains have a bad aspect ratio. But what is more important is that each rectangular subdomain includes a full set of four unconnected pieces of stiff fiber. Recalling the mode shape visualizations of section 4.2.3, the bad modes at large time step sizes are especially composed by the different combinations of rigid body movements of the fibers within the soft matrix. Of course, in the case of the rectangular decomposition, each substructure contributes a large number of such combinations. Nevertheless, the None algorithm takes less local solves for the rectangular decomposition than for the METIS decomposition, which is caused by the special feature of the rectangular decomposition that each substructure has at most two neighbors.

5.3.4 Effect of Increasing the Ritz Space Size

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

Figure 5.13 and figure 5.14 show the performance by means of local solves and the effective operator spectra for the case *Stripes/Metis/Mixed* that is illustrated in figure 5.7. The Ritz space size, averaged over all substructures as each substructure

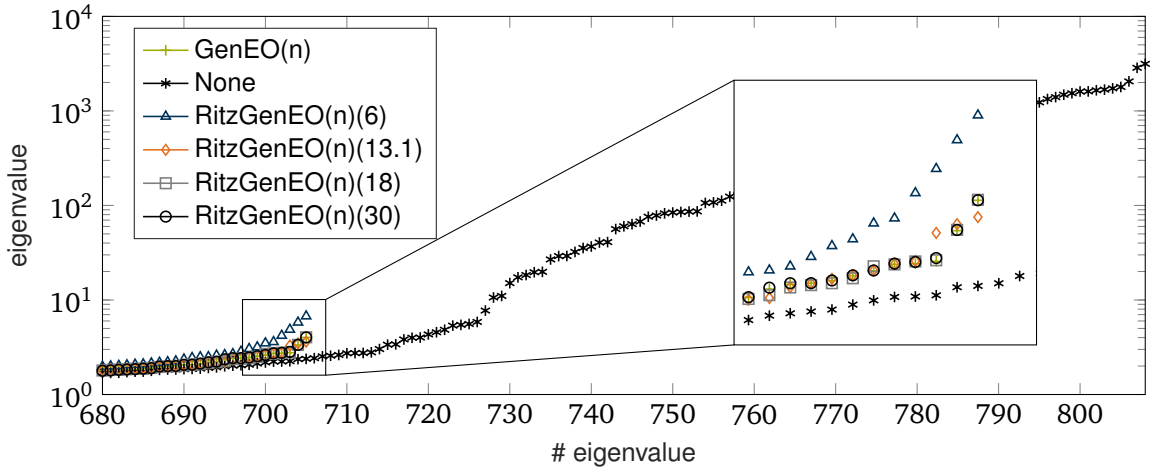


Figure 5.14. Effective spectra for the different Ritz space sizes, governing the convergence of the CG algorithm, for the case *Stripes/Metis/Mixed*. The ordinate shows the eigenvalues of the projected preconditioned operator $\boldsymbol{H}\boldsymbol{P}_C^T\boldsymbol{F}$ while the abscissa gives its number, sorted in ascending order. Zero eigenvalues, whether they are caused by cross points, i.e., redundant Lagrange multipliers, or deflation by projection, are ignored.

can use a different size, is given for each variant in the legend in brackets. The Ritz space size of 13.1 results from the automatic selection method in section 5.3.2, the others were set manually.

The results confirm the assumption on which the Ritz space basis selection method in section 5.3.2 was built: Not all available solution increments must be taken into account to build an efficient coarse space, but only those, where the τ-criterion identifies bad convergence and the minimization step subsequently computes a favorable combination of neighboring contributions. The automatic selection criterion leads to the small Ritz space size of 13.1 but already achieves the same performance as GenEO. A smaller Ritz space, for example 6 basis vectors, shows clearly inferior performance. The criterion has thus not selected an unnecessary large Ritz space. On the other hand, approximating the GenEO coarse space much more precisely, e.g., by a Ritz space size of 30, is not efficient. The number of local solves does not decrease any further. Only the spectra shown in figure 5.14 illustrate how the larger Ritz spaces with 18 and 30 basis vectors converge ever closer to GenEO. However, the exact distribution of eigenvalues is not relevant, as long as their general level and their clustering lead to the same computational performance.

5.3.5 Direct Employment of Ritz Spaces as Coarse Space

Varying the parameters n^s and k^s of the local Ritz approximation method shows that a Ritz space size that is only marginally larger than the number of extracted GenEO vector approximations $\boldsymbol{V}_W^{(s)}\boldsymbol{q}_i^{(s)}$, already leads to very good results. This observation gives rise to the idea to simply use the complete Ritz space as coarse space, equivalent

to setting $k^s = n^s$ for all substructures. This idea, the according algorithm, and results have been published first in [4]. As in exact arithmetic, only the space itself that is used as coarse space is relevant, regardless of the basis chosen. This means that no eigenvalue problem needs to be solved anymore. Noting that $\boldsymbol{HF}^{(s)}\boldsymbol{V}_W^{(s)} = \boldsymbol{HB}^{(s)}\boldsymbol{V}^{(s)}$, the coarse space can be directly constructed as

$$\boldsymbol{C} = \left[\boldsymbol{HF}^{(1)}\boldsymbol{V}_W^{(1)} \,\middle|\, \boldsymbol{HF}^{(2)}\boldsymbol{V}_W^{(2)} \,\middle|\, \dots \,\middle|\, \boldsymbol{HF}^{(N_s)}\boldsymbol{V}_W^{(N_s)} \right]. \tag{5.7}$$

It should be recalled that the spaces $\boldsymbol{V}_W^{(s)}$, constructed as

$$\boldsymbol{V}_W^{(s)} = \left[\boldsymbol{W}_0\boldsymbol{\alpha}_0 \,\middle|\, \boldsymbol{W}_1\boldsymbol{\alpha}_1 \,\middle|\, \dots \,\middle|\, \boldsymbol{W}_{n^s-1}\boldsymbol{\alpha}_{n^s-1} \right],$$

only consist of the solution increments $\Delta\boldsymbol{\lambda}_i$. Consequently, the coarse space construction can also be expressed as

$$\boldsymbol{C}^{(s)} = \boldsymbol{HF}^{(s)}\boldsymbol{V}_W^{(s)} = \left[\boldsymbol{HF}^{(s)}\Delta\boldsymbol{\lambda}_1 \,\middle|\, \boldsymbol{HF}^{(s)}\Delta\boldsymbol{\lambda}_2 \,\middle|\, \dots \,\middle|\, \boldsymbol{HF}^{(s)}\Delta\boldsymbol{\lambda}_{k^s} \right]$$

$$\boldsymbol{C} = \left[\boldsymbol{C}^{(1)} \middle| \boldsymbol{C}^{(2)} \middle| \dots \middle| \boldsymbol{C}^{(N_s)} \right]. \tag{5.8}$$

Each substructure contributes $k^s = n^s$ basis vectors to the coarse space. This results in a coarse space size of

$$\operatorname{rank}(\boldsymbol{C}) = \sum_{s=1}^{N_s} n^s$$

In the following, this algorithm is referred to as RitzDirect.

What remains, is the choice of how many iterations n^s should be included in the space $\boldsymbol{V}_W^{(s)}$ for a certain substructure s. There is no such thing as a Ritz value available that would allow the application of the jump criterion or to sort them in order of importance. Here, the values of $n^s = k^s$ in RitzDirect are chosen as balanced as possible and such that the overall size of the coarse space is equal to that of the GenEO variant to ensure good comparability. To obtain a reasonable coarse space size for RitzDirect automatically, the selection method proposed in section 5.3.2 to choose the sizes of the approximation spaces $\boldsymbol{V}^{(s)}$ could be applied with a more restrictive criterion.

To assess the capabilities of the proposed method, the reference algorithms listed in table 5.1, RitzGenEO, and RitzDirect are applied to the *Stripes/Metis/Mixed* example. The computational cost, described by the average number of local Dirichlet and Neumann solves per substructure, is shown in the left of figure 5.15. The according effective operator spectra are shown in figure 5.16 The results show that the reduction in local solves achieved by RitzDirect is of course inferior to that of RitzGenEO, but still very close to the original GenEO variant. Nevertheless, it leads to significantly better performance than the plain reuse strategy. This is a remarkable result, considering that no eigenproblem and not any other system of equations needs to be solved to obtain the respective coarse space. Only a few additional local solves, which are included in the results, are necessary to compute the application of the preconditioner $\boldsymbol{H}$ to the localized products $\boldsymbol{F}^{(s)}\Delta\lambda_i$. Consequently, the coarse space (5.8) is easy to implement and therefore recommended in cases where, for example, only few right hand sides must be solved. Otherwise, the application of the full RitzGenEO approximation might be superior in the long run.

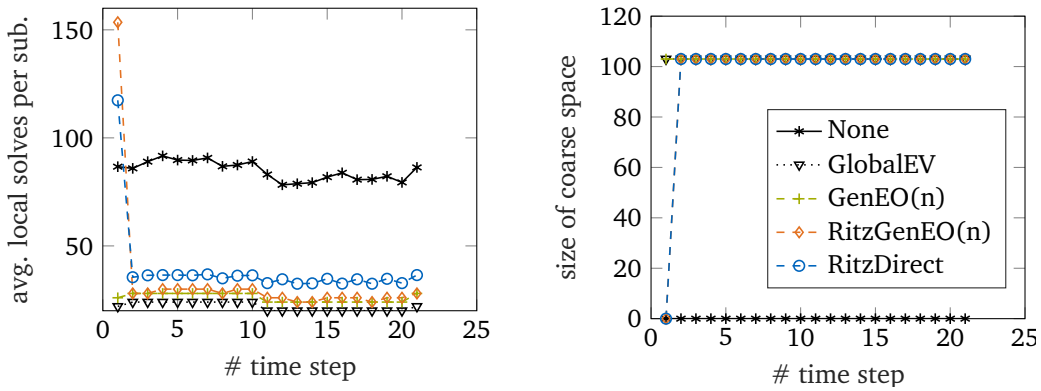


Figure 5.15. Performance of the RitzDirect method in comparison to others, measured by the average number of local solves per substructure at equal target coarse space sizes.

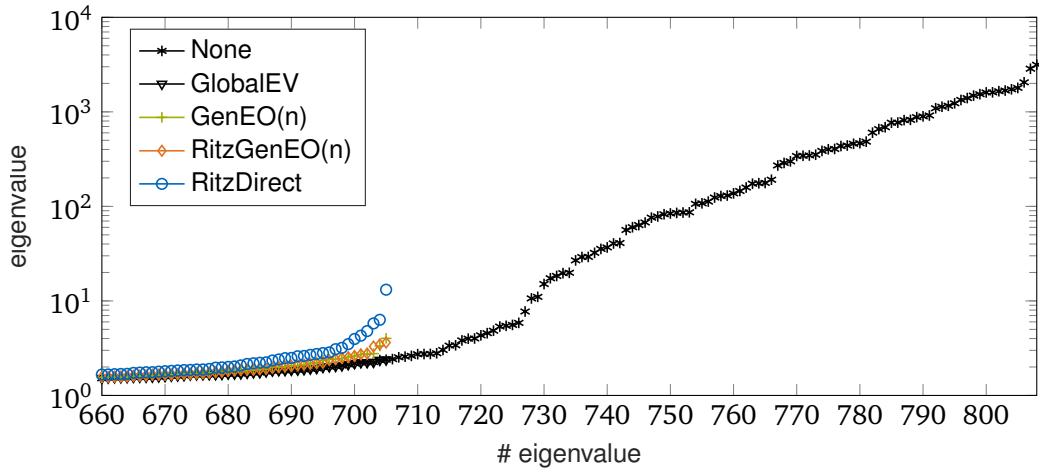


Figure 5.16. Spectrum of the projected preconditioned operator $\boldsymbol{H}\boldsymbol{P}_C^T\boldsymbol{F}$. The ordinate shows the eigenvalues while the abscissa gives its number, sorted in ascending order. Zero eigenvalues, whether they are caused by cross points, i.e., redundant Lagrange multipliers, or deflation by projection, are ignored.

5.4 Conclusion

In this chapter, the application of adaptive multipreconditioned FETI to problems of linear structural dynamics has been investigated. In particular, two main aspects have been considered. First, the possible degeneration of the search space was shown and two different ways to handle it were introduced. Second, the options for solution space recycling within the multipreconditioned framework have been highlighted and three recycling strategies were introduced.

The solution space degeneration occurs in dynamic simulations that start from a state of equilibrium and it can be handled mathematically well by a rank revealing LDLT factorization during the minimization step, which seeks for the simultaneous, optimal step lengths in the multiple directions. However, this is inefficient because the rank revealing LDLT factorization is costly in computational and communicational resources. To relieve this operation from the unnecessary, numerically extremely small and thus linearly dependent directions, the very simple, cheap and effective $\epsilon_{r,\text{sub}}$-

criterion was presented. Used in addition to the rank revealing LDLT factorization, it identifies the unnecessary directions from inactive substructures.

In case the solution space should be recycled for accelerating later solves, this simple identification of inactive substructures and the corresponding reduced search space is not the optimal scenario. Instead, the search space should be fully populated and contain as much as possible information about all substructures, even about those that are still inactive in the first time step. To realize this, a structured initialization vector for the interface forces instead of zero was proposed. This activation strategy is reliably capable to excite all substructures, prevent search space degeneration and forms the basis of any recycling strategy.

The aim of recycling strategies is to efficiently reuse information from prior solution processes, when the same linear operator is solved repeatedly. Three recycling strategies, specifically tailored to multipreconditioning, have been presented and assessed by application to three test cases. All three methods work by building a coarse space that is then employed to accelerate the AMP-FETI algorithm.

The first method, called PlainReuse here, is a straightforward application of an existing method for single-preconditioned FETI. It reuses the plain solution space without further processing as coarse space. It showed minor performance in the results. Nevertheless, its performance is much better than plain reuse of the solution spaces in single-preconditioning because the individual contributions of the substructures as basis vectors can represent the detrimental local effects much better. It is very easy to implement and can thus serve as a quick solution to accelerate multiple right-hand side problems in AMP-FETI.

The second method, called RitzGenEO, approximates the local eigenproblems of GenEO, which are known to provide highly efficient coarse spaces. This is done by applying a Ritz ansatz and consists of three steps. Build a Ritz space basis in which the approximation is sought, construct and solve the local eigenproblems, and finally choose which of their solutions are used to construct the coarse space. The idea to compute a Ritz approximation of GenEO using a prior solution space of AMP-FETI is completely novel and has first been proposed by the author in [3]. The specific choice of the Ritz space basis is crucial. Not only a very effective basis for the Ritz space, but also a selection method to restrict the basis' size has been presented, such that the highly efficient RitzGenEO method results. It achieves about the same performance as the original GenEO with just a rough and thus cheap approximation of it. For the solution of multiple right-hand side problems by a dual domain decomposition technique, RitzGenEO in combination with adaptive multipreconditioning is currently the most advanced and efficient way available.

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

Finally, following the observation that even a very rough approximation by a very

small Ritz space is already effective, a third method was proposed. By selecting only a very small sized Ritz space and simply using it completely as coarse space, the necessary assembly and solution of the eigenproblems is eliminated. What remains, is an extremely cheap method, which costs only little more than the plain reuse strategy but performs rather close to the GenEO coarse space.

What should be investigated in more detail in the future, is the superiority of the AMP-FETI method over the classical FETI method in structural dynamics problems. For the case *Stripes/Metis/Mixed* at $\Delta t = T_3$, the results show an approximate reduction of the local solves from about 190 for FETI to about 90 for AMP-FETI. However, this should be verified for larger examples and different material heterogeneities and time scales.

Chapter 6

Closure

The major topic of this thesis was the development of methods to optimize the efficient applicability of deflated single- and multipreconditioned FETI algorithms to problems of linear structural dynamics. As it is one of the most crucial aspects for high efficiency in this context, the focus was set on the use of coarse spaces, either constructed by a priori computations or by recycling strategies, to accelerate the solution of the multiple right-hand side problems that arise in linear dynamics. This chapter summarizes the most important findings of this thesis and proposes future research topics that could contribute to the development of parallel iterative methods that enable the efficient use of modern parallel computers to simulate the dynamics of mechanical structures.

6.1 Conclusion

Chapter 2 and 3 thoroughly introduced the framework of dual domain decomposition and in particular the deflation of the CG solution algorithm in FETI. This theory review is specifically tailored to the later application to dynamics and the according key aspects, which are the construction and application of coarse spaces and recycling strategies. The specific structure and characteristics of the operators that result from applying time integration schemes are discussed and the application of FETI to solve them iteratively and in parallel is illustrated. On this basis, the chapters explain where the current state of the art lacks of efficiency and which fundamental concepts could be applied to improve it in the special case of linear structural dynamics.

Chapter 4 focused on the numerical assessment of a priori coarse spaces, recycling strategies, and the comparison of performance between them. Furthermore, the chapter enlightened the specific characteristics of the interface problem in linear structural dynamics as function of heterogeneity and the time scale. The visualizations of the GenEO coarse space showed that for very large time steps, the high part of the spectrum of the interface problem consists of substructure modes that are dominated by overall deformation. These introduce strong coupling between many substructures, which is a weakness of the undeflated FETI algorithm and thus requires

according coarse spaces. For very small time steps that are much smaller than necessary to resolve global, physical modes, the important modes that constitute the high part of the spectrum become very localized and confined to a few elements. These effects are covered very well by the preconditioners and coarse spaces are not necessary anymore. Instead, they can even be detrimental to the overall efficiency. The transition zone for the considered homogeneous and heterogeneous examples was found to be situated between time step sizes of $T_3/10$ and $T_3/100$. However, it is difficult to generalize these ranges for arbitrary problems. In the single-preconditioned framework, the SRKS recycling strategy in combination with the newly proposed target condition number criterion ϵ_κ should always be preferred over a priori coarse spaces and other recycling strategies. The ϵ_κ criterion is based only on the absolute Ritz values and not on their convergence and is thus easier to compute and easier to implement. It is the only method that was found to provide highly efficient coarse spaces whose chosen subspace and size is always tailor-made for the specific problem in terms of time scale and heterogeneity. It automatically takes care of the specific requirements on the coarse space, whether it should be large and contain rigid body mode-like components, include specific deformation modes in case of heterogeneous structures, or even be empty for very small time steps. The analysis of principal subspace angles showed that the precise reconstruction of eigenvectors is practically limited to very few eigenvectors at the high end of the spectrum.

Chapter 5 investigated possible issues in the application of multipreconditioning to problems of linear structural dynamics, and three new recycling methods were proposed and assessed by numerical experiments. Setting the initial conditions to a state of equilibrium, which is the most common case, in combination with a localized application of force, can lead to degenerated search spaces. The rank revealing LDLT factorization was found to handle them properly but at unnecessarily high cost. The newly proposed ϵ_r criterion identifies unimportant contributions of inactive substructures by their relative portion of the error energy. It should be used in between the τ-criterion and the rank revealing LDLT factorization, and it was found to handle this problem comprehensively and at negligible additional cost. However, it is not optimal in case that recycling algorithms should be applied because these rely on fully populated search spaces. In the considered case, some substructures are still inactive and exhibit no interface error at the beginning of a time step. As a consequence, the solution process is not able to identify important characteristics of them. An activation strategy was proposed that cures this issue, enriches the search space, and thus lays the optimal foundation for applying a recycling strategy. The plain reuse recycling strategy, transferred in a straightforward way from single- to multipreconditioning, showed much higher potential than in single-preconditioning but still lacks of a useful criterion to choose a reasonable coarse space size and is inferior to the other strategies considered. The RitzGenEO method, which approximates the GenEO coarse space within a part of the former solution space, was proposed and achieved excellent performance close to the exact GenEO coarse space. A very important finding of the investigations is that a very rough approximation of GenEO is sufficient to obtain very good performance. The proposed criterion to automatically tune this approximation quality by setting the size of the approximation space was shown to be close to optimal for the considered examples. This important result was developed further to another new recycling strategy that does not need to solve approximated

eigenproblems anymore. The new strategy, called RitzDirect, showed a remarkable efficiency and constitutes the best combination of an easy to implement and highly efficient recycling strategy for adaptive multipreconditioned FETI that is available at this time.

6.2 Outlook

In the field of parallel domain decomposition methods for structural mechanics, numerous extensive areas of research have to be combined. The mathematical problem to be solved is provided by the modeling and discretization of the physical problem using the finite element method. This includes meshing the geometries, applying element formulations, material laws and time integration schemes. To solve it, solution algorithms belonging to the area of numerical mathematics have to be applied. For this to work all together, it must be implemented as executable computer program, using suitable programming languages. In case that parallel computing hardware should be used, sophisticated implementations that use parallel programming models, for example, the message passing model for distributed memory machines, are necessary. Not only the actual algorithms must be implemented very efficiently but also the communication between the parallel tasks.

Because it is not possible to cover all these aspects in a single thesis, this work is limited to the development of the solution methods and their assessment by application to rather academic examples. To implement the methods, a comprehensive simulation code based on Matlab was developed that covers everything between the meshing and the visualization of the results. However, it is not able to run in parallel on high performance computing clusters. This restriction made it possible to focus on the development of the methods and to investigate a large variety of solution algorithms, coarse spaces, and recycling strategies. Accordingly, many possible future directions to continue this research remain open. The ones considered as most important are presented in the following.

Application to industrial examples. In this thesis, only small two dimensional examples have been considered. They include much less dof than most detailed analyses of real mechanical parts would require. Also the number of subdomains was chosen rather small to preserve a reasonable ratio between the number of dof in the global structure and in one subdomain. For a further verification of the effectiveness of the methods, they should be applied to industrial examples. This means in particular three dimensional problems and a large number of dof. As a consequence, a larger number of subdomains would be possible and aspects like numerical scalability could be investigated. Typical examples would be woven composites like investigated in [107] or automotive tires as considered in [85].

State of the art implementation. To analyze the efficiency of the a priori coarse space and the recycling strategies considered in this thesis, an estimate of the computational cost and gain was computed by a formal weighting of the algorithmic steps. This weighting was based on the results of state of the art implementations applied to industrial examples in other publications. With the Matlab code that was used to generate the results of this thesis, a measurement of real wall clock times of parallel computations was not possible. To further verify the efficiency of the methods proposed in this thesis, such measurements should be considered in the future. For this task, the methods must be integrated into state of the art implementations of AMP-FETI that allow to solve large industrial problems on modern high performance computing hardware with several thousand cores. A very careful implementation of interprocessor communication is crucial herein. This will also enable the analysis of the parallel scalability of the methods.

Update global Ritz approximations in nonlinear problems. This thesis was focused to linear dynamic problems, which is a special case because a series of identical operators with changing right-hand sides must be solved. The proposed methods were accordingly tailored to this requirement. Nevertheless, they offer a high potential for nonlinear problems as well. The most important aspect in this case will be the slight changes of the operator from step to step. In this case, an identification of a fixed set of Ritz vectors has been shown in [92] to already significantly improve the performance. A procedure that would allow to update the global Ritz approximations of the SRKS recycling strategy along with the changing operator could lead to a further substantial improvement of the method. In the linear case, updating Ritz approximations has been considered in [110], but the results of this thesis, i.e., the analysis of principal angles between subspaces in section 4.2.3, indicated that a simple extension of the coarse space already leads to very good results. In contrast, nonlinear problems might benefit from a real update on the existing coarse space basis vectors from time to time.

Update local Ritz approximations in nonlinear problems. The recycling strategy RitzGenEO was limited to the one-time computation of a set of Ritz approximations based on the solution space generated in the first time step. For the application to nonlinear problems, the strategy could be enhanced. Similar to the updating of global Ritz approximations, also the local Ritz approximations of GenEO modes could be updated within the RitzGenEO strategy. One possibility would be to make a real update of the existing set of GenEO vector approximations. The alternative is to rate the quality of a GenEO vector approximation and decide if a complete recalculation of the Ritz approximation should be done. A major advantage of the combination of multipreconditioning and RitzGenEO is the possibility to do all this individually for each substructure. If in a nonlinear problem, only few substructures change their state to such an extent that their bad modes also change significantly, only those would need to update or recalculate their contributions to the coarse space.

Application to nonlinear relocalizations. A recent development in the application of iterative domain decomposition techniques to nonlinear problems are nonlinear relocalizations, which have been investigated in [111] and [112] and within the FETI-DP framework in [113]. Nonlinear relocalizations constitute an additional step within each global Newton iteration. In this step, local Newton iterations are run on each substructure until their internal reaction force is equilibrated with the external interface force. These procedures have been designed to account for very localized nonlinearities that affect only a few substructures. Accordingly, the combination with multipreconditioning and the RitzGenEO recycling strategy holds great potential because both techniques also allow to account very specifically for locally confined effects. It should be investigated if the strategies proposed in this work are capable of creating very selective coarse spaces that cover in particular the bad modes of the few affected substructures. A first approach was already implemented and analyzed in a master's thesis supervised by the author and presented in [114].

Appendix A

Example Structures and Further Results

Example Structures

This section illustrates the example structures that are used throughout this work. Figure A.1 shows the geometry, the mesh, the decomposition, and the Dirichlet boundary conditions. Table A.1 summarizes the physical properties of the examples. In particular, the ratio of the materials' Young's moduli, the Poisson ratios and the periods of the most important eigenfrequencies are listed. All four variants of mesh, material and decomposition are two-dimensional, linear finite element problems and use the same underlying geometry.

The mesh was generated by Gmsh [94] and decomposed into 18 substructures, either automatically by the built-in METIS [95] implementation or manually into rectangular regions. In the heterogeneous structures, the four gray colored thin fibers have the same material properties like the identically colored homogeneous structure. In the heterogeneous examples, the space between the fibers is filled by a much softer matrix, which is colored blue. For the heterogeneous structure with METIS decomposition, a much finer meshed variant, whose elements have about one third of the size of the elements of the coarser mesh, was created and used in chapter 4. The nodes on the left edge are fixed by Dirichlet boundary conditions for all structures.

Parameter		Example 1	Example 2	Example 3	Example 4
Material distribution	-	Stripes	Stripes(Fine)	Stripes	Homogeneous
Decomposition	-	Metis	Metis	Rectangular	Metis
Youngs mod. fiber/matrix	$\frac{E_2}{E_1}$	10^4	10^4	10^4	1
Density fiber/matrix	$\frac{\rho_2}{\rho_1}$	10^1	10^1	10^1	1
Poisson ratio matrix	ν_1	0.49	0.49	0.49	0.3
Poisson ratio fiber	ν_2	0.3	0.3	0.3	0.3
Period of 1. eigenfreq.	T_1	3.2×10^{-1} s	3.4×10^{-1} s	3.2×10^{-1} s	3.8×10^{-2} s
Period of 2. eigenfreq.	T_2	6.6×10^{-2} s	7.5×10^{-2} s	6.7×10^{-2} s	7.6×10^{-3} s
Period of 3. eigenfreq.	T_3	4.2×10^{-2} s	4.2×10^{-2} s	4.2×10^{-2} s	6.2×10^{-3} s
Period of highest eig.f.	T_{min}	1.4×10^{-5} s	3.2×10^{-6} s	1.5×10^{-5} s	1.3×10^{-5} s

Table A.1. Physical properties of the example structures.

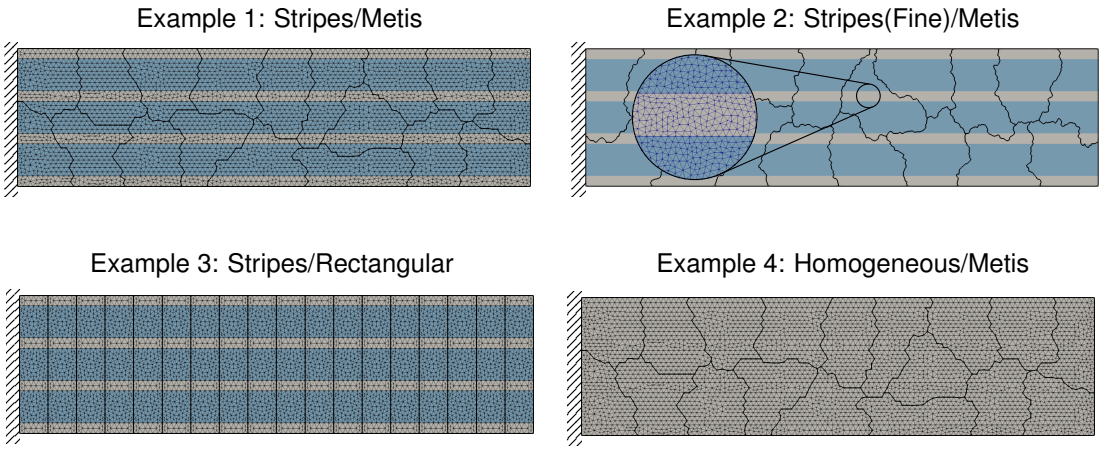


Figure A.1. Structural problem setup.

GenEO Value Distributions for the Homogeneous Case

Figure A.2 shows the GenEO value distributions investigated in section 4.2.2 for the case *Homogeneous/Metis/Mixed*.

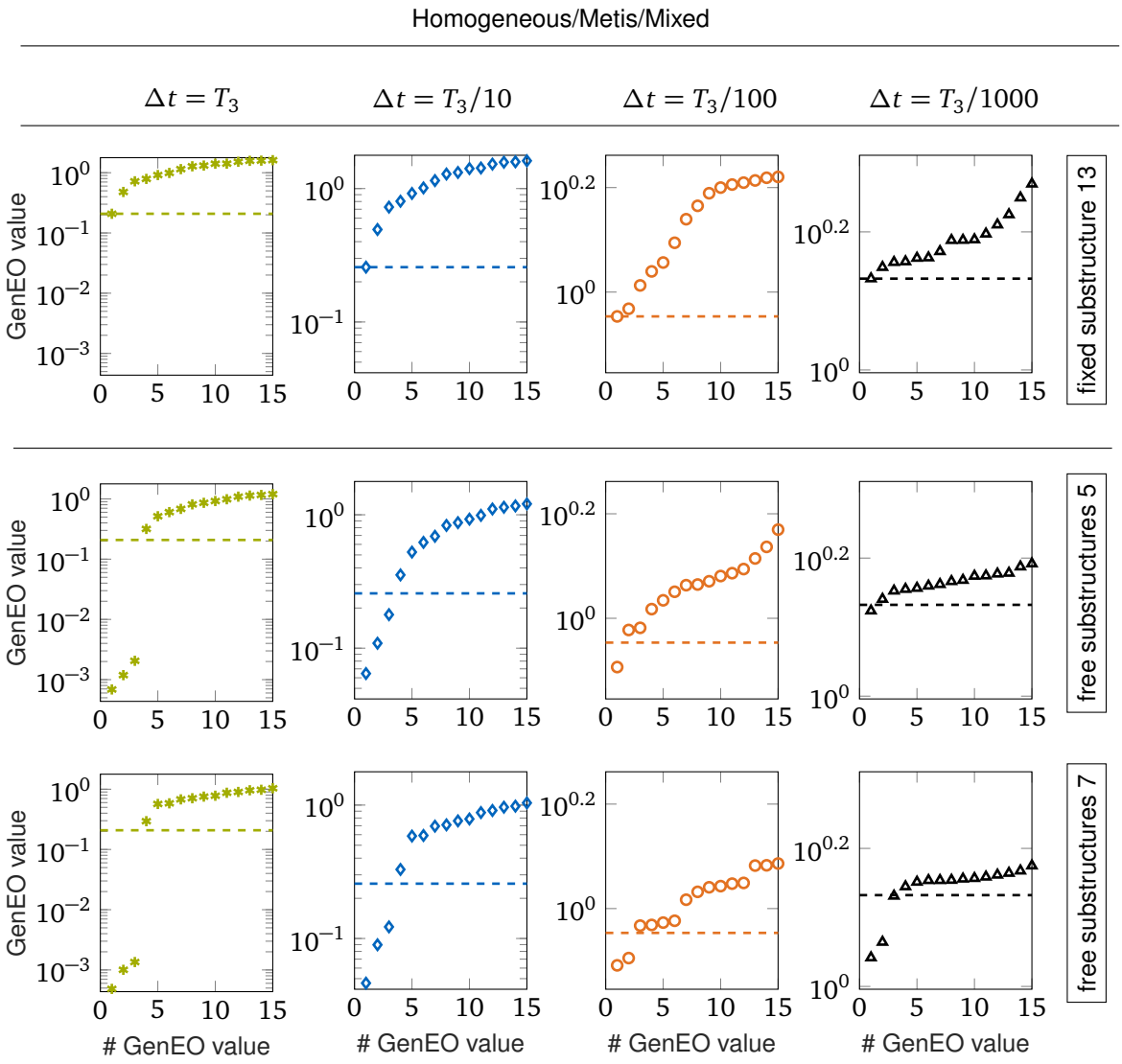


Figure A.2. GenEO values of three substructures for the case *Homogeneous/Metis/Mixed*. For each time step size, the smallest GenEO value of the fixed substructure is indicated by a dashed line.

A Priori Coarse Space Spectra for the Homogeneous Case

Figure A.3 and figure A.4 show the full results of the investigation in section 4.2.4 for the case *Homogeneous/Metis/Mixed*.

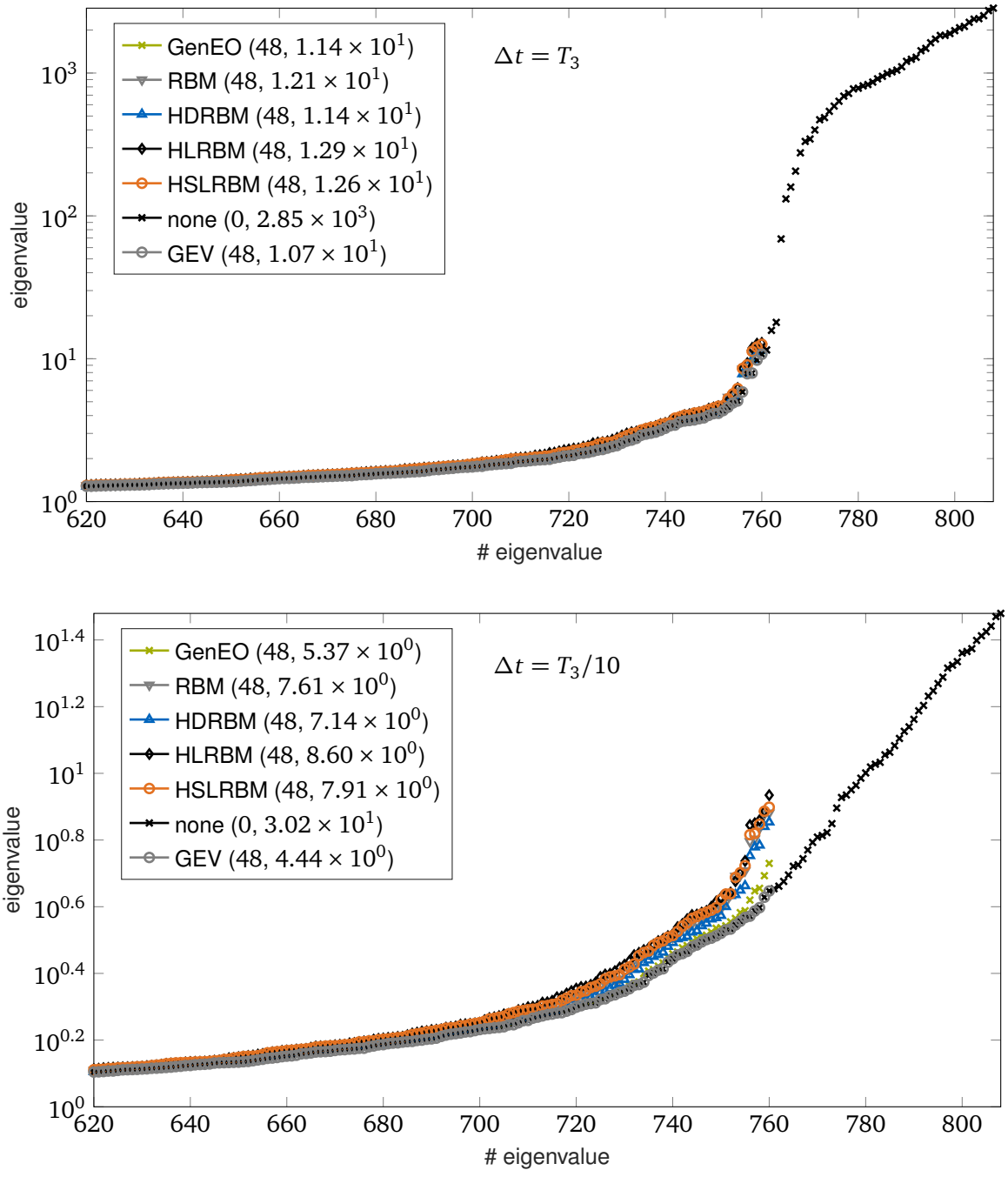


Figure A.3. Spectra of the deflated operator $\boldsymbol{P}_C\boldsymbol{H}\boldsymbol{P}_C^T\boldsymbol{F}$ for the case *Homogeneous/Metis/Mixed* and all coarse spaces of section 4.2.4 for the time step sizes $\Delta t = T_3$ and $\Delta t = T_3/10$. In the legend, the coarse space sizes and the resulting effective condition numbers are given.

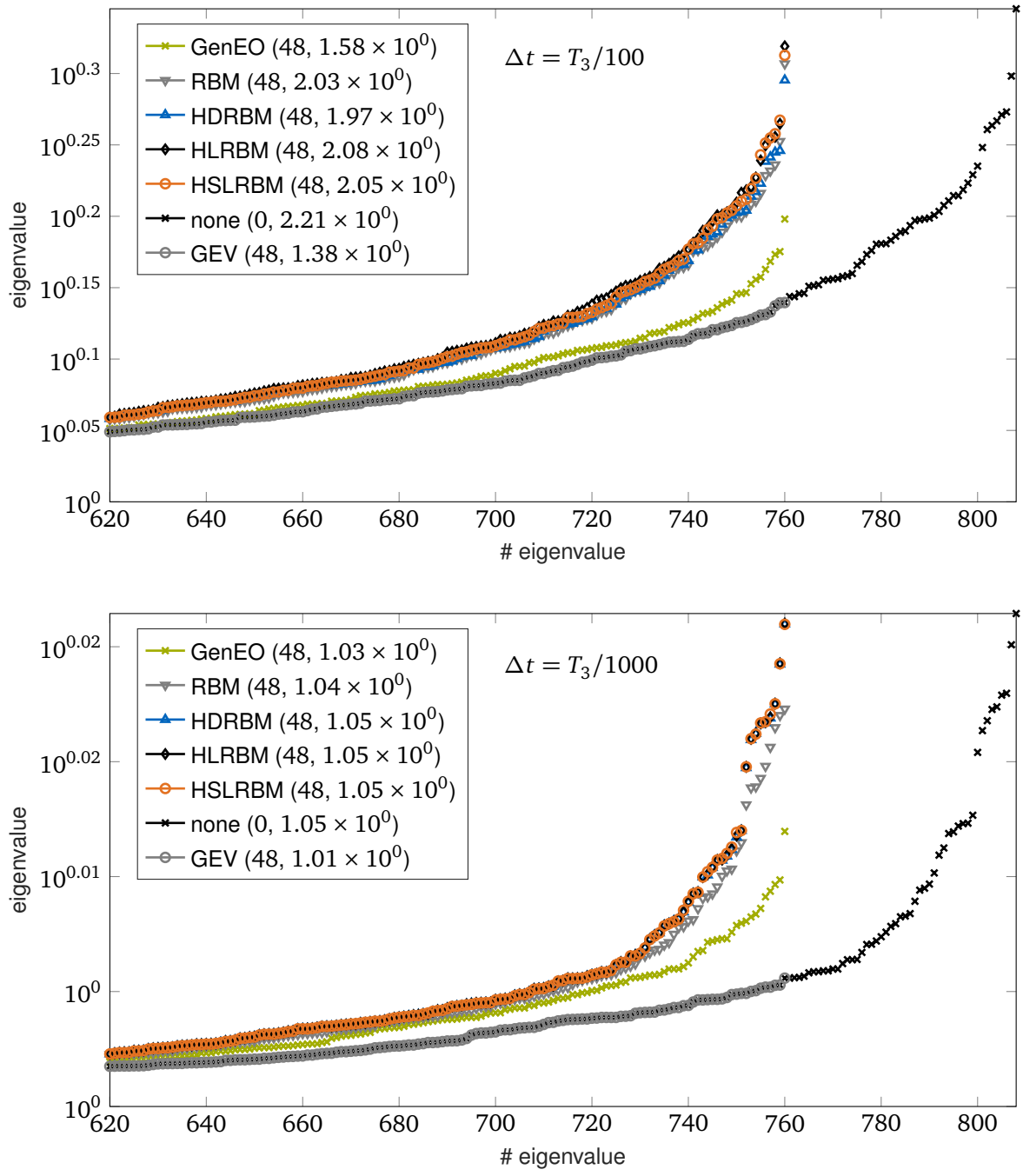


Figure A.4. Spectra of the deflated operator $\boldsymbol{P}_C\boldsymbol{H}\boldsymbol{P}_C^T\boldsymbol{F}$ for the case *Homogeneous/Metis/Mixed* and all coarse spaces of section 4.2.4 for the time step sizes $\Delta t = T_3/100$ and $\Delta t = T_3/1000$. In the legend, the coarse space sizes and the resulting effective condition numbers are given.

A Priori Coarse Space Spectra for the Heterogeneous Case

Figure A.5 and figure A.6 show the full results of the investigation in section 4.2.4 for the case *Heterogeneous/Metis/Mixed*.

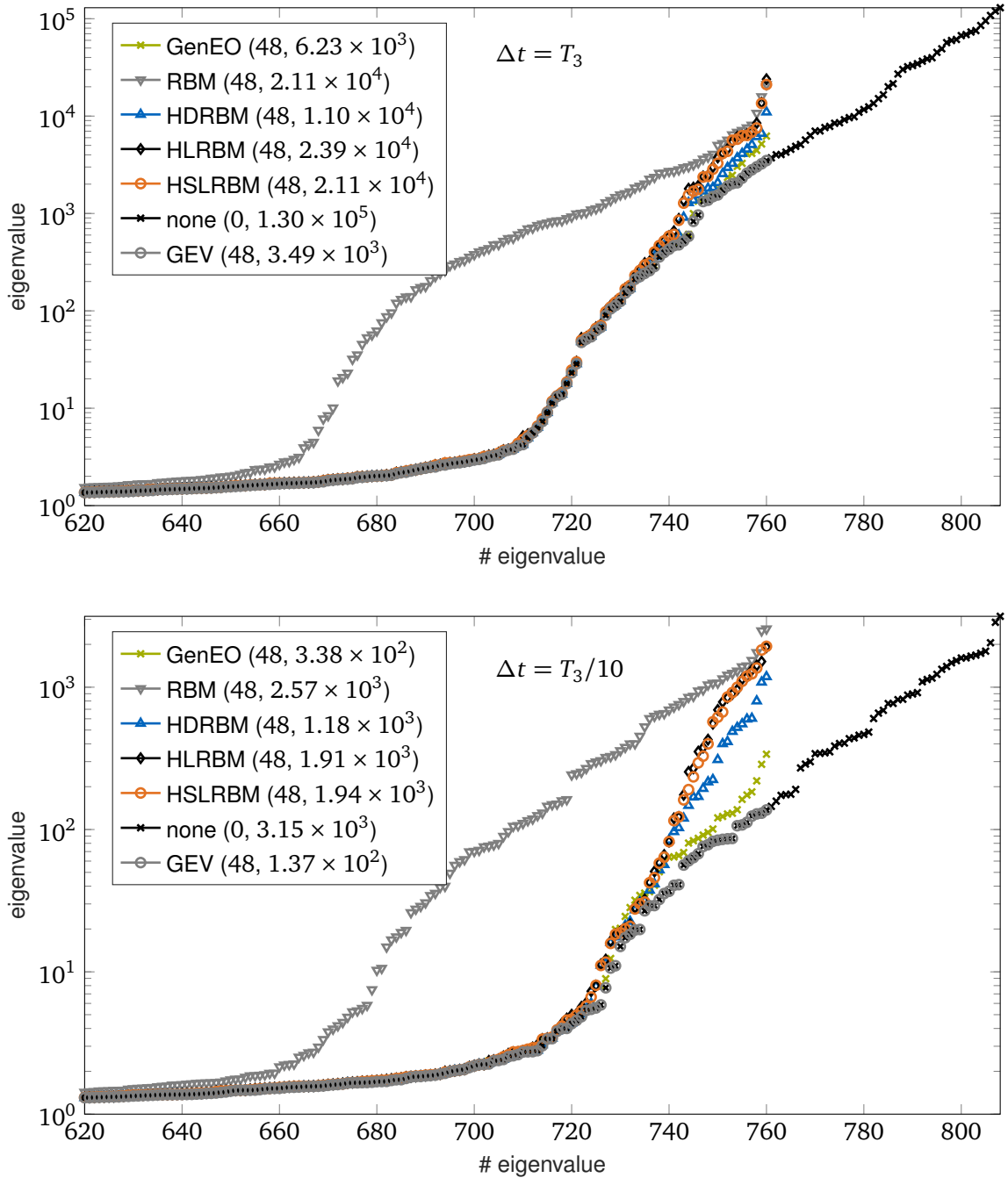


Figure A.5. Spectra of the deflated operator $\boldsymbol{P}_C\boldsymbol{H}\boldsymbol{P}_C^T\boldsymbol{F}$ for the case *Heterogeneous/Metis/Mixed* and all coarse spaces of section 4.2.4 for the time step sizes $\Delta t = T_3$ and $\Delta t = T_3/10$. In the legend, the coarse space sizes and the resulting effective condition numbers are given.

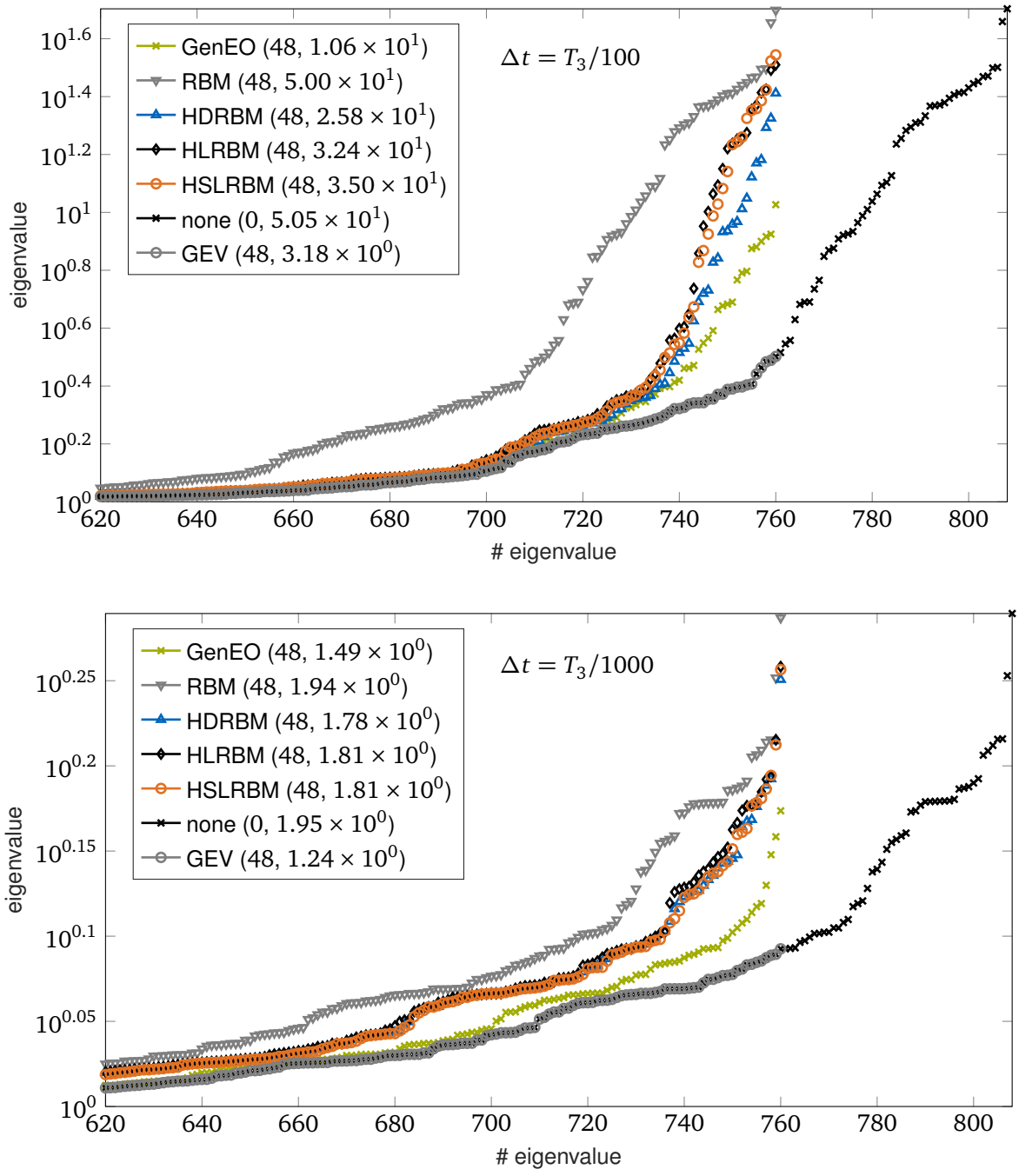


Figure A.6. Spectra of the deflated operator $\boldsymbol{P}_C\boldsymbol{H}\boldsymbol{P}_C^T\boldsymbol{F}$ for the case *Heterogeneous/Metis/Mixed* and all coarse spaces of section 4.2.4 for the time step sizes $\Delta t = T_3/100$ and $\Delta t = T_3/1000$. In the legend, the coarse space sizes and the resulting effective condition numbers are given.

Bibliography

[1] Hennessy, J. and Patterson, D. *Computer Architecture*. 6th Ed. Katey Birtcher, 2017.

[2] Leistner, M., Rixen, D., and Gosselet, P. “Application of multipreconditioned iterative algorithms in dual domain decomposition methods for structural dynamics”. In: *Proc. Appl. Math. Mech.* 17.1 (2017), pp. 329–330.

[3] Leistner, M. C., Gosselet, P., and Rixen, D. J. “Recycling of solution spaces in multipreconditioned FETI methods applied to structural dynamics”. In: *International Journal for Numerical Methods in Engineering* 116.2 (2018), pp. 141–160.

[4] Leistner, M. C., Gosselet, P., and Rixen, D. J. “A simplification of the Ritz-GenEO recycling strategy for adaptive multi-preconditioned FETI applied to multiple right-hand sides”. In: *Proc. Appl. Math. Mech.* accepted (2018).

[5] Wriggers, P. *Nonlinear Finite Element Methods*. Springer, 2008.

[6] Kim, N.-H. *Introduction to Nonlinear Finite Element Analysis*. 2015.

[7] Bathe, K.-J. *Finite Element Procedures. Second Edition*. 2006.

[8] Doyle, J. F. *Nonlinear Structural Dynamics Using FE Methods*. 2015.

[9] Przemieniecki, J. S. “Matrix structural analysis of substructures”. In: *AIAA Journal* 1.1 (1963), pp. 138–147.

[10] Bjørstad, P. E. and Widlund, O. B. “Solving elliptic problems on regions partitioned into substructures”. In: *Elliptic Problem Solvers*. Elsevier, 1984, pp. 245–255.

[11] Dihn, Q., Glowinski, R., and Périaux, J. “Solving elliptic problems by domain decomposition methods wtih applications”. In: *Elliptic Problem Solvers*. Elsevier, 1984, pp. 395–426.

[12] Bjørstad, P. E. and Widlund, O. B. “Iterative methods for the solution of elliptic problems on regions partitioned into substructures”. In: *SIAM Journal on Numerical Analysis* 23.6 (1986), pp. 1097–1120.

[13] Bramble, J. H., Pasciak, J. E., and Schatz, A. H. “An iterative method for elliptic problems on regions partitioned into substructures”. In: *Mathematics of Computation* 46.174 (1986), pp. 361–369.

[14] Hestenes, M. R. and Stiefel, E. “Methods of conjugate gradients for solving linear systems”. In: *Journal of Research of the Natural Bureau of Standards* 49.6 (1952), pp. 409–436.

[15] Bramble, J. H., Pasciak, J. E., and Schatz, A. H. “The construction of preconditioners for elliptic problems by substructuring. I”. In: *Mathematics of Computation* 47.175 (1986), pp. 103–134.

[16] Bramble, J. H., Pasciak, J. E., and Schatz, A. H. “The construction of preconditioners for elliptic problems by substructuring. II”. In: *Mathematics of Computation* 49.179 (1987), pp. 1–16.

[17] Bramble, J. H., Pasciak, J. E., and Schatz, A. H. “The construction of preconditioners for elliptic problems by substructuring, III”. In: *Mathematics of Computation* 51.184 (1988), pp. 415–430.

[18] Bramble, J. H., Pasciak, J. E., and Schatz, A. H. “The construction of preconditioners for elliptic problems by substructuring, IV”. In: *Mathematics of Computation* 53.187 (1989), pp. 1–24.

[19] Mandel, J. “Balancing domain decomposition”. In: *Communications in Numerical Methods in Engineering* 9.3 (1993), pp. 233–241.

[20] Farhat, C. and Roux, F.-X. “A method of finite element tearing and interconnecting and its parallel solution algorithm”. In: *International Journal for Numerical Methods in Engineering* 32.6 (1991), pp. 1205–1227.

[21] Farhat, C. “A lagrange multiplier based divide and conquer finite element algorithm”. In: *Computing Systems in Engineering* 2.2-3 (1991), pp. 149–156.

[22] Farhat, C. and Roux, F.-X. “An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems”. In: *SIAM Journal on Scientific and Statistical Computing* 13.1 (1992), pp. 379–396.

[23] Farhat, C., Lesoinne, M., Le Tallec, P., Pierson, K., and Rixen, D. J. “FETI-DP: A dual-primal unified FETI method part I: a faster alternative to the two-level FETI method”. In: *International Journal for Numerical Methods in Engineering* 50.7 (2001), pp. 1523–1544.

[24] Chan, T. F. and Goovaerts, D. “On the relationship between overlapping and nonoverlapping domain decomposition methods”. In: *SIAM Journal on Matrix Analysis and Applications* 13.2 (1992), pp. 663–670.

[25] Chan, T. F. and Mathew, T. P. A. “Domain decomposition algorithms”. In: *Acta Numerica* 3 (1994), pp. 61–143.

[26] Smith, B. F., Bjørstad, P. E., and Gropp, W. *Domain decomposition : parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, 2004.

[27] Toselli, A. and Widlund, O. B. *Domain Decomposition Methods - Algorithms and Theory*. Springer-Verlag Berlin Heidelberg, 2005.

[28] Mathew, T. P. A. *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations*. Vol. 61. Lecture Notes in Computational Science and Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[29] Dolean, V., Jolivet, P., and Nataf, F. *An Introduction to Domain Decomposition Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2015.

[30] Gosselet, P. and Rey, C. “Non-overlapping domain decomposition methods in structural mechanics”. In: *Archives of Computational Methods in Engineering* 13.4 (2006), pp. 515–572.

[31] Strang, G. *Linear algebra and its applications*. 4th Ed. Belmont, CA: Thomson, Brooks/Cole, 2006.

[32] Farhat, C. and Géradin, M. “On the general solution by a direct method of a large-scale singular system of linear equations: application to the analysis of floating structures”. In: *International Journal for Numerical Methods in Engineering* 41.4 (1998), pp. 675–696.

[33] Géradin, M. and Rixen, D. J. *Mechanical vibrations: theory and application to structural dynamics*. 3rd. Wiley & Sons, 2015.

[34] Farhat, C. and Roux, F.-X. “Implicit parallel processing in structural mechanics”. In: *Computational Mechanics Advances* 2.1 (1994), pp. 1–124.

[35] Rixen, D. J. “Substructuring and dual methods in structural analysis”. PhD thesis. Universite de Liege, 1997.

[36] Rixen, D. J. and Farhat, C. “A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems”. In: *International Journal for Numerical Methods in Engineering* 44.4 (1999), pp. 489–516.

[37] Golub, G. H. and Loan, C. F. van. *Matrix Computations*. Fourth. JHU Press, 2013.

[38] Rixen, D. J. “Dual Schur complement method for semi-definite problems”. In: *Contemporary Mathematics* 18.1 (1998), pp. 341–348.

[39] Farhat, C., Mandel, J., and Roux, F.-X. “Optimal convergence properties of the FETI domain decomposition method”. In: *Computer Methods in Applied Mechanics and Engineering* 115.3 (1994), pp. 365–385.

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

[41] Farhat, C. and Rixen, D. J. “A new coarsening operator for the optimal preconditioning of the dual and primal domain decomposition methods: application to problems with severe coeffcient jumps”. In: *Proceedings of the Seventh Copper Mountain Conference on Multigrid Methods*. Ed. by Duane Melson, N., Manteuffel, T., McCormick, S., and Douglas, C. 1995, pp. 301–316.

[42] Marchuk, G. and Kuznetsov, Y. A. “Theory and applications of the generalized conjugate gradient method”. In: *Advances in Mathematics, Supplementary Studies* 10 (1986), pp. 153–167.

[43] Nicolaides, R. A. “Deflation of conjugate gradients with applications to boundary value problems”. In: *SIAM Journal on Numerical Analysis* 24.2 (1987), pp. 355–365.

[44] Dostál, Z. “Conjugate gradient method with preconditioning by projector”. In: *International Journal of Computer Mathematics* 23.3-4 (1988), pp. 315–323.

[45] Farhat, C., Chen, P.-S., and Mandel, J. “A scalable Lagrange multiplier based domain decomposition method for time-dependent problems”. In: *International Journal for Numerical Methods in Engineering* 38.22 (1995), pp. 3831–3853.

[46] Farhat, C. and Mandel, J. “The two-level FETI method for static and dynamic plate problems Part I: An optimal iterative solver for biharmonic systems”. In: *Computer Methods in Applied Mechanics and Engineering* 155.1 (1998), pp. 129–151.

[47] Farhat, C., Chen, P.-S., Mandel, J., and Roux, F.-X. “The two-level FETI method part II: extension to shell problems, parallel implementation and performance results”. In: *Computer Methods in Applied Mechanics and Engineering* 155.1 (1998), pp. 153–179.

[48] Roux, F.-X. and Farhat, C. “Parallel implementation of direct solution strategies for the coarse grid solvers in 2-level FETI method”. In: *Contemporary Mathematics*. Ed. by Mandel, J., Farhat, C., and Cai, X.-C. Vol. 218. 1998, pp. 158–173.

[49] Gaul, A., Gutknecht, M. H., Liesen, J., and Nabben, R. “A framework for deflated and augmented Krylov subspace methods”. In: *SIAM Journal on Matrix Analysis and Applications* 34.2 (2013), pp. 495–518.

[50] Paige, C. C. “Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem”. In: *Linear Algebra and its Applications* 34 (1980), pp. 235–258.

[51] Roux, F.-X. “Acceleration of the outer conjugate gradient by reorthogonalization for a domain decomposition method for structural analysis problems”. In: *Proceedings of the 3rd International Conference on Supercomputing*. ICS ’89. New York, NY, USA: ACM, 1989, pp. 471–476.

[52] LeVeque, R. J. *Finite Difference Methods for Ordinary and Partial Differential Equations*. Society for Industrial and Applied Mathematics, 2007.

[53] Strakoš, Z. “On the real convergence rate of the conjugate gradient method”. In: *Linear Algebra and its Applications* 154-156 (1991), pp. 535–549.

[54] Mandel, J. and Tezaur, R. “Convergence of a substructuring method with Lagrange multipliers”. In: *Numerische Mathematik* 73.4 (1996), pp. 473–487.

[55] Klawonn, A. and Widlund, O. B. “FETI and Neumann-Neumann iterative substructuring methods: connections and new results”. In: *Communications on Pure and Applied Mathematics* 54.1 (2001), pp. 57–90.

[56] Roux, F.-X. “Spectral analysis of the interface operators associated with the preconditioned saddle-point principle domain decomposition method”. In: *Domain Decomposition Methods for Partial Differential Equations*. Ed. by Keyes, D., Chan, T., Meurant, G., Scroggs, J., and Voigt, R. Norfolk: Siam, 1992, pp. 73–90.

[57] Sluis, A. van der and Vorst, H. A. van der. “The rate of convergence of conjugate gradients”. In: *Numerische Mathematik* 48.5 (1986), pp. 543–560.

[58] Spillane, N. and Rixen, D. J. “Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms”. In: *International Journal for Numerical Methods in Engineering* 95.11 (2013), pp. 953–990.

[59] Farhat, C., Crivelli, L., and Roux, F.-X. “A transient FETI methodology for large-scale parallel implicit computations in structural mechanics”. In: *International Journal for Numerical Methods in Engineering* 37.11 (1994), pp. 1945–1975.

[60] Farhat, C., Crivelli, L., and Géradin, M. “Implicit time integration of a class of constrained hybrid formulations - Part I: spectral stability theory”. In: *Computer Methods in Applied Mechanics and Engineering* 125.1 (1995), pp. 71–107.

[61] Zhou, X. and Tamma, K. K. “Design, analysis, and synthesis of generalized single step single solve and optimal algorithms for structural dynamics”. In: *International Journal for Numerical Methods in Engineering* 59.5 (2004), pp. 597–668.

[62] Har, J. and Tamma, K. K. *Advances in computational dynamics of particles, materials and structures*. Chichester, UK: John Wiley & Sons, Ltd, 2012.

[63] Erlicher, S., Bonaventura, L., and Bursi, O. S. “The analysis of the Generalized-α method for non-linear dynamic problems”. In: *Computational Mechanics* 28.2 (2002), pp. 83–104.

[64] Chan, S. P., Cox, H. L., and Benfield, W. A. “Transient analysis of forced vibrations of complex structural-mechanical systems”. In: *Journal of the Royal Aeronautical Society* 66.619 (1962), pp. 457–460.

[65] Géradin, M. “A classification and discussion of integration operators for transient structural response”. In: *12th Aerospace Sciences Meeting*. Reston, Virigina: American Institute of Aeronautics and Astronautics, 1974.

[66] Hulbert, G. M. “Limitations on linear multistep methods for structural dynamics”. In: *Earthquake Engineering & Structural Dynamics* 20.2 (1991), pp. 191–196.

[67] Hilber, H. M., Hughes, T. J. R., and Taylor, R. L. “Improved numerical dissipation for time integration algorithms in structural dynamics”. In: *Earthquake Engineering & Structural Dynamics* 5.3 (1977), pp. 283–292.

[68] Hilber, H. M. and Hughes, T. J. R. “Collocation, dissipation and ’overshoot’ for time integration schemes in structural dynamics”. In: *Earthquake Engineering & Structural Dynamics* 6.1 (1978), pp. 99–117.

[69] Bathe, K.-J. and Wilson, E. L. “Stability and accuracy analysis of direct integration methods”. In: *Earthquake Engineering & Structural Dynamics* 1.3 (1972), pp. 283–291.

[70] Farhat, C., Chen, P.-S., Risler, F., and Roux, F.-X. “A unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers”. In: *International Journal for Numerical Methods in Engineering* 42.2 (1998), pp. 257–288.

[71] Courant, R., Friedrichs, K., and Lewy, H. “Über die partiellen Differenzengleichungen der mathematischen Physik”. In: *Mathematische Annalen* 100.1 (1928), pp. 32–74.

[72] Dahlquist, G. G. “A special stability problem for linear multistep methods”. In: *BIT* 3.1 (1963), pp. 27–43.

[73] Newmark, N. M. “A method of computation for structural dynamics”. In: *Journal of the Engineering Mechanics Division* 85.85 (1959), pp. 67–94.

[74] Chung, J. and Hulbert, G. M. “A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method”. In: *Journal of Applied Mechanics* 60.2 (1993), pp. 371–375.

[75] Wood, W. L., Bossak, M., and Zienkiewicz, O. C. “An alpha modification of Newmark’s method”. In: *International Journal for Numerical Methods in Engineering* 15.10 (1980), pp. 1562–1566.

[76] Chang, S. Y. “Dissipative, noniterative integration algorithms with unconditional stability for mildly nonlinear structural dynamic problems”. In: *Nonlinear Dynamics* 79.2 (2014), pp. 1625–1649.

[77] Owren, B. and Simonsen, H. “Alternative integration methods for problems in structural dynamics”. In: *Computer Methods in Applied Mechanics and Engineering* 122.1-2 (1995), pp. 1–10.

[78] Piché, R. “An L-stable Rosenbrock method for step-by-step time integration in structural dynamics”. In: *Computer Methods in Applied Mechanics and Engineering* 126.3-4 (1995), pp. 343–354.

[79] Farhat, C., Crivelli, L., and Géradin, M. “On the spectral stability of time integration algorithms for a class of constrained dynamics problems”. In: *34th Structures, Structural Dynamics and Materials Conference*. Structures, Structural Dynamics, and Materials and Co-located Conferences. American Institute of Aeronautics and Astronautics, 1993, pp. 80–97.

[80] Arnold, M. and Brüls, O. “Convergence of the generalized-α scheme for constrained mechanical systems”. In: *Multibody System Dynamics* 18.2 (2007), pp. 185–202.

[81] Fragakis, Y. and Papadrakakis, M. “The mosaic of high-performance domain decomposition methods for structural mechanics - Part II: formulation enhancements, multiple right-hand sides and implicit dynamics”. In: *Computer Methods in Applied Mechanics and Engineering* 193.42-44 (2004), pp. 4611–4662.

[82] Dostál, Z., Kozubek, T., Brzobohatý, T., Markopoulos, A., and Vlach, O. “Scalable TFETI with optional preconditioning by conjugate projector for transient frictionless contact problems of elasticity”. In: *Computer Methods in Applied Mechanics and Engineering* 247-248.Supplement C (2012), pp. 37–50.

[83] Hulbert, G. M. and Hughes, T. J. R. “An error analysis of truncated starting conditions in step-by-step time integration: consequences for structural dynamics”. In: *Earthquake Engineering & Structural Dynamics* 15.7 (1987), pp. 901–910.

[84] Bhardwaj, M., Day, D., Farhat, C., et al. “Application of the FETI method to ASCI problems - scalability results on 1000 processors and discussion of highly heterogeneous problems”. In: *International Journal for Numerical Methods in Engineering* 47.1-3 (2000), pp. 513–535.

[85] Spillane, N. “Robust domain decomposition methods for symmetric positive definite problems”. PhD thesis. Paris: Université Pierre et Marie Curie, 2014.

[86] Farhat, C., Crivelli, L., and Roux, F.-X. “Extending substructure based iterative solvers to multiple load and repeated analyses”. In: *Computer Methods in Applied Mechanics and Engineering* 117.1 (1994), pp. 195–209.

[87] Farhat, C. and Chen, P.-S. “Tailoring domain decomposition methods for efficient parallel coarse grid solution and for systems with many right hand sides”. In: *Contemporary Mathematics*. Ed. by Keyes, D. E. and Xu, J. Vol. 180. 1994, pp. 401–406.

[88] Roux, F.-X. “Parallel implementation of a domain decomposition method for non-linear elasticity problems”. In: *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*. Ed. by Keyes D., Saad, Y., and Truhlar D.G. Society for Industrial and Applied Mathematics, 1995, pp. 161–175.

[89] Rey, C. and Léné, F. “Reuse of Krylov spaces in the solution of large-scale nonlinear elasticity problems”. In: *Ninth International Conference on Domain Decomposition Methods*. 1998, pp. 465–471.

[90] Risler, F. and Rey, C. “On the reuse of ritz vectors for the solution to nonlinear elasticity problems by domain decomposition methods”. In: *Contemporary Mathematics*. Ed. by Mandel, J., Farhat, C., and Cai, X.-C. Vol. 218. 1998, pp. 334–340.

[91] Gosselet, P. and Rey, C. “On a selective reuse of Krylov subspaces in Newton-Krylov approaches for nonlinear elasticity”. In: *Fourteenth international conference on domain decomposition methods* (2003), pp. 419–426.

[92] Gosselet, P., Rey, C., and Pebrel, J. “Total and selective reuse of Krylov subspaces for the resolution of sequences of nonlinear structural problems”. In: *International Journal for Numerical Methods in Engineering* 94.1 (2013), pp. 60–83.

[93] Rey, C. “Une technique d’accélération de la résolution de problèmes d’élasticité non linéaire par décomposition de domaines”. In: *Comptes rendus de l’Académie des sciences. Série II, Mécanique, physique, chimie, astronomie* 322.8 (1996), pp. 601–606.

[94] Geuzaine, C. and Remacle, J.-F. “Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities”. In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331.

[95] Karypis, G. and Kumar, V. “A fast and high quality multilevel scheme for partitioning irregular graphs”. In: *SIAM Journal on Scientific Computing* 20.1 (1998), pp. 359–392.

[96] Björck, A. and Golub, G. H. “Numerical methods for computing angles between linear subspaces”. In: *Mathematics of Computation* 27.123 (1973), pp. 579–579.

[97] Wedin, P. A. “On angles between subspaces of a finite dimensional inner product space”. In: *Matrix Pencils*. Ed. by Kågström, B. and Ruhe, A. Springer, Berlin, Heidelberg, 1983, pp. 263–285.

[98] Knyazev, A. V. and Argentati, M. E. “Principal angles between subspaces in an A-based scalar product: algorithms and perturbation estimates”. In: *SIAM Journal on Scientific Computing* 23.6 (2002), pp. 2008–2040.

[99] Parlett, B. N. *The symmetric eigenvalue problem*. Prentice-Hall, Englewood Cliffs, N.J., 1980.

[100] Saad, Y. “On the rates of convergence of the Lanczos and the block-Lanczos methods”. In: *SIAM Journal on Numerical Analysis* 17.5 (1980), pp. 687–706.

[101] Sluis, A. van der and Vorst, H. A. van der. “The convergence behavior of ritz values in the presence of close eigenvalues”. In: *Linear Algebra and its Applications* 88-89 (1987), pp. 651–694.

[102] Rey, C. and Risler, F. “A Rayleigh–Ritz preconditioner for the iterative solution to large scale nonlinear problems”. In: *Numerical Algorithms* 17.3/4 (1998), pp. 279–311.

[103] Risler, F. and Rey, C. “Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems”. In: *Numerical Algorithms* 23.1 (2000), pp. 1–30.

[104] Gosselet, P., Rixen, D. J., Roux, F.-X., and Spillane, N. “Simultaneous FETI and block FETI: robust domain decomposition with multiple search directions”. In: *International Journal for Numerical Methods in Engineering* 104.10 (2015), pp. 905–927.

[105] Bridson, R. and Greif, C. “A multipreconditioned conjugate gradient algorithm”. In: *SIAM Journal on Matrix Analysis and Applications* 27.4 (2006), pp. 1056–1068.

[106] Spillane, N. “An adaptive multipreconditioned conjugate gradient algorithm”. In: *SIAM Journal on Scientific Computing* 38.3 (2016), A1896–A1918.

[107] Bovet, C., Parret-Fréaud, A., Spillane, N., and Gosselet, P. “Adaptive multipreconditioned FETI: scalability results and robustness assessment”. In: *Computers & Structures* 193 (2017), pp. 1–20.

[108] Higham, N. J. “Analysis of the Cholesky decomposition of a semi-definite matrix”. In: *Reliable Numerical Computation* (1990). Ed. by Cox, M. G. and Hammarling, S. J., pp. 161–185.

[109] Jolivet, P., Dolean, V., Hecht, F., et al. “High performance domain decomposition methods on massively parallel architectures with FreeFem++”. In: *Journal of Numerical Mathematics* 20.3-4 (2012), pp. 287–302.

[110] Saad, Y., Yeung, M., Erhel, J., and Guyomarc’h, F. “A deflated version of the conjugate gradient algorithm”. In: *SIAM Journal on Scientific Computing* 21.5 (2000), pp. 1909–1926.

[111] Pebrel, J., Rey, C., and Gosselet, P. “A nonlinear dual-domain decomposition method: application to structural problems with damage”. In: *International Journal for Multiscale Computational Engineering* 6.3 (2008), pp. 251–262.

[112] Negrello, C., Gosselet, P., Rey, C., and Pebrel, J. “Substructured formulations of nonlinear structure problems - influence of the interface condition”. In: *International Journal for Numerical Methods in Engineering* 107.13 (2016), pp. 1083–1105.

[113] Klawonn, A., Lanser, M., and Rheinbach, O. “Nonlinear FETI-DP and BDDC methods”. In: *SIAM J. Sci. Comput.* 36.2 (2014), A737–A765.

[114] Seibold, A. S., Leistner, M. C., and Rixen, D. J. “Localization of nonlinearities and recycling in dual domain decomposition”. In: *Presented at the Twenty-fifth DD meeting in St. John’s, Newfoundland, Canada, submitted for proceedings*. 2018, pp. 1–8.