Delft University of Technology

Methods for Efficient Analysis of Vibro-Acoustic Problems

Tabak, Umut

DOI
10.4233/uuid:8e548e09-ab1e-45ad-8700-6bc5aa12997e

Publication date
2020

Document Version
Final published version

Citation (APA)
Tabak, U. (2020). *Methods for Efficient Analysis of Vibro-Acoustic Problems*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:8e548e09-ab1e-45ad-8700-6bc5aa12997e

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Methods for Efficient Analysis of Vibro-Acoustic Problems

Methods for Efficient Analysis of Vibro-Acoustic Problems

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
9 March 2020 at 12:30 o'clock

by

Umut TABAK

Master of Science in Mechanical Engineering
Dokuz Eylul University, Izmir, Turkey
born in Izmir, Turkey

This dissertation has been approved by the promotors.

Promotor: Prof. dr. ir. Daniel J. Rixen
Promotor: Prof. dr. Peter G. Steeneken

Composition of the doctoral committee:

Rector Magnificus,	voorzitter
Prof. dr. ir. Daniel J. Rixen	Technische Universität München
Prof. dr. Peter G. Steeneken	Technische Universiteit Delft
Independent members:	
Prof. dr. M. Géradin	University of Liège, Belgium
Prof. dr. N. B. Roozen	Katholieke Universiteit Leuven, Belgium
Prof. dr. C. Vuik	Delft University of Technology
Prof. dr. L. J. Sluys	Delft University of Technology
Prof. dr. A. van Keulen	Delft University of Technology

TU Delft Delft University of Technology

NWO Nederlandse Organisatie voor Wetenschappelijk Onderzoek

Keywords: Vibro-acoustics, Lanczos, component mode synthesis, model order reduction

Printed by: Ipskamp Printing

Cover image: https://pixabay.com

Copyright © 2020 by Umut Tabak

ISBN 978-94-028-1966-3

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

"So, verily, with every difficulty, there is relief",
"Verily, with every difficulty, there is relief".
The Holy Qur'an, Surah Al-Inshirah (Translation: Relief), 94/5 and 94/6

To my family

Contents

Summary xi

Samenvatting xiii

1 Introduction 1
- 1.1 Background 1
- 1.2 Problem statement and motivation 2
- 1.3 Objectives and outline 3
- 1.4 Contributions of the thesis 4

2 Vibro-acoustic Modelling 7
- 2.1 Introduction. 7
- 2.2 Structural domain and discretization 8
- 2.3 Acoustic domain and discretization 11
- 2.4 Coupling conditions, coupled equations and properties. 13
- 2.5 Computational platform 16
- 2.6 Summary 17

I Investigations on Eigenvalue Solution Strategies for Vibro-Acoustic Simulations 19

3 Iterative Reduced Correction Algorithm, *IRCA* 21
- 3.1 Introduction. 21
- 3.2 Overview on Subspace iteration 22
- 3.3 Iterative reduced correction algorithm, *IRCA*. 24
  - 3.3.1 Modal interaction model and the starting basis. 24
  - 3.3.2 Correction vectors and their computations 26
  - 3.3.3 Enrichment with residual vectors 29
  - 3.3.4 Projection space and the interaction problem. 30
  - 3.3.5 Energy based convergence check 31
- 3.4 Notes on coupled residual, convergence and relation with Krylov vectors. 34
- 3.5 Additional practical considerations 36
  - 3.5.1 Remarks on the target mode count and basis size 36
  - 3.5.2 Convergence acceleration: use of buffer vectors 36
- 3.6 Numerical case studies 38
  - 3.6.1 Test case 1, air filled cavity 39
  - 3.6.2 Test case 2, water filled cavity 40
- 3.7 Conclusions and discussion. 41

4 *vibro*-Lanczos, a Symmetric Eigenvalue Solver for Vibro-Acoustic Problems 43
4.1 Introduction. 43
4.2 Nonsymmetric eigenvalue solvers for vibro-acoustic problems 45
  4.2.1 Two-sided Lanczos method. 45
  4.2.2 Arnoldi method 47
  4.2.3 Exploiting the matrix topology for computations. 49
4.3 Symmetric forms of the nonsymmetric problem. 49
4.4 Symmetric *vibro*-Lanczos algorithm 50
  4.4.1 Overview of the symmetric Lanczos eigensolver. 50
  4.4.2 Symmetric Lanczos iterations for vibro-acoustics 51
  4.4.3 Details on filtering rigid modes for the symmetric representation 54
  4.4.4 Convergence check 56
  4.4.5 Cost reduction with partial or selective orthogonalization . . . 56
  4.4.6 Implementation details for partial reorthogonalization 58
  4.4.7 Remarks on spectral transformation with symmetry 59
4.5 Numerical case studies 60
  4.5.1 Test case 1 61
  4.5.2 Test case 2 63
4.6 Summary and conclusions 68

II Investigations on Substructuring Techniques for Vibro-Acoustic Simulations 69

5 Locally Symmetric Craig-Bampton Approach for Vibro-Acoustic Problems, *vibro*-LsCB 71
5.1 Introduction. 71
5.2 General overview on Reduction and Component Mode Synthesis . . . 72
  5.2.1 Dynamic model reduction by modal tranctation 72
  5.2.2 Component Mode Synthesis, a non-mathematical overview . . 74
5.3 Craig-Bampton reduction for vibro-acoustics 75
  5.3.1 Brief overview of Craig-Bampton approach 75
  5.3.2 Options for symmetric vibro-acoustic component matrices and its advantage 77
  5.3.3 Ingredients of the reduction basis for vibro-acoustic components, $\mathbf{R}_{cb}$ 79
  5.3.4 Reduced matrices and their structure 84
5.4 Assembling reduced component matrices 88
  5.4.1 Component coupling with symmetric matrices 88
  5.4.2 Effect of symmetric reduced component matrices on assembly. 95
  5.4.3 Assembly of τ-symmetric reduced vibro-acoustic component matrices 96

5.5 Some notes on implementation and analysis of assembled system . . . 102
5.6 Numerical examples . . . 104
  5.6.1 Test case 1: academic model . . . 106
  5.6.2 Test case 2: *pseudo* industrial-sized model . . . 108
5.7 Summary and conclusions . . . 108

6 Extending *vibro*-LsCB for Systems with Singularities 113
6.1 Introduction . . . 113
6.2 Solution with singularities . . . 114
6.3 Representation of symmetric component matrices with singularities . 116
  6.3.1 Format of the reduced component matrices in the extended coordinates . . . 119
  6.3.2 Effect of singularity transformations on compatibility relations of reduced component matrices . . . 120
  6.3.3 Generalized primal assembly . . . 122
6.4 Numerical example . . . 125
6.5 Summary and conclusions . . . 128

7 Conclusions & Recommendations 129
7.1 Conclusions of the thesis . . . 129
  7.1.1 Developments on eigenvalue solution techniques . . . 129
  7.1.2 Developments on model reduction techniques . . . 130
7.2 Recommendations for further investigations. . . . 132

A Model Reduction Techniques in Different Application Fields 135
A.1 Introduction. . . . 135
A.2 Review of model reduction techniques. . . . 136
  A.2.1 Mode displacement methods . . . 137
  A.2.2 Krylov subspace based model order reduction . . . 142
  A.2.3 Balanced truncation . . . 146
A.3 Qualitative comparison on model reduction methods. . . . 152
  A.3.1 Projection . . . 152
  A.3.2 General comparison. . . . 153
  A.3.3 Moment matching and model truncation augmentation . . . 156
  A.3.4 Computational aspects . . . 157
  A.3.5 Preservation of properties . . . 158
A.4 Illustrative example. . . . 159
A.5 Conclusions . . . 163

B Pre- and post-multiplier matrices 167
B.1 Pre-multiplication matrices . . . 167
B.2 Post-multiplication matrices . . . 168

C Convergence Proofs on Subspace Iteration 169
C.1 Refresher on inverse iteration and its convergence . . . 169
C.2 Convergence of Subspace iteration. . . . 171

D Proofs on 3-term Recurrence Relations for Partial Reorthogonalization 173

E Relation of Left and Right Eigenvector with τ 175

Curriculum Vitae 189

List of Publications 191

Acknowledgements 193

Summary

In this thesis, we present numerical techniques for analyzing vibro-acoustic problems that emerge from the coupling of elastic structures with fluid domains. The main analysis tool used for numerical modelling is the Finite Element Technique (FEM). By using FEM, the goal of our research was to either improve the already available techniques or propose new ones for problems involving this type of coupling. The thesis is divided into two main topics/parts which can be followed independently.

The first part of the thesis focuses on developing eigenvalue solvers for the non-symmetric coupled vibro-acoustic eigenvalue problems. Two iterative techniques are investigated for this purpose as follows:

- For the first technique in Chapter 3, starting with the uncoupled modes of the structural and fluid domains, these initial vector blocks are improved in an iterative manner similar to the well known Subspace iteration technique. Each iteration step builds a projection space enriched with the so-called *correction vectors*. These correction vectors are composed of some vectors represented in the space of the decoupled basis vectors, modal truncation vectors and static residual vectors. Practical examples are provided to demonstrate the accuracy of the technique.

- The second technique outlined in Chapter 4 builds on a transformation matrix which is used to transform the original non-symmetric eigenvalue problem into a symmetric one. However, the symmetric counter parts of the original non-symmetric system stiffness or mass matrix become fully populated in closed form which is of little practical use from a numerical efficiency point of view. Nevertheless, it is shown that an efficient Lanczos type iteration scheme can be built without explicitly constructing these matrices. By using the theoretical three-term recurrence formulas, we show that a partial re-orthogonalization scheme can be integrated in the iteration process to reduce the numerical cost of the repeated orthogonalization operations. The developed solver, *vibro*-Lanczos, is benchmarked against the well-known non-symmetric eigenvalue solution techniques on different test cases. It is shown that the results are highly accurate and can bring a CPU time saving on the order of 10% for large target eigenvalue counts.

The second part of the thesis focuses on a Model Order Reduction (MOR) technique for coupled vibro-acoustic problems. To reduce the cost of simulation in those problems, a Craig-Bampton (CB) based Component Mode Synthesis (CMS) approach is detailed in this part as follows:

- Based on the transformation matrix used in Chapter 4 and starting with Chapter 5, we present the development of an efficient CMS based approach for an-

alyzing vibro-acoustic problems, namely, *vibro*-LsCB. We show that we can create, analyze and perform the reduction on *locally* symmetric component matrices. Efficient numerical solution schemes are outlined which emerge due to the use of the transformation matrix. We show the details on component coupling either in a primal or dual framework. Initial performance tests show that the proposed technique can accurately predict the global system dynamics.

- Chapter 6 extends the technique presented in Chapter 5 to systems where the operator matrices used in the numerical operations become singular. These singularities originate from the free-free components where no constraints are applied on. We develop and show the mathematical details where we can still end up with *locally* symmetric components in this case also. As a result, the method presented in Chapter 5 is generalized within this chapter. The accuracy of the technique is illusrated with a 2D-academic test problem.

Samenvatting

In dit proefschrift presenteren we numerieke technieken voor het analyseren van vibro-akoestische problemen die ontstaan uit de koppeling van elastische structuren met vloeistof domeinen. Het belangrijkste analyse instrument dat gebruikt wordt voor numerieke modellering is de Eindige Elementen Techniek (FEM). Door FEM te gebruiken is het doel van ons onderzoek om, of de al bestaande techniek te verbeteren, of nieuwe technieken voor te stellen voor problemen met dit type koppeling. Het proefschrift is opgedeeld in twee hoofd onderwerpen/delen die onafhankelijk van elkaar gelezen kunnen worden. Het eerste deel van het proefschrift richt zich op de ontwikkeling van eigenwaarde oplossers voor niet-symmetrische gekoppelde vibro-akoestische eigenwaarde problemen. Met dit doel worden twee iteratieve technieken bestudeerd:

- Voor de eerste techniek in hoofdstuk 3, met als uitgangspunt de ongekoppelde modes van de structurele en fluidische domeinen, worden deze initiële vector blokken verbeterd op een iteratieve manier, net als de bekende Subspace Iteration Techniek. Elke iteratiestap bouwt een projectieruimte op die verrijkt is met de zogenaamde correctievectoren. Deze correctievectoren bestaan uit vectoren die uitgedrukt worden in de ruimte van de ongekoppelde basisvectoren, modale truncatievectoren en statische residuvectoren. Praktische voorbeelden worden gegeven om de nauwkeurigheid van de techniek aan te tonen.

- De tweede techniek, die in hoofdstuk 4 beschreven wordt, bouwt op een transformatiematrix die gebruikt wordt om het originele niet-symmetrische eigenwaarde probleem te transformeren in een symmetrisch probleem. Maar, de symmetrische versie van de originele niet-symmetrische stijfheids- of massamatrix van het systeem wordt volledig gevuld in gesloten vorm, en is dus praktisch weinig bruikbaar vanuit een oogpunt van numerieke efficiëntie. Desalniettemin, wordt aangetoond dat een efficiënte, Lanczos-type, iteratiemethode opgebouwd kan worden zonder expliciet deze matrices te construeren. Door de theoretische drie-term ‘recurrence’ formules te gebruiken, tonen we aan dat een gedeeltelijke re-orthogonalisatie methode geïntegreerd kan worden in het iteratieproces, om de numerieke kosten van de herhalende orthogonalisatie stappen te reduceren. De ontwikkelde oplosser, vibro-Lanczos, wordt gebenchmarkt tegen bekende niet-symmetrische eigenwaarde oplostechnieken voor verschillende proefproblemen. Het wordt aangetoond dat de resultaten zeer accuraat zijn en een computer tijdsbesparing van zo’n 10% kunnen geven voor grote aantallen doeleigenwaardes.

Het tweede deel van het proefschrift richt zich op een Model Orde Reductie (MOR) techniek voor gekoppelde vibro-akoestische problemen. Om de simulatiekosten in deze problemen te reduceren, wordt een Craig-Bampton (CB) gebaseerde Component Mode Synthese (CMS) aanpak in dit deel als volgt beschreven:

- Op basis van de transformatiematrix die in hoofdstuk 4 gebruikt werd, en vanaf hoofdstuk 5, presenteren we de ontwikkeling van een efficiënte CMS gebaseerde aanpak voor het analyseren van vibro-akoestische problemen, namelijk, vibro-LsCB. We tonen aan dat we creatie, analyse en reductie kunnen toepassen op lokaal symmetrische component-matrices. Efficiënte numerieke oplosmethodes worden beschreven, die ontstaan door het gebruik van de transformatiematrix. We laten de details van component koppeling zien, of in een primaire of in een duale opzet. Eerste prestatietests laten zien dat de voorgestelde techniek accuraat de globale systeemdynamica kan voorspellen.

- Hoofdstuk 6 breidt de in hoofdstuk 5 gepresenteerde techniek uit naar systemen waar de operatiematrices, die gebruikt worden in de numerieke methode, singulair worden. Deze singulariteiten ontstaan uit de ‘free-free’ componenten waar geen beperkingen aan worden opgelegd. Ook in dit geval ontwikkelen en tonen we de wiskundige details waar we nog kunnen uitkomen op lokaal symmetrische componenten. Als eindresultaat wordt de methode, die in hoofdstuk 5 geïntroduceerd werd, in dit hoofdstuk gegeneraliseerd. De accuratesse van de techniek wordt geïllustreerd met een 2D academisch testprobleem.

1

Introduction

1.1. Background

Nobody wants to drive a car which has a lot of booming noise. Equally, nobody will also prefer to be on a transatlantic flight spending hours with a similar distracting noise, most from the engines. Eventually, no customer will be happy with the rattling parts of a product that make some irritating sound. All together, these characteristics contribute to the concept known as **N**oise, **V**ibration and **H**arshness (**NVH**). In the last decades, the pressure to improve these combined characteristics has increased substantially due to increased comfort requirements.

An example from automotive industry is as follows:

Vibration energy from an engine travels through the engine mounts, into the structure, and through the car seat into the driver. ***But energy from the same source can take a similar path through the structure to become acoustic noise when structure and the internal fluid are coupled and, subsequently, this interaction reflects sound into the cabin.*** *Optimizing these factors is therefore of utmost importance for the overall experience of the vehicle* [28].

This short overview draws the attention to the importance of the prediction of these qualities at early design stages. More importantly, the bold lines above bring us to the focus of the dissertation, namely,

investigation of problems with ***structural-acoustic*** *or* ***vibro-acoustic*** *coupling.*

For problems involving structural-acoustic coupling, it is quite important to predict the coupled dynamic response of structures with internal and/or external fluid domains. Prediction of the acoustic response or pressure profile in the passenger cabin of an automobile or of an aircraft is an important example that can help engineers to improve the above mentioned qualities. Another example is the analysis of exterior sound fields created by vibrating objects. In these examples, the structure and the fluid are in strong interaction and, therefore, coupled. This mutual coupling also has a strong influence on the dynamic properties of the systems and, most of the time, can not be neglected. Eventually, these systems should be analyzed by taking this coupling into account for realistic performance predictions.

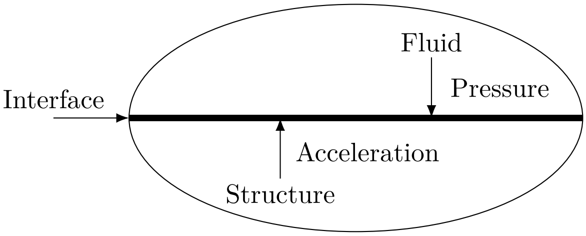


Figure 1.1: Conceptual interaction between the structural and fluid domains

1.2. Problem statement and motivation

In this thesis, we mainly concentrate on structural-acoustic problems where elastic structures are interacting with enclosed acoustic fluid cavities. A simplified schematic representation of this type of coupling is depicted in Figure 1.1. In this type of coupling, the pressure on the fluid side acts on the structural domain over the interface. Similarly, again over the interface, the normal acceleration of the structural domain is transmitted to the fluid domain. Systems including this coupling is extensively investigated since the original article by Craggs [34] where it is shown that the coupled system in displacement-pressure formulation results in non-symmetric matrices with the coupling terms included. An extensive mathematical exposition is also provided in [105].

However, from a modelling point of view, we can not end up with closed form analytical solutions for most of these challenging problems. This is because of the the complex shapes of the domains we work with and the associated boundary conditions involved. Therefore, it is quite common to resort to approximate solution techniques and methods. The most notable of these modelling approaches for the structural-acoustic problems are the finite element method [78], [55], [17] and the boundary element method [25], [26]. Boundary element method is generally used for external pressure field predictions. It is a widely accepted practice to use the finite element method for elastic structures interacting with enclosed cavities. We follow this practice and use the finite element method in this thesis. We assume the displacements and the pressure fluctuations due to this coupling is small so we use the linear field assumptions in both of the domains.

The number of degrees of freedom (DOFs) of the finite element models, namely, the size of the matrices, resulting from a structural-acoustic problem is likely to be very large. Generally, the complexity of the domains and the increase in the upper frequency limit are the reasons of the substantial increase in the number of DOFs for a correct representation [138].

Generally, the resulting large matrices are used to perform,

- A frequency sweep analysis, where we have to solve a linear system of equations of the form $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ at different discrete frequencies with a fine enough resolution to represent the Frequency Response Functions (FRFs). In this analysis, $\mathbf{A}$ is a frequency-dependent dynamic stiffness matrix [138] to be factorized for the solution of the linear system at each frequency step which is a costly numerical operation [62],

- An eigenvalue analysis to determine the major dynamic system parameters of the coupled system representation.

The solution of these problems also require excessive amount of storage space even if the matrices involved are kept in sparse format [134], [136]. Moreover, as shortly mentioned above, the problems arising from structural-acoustic coupling result in nonsymmetric matrices [105] which require nonsymmetric solvers during the above mentioned numerical operations. Generally, numerical solution costs of system equations involving nonsymmetric matrices are higher than their symmetric counterparts [62].

Another challenge in analyzing systems with these large matrices becomes even more important if the designers need to perform many simulation and optimization cycles, which is most of the time the case. In this respect, investigations of model order reduction (MOR) techniques are also important for this class of problems. With these techniques, the initial model sizes can be substantially decreased and the resulting systems can be analyzed quicker with an acceptable accuracy. Several approaches exist in the structural dynamics field since 60s and 70s [37], [100], [132]. In these approximations, the complete problem domain is divided into smaller domains and the reduction is performed independently on each of these domains, namely, the *component* domains. Later on, these domains are coupled to approximate the global system dynamics [90]. This practice results in significant reductions on the sizes of the matrices both on the component level and on the coupled system level. Moreover, the analysis on these components can be performed in a parallel nature as well with the best fitting methods. This approach has several advantages over solving the complete problem representation. Namely, instead of operating on the large system matrices of the complete problem, we perform the numerical operations on smaller matrices which can reduce the computational costs and time significantly.

To solve structural-acoustic problems, numerical analysts and engineers need to develop computationally efficient and reliable solution techniques. Therefore, improvement of the already existing methods is an active research area. And, in this thesis, the main focus is on the development of *new efficient solution and reduction techniques for structural-acoustic* problems.

1.3. Objectives and outline

Given the scope and motivation for the development of efficient numerical solution techniques for structural-acoustic problems, the research objectives of this thesis can be summarized as follows:

- *Investigate the current eigenvalue solution techniques used for structural-acoustic problems. Subsequently, improve these techniques and/or propose new numerical solution techniques for these problems,*

- *Investigate the currently applied model order reduction(MOR) techniques and develop new techniques for structural-acoustic problems,*

- *Develop numerical analysis tools which can be used to analyze industrial scale structural-acoustic problems.*

To achieve the above mentioned goals, this thesis is divided into two parts. Namely, the first part of the thesis focuses on developing eigenvalue solution techniques for structural-acoustic problems and it constitutes the research aimed at improving the already available solution techniques. This part constitutes Chapters 3 and 4. The second part of the thesis focuses on developing a component mode synthesis[1] approach for structural-acoustic problems. This part constitutes Chapters 5 and 6.

A brief outline of the contents is given as follows:

Chapter 2 introduces the main equations for the structural and acoustic domains along with the coupling relations and conditions between these two domains. The discretization process, which results in the nonsymmetric coupled system matrices, is also outlined in this chapter. The equations and the theory introduced in this chapter forms the basis of the thesis.

In **Chapter 3**, we detail the development of a Subspace like solver for the solution of the eigenvalue problems resulting from structural-acoustic problems. Starting from the uncoupled mode shape vectors, an iterative scheme is detailed which improves these initial vectors to end up with the coupled system eigenvalue/vector approximations.

Chapter 4 covers the development of a new symmetric solution technique for the eigenvalue problems arising from structural-acoustic coupling. This chapter also includes short information on the available solution techniques and, subsequently, the improved version of the symmetric solver is detailed and benchmarked against the current solvers available in the literature.

Chapter 5 proposes a new Craig-Bampton type reduction technique for structural-acoustic problems. This can also be viewed as an extension of the research presented in **Chapter 4**. It is centralized around the idea of building symmetric reduced component matrices in a numerically efficient manner.

Chapter 6 covers the extension of the technique presented in **Chapter 5** for components involving singularities which occur during the transformation of the component matrices to symmetric formats. Moreover, a generalized primal assembly framework is also presented for general substructuring applications for systems involving structural-acoustic coupling.

Chapter 7 summarizes the conclusions of the research performed in this thesis and outlines some research areas for possible future investigations.

1.4. Contributions of the thesis

In the view of the above mentioned objectives, we propose several scientific contributions in this thesis which can be listed as follows:

1. An iterative correction algorithm based on the well-known Subspace iteration technique is developed [148], [150].

---

[1] Substructuring

2. For the original non-symmetric eigenvalue problem resulting from the coupled structural-acoustic problems, we proposed a numerically efficient symmetric Lanczos eigenvalue solver [151], namely, *vibro*-Lanczos. The solver also makes use of a *partial-orthogonalization* scheme to decrease the numerical cost of the full-orthogonalization operations used in Lanczos iterations.

3. A numerically efficient and general Craig-Bampton (CB) type reduction technique is proposed for vibro-acoustic problems, namely, *vibro*-LsCB. The technique is built around the central idea of creating symmetric reduced component matrices and performing the operations on the component level with symmetric solvers. Another journal manuscript is under preparation based on this research.

4. Although not directly presented and used for structural-acoustic problems, different MOR techniques are reviewed [23] and compared. These different techniques mainly originate from the following fields: structural dynamics, systems and control and mathematics. The resulting research is also presented in Appendix A.

5. A powerful and easy to use numerical platform has been developed both in MATLAB and C++ with functional and object-oriented programming approaches. This platform interfaces with the commerial finite element code ANSYS to extract system and/or element matrices, node, connectivity information from the finite element models built in ANSYS. It is also used to interface the component related information used for model reduction approaches developed in this thesis. As a result, the developed numerical platform is capable of analyzing both academic and industrial problems efficiently either in MATLAB or in C++.

2

Vibro-acoustic Modelling

In this chapter, we investigate the analysis of interactions between flexible structural domains which are in contact with enclosed fluid domains. Governing equations of the two domains are presented along with the mathematical details of the interface coupling. Later on, discretized forms of the coupled problem representation is derived in a finite element framework. This chapter is intended to form a basis for the rest of the thesis.

2.1. Introduction

Generally, the fluid surrounding a structure has an influence on the behaviour of the structure and similarly the pressure waves inside the fluid domain is also influenced by the displacement field of the structure. In general, it is best to analyze this bidirectional interaction simultaneously. Nowadays, it is quite common to use advanced computational tools in order to predict noise levels radiated from different engineering structures such as fans or loudspeaker systems in free space. Moreover, similar tools can also become quite useful for the predictions performed for closed cavities which are interacting with flexible structural walls.

An overview and review of different numerical techniques for low frequency applications can be found in [11]. Namely, two main mathematical methods for modelling acoustic and vibro-acoustic problems are discussed, namely, the finite element method [64, 34, 123, 109] and the boundary element method [16, 26, 31]. Generally, finite element method is preferred for internal acoustic problems, whereas boundary element method is the preferred method for external radiation problems. An overview and comparison of these methods from a numerical perspective is provided in [40].

The formulation of coupled structural-acoustic problems using the finite element method is described thoroughly in [105]. In the structural domain, the main variable is the displacement. For the fluid domain, several different variables can be used, namely, displacement, pressure and the fluid displacement potential. [30, 2] are references where the fluid is represented in terms of displacement formulation. Although the fluid does not carry the shear loads, normal modes with rotational motion are introduced. Several techniques to make the eigenvalues of these rotational modes equal to zero were proposed therein along with some techniques based on reduced integration schemes.

By using the pressure formulation in the acoustic domain, the problem with the non-physical modes can be circumvented. Pressure based finite element formulation is used in [34, 123] to solve the eigenvalue problems related to complex shaped three dimensional cavities. The advantage of these two field formulations is that the fluid variable, pressure, is a scalar. Therefore, each node of the finite element model has only one degree of freedom in the fluid domain. Moreover, pressure is a variable which can be measured for the validation of the numerical method being used. Unfortunately, the resulting system of equations are non-symmetric. In effect, solvers that can handle non-symmetric matrices must be used in order to perform the solution of the system equations which is a drawback. However, some authors proposed ways to create symmetric system matrices by using two different variables in the fluid domain resulting in a three field representation with the displacement variable of the structural domain. Namely, in the fluid domain, pressure and fluid displacement potential [68] are used. However, this practice results in doubling the number of degrees of freedom in the fluid domain which substantially increases the numerical cost of the operations. In this thesis, we performed the investigations by using the displacement-pressure formulation using the finite element method.

In this chapter, basic principles of vibro-acoustic modelling for interior problems is introduced where the mutual interaction between the structure and the fluid is also taken into account. First, the governing equations[1] in the structural and acoustic domains are outlined, respectively. Subsequently, the mathematical details of the vibro-acoustic coupling conditions are provided. In order to solve these kinds of problems computationally, the corresponding strong forms of the differential equations are transformed into their weak forms. Then, the weak form is discretised with the finite element method leading to a coupled system of algebraic equations which can be solved numerically. The mathematical details of the discretisation process is also shortly derived in this chapter. This chapter is organized as follows: Section 2.2 briefly outlines the structural equations and the finite element discretization process for the structural domain. In a similar manner, Section 2.3 outlines the same process performed over the acoustic domain. Section 2.4 briefly explains the coupling conditions and provides some insight into the properties of the resulting system equations.

2.2. Structural domain and discretization

With the assumption of small displacements and following the Newton's second law of motion, the differential equation[2] of a structure under a body load $\boldsymbol{b}_s$ is written in matrix notation [116, 55] as follows:

$$\nabla_s^{\mathrm{T}} \boldsymbol{\sigma}_s + \boldsymbol{b}_s = \rho_s \ddot{\boldsymbol{u}} \tag{2.1}$$

where $\boldsymbol{u}$ represents the displacement field vector, $\boldsymbol{b}_s$ represents the body load vector. The right hand side in (2.1) represents the inertia force of the structure with ρ_s as

---

[1] Strong forms of the equations.

[2] The balance equation

the density of the material. Namely,

$$\boldsymbol{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}, \quad \boldsymbol{b}_s = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \tag{2.2}$$

The differential operator in (2.1) is given as,

$$\nabla_s = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{bmatrix} \tag{2.3}$$

In matrix notation, the stresses and strains are defined as follows:

$$\boldsymbol{\sigma}_s = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yz} \end{bmatrix}, \quad \boldsymbol{\varepsilon}_s = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{bmatrix}. \tag{2.4}$$

with $\gamma_{xy} = 2\varepsilon_{xy}$, $\gamma_{xzz} = 2\varepsilon_{xz}$ and $\gamma_{yz} = 2\varepsilon_{yz}$. With these definitions, the kinematic relation between the strains and the displacements is written as

$$\boldsymbol{\varepsilon}_s = \nabla_s \boldsymbol{u} \tag{2.5}$$

Similarly, for an isotropic linear material, the strains and the stresses are related with the constitutive equation, namely,

$$\boldsymbol{\sigma}_s = \mathbf{D}_s \boldsymbol{\varepsilon}_s \tag{2.6}$$

where $\mathbf{D}_s$ represents the constitutive matrix for an isotropic material [154].

To arrive at the discretized form of the equations, a weak form of the differential equation is constructed by multiplying (2.1) by different test functions in the three different directions in space. Namely, the arbitrary weight functions can be written as $\mathbf{v} = [v_x, v_y, v_z]^{\mathrm{T}}$.

One can multiply the first balance equation in (2.1) by the weight function, v_x, and integrate over the domain to arrive at the weak form of the differential equation[3], namely,

$$\int_{V_s} v_x \left(\nabla \cdot \hat{\boldsymbol{\sigma}}_x + \boldsymbol{b}_x - \rho_s \ddot{u}_x \right) dV = 0 \tag{2.7}$$

---

[3] It is important to note that the balance equation in different directions can also be written with the help of divergence and that property was used here.

with $\hat{\boldsymbol{\sigma}}_x = [\sigma_{xx}, \sigma_{xy}, \sigma_{xz}]^{\mathrm{T}}$. Namely, $\hat{\boldsymbol{\sigma}}_x$ is the one of the three traction vectors on an infinitesimal element [154].

The first term in (2.7) can be transformed by using the Green-Gauss theorem [55] resulting in

$$\int_{S_s} v_x \hat{\sigma}_x^{\mathrm{T}} \boldsymbol{n}_s dS - \int_{V_s} (\nabla v_x)^{\mathrm{T}} \hat{\sigma}_x dV + \int_{V_s} v_x \boldsymbol{b}_x dv - \int_{V_s} v_x \rho_s \ddot{u}_x dV = 0 \tag{2.8}$$

where $\boldsymbol{n}_s$ represents the normal vector pointing out from the structural domain written as $\boldsymbol{n}_s = [n_{s,x}, n_{s,y}, n_{s,z}]^{\mathrm{T}}$.

If we perform similar operations in the other directions and add up the resulting three equations, we can write the weak form of the structural differential equations in matrix format as follows, namely,

$$\int_{V_s} (\nabla_s \mathbf{v})^{\mathrm{T}} \boldsymbol{\sigma}_s dV + \int_{V_s} \mathbf{v}^{\mathrm{T}} \rho_s \ddot{\boldsymbol{u}} dV - \int_{S_s} \mathbf{v}^{\mathrm{T}} \boldsymbol{t}_s dS - \int_{V_s} \mathbf{v}^{\mathrm{T}} \boldsymbol{b} dV = 0 \tag{2.9}$$

It is important to note, when transforming from (2.8) to (2.9), the equilibrium between the forces on the surfaces of the domain and the internal forces expressed by the stress tensor requires that $\boldsymbol{t}_s = \mathbf{S_s} \boldsymbol{n}_s$. In this representation, the stress tensor $\mathbf{S}_s$ is defined as

$$\mathbf{S}_s = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \tag{2.10}$$

If the structural domain is discretized using suitable elements then the integrations in (2.9) must be performed over the element domains and the contributions of each element must be added to end up with the system matrices [17].

The approximation of the displacements and the weight functions over an element is introduced respectively by using the shape functions

$$\boldsymbol{u} = \mathbf{N}_s \boldsymbol{u}_s, \quad \mathbf{v}_s = \mathbf{N}_s \boldsymbol{c}_s \tag{2.11}$$

By using (2.5), the strains can be expressed as

$$\varepsilon_s = \nabla_s \mathbf{N}_s \boldsymbol{u}_s \tag{2.12}$$

Substituting (2.6) and (2.12) in (2.9), we can write the finite element formulation for the structural domain as follows,

$$\mathbf{M}_s \ddot{\boldsymbol{u}}_s + \mathbf{K}_s \boldsymbol{u}_s = \boldsymbol{f}_s + \boldsymbol{f}_b \tag{2.13}$$

where

$$\mathbf{K}_s = \sum_e \int_{V_s^{(e)}} \left(\nabla_s \mathbf{N}_s\right)^{\mathrm{T}} \mathbf{D}_s \nabla_s \mathbf{N}_s dV \tag{2.14a}$$

$$, \mathbf{M}_s = \sum_e \int_{V_s^{(e)}} \rho_s \mathbf{N}_s^{\mathrm{T}} \mathbf{N}_s dV \tag{2.14b}$$

$$\boldsymbol{f}_s = \sum_{e_{sf}} \int_{S_{sf}^{(e)}} \mathbf{N}_s^{\mathrm{T}} \boldsymbol{t}_s dS \tag{2.14c}$$

$$\boldsymbol{f}_b = \sum_e \int_{V_s^{(e)}} \mathbf{N}_s^{\mathrm{T}} \boldsymbol{b} dV \tag{2.14d}$$

This approach is called the Galerkin discretization of the weak form [33]. In a finite element framework, the integral expressions of the Galerkin weak-form, provided in (2.9), are converted to integrals over the discretized element domains. Subsequently, with a mapping between the local and global degrees of freedom, the results of above integration operations, which are the element matrices, are then assembled into the system matrices of the domain under consideration [55, 116].

In (2.14), $\sum_e$ represents the assembly of the local element matrices or element force vectors over the domain. Similarly, $\sum_{e_{sf}}$ represents the assembly over the interface elements. Moreover, it is important to note that, for $\boldsymbol{f}_s$, we only have to consider the elements where the traction vector $\boldsymbol{t}_s$ is defined. In the context of vibro-acoustic problems studied in this thesis, this interface is the surface between the structural and acoustic domain, S_{sf}.

2.3. Acoustic domain and discretization

In this thesis, we assume an inviscid[4], compressible and irrotational fluid which undergoes small translations only. Moreover, linear stress-strain laws are assumed. Fluid is an ideal gas without any mean flow[5]. In all the mathematical derivations, perturbations associated with the fluid domain are very small so that their products can be neglected during the mathematical operations.

In the current context, Helmholtz equation is used to describe the motion of the acoustic domain [106]. The main variable is the excess pressure, p, which represents the small fluctuations in pressure around the mean pressure of the medium. Namely,

$$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0 \tag{2.15}$$

where c is the speed of sound in the fluid domain. Derivation of (2.15) can be performed by using the linearized continuity and linearized Euler equations along with the state of stress of an ideal gas. The details can be consulted in well documented references [106, 88, 81]. In this thesis, we did not consider any external sources in (2.15).

---

[4] No viscous effects are considered.

[5] There is no ambient flow.

In order to develop the finite element formulation in the acoustic domain, (2.15) is multiplied by a test function, v_f, and integrated over the volume of the fluid, namely,

$$\int_{V_f} v_f \left(\nabla^2 p - \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \right) dV = 0 \tag{2.16}$$

Since $\nabla^2 p = \nabla \cdot (\nabla p)$ and by using the Green-Gauss theorem [55], we can write the first term in (2.16) by using the matrix notation as follows,

$$\int_{V_f} v_f \nabla \cdot (\nabla p) dV = \int_S v_f (\nabla p)^{\mathrm{T}} \boldsymbol{n}_f dS - \int_{V_f} (\nabla v_f)^{\mathrm{T}} (\nabla p) dV = 0 \tag{2.17}$$

where the normal vector, $\boldsymbol{n}_f$, represents the normal that points out from the fluid domain. Using (2.17), we can get to the weak form of the equations, namely,

$$\int_{V_f} (\nabla v_f)^{\mathrm{T}} (\nabla p) dV + \frac{1}{c^2} \int_{V_f} v_f \frac{\partial^2 p}{\partial t^2} dV - \int_S v_f (\nabla p)^{\mathrm{T}} \boldsymbol{n}_f dS = 0 \tag{2.18}$$

2

Similar to the structural domain, when the fluid domain, V_f, is discretized, acoustic pressure variable, p, and the arbitrary test function, v_f, over an element are written as follows, namely,

$$p = \mathbf{N}_f \boldsymbol{p}_f, \quad v_f = \mathbf{N}_f \boldsymbol{c_f} \tag{2.19}$$

where $\boldsymbol{p}_f$ contains the nodal pressures, and $\boldsymbol{c}_f$ are the arbitrary nodal weight constants.

Following the briefly outlined scheme and using the shape function representations of the field variable over the elements, namely (2.19), (2.18) can be put into the discretized format, namely,

$$\mathbf{M}_f \ddot{\boldsymbol{p}} + \mathbf{K}_f \boldsymbol{p} - \boldsymbol{f}_f = \mathbf{0} \tag{2.20}$$

where

$$\mathbf{M}_f = \sum_e \frac{1}{c^2} \int_{V_f^{(e)}} \mathbf{N}_f^{\mathrm{T}} \mathbf{N}_f dV, \tag{2.21a}$$

$$\mathbf{K}_f = \sum_e \int_{V_f^{(e)}} (\nabla \mathbf{N}_f)^{\mathrm{T}} (\nabla \mathbf{N}_f) dV, \tag{2.21b}$$

$$\boldsymbol{f}_f = \sum_{e_{e_{sf}}} \int_{S_{sf}^{(e)}} \mathbf{N}_f^{\mathrm{T}} \boldsymbol{n}_f^{\mathrm{T}} \nabla p dS. \tag{2.21c}$$

In (2.21), $\sum_e$ and $\sum_{e_{sf}}$ represent the assembly of the local element matrices over the domain and over the interface, respectively. It is important to note that the acoustic stiffness matrix, $\mathbf{K}_f$, is related to the acoustic kinetic energy. Similarly, the acoustic mass matrix, $\mathbf{M}_f$ is related to the acoustic potential energy [48]. (2.21c) represents the interface coupling terms which provides the effect of the structure on the fluid due to the pressure gradient on the interface. A similar coupling was also considered for the structural domain resulting in a mutual interface coupling over S_{sf} between the two domains.

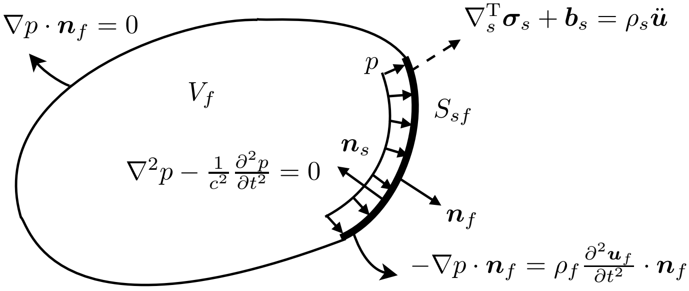

$\nabla p \cdot \boldsymbol{n}_f = 0$
$\nabla_s^{\mathrm{T}} \boldsymbol{\sigma}_s + \boldsymbol{b}_s = \rho_s \ddot{\boldsymbol{u}}$
p
V_f
S_{sf}
$\boldsymbol{n}_s$
$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0$
$\boldsymbol{n}_f$
$-\nabla p \cdot \boldsymbol{n}_f = \rho_f \frac{\partial^2 \boldsymbol{u}_f}{\partial t^2} \cdot \boldsymbol{n}_f$

Figure 2.1: Simple structure coupled to an acoustic domain on the interface S_{sf}

2.4. Coupling conditions, coupled equations and properties

In vibro-acoustic problems, the coupling between the structural and acoustic domain is accomplished over the interface between the structure and the coupling fluid. In most practical engineering problems, this bidirectional coupling can not be neglected and must be taken into account. Simple mechanics dictates two conditions for this coupling to hold on the interface. Namely, over the interface, the normal velocities on the structural and acoustic domains must match. Second, the equilibrium relations must hold over the interfaces.

A simple coupled problem including the structural and acoustic domains is depicted in Figure 2.1 with the coupling surface, S_{sf}. Additionally, the considered boundary conditions are also shown in this figure. Pressure acting on the structural domain is also shown in the same figure[6]. The corresponding normal vectors of the coupling domains are also shown therein for clarity. It is important to note that the structural normal vector, $\boldsymbol{n}_s$, points outwards from the structural domain. Similarly, the normal vector of the acoustic domain also points outwards from the fluid domain. This convention results in the following relation between the normal vectors, namely,

$$\boldsymbol{n}_f = -\boldsymbol{n}_s \tag{2.22}$$

The linearized Euler equation reads as

$$-\nabla p = \rho_f \frac{\partial \boldsymbol{v}_f}{\partial t} \tag{2.23}$$

where $\boldsymbol{v}_f$ is the velocity of the fluid particles and ρ_f represents the density of the fluid that is in contact with the structural domain. If we use the continuity of the velocities on the interface in the normal direction, namely, $\boldsymbol{v}_s = \boldsymbol{v}_f$, we can write (2.23) as follows,

$$-\nabla p \cdot \boldsymbol{n}_f = \rho_f \frac{\partial \boldsymbol{v}_s}{\partial t} \cdot \boldsymbol{n}_f \equiv \rho_f \frac{\partial^2 \boldsymbol{u}_s}{\partial t^2} \cdot \boldsymbol{n}_f \tag{2.24}$$

---

[6] No other tractions apply on the structural domain except the pressure of the fluid domain

(2.24) can be also be interpreted as a boundary condition for the Helmholtz equation given in (2.15). Physically, the displacement or velocity represents the boundary condition for the fluid domain.

Second condition results from the relation between the tractions on the interface[7], namely,

$$\mathbf{S}_s \boldsymbol{n}_s \equiv \boldsymbol{t}_s = -p \boldsymbol{n}_s \tag{2.25}$$

where $\boldsymbol{t}_s$ represents the traction[8] on the structural domain due to the acoustic domain. This traction represents the boundary condition on the structural domain. As a result, both of the coupling conditions (2.24) and (2.25) have to be satisfied in the case of a coupled problem representation since the effect of the structure on the fluid and vice versa can not be neglected.

In Sections 2.2 and 2.3, the discretization of the structural traction vector was provided in (2.14c) and, similarly, the discretization of the source term was provided in (2.21c). By using the conditions outlined for coupling, namely, (2.24) and (2.25), we can convert (2.14c) as follows, namely,

$$\boldsymbol{f}_s = \sum_{e_{sf}} \int_{S_{sf}^{(e)}} \mathbf{N}_s^{\mathrm{T}} \boldsymbol{t}_s dS \xrightarrow{(2.22),\ (2.25)} \boldsymbol{f}_s = \underbrace{\left(\sum_{e_{sf}} \int_{S_{sf}^{(e)}} \mathbf{N}_s^{\mathrm{T}} \boldsymbol{n}_f \mathbf{N}_f dS \right)}_{\mathbf{K}_{sf}} \boldsymbol{p}_f \tag{2.26a}$$

$$\boldsymbol{f}_s = \mathbf{K}_{sf} \boldsymbol{p}_f \tag{2.26b}$$

Similarly, we can also rewrite (2.21c), namely,

$$\boldsymbol{f}_f = \sum_{e} \int_{S_{sf}^{(e)}} \mathbf{N}_f^{\mathrm{T}} \boldsymbol{n}_f^{\mathrm{T}} \nabla p dS \xrightarrow{(2.24)} \boldsymbol{f}_f = -\rho_f \underbrace{\left(\sum_{e_{sf}} \int_{S_{sf}^{(e)}} \mathbf{N}_f^{\mathrm{T}} \boldsymbol{n}_f^{\mathrm{T}} \mathbf{N}_s dS \right)}_{\mathbf{M}_{fs}} \ddot{\boldsymbol{u}}_s \tag{2.27a}$$

$$\boldsymbol{f}_f = -\mathbf{M}_{fs} \ddot{\boldsymbol{u}}_s \tag{2.27b}$$

In these transformations, we also used the pressure and displacement field representations, namely, $p = \mathbf{N}_f \boldsymbol{p}_f$ and $\boldsymbol{u} = \mathbf{N}_s \boldsymbol{u}_s$ in (2.26a) and (2.27a), respectively.

In the light of the above discussion, we can combine the field equations provided in (2.13) and (2.20) to end with the coupled system equations of the vibro-acoustic problem, namely,

$$\underbrace{\begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{M}_{fs} & \mathbf{M}_f \end{bmatrix}}_{\mathbf{M}_c} \begin{bmatrix} \ddot{\boldsymbol{u}}_s \\ \ddot{\boldsymbol{p}}_f \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{K}_s & -\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}}_{\mathbf{K}_c} \begin{bmatrix} \boldsymbol{u}_s \\ \boldsymbol{p}_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_b \\ \mathbf{0} \end{bmatrix} \tag{2.28}$$

---

[7] Due to Newtons's third law of motion, namely, action and reaction principle

[8] pressure

where $\mathbf{M}_c$ and $\mathbf{K}_c$ represent the coupled system mass and stiffness matrices of the vibro-acoustic problem, respectively. $\mathbf{K}_{sf}$ and $\mathbf{M}_{sf}$ in (2.28) represent the, generally, rectangular interface coupling blocks for the problem. Moreover, investigating the final equations provided in (2.26a) and (2.27a) closer, we can also see that $\mathbf{M}_{fs} = \rho_f \mathbf{K}_{sf}^{\mathrm{T}}$. We can also simplify (2.28) further by scaling the second block row with ρ_f. Namely,

$$\underbrace{\begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f \end{bmatrix}}_{\mathbf{M}_c} \begin{bmatrix} \ddot{\boldsymbol{u}}_s \\ \ddot{\boldsymbol{p}}_f \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{K}_s & -\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}}_{\mathbf{K}_c} \begin{bmatrix} \boldsymbol{u}_s \\ \boldsymbol{p}_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_b \\ \mathbf{0} \end{bmatrix} \tag{2.29}$$

where $\mathbf{K}_f \leftarrow \mathbf{K}_f/\rho_f$ and $\mathbf{M}_f \leftarrow \mathbf{M}_f/\rho_f$. It is important to note that in (2.29), the source terms related to the acoustic partition is not there. However, if we would like to perform a forced response analysis with the source terms, then we must also perform the scaling on the source side as well. To conclude, (2.29) represents the linearized, undamped coupled equations of motion of the vibro-acoustic systems to be investigated in the context of this thesis.

By selecting displacement and pressure as our main variables in the structural and fluid domains, respectively, we have shown that we end up with the nonsymmetric coupled system in (2.29). This is a two field formulation which have $(m_s + m_f)$ degrees of freedom in total where m_s and m_f represent the size of the structural and acoustic partitions, respectively. Some other approaches were proposed that use the acoustic velocity potential as the field variable in the acoustic domain, namely, [46],[140]. By using this choice, one can still end up with symmetric mass and stiffness matrices in a two field representation which results in $(m_s + m_f)$ degrees of freedom in total. Unfortunately, the coupling conditions results in a skew-symmetric coupling matrix which is considered as the damping matrix in this approach. Moreover, as shortly mentioned in the introduction, several other methods to end up with symmetric system matrices also exist and documented in [105, 68]. A common down side of these approaches is the substantial increase in the total number of degrees of freedom of the systems. Namely, the use of the air particle displacement potential as mentioned in [68] results in a mixed formulation in the acoustic domain which results in an increase in the total system size. As a result, the size of the resulting symmetric matrices are of the order of $(m_s + 2m_f)$ in comparison to the $(m_s + m_f)$ of the above mentioned size of the two field formulation. Therefore, we base our selection of the *displacement-pressure* formulation on the shortly outlined justifications.

Apart from the above mentioned points, it is good to add some additional comments on the *displacement-pressure* formulation, namely,

- The system given in (2.29) is composed of two different physics of two different orders, namely, the displacement and pressure. This practice results in an overall system that is ill-conditioned which is a serious drawback for the numerical operations to be performed. Therefore, scaling of the system matrices are proposed by several authors [102] in the context of substructuring. In our numerical tests, we generally applied a simple diagonal scaling on the matrices.

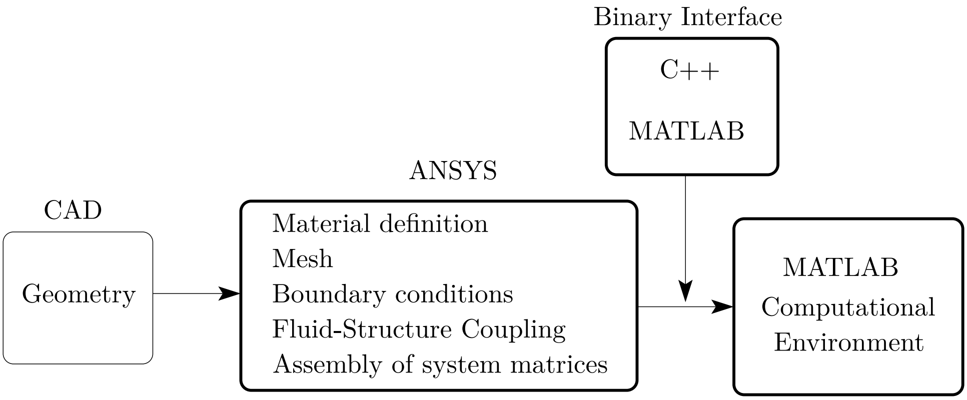


Figure 2.2: Computational platform

- Even if the coupled system equations are nonsymmetric, the matrices that build up the coupled system are all symmetric except the off-diagonal coupling blocks.

- Nonsymmetric eigenvalue problems might result in eigenvalues and eigenvectors in complex conjugate pairs [91] in contrast to real symmetric eigenvalue problems. However, it can be shown due to the relations between the off-diagonal coupling blocks, (2.29) results in real eigenvalues and eigenvectors in the absence of damping [161].

- To perform the modal decoupling operations similar to the ones performed for the symmetric system matrices [63], we need the left eigenvectors of the problem. Left eigenvectors, $\boldsymbol{\phi}_l$, originate from the from left eigenvalue problem $\left(\mathbf{K}_c^{\mathrm{T}} - \omega^2 \mathbf{M}_c^{\mathrm{T}}\right) \boldsymbol{\phi}_l = \mathbf{0}$. By using the left eigenvectors and performing the mass normalization, we can write

$$\boldsymbol{\phi}_l^{\mathrm{T}} \mathbf{M}_c \boldsymbol{\phi}_r = \mathbf{I} \tag{2.30a}$$

$$\boldsymbol{\phi}_l^{\mathrm{T}} \mathbf{K}_c \boldsymbol{\phi}_r = \mathbf{diag}(\omega_i^2) \tag{2.30b}$$

  where $\boldsymbol{\phi}_r$ are the right eigenvectors originating from (2.29). Moreover, the left and right eigenvectors of (2.29) are related as proven in Appendix E.

2.5. Computational platform

This short section describes the simulation platform used for the numerical tests performed in this thesis. The operations performed with the platform is summarized in the flowchart presented in Figure 2.2. The main component of the platform is ANSYS [6] which is one of the commercial finite element codes in the market. The input to the platform is the model geometry which can either be processed by different CAD software or can be built up by the preprocessor of ANSYS itself. The usual finite element modelling steps are carried out in ANSYS through APDL, namely, *ANSYS Parametric Design Language*. Besides, we mainly use ANSYS as a preprocessor for finite element operations such as matrix assembly, application of boundary conditions and etc. Subsequently, we extract the information from the output binary files written by ANSYS with the help of developed interfaces. Namely, we developed two interfaces in MATLAB (MathWorks, Natick, MA, USA) and in C++ [145] which can handle the models prepared in ANSYS efficiently. Eventually, we mostly used MATLAB as our main computational platform.

2.6. Summary

In this chapter, we presented the governing equations of the coupled vibro-acoustic problem. Details of the discretization with finite element method was also shown along with the coupling formulation. Moverover, a very brief overview on the computational framework used throughout the thesis was also provided. This short chapter forms a basis for the thesis.

I

Investigations on Eigenvalue Solution Strategies for Vibro-Acoustic Simulations

3

Iterative Reduced Correction Algorithm, *IRCA*

In this chapter, a Subspace-like eigenvalue solver is investigated for the coupled eigenvalue problems resulting from vibro-acoustics. A projection space similar to the one used in the Subspace Iteration method is built up iteratively and used. This basis includes the corrections from the mutually coupling physics. Moreover, it is also enriched with extra residual vectors. By relating the inverse iteration step with a symmetric scaling transformation along with the use of residual vectors, some proofs on improving convergence are provided. The research presented in this chapter is partially based on the conference articles [148, 150].

3.1. Introduction

In Chapter 2, we provided the derivation of the main equations that govern the vibro-acoustic problems. We also mentioned that the problem can be written in different formats by selecting different variables to represent acoustic domain. Solution of the coupled problem representation is almost always the best option. Moreover, it might be the only option whenever the impedances of the two coupling domains match which eventually results in a strongly coupled problem. However, solution of the fully coupled problem is generally time consuming. In addition, the numerical conditioning of the coupled system matrices can introduce more severe challenges for the solution techniques.

Due to the briefly outlined difficulties, cheaper indirect solution approaches were proposed which are all based on the modal interaction models presented in [158], [48]. Generally in these methods, the structural nodal displacements are represented in terms of a summation over the *in vacuo* normal modes of the structural domain. In a similar manner, the nodal acoustic pressures are represented in terms of a summation over the *rigid wall* acoustic pressure normal modes. But, as mentioned shortly before, the coupling impedances on the interacting surfaces can match when a heavy fluid is used. In this case, the use of the modal interaction model generally fails to provide accurate estimates of the eigenvalues and eigenvectors.

In the literature, different techniques were studied and documented which use the *in vacuo* normal modes of the structural domain and the *rigid wall* cavity modes of the subsystems, namely, the structural and the fluid domains. These are called the decoupled system modes in the presentation from now on. In [107], the authors used the decoupled system modes and, later on, with some algebraic manipulations, formulated the eigenvalue and frequency response problems in a symmetric manner. By using a relatively high number of decoupled mode vectors, they ended up with results that have fair accuracy while air was used as the coupling fluid. [137] also uses the decoupled system modes and outlines an algebraic approach to write the system representation in a symmetric format by using either the displacement-pressure($\boldsymbol{u}-\boldsymbol{p}$) formulation or the displacement-fluid displacement potential($\boldsymbol{u}-\boldsymbol{\Psi}$) formulation. That author also used the relations between the left and right eigenvectors while building up the modal interaction model. The author reported fair accuracy even by using a relatively high number of decoupled mode vectors from the acoustic domain. Later, [152] proposed to enrich the modal interaction model with pseudo-static correction vectors resulting from the structural and acoustic domains, respectively. As a result, they reported significant improvements on the system responses with the use of pseudo-static correction approach. [3, 4] combined the approaches proposed in [137] and [152] in order to come up with a symmetric representation for forced response calculations. [144] also starts from the decoupled system modes. They proposed a technique that uses some residual vectors in order to enrich the modal interaction model. In order to select important residual vectors, a two level vector selection approach is used. First, they performed an initial screening by using the off-diagonal coupling blocks of the coupled problem representation. Subsequently, they used the results of the previous step to perform a second selection by using some strain energy like norms. In this way, they tried to select the best vectors that will eventually appear in the modal interaction basis. To the authors knowledge, [24] proposed the only iterative technique that intends to improve the decoupled system modes without any convergence proofs.

In this chapter, our aim is to outline the development of a Subspace like iterative eigenvalue solution technique which improves the above mentioned decoupled system modes substantially in order to accurately predict the dynamic response of the fully coupled problems. This point is especially important for strongly coupled problems where most of the briefly outlined decoupled projection approaches fail. This chapter is organized as follows. In Section 3.2, we shortly outline the symmetric Subspace iteration technique to form a basis for the following discussions. Section 3.3 provides the details of the developed technique. Namely, it provides the mathematical details of the enrichment vector computations, projection and the resulting form of the interaction problem. In Section 3.4, a connection between the symmetric and nonsymmetric cases is established through a pre-multiplication matrix and a rather simple convergence proof is provided. Section 3.5 provides some practical considerations over the developed technique. Moderate sized academic test cases are presented in Section 3.6. Some discussion and conclusions are provided in Section 3.7.

3.2. Overview on Subspace iteration

Since the algorithm developed in this chapter is similar in nature to the Subspace iteration technique, we would like to briefly outline the Subspace iteration technique based on [63, 110] in order to form a basis for the following discussions [19].

Subspace iteration is useful in evaluating the dominant eigenvalues and eigenvectors of large-order generalized linear eigenvalue problems. Transformation to a standard linear eigenvalue problem format is not necessary and it can benefit from the full sparsity of the stiffness and mass matrices of the problem at hand. The process is iterative in nature and is mostly based on the power iteration method (or the inverse iteration method) where, in this case, several vectors are simultaneously iterated upon. It also uses the properties of the Rayleigh quotient in order to extract the best approximates from a selected reduction basis.

The eigenvectors $\mathbf{\Phi}$ of the generalized eigenvalue problem

$$\mathbf{K\Phi} = \mathbf{M\Phi\Omega}, \tag{3.1}$$

span an n-dimensional space. In this case, (3.1) is simultaneously iterated, as in the case of the inverse-iteration, with a set of p linearly independent vectors where $p \ll n$. Iterating over this p-dimensional subspace, the vectors are gradually improved so that it will eventually span the pth-order subspace (associated with the p dominant eigenvalues and eigenvectors) of the original n-dimensional space.

The iterative algorithm can be stated as follows. One starts from an initial block basis, $\mathbf{X}_0$, which is of size $n \times p$, and iterates over to improve the starting vectors in the initial basis. In other words, it is a block generalization of the inverse iteration method. The iteration equation is the same as the one in the inverse iteration but one iterates over a block of vectors, namely,

$$\mathbf{K}\tilde{\mathbf{X}}_{k+1} = \mathbf{M}\mathbf{X}_k. \tag{3.2}$$

At this point one solves for $\tilde{\mathbf{X}}_{k+1}$ and form the interaction problem of the subspace iteration, namely,

$$\tilde{\mathbf{K}} = \tilde{\mathbf{X}}_{k+1}^{\mathrm{T}}\mathbf{K}\tilde{\mathbf{X}}_{k+1}, \tag{3.3}$$

$$\tilde{\mathbf{M}} = \tilde{\mathbf{X}}_{k+1}^{\mathrm{T}}\mathbf{M}\tilde{\mathbf{X}}_{k+1}, \tag{3.4}$$

and subsequently solve the reduced eigenvalue problem of order $p \times p$. This problem is supposed to be small so that it can be solved with an eigenvalue solver that uses direct solution techniques. One prominent candidate is the QR algorithm which determines all the eigenvalues and eigenvectors of the reduced problem, namely,

$$\tilde{\mathbf{K}}\mathbf{V}_{k+1} = \tilde{\mathbf{M}}\mathbf{V}_{k+1}\tilde{\mathbf{\Omega}}_p. \tag{3.5}$$

At the next step, the improved vectors are recovered by

$$\mathbf{X}_{k+1} = \tilde{\mathbf{X}}_{k+1}\mathbf{V}_{k+1}. \tag{3.6}$$

At the end of iteration steps, the convergence of the iterate vectors are checked according to the error-measure criterion that is provided (assuming no rigid body mode vectors are present) as

$$\epsilon = \frac{\|\mathbf{K}\boldsymbol{X}_k - \lambda_k\mathbf{M}\boldsymbol{X}_k\|}{\|\mathbf{K}\boldsymbol{X}_k\|}. \tag{3.7}$$

As the number of iterations increase, the algorithm converges to the first p dimensional subspace of eigenvalues and eigenvectors. The steps of the algorithm is summarized in Algorithm 3.1.

Algorithm 3.1 Subspace iteration algorithm

```
Input matrices and tolerance, $\mathbf{K}$, $\mathbf{M}$ and $\epsilon$
Input initial block with a block size of $p$, $\mathbf{X}_1$
$E = 1$ and $k = 0$
while $E \geq \epsilon$ do
    // Inverse iteration and projection
    $\mathbf{Z}_k = \mathbf{M}\mathbf{X}_k$
    $\tilde{\mathbf{X}}_{k+1} = \mathbf{K}^{-1}\mathbf{Z}_k$
    $\tilde{\mathbf{K}} = \tilde{\mathbf{X}}_{k+1}^T \mathbf{K} \tilde{\mathbf{X}}_{k+1}$
    $\tilde{\mathbf{M}} = \tilde{\mathbf{X}}_{k+1}^T \mathbf{M} \tilde{\mathbf{X}}_{k+1}$
    // Solution of the interaction problem and recovery
    $\tilde{\mathbf{K}}\mathbf{V}_{k+1} = \tilde{\mathbf{M}}\mathbf{V}_{k+1}\tilde{\mathbf{\Omega}}_p$ where $\tilde{\mathbf{\Omega}}_p = diag([\lambda_1, \lambda_2, \ldots, \lambda_p])$
    $\mathbf{X}_{k+1} = \tilde{\mathbf{X}}_{k+1}\mathbf{V}_{k+1}$
    // Convergence check
    $E = \max\limits_{k=1,\ldots,p} \left( \frac{\|\mathbf{K}\boldsymbol{X}_k - \lambda_k \mathbf{M}\boldsymbol{X}_k\|}{\|\mathbf{K}\boldsymbol{X}_k\|} \right)$
end while
Output $\mathbf{X}_{k+1} \rightarrow \mathbf{\Phi}_p$, $\tilde{\mathbf{\Omega}}_p \rightarrow \mathbf{\Omega_p}$
```

3.3. Iterative reduced correction algorithm, *IRCA*

As pointed out in Section 3.2, Subspace iteration iterates over a block of vectors simultaneously in order to come up with the eigenvalues and eigenvectors in the low frequency band.

In this section, a similar algorithm will be presented for the solution of the coupled eigenvalue problems arising in vibro-acoustic problems. The name of the algorithm emerges from the idea that several correction and residual vectors are computed by taking into account the mutual coupling effects between the coupling physics. Our aim is to show that the correction vectors enriched with some residual vectors help us to end up with a Subspace like correction algorithm. First, we briefly outline the starting vectors in Section 3.3.1. Sections 3.3.2 and 3.3.3 provide the details about the computations of enrichment vectors which are the main ingredients of the projection subspace.

3.3.1. Modal interaction model and the starting basis

The starting basis used to initiate the complete iterative process is based on the modal interaction model which is composed of two independent vector bases [48]. Namely, we start with the fully coupled problem representation presented in (2.29) where we neglect the coupling between the structural and fluid domains to end with the decoupled eigenvalue problems. To summarize,

- The first component of the complete basis is composed of the *in vacuo* modes of the structural domain where the effect of the fluid domain is completely neglected. These *in vacuo* structural modes are computed from the structural eigenvalue problem, namely,

$$(\mathbf{K}_s - \lambda_s \mathbf{M}_s)\,\boldsymbol{\phi}_s = \mathbf{0}, \tag{3.8}$$

- The second component of the basis is composed of the *rigid wall* acoustic modes of the fluid domain. Similar to the above modes, the effect of the structure is neglected. The *rigid wall* acoustic modes are also calculated by solving the eigenvalue problem for the fluid domain alone, namely,

$$(\mathbf{K}_f - \lambda_f \mathbf{M}_f)\,\boldsymbol{\phi}_f = \mathbf{0}, \tag{3.9}$$

In the current implementations, the algorithm does not extract the complete converged set of modes of the structural and fluid domains in (3.8) and (3.9). Namely, Lanczos eigenvalue extraction method generally requires, at least, $3n$ to $4n$ iterations in order to build up a space that is rich enough to extract the first n eigenvectors that are sought [63]. In order to reduce the computational cost, it is proposed to perform a Lanczos run which consists a lower number of iterations than it would be needed to find the exact converged modes. Namely, if n vectors are to be used from either structural or fluid domains, then it is proposed to iterate n times which is equal to the number of the selected vectors for the target domain. It should be noted that these vectors are not the fully converged mode vectors of the Lanczos eigenvalue extraction algorithm. In other words, they originate from the tridiagonal eigenvalue problem used at the end of the Lanczos run with a lower number of iterations. These vectors are called the pseudo vectors in the rest of the discussion.

In the modal interaction model, the results of eigenvalue problems in (3.8) and (3.9), $\boldsymbol{\phi}_s$ and $\boldsymbol{\phi}_f$, are used for the representation of the main variables of the fully coupled problem representation. Namely, the displacements, $\boldsymbol{u}$, are formulated in terms of a summation over the *in vacuo* structural modes,

$$\boldsymbol{u} = \sum_{m=1}^{ns} \boldsymbol{\phi}_s \eta_s = \boldsymbol{\Phi}_s \boldsymbol{\eta}_s. \tag{3.10}$$

Similarly, the acoustic pressure variable of the fluid domain is represented as a summation over the *rigid wall* acoustic normal modes,

$$\boldsymbol{p} = \sum_{m=1}^{nf} \boldsymbol{\phi}_f \eta_f = \boldsymbol{\Phi}_f \boldsymbol{\eta}_f. \tag{3.11}$$

If the modal bases expressions from (3.10) and (3.11) are used to built up a projection space in a Rayleigh-Ritz sense, then the modal projection basis reads as,

$$\mathbf{T}_m = \begin{bmatrix} \boldsymbol{\Phi}_s & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_f \end{bmatrix}. \tag{3.12}$$

Projection of the coupled eigenvalue problem, (2.29), on this modal basis, $\mathbf{T}_m$, results in

$$\left(\underbrace{\begin{bmatrix}\boldsymbol{\Omega}_s & -\boldsymbol{\Phi}_s^{\mathrm{T}}\mathbf{K}_{sf}\boldsymbol{\Phi}_f \\ \mathbf{0} & \boldsymbol{\Omega}_f\end{bmatrix}}_{\tilde{\mathbf{k}}} - \tilde{\lambda}\underbrace{\begin{bmatrix}\mathbf{I}_s & \mathbf{0} \\ \boldsymbol{\Phi}_f^{\mathrm{T}}\mathbf{K}_{sf}^{\mathrm{T}}\boldsymbol{\Phi}_s & \mathbf{I}_f\end{bmatrix}}_{\tilde{\mathbf{m}}}\right)\begin{bmatrix}\tilde{\boldsymbol{u}} \\ \tilde{\boldsymbol{p}}\end{bmatrix} = \mathbf{0} \tag{3.13}$$

In order to find the starting vectors of the iterative process, we have to solve the reduced eigenvalue problem in (3.13) and recover the physical vectors with the help of a back transformation as follows

$$\begin{bmatrix}\tilde{\boldsymbol{\Phi}}_s \\ \tilde{\boldsymbol{\Phi}}_f\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Phi}_s & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_f\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{u}} \\ \tilde{\boldsymbol{p}}\end{bmatrix} \tag{3.14}$$

It is important to emphasize that we start the iterative process outlined in this Chapter by using a limited number of modes resulting from (3.14). Moreover, the actual selection criteria will be outlined in Section 3.5.1 with more details. In brief, by solving a reduced problem as given in (3.13), we compute a first rough approximation of the coupled modes of our problem.

3.3.2. Correction vectors and their computations

Considering the coupled problem representation in (2.29) without the force terms, one could separate the rows of that equation and write the equations as

$$(\mathbf{K}_s - \lambda\mathbf{M}_s)\,\boldsymbol{C}_s = \mathbf{K}_{sf}\boldsymbol{\phi}_f, \tag{3.15}$$

$$(\mathbf{K}_f - \lambda\mathbf{M}_f)\,\boldsymbol{C}_f = \lambda\mathbf{K}_{sf}^{\mathrm{T}}\boldsymbol{\phi}_s. \tag{3.16}$$

Due to the non-symmetric nature of the problem, (3.15) and (3.16) are coupled through the coupling blocks in the system mass and stiffness matrices. And these two equations are the driving idea behind the correction computations. Namely, to include the effect of mutually interacting physics as a force correction on its counterpart. Following, with these force corrections, we would like to correct and complement the missing information on the results of the modal interaction model in an iterative sense.

One starts by using the initial vectors that were generated in Section 3.3.1. Following, (3.15) and (3.16) are written, on a per mode basis, as

$$\left(\mathbf{K}_s - \tilde{\lambda}\mathbf{M}_s\right)\boldsymbol{C}_s = \mathbf{K}_{sf}\tilde{\boldsymbol{\phi}}_f, \tag{3.17}$$

$$\left(\mathbf{K}_f - \tilde{\lambda}\mathbf{M}_f\right)\boldsymbol{C}_f = \tilde{\lambda}\mathbf{K}_{sf}^{\mathrm{T}}\tilde{\boldsymbol{\phi}}_s, \tag{3.18}$$

where $\boldsymbol{C}_s$ and $\boldsymbol{C}_f$ represent the correction vectors. $\tilde{\boldsymbol{\phi}}_s$ and $\tilde{\boldsymbol{\phi}}_f$ represent the vectors in (3.14), namely, in $\tilde{\boldsymbol{\Phi}}_s$ and $\tilde{\boldsymbol{\Phi}}_f$, respectively. We can reasonably think that these corrections might be good candidates to represent the missing counterparts of the modal interaction model solutions. Namely, $\boldsymbol{C}_s$ could be thought of as a complement for the $\tilde{\boldsymbol{\phi}}_f$ and $\boldsymbol{C}_f$ as a complement for the $\tilde{\boldsymbol{\phi}}_s$. With this scheme, we intend to improve the results of the modal interaction problem outlined in Section (3.3.1).

There is an important computational point to note in (3.17) and (3.18). Namely, the dynamic stiffness matrices given by $(\mathbf{K}_s - \tilde{\lambda}\mathbf{M}_s)$ and $(\mathbf{K}_f - \tilde{\lambda}\mathbf{M}_f)$ are frequency dependent. And, subsequently, our intention is to update the frequencies at each step of the iteration process. In correct mathematical terms, this practice will bring the need to update dynamic stiffness matrices at each iteration which requires a separate factorization on each iteration. This practice is definitely unacceptable and will increase the computational costs unacceptably.

As mentioned above, the solution of equations (3.17) and (3.18) for each update of the frequency array can rather be expensive and is not much of a practical use. In order to partially circumvent this problem, we would like to use the expansion of the correction vectors, $\boldsymbol{C}_s$ and $\boldsymbol{C}_f$, in the space of the pseudo vectors $\boldsymbol{\Phi}_s$ and $\boldsymbol{\Phi}_f$. Although we know that the ingredients of the modal interaction model do not satisfy the exact boundary conditions of the problem, they are the only available information that we can use for the calculations that result from the problem physics itself. Moreover, one step further was already taken where we used them to get to some approximate coupled vectors through the use of the modal interaction problem represented in (3.13). In addition to the expansion, we would like to also take into account the modal truncation vectors that result from this expansion process [41]. Briefly speaking, we propose to use two vectors in order to represent the correction vectors in either the structural or fluid domains. The details of these two approaches are outlined next.

Expansion in the space of pseudo vectors

We can write the correction equations, (3.15) or (3.16), in the form of a general linear system, namely,

$$(\mathbf{K} - \lambda\mathbf{M})\,\boldsymbol{C} = \boldsymbol{f}. \tag{3.19}$$

As mentioned above, we represent the correction vector as a summation over either the *in vacuo* structural modes or the *rigid wall* acoustic modes, for the structural and the fluid domains, respectively. We can write the structural and fluid correction vectors as,

$$\boldsymbol{C}_s = \sum_{m=1}^{n_s} \Phi_{s,m}\eta_m = \boldsymbol{\Phi}_s\boldsymbol{\eta}_s \tag{3.20a}$$

$$\boldsymbol{C}_f = \sum_{m=1}^{n_f} \Phi_{f,m}\eta_m = \boldsymbol{\Phi}_f\boldsymbol{\eta}_f \tag{3.20b}$$

respectively.

In (3.20a) and (3.20b), n_s and n_f represent the number of pseudo vectors used in the expansion. We know that these pseudo vectors blocks, $\boldsymbol{\Phi}_s$ and $\boldsymbol{\Phi}_f$, are **K**- and **M**-orthogonal on the uncoupled system mass and stiffness matrices of the corresponding physics. This fact makes the computations cheaper. Using the orthogonality properties of the pseudo vectors, the expansion equation for one mode reads as,

$$(\lambda_{\star,m} - \lambda)\,\eta_m = \Phi_{\star,m}^{\mathrm{T}}\boldsymbol{f}, \tag{3.21a}$$

$$\eta_m = \frac{\Phi_{\star,m}^{\mathrm{T}}\boldsymbol{f}}{(\lambda_{\star,m} - \lambda)} \tag{3.21b}$$

where $\lambda_{\star,m}$ and $\Phi_{\star,m}$ represents the eigenvalues and the pseudo vectors found by either (3.8) or (3.9). In brief, the expansion is computed for the structural and acoustic domains independently in order to represent the correction vector, $\boldsymbol{C}$, in this space. Eventually, the correction vector is represented in the space of the pseudo vectors as,

$$\boldsymbol{C}_{exp} = \sum_{m=1}^{n} \Phi_{\star,m} \frac{\Phi_{\star,m}^{\mathrm{T}}\boldsymbol{f}}{(\lambda_{\star,m} - \lambda)} \tag{3.22}$$

where, in (3.22), n represents the column size of the vector spaces $\boldsymbol{\Phi}_s$ or $\boldsymbol{\Phi}_f$.

Modal truncation vectors

Since the pseudo vectors do not satisfy the interface conditions of the coupled vibro-acoustic problem, in general, it might be necessary to keep a large number of vectors in the expansion space [146, 152] which hinders the application of the modal interaction model to a large class of problems. This brings the need to find extra sources for the improvement of the convergence of these kinds of methods.

At this point, we propose to add the residual vectors or Modal Truncation Vectors (MTVs) associated to this forcing function [41, 42]. In order to shortly explain the theory of modal truncation idea, we will start with a general undamped forced response problem. Following the common notation used, we can write,

$$\mathbf{M}\ddot{\boldsymbol{u}} + \mathbf{K}\boldsymbol{u} = \boldsymbol{f}. \tag{3.23}$$

Following the common projection concepts of the model reduction literature where the equation is expanded in the available modes (or the pseudo mode vectors) of the system, namely, $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{\eta}_m$, one is able to write in block format,

$$\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{M}\boldsymbol{\Phi}\ddot{\boldsymbol{\eta}} + \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{K}\boldsymbol{\Phi}\boldsymbol{\eta} = \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{f}. \tag{3.24}$$

The MTV concept considers the omitted part of the spatial distribution of the force, namely, the part that is not represented in the projected forcing term $\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{f}$ in (3.24). We advise the reader to consult the above references on MTVs for a further discussion and the derivation of the equations therein. Following the mathematical derivations used in [41], one could easily write the omitted part of the applied force as

$$\boldsymbol{f}_t = \boldsymbol{f} - \mathbf{M}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{f}. \tag{3.25}$$

where $\boldsymbol{f}_t$ represents the truncated part of projected force vector, $\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{f}$. In other words, the force is projected onto the orthogonal complement of the pseudo vector set. Eventually, these projected forces are used to produce extra static residual enrichment vectors by solving,

$$\mathbf{K}\boldsymbol{C}_{mtv} = \boldsymbol{f}_t, \tag{3.26}$$

where $\boldsymbol{C}_{mtv}$ represents the static residual contribution of the truncated part of the projected force, $\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{f}$.

Intermediate summary and remarks on the correction equations
As a recap, instead of solving the actual correction equations (3.15) or (3.16), we intend to express the correction vectors, $\boldsymbol{C}_s$ or $\boldsymbol{C}_{\boldsymbol{f}}$, as a combination of two vectors. Namely,

- The first vector is the vector found with the expansion relation, (3.22),
- The second vector is computed as a residual force correction from (3.26).

Instead of using the previously outlined process, we can also use iterative solution techniques to solve the correction equations, an efficient solution scheme was also developed and the related details can be found in [150]. Another important observation is on the computation of the MTVs with (3.26). The residual vectors, $\boldsymbol{C}_{mtv}$, are computed on a per physics basis and we already performed the factorization of the system matrices for the initial Lanczos runs[1]. Therefore, we can use the available factorization information and only perform solutions to compute the residual vectors.

3.3.3. Enrichment with residual vectors

We outlined the modal interaction model in Section 3.3.1 from which we computed the initial vectors to start our iterative process. It is important to remember that these starting vectors are only first rough approximations of the coupled system modes. Although we got one step closer to the full coupled system modes with the modal interaction model, especially, for strongly coupled problems, we still can not find accurate representations of the actual coupled system modes. In other words, the residuals resulting from the use of these initial vectors, $\tilde{\boldsymbol{\Phi}}_{\mathbf{s}}$ and $\tilde{\boldsymbol{\Phi}}_{\mathbf{f}}$, will still be large. On the other hand, the static response of these residual vectors can provide some extra enrichment vectors when integrated into our projection space. Similar ideas were also proposed before by several authors for enrichment purposes [24, 153, 144].

However, in all the above cited references, they used the uncoupled residuals in their implementations. Namely, the residual vectors were computed for the structural and acoustic domains independently. This means that the coupling term, $\mathbf{K}_{sf}$, inside the coupled stiffness matrix were not used during the residual vector computations. At this point, we propose to use a slightly improved residual vector computation which also takes into account the coupling term, $\mathbf{K}_{sf}$, with a little extra cost due to the special topology of the coupled system stiffness matrix. By

---

[1] For extracting $\boldsymbol{\Phi}_s$ and $\boldsymbol{\Phi}_f$

investigating the coupled nonsymmetric stiffness matrix, $\mathbf{K}_c$ from (2.29), we can observe the block zero in the lower left corner, with this property in the mind, it is possible to write the coupled residual relation in blocks as follows:

$$\begin{bmatrix}\mathbf{K}_s & -\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{K}_f\end{bmatrix}\begin{bmatrix}\mathbf{X}_{R,s} \\ \mathbf{X}_{R,f}\end{bmatrix} = \begin{bmatrix}\mathbf{R}_s \\ \mathbf{R}_f\end{bmatrix}, \tag{3.27}$$

where, for a given block of approximate eigenvalues and eigenvectors, $(\tilde{\mathbf{\Omega}}, \tilde{\mathbf{\Phi}})$, the block residuals for the structural and acoustic domains are written as follows:

$$\mathbf{R}_s = \mathbf{K}_s\tilde{\mathbf{\Phi}}_s - \mathbf{M}_s\tilde{\mathbf{\Phi}}_s\tilde{\mathbf{\Omega}} - \mathbf{K}_{sf}\tilde{\mathbf{\Phi}}_f \tag{3.28a}$$

$$\mathbf{R}_f = \mathbf{K}_f\tilde{\mathbf{\Phi}}_f - \mathbf{M}_f\tilde{\mathbf{\Phi}}_f\tilde{\mathbf{\Omega}} - \mathbf{K}_{sf}^{\mathrm{T}}\tilde{\mathbf{\Phi}}_s\tilde{\mathbf{\Omega}} \tag{3.28b}$$

where $\tilde{\mathbf{\Phi}}_s$ and $\tilde{\mathbf{\Phi}}_f$ represent the structural and acoustic partitions of the approximate vector block $\tilde{\mathbf{\Phi}}$.

Returning to (3.27), due to the large zero block in the second row, the solution of this system can be performed in an efficient manner by using the independent factorizations of the structural and acoustic stiffness matrices, namely,

$$\mathbf{K}_s\mathbf{X}_{R,s} - \mathbf{K}_{sf}\mathbf{X}_{R,f} = \mathbf{R}_s, \tag{3.29a}$$

$$\mathbf{K}_f\mathbf{X}_{R,f} = \mathbf{R}_f, \tag{3.29b}$$

The only addition cost over the above mentioned references is the matrix vector multiplication performed for $\mathbf{K}_{sf}\mathbf{X}_{R,f}$ and an addition to find the modified right hand side in (3.29a) during the solution process. If a singularity exists on the fluid side, then a pseudo inverse should be employed in the solutions for the fluid side, namely, $\mathbf{K}_f^{+}$ [63, 49]. Eventually, after performing the solutions for the residual vectors in (3.29a) and (3.29b), $\mathbf{X}_{R,s}$ and $\mathbf{X}_{R,f}$ are the vector blocks that can be integrated in the projection basis.

3.3.4. Projection space and the interaction problem

In sections 3.3.2 and 3.3.3, we outlined the main sources of the vectors that build up the projection space, namely, the correction vector expansions along with MTVs and the residual vector components. In this section, we are going to briefly outline the projection space and the projection operations performed in order to end up with a reduced system representation or the so called the interaction problem [63].

Projection matrix

The coupled problem representation was given in (2.29) which results in real eigenvalues and vectors. This fact was proven in [161]. Therein, the authors used the relation between the off-diagonal coupling blocks in order to prove that the problem reveals real eigenvalues and eigenvectors. Namely, $\mathbf{M}_{fs} = \rho\mathbf{K}_{sf}^{\mathrm{T}}$. Following the same reasoning provided therein and projecting the problem onto a block diagonal basis results in the same kind of relations for the reduced problem. As a result, we can conclude that we are guaranteed to end up with a real eigenspace after the projection on a block diagonal basis. The general form of the block diagonal basis, $\mathbf{T}$, can be written as,

$$\mathbf{T} = \begin{bmatrix} \mathbf{\Psi}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{\Psi}_f \end{bmatrix} \tag{3.30}$$

$\mathbf{\Psi}_s$ and $\mathbf{\Psi}_f$ are mainly built with the enrichment vectors computed in Sections 3.3.1, 3.3.2 and 3.3.3, namely

$$\mathbf{\Psi}_s = \begin{bmatrix} \tilde{\mathbf{\Phi}}_s & \mathbf{C}_{exp,s} & \mathbf{C}_{mtv,s} & \mathbf{X}_{R,s} \end{bmatrix}, \tag{3.31a}$$

$$\mathbf{\Psi}_f = \begin{bmatrix} \tilde{\mathbf{\Phi}}_f & \mathbf{C}_{exp,f} & \mathbf{C}_{mtv,f} & \mathbf{X}_{R,f} \end{bmatrix}. \tag{3.31b}$$

These blocks, $\mathbf{\Psi}_s$ and $\mathbf{\Psi}_f$, are repeatedly orthogonalized independently to avoid numerical problems due to the appearance of collinear vectors. During the iterations, some mode vectors converge in the earlier steps of the iteration procedure. In case some converged modes are detected during the iteration process, correction vectors are not computed for the converged vectors anymore.

Projection and the interaction problem

Considering a system without external excitation and projecting (2.29) on (3.30) results in,

$$\underbrace{\begin{bmatrix} \hat{\mathbf{M}}_s & \mathbf{0} \\ \hat{\mathbf{M}}_{fs} & \hat{\mathbf{M}}_f \end{bmatrix}}_{\hat{\mathbf{m}}} \begin{bmatrix} \ddot{\hat{\boldsymbol{u}}} \\ \ddot{\hat{\boldsymbol{p}}} \end{bmatrix} + \underbrace{\begin{bmatrix} \hat{\mathbf{K}}_s & \hat{\mathbf{K}}_{sf} \\ \mathbf{0} & \hat{\mathbf{K}}_f \end{bmatrix}}_{\hat{\mathbf{k}}} \begin{bmatrix} \hat{\boldsymbol{u}} \\ \hat{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{3.32}$$

where

$$\hat{\mathbf{M}}_s = \mathbf{\Psi}_s{}^{\mathrm{T}} \mathbf{M}_s \mathbf{\Psi}_s, \quad \hat{\mathbf{M}}_f = \mathbf{\Psi}_f{}^{\mathrm{T}} \mathbf{M}_f \mathbf{\Psi}_f, \quad \hat{\mathbf{M}}_{fs} = \mathbf{\Psi}_f{}^{\mathrm{T}} \mathbf{K}_{sf}^{\mathrm{T}} \mathbf{\Psi}_s, \tag{3.33a}$$

$$\hat{\mathbf{K}}_s = \mathbf{\Psi}_s{}^{\mathrm{T}} \mathbf{K}_f \mathbf{\Psi}_f, \quad \hat{\mathbf{K}}_f = \mathbf{\Psi}_f{}^{\mathrm{T}} \mathbf{K}_f \mathbf{\Psi}_f, \quad \hat{\mathbf{K}}_{sf} = -\mathbf{\Psi}_s{}^{\mathrm{T}} \mathbf{K}_{sf} \mathbf{\Psi}_f. \tag{3.33b}$$

From (3.32), the reduced eigenvalue problem or the interaction problem reads as:

$$\left(\hat{\mathbf{k}} - \hat{\lambda} \hat{\mathbf{m}} \right) \hat{\phi} = \mathbf{0} \tag{3.34}$$

Later on, the recovery for the physical eigenvectors is accomplished with the back transformation

$$\tilde{\mathbf{\Phi}} = \mathbf{T} \hat{\mathbf{\Phi}}, \quad \hat{\mathbf{\Phi}} = [\hat{\phi}_1, \hat{\phi}_2, ..., \hat{\phi}_r] \tag{3.35}$$

where r represents the size of the interaction problem.

3.3.5. Energy based convergence check

Structural error measures for the convergence check of certain modes are well developed and documented in the literature [17] and [15]. Error check on mode vectors mentioned in [17] is summarized here for convenience. Namely, if $\tilde{\boldsymbol{\phi}}$ and $\tilde{\lambda}_s$ are one of the approximate eigenpairs of a structural system, the relative error measure check is given by,

$$\epsilon = \frac{\|\mathbf{K}_s\tilde{\boldsymbol{\phi}} - \tilde{\lambda}_s\mathbf{M}_s\tilde{\boldsymbol{\phi}}\|}{\|\mathbf{K}_s\tilde{\boldsymbol{\phi}}\|}, \tag{3.36}$$

where $\mathbf{K}_s$ and $\mathbf{M}_s$ are the structural stiffness and mass matrices, respectively. For convergence, ϵ should be small, on the order of 10^{-8} or 10^{-9}. Physically, this makes sense since nominator in (3.36) gives the residual of the force and this force is expected to be small in comparison to elastic forces that are created by the displacement vector given in the nominator in the same equation. However, direct application of this equation to a coupled vibro-acoustic problem will not result in sensible values for the values of the convergence indicator ϵ. The reason is that (2.29) is composed of two physics of very different orders, namely, the structural displacements, $\boldsymbol{u}$ and fluid pressures, $\boldsymbol{p}$. In applications, fluid pressures, $\boldsymbol{p}$, are orders of smaller in magnitude in comparison to that of the structural displacements, $\boldsymbol{u}$.[2] Mixing and summing up of these terms in a common Euclidean norm as given in (3.36) results in operations on expressions of very different order. This is not also a physically sound approach because operations on a relatively heavier domain, namely, the structure, are being mixed with a relatively lighter domain, namely, the fluid.

A next reasonable approach is to search for a convergence check on a per physics basis. It is mentioned in the literature that the use of the Euclidean norm neither takes into account the different scaling between different degrees of freedom, namely, the translations and rotations for a structural problem, nor the differences on the scaling of the participating physics. In this context, the latter is exactly the reason of the problem for this type of application. Thus, it is proposed in [15, 14] to use a strain energy norm or **K**-norm instead. For any arbitrary vector, $\boldsymbol{x}$, this norm is defined as

$$\|\boldsymbol{x}\|_{\mathbf{K}} = \sqrt{\boldsymbol{x}^{\mathrm{T}}\mathbf{K}\boldsymbol{x}}. \tag{3.37}$$

It is also possible to define an **M**-norm as well.

The proposed approach here is inline with the ideas discussed in [15, 85]. Namely, the use of the strain energy norm of the residuals per physics is suggested. The static responses of the residuals might be computed over a general problem with a given stiffness matrix **K** as

$$\boldsymbol{q}_R = \mathbf{K}^{-1}\boldsymbol{R}, \tag{3.38}$$

where $\boldsymbol{q}_R$ represents the residual vector generated with the considered approximate vector. In the context of a structural dynamics problem, the residual vector, $\boldsymbol{R}$, is written as,

$$\boldsymbol{R} = \mathbf{K}_s\tilde{\boldsymbol{\phi}} - \tilde{\lambda}\mathbf{M}_s\tilde{\boldsymbol{\phi}}, \tag{3.39}$$

---

[2] Structural displacement vector, $\boldsymbol{u}$, has also different components, such as translations and rotations which might have different scaling in itself.

where $(\tilde{\lambda}, \tilde{\boldsymbol{\phi}})$ together represent an approximate eigenpair. Convergence is achieved if the ratio of the relative strain energy of the residual vector to the strain energy of the corresponding vector, $\tilde{\boldsymbol{\phi}}$,

$$\epsilon_{sen} = \frac{\|\boldsymbol{q}_R\|_{\mathbf{K}}}{\|\tilde{\boldsymbol{\phi}}\|_{\mathbf{K}}}, \tag{3.40}$$

is below a certain tolerance limit ϵ_{tol}. Subscript *sen* in (3.40) represents the *strain energy*. This error could be computed efficiently since the strain energy is represented by some matrix vector products, namely,

$$\|\boldsymbol{q}_R\|_{\mathbf{K}} = \sqrt{\boldsymbol{q}_R^{\mathrm{T}}\mathbf{K}\boldsymbol{q}_R} \overset{(3.38)}{\equiv} \sqrt{\boldsymbol{q}_R^{\mathrm{T}}\mathbf{K}\mathbf{K}^{-1}\boldsymbol{R}} \equiv \sqrt{\boldsymbol{q}_R^{\mathrm{T}}\boldsymbol{R}}. \tag{3.41}$$

In the discussion here, $\tilde{\boldsymbol{\phi}}$ is one of the vectors present in the recovered vector block after the solution of the interaction problem, namely, (3.35). Moreover, by use of the projection space, $\mathbf{T}$, and the orthogonality properties[3] between the results of the interaction problem, we can prove that the denominator in (3.40) can directly be written as $\hat{\lambda}$ which is a direct result of the interaction problem.

The briefly outlined scheme can also be used in a mono-physical problem. However, in the case of a coupled vibro-acoustic problem, if the approximate mode vectors are computed through the interaction problem as given in (3.32) and, subsequently, if we separate these vectors into structural and acoustic parts, they do not satisfy the orthogonality relations with respect to the mass and stiffness matrices of the structural and fluid domains anymore. Therefore, another approach that considers the different scaling in different domains[4] of the problem is necessary. One possible solution to this challenge is to define the strain energy norms for the structural and fluid parts separately. Namely,

$$\epsilon^k_{s,sen} = \frac{\|\boldsymbol{q}^k_{R,s}\|_{\mathbf{K}_s}}{\|\tilde{\boldsymbol{\phi}}^k_s\|_{\mathbf{K}_s}}, \tag{3.42a}$$

$$\epsilon^k_{f,sen} = \frac{\|\boldsymbol{q}^k_{R,f}\|_{\mathbf{K}_f}}{\|\tilde{\boldsymbol{\phi}}^k_f\|_{\mathbf{K}_f}}. \tag{3.42b}$$

(3.42a) represents the strain energy error measure for the structural part and (3.42b) represents the strain energy error measure for the fluid part, respectively. In both (3.42a) and (3.42b), k represents the mode number for which the partial error measures are computed on.[5] Subscripts s and f represent that these ratios are computed for structural and fluid domains, respectively. Similarly, $\mathbf{K}_s$ and $\mathbf{K}_f$ represent the structural and fluid stiffness matrices, respectively. $\tilde{\boldsymbol{\phi}}_s$ and $\tilde{\boldsymbol{\phi}}_f$ are the structural and fluid parts of the recovered vectors computed by using the recovery process after the interaction problem solution, namely, by (3.35).

---

[3] Stiffness and mass orthogonality between the vectors

[4] Namely, over the structural and acoustic dofs

[5] In the following discussions, k will be dropped for simplicity.

By using (3.38) and (3.41), we can write (3.42a) and (3.42b) as follows:

$$\epsilon_{s,sen} = \frac{\boldsymbol{q}_{R,s}\boldsymbol{R}_s}{\tilde{\boldsymbol{\phi}}_s^{\mathrm{T}}\mathbf{K}_s\tilde{\boldsymbol{\phi}}_s}, \tag{3.43a}$$

$$\epsilon_{f,sen} = \frac{\boldsymbol{q}_{R,f}\boldsymbol{R}_f}{\tilde{\boldsymbol{\phi}}_f^{\mathrm{T}}\mathbf{K}_f\tilde{\boldsymbol{\phi}}_f}. \tag{3.43b}$$

In Section 3.3.3, we outlined the computation of the residual vectors as an extra source of enrichment. These residuals can directly be used for the convergence check calculations. Namely, $\mathbf{X}_{R,s}$ and $\mathbf{X}_{R,f}$ can directly be used in the convergence checks by replacing them with $\boldsymbol{q}_{R,s}$ and $\boldsymbol{q}_{R,f}$, respectively. In conclusion, the residual vectors are used for two purposes. Namely, the first purpose is to enrich the projection space for the subsequent iterations. And the second purpose is to perform an energy based convergence check for mode vectors on a per physics basis. Eventually, the error measure estimation for the complete mode vector is performed with a geometric mean which uses the strain energy norms, $\epsilon_{s,sen}$ and $\epsilon_{f,sen}$ [85], namely,

$$\epsilon = \sqrt{\epsilon_{s,sen}\epsilon_{f,sen}}, \tag{3.44}$$

which is computed for the mode vectors under consideration. Partial error norms computed on the coupling domains, namely, (3.43a) and (3.43b), can also be of different orders and the use of a geometric mean facilitates an efficient means to average these partial error measures. If ϵ is smaller than a given tolerance then the mode under consideration is marked as converged and the correction vectors for this specific mode vector are not computed anymore. In the numerical tests performed, the convergence tolerance was set to 10^{-8} which is referred to as ϵ_{irca} from now on.

3.4. Notes on coupled residual, convergence and relation with Krylov vectors

The convergence of Subspace iteration is outlined in Appendix C.2 for symmetric systems, but our problem is a generalized eigenvalue problem with nonsymmetric matrices to which these proofs do not apply directly. However, several transformation matrices were presented before [52, 53] where a symmetric representation can be constructed for vibro-acoustic problems. In [52], the author proposed two premultiplication and two post-multiplication matrices from which symmetric systems can be built. The transformation matrices and the resulting system matrices are presented in detail in Appendix B.

As presented in Appendix B, the systems either end up with fully populated symmetric stiffness or mass matrices in closed form and do not have a direct practical use for numerical applications. However, they are still useful and can provide us with a tool that we can use to show the relations between these residual vectors and the Krylov vectors. Moreover, since there is a close relation and connection with Krylov vectors used in Lanczos iterations, we believe that these vectors also improve the convergence rate of the technique outlined in this Chapter. Namely, by using the pre-multiplication matrices presented in Appendix B here, namely as $\mathbf{P}$, the generalized eigenvalue problem reads as

$$\mathbf{P}\mathbf{K}_c\boldsymbol{\phi} = \lambda\mathbf{P}\mathbf{M}_c\boldsymbol{\phi} \longrightarrow \mathbf{P}\left(\mathbf{K}_c\boldsymbol{\phi} = \lambda\mathbf{M}_c\boldsymbol{\phi}\right) \tag{3.45}$$

where $\mathbf{K}_c$ and $\mathbf{M}_c$ are the nonsymmetric system matrices of the original generalized eigenvalue problem from (2.29).

Considering the inverse iteration relation for (3.45), we can realize that the premultiplier matrix, $\mathbf{P}$, does not have any effect on the resulting iteration vectors. Mathematically, the inverse iteration step for the system in (3.45) is equivalent to

$$\mathbf{K}_c\boldsymbol{\phi}_{k+1} = \mathbf{M}_c\boldsymbol{\phi}_k \tag{3.46}$$

This implicitly means that, even if we use (3.46) with nonsymmetric system matrices, we are still iterating in the same space of the vectors that would eventually be found with the symmetric representation in (3.45).

As outlined in Section 3.3.3, one of the ingredients of the projection space is the static response of the residual vectors from the previous step, namely,

$$\boldsymbol{q}_R = \mathbf{K}_c^{-1}\underbrace{\left(\mathbf{K}_c\tilde{\boldsymbol{\phi}} - \hat{\lambda}\mathbf{M}_c\tilde{\boldsymbol{\phi}}\right)}_{\boldsymbol{R}} \equiv \left(\tilde{\boldsymbol{\phi}} - \hat{\lambda}\mathbf{K}_c^{-1}\mathbf{M}_c\tilde{\boldsymbol{\phi}}\right) \tag{3.47}$$

where $\hat{\lambda}$ and $\tilde{\boldsymbol{\phi}}$ originate from the interaction problem with the recovery step in (3.35). And, $\boldsymbol{R}$ is the residual vector for the eigenpair $(\hat{\lambda}, \tilde{\boldsymbol{\phi}})$.

One can observe from (3.47) that $\boldsymbol{q}_R$ has two components. Namely,

- The first component is $\tilde{\boldsymbol{\phi}}$, the direct result of the interaction problem,
- The second one is $\mathbf{K}_c^{-1}\mathbf{M}_c\tilde{\boldsymbol{\phi}}$ multiplied with $\hat{\lambda}$. This component lives in the same space as the inverse iteration result for the next step as mentioned in Algorithm 3.1. The results of the inverse iteration step are also known as the Krylov vectors. Therefore, we can conclude that the coupled residual response found with (3.47) implicitly includes components of Krylov vectors scaled by $\hat{\lambda}$. Besides, it is shown in Appendix C.2 that the convergence is mainly based on the inverse iteration operation with which the contribution of the higher order modes vanish in the limit. This fact is based on the expansion of the vectors in the space of eigenvectors. Similarly, for the technique discussed here, we are performing a similar iteration but the ingredients are not the Krylov vectors directly but has components of Krylov vectors. Namely, they are a linear combination of the results of the interaction problem, $\tilde{\boldsymbol{\phi}}$, and the Krylov vectors, $\mathbf{K}_c^{-1}\mathbf{M}_c\tilde{\boldsymbol{\phi}}$ scaled by $\hat{\lambda}$. Due to their close relation to Krylov vectors, we can expect an improvement in the convergence of the method by using the above outlined coupled residual vectors in the projection matrix $\mathbf{T}$.

Briefly, by using the coupled residual response, we are including vector components which have close relations to real Krylov vectors. Moreover, we are iterating in a richer space in addition to the expansion vectors and the modal truncation vectors.

3.5. Additional practical considerations

3.5.1. Remarks on the target mode count and basis size

In order to start the iterations, we proposed to use the results of the modal interaction model. Namely, we started the iterative process with two sets of decoupled vectors from the structural and fluid domains, respectively. For this initial step, we advise to use an equal number of mode vectors from the structural and acoustic domains, respectively. However, this is not strictly necessary since we can also select the number of vectors to start the iterations with a simple criterion as follows:

$$n = \frac{min(n_s, n_f)}{c}. \tag{3.48}$$

In (3.48), n represents the number of improved vectors to be used at the end of the modal interaction step, namely, the number of recovered vectors used in (3.14). Besides, n_s represents the number of vectors computed for the structural domain and, similarly, n_f represent the number of vectors computed for the acoustic domain. c is an integer number which independently sets the number of target modes to be computed in the low frequency band for the iterative process. Our numerical experience and tests show that we can select c as 3. This approach physically makes sense due to the following reason which is closely related to the convergence check and the number of vectors that converge during the iterations. Namely, as the iterations advance, we start to detect convergence in mode vectors in the low frequency band. Subsequently, we do not solve for the correction vectors for these converged mode vectors anymore. The implication of this implementation is that the number of the vectors in the projection space, $\mathbf{T}$, decreases as the iterations advance. In other words, we decrease the number of enrichment vectors that we use to build the projection space. This is mainly performed to decrease the numerical cost of the operations. Therefore, we can not expect convergence to a high number of modes and we have to put a practical upper limit on the number of vectors that we are searching in the low frequency band. This means that, although we start the iterative process with a high number of vectors, in the course of the iterations, we intend to decrease the number of vectors used to build up the projection space, $\mathbf{T}$.

3.5.2. Convergence acceleration: use of buffer vectors

The convergence rate of Subspace Iteration depends on λ_i/λ_{n+1} [63, 18], where n is target mode count computed heuristically with (3.48). It is worth noting that, here, the target mode count, n, replaces the target block size, p, of the original Subspace Iteration[6]. Inline with [89], we propose to integrate some buffer vectors into the reduction space, $\mathbf{T}$, in order to accelerate the convergence slightly. Number of buffer vectors is selected as 8 according to the criterion adapted from [89]. Namely, a total number of iteration vectors of $min(2p, p+8)$ should be selected according to the original criterion. However, in our technique, we modified this criterion to decrease the additional number of improved vectors, $\tilde{\mathbf{\Phi}}_s$ and $\tilde{\mathbf{\Phi}}_f$, which are directly included in the projection space, $\mathbf{T}$, (3.30).

---

[6] Namely, see Section 3.2

In the light of the practical considerations and the previous theoretical discussions, the complete set of steps is summarized in Algorithm 3.2.

Algorithm 3.2 Iterative reduced correction algorithm, Irca

```
Input $n_s$ and $n_f$    ▷ pseudo vector sizes for structural and acoustic domains
Input convergence tolerance, $\epsilon_{irca} = 10^{-8}$
Set number of buffer vectors, $b = 8$
$(\mathbf{K}_s - \lambda_s \mathbf{M}_s)\, \boldsymbol{\phi}_s = \mathbf{0}$    ▷ Extract $n_s$ pseudo vectors, $\boldsymbol{\Phi}_s$
$(\mathbf{K}_f - \lambda_f \mathbf{M}_f)\, \boldsymbol{\phi}_f = \mathbf{0}$    ▷ Extract $n_f$ pseudo vectors, $\boldsymbol{\Phi}_f$
$\mathbf{T}_m = \begin{bmatrix} \boldsymbol{\Phi}_s & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_f \end{bmatrix}$    ▷ Create the initial projection space, $\mathbf{T}_m$
$\tilde{\mathbf{k}} = \mathbf{T}_m^{\mathrm{T}} \mathbf{K}_c \mathbf{T}_m, \quad \tilde{\mathbf{m}} = \mathbf{T}_m^{\mathrm{T}} \mathbf{M}_c \mathbf{T}_m$    ▷ Project the problem on $\mathbf{T}_m$ to get (3.13)
$\tilde{\mathbf{k}}\tilde{\Phi} = \tilde{\lambda}\tilde{\mathbf{m}}\tilde{\Phi}$    ▷ Solve the modal interaction problem
$n = \frac{min(n_s, n_f)}{c}$    ▷ Compute the target mode count for $c = 3$
$\begin{bmatrix} \tilde{\boldsymbol{\Phi}}_s \\ \tilde{\boldsymbol{\Phi}}_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_s & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_f \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \tilde{\boldsymbol{p}} \end{bmatrix}$    ▷ Recover the first $n$ physical vectors
$\mathbf{R}_s = \mathbf{K}_s \tilde{\boldsymbol{\Phi}}_s - \mathbf{M}_s \tilde{\boldsymbol{\Phi}}_s \tilde{\boldsymbol{\Omega}} - \mathbf{K}_{sf} \tilde{\boldsymbol{\Phi}}_f$    ▷ Compute the residuals for domains
$\mathbf{R}_f = \mathbf{K}_f \tilde{\boldsymbol{\Phi}}_f - \mathbf{M}_f \tilde{\boldsymbol{\Phi}}_f \tilde{\boldsymbol{\Omega}} - \mathbf{K}_{sf}^{\mathrm{T}} \tilde{\boldsymbol{\Phi}}_s \tilde{\boldsymbol{\Omega}}$    ▷ $\tilde{\boldsymbol{\Omega}} = diag(\tilde{\lambda})$
$\begin{bmatrix} \mathbf{X}_{R,s} \\ \mathbf{X}_{R,f} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_s & -\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R}_s \\ \mathbf{R}_f \end{bmatrix}$    ▷ Compute the static residual responses
while $\epsilon > \epsilon_{irca}$ do    ▷ Loop until convergence
    $\mathbf{C}_s^{app} \approx [\mathbf{C}_{exp,s} \quad \mathbf{C}_{mtv,s}]$    ▷ Compute correction approximations
    $\mathbf{C}_f^{app} \approx [\mathbf{C}_{exp,f} \quad \mathbf{C}_{mtv,f}]$
    $\boldsymbol{\Psi}_s = [\tilde{\boldsymbol{\Phi}}_s \quad \mathbf{C}_s^{app} \quad \mathbf{X}_{R,s}]$    ▷ Build the projection space and orthogonalize
    $\boldsymbol{\Psi}_f = [\tilde{\boldsymbol{\Phi}}_f \quad \mathbf{C}_f^{app} \quad \mathbf{X}_{R,f}]$
    $\mathbf{T} = \begin{bmatrix} \boldsymbol{\Psi}_s & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_f \end{bmatrix}$
    $\tilde{\mathbf{k}} = \mathbf{T}^{\mathrm{T}} \mathbf{K}_c \mathbf{T}, \quad \tilde{\mathbf{m}} = \mathbf{T}^{\mathrm{T}} \mathbf{M}_c \mathbf{T}$    ▷ Project the problem on $\mathbf{T}$
    $\tilde{\mathbf{k}}\tilde{\tilde{\Phi}} = \tilde{\lambda}\tilde{\mathbf{m}}\tilde{\tilde{\Phi}}$    ▷ Solve the interaction problem
    $\begin{bmatrix} \tilde{\boldsymbol{\Phi}}_s \\ \tilde{\boldsymbol{\Phi}}_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Psi}_s & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_f \end{bmatrix} \begin{bmatrix} \tilde{\tilde{\boldsymbol{\Phi}}}_{\boldsymbol{s}} \\ \tilde{\tilde{\boldsymbol{\Phi}}}_{\boldsymbol{f}} \end{bmatrix}$    ▷ Recover $(n + b)$ vectors
    $\mathbf{R}_s = \mathbf{K}_s \tilde{\boldsymbol{\Phi}}_s - \mathbf{M}_s \tilde{\boldsymbol{\Phi}}_s \tilde{\boldsymbol{\Omega}} - \mathbf{K}_{sf} \tilde{\boldsymbol{\Phi}}_f$    ▷ Compute the residuals for domains
    $\mathbf{R}_f = \mathbf{K}_f \tilde{\boldsymbol{\Phi}}_f - \mathbf{M}_f \tilde{\boldsymbol{\Phi}}_f \tilde{\boldsymbol{\Omega}} - \mathbf{K}_{sf}^{\mathrm{T}} \tilde{\boldsymbol{\Phi}}_s \tilde{\boldsymbol{\Omega}}$
    $\begin{bmatrix} \mathbf{X}_{R,s} \\ \mathbf{X}_{R,f} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_s & -\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R}_s \\ \mathbf{R}_f \end{bmatrix}$    ▷ Compute the residual responses
    Check convergence by using (3.43a), (3.43b) and (3.44)
    if Convergence achieved on some vectors then
        Do not compute the correction vectors, in $\mathbf{C}_s^{app}$ and $\mathbf{C}_f^{app}$, for these
    else
        Compute correction vectors for the rest of the target vectors
    end if
end while
```

parameter	value
dimensions	0.9m×0.7m×0.4 m
shell thickness	0.002 m
E-modulus	71 GPa
Poisson's ratio	0.3
shell density	2800 kg/m^3
structural dofs	9132
fluid dofs	34776
total dofs	43908

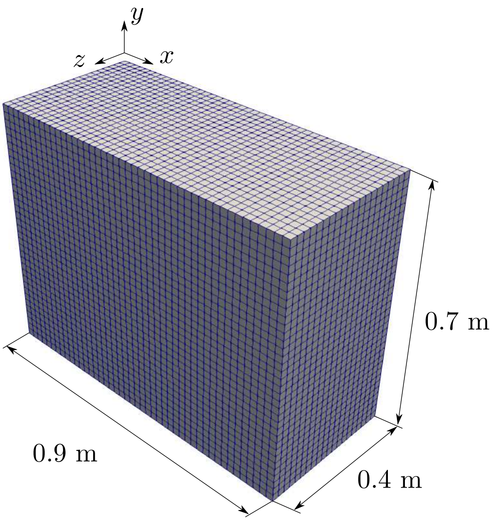


Figure 3.1: Test model, parameters(left) and the meshed model with dimensions(right)

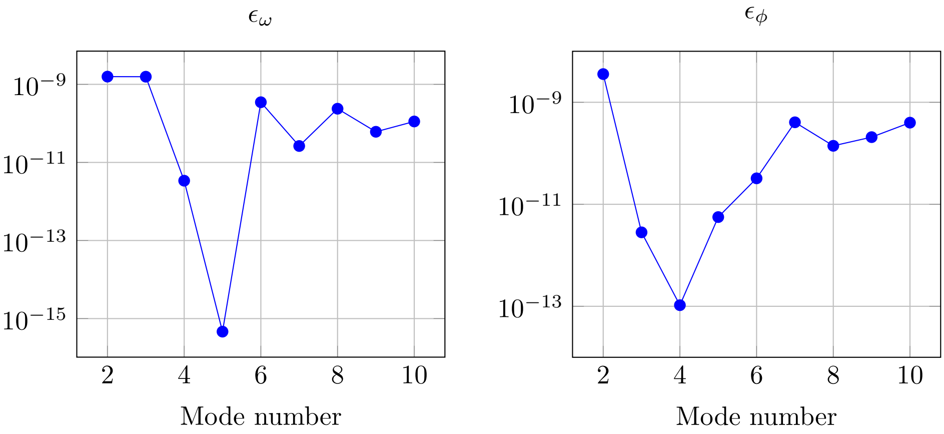
3.6. Numerical case studies

To test the ideas that were discussed in the theoretical analysis, a 3-dimensional cavity model adapted from [152] was investigated. The finite element models were generated in the commercial FE package `ANSYS`[6] which has the capability to model the fluid-structure interaction and the associated coupling conditions. For numerical tests, shell elements, `SHELL181` in `ANSYS`, are coupled to three dimensional fluid elements, `FLUID30` of `ANSYS`. For this problem, the edges of the coupling elastic shell surface(the largest face towards the reader in Figure 3.1) is simply supported to simulate a more flexible structural domain in order to increase the level of coupling. It is important to note that no prescribed pressure was imposed on the surfaces of the domain. The coupled non-symmetric system stiffness and mass matrices are then exported from `ANSYS` binary result files by the use of the interface which is developed in `C++` [145]. Subsequently, all the computations were performed in `MATLAB`. Numerical tests were performed with two different fluids which fill the cavity. The first fluid that is used in the tests is air and the second fluid used is water. The density of air was selected as 1.215 kg/m^3 along with a speed of sound of 343 m/s. Density and the speed of sound for water were selected as 1000 kg/m^3 and 1500 m/s, respectively.

For each of these different fluid configurations, the following results are provided separately:

- Relative frequency errors computed with respect to the results of the commercial FE code `ANSYS`, namely,

$$\epsilon_\omega = \frac{(\omega_{IRCA} - \omega_{ANSYS})}{\omega_{ANSYS}} \tag{3.49}$$



(a) Relative errors on frequencies **(b)** Errors on the eigenvectors

Figure 3.2: Errors on frequencies and eigenvectors, air filled cavity

- Accuracy measures of the computed eigenvectors with respect to the results computed with `ANSYS`. In these comparisons, we intend to use the Modal Assurance Criterion, MAC [47]. Namely, an error measure is computed by using

$$\epsilon_\phi = 1 - \mathrm{MAC} \equiv 1 - \frac{({\phi_{IRCA}}^{\mathrm{T}} \phi_{ANSYS})^2}{({\phi_{IRCA}}^{\mathrm{T}} \phi_{IRCA})({\phi_{ANSYS}}^{\mathrm{T}} \phi_{ANSYS})} \tag{3.50}$$

  where ϕ_{IRCA} represents the mode vectors computed by *IRCA* and, similarly, ϕ_{ANSYS} represents the mode vectors computed by `ANSYS`.

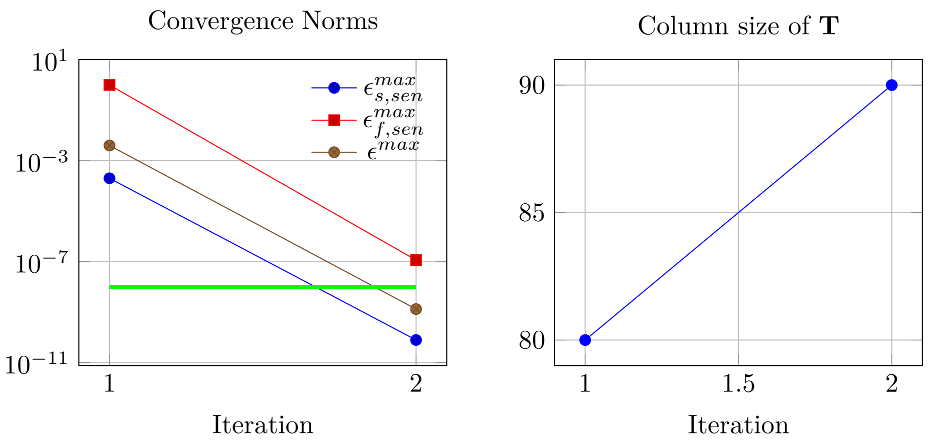
- The convergence indicators computed with (3.43a), (3.43b) and (3.44),

- The change in the column size of **T**, namely, the projection matrix,

3.6.1. Test case 1, air filled cavity

Generally, if air is used as the fluid filling the cavity, this practice results in a weak coupling due to the relatively low stiffness of air in comparison to heavier fluids like water. For these kinds of weakly coupled problems, a modal interaction model might provide fair accuracy for the coupled system response for a very limited number of low frequency modes. However, our aim is to show the performance of the developed technique as an eigenvalue solver. Therefore, we investigate the convergence properties and the accuracy of the method for different number of input vectors. As mentioned in Section 3.3.1, these vectors result from the solution of the two decoupled eigenvalue problems, namely, provided in (3.8) and (3.9).

The number of input vectors, namely, the *in vacuo* structural modes and the *ridig wall* cavity modes were selected as 30 for this case. The number of target modes computed with (3.48) is 10 for this test case.



(a) Convergence estimators

(b) Column size of **T**

Figure 3.3: Convergence norms and change in basis size, air filled cavity

Figures 3.2a and 3.2b show the frequency and eigenvector errors with respect to the result computed by the commercial code ANSYS. In these comparisons, the first frequency is numerically always 0 due to the singularity of the acoustic stiffness matrix therefore the comparisons start with the second mode. The error levels for the selected target mode numbers are significantly small which shows that the accuracy of the tested algorithm is quite acceptable for practical applications.

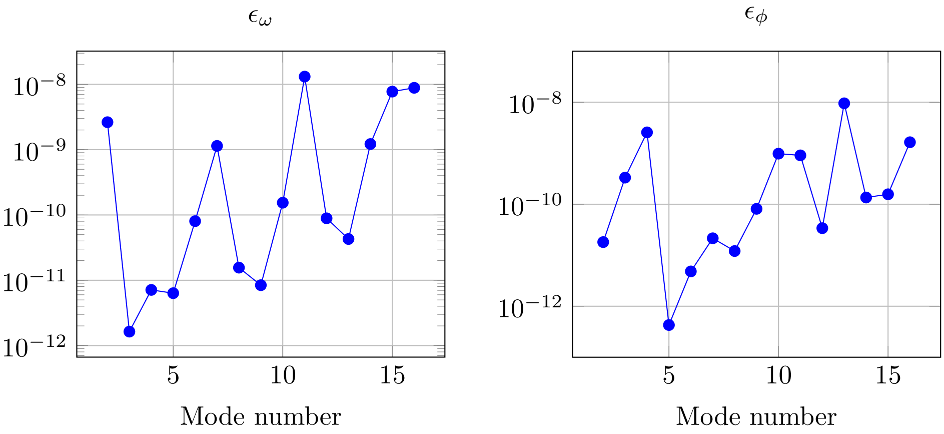
Figure 3.3a shows the error norms and their change during the iterations. The bold green line in Figure 3.3a represent the selected convergence tolerance for the numerical tests, namely, $\epsilon_{irca} = 1e-8$. Figure 3.3b show the change in the column size of the projection basis, **T**, during the iterations.

3.6.2. Test case 2, water filled cavity

Similar to the air filled cavity case outlined above, results are also presented for the cavity filled with water. Besides, the reason to use water in our tests is intentional. Namely, water has a much higher density than air which results in an increased coupling between the structure and the fluid. With this selection, the developed method is also tested for strongly coupled problems.

For this case, we started with 50 initial vectors from the structural and fluid domains. With our heuristic target mode selection criteria, (3.48), we aim to get convergence on 16 modes. Similar to the previous case, we present the results for the errors on the frequencies and on the eigenvectors, respectively, along with some convergence related properties.

Figure 3.4a shows the relative errors of the computed frequencies which are again significantly small. Figure 3.4b shows the error on the computed eigenvectors. Overall, it is important to note that the error levels shown in Figures 3.4a and 3.4b are significantly small. Investigating Figure 3.5a, we can again observe the convergent behavior of the method. However, in this case, the initial basis is not rich enough



(a) Relative errors on frequencies **(b)** Errors on the eigenvectors

Figure 3.4: Errors on frequencies and eigenvectors, water filled cavity

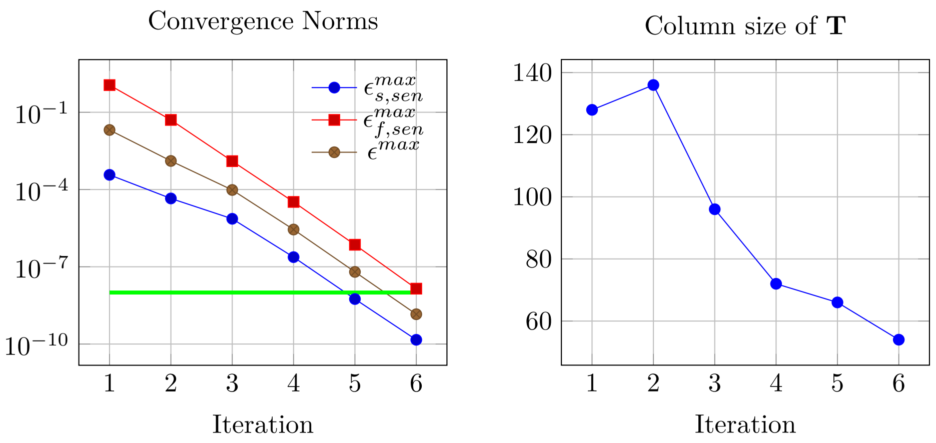
to represent the coupling between the domains and it requires more iterations for the algorithm to get convergence to the selected number of low frequency modes. In comparison to the air filled cavity case, the total number of iterations increased from 2 to 6. Basis size evaluation in Figure 3.5b is also different from that of the air case. Namely, at first, the size of the basis increased in both of the tests cases. But, for this test case, as we iterate, it started to decrease since corrections for the converged mode vectors are not computed. This is also inline with the expected behaviour.

3.7. Conclusions and discussion

In this chapter, we proposed an iterative algorithm, similar to Subspace iteration technique, for the solution of the vibro-acoustic eigenvalue problems. Widely available practice in the literature is to use the uncoupled mode vectors from the coupling domains and to project the problem on this basis in order to end up with a reduced problem to approximate the coupled vectors. However, these uncoupled vectors do not satisfy the boundary conditions on the coupling interfaces. Therefore, one generally needs a large set of these initial uncoupled vectors in the projection operations to end up with a fair accuracy on the coupled results.

To cure these problems with the uncoupled mode vectors,

- We proposed to also start the iterations with the uncoupled(in-vacuo) structural and acoustic mode vectors. But, at each iteration, we improved the starting vectors in a Rayleigh-Ritz sense similar to what is performed in Subspace iterations. In these iterations, some correction vectors were also used which approximate the forced responses of the coupling effects from the other domain. In addition, some static residual vectors were also used for the enrichment of the projection space.



(a) Convergence estimators **(b)** Column size of **T**

Figure 3.5: Convergence norms and change in basis size, water filled cavity

- Based on a scaling matrix presented in [52] which helps to convert the original nonsymmetric problem into a symmetric one, we show that the static response of the coupled residuals implicitly include components of the Krylov vectors. With our numerical experience, we have verified that the residuals vectors improve the convergence of the technique significantly.

Although the technique is attractive and provides results with a good accuracy, Subspace iteration like techniques discard the information generated in the previous iteration steps and, therefore, the computational cost of the operations is higher in comparison to Lanczos type Krylov Subspace based methods [63]. It is also possible to come up with a iteration scheme that uses symmetric matrices by using the transformation matrix provided in [52]. We have also investigated this option and found out that a robust and symmetric Subspace iteration scheme can be built which was not reported here. However, we have realized that efficient Lanczos iterations (see next chapter) are possible and we did not investigate further Subspace alike options. Therefore, a logical follow up and extension of this research will be presented in the next chapter. Namely, chapter 4 addresses the above mentioned deficiencies of the Subspace iteration technique and details an efficient iteration scheme.

4

vibro-Lanczos, a Symmetric Eigenvalue Solver for Vibro-Acoustic Problems

In Chapter 3, we detailed the development of a Subspace-like eigenvalue solver for vibro-acoustic problems. Subsequently, we also provided proofs on the convergence of the method based on some transformation matrices provided in the literature[52]. By using these transformation matrices, it is possible to write the original nonsymmetric coupled system matrices in a symmetric format. This chapter discusses the development of a symmetric eigenvalue solver for coupled vibroacoustic eigenvalue problems based on the above mentioned transformation matrices. Therefore, the contents of the chapter can be considered as an improvement over the results in Chapter 3. Namely, original nonsymmetric system matrices are replaced by their symmetric equivalents. And the resulting matrices are efficiently used in order to result in a truly symmetric eigenvalue solver for the originally nonsymmetric eigenvalue problem. In order to further improve the numerical efficiency, a partial orthogonalization scheme is also integrated during the numerical implementation. Eventually, the proposed solver is compared with the well known nonsymmetric Lanczos and Arnoldi methods. Results presented in this chapter were published in [151].

4.1. Introduction

Efficient solution of eigenvalue problems for vibro-acoustic systems is still an important problem for industrial design challenges. In this chapter, we are going to consider the application of the Lanczos method on the nonsymmetric generalized eigenvalue problem which emerges from the coupled vibro-acoustic systems as presented in Chapter 2, namely,

$$\left(\begin{bmatrix}\mathbf{K}_s & -\mathbf{K}_{sf}\\ \mathbf{0} & \mathbf{K}_f\end{bmatrix} - \lambda\begin{bmatrix}\mathbf{M}_s & \mathbf{0}\\ \mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f\end{bmatrix}\right)\begin{bmatrix}\boldsymbol{u}_s\\ \boldsymbol{p}_f\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix} \longrightarrow (\mathbf{K}-\lambda\mathbf{M})\,\boldsymbol{\phi} = \mathbf{0} \qquad (4.1)$$

[1] The ultimate goal in solving (4.1) is to find a large number of eigenvalues and eigenvectors efficiently for challenging industrial-sized problems. In (4.1), **K** and **M** are large sparse matrices. In the current context, the computational challenge originates,

---

[1] Writing (4.1), the subscript **c** used in $\mathbf{K}_c$ and $\mathbf{M}_c$ defined in Chapter 2 is be dropped for convenience.

basically, from two important points. Firstly, in the classical displacement-pressure ($\boldsymbol{u} - \boldsymbol{p}$) formulation, the operator matrices of the original problem represented in (4.1), namely $\mathbf{K}$ and $\mathbf{M}$ are nonsymmetric [105]. Therefore, one has to resort to solution methods for nonsymmetric systems. Secondly, the nature of the problem necessitates the use of two different physics, namely, a structural domain is coupled to a fluid domain over an interfacing surface. With the introduced coupling, the resulting discretized system matrices include terms which are orders of magnitude different resulting in ill-conditioned problems. In most of the practical engineering applications, these scaling differences put severe restrictions on the use of iterative solution techniques which could circumvent the factorization costs of large problems.

Prominent alternatives for the solution of the original nonsymmetric eigenvalue problem are Arnoldi [10] and nonsymmetric Lanczos methods [136]. Moreover, a specific implementation of the two-sided nonsymmetric Lanczos method for vibro-acoustic problems was investigated in [125]. Arnoldi can be classified as an orthogonal projection method whereas nonsymmetric Lanczos method is an oblique projection method where orthogonal projection methods are found to be performing better from a conditioning point of view [38]. Following the line of nonsymmetric solvers, a useful computational performance overview on the comparison of the nonsymmetric eigenvalue solvers for fluid-structure interaction problems is provided in [9]. Basically, Arnoldi type methods suffer from the fact that, as the size of the Krylov subspace grows, orthogonalization costs increase. In order to circumvent this problem, several restarting schemes were proposed [92]. The main problem encountered in the nonsymmetric Lanczos methods is the possibility of break-down during the biorthogonalization of the left and right Krylov basis vectors. Moreover, due to the generation of two subspaces, namely, the left and right Krylov subspaces, these methods are more expensive in comparison to the one-sided Arnoldi type methods. Approaches to cure these break-down problems were proposed in [56]. It is worth noting that symmetric methods do not suffer from the above mentioned problems and a well known theory on error bound computations exists [38].

Some symmetric alternatives for the ($\boldsymbol{u}-\boldsymbol{p}$) formulation were proposed in the literature, namely, symmetrization techniques such as those proposed in [84, 52] exist. However, direct use of these approaches is not practical due the high computational cost of the operations with the resulting matrices. Other symmetric alternatives which use the velocity potential as the unknown variable for the fluid part were also developed [46, 115]. Another promising alternative leading to a symmetric eigenvalue problem uses both the pressure and the displacement potentials as unknowns [105]. Although this formulation doubles the total degrees of freedom over the fluid domain, the symmetric nature is still advantageous from a computational perspective. This approach has already been integrated in commercial codes [147].

The aim of this chapter is to outline and detail a new symmetric Lanczos type solution technique for vibro-acoustic eigenresponses which uses the common displacement-pressure ($\boldsymbol{u}-\boldsymbol{p}$) formulation. This method makes use of the transformation matrices which is proposed and formulated in [52]. Implicitly, all the iterations are completely performed with symmetric matrices. However, these symmetric matrices are never formed and used explicitly. In this scheme, Lanczos process generates the same iterates as the well known Arnoldi algorithm but the numerical operations are arranged in a way that a reduced symmetric tridiagonal is formed.

We will consider completely closed volumes which are coupled to elastic structures through a coupling surface. These kinds of systems reveal a zero-frequency constant pressure mode, which is a result of the singularity of the fluid stiffness matrix, $\mathbf{K}_f$[2]. Extensive mathematical details are provided in [105] for these kinds of systems and we directly refer and build upon the discussions provided therein. It is important to emphasize that a closed volume problem is a general case for acoustics. If some pressure boundary condition is imposed on the model, the above mentioned *acoustic* ridid body mode is not present and the methods outlined here can also be applied without modifications.

The accuracy benchmark of the results are performed against the results of the Arnoldi algorithm. It is observed that relative errors on computed eigenvalues and eigenvectors are small. In contrary to common implementations and research in the field, our research shows that the eigenresponses of a vibro-acoustic system can be predicted efficiently in a symmetric format with the presented approach.

The organization of this chapter is as follows. Section 4.2 provides an overview on nonsymmetric solution techniques for vibro-acoustic eigenvalue problems. Section 4.3 discusses possible paths in order to transform the problem into a symmetric representation. Section 4.4 provides an overview on the symmetric Lanczos iterations detailing the efficient inverse iteration steps along with the projections that result in symmetry. In Section 4.4, we also discuss the details of the implemented orthogonalization scheme along with the remarks on shifting and Sturm sequence checks. Finally, the discussed concepts are tested on two model problems in Section 4.5. Section 4.6 summarizes the main conclusions.

4.2. Nonsymmetric eigenvalue solvers for vibro-acoustic problems

In this small section, we shortly introduce the two-sided Lanczos and Arnoldi methods in order to later outline the advantages of the proposed method with respect to these two methods.

4.2.1. Two-sided Lanczos method

The two-sided Lanczos method is an iterative projection method for the nonsymmetric eigenvalue problem represented in (4.1). Starting from two vectors, $\boldsymbol{p}_1 \neq 0$ and $\boldsymbol{q}_1 \neq 0$, and with the use of two matrix operators $\mathbf{K}^{-\mathrm{T}}\mathbf{M}^{\mathrm{T}}$ and $\mathbf{K}^{-1}\mathbf{M}$, it generates two growing orthogonal Krylov bases, namely the left and the right bases.

The left basis spans [125, 133]

$$\begin{aligned}\mathcal{K}^k(\mathbf{K}^{-\mathrm{T}}\mathbf{M}^{\mathrm{T}}, \boldsymbol{p}_1) &= span\{\boldsymbol{p}_1, \left(\mathbf{K}^{-\mathrm{T}}\mathbf{M}^{\mathrm{T}}\right)\boldsymbol{p}_1, \ldots, \left(\mathbf{K}^{-\mathrm{T}}\mathbf{M}^{\mathrm{T}}\right)^{k-1}\boldsymbol{p}_1\} \\ &\equiv span\{\mathbf{P}_k\}\end{aligned} \tag{4.2}$$

---

[2] By analogy with structures, we will call in what follows the constant pressure mode *the rigid body mode*

similarly the right basis spans

$$\begin{aligned} \mathcal{K}^k(\mathbf{K}^{-1}\mathbf{M}, \boldsymbol{q}_1) &= span\{\boldsymbol{q}_1, \left(\mathbf{K}^{-1}\mathbf{M}\right)\boldsymbol{q}_1, \dots, \left(\mathbf{K}^{-1}\mathbf{M}\right)^{k-1}\boldsymbol{q}_1\} \\ &\equiv span\{\mathbf{Q}_k\} \end{aligned} \tag{4.3}$$

Conceptually, at each step of the process, the nonsymmetric problem representation given by (4.1) is projected on these two subspaces, $\mathbf{P}_k$ and $\mathbf{Q}_k$ to find a reduced order problem from which the eigensolutions are computed. In addition, these two sets of vectors are also $\mathbf{M}$-biorthogonalized during the iteration process, namely, to satisfy $\mathbf{P}_k^{\mathrm{T}}\mathbf{M}\mathbf{Q}_k = \mathbf{I}$.

Lanczos [32] shows that three-term recurrence relations hold for the left and right iteration vectors, namely,

$$\tilde{\boldsymbol{p}}_{k+1} = \mathbf{K}^{-\mathrm{T}}\mathbf{M}^{\mathrm{T}}\boldsymbol{p}_k - \alpha_k\boldsymbol{p}_k - \delta_k\boldsymbol{p}_{k-1} \quad \text{with} \quad \delta_1\boldsymbol{p}_0 = 0, \tag{4.4}$$

$$\tilde{\boldsymbol{q}}_{k+1} = \mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}_k - \alpha_k\boldsymbol{q}_k - \beta_k\boldsymbol{q}_{k-1} \quad \text{with} \quad \beta_1\boldsymbol{q}_0 = 0 \tag{4.5}$$

By using the orthogonality relations between the left and right vectors, it can be shown that

$$\alpha_k = \boldsymbol{p}_k^{\mathrm{T}}\mathbf{M}\mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}_k \tag{4.6}$$

$$\delta_{k+1} = |\boldsymbol{p}_{k+1}^{T}\mathbf{M}\boldsymbol{q}_{k+1}|^{1/2} \tag{4.7}$$

$$\beta_{k+1} = \delta_{k+1} sign(\boldsymbol{p}_{k+1}^{T}\mathbf{M}\boldsymbol{q}_{k+1}) \tag{4.8}$$

Therefore, subsequent iterate vectors are found to be

$$\boldsymbol{p}_{k+1} = \frac{\tilde{\boldsymbol{p}}_{k+1}}{\delta_{k+1}} \tag{4.9}$$

$$\boldsymbol{q}_{k+1} = \frac{\tilde{\boldsymbol{q}}_{k+1}}{\beta_{k+1}} \tag{4.10}$$

In essence, by using the relations from (4.6) to (4.10), one is generating the orthogonalization coefficients for the Krylov sequences. The coefficients α_k, δ_{k+1} and β_{k+1} build up the nonsymmetric tridiagonal matrix $\mathbf{T}_k$ column by column which represents the reduced eigenvalue problem at step k, namely,

$$\mathbf{T}_k\boldsymbol{\eta} = \tilde{\lambda}_k\boldsymbol{\eta}_k \tag{4.11}$$

where the nonsymmetric tridiagonal matrix is written as

$$\mathbf{T}_k = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ \delta_2 & \alpha_2 & \beta_3 & & & \\ & \delta_3 & \alpha_3 & & & \\ & & & \ddots & & \\ & & & & \alpha_{k-1} & \beta_k \\ & & & & \delta_k & \alpha_k \end{bmatrix} \tag{4.12}$$

In (4.11), $\tilde{\lambda}_k$ represents the reciprocals of the approximated eigenvalues and similarly $\mathbf{Q}_k \boldsymbol{\eta}_k$ represent the right eigenvector approximations.

In order to iterate in the space of the elastic modes, the results of the inverse iteration steps, namely, $\hat{\boldsymbol{p}}_{k+1} = \mathbf{K}^{-\mathrm{T}}\mathbf{M}^{\mathrm{T}}\boldsymbol{p}_k$ and $\hat{\boldsymbol{q}}_{k+1} = \mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}_k$, are explicitly $\mathbf{M}$-orthogonalized with respect to the left and right rigid mode vectors, $\boldsymbol{\phi}_l^O$ and $\boldsymbol{\phi}_r^O$. If the stiffness matrix of the system is singular then it is not possible to invert the matrix and pseudo-inverses should be used [63].

Although, mathematically speaking, three-term recurrence relations necessitate only the orthogonalization of the current iteration vector with respect to the two previous ones, loss of orthogonality between the generated Lanczos vectors due to finite precision arithmetic on a computer is a well known fact. In the course of this chapter, a full orthogonalization scheme is followed to remedy this problem for the two-sided solver. Namely, the current iteration vectors are explicitly made $\mathbf{M}$-orthogonal to the previous vectors by using the two-sided Modified Gram Schmidt technique [120].

The convergence of the method can be controlled either with the norm of the residual of the original eigenvalue problem after recovery [63] or with a very economical residual norm provided in [118]. Namely, for the right eigenvectors, we used

$$\epsilon = \frac{||\mathbf{K}^{-1}\mathbf{M}\tilde{\boldsymbol{\phi}}_k - \tilde{\lambda}_k\tilde{\boldsymbol{\phi}}_k||}{||\tilde{\boldsymbol{\phi}}_k||} \equiv \frac{|\delta_{k+1}||\boldsymbol{\eta}_{k,i}|}{||\tilde{\boldsymbol{\phi}}_k||} \tag{4.13}$$

$\tilde{\boldsymbol{\phi}}_k$ and $\tilde{\lambda}_k$ represent the right eigenvector and the associated eigenvalue approximations for the k^{th} mode, respectively. Recalling from (4.11), $\boldsymbol{\eta}_k$ represents the k^{th} right eigenvector of the nonsymmetric tridiagonal matrix and $\boldsymbol{\eta}_{k,i}$ is its last element. In general, the convergence of the left and right eigenvectors should be monitored simultaneously. However, it was proven in [161] that the left and the right vibro-acoustic eigenvectors are related. In brief, we only monitored the convergence of the right eigenvectors in the tests. It is possible to extend the process to operate in blocks without any difficulties where the tridiagonal matrix $\mathbf{T}_k$ is generated in a block fashion. In fact, it might not always be possible to $\mathbf{M}$-biorthogonalize the vectors due to break-down, namely, $\delta_{k+1} = 0$, then restarting is necessary with new starting vectors $\boldsymbol{p}_1$ and $\boldsymbol{q}_1$ [120, 71][3]. Possible strategies to cope with these breakdown problems are proposed in [56, 159].

4.2.2. Arnoldi method

In contrast to the two-sided Lanczos solver, Arnoldi is an orthogonal projection method which can be used to compute the eigenvalues and eigenvectors of nonsymmetric matrices. In essence, it was introduced in order to transform a dense matrix into Hessenberg form [10]. In this respect, for the computation of the eigenvalues and eigenvectors of very large problems, a partial reduction to Hessenberg format is accomplished during Arnoldi iterations.

In a manner similar to the two-sided Lanczos solver outlined previously, Arnoldi iteration can also be viewed as a projection method onto successive Krylov sub-

---

[3] In the large number of numerical tests conducted, we did not experience any break-down problems.

spaces. Namely, starting from the real nonsymmetric eigenvalue problem, (4.1), and an initial vector $\boldsymbol{q}_1 \neq 0$, Arnoldi iterations build a Krylov basis spanning $\mathcal{K}^k(\mathbf{K}^{-1}\mathbf{M}, \boldsymbol{q}_1) = span\{\boldsymbol{q}_1, \mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}_1, \ldots, (\mathbf{K}^{-1}\mathbf{M})^{k-1}\boldsymbol{q}_1\} \equiv span\{\mathbf{Q}_k\}$. In contrast to the two-sided Lanczos iterations, a k-term recurrence is used in the iterations where the general recurrence relation at step k is written as,

$$h_{k+1,k}\boldsymbol{q}_{k+1} = \mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}_k - h_{1,k}\boldsymbol{q}_1 - \cdots - h_{k,k}\boldsymbol{q}_k \equiv \mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}_k - \sum_{p=1}^{k} h_{p,k}\boldsymbol{q}_p \tag{4.14}$$

where the h-terms represent the coefficients of the Hessenberg matrix. From a different perspective, the h-terms can also be viewed as the orthogonality coefficients resulting from an orthogonalization operation [136]. From (4.14), one can understand that the components of all the previous vectors should be subtracted from the result of the inverse iteration which is a serious drawback of Arnoldi type iteration methods and several restarting schemes were proposed to alleviate this problem [92]. Moreover, an incomplete orthogonalization scheme for Arnoldi method was also proposed [135]. The main idea is to perform the orthogonalization operations over a limited number of previous vectors. However, results in [135] suggest that this implementation significantly increases the number of iterations with respect to a normal Arnoldi implementation. Therefore, we did not use this approach during the tests performed in this research.

Conceptually, (4.14) can also be written in a matrix form, namely,

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_k = \mathbf{Q}_k\mathbf{H}_k + h_{k+1,k}\boldsymbol{q}_{k+1}\boldsymbol{e}_k^{\mathrm{T}} \tag{4.15}$$

where $\boldsymbol{e}_k$ is the unit vector at step k of the iteration process. And $\mathbf{H}_k$ is the upper Hessenberg matrix written as

$$\mathbf{H}_k = \begin{bmatrix} h_{1,1} & & \cdots & & h_{1,n} \\ h_{2,1} & h_{2,2} & & & h_{2,n} \\ & & & & \\ & & \ddots & & \\ & & & h_{k,k-1} & h_{k,k} \end{bmatrix} \tag{4.16}$$

At each step, (4.16) is used to find the Arnoldi eigenvalue and eigenvector estimates of the system. Similar to the two-sided Lanczos solver outlined, the convergence of the process is monitored with the residual error estimators [136] in a cheap way. Namely, to compute the residual error estimate for mode k, we can use

$$\epsilon = \frac{||\mathbf{K}^{-1}\mathbf{M}\tilde{\boldsymbol{\phi}}_k - \tilde{\lambda}_k\tilde{\boldsymbol{\phi}}_k||}{||\tilde{\boldsymbol{\phi}}_k||} \equiv \frac{|h_{k+1,k}||\boldsymbol{\phi}_{k,i}^{\mathbf{H}_k}|}{||\boldsymbol{\phi}_k^{\mathbf{H}_k}||} \tag{4.17}$$

where $\tilde{\boldsymbol{\phi}}_k$ and $\tilde{\lambda}_k$ represent the eigenvector and reciprocal eigenvalue approximations, respectively. We recover the eigenvector approximations with $\tilde{\boldsymbol{\phi}}_k = \mathbf{Q}_k\boldsymbol{\phi}_k^{\mathbf{H}_k}$ where $\mathbf{Q}_k$ represent the L^2-orthogonal Arnoldi vectors and $\boldsymbol{\phi}_k^{\mathbf{H}_k}$ is the k^{th} eigenvector of (4.16). L^2-orthogonality is used to simplify the denominator in (4.17), namely, $||\tilde{\boldsymbol{\phi}}_k|| = \sqrt{{\boldsymbol{\phi}_k^{\mathbf{H}_k}}^{\mathrm{T}}\mathbf{Q}_k^{\mathrm{T}}\mathbf{Q}_k\boldsymbol{\phi}_k^{\mathbf{H}_k}} \equiv ||\boldsymbol{\phi}_k^{\mathbf{H}_k}||$. Moreover, $\boldsymbol{\phi}_{k,i}^{\mathbf{H}_k}$ is the last element of $\boldsymbol{\phi}_k^{\mathbf{H}_k}$.

4.2.3. Exploiting the matrix topology for computations

In the implementations of the two-sided Lanczos and Arnoldi methods, we have to perform the factorization of $\mathbf{K}^{\mathrm{T}}$ and $\mathbf{K}$. However, by looking at the nonsymmetric stiffness matrix of the problem defined in (4.1), we can easily realize a computationally advantageous fact that both of these matrices have a large zero-block which is quite useful in order to decrease the computational cost of the factorization operations. Namely, for the right basis, writing a simple linear system of equations with $\mathbf{K}$ results in,

$$\begin{bmatrix} \mathbf{K}_s & -\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_s \\ \boldsymbol{x}_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_s \\ \boldsymbol{b}_f \end{bmatrix} \tag{4.18}$$

It is possible to separate the rows of (4.18) and come up with two one-way coupled system of equations, namely,

$$\mathbf{K}_s \boldsymbol{x}_s - \mathbf{K}_{sf} \boldsymbol{x}_{\boldsymbol{f}} = \boldsymbol{b}_s, \tag{4.19}$$

$$\mathbf{K}_f \boldsymbol{x}_f = \boldsymbol{b}_f, \tag{4.20}$$

which can be solved with the independent factorizations of the individual stiffness matrices of the participating physics, namely, $\mathbf{K}_s$ and $\mathbf{K}_f$. With this scheme, it is possible to circumvent the factorization of the full nonsymmetric stiffness matrix. Moreover, the same principle equally applies when solving the linear systems with $\mathbf{K}^{\mathrm{T}}$. Therefore, the special topology of the system can be used to result in a very efficient solution scheme for the problem at hand due to the independent block factorizations. In the case of closed volumes, $\mathbf{K}_f$ is singular whereas $\mathbf{K}_s$ is generally regular due to the application of proper boundary conditions. In these kinds of problems, a pseudo inversion is necessary for solving (4.20) [49].

4.3. Symmetric forms of the nonsymmetric problem

Approaches to create symmetric representations of the nonsymmetric coupled vibroacoustic problem defined in equation (4.1) exist and these are outlined in [52] and [53] which also support the explanations with a theorem provided in [118]. In the light of these references, matrix scaling and eigenvector augmentation methods were used respectively to end up with symmetric representations of the original nonsymmetric coupled eigenvalue problem.

[52] presents four scaling matrices that could result in a symmetric representation of the original nonsymmetric problem. Namely, two pre-multiplier and two post-multiplier matrices are shown that could result in a symmetric system representation. The pre-multiplier matrices are denoted by κ_1 and κ_2, respectively. Similarly, post-multiplier matrices are called τ_1 and τ_2[4], namely,

$$\kappa_1 = \begin{bmatrix} \mathbf{K}_s \mathbf{M}_s^{-1} & \mathbf{0} \\ -\mathbf{K}_{sf}^{\mathrm{T}} \mathbf{M}_s^{-1} & \mathbf{I} \end{bmatrix}, \qquad \kappa_2 = \begin{bmatrix} \mathbf{I} & \mathbf{K}_{sf} \mathbf{K}_f^{-1} \\ \mathbf{0} & \mathbf{M}_f \mathbf{K}_f^{-1} \end{bmatrix} \tag{4.21}$$

---

[4] It must be noted here that which of the transformations will be used later will depend on the properties of the matrices used to develop the algorithms.

$$\tau_1 = \begin{bmatrix} \mathbf{K}_s^{-1}\mathbf{M}_s & \mathbf{K}_s^{-1}\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \qquad \tau_2 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{M}_f^{-1}\mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f^{-1}\mathbf{K}_f \end{bmatrix} \tag{4.22}$$

It is possible to use each of these transformation matrices in order to come up with an efficient iteration scheme. By using either κ_1 or κ_2, we are not changing the space of the iteration vectors since it is a pre-multiplication from the left. In contrast, if τ_1 is selected as a post-multiplier matrix, then the resulting vectors change their corresponding basis, namely

$$\begin{bmatrix} \boldsymbol{\phi}_s \\ \boldsymbol{\phi}_f \end{bmatrix} = \tau_1 \begin{bmatrix} \tilde{\boldsymbol{\phi}}_s \\ \boldsymbol{\phi}_f \end{bmatrix} \tag{4.23}$$

In closed form, the system matrices built up with either κ_1 or κ_2 can be written respectively as,

$$\left(\begin{bmatrix} (\mathbf{K}_s\mathbf{M}_s^{-1}\mathbf{K}_s & -\mathbf{K}_s^{-1}\mathbf{M}_s^{-1}\mathbf{K}_{sf} \\ -\mathbf{K}_{sf}^{\mathrm{T}}\mathbf{M}_s^{-1}\mathbf{K}_s & (\mathbf{K}_f + \mathbf{K}_{sf}^{\mathrm{T}}\mathbf{M}_s^{-1}\mathbf{K}_{sf}) \end{bmatrix} - \lambda \begin{bmatrix} \mathbf{K}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_f \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{\phi}_s \\ \boldsymbol{\phi}_f \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{4.24}$$

$$\left(\begin{bmatrix} \mathbf{K}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_f \end{bmatrix} - \lambda \begin{bmatrix} (\mathbf{M}_s + \mathbf{K}_{sf}\mathbf{K}_f^{-1}\mathbf{K}_{sf}^{\mathrm{T}}) & \mathbf{K}_{sf}\mathbf{K}_f^{-1}\mathbf{M}_f \\ \mathbf{M}_f\mathbf{K}_f^{-1}\mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f\mathbf{K}_f^{-1}\mathbf{M}_f \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{\phi}_s \\ \boldsymbol{\phi}_f \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{4.25}$$

Looking at (4.24) and (4.25), it is easy to realize that either the stiffness matrix or the mass matrix become fully populated when κ_1 or κ_2 is used, respectively. (4.24) and (4.25) are provided here for convenience in order to show the structure of the resulting symmetric system matrices. Due to the fully populated nature of either the stiffness or mass matrices, explicit construction and numerical implementation is not a feasible approach. In Section 4.4, we are going to provide an analysis showing that it is not necessary to set up neither of these forms explicitly.

4.4. Symmetric *vibro*-Lanczos algorithm

4.4.1. Overview of the symmetric Lanczos eigensolver

The generalized eigenvalue problem with symmetric and real system matrices is written as

$$\mathbf{K}_{sym}\boldsymbol{\phi}_{sym} = \lambda \mathbf{M}_{sym}\boldsymbol{\phi}_{sym} \tag{4.26}$$

Starting with an initial vector $\boldsymbol{q}_1 \neq 0$, Lanczos iterations build up a Krylov basis which spans $\mathcal{K}^k(\mathbf{K}_{sym}^{-1}\mathbf{M}_{sym}, \boldsymbol{q}_1) = span\{\boldsymbol{q}_1, \mathbf{K}_{sym}^{-1}\mathbf{M}_{sym}\boldsymbol{q}_1, \ldots, (\mathbf{K}_{sym}^{-1}\mathbf{M}_{sym})^{k-1}\boldsymbol{q}_1\} \equiv span\{\mathbf{Q}_k\}$. In exact arithmetic, due to the symmetric matrices, recursive orthogonality relations apply and the main three-term recurrence relation can be written as [136]

$$\tilde{\boldsymbol{q}}_{k+1} = \mathbf{K}_{sym}^{-1}\mathbf{M}_{sym}\boldsymbol{q}_k - \alpha_k\boldsymbol{q}_k - \beta_k\boldsymbol{q}_{k-1} \tag{4.27}$$

At each step of the iteration process, a new vector is generated with one inverse iteration step and some additional vector operations. α_k and β_{k-1} represent the $\mathbf{M}_{sym}$-orthogonalization coefficients. By using (4.27), one is implicitly generating the coefficients of a symmetric tridiagonal matrix column by column. With the use of these orthogonalization coefficients, the original symmetric eigenvalue problem representation is transformed into a reduced symmetric tridiagonal eigenvalue problem. As a result, at step k of the shortly outlined process, one ends up with

$$\mathbf{T}_k \boldsymbol{\eta} = \tilde{\lambda}_k \boldsymbol{\eta}_k \tag{4.28}$$

where the symmetric tridiagonal matrix is written as

$$\mathbf{T}_k = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & & & \\ & & & \ddots & & \\ & & & & \alpha_{k-1} & \beta_k \\ & & & & \beta_k & \alpha_k \end{bmatrix} \tag{4.29}$$

$\tilde{\lambda}_k$ represents the reciprocals of the approximated eigenvalues and $\mathbf{Q}_k \boldsymbol{\eta}_k$ represent the eigenvector approximations. Similar to the nonsymmetric solvers, the convergence of the method can be controlled either with the norm of the residual of the original eigenvalue problem after recovery [63] or with a very economical residual norm provided in [118]. It is also possible to extend the symmetric version to operate in blocks without any difficulties.

The procedure is summarized in Algorithm 4.1 starting from $\tilde{\boldsymbol{q}}_1 \neq \mathbf{0}$. $\mathbf{U}$ represents the rigid body vector block if there are any. If no rigid body modes are present, this step should be skipped. For numerical stability reasons, we used the modified Gram-Schmidt algorithm (between lines 14 and 17). There is a subtle point which needs further explanation in Algorithm 4.1. Namely, between lines 11 and 13, the result of the inverse iteration vector is orthogonalized with respect to vectors $\boldsymbol{q}_k$ and $\boldsymbol{q}_{k-1}$. Considering this fact, the full reorthogonalization given on line 14 should only extend up to vector $\boldsymbol{q}_{k-2}$. However, we observed that performing the orthogonalization with $\boldsymbol{q}_k$ and $\boldsymbol{q}_{k-1}$ once more during the full reorthogonalization as shown in Algorithm 4.1 gives better results.

4.4.2. Symmetric Lanczos iterations for vibro-acoustics

The symmetric eigenvalue problem of vibro-acoustics was provided in (4.24) or (4.25) in closed form. An efficient solution scheme is detailed here which is used to perform the inverse iteration step of the Lanczos iterations for the symmetric version of the eigenvalue problem.

In Arnoldi iterations or similarly for the right vectors of the two-sided Lanczos iterations, a new vector is generated at the inverse iteration step, namely, $\hat{\boldsymbol{q}}_{k+1} = \mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}_k$. The same analysis is also valid for the system equations that are built up with either κ_1 or κ_2. In the discussion, κ_1 is used and the symmetric form of the

Algorithm 4.1 Symmetric Lanczos algorithm

```
 1: $\tilde{\boldsymbol{q}}_1 \neq \mathbf{0}$
 2: $\tilde{\boldsymbol{q}}_1 = \tilde{\boldsymbol{q}}_1 - \mathbf{U}\mathbf{U}^{\mathrm{T}}\mathbf{M}_{sym}\tilde{\boldsymbol{q}}_1$          ▷ Filter the rigid body modes
 3: $\boldsymbol{z} = \mathbf{M}_{sym}\tilde{\boldsymbol{q}}_1$          ▷ scale vector $\tilde{\boldsymbol{q}}_1$:$\boldsymbol{q}_1^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_{\mathbf{1}} = 1$
 4: $d = \sqrt{(\tilde{\boldsymbol{q}}_1^{\mathrm{T}}\boldsymbol{z})}, \beta_1 = d$
 5: $\boldsymbol{q}_1 = \tilde{\boldsymbol{q}}_1/d$
 6: $\mathbf{Q} \leftarrow \boldsymbol{q}_1$          ▷ Add vector to Lanczos vector basis
 7: for $(k = 1, 2, ...)$ do          ▷ Loop until convergence
 8:     $\boldsymbol{R}_k^i = \mathbf{M}_{sym}\mathbf{q}_k \equiv \boldsymbol{z}/d$          ▷ Create inertial force
 9:     $\hat{\boldsymbol{q}}_{k+1} = \mathbf{K}_{sym}^{-1}\boldsymbol{R}_k^i$          ▷ Inverse iteration step, if $\mathbf{K}_{sym}$ is singular, use $\mathbf{K}_{sym}^{+}$
10:     $\hat{\boldsymbol{q}}_{k+1} = \hat{\boldsymbol{q}}_{k+1} - \mathbf{U}\mathbf{U}^{\mathrm{T}}\mathbf{M}_{sym}\hat{\boldsymbol{q}}_{k+1}$          ▷ Filter the rigid body modes
11:     $\hat{\boldsymbol{q}}_{k+1} = \hat{\boldsymbol{q}}_{k+1} - \beta_k\boldsymbol{q}_{k-1}$          ▷ Apply the 3-term recurrence
12:     $\alpha_k = \hat{\boldsymbol{q}}_{k+1}^{\mathrm{T}}\boldsymbol{R}_k^i$
13:     $\hat{\boldsymbol{q}}_{k+1} = \hat{\boldsymbol{q}}_{k+1} - \alpha_k\boldsymbol{q}_k$
14:     for $j = 1, ..., k$ do
15:         $a = {\boldsymbol{R}_j^i}^{\mathrm{T}}\hat{\boldsymbol{q}}_{k+1}$
16:         $\hat{\boldsymbol{q}}_{k+1} = \hat{\boldsymbol{q}}_{k+1} - a\boldsymbol{q}_j$
17:     end for
18:     $\boldsymbol{z} = \mathbf{M}_{sym}\tilde{\boldsymbol{q}}_{k+1}$          ▷ scale vector $\tilde{\boldsymbol{q}}_{k+1}$:$\boldsymbol{q}_{k+1}^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_{k+1} = 1$
19:     $\beta_{k+1} = \sqrt{(\tilde{\boldsymbol{q}}_{\mathbf{k+1}}^{\mathrm{T}}\boldsymbol{z})} \equiv d$
20:     $\boldsymbol{q}_{k+1} = \tilde{\boldsymbol{q}}_{k+1}/\beta_{k+1}$
21:     $\mathbf{T} \leftarrow \alpha_k, \beta_{k+1}$          ▷ Fill the tridiagonal matrix and compute error
22:     if Converged then          ▷ Convergence check after some iterations
23:         output eigenvalues/vectors and stop
24:     end if
25:     $\mathbf{Q} \leftarrow \boldsymbol{q}_{k+1}$          ▷ Add new vector to Lanczos vector block
26: end for
```

inverse iteration equation is written as (line 8 and 9 in Algorithm 4.1),

$$\kappa_1\mathbf{K}\hat{\boldsymbol{q}}_{k+1} = \kappa_1\mathbf{M}\boldsymbol{q}_k \tag{4.30}$$

Since κ_1 is a regular pre-multiplication operator, we can conclude that the results of the inverse iteration operations for Arnoldi, for the right vectors of the two-sided Lanczos method and for the symmetric variant proposed here are the same. And they all can be computed with the equations provided in (4.19) and (4.20).

In this respect, considering (4.19) and (4.20) while forming the inverse iteration equation as $\hat{\boldsymbol{q}}_{k+1} = \mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}_k$, we can write the fluid partition of the improved vector independently from row 2 as,

$$\hat{\boldsymbol{q}}_{k+1,f} = \mathbf{K}_f^{-1}\left(\mathbf{K}_{sf}^{\mathrm{T}}\boldsymbol{q}_{k,s} + \mathbf{M}_f\boldsymbol{q}_{k,f}\right) \tag{4.31}$$

And, therefore, by using (4.19), it is possible to write the structural part of the solution vector as

$$\hat{\boldsymbol{q}}_{k+1,s} = \mathbf{K}_s^{-1}\left(\mathbf{M}_s\boldsymbol{q}_{k,s} + \mathbf{K}_{sf}\hat{\boldsymbol{q}}_{k+1,f}\right) \tag{4.32}$$

In brief, the equations to be solved for the two uncoupled physics are summarized as,

$$\hat{\boldsymbol{q}}_{k+1,f} = \mathbf{K}_f^{-1}\left(\mathbf{K}_{sf}^{\mathrm{T}}\boldsymbol{q}_{k,s} + \mathbf{M}_f\boldsymbol{q}_{k,f}\right) \tag{4.33}$$

$$\hat{\boldsymbol{q}}_{k+1,s} = \mathbf{K}_s^{-1}\left(\mathbf{M}_s\boldsymbol{q}_{k,s} + \mathbf{K}_{sf}\hat{\boldsymbol{q}}_{k+1,f}\right) \tag{4.34}$$

A special treatment is necessary for the solution of (4.33) if the operator matrix is singular. In those cases, the solution of (4.33) is only possible if the right hand side vector is self-equilibrated [63]. Following the generation of the self-equilibrated load, a constrained problem is solved with this self-equilibrated load vector. The constrained problem is generated by deleting the last row and column in the singular $\mathbf{K}_f$ to render it nonsingular. This constrained matrix is represented with $\mathbf{K}_{f,11}$. This concept is also referred to as inertia-relief in the field of substructuring [36].

The symmetric Lanczos process generates an $\mathbf{M}_{sym}$-orthogonal Lanczos vector set where the vectors are explicitly multiplied by the mass matrix as shown on lines 8 and 18 of Algorithm 4.1. However, Algorithm 4.1 is only given here in order to provide an overview for the symmetric Lanczos algorithm and to stress out the modifications for vibro-acoustic eigenvalue problems. In the context of this chapter, we should mention that the inverse iteration steps, namely, lines 8 and 9, are performed with the nonsymmetric matrices since they produce the same result, namely, $\mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}_k$. These equations were provided in (4.33) and (4.34).

With the symmetric system representation formed with κ_1, the mass matrix of the system is in block diagonal format which can be observed in (4.24). Namely,

$$\mathbf{M}_{sym} = \kappa_1\mathbf{M} \equiv \begin{bmatrix} \mathbf{K}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_f \end{bmatrix} \tag{4.35}$$

Although the mass matrix built with κ_2 transformation is dense, the mass matrix built with κ_1 is block diagonal. In this format, we can generate $\mathbf{M}_{sym}$-orthogonal Lanczos vectors efficiently. We can summarize the important operational points as follows:

- Inverse iteration is performed with $\mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}_k$.
- In order to $\mathbf{M}_{sym}$-scale the vectors on lines 3 or 18, we use $\kappa_1\mathbf{M}$ which is block diagonal.
- The resulting matrix-vector multiplications performed during the $\mathbf{M}_{sym}$-scaling operations are stored for the computation of α_k coefficients. Later, these are also used for the reorthogonalization operations.

Moreover, the norm inherent to the orthogonalization performed with respect to $\mathbf{M}_{sym}$ in the proposed approach performs a scaling and treats the pressure and displacement variables in an energy-like norm. The L^2-orthogonality of the Arnoldi process mixes the pressure and displacement variables which could lead to ill-conditioning during the orthogonalization process.

We can summarize the similarities and differences of Arnoldi and the symmetric approach proposed here as follows:

- The results of the inverse iteration steps for Arnoldi and the symmetric Lanczos are the same due to the pre-multiplication operations. Hence both methods generate the same same Krylov subspace.

- The vectors, $\boldsymbol{q}_k$, generated during the complete iteration cycle will be different because they satisfy different orthogonality relations. Namely, for the symmetric approach, they satisfy $\mathbf{M}_{sym}$-orthogonality, whereas, for Arnoldi, they are L^2-orthogonal. However, the subspaces generated in both cases are equivalent.

- For the Arnoldi algorithm, full orthogonalization is required. Whereas, a tri-orthogonality relation exists for the symmetric Lanczos process, at least in theory. This relation can be used in a partial reorthogonalization scheme which is going to be the subject of Section 4.4.5.

- In the symmetric case, a symmetric reduced tridiagonal is formed since the problem is projected onto a symmetric representation implicitly. In contrast, Arnoldi generates an upper Hessenberg matrix whereas two-sided Lanczos algorithm generates a nonsymmetric tridiagonal. Moreover, the transformation to a symmetric representation brings no extra costs in comparison to Arnoldi.

4.4.3. Details on filtering rigid modes for the symmetric representation

Computing the right and left rigid modes

To compute the right rigid mode of the system, one should solve

$$\begin{bmatrix} \mathbf{K}_s & -\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_s^{Rigid} \\ \boldsymbol{\phi}_f^{Rigid} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{4.36}$$

For a closed volume, the stiffness matrix of the fluid side of the system is singular which means that $\boldsymbol{\phi}_f^{Rigid}$ corresponds to the null-space of $\mathbf{K}_f$.

The set of all m independent solutions, $\boldsymbol{r}_i$, of $\mathbf{K}_f$,

$$\mathbf{K}_f \boldsymbol{r}_i = \mathbf{0} \qquad i = 1, 2, ..., m, \tag{4.37}$$

forms the null space of $\mathbf{K}_f$.

If m null space vectors $\boldsymbol{r}_i$ for a matrix $\mathbf{K}_f$ of dimension $n \times n$ exist, then in (4.37), only $n - m$ rows and columns are linearly independent. The null space of $\mathbf{K}_f$ is, in this case, composed of only one vector which is the so-called rigid body mode of the fluid domain exhibiting a uniform pressure at all nodes . One can assume to partition the system in the light of this discussion as,

$$\begin{bmatrix} \mathbf{K}_f^{11} & \mathbf{K}_f^{12} \\ \mathbf{K}_f^{21} & \mathbf{K}_f^{22} \end{bmatrix} \tag{4.38}$$

where $\mathbf{K}_f^{11}$ represents the non-singular square matrix whose dimension is equal to the rank of $\mathbf{K}_f$ which is in this case $n - 1$. Mathematically speaking, $\mathbf{K}_f^{11}$ is a matrix of size $(n-1)\times(n-1)$ and $\mathbf{K}_f^{12}$ is a column vector of size $(n-1)\times 1$. Additionally, $\mathbf{K}_f^{22}$ is a matrix of size 1×1 which is a scalar for this special setting. Therefore, the null space is written as [49],

$$\boldsymbol{r}_1 = \boldsymbol{\phi}_f^{Rigid} \equiv \begin{bmatrix} -\mathbf{K}_f^{11^{-1}}\mathbf{K}_f^{12} \\ 1 \end{bmatrix} \tag{4.39}$$

Having found $\boldsymbol{\phi}_f^{Rigid}$, one could use (4.36) to compute the structural part of the right rigid mode vector, namely,

$$\boldsymbol{\phi}_s^{Rigid} = \mathbf{K}_s^{-1}\mathbf{K}_{sf}\boldsymbol{\phi}_f^{Rigid} \tag{4.40}$$

With this computational overview, the right rigid body mode vector is written as,

$$\boldsymbol{\phi}_r^O = \begin{bmatrix} \mathbf{K}_s^{-1}\mathbf{K}_{sf}\boldsymbol{\phi}_f^{Rigid} \\ \boldsymbol{\phi}_f^{Rigid} \end{bmatrix} \tag{4.41}$$

To compute the left rigid mode vector, the system has to be transposed to solve for the left rigid mode vector, namely, we have to solve

$$\begin{bmatrix} \mathbf{K}_s & \mathbf{0} \\ -\mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{K}_f \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\phi}}_s^{Rigid} \\ \tilde{\boldsymbol{\phi}}_f^{Rigid} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{4.42}$$

where $\tilde{\bullet}$ is used to represent the left rigid body mode vector. Using the block structure of (4.42), one could write

$$\mathbf{K}_s\tilde{\boldsymbol{\phi}}_s^{Rigid} = \mathbf{0} \tag{4.43}$$

$$-\mathbf{K}_{sf}^{\mathrm{T}}\tilde{\boldsymbol{\phi}}_s^{Rigid} + \mathbf{K}_f\tilde{\boldsymbol{\phi}}_f^{Rigid} = \mathbf{0} \tag{4.44}$$

resulting in

$$\boldsymbol{\phi}_l^O = \begin{bmatrix} \tilde{\boldsymbol{\phi}}_s^{Rigid} \\ \tilde{\boldsymbol{\phi}}_f^{Rigid} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\phi}_f^{Rigid} \end{bmatrix} \tag{4.45}$$

The rigid body vectors are scaled with respect to the mass matrix such that $\boldsymbol{\phi}_l^{O^{\mathrm{T}}}\mathbf{M}\boldsymbol{\phi}_r^O = 1$.

Filtering the rigid mode

In order to iterate in the space of the elastic modes, the zero-frequency pseudo mode vector has to be filtered out from the solution space. Since the iterations advance by iterating over the right Krylov subspace of the original nonsymmetric problem, the right pseudo-mode vector has to be filtered out from iteration vectors that result from the inverse iteration step[5]. Namely,

$$\hat{\boldsymbol{q}}_{k+1} \leftarrow (\mathbf{I} - \boldsymbol{\phi}_r^O\boldsymbol{\phi}_l^{O^{\mathrm{T}}}\mathbf{M})\hat{\boldsymbol{q}}_{k+1} \equiv (\mathbf{I} - \boldsymbol{\phi}_r^O\boldsymbol{\phi}_r^{O^{\mathrm{T}}}\mathbf{M}_{sym})\hat{\boldsymbol{q}}_{k+1} \tag{4.46}$$

due to $\boldsymbol{\phi}_l^O = \kappa_1^{\mathrm{T}}\boldsymbol{\phi}_r^O$ and $\mathbf{M}_{sym} = \kappa_1\mathbf{M}$. In (4.46), $\boldsymbol{\phi}_l^O$ and $\boldsymbol{\phi}_r^O$ represent the left and right zero-frequency pseudo mode vectors, respectively.

---

[5] Similar filtering operations were also used in the implementations of the two-sided Lanczos and Arnoldi methods.

4.4.4. Convergence check

Similar to the outlined nonsymmetric alternatives, the convergence of the symmetric method is also monitored with a relative residual error estimate [63],

$$\epsilon = \frac{||\mathbf{K}^{-1}\mathbf{M}\tilde{\boldsymbol{\phi}}_k - \tilde{\lambda}_k\tilde{\boldsymbol{\phi}}_k||_{\mathbf{M}}}{||\tilde{\boldsymbol{\phi}}_k||_{\mathbf{M}}} \equiv \frac{\beta_{i+1}|\boldsymbol{\phi}^{\mathbf{T}_k}_{k,i}|}{||\boldsymbol{\phi}^{\mathbf{T}_k}_k||} \tag{4.47}$$

where all the matrices are considered to be the symmetric in this case. Besides, $\tilde{\boldsymbol{\phi}}_k$ and $\tilde{\lambda}_k$ represent the eigenvector and reciprocal eigenvalue approximations, respectively. It is worth noting that the denominator in (4.47) was simplified by using $\tilde{\boldsymbol{\phi}}_k = \mathbf{Q}_k\boldsymbol{\phi}^{\mathbf{T}_k}_k$ and $\mathbf{M}_{sym}$-orthogonality among the Lanczos vectors, $\mathbf{Q}_k$. Namely, $||\tilde{\boldsymbol{\phi}}_k||_{\mathbf{M}} = \sqrt{\boldsymbol{\phi}^{\mathbf{T}_k}_k{}^{\mathrm{T}}\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}\mathbf{Q}_k\boldsymbol{\phi}^{\mathbf{T}_k}_k} \equiv ||\boldsymbol{\phi}^{\mathbf{T}_k}_k||$. $\boldsymbol{\phi}^{\mathbf{T}_k}_k$ represents the k^{th} eigenvector of the symmetric tridiagonal problem and $\boldsymbol{\phi}^{\mathbf{T}_k}_{k,i}$ is its last element.

4.4.5. Cost reduction with partial or selective orthogonalization

Due to the finite precision computations performed on a computer, the symmetric Lanczos method is advised to be used with full reorthogonalization where theoretical three-term recurrence relations are not enough to keep the orthogonality among the Lanczos vectors [62]. However, several partial and selective orthogonalization schemes were proposed in the literature in order to decrease the cost of orthogonalization operations during the symmetric Lanczos iterations [118, 142, 141]. Even though the orthogonalization is incomplete in these procedures, accurate eigenvalues and eigenvectors can be found.

A brief outline for these methods is provided next [119]:

- Selective: If $\boldsymbol{q}$ is the current Lanczos vector and $\mathbf{K}^{-1}\mathbf{M}$ is the operator, then one has to orthogonalize $\mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}$ against the two earlier Lanczos vectors. It is also known that
    - this is good enough provided the Krylov subspace has not yet captured any eigenvector,
    - if the Krylov subspace is big enough to contain a few eigenvectors, then $\mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}$ will have significant components in those eigenvectors. If eigenvectors are recovered by transforming to the physical space with $\mathbf{Q}_k\eta_k$ as soon as convergence is detected then it might be much faster to orthogonalize against the converged eigenvectors rather than with respect to all the Lanczos vectors. However, this practice brings the need to solve the tridiagonal eigenvalue problem at each step. To decide on this, the recurrence developed in [118] can be used.
- Partial: If the Krylov subspace is being constructed without computing eigenvectors as soon as convergence is detected, then selective reorthogonalization is not an option. However, it is observed that the orthogonality is being lost in a systematic way [142]. In [142], a recurrence relation is shown which predicts

> when orthogonality was lost at the $\sqrt{eps}$[6]. At that point the new Lanczos iteration vector $\boldsymbol{q}$ are reorthogonalized against all previous ones. It was also proven in [141] that if $\|\mathbf{I}-\mathbf{Q}^T\mathbf{M}_{sym}\mathbf{Q}\| < \sqrt{eps}$ then the resulting tridiagonal matrix, $\mathbf{T}_k$, computed with the symmetric Lanczos algorithm is correct up to machine precision *eps*, although the Lanczos vectors themselves are not orthogonal to working precision. In conclusion, semi-orthogonality was reported to be good enough for computations. However, in our numerical tests, we have experienced that the $\sqrt{eps}$ bound sometimes results in double frequencies for ill-conditioned problems. Therefore, we have tightened for an order depending on the machine precision. Namely, in the machines that we performed our tests $\sqrt{eps}$ was on the order of 10^{-8} and we used an orthogonality tolerance of 10^{-9} instead. This is also inline with the experience and informal analysis performed in [118].

Although it was proven in Section 4.4 that Arnoldi and symmetric Lanczos iterate in the same space, there is a subtle point that we can use advantageously in the case of iterations with a symmetric system. Namely, in the Arnoldi method as outlined in Section 4.2.2, all the orthogonality coefficients should be computed accurately in order to represent the reduced upper Hessenberg matrix. Otherwise the computation of the eigenvalues and eigenvectors with the reduced upper Hessenberg matrix will be affected and wrong results might emerge. Similar to Arnoldi, coefficients that are used to form the reduced symmetric tridiagonal matrix are also the orthogonalization coefficients. But in the symmetric case, most of these orthogonality coefficients should vanish in order to end up with a symmetric tridiagonal matrix and this is explicitly forced with the full orthogonalization scheme. However, as outlined in [142], [110] and briefly above, we can keep the Lanczos vectors orthogonal up to the order of 10^{-9} and still come up with accurate results in the symmetric cases. In this work, partial reorthogonalization is used during the numerical tests.

This can be accomplished with a recurrence relation provided in [142, 110] (see also Appendix D) and modified here for $\mathbf{M}_{sym}$-orthogonality, namely,

$$\beta_{k+1}\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_{k+1} = (\mathbf{T}_k - \alpha_k\mathbf{I})\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_k - \beta_k\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_{k-1}, \quad k \geq 2 \tag{4.48}$$

In (4.48), $\mathbf{T}_k$ is the symmetric tridiagonal formed at step k. $\mathbf{M}_{sym}\mathbf{Q}_k$ represents the matrix-vector multiplications of the Lanczos vectors stored during the previous iterations. Theoretically $\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_{k+1}$ should be zero, but in finite arithmetic it is not. The orthogonality error at step $k+1$ (left-hand side of (4.48)) can be estimated recursively (right-hand side of (4.48)), based on the previous orthogonality coefficient vectors $\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_k$ and $\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_{k-1}$. (4.48) is a result of the original three-term recurrence formula devised in [142] which is used to monitor loss of orthogonality among the Lanczos vectors. In the original theory [142], (4.48) includes additional terms related to round-off errors, however, for our implementations, those terms were neglected. More mathematical details on the derivation of (4.48) are presented in Appendix D.

---

[6] eps represents the machine precision which was approximately 2.22×10^{-16} during the tests.

Moreover, [142] advises to perform a full reorthogonalization at steps k and $k+1$ if $\|\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_{k+1}\|_\infty > 10^{-9}$ at step k is satisfied. This practice is used in the numerical implementations in this chapter to keep the coefficients at a small level for the next few iteration cycles. Detailed proofs on (4.48) are provided in [142] for L^2-orthogonality.

The same kind of recurrence relations can also be written in the case of block iterations [66] however the algebra becomes more complex than it is for the single vector iterations. Therefore, in the course of the current research, block iterations were not considered.

4.4.6. Implementation details for partial reorthogonalization

As outlined in Section 4.4.5, (4.48) is a cheap way to estimate the orthogonality coefficient vector, $\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\hat{\boldsymbol{q}}_{k+1}$, which is used to monitor the loss of orthogonality among Lanczos vectors without explicitly forming these products. In this small section, we are going to show the implementation details of (4.48) used during the numerical tests. As mentioned in Section 4.4.5, (4.48) is valid from step 2 of the Lanczos process. For the implementation of the Lanczos process, we divided the complete iteration cycle into two blocks. Namely, the first block consists of the first iteration of the Lanczos iteration process where vectors $\mathbf{Q}_2^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_2$ and $\mathbf{Q}_2^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_1$ are initialized in order to provide the starting vectors for the recurrence provided in (4.48). It is important to note that after the first iteration step we are sure that $\mathbf{M}_{sym}$-orthogonality among Lanczos vectors, $\boldsymbol{q}_2$ and $\boldsymbol{q}_1$, is not lost since the orthogonality was forced explicitly. Therefore, it is safe to explicitly write

$$\mathbf{Q}_2^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_2 = [eps, 1]^{\mathrm{T}}, \quad \mathbf{Q}_2^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_1 = [1, eps]^{\mathrm{T}} \tag{4.49}$$

where $\mathbf{Q}_2$ represents the block of Lanczos vectors, namely, $\mathbf{Q}_2 = [\boldsymbol{q}_1, \boldsymbol{q}_2]$.

The second block of the iteration process starts at step 2 in (4.48) where the Lanczos iterations advance as outlined before. During the iterations, loss of orthogonality is continuously monitored by using (4.48). Writing (4.48) explicitly for $k = 2$ gives

$$\beta_3\mathbf{Q}_2^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_3 = (\mathbf{T}_2 - \alpha_2\mathbf{I})\mathbf{Q}_2^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_2 - \beta_2\mathbf{Q}_2^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_1 \tag{4.50}$$

from which the orthogonality prediction vector $\mathbf{Q}_2^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_3$ can be computed for the new vector $\boldsymbol{q}_3$. In order to generalize and show the details on how we decide to perform a reorthogonalization [110], we repeat (4.48) with some additional vector notations, namely,

$$\beta_{k+1}\underbrace{\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_{k+1}}_{\boldsymbol{X}_k} = (\mathbf{T}_k - \alpha_k\mathbf{I})\underbrace{\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_k}_{\boldsymbol{R}_k} - \beta_k\underbrace{\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_{k-1}}_{\boldsymbol{Z}_k} \tag{4.51}$$

Decision on reorthogonalization and the update of the vectors used in (4.51) are performed by using the pseudo-code provided in Algorithm 4.2. $\mathcal{E}$ is a vector of appropriate size having all its elements set to *eps*. As suggested in [142], if a need for reorthogonalization is detected at step k then a full reorthogonalization is also

Algorithm 4.2 Vector updates and reorthogonalization at step k

```
if $\|\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_{k+1}\|_\infty \leq 10^{-9}$ then
    No reorthogonalization, $\boldsymbol{R}_{k+1} = [\boldsymbol{X}_k, 1]^{\mathrm{T}}$, $\boldsymbol{Z}_{k+1} = [\boldsymbol{R}_k, eps]^{\mathrm{T}}$
else
    Full reorthogonalization, $\boldsymbol{R}_{k+1} = [\mathcal{E}, 1]^{\mathrm{T}}$, $\boldsymbol{Z}_{k+1} = [\boldsymbol{R}_k, eps]^{\mathrm{T}}$
end if
```

performed at step $k+1$. After this second reorthogonalization, the contents of the vectors $\boldsymbol{R}_k$ and $\boldsymbol{Z}_k$ are set as follows: $\boldsymbol{R}_k = [\mathcal{E}, 1]^{\mathrm{T}}$, $\boldsymbol{Z}_k = [\mathcal{E}^*, 1, eps]^{\mathrm{T}}$ where $\mathcal{E}^*$ is a vector similar to $\mathcal{E}$ with one less element.

In brief, by using (4.51), we can come up with cheap estimates of the orthogonality coefficients without explicitly computing them. Coefficient vectors used on the right side of (4.51) are also updated cheaply as outlined in Algorithm 4.2. Moreover, no extra matrix-vector multiplications are performed during these predictions.

4.4.7. Remarks on spectral transformation with symmetry

In order to accelerate convergence or to extract eigenvalues in a specific part of the spectrum, shifting can be applied on the original problem [45]. By selecting a shift value of σ, the original problem can be written as

$$(\mathbf{K} - \sigma\mathbf{M})\boldsymbol{\phi} = \hat{\lambda}\mathbf{M}\boldsymbol{\phi}, \quad \hat{\lambda} = \lambda - \sigma \tag{4.52}$$

λ represents the original eigenvalues of the problem whereas $\hat{\lambda}$ represents the shifted ones. In the case of the nonsymmetric problem representation, the application of a shift introduces an important difference. Namely, we can not represent the shifted stiffness matrix of the system in a block diagonal fashion. However, the symmetry relations still apply but the stiffness matrix includes terms on the off-diagonal blocks. Mathematically speaking, the eigenvalue problem is written as,

$$\kappa_1(\mathbf{K} - \sigma\mathbf{M})\boldsymbol{\phi} = \hat{\lambda}\kappa_1\mathbf{M}\boldsymbol{\phi} \tag{4.53}$$

In a general sense, this puts a limitation on the presented method for shifting since we can not use independent factorization of the $\mathbf{K}_s$ and $\mathbf{K}_f$ blocks. However, the inverse iterations can still be performed with a symmetric stiffness matrix. Namely, we can write the inverse iteration relation as

$$\begin{bmatrix} (\mathbf{K}_s - \sigma\mathbf{M}_s) & \mathbf{K}_{sf} \\ \sigma\mathbf{K}_{sf}^{\mathrm{T}} & (\mathbf{K}_f - \sigma\mathbf{M}_f) \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{q}}_{k+1,s} \\ \hat{\boldsymbol{q}}_{k+1,f} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ -\mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_{k,s} \\ \boldsymbol{q}_{k,f} \end{bmatrix} \tag{4.54}$$

It is possible to scale the second row in (4.54) by σ in order to end up with a symmetric stiffness matrix, namely

$$\begin{bmatrix} (\mathbf{K}_s - \sigma\mathbf{M}_s) & \mathbf{K}_{sf} \\ \mathbf{K}_{sf}^{\mathrm{T}} & (\frac{\mathbf{K}_f}{\sigma} - \mathbf{M}_f) \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{q}}_{k+1,s} \\ \hat{\boldsymbol{q}}_{k+1,f} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \frac{-\mathbf{K}_{sf}^{\mathrm{T}}}{\sigma} & \frac{\mathbf{M}_f}{\sigma} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_{k,s} \\ \boldsymbol{q}_{k,f} \end{bmatrix} \tag{4.55}$$

With (4.55), we can perform the factorization of a symmetric stiffness matrix and use it for inverse iteration step. Besides, $\mathbf{M}_{sym}$-orthogonalization operations can still be performed with the block diagonal symmetric mass matrix, $\kappa_1 \mathbf{M}$.

In order to be sure that no eigenvalues are missed during the computations, Sturm sequence checks should generally be employed [17]. This brings some extra factorization costs on the problem and should be used with care. In the case of vibro-acoustic eigenvalue problem, we can write the stiffness matrix of the shifted problem in a symmetric format as shown in (4.55). Therefore, Sturm sequence checks can be performed with a symmetric matrix which can save some factorization cost.

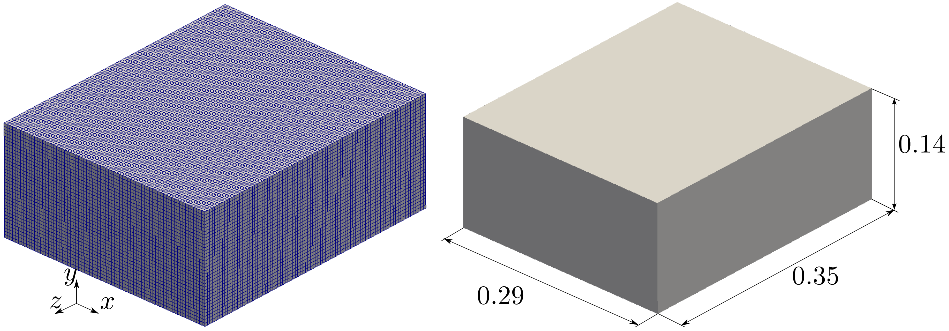
4.5. Numerical case studies

In order to analyze the numerical performance of the presented method, we performed analyses on two vibro-acoustic models. The aim of the performed analyses was to compare the accuracy of the implicitly symmetric solver, namely *vibro*-Lanczos, to the well-known two-sided Lanczos and Arnoldi methods which were briefly outlined in Sections 4.2.1 and 4.2.2.

Finite element models used in the numerical experiments were created in ANSYS by using the implemented fluid structure interface [6]. The structural part of the models were meshed using quadrilateral shell elements, namely, SHELL63 in ANSYS, which include both rotational and translational degrees of freedom. The fluid part of the models were meshed using hexahedral FLUID30 elements which use pressure variables as the main degrees of freedom. A special variant of FLUID30 elements were also used in order to represent the coupling between the structural and fluid domains. Having set up the models and the properties, the coupled system matrices were extracted by the use of the binary interface developed in C++ [145] by the author. All the computations were programmed and performed in MATLAB environment. An important ingredient of the Lanczos procedure is the linear solver which is used to factorize the operator matrices and build up the Krylov basis. In the numerical implementation, MA97 is used as a direct linear solver which is a part of Harwell Subroutine Library [77]. In order to speed up the computations, the MA97 interface was compiled with Intel compiler and the associated Intel Math Kernel Library for optimized BLAS implementations [82, 83].

We performed a thorough performance analysis of the presented symmetric algorithm. Namely, in the following two case study sections, the symmetric Lanczos procedure is compared with the nonsymmetric solvers outlined in Sections 4.2.1 and 4.2.2. In all of the numerical experiments, the minimum iteration number was set to $2 \times neig + 10$ where $neig$ represents the target eigenvalue count. A convergence tolerance of 10^{-10} was used on the relative residual error estimates provided in (4.13), (4.17) and (4.47), respectively. We analyzed the performance of the presented method in the following points:

- The CPU times for the solution process were compared for the presented three different methods
- In order to verify the correctness of the eigensolutions computed, we evaluate an error measure comparing the eigenvalues of the symmetric solver and those



(a) Mesh of the 3D box **(b)** Dimensions of the 3D box in [m]

Figure 4.1: 3D box model

found by the Arnoldi solver at convergence. We used the following error criteria:

$$\Delta_\omega^k = \frac{\left|\omega_{\text{sym}}^k - \omega_{\text{Arnoldi}}^k\right|}{\omega_{\text{Arnoldi}}^k}, \quad k = 1, \ldots, neig \tag{4.56}$$

- Accuracy was also verified for the eigenvectors, considering Modal Assurance Criteria (MAC [5]) values between corresponding mode shapes computed with the symmetric Lanczos and with the Arnoldi algorithms. To represent an error-like term (1-MAC) is used as a performance criteria. MAC between two vectors, $\boldsymbol{\phi}_{\text{sym}}^k$ and $\boldsymbol{\phi}_{\text{Arnoldi}}^k$, is computed as presented in (4.58),

$$\begin{align} \Delta_{\text{MAC}}^k &= 1 - \text{MAC}(\boldsymbol{\phi}_{\text{sym}}^k, \boldsymbol{\phi}_{\text{Arnoldi}}^k) \tag{4.57} \\ &= 1 - \frac{({\boldsymbol{\phi}_{\text{sym}}^k}^{\mathrm{T}} \boldsymbol{\phi}_{\text{Arnoldi}}^k)^2}{({\boldsymbol{\phi}_{\text{sym}}^k}^{\mathrm{T}} \boldsymbol{\phi}_{\text{sym}}^k)({\boldsymbol{\phi}_{\text{Arnoldi}}^k}^{\mathrm{T}} \boldsymbol{\phi}_{\text{Arnoldi}}^k)}, \quad k = 1, \ldots, neig \tag{4.58} \end{align}$$

In the numerical tests and the comparisons reported to evaluate the performance of *vibro*-Lanczos, we used the partial orthogonalization scheme presented in Sections 4.4.5 and 4.4.6.

4.5.1. Test case 1

The first model considered is a completely sealed 3-dimensional cubic cavity filled with water which has a density of 1000 kg/m^3 and the speed of sound in water is taken as 1500 m/s. The top surface of this cavity is modelled as a simply supported thin shell structure of 2 mm thickness, which is coupled to the fluid domain. The rest of the walls of the fluid domain are considered rigid. The specific properties of the elastic shell surface are as follows: density, 2800 kg/m^3 , Young modulus, 70 GPa and poisson ratio, 0.3, respectively. A schematic representation of the model with the finite element mesh is shown in Figure 4.1 along with the dimensions. The model consists of 117740 elements resulting in 145331 degrees of freedom (dofs) in

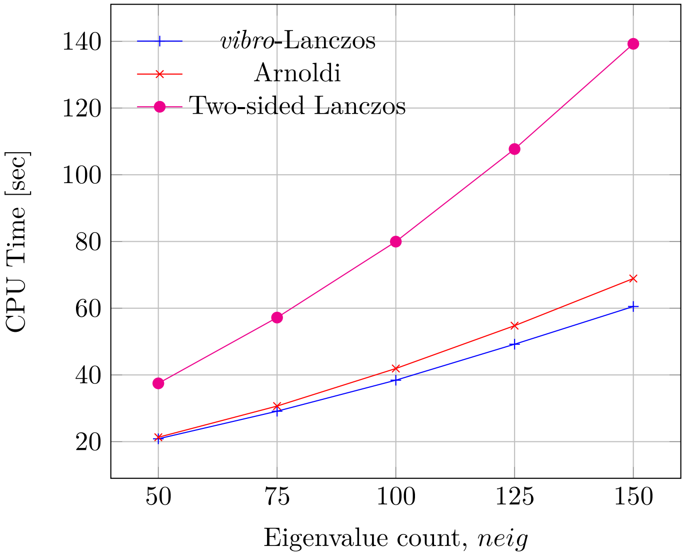


Figure 4.2: CPU Time for the three variants, 3D Box model

Table 4.1: CPU time saving of *vibro*-Lanczos compared to the two-sided Lanczos and the Arnoldi method (3D Box)

	neig				
	50	75	100	125	150
Two-sided Lanczos	%44.34	%49.08	%51.95	%54.315	%56.54
Arnoldi	%2.16	%5.02	%8.34	%10.15	%12.16

total. The structural part of the model consists of 23850 dofs. The structural stiffness matrix has 1182884 nonzero (nnz) elements with a bandwidth of 412. The fluid part of the system stiffness matrix consists of 121481 dofs. And it has 2425657 nnz elements in a bandwidth of 3486.

By looking at the elapsed CPU times for the solution sequences (Figure 4.2), one can observe that the two-sided Lanczos solver is almost twice as expensive as the Arnoldi's solution method. And it is the most expensive method of the three variants. The proposed symmetric solution approach can bring some important savings when implemented with the partial reorthogonalization scheme. An overview on the savings with respect to Arnoldi and the two-sided Lanczos solvers is provided on Table 4.1. For large target eigenvalue counts like 100, 125 and 150, the saving of the proposed symmetric approach over Arnoldi is around %10.

Maximum and minimum of the relative frequency errors computed with (4.56) for the selected target eigenvalue count is provided in Figure 4.3. Besides, associated maximum errors (1-MAC) on the eigenvectors computed with (4.58) are also provided on the same Figure. As also observed from Figure 4.3, the computed maxi-

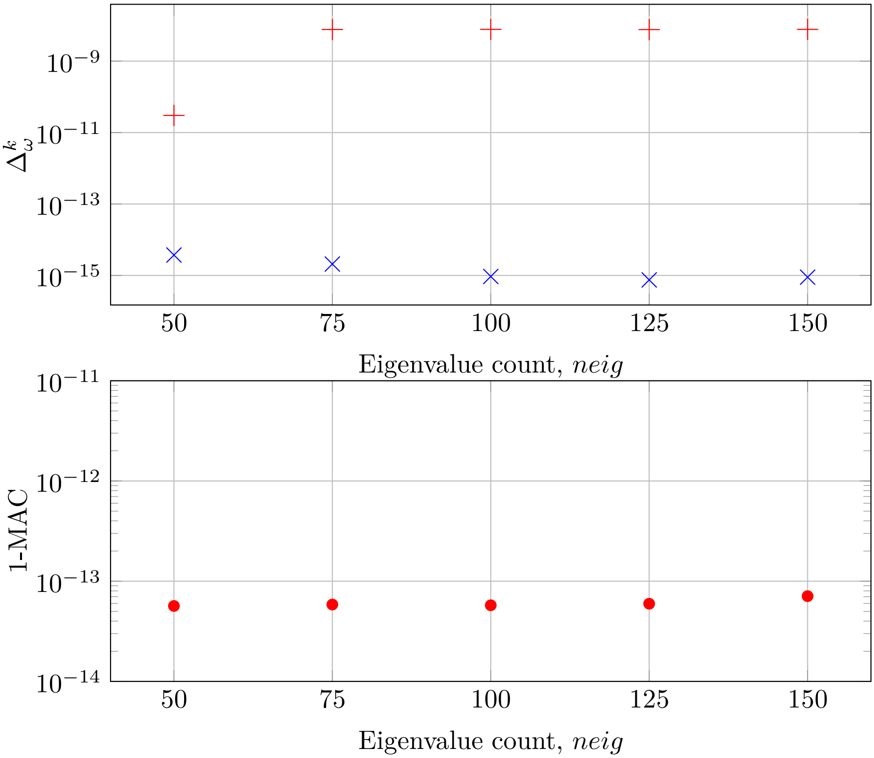
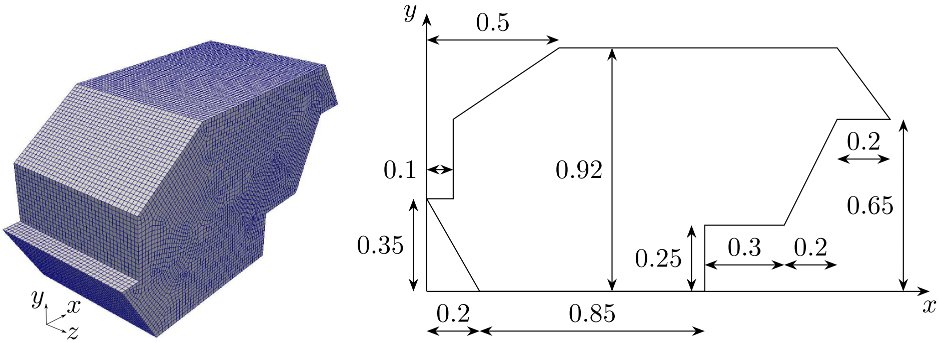


Figure 4.3: Max(+)/Min(x) error levels on frequencies and Max errors (1-MAC) on mode vectors (3D Box)

mum error levels are quite small signifying that they are in good agreement with the results computed with the nonsymmetric Arnoldi solver. It is important to note that the convergence of the presented methods is monitored with the relative residual estimates provided in (4.13), (4.17) and (4.47) where the stopping tolerances were set to 10^{-10}. However, the literature in the field suggests that although the residual estimates are useful cheap estimates for convergence checks, they do not always provide the true error levels on the eigenvalues as can be seen in Figure 4.3 [136].

4.5.2. Test case 2

The second model that is considered is a simplified model of a vehicle cavity where the coupling fluid inside the domain is air. On this model, two surfaces, namely, the windscreen and the roof of the cavity, are used as simply supported thin elastic structures which are coupled with the fluid domain inside the cavity. The mentioned elastic surfaces are modelled with thin shell elements of 1mm thickness. Similar to the first test case, the rest of the walls of the fluid domain are considered rigid. The properties of air is chosen as follows: density, 1.225 kg/m^3, speed of sound 343.2 m/s. The properties of the elastic structural surfaces are as follows: density, 7860



(a) Mesh of the cavity **(b)** Dimensions of the cavity section in [m]

Figure 4.4: Simplified vehicle cavity

kg/m^3, Youngs modulus, 210 GPa and poisson ratio, 0.3. The dimensions of the section of the cavity are provided in Figure 4.4b in m and the depth of the cavity in z direction is 0.75m. A snapshot of the generated mesh is provided in Figure 4.4a along with the coordinate axis. The model consists of 283559 elements resulting in a system with 316808 degrees of freedom.

Structural part of this model consists of 30290 dofs. And the associated stiffness matrix has 1066492 nnz elements in a bandwidth of 639. The fluid part of the stiffness matrix consists of 286518 dofs. And it has 7489298 nnz elements in a bandwidth of 7040.

Examining the CPU time comparisons provided in Figure 4.5, we can conclude that the elapsed CPU times show a similar trend to the analysis performed for the previous test case. The detailed information on percentage savings with respect to Arnoldi and two-sided Lanczos solvers is provided in Table 4.2. Similar savings to the previous test case are also observed for this model. $\mathbf{M}_{sym}$-orthogonality among the Lanczos vectors is continuously monitored with (4.48) for both of the numerical examples. As outlined in Section 4.4.5, whenever $\|\mathbf{Q}_k^{\mathrm{T}}\mathbf{M}_{sym}\boldsymbol{q}_{k+1}\|_\infty > 10^{-9}$, a full reorthogonalization is performed at the current and the following iteration step. In order to show the gain with the partial reorthogonalization scheme on this test case, we compared the orthogonalization costs of the symmetric Lanczos with full orthogonalization and partial reorthogonalization approaches. Theoretical reasoning on the applicability of partial reorthogonalization was summarized in Section 4.4.5. For these comparisons, we selected target eigenvalue counts 50 and 150.

We sketched a table where we show the detailed information on the performance comparison of the outlined schemes. Namely, Table 4.3 provides an overview on the performance of full and partial reorthogonalization schemes. IC in column 2 shows the iteration counts to convergence for the selected target eigenvalue counts shown in column 1. FS in column 3 shows the total number of full reorthogonalization steps which should have been performed if it was implemented as a normal symmetric

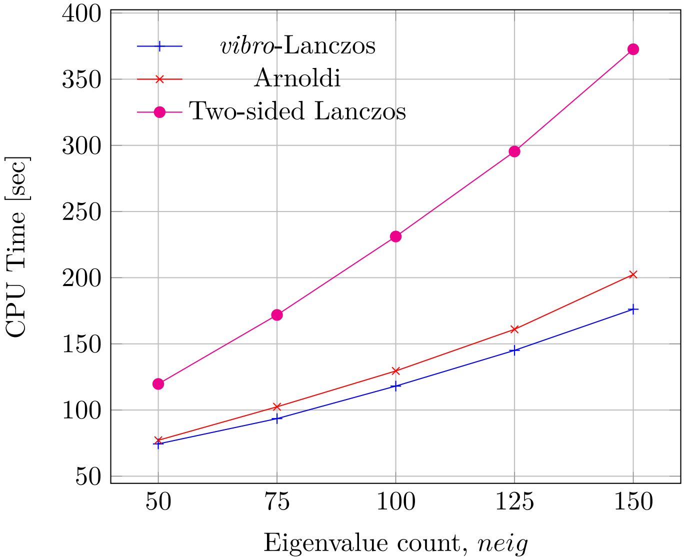


Figure 4.5: CPU Time for the three variants, simplified vehicle cavity model

Table 4.2: CPU time saving of *vibro*-Lanczos compared to the two-sided Lanczos and the Arnoldi method (simplified vehicle cavity)

	neig				
	50	75	100	125	150
Two-sided Lanczos	%37.83	%45.58	%48.9	%50.88	%52.77
Arnoldi	%3.56	%6.02	%8.82	%10.01	%12.97

Table 4.3: Partial reorthogonalization performance (simplified vehicle cavity)

neig	IC	FS	PS	CF[FLOPS]	CP[FLOPS]	% Saving
50	111	111	48	7.8771×10^{9}	3.7713×10^{9}	52.12
150	311	311	176	6.14814×10^{10}	3.8402×10^{10}	37.54

Lanczos process with full reorthogonalization [7]. Whereas, PS in column 4 represents the total number of partial reorthogonalization steps. CF in column 5 represents the cost of floating point operations (FLOPS) for full reorthogonalization. Similarly, CP in column 6 represents the FLOPS for partial reorthogonalization.

We point to Algorithm 4.1 in order to provide a brief outline on the computation of FLOPS. Namely, the orthogonalization operations are performed on lines 15 and 16. Line 15 is a dot product where m multiplications and $(m-1)$ additions are performed. Line 16 is composed of a vector scaling followed by a vector subtraction. Operational cost related Line 16 is m multiplications plus m subtractions. And these numbers are related to $a\boldsymbol{q}_j$ and $(\hat{\boldsymbol{q}}_{k+1} - a\boldsymbol{q}_j)$, respectively. m is the size of the vectors used in the orthogonalization operations which is 316808 for this case. In total, $(4m-1)$ FLOPS is performed in one step of the reorthogonalization iterations. If the total iteration count is given by k then the total cost of reorthogonalization is given by the sum of the orthogonalization costs per iteration which is $\sum_{n=1}^{k}(4m-1)n$ FLOPS.

It is good to note that Arnoldi performs an explicit reorthogonalization at each iteration where the costs are similar to column 5 in Table 4.3. However, with partial reorthogonalization, we can see that considerable gains can be achieved for the symmetric case by comparing columns 5 and 6 of the same table. Our numerical experience also suggests that the frequency of partial reorthogonalization increases for ill-conditioned problems significantly. Note however that, the reported frequencies of partial reorthogonalization in [142] and [156] are lower than the ones we have found in this study.

For this test case, we also compared the convergence of the first and last eigenvalues. Namely, we tried to represent the convergence by showing the relative errors on the frequency values with respect to the values found by Arnoldi iterations. We performed these checks at different iteration steps, k, by using the following global error criterion

$$\frac{|\omega_i(\mathbf{T}_k) - \omega_i^{\text{Arnoldi}}|}{\omega_i^{\text{Arnoldi}}}, \quad \text{for} \quad i = 1, \text{neig} \tag{4.59}$$

where $\omega_i^{\text{Arnoldi}}$ are the frequencies computed by Arnoldi and $\omega_i(\mathbf{T}_k)$ are the frequencies computed with the symmetric tridiagonal of the proposed approach.

With (4.59), we plotted the resulting convergence behavior in Figures 4.6 and 4.7 for the first and last eigenvalues while solving for 50 and 150 eigenvalues, respectively. We also verified that the convergence behavior shows a similar trend for

---

[7] Note that FS and IC are equal in the full reorthogonalization scheme.

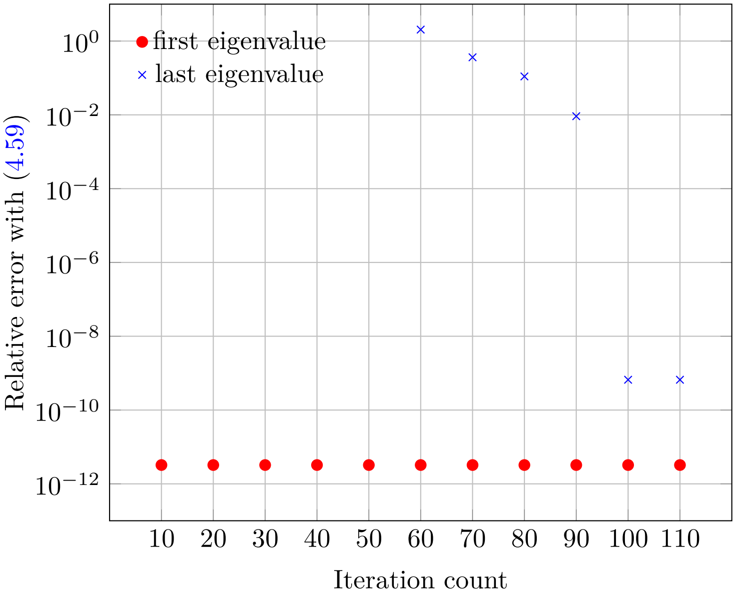


Figure 4.6: Convergence history of the symmetric solver for 50 eigenvalues (simplified vehicle cavity)

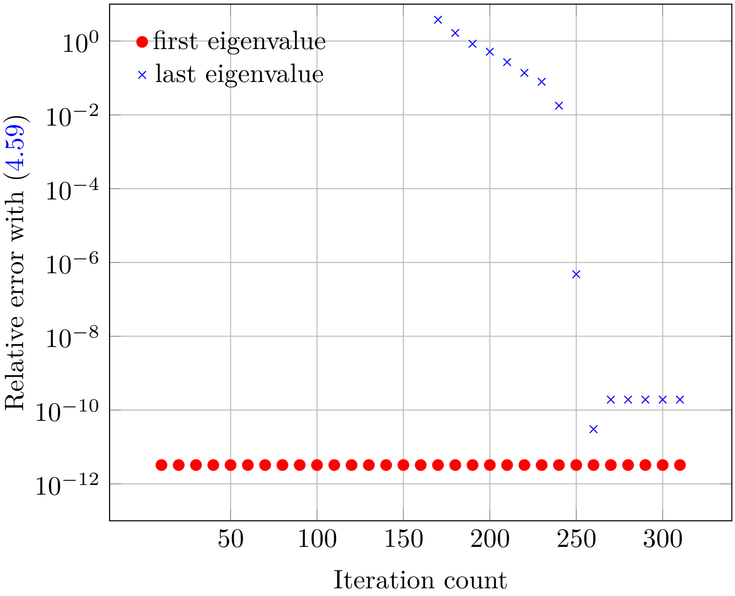


Figure 4.7: Convergence history of the symmetric solver for 150 eigenvalues (simplified vehicle cavity)

the first test case. As we can observe from the convergence plots, the first eigenvalue converges in the earlier steps of iterations which is inline with the theoretical expectation [136]. In contrast, the convergence of the last eigenvalue speeds up as the number of iterations increase.

4.6. Summary and conclusions

In this chapter, we proposed a symmetric Lanczos method that efficiently computes the solution for the vibro-acoustic eigenvalue problems. These problems can be challenging due to the nonsymmetric system matrices and their ill-conditioned nature. For the original problem representation, it is necessary to resort to nonsymmetric eigenvalue solution techniques. To our knowledge, an efficient symmetric Lanczos implementation has not been proposed before for these systems using the displacement-pressure formulation without adding extra variables like velocity potential.

Although the closed form representation of the system matrices can become fully populated depending on the transformation matrix used, it is shown that it is not necessary to use the matrices in those formats. Namely, the numerical operations can be arranged such that an efficient symmetric implementation is possible. Further, we have shown that the proposed symmetric Lanczos solver and the Arnoldi algorithm generate the same subspace to find the eigensolution of the considered problem. In addition to its symmetric nature, several other advantages of the method can be summarized as follows:

- The orthogonality between the basis vectors is ensured with respect to a symmetric mass matrix (thereby providing an inherent scaling), whereas Arnoldi iterations produce the orthogonality with respect to an L^2-norm. In such a L^2-norm pressure and displacement variables are mixed which could lead to ill-conditioning during the orthogonalization process.

- For the symmetric Lanczos method, theoretical three-term recurrence relations exist. Therefore, computational savings can be achieved by using either partial or selective orthogonalization schemes.

- The method can bring important computational savings with respect to the two-sided Lanczos solver. Moreover, for large target eigenvalue counts, the savings of the method with respect to the Arnoldi solver is on the order 10% in our examples.

Hence we can conclude that the proposed symmetric Lanczos algorithm for vibro-acoustic problems can advantageously replace the commonly used non-symmetric Lanczos or Arnoldi algorithms.

II

Investigations on Substructuring Techniques for Vibro-Acoustic Simulations

5

Locally Symmetric Craig-Bampton Approach for Vibro-Acoustic Problems, *vibro*-LsCB

In this chapter, we present and propose a new Craig-Bampton type component mode synthesis technique for coupled structural-acoustic problems. In this technique, original nonsymmetric component matrices originating from structural-acoustic coupling are converted to symmetric forms in numerical operations. To build up the Craig-Bampton projection space efficiently, operations are arranged and performed by using symmetric components while these symmetric forms are never built explicitly. The advantage of this new technique is that we can use symmetric solvers to calculate the ingredients of the Craig-Bampton basis on the component level. By using the current technique, we show that locally-symmetric Craig-Bampton matrices are built. Moreover, the resulting reduced symmetric component matrices have the nice sparsity patterns resulting from a typical Craig-Bampton reduction. In the next step, we demonstrate how we can assemble the reduced components with either primal or dual coupling approaches where we also show their relations. Last, we performed some numerical tests on two-dimensional test problems to show its accuracy and reduction performance. An earlier version of the research presented in this chapter was presented in [149]. A journal manuscript based on the this chapter is under preparation.

5.1. Introduction

In computational modelling of real-world engineering problems, investigation of model reduction techniques is still an active research area. The first techniques were proposed in 60s and 70s, mainly, in the area of structural engineering. A recent comparison and overview of different techniques in different fields was reviewed in [23]. Pioneering work in this field is known as the Craig-Bampton technique [37] which is an extension of the method proposed by Hurty earlier [79], [80] with fixed interface dynamic modes. Rubin [132] and MacNeal [101] later proposed free-interface versions by also including attachment modes in the reduction space. Finally, the reduced components are assembled together for the prediction of global system dynamics. An extensive overview of methods can be found in the relatively recent source [35].

Extension of these methods to coupled vibro-acoustic problems were investigated mainly for piping systems in [102], [75], [74]. Commonly, in these investigations, the researchers used the Craig-Bampton reduction technique as the main method. However, they omitted some important coupling terms in the representation of the constraint modes whereas they still used the coupled fixed interface modes for the dynamic characterization of the reduced problems. With this practice, it is not possible to guarantee a statically complete representation. In spite of this deficiency, they reported good accuracy on the results which is also, to some extent, validated with measurements [75].

In this chapter, we are proposing an improved and efficient method extending the above mentioned references. To start with, we also base our development on the well-known Craig-Bampton technique but we also make use of the numerical experience gained in Chapter 4. Namely, as a bi-product of our previous research, we propose to extend the idea to create symmetric component matrices in an efficient manner. In contrast to the above mentioned references, we do not omit any terms while building up the projection space. And, therefore, we guarantee a statically complete representation. We show that we can build a sequence of efficient computational steps to compute the ingredients of the Craig-Bampton projection space with symmetric matrices. Due to the symmetry property, the reduced matrices of the components also exhibit the nice sparsity patterns resulting from a conventional Craig-Bampton reduction [63]. Although, on the component level, the resulting reduced matrices turn out to be symmetric, we show that the resulting global coupled system matrices are nonsymmetric. Namely, the transformation of the component matrices to symmetric format requires an update of the compatibility conditions. This practice results in using different left and right null spaces to couple the reduced symmetric component matrices and therefore results in nonsymmetric reduced global system matrices.

The outline of this chapter is as follows: in Section 5.2, we provide a general overview on reduction and component mode synthesis. Section 5.3 provides the details of the Craig-Bampton reduction technique and its extension to vibro-acoustic problems with a symmetric matrix perspective in the mind. We show the details of component coupling in general and specifically for the current technique in Section 5.4. Section 5.5 provides some details on the efficient numerical implementation of the technique. We perform some numerical experiments in Section 5.6 to show the results for the evaluation of the technique. Main conclusions are summarized in Section 5.7.

5.2. General overview on Reduction and Component Mode Synthesis

5.2.1. Dynamic model reduction by modal trunctation

In the analysis of complex systems, it is quite common to reduce the number of degrees of freedom that are effectively used for the analysis. To accomplish this reduction, one needs a reduction basis which has a lower dimension than the original degrees of freedom of the model used. One of the easiest and commonly accepted practices is to select some of the low frequency modes of the system in order to build up this set, namely,

$$\boldsymbol{x} = \mathbf{R}\boldsymbol{\eta} \tag{5.1}$$

where $\boldsymbol{\eta}$ represents the transformed and significantly reduced set of coordinates in the new space, which are also called the modal coordinates. The original set of degrees of freedom, namely, $\boldsymbol{x}$, is approximated with a lower dimensional subspace of vectors found in the matrix, $\mathbf{R}$. $\mathbf{R}$ is written explicitly as follows,

$$\mathbf{R} = \begin{bmatrix} \boldsymbol{\phi}_1 & \ldots & \boldsymbol{\phi}_m \end{bmatrix}, \quad m \ll n \tag{5.2}$$

In this representation, the vectors in $\mathbf{R}$, namely, $\boldsymbol{\phi}_m$, are the first m eigenvectors of the eigenvalue problem

$$\left(\mathbf{K} - \lambda_m \mathbf{M}\right) \boldsymbol{\phi}_m = \mathbf{0} \tag{5.3}$$

Besides, n represents the original problem size. The aim of this analysis is to be able to represent the dynamic system response with a significant reduction in size. In addition to the reduction in size, we have to also achieve a sufficient accuracy level. However, we should mention that, with this selection, we have to use the full (unreduced) system matrices, $\mathbf{K}$ and $\mathbf{M}$, to solve for its eigenvectors first, which is a serious drawback. However, practically, we would like to perform the reduction without computing its eigenvectors(eigenmodes).

Considering the equation of motion of an undamped system representation, namely,

$$\mathbf{M}\ddot{\boldsymbol{x}} + \mathbf{K}\boldsymbol{x} = \boldsymbol{F} \tag{5.4}$$

we can now replace, $\boldsymbol{x}$, by the expression given in (5.1) to write

$$\mathbf{M}\mathbf{R}\ddot{\boldsymbol{\eta}} + \mathbf{K}\mathbf{R}\boldsymbol{x} = \boldsymbol{F} + \boldsymbol{r} \tag{5.5}$$

since the original vector of degrees of freedom, $\boldsymbol{x}$, is approximated by using the subspace spanned by the vectors in $\mathbf{R}$, a force residual has to be introduced in (5.5). Projecting the system in (5.5) onto $\mathbf{R}$[1] results in

$$\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\ddot{\boldsymbol{\eta}} + \mathbf{R}^{\mathrm{T}}\mathbf{K}\mathbf{R}\boldsymbol{\eta} = \mathbf{R}^{\mathrm{T}}\boldsymbol{F} + \mathbf{R}^{\mathrm{T}}\boldsymbol{r} \tag{5.6}$$

At this point, we enforce the condition that the residual vanishes in the selected space, $\mathbf{R}$, namely, $\mathbf{R}^{\mathrm{T}}\boldsymbol{r} = \mathbf{0}$. It is important to note that applying a projection like this always induces a force (equilibrium) error which is not directly visible in the reduced representation. But we can compute this error *a-posteriori* by using the approximate solution back in the full system representation.

---

[1] Multiplying the equation from left with $\mathbf{R}^{\mathrm{T}}$

With this assumption, we end up with the modally reduced problem of dimension m:

$$\mathbf{M}_m \ddot{\boldsymbol{\eta}} + \mathbf{K}_m \boldsymbol{\eta} = \boldsymbol{f}_m, \quad \text{where} \quad \begin{aligned} \mathbf{M}_m &= \mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R} \\ \mathbf{K}_m &= \mathbf{R}^{\mathrm{T}}\mathbf{K}\mathbf{R} \\ \boldsymbol{f}_m &= \mathbf{R}^{\mathrm{T}}\boldsymbol{F} \end{aligned} \tag{5.7}$$

After solving the reduced problem in (5.7), we can expand back to the physical domain by using (5.1).

5.2.2. Component Mode Synthesis, a non-mathematical overview

Computing the modes of the full system is not very practical if one tries to reduce the cost of finding a solution for the system. Another approach consists in dividing the full problem into substructures/components and, subsequently, performing the reduction by using a set of low frequency eigenmodes. This technique is an efficient and well-known procedure to reduce the size of the system equations. Different efficient reduction techniques were proposed by different researchers. These methods mainly differ in the selection of the ingredients of the reduction basis, **R**. Namely, **R** can be composed of some fixed interface modes along with some static constraint modes leading to the well-known Craig-Bamptom method [37]. In some other class of methods, the reduction basis, **R**, is built with some free-interface and attachment modes which are known as free-interface methods [132],[101]

During the analysis of very large complex systems, we can divide the domain of the considered complete model, Ω_c, into some non-overlapping component domains, Ω_k, as shown in Figure 5.1.

With this representation, we are setting up some partial domains which we can analyze independently. Each domain has its internal nodes leading to internal degrees of freedom. Similarly, each domain has some interface nodes leading to some inteface degrees of freedom. From this point on, variables related to the internals of a component are going to be denoted with the subscript $_i$. And the variables related to the interfaces of a component are going to be denoted with the subscript $_b$. Moreover, we are going to use the $_i$ and $_b$ notations heavily on the component level representations and during the theory discussed further on.

Component Mode Synthesis procedure involves 3 basic steps:

1. Division of the selected domain into components,
2. Definition of the ingredients of the reduction space, **R**, and reduction on the component level,
3. Coupling (assembly) of the reduced component matrices to assemble the global reduced system matrices. This step is achieved by enforcing the *compatibility* and *equilibrium* conditions between the interfaces of the different components used.

These steps are also summarized schematically in Figure 5.1.

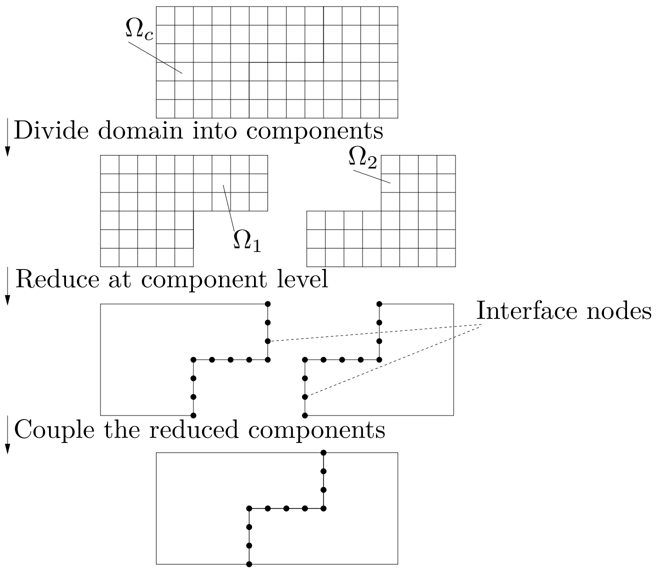

Ω_c
Divide domain into components
Ω_2
Ω_1
Reduce at component level
Interface nodes
Couple the reduced components

Figure 5.1: Component Mode Synthesis steps on a simple domain, Ω_c

5.3. Craig-Bampton reduction for vibro-acoustics

In this section, we will outline the details on how we can come up with reduced *vibro-acoustic* component matrices by using the Craig-Bampton reduction technique [37] which was originally developed for problems with symmetric component matrices. In Section 5.3.1, we will briefly outline the Craig Bampton technique used for component level reduction. As a follow up, in Section 5.3.2, we will outline some options to compute symmetric *vibro-acoustic* component matrices. Computational details on how to compute the ingredients of a CB basis will be outlined in Section 5.3.3.

5.3.1. Brief overview of Craig-Bampton approach

In this short subsection, we are going to provide a brief overview of the Craig-Bampton (CB) approach for structural systems with displacement degrees of freedom only. The general transformation equation from physical coordinate space to generalized component modal space reads as

$$\boldsymbol{u}^{(s)} = \mathbf{R}^{(s)}\boldsymbol{\eta}^{(s)} \tag{5.8}$$

where $\mathbf{R}^{(s)}$ represents the reduction basis, transformation matrix or a Ritz basis. $\boldsymbol{u}^{(s)}$ represents the displacement degrees of freedom of the component and $\boldsymbol{\eta}^{(s)}$ represents the generalized degrees of freedom. In (5.8), (s) superscript is used to represent the component or substructure under consideration.

CB approach is an improved version of Guyan's static condensation idea [72] with fixed interface modes. The reduction basis is constructed by using the static constraint modes and the fixed interface component modes. The first set is necessary to facilitate the component coupling on the interfaces. It also guarantees that a statically complete representation is set up in the absence of dynamic contributions. The second set of vectors, namely fixed interface modes, represent the internal dynamics of the component when the interfaces of the component are fixed.

For the component under consideration, we are going to use the following partitioned form of the general equation of motion,

$$\begin{bmatrix} \mathbf{M}_{bb} & \mathbf{M}_{bi} \\ \mathbf{M}_{ib} & \mathbf{M}_{ii} \end{bmatrix}^{(s)} \begin{bmatrix} \ddot{\boldsymbol{u}}_b \\ \ddot{\boldsymbol{u}}_i \end{bmatrix}^{(s)} + \begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bi} \\ \mathbf{K}_{ib} & \mathbf{K}_{ii} \end{bmatrix}^{(s)} \begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_i \end{bmatrix}^{(s)} = \begin{bmatrix} \boldsymbol{f}_b \\ \boldsymbol{f}_i \end{bmatrix}^{(s)} + \begin{bmatrix} \boldsymbol{g}_b \\ \mathbf{0} \end{bmatrix}^{(s)} \tag{5.9}$$

where subscript $_b$ represents the boundary degrees of freedom and subscript $_i$ represents the internal degrees of freedom of the component. Moreover, in (5.9), $\boldsymbol{f}$ represents the external forces and $\boldsymbol{g}$ represents the interface force due to the neighbouring components. From this point on, we will drop the (s) superscript from the component equations as presented in (5.9). Note that the system partitioning is done on the basis that the application points of the external forces are used as the parts of the partition boundary. This is not a must but, in practice, leaves the internal degrees of freedom excitation free.

Constraint modes are calculated by statically imposing a unit displacement on the interface degrees of freedom one at a time while keeping the displacements of the other interace degrees of freedom zero. It is also assumed that there are no reaction forces internally, namely,

$$\begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bi} \\ \mathbf{K}_{ib} & \mathbf{K}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{I}_b \\ \boldsymbol{\Psi}_c \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b \\ \mathbf{0} \end{bmatrix} \tag{5.10}$$

where $\mathbf{R}_b$ represents the unknown interface reactions. By using the second block row, we can write the constraint mode matrix as follows,

$$\boldsymbol{\Psi}_c = -\mathbf{K}_{ii}^{-1}\mathbf{K}_{ib} \tag{5.11}$$

which results in the final expression for the constraint modes, namely,

$$\begin{bmatrix} \mathbf{I}_b \\ \boldsymbol{\Psi}_c \end{bmatrix} = \begin{bmatrix} \mathbf{I}_b \\ -\mathbf{K}_{ii}^{-1}\mathbf{K}_{ib} \end{bmatrix} \tag{5.12}$$

Component fixed interface normal modes of the structure are obtained by restraining the boundary degrees of freedom. Namely, solving the undamped eigenvalue problem for the internal partition of the component matrices,

$$(\mathbf{K}_{ii} - \lambda_j \mathbf{M}_{ii}) \{\boldsymbol{\phi}_{ii}\}_j = \mathbf{0}, \qquad j = 1, 2, ..., m \tag{5.13}$$

where $\lambda_j, \{\boldsymbol{\phi}_{ii}\}_j$ are the j^{th} eigenvalue and the corresponding mode shape, respectively. And m is the number of the normal modes selected for computations. Generally, fixed interface modes are selected according to an target frequency, f_T. In this respect, fixed interface modes with frequencies of up to two times the selected frequency, f_T, are kept in the CB basis. In a matrix representation, the fixed interface modes read as,

$$[\boldsymbol{\Phi}_{ii}] = \left[\{\boldsymbol{\phi}_{ii}\}_1 \quad \{\boldsymbol{\phi}_{ii}\}_2 \quad \dots \quad \{\boldsymbol{\phi}_{ii}\}_m\right] \tag{5.14}$$

The overall reduction basis, $\mathbf{R}_{cb}$, for the component is written as,

$$\begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_i \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I}_b & \mathbf{0} \\ \mathbf{\Psi}_c & \mathbf{\Phi}_{ii} \end{bmatrix}}_{\mathbf{R}_{cb}} \begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{\eta}_i \end{bmatrix} \tag{5.15}$$

In the Craig-Bampton approach, as in (5.15), boundary degrees of freedom $\boldsymbol{u}_b$ is kept and used for component coupling. The internal degrees of freedom of the component is now represented as a linear combination of fixed interface modes and static constraint modes.

5.3.2. Options for symmetric vibro-acoustic component matrices and its advantage

By following a similar notation to the one used in (5.9), we can write the equation of motion for a vibro-acoustic component as follows,

$$\mathbf{M}_c\ddot{\boldsymbol{q}} + \mathbf{K}_c\boldsymbol{q} = \boldsymbol{f} + \boldsymbol{g} \tag{5.16}$$

where

$$\mathbf{M}_c = \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f \end{bmatrix} \tag{5.17a}$$

$$\mathbf{K}_c = \begin{bmatrix} \mathbf{K}_s & -\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix} \tag{5.17b}$$

are the vibro-acoustic component matrices and

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{bmatrix} \tag{5.17c}$$

represents the degrees of freedom of the component.

In (5.16), $\boldsymbol{f}$ represents the externally applied forces/sources on the component and $\boldsymbol{g}$ represents the forces/fluxes[2] that originate from the neighbouring components.

For structural-acoustic problems, to end up with symmetric components matrices in (5.16), we can use two approaches to scale the component equations represented with the original nonsymmetric component matrices. These will shortly be outlined next. It is important to note that we used these matrices in Section 4.3 while explaining the details of the symmetric eigenvalue solver, namely, presented in (4.21) and (4.22). At this point, we extend on the same approach but use the transformation matrices that result in a block diagonal stiffness matrix.

---

[2] Since we are now working with components involving acoustics, acoustic sources and fluxes are involved.

1. Multiplying (5.16) from left with the scaling matrix, κ, namely,

$$\kappa = \begin{bmatrix} \mathbf{I} & \mathbf{K}_{sf}\mathbf{K}_f^{-1} \\ \mathbf{0} & \mathbf{M}_f\mathbf{K}_f^{-1} \end{bmatrix} \tag{5.18}$$

   In (5.18), we assumed that $\mathbf{K}_f$ is a regular invertible matrix.

2. In (5.16), applying a coordinate transformation for the original degrees of freedom, $\boldsymbol{q}$, namely by using,

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_s^{-1}\mathbf{M}_s & \mathbf{K}_s^{-1}\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \boldsymbol{p} \end{bmatrix} \equiv \tau\tilde{\boldsymbol{q}} \tag{5.19}$$

   where the transformation matrix τ is used for the transformation[3]. Similar to κ-matrix above, in (5.19), we assume that $\mathbf{K}_s$ is invertible so that the numerical operations can be performed without problems[4]. In the new coordinate space, $\tilde{\boldsymbol{q}}$, $\tilde{\boldsymbol{u}}$ has the meaning of acceleration. This is the acceleration for which the related inertia would create as the static response together with the pressure on the interface.

The tranformed symmetric equations of motions of the component can be written for κ and τ cases, respectively, as follows

$$\kappa\mathbf{M}\ddot{\boldsymbol{q}} + \kappa\mathbf{K}\boldsymbol{q} = \kappa\boldsymbol{f} + \kappa\boldsymbol{g} \tag{5.20}$$

$$\mathbf{M}\tau\ddot{\tilde{\boldsymbol{q}}} + \mathbf{K}\tau\tilde{\boldsymbol{q}} = \boldsymbol{f} + \boldsymbol{g} \tag{5.21}$$

In the subsequent discussions, systems written with κ and τ matrices are going to be called κ-symmetric system and τ-symmetric system, respectively.

In the light of the ongoing discussion, we can write the symmetric forms of the component matrices with κ and τ, respectively, as follows,

$$\underbrace{\begin{bmatrix} (\mathbf{M}_s + \mathbf{K}_{sf}\mathbf{K}_f^{-1}\mathbf{K}_{sf}^{\mathrm{T}}) & \mathbf{K}_{sf}\mathbf{K}_f^{-1}\mathbf{M}_f \\ \mathbf{M}_f\mathbf{K}_f^{-1}\mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f\mathbf{K}_f^{-1}\mathbf{M}_f \end{bmatrix}}_{\mathbf{M}_{sym}} \begin{bmatrix} \ddot{\boldsymbol{u}} \\ \ddot{\boldsymbol{p}} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{K}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_f \end{bmatrix}}_{\mathbf{K}_{sym}} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{bmatrix} = \kappa \begin{bmatrix} \boldsymbol{f}_s \\ \boldsymbol{q}_f \end{bmatrix} \tag{5.22}$$

$$\underbrace{\begin{bmatrix} \mathbf{M}_s\mathbf{K}_s^{-1}\mathbf{M}_s & \mathbf{M}_s\mathbf{K}_s^{-1}\mathbf{K}_{sf} \\ \mathbf{K}_{sf}^{\mathrm{T}}\mathbf{K}_s^{-1}\mathbf{M}_s & (\mathbf{M}_f + \mathbf{K}_{sf}^{\mathrm{T}}\mathbf{K}_s^{-1}\mathbf{K}_{sf}) \end{bmatrix}}_{\mathbf{M}_{sym}} \begin{bmatrix} \ddot{\tilde{\boldsymbol{u}}} \\ \ddot{\boldsymbol{p}} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}}_{\mathbf{K}_{sym}} \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_s \\ \boldsymbol{q}_f \end{bmatrix} \tag{5.23}$$

In (5.22) and (5.23), the stiffness matrices of the symmetric representations are in block diagonal format and still keep their sparse structure. In contrast, the mass

---

[3] It must be noted that the transformation in (5.19) is a change of basis.

[4] However, this is not strictly necessary and is going to be generalized later for singular matrices

matrices of both of the represetations are now fully populated and lost their sparse nature. At first sight, this can be seen as a numerical disadvantage. However, in Section 5.3.3, we are going to outline some practical ways to overcome this numerical difficulty when we need to solve for the fixed interface modes of the components.

Additionally, this practice has one more advantage from the reduction perspective. Namely, since we created symmetric component matrices, we can use the same projection matrix on the left and right while performing the reduction as presented in (5.7). Looking from a different perspective, by using either κ or τ matrices, we implicitly end up using different left and right projection spaces for the original non-symmetric component matrices. Namely, by using τ, we can conceptually represent the reduction on the component level as follows

$$\mathbf{R}^{\mathrm{T}}\mathbf{M}\tau\mathbf{R}\ddot{\tilde{\boldsymbol{\eta}}}+\mathbf{R}^{\mathrm{T}}\mathbf{K}\tau\mathbf{R}\tilde{\boldsymbol{\eta}}=\mathbf{R}^{\mathrm{T}}\boldsymbol{f}+\mathbf{R}^{\mathrm{T}}\boldsymbol{g} \tag{5.24}$$

In (5.24), $\tau\mathbf{R}$ can be interpreted as the right projection space for the original non-symmetric system. Similarly, $\mathbf{R}$ can be seen as the left projection space. Eventually, the component matrices are implicitly projected on different subspaces left and right. With this practice, we still guarantee real eigenvalues for the original non-symmetric component matrices because, eventually, the resulting reduced matrices will be symmetric. Some researchers used the same projection space on the left and right [75, 102]. However, they had to explicitly use a block diagonal decoupled projection basis to retain the relations between the off-diagonal coupling blocks of the component matrices[5]. Only in this way, they could still guarantee real eigenvalues on the reduced problem. In comparison to these references, here, we present a more natural approach by using a transformation matrix to retain the original properties of the non-symmetric component matrices.

5.3.3. Ingredients of the reduction basis for vibro-acoustic components, $\mathbf{R}_{cb}$

Fixed interface modes

In general, the fixed-interface modes of a component, which has symmetric matrices, can be computed with a symmetric Lanczos solver. The details can be found in [63, 151]. In order to solve for the internal modes or fixed interface modes of of any components, we have to solve the symmetric eigenvalue problem, namely,

$$\mathbf{K}_{sym,ii}\boldsymbol{\phi}_{sym,ii}=\lambda_{sym,ii}\mathbf{M}_{sym,ii}\boldsymbol{\phi}_{ii} \tag{5.25}$$

where $\mathbf{K}_{sym}$ and $\mathbf{M}_{sym}$ represent the symmetric matrices of the component under consideration. It is important to emphasize one point here. Namely, (5.25) is not the symmetric form of the internal vibro-acoustic problem $(\kappa_{ii}\mathbf{K}_{ii},\kappa_{ii}\mathbf{M}_{ii})$ or $(\mathbf{K}_{ii}\tau_{ii},\mathbf{M}_{ii}\tau_{ii})$ but the partition of the matrices that correspond to the internal degrees of freedom of the symmetric vibro-acoustic problem of the component. Mathematically speaking, these matrices can be written in pairs as $((\kappa\mathbf{K})_{ii},(\kappa\mathbf{M})_{ii})$ or $((\mathbf{K}\tau)_{ii},(\mathbf{M}\tau)_{ii})$.

---

[5] We used a similar approach with a block diagonal projection space in Chapter 3 to retain the properties of the original system matrices.

While solving the eigenvalue problem in (5.25), there are two important costs to pay. The details of these operations are outlined next, namely,

1. **Factorization of $\mathbf{K}_{sym,ii}$**: We can write the internal partition of the stiffness matrix, $\mathbf{K}_{sym,ii}$, provided for the pre- and post-multiplication matrices, κ and τ, respectively, as follows,

$$\mathbf{K}^{\kappa}_{sym,ii} = \begin{bmatrix} \mathbf{K}_{s,ii} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{f,ii} \end{bmatrix} \tag{5.26a}$$

$$\mathbf{K}^{\tau}_{sym,ii} = \begin{bmatrix} \mathbf{M}_{s,ii} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{f,ii} \end{bmatrix} \tag{5.26b}$$

   The advantage of using either of the above mentioned forms is their block-diagonal structure. And this practice decreases the numerical cost of the factorization.

2. **Inverse iteration step for the Lanczos iterations:** During the Lanczos iterations, after the factorization is performed, we have to perform the solution for the next iterate vector in the sequence, namely with,

$$\tilde{\boldsymbol{x}}_i^{k+1} = \mathbf{K}^{-1}_{sym,ii}\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k \tag{5.27}$$

   where $\boldsymbol{x}_i^k$ is the current iteration vector and $\tilde{\boldsymbol{x}}_i^{k+1}$ is the next iteration vector in the sequence. Moreover, for this step, we have to perform the matrix-vector multiplication $\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$ at step k of the iteration process. However, as outlined before in Section 5.3.2 and in [151], it is not possible to compute $\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$ explicitly because of the fully-populated nature of the mass matrix. In brief, we can perform the inverse iteration step of the Lanczos process provided that we can also perform the matrix vector multiplication $\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$ efficiently.

An efficient way to compute $\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$

As outlined above, we can not explicitly compute and extract the internal partition of the mass matrix for the calculations to be performed for the internal modes. The aim of this subsection is to show that we can arrange the operations to compute $\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$ efficiently.

In Section 5.3.2, we outlined two options for creating symmetric component matrices. For the following computations, we propose to use the post-multiplication option, namely, τ. The reason of this approach can be understood after writing the symmetric mass matrix explicitly with the help of τ matrix. Namely, the explicit symmetric mass matrix reads as,

$$\mathbf{M}_{sym} = \mathbf{M}\mathcal{T} = \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{K}_s^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\mathbf{A}_\tau} \underbrace{\begin{bmatrix} \mathbf{M}_s & \mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\mathbf{B}_\tau} \tag{5.28}$$

In this representation, τ was written in the form of multiplication of two matrices, namely, with $\mathbf{A}_\tau$ and $\mathbf{B}_\tau$ [52, 53, 51]. After presenting the multiplication explicitly, as in (5.28), we can come to the reason why we selected the τ option for the operations. Namely, considering a dummy vector $\hat{\boldsymbol{x}}$ for the multiplication with the expression in (5.28), we can write,

$$\mathbf{M}_{sym}\hat{\boldsymbol{x}} = \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{K}_s^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{M}_s & \mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\tau=\mathbf{A}_\tau \mathbf{B}_\tau} \begin{bmatrix} \hat{\boldsymbol{x}}_s \\ \hat{\boldsymbol{x}}_f \end{bmatrix} \tag{5.29}$$

While performing the multiplications from right to left in (5.29), we can observe that, during the first two matrix-vector multiplications, the fluid side of the input vector, $\hat{\boldsymbol{x}}_f$, is not affected by the multiplication with τ matrix and is transferred directly until the last matrix-vector multiplication with the coupled non-symmetric component mass matrix. Therefore, on the fluid side, the number of matrix-vector multiplications is minimized. This is mainly because of the identity matrices found on the fluid rows in (5.29).

Looking at the pre-multiplication option with κ, the same reasoning leads to the following relation,

$$\mathbf{M}_{sym}\hat{\boldsymbol{x}} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{M}_f \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f^{-1} \end{bmatrix}}_{\kappa=\mathbf{A}_\kappa \mathbf{B}_\kappa} \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}}_s \\ \hat{\boldsymbol{x}}_f \end{bmatrix} \tag{5.30}$$

In (5.30), we can not observe a similar partial direct transfer relation for the fluid rows as it was for the τ case. To summarize, we can mention the following points for the comparison between τ and κ usage, namely,

- As briefly mentioned above, by using κ, additional numerical operations must be performed on the fluid partition. This is a disadvantage for problems with large fluid domains which results in large matrices after the discretization process.

- Besides, for closed cavities, the stiffness matrix representing the fluid side of the problem becomes singular [114] and this problem appears on every component having a fluid domain without any boundary conditions except the interface boundary condition on the coupling surface. Therefore, we also have to tackle this singularity problem for every component properly. It might also be good to emphasize that when using τ for the solution of the fixed interface modes by using (5.26b), we do not have the singularity problem. Because, for the fluid side of the problem, the internal partition is used assuming that the boundary pressure degrees of freedom are fixed, $\boldsymbol{p}_b = 0$.

- A similar singularity problem also appears for components that are not constrained on the structural side when τ matrix is used. However depending on the partition and the speficic boundary conditions used on the partition, this might not always result in a singular stiffness matrix on the structural side.

> In contrast, when κ is used for creating symmetric component matrices, this singularity problem always exists for the fluid side. We are going to attack this specific singularity problem related to τ matrix in detail in Chapter 6.

Having provided the above extra points, we can make some useful computational arrangements in order to compute $\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$ for the symmetric component matrix. We are going to start the discussion from a simple matrix vector product to show the big picture. Namely, assume for a moment that we can compute the symmeric mass matrix, $\mathbf{M}_{sym}$, explicitly[6] and partition it according to boundary and internal degrees of freedom. Then, to find the matrix vector multiplication, $\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$, we can write

$$\begin{bmatrix}\boldsymbol{r}_b\\ \boldsymbol{r}_i\end{bmatrix} = \begin{bmatrix}\mathbf{M}_{sym,bb} & \mathbf{M}_{sym,bi}\\ \mathbf{M}_{sym,,ib} & \mathbf{M}_{sym,ii}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_b^k\\ \boldsymbol{x}_i^k\end{bmatrix} \tag{5.31}$$

and set the boundary partition of $\boldsymbol{x}^k$ to $\mathbf{0}$, namely, $\boldsymbol{x}_b^k = 0$, to get to

$$\begin{bmatrix}\boldsymbol{r}_b\\ \boldsymbol{r}_i\end{bmatrix} = \begin{bmatrix}\mathbf{M}_{sym,bi}\boldsymbol{x}_i^k\\ \mathbf{M}_{sym,ii}\boldsymbol{x}_i^k\end{bmatrix} \tag{5.32}$$

Eventually, $\boldsymbol{r}_i$ is the resulting vector, we are looking for.

Now, we have to adjust our computations for the actual case where we can not explicitly compute $\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$ as mentioned above in (5.31) and (5.32). Returning to (5.28), we have shown that the symmetric matrix can be written as a multiplication of three matrices. And we will use this fact here in order to find $\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$ at step k with some additional operations. Namely, the operations can be summarized as follows:

$$\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k = \left(\begin{bmatrix}\mathbf{M}_s & \mathbf{0}\\ \mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f\end{bmatrix}\begin{bmatrix}\mathbf{K}_s^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{M}_s & \mathbf{K}_{sf}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_s^k\\ \boldsymbol{x}_f^k\end{bmatrix}_{b=0}\right)_i \tag{5.33}$$

In (5.33), it is important to note that we did not perform any ordering on the matrices which was performed for the conceptual representations in (5.31) or in (5.32). Instead, we propose to set the boundary degrees of freedom for vector, $\boldsymbol{x}^k$, to zero and perform the multiplication in this way. The subscript $_{b=0}$ on the vector $\boldsymbol{x}^k$, in (5.33), is used to show that the boundary degrees of freedom are set to zero. After setting the related boundary degrees of freedom to zero, the rest of the multiplications are performed from right to left. Eventually, the degrees of freedom that corresponds to the internal degrees of freedom are extracted from the resulting vector. This is shown with the subscript $_i$ on the complete expression in (5.33).

Our aim was to compute $\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$ and, to achieve this goal, we are not only using the internal partitions of the component matrices but the full component matrices as shown in (5.33) explicitly. The important point to note from a substructuring perspective is that we are still using the component matrices and operations are performed on the component level. This practice is still inline with the central idea of *operations on the component level* coined in [37].

---

[6] Without considering the numerical efficiency issues

Inverse iteration

Having found $\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$, we have to perform the inverse iteration with the block diagonal stiffness to find the next iteration vector in the sequence. Namely, repeating here (5.27),

$$\tilde{\boldsymbol{x}}_{k+1} = \mathbf{K}_{sym,ii}^{-1}\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$$

where

$$\mathbf{K}_{sym,ii} = \begin{bmatrix} \mathbf{M}_{s,ii} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{f,ii} \end{bmatrix} \tag{5.34}$$

A particularly useful side of the block diagonal stiffness matrix in (5.34) is that $\mathbf{K}_{f,ii}$ is not singular. Moreover, for realistic three dimensional problems, the fluid part of these matrices is generally larger in size than their structural counterparts. We can use this information in order to decrease the factorization costs by using iterative solution techniques for systems involving $\mathbf{K}_{f,ii}$. Namely, a hydrid-solver can be developed to perform the inverse iteration step. Since $\mathbf{K}_{sym,ii}$ is a block diagonal matrix, the structural and the fluid parts of the matrix can be factorized independently for the following forward-backward solution steps. Moreover, our previous experience shows that $\mathbf{K}_{f,ii}$ is suitable for iterative solution techniques, such as the Preconditioned Conjugate Gradient method [76]. And the convergence is quite fast in the number of iterations. Although, from a condition number point of view, the above mentioned matrices are advantageous, during the inverse iteration process of the Lanczos technique, we have to perform solutions for many right hand side vectors which makes direct solvers a better candidate for the tests performed here. Therefore, we did not use iterative solution techniques for the inverse iterations performed with $\mathbf{K}_{f,ii}$[7].

With the shortly outlined methods in this Section, we can efficiently compute the fixed interface modes of the component by using the Lanczos technique [63].

5

Constraint modes

As briefly outlined in Section 5.3.1, the constraint modes of the component are written as follows,

$$\boldsymbol{\Psi}_c = -\mathbf{K}_{sym,ii}^{-1}\mathbf{K}_{sym,ib} \tag{5.35}$$

where, in this case, we used the symmetric component stiffness matrix and the related partitions of that matrix. The explicit form of the contraint modes can also be written in a block diagonal format, namely, as follows,

$$\boldsymbol{\Psi}_c = \begin{bmatrix} \boldsymbol{\Psi}_{c,s} \\ \boldsymbol{\Psi}_{c,f} \end{bmatrix} \equiv \begin{bmatrix} -\mathbf{M}_{s,ii}^{-1}\mathbf{M}_{s,ib} & \mathbf{0} \\ \mathbf{0} & -\mathbf{K}_{f,ii}^{-1}\mathbf{K}_{f,ib} \end{bmatrix} \tag{5.36}$$

---

[7] This is a possible future research topic for decreasing costs of the solvers in this component mode synthesis technique

Overall reduction basis

Having outlined the details of the calculation of the ingredients of the reduction basis on a τ-symmetric system, we can shortly outline the order of degrees of freedom to be used for the CB transformation and the overall reduction basis $\mathbf{R}_{cb}$.

The degrees of freedom of a component used for a CB transfomation over a vibro-acoustic component is written as follows,

$$\tilde{\boldsymbol{q}} = \begin{bmatrix} \tilde{\boldsymbol{q}}_b \\ \tilde{\boldsymbol{q}}_i \end{bmatrix} \equiv \begin{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \boldsymbol{p} \end{bmatrix}_b \\ \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \boldsymbol{p} \end{bmatrix}_i \end{bmatrix} \tag{5.37}$$

It is important to note that, for the selection in (5.37), we keep both the structural and fluid degrees of freedom on the interfaces and on the internal regions of the components. We believe that this is a natural partitioning for, almost, all the vibro-acoustic problems both in two and three dimensions. This approach is general for any partitioning of the vibro-acoustic domain: part of the interface could be fully structural or acoustical (then $\boldsymbol{p}_b$ or $\boldsymbol{u}_b$ would not exist), whereas another part could be coinciding with any vibro-acoustic boundary.

Under the light of the ongoing discussion, we can write the overall reduction basis explicitly as follows,

$$\begin{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \boldsymbol{p} \end{bmatrix}_b \\ \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \boldsymbol{p} \end{bmatrix}_i \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\mathbf{M}_{s,ii}^{-1}\mathbf{M}_{s,ib} & \mathbf{0} & \boldsymbol{\Phi}_{sym,ii,s} \\ \mathbf{0} & -\mathbf{K}_{f,ii}^{-1}\mathbf{K}_{f,ib} & \boldsymbol{\Phi}_{sym.ii,f} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \boldsymbol{p} \end{bmatrix}_b \\ \begin{bmatrix} \boldsymbol{\eta}_{\tilde{u},i} \\ \boldsymbol{\eta}_{p,i} \end{bmatrix} \end{bmatrix} \equiv \mathbf{R}_{cb} \begin{bmatrix} \tilde{\boldsymbol{q}}_b \\ \boldsymbol{\eta}_i \end{bmatrix} \tag{5.38}$$

or, more compactly,

$$\tilde{\boldsymbol{q}} = \mathbf{R}_{cb}\tilde{\boldsymbol{\eta}}, \qquad \tilde{\boldsymbol{\eta}} = \begin{bmatrix} \tilde{\boldsymbol{q}}_b \\ \boldsymbol{\eta}_i \end{bmatrix} \tag{5.39}$$

where, in (5.38), $\boldsymbol{\eta}_{\tilde{u},i}$ and $\boldsymbol{\eta}_{p,i}$ represent the modal coordinates of the internal structural and fluid parititons after the CB transformation.

5.3.4. Reduced matrices and their structure

Stiffness matrix, k

The closed form of the reduced stiffness matrix after a Craig-Bampton type reduction for a system with symmetric matrices is written as [63],

$$\mathbf{k} = \mathbf{R}_{cb}^{\mathrm{T}}(\mathbf{K}\tau)_{cb}\mathbf{R}_{cb} \equiv \mathbf{R}_{cb}^{\mathrm{T}}\mathbf{K}_{sym,cb}\mathbf{R}_{cb} \equiv \begin{bmatrix} \mathbf{k}_{bb} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega}_{ii} \end{bmatrix} \tag{5.40}$$

where $\mathbf{K}_{sym,cb}$ is the reordered form of the symmetric stiffness matrix, $\mathbf{K}\tau$[8]. In (5.40), $\boldsymbol{\Omega}_{ii}$ represents the diagonal matrix of internal eigenvalues resulting from (5.25),

---

[8] Otherwise, it is not possible to multiply correct terms with each other in a CB representation/ordering.

namely,

$$\mathbf{\Omega}_{ii} = \begin{bmatrix} \lambda_i^1 & & \\ & \ddots & \\ & & \lambda_i^m \end{bmatrix} \tag{5.41}$$

and m represents the selected number of internal modes for the component. $\mathbf{k}_{bb}$ represents the stiffness matrix which is statically condensed on the the interfaces of the component and written as follows,

$$\mathbf{k}_{bb} = \mathbf{K}_{sym,bb} - \mathbf{K}_{sym,bi}\mathbf{K}_{sym,ii}^{-1}\mathbf{K}_{sym,ib} \tag{5.42}$$

In order to find (5.42), we have to partition the block diagonal stiffness matrix of the component provided in (5.26b), namely,

$$\mathbf{K}_{sym} = \begin{bmatrix} \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}_{bb} & \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}_{bi} \\ \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}_{ib} & \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}_{ii} \end{bmatrix} \tag{5.43}$$

Placing the appropriate matrices into the closed format representation of the reduced stiffness matrix, namely in (5.42), we can write the boundary partition of the reduced stiffness matrix of any component as

$$\mathbf{k}_{bb} = \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}_{bb} - \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}_{bi} \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}_{ii}^{-1} \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}_{ib} \tag{5.44}$$

which can be reduced to a block diagonal matrix, namely,

$$\mathbf{k}_{bb} = \begin{bmatrix} \mathbf{M}_{s,bb} - \mathbf{M}_{s,bi}\mathbf{M}_{s,ii}^{-1}\mathbf{M}_{s,ib} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{f,bb} - \mathbf{K}_{f,bi}\mathbf{K}_{f,ii}^{-1}\mathbf{K}_{f,ib} \end{bmatrix} \tag{5.45}$$

or, equivalently into,

$$\mathbf{k}_{bb} = \begin{bmatrix} \mathbf{M}_{s,bb} + \mathbf{M}_{s,bi}\mathbf{\Psi}_{c,s} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{f,bb} + \mathbf{K}_{f,bi}\mathbf{\Psi}_{c,f} \end{bmatrix} \tag{5.46}$$

by using the definitions of $\mathbf{\Psi}_{c,s}$ and $\mathbf{\Psi}_{c,f}$ from (5.35).

It is worth mentioning that $\mathbf{k}_{bb}$ found with a normal Craig&Bampton transformation in (5.40) is a fully populated matrix. For vibroacoustic problems, however, we can write $\mathbf{k}_{bb}$ as a block diagonal matrix composed of two smaller fully populated matrices as shown in (5.46).

Mass matrix, m

In contrast to the stiffness matrix, we do not have a special structure for the symmetric mass matrix of component. In order to end up with the symmetric reduced mass matrix, we have to perform the projection written as

$$\mathbf{m} = \mathbf{R}_{cb}^{\mathrm{T}}(\mathbf{M}\mathcal{T})_{cb}\mathbf{R}_{cb} \equiv \mathbf{R}_{cb}^{\mathrm{T}}\mathbf{M}_{sym,cb}\mathbf{R}_{cb} \tag{5.47}$$

Assuming that we can perform the above projection, we can write the closed form representation of the reduced mass matrix [63] as follows, namely,

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}_{bb} & \mathbf{m}_{bm} \\ \mathbf{m}_{mb} & \mathbf{I} \end{bmatrix} \tag{5.48}$$

with the fully populated matrices

$$\begin{aligned} \mathbf{m}_{bb} =& \mathbf{M}_{bb} - \mathbf{M}_{bi}\mathbf{K}_{ii}{}^{-1}\mathbf{K}_{ib} \\ & - \mathbf{K}_{ib}\mathbf{K}_{ii}{}^{-1}\mathbf{M}_{ib} + \mathbf{K}_{bi}\mathbf{K}_{ii}{}^{-1}\mathbf{M}_{ii}\mathbf{K}_{ii}{}^{-1}\mathbf{K}_{ib} \end{aligned} \tag{5.49}$$

$$\mathbf{m}_{mb} = \mathbf{m}_{bm}{}^{\mathrm{T}} \equiv \mathbf{\Phi}_{ii}{}^{\mathrm{T}} \left(\mathbf{M}_{ib} - \mathbf{M}_{ii}\mathbf{K}_{ii}{}^{-1}\mathbf{K}_{ib}\right) \tag{5.50}$$

However, the closed form of the symmetric mass matrix for the component was written in (5.23) and it was also mentioned before that, from a numerical perspective, it is not feasible to compute the closed form representation of the mass matrix due to its fully populated nature. This, in turn, means that we can not extract the related matrix partitions, $\mathbf{M}^{bb}$, $\mathbf{M}^{ib}$ and $\mathbf{M}^{ii}$, in (5.49) and in (5.50). As a first impression, it might not seem possible to compute the reduced mass matrix in an efficient manner. However, we can make use of the same trick we used to compute $\mathbf{M}_{sym,ii}\boldsymbol{x}_i^k$ in Section 5.3.3 along with a reordering on the rows of $\mathbf{R}_{cb}$.

Namely, in a matrix multiplication operation written as $\mathbf{C} = \mathbf{A}^{\mathrm{T}}\mathbf{B}\mathbf{A}$, the resulting matrix, $\mathbf{C}$, is not going to change if we can order the rows of matrix $\mathbf{A}$[9] and also both the rows and columns of matrix $\mathbf{B}$[10] in the same order. In other words, this order change does not affect the numerical values and the location of the terms of the resulting matrix $\mathbf{C}$. Namely,

$$\mathbf{C} = \mathbf{A}^{\mathrm{T}}\mathbf{B}\mathbf{A} \equiv \tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{B}}\tilde{\mathbf{A}} \tag{5.51}$$

where $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ represent the reordered matrices. Therefore, we can use this simple algebraic property to convert the order of rows in $\mathbf{R}_{cb}$ to perfom the multiplication in (5.47). Namely, the matrices representing the symmetric mass matrix in (5.47) is not ordered in this specific case but only $\mathbf{R}_{cb}$ has to be ordered. To perform the reordering on rows of $\mathbf{R}_{cb}$, we have to find the mapping between the degrees of freedom in the CB order and the given $(\boldsymbol{u} - \boldsymbol{p})$ order over the components. Namely, the list of the degrees of fredom in

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_b \\ \boldsymbol{q}_i \end{bmatrix} \equiv \begin{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{bmatrix}_b \\ \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{bmatrix}_i \end{bmatrix} \tag{5.52}$$

has be to re-ordered back to the list of degrees of freedom in

$$\boldsymbol{q}_{\boldsymbol{up}} = \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{bmatrix} \tag{5.53}$$

---

[9] which corresponds to $\mathbf{R}_{cb}$ in actual operation

[10] which corresponds to $\mathbf{M}_{\mathcal{T}}$ in actual operation

In brief, we have to set up and use a proper mapping between the list of degrees of freedom in $\boldsymbol{q}$ and $\boldsymbol{q_{up}}$. We are going to denote this specific mapping between the order of degrees of freedom with $\boldsymbol{q} \leftrightarrow \boldsymbol{q_{up}}$.

By using the mapping between $\boldsymbol{q_{up}}$ and $\boldsymbol{q}$, we can reorder the rows of $\mathbf{R}_{cb}$ to $(\boldsymbol{u}-\boldsymbol{p})$ order over the component, namely,

$$\mathbf{R}_{cb} \xrightarrow{(\boldsymbol{q} \leftrightarrow\, \boldsymbol{q_{up}})} \begin{bmatrix} \mathbf{R}_{cb,\boldsymbol{u}} \\ \mathbf{R}_{cb,\boldsymbol{p}} \end{bmatrix} \tag{5.54}$$

where $\mathbf{R}_{cb,\boldsymbol{u}}$ represents the part of $\mathbf{R}_{cb}$ that matches with the structural degrees of freedom of the component. Similarly, $\mathbf{R}_{cb,\boldsymbol{p}}$ represents the part of $\mathbf{R}_{cb}$ that corresponds to the fluid degrees of freedom.

After the ordering, we can write the explicit form of the reduced mass matrix in (5.47), namely,

$$\mathbf{m} = \begin{bmatrix} \mathbf{R}_{cb,\boldsymbol{u}} \\ \mathbf{R}_{cb,\boldsymbol{p}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f \end{bmatrix} \begin{bmatrix} \mathbf{K}_s^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{M}_s & \mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{cb,\boldsymbol{u}} \\ \mathbf{R}_{cb,\boldsymbol{p}} \end{bmatrix} \tag{5.55}$$

Another interesting and important note for the computations in (5.55) is that we can store some information and reuse this information later. Namely, the stored information is going to be used during the coupling of component matrices as will be explained in Section 5.5. To show this, we are going to perform the operations in the following order:

- First perform the multiplication of the following matrix terms:

$$\begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{K}_s^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{M}_s & \mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{cb,\boldsymbol{u}} \\ \mathbf{R}_{cb,\boldsymbol{p}} \end{bmatrix} \tag{5.56}$$

  - While performing the operations, we can store the partial result in the first multiplication from the right

$$\mathbf{a} = \mathbf{M}_s \mathbf{R}_{cb,\boldsymbol{u}} + \mathbf{K}_{sf} \mathbf{R}_{cb,\boldsymbol{p}} \tag{5.57}$$

- With this substitution given in (5.57), (5.56) can be written as,

$$\begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{K}_s^{-1}\mathbf{a} \\ \mathbf{R}_{cb,\boldsymbol{p}} \end{bmatrix} \tag{5.58}$$

- Next, perform the multiplication of the first two matrices in (5.55) by also making use of (5.57), namely,

$$\mathbf{b} = \left[\left(\mathbf{R}_{cb,\boldsymbol{u}}^{\mathrm{T}} \mathbf{M}_s + \mathbf{R}_{cb,\boldsymbol{p}}^{\mathrm{T}} \mathbf{K}_{sf}^{\mathrm{T}} \right) \quad \mathbf{R}_{cb,\boldsymbol{p}}^{\mathrm{T}} \mathbf{M}_f \right] \equiv \left[\mathbf{a}^{\mathrm{T}} \quad \mathbf{R}_{cb,\boldsymbol{p}}^{\mathrm{T}} \mathbf{M}_f \right] \tag{5.59}$$

- Eventually, compute the resulting reduced mass matrix with (5.58) and (5.59)

$$\mathbf{m} = \underbrace{\left[\mathbf{a}^{\mathrm{T}} \quad \mathbf{R}_{cb,\boldsymbol{p}}^{\mathrm{T}} \mathbf{M}_f \right]}_{\mathbf{b}} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} \equiv \mathbf{a}^{\mathrm{T}} \mathbf{K}_s^{-1} \mathbf{a} + \mathbf{R}_{cb,\boldsymbol{p}}^{\mathrm{T}} \mathbf{M}_f \mathbf{R}_{cb,\boldsymbol{p}} \tag{5.60}$$

(5.60) has the same format as presented in (5.48) provided that the fixed interface modes computed with the techniques outlined in Section 5.3.3 are mass normalized.

5.4. Assembling reduced component matrices

In Section 5.3, we outlined the details on how we can end up with reduced symmetric component matrices with all the computational details. In this section, we will show how these reduced component matrices can be assembled together to compute the global system dynamics. The general framework presented in this section is based on [90]. The outline is as follows. In Section 5.4.1, we are going to outline the general mathematical framework of component coupling. In Section 5.4.2, we are going to show the effects of creating symmetric matrices on the compatibility and equilibrium relations between the components. In Section 5.4.3, we are going to provide the details of coupling reduced component matrices found in Section 5.3.

5.4.1. Component coupling with symmetric matrices

Coupling framework in the physical domain

In general when a physical domain is modelled with the finite element technique and subsequently divided into components, the equation of motion of a component s, which is in connection with other components, can be written as,

$$\mathbf{M}^{(s)}\ddot{\boldsymbol{q}}^{(s)} + \mathbf{K}^{(s)}\boldsymbol{q}^{(s)} = \boldsymbol{f}^{(s)} + \boldsymbol{g}^{(s)} \tag{5.61}$$

where $\mathbf{M}^{(s)}$ represents the symmetric mass matrix of the component, $\mathbf{K}^{(s)}$ represents the symmetric stiffness matrix of the component, $\boldsymbol{q}^{(s)}$ represents the vector of some physical degrees of freedom, $\boldsymbol{f}^{(s)}$ and $\boldsymbol{g}^{(s)}$ represent the external forces acting on the component, if any, and the vector of interface forces that originate from the neighbouring components, respectively.

Components interacting with other components can be coupled provided that the two following conditions are met, namely,

1. Displacements at the interfaces of the components should match, which is the compatibility condition,
2. Force equilibrium on the interfaces should be satisfied, which is the equilibrium condition.

The general system of equations of motion of k components which are to be coupled can written in a block diagonal format as

$$\mathbf{M}\ddot{\boldsymbol{q}} + \mathbf{K}\boldsymbol{q} = \boldsymbol{f} + \boldsymbol{g} \tag{5.62}$$

with

$$\mathbf{M} = \mathrm{diag}(\mathbf{M}^{(1)}, \ldots, \mathbf{M}^{(k)}) \equiv \begin{bmatrix} \mathbf{M}^{(1)} & & \\ & \ddots & \\ & & \mathbf{M}^{(k)} \end{bmatrix} \tag{5.63}$$

$$\mathbf{K} = \mathrm{diag}(\mathbf{K}^{(1)}, \ldots, \mathbf{K}^{(k)}) \tag{5.64}$$

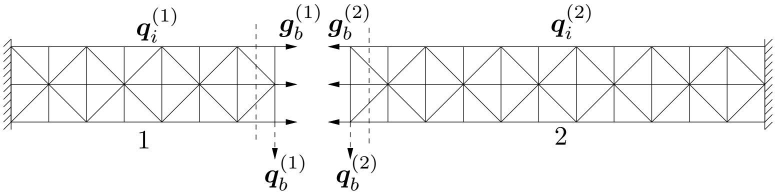


Figure 5.2: A simple domain divided into two components, no external forces

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}^{(1)} \\ \vdots \\ \boldsymbol{q}^{(k)} \end{bmatrix}, \quad \boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}^{(1)} \\ \vdots \\ \boldsymbol{f}^{(k)} \end{bmatrix}, \quad \boldsymbol{g} = \begin{bmatrix} \boldsymbol{g}^{(1)} \\ \vdots \\ \boldsymbol{g}^{(k)} \end{bmatrix} \tag{5.65}$$

In the following two small subsections, we are going to shortly outline the displacement and force compatibility requirements in order to couple components with matching meshes. To explain the concepts, we are going to make use of the simple model depicted in Figure 5.2. Figure 5.2 also depicts the degrees of freedom used for the interfaces, $\boldsymbol{q}_b$, and the degrees of freedom internal to the component, namely, $\boldsymbol{q}_i$. In the same figure, we can also see the connecting interface forces, namely, $\boldsymbol{g}_b^{(1)}$ and $\boldsymbol{g}_b^{(2)}$ for component 1 and 2, respectively.

Compatibility condition and Boolean matrices

Compatibility condition requires that the displacements on the interfaces must be equal. Investigating Figure 5.2, this condition can be easily written in the following form, namely

$$\boldsymbol{q}_b^{(1)} = \boldsymbol{q}_b^{(2)} \longleftrightarrow \boldsymbol{q}_b^{(1)} - \boldsymbol{q}_b^{(2)} = \boldsymbol{0} \tag{5.66}$$

Recalling the definition of $\boldsymbol{q}$ from (5.65), we can also partition the individual components of that vector into internal and boundary degrees of freedom, namely

$$\boldsymbol{q}^{(k)} = \begin{bmatrix} \boldsymbol{q}_b^{(k)} \\ \boldsymbol{q}_i^{(k)} \end{bmatrix} \tag{5.67}$$

With this scheme, the complete set of degrees of freedom in Figure 5.2 can be written as,

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_b^{(1)} \\ \boldsymbol{q}_i^{(1)} \\ \boldsymbol{q}_b^{(2)} \\ \boldsymbol{q}_i^{(2)} \end{bmatrix} \tag{5.68}$$

By making use of the definition in (5.68), we can represent the compatibility condition presented in (5.66) with a signed Boolean matrix notation which is introduced in [90] leading to

$$\begin{bmatrix}\mathbf{I} & \mathbf{0} & -\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{q}_b^{(1)}\\ \boldsymbol{q}_i^{(1)}\\ \boldsymbol{q}_b^{(2)}\\ \boldsymbol{q}_i^{(2)}\end{bmatrix} = \mathbf{0} \Longleftrightarrow \mathbf{B}\boldsymbol{q} = \mathbf{0} \tag{5.69}$$

where the *identity* blocks correspond to matching interface degrees of freedom. It is also possible to present (5.69) as follows

$$\begin{bmatrix}\mathbf{B}^{(1)} & \mathbf{B}^{(2)}\end{bmatrix}\begin{bmatrix}\boldsymbol{q}^{(1)}\\ \boldsymbol{q}^{(2)}\end{bmatrix} = \mathbf{0} \qquad \text{with} \quad \mathbf{B}^{(1)} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}, \mathbf{B}^{(2)} = \begin{bmatrix}-\mathbf{I} & \mathbf{0}\end{bmatrix} \tag{5.70}$$

By extending (5.70), we can generalize the signed Boolean compatibility condition (5.70) for k components as follows [35],

$$\mathbf{B}\boldsymbol{q} = \begin{bmatrix}\mathbf{B}^{(1)} & \dots & \mathbf{B}^{(k)}\end{bmatrix}\begin{bmatrix}\boldsymbol{q}^{(1)}\\ \vdots\\ \boldsymbol{q}^{(k)}\end{bmatrix} \equiv \mathbf{0} \tag{5.71}$$

Each $\mathbf{B}^{(k)}$ in $\mathbf{B}$ is written as

$$\mathbf{B}^{(k)} = \begin{bmatrix}\mathbf{B}_b^{(k)} & \mathbf{0}\end{bmatrix} \tag{5.72}$$

where the zero block corresponds to the internal degrees of freedom and $\mathbf{B}_b^{(k)}$ corresponds to the boundary degrees of freedom.

It is important to note that (5.71) represents the constraints on the matching degrees of freedom on the interfaces of the components. Namely, each row in the compatibility matrix represents a constraint equation. Therefore, the row size of the compatibility matrix is equal to the number of constraints on the degrees of freedom. A more detailed presentation with a simple example is also given in the Appendix of [90].

Another form of compatibility condition is closely related to the finite element assembly procedure. Namely, it is represented as a mapping between the degrees of freedom of the components and a unique set of degrees of freedom representing the coupled or global system behavior. For the system in Figure 5.2, we can define this mapping as follows:

$$\begin{bmatrix}\boldsymbol{q}_b^{(1)}\\ \boldsymbol{q}_i^{(1)}\\ \boldsymbol{q}_b^{(2)}\\ \boldsymbol{q}_i^{(2)}\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{I} & \mathbf{0}\\ \mathbf{I} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{q}_b^{(1)}\\ \boldsymbol{q}_i^{(1)}\\ \boldsymbol{q}_i^{(2)}\end{bmatrix} \longleftrightarrow \boldsymbol{q} = \mathbf{L}\tilde{\boldsymbol{q}} \tag{5.73}$$

where $\tilde{\boldsymbol{q}}$ represents the unique set of degrees of freedom of the coupled system[11]. $\mathbf{L}$ represents the *Boolean localization matrix*. Similar matrices to the *Localization*

---

[11] It is important to reemphasize that there are no repetitions in this set.

matrix are also used in finite element literature to perform the direct assembly of element matrices into a global system matrix[55]. However, in the context of component mode synthesis, these matrices are used to assemble components, in other words *super-elements*, which are built up of many elements.

It is possible to device a useful relation between **B** and **L**. We can observe this relation by using (5.73) in (5.71) which results in

$$\mathbf{B}\mathbf{L}\tilde{\boldsymbol{q}} = \mathbf{0} \Longrightarrow \begin{cases} \mathbf{L} = \mathrm{null}(\mathbf{B}) \\ \mathbf{B}^{\mathrm{T}} = \mathrm{null}(\mathbf{L}^{\mathrm{T}}) \end{cases} \tag{5.74}$$

As a conclusion, **L** is the nullspace of **B**. **L** and **B** are used in the primal and dual assembly approaches which are going to be discussed in the following sections.

Force equilibrium on the interfaces

Force equilibrium between the matching interface degrees of freedom is the second condition for the assembly of components. Originating from Newton's third law of motion, it states that equal but opposite forces must act on the individual components connecting to a common interface. Returning to Figure 5.2, we can define this relation for the connecting interface line as follows:

$$\boldsymbol{g}_b^{(1)} + \boldsymbol{g}_b^{(2)} = \mathbf{0} \tag{5.75}$$

If we define the interface forces per component as follows

$$\boldsymbol{g}^{(k)} = \begin{bmatrix} \boldsymbol{g}_b^{(k)} \\ \mathbf{0} \end{bmatrix} \tag{5.76}$$

then the global interface force vector for the system in Figure 5.2 can be written as

$$\boldsymbol{g} = \begin{bmatrix} \boldsymbol{g}^{(1)} \\ \boldsymbol{g}^{(2)} \end{bmatrix} \tag{5.77}$$

The force compatibility can also be extracted by multiplying the global interface force vector $\boldsymbol{g}$ with the transpose of **L** and equating the result to **0**, namely,

$$\mathbf{L}^{\mathrm{T}}\boldsymbol{g} = \mathbf{0} \longrightarrow \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{g}_b^{(1)} \\ \mathbf{0} \\ \boldsymbol{g}_b^{(2)} \\ \mathbf{0} \end{bmatrix} = \mathbf{0} \longrightarrow \begin{bmatrix} \boldsymbol{g}_b^{(1)} + \boldsymbol{g}_b^{(2)} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{5.78}$$

where we can observe that the first row gives the force compatibility we search for. The rest of the rows pertain to the internal degrees of freedom of the components which do not contribute to the force compatibility on the interfaces but kept here for completeness.

The global interface force vector $\boldsymbol{g}$ can also be written by considering a common set of *Lagrange multipliers*, $\boldsymbol{\lambda}$. Namely, these *Lagrange multipliers* represent the intensity of the forces connecting the matching interface degrees of freedom. Since

$\boldsymbol{g}_b^{(1)}$ and $\boldsymbol{g}_b^{(2)}$ must act in opposite directions, we can write the global interface force vector with the help of the above defined Boolean compatibility matrix, $\mathbf{B}$, namely,

$$\boldsymbol{g} = -\mathbf{B}^{\mathrm{T}}\boldsymbol{\lambda} \equiv \begin{bmatrix} -\boldsymbol{\lambda} \\ \mathbf{0} \\ \boldsymbol{\lambda} \\ \mathbf{0} \end{bmatrix} \tag{5.79}$$

It is important to note at this point that, by defining the interface forces with (5.79), the interface equilibrium is automatically satisfied, namely,

$$\mathbf{L}^{\mathrm{T}}\boldsymbol{g} = \mathbf{0} \longrightarrow -\mathbf{L}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{0} \tag{5.80}$$

which is also proven once more by (5.74).

Equations (5.62), (5.71) and (5.78) represent the general format of the equations which are going to be used for coupling different components, namely,

$$\mathbf{M}\ddot{\boldsymbol{q}} + \mathbf{K}\boldsymbol{q} = \boldsymbol{f} + \boldsymbol{g} \tag{5.81a}$$

$$\mathbf{B}\boldsymbol{q} = \mathbf{0} \tag{5.81b}$$

$$\mathbf{L}^{\mathrm{T}}\boldsymbol{g} = \mathbf{0} \tag{5.81c}$$

Two different coupling procedures can be used in order to couple different components. Namely, primal and dual assembly approaches can be employed [90]. In the next two sections, we are going to outline these two approaches shortly.

Primal assembly in physical domain

Primal assembly approach works by transforming the dual set of degrees of freedom defined in (5.65) into a vector having a unique set of degrees of freedom by using (5.73) and (5.78). In essence, it is similar to the finite element assembly of element matrices. During this transformation, the interface forces are also eliminated. Namely, we can write (5.81a)-(5.81c) as follows

$$\mathbf{M}\mathbf{L}\ddot{\tilde{\boldsymbol{q}}} + \mathbf{K}\mathbf{L}\tilde{\boldsymbol{q}} = \boldsymbol{f} + \boldsymbol{g} \tag{5.82}$$

$$\mathbf{L}^{\mathrm{T}}\boldsymbol{g} = \mathbf{0} \tag{5.83}$$

noting that, due to (5.74), (5.81b) always holds. Premultiplication of (5.82) by $\mathbf{L}^{\mathrm{T}}$, noting also $\mathbf{L}^{\mathrm{T}}\boldsymbol{g} = \mathbf{0}$ with which we cancel the internal connecting forces, results in the primal assembled system of equations in the physical coordinate space,

$$\tilde{\mathbf{M}}\ddot{\tilde{\boldsymbol{q}}} + \tilde{\mathbf{K}}\tilde{\boldsymbol{q}} = \tilde{\boldsymbol{f}} \longleftrightarrow \begin{cases} \tilde{\mathbf{M}} & = \mathbf{L}^{\mathrm{T}}\mathbf{M}\mathbf{L} \\ \tilde{\mathbf{K}} & = \mathbf{L}^{\mathrm{T}}\mathbf{K}\mathbf{L} \\ \tilde{\boldsymbol{f}} & = \mathbf{L}^{\mathrm{T}}\boldsymbol{f} \end{cases} \tag{5.84}$$

Dual assembly in physical domain

In a dual assembly framework, *Lagrange multipliers*, namely, the interface forces, also become a part of the system equations. Namely, by integrating (5.79) into (5.81a)-

(5.81c), we can write the dual representation of the problem as follows:

$$\mathbf{M}\ddot{\boldsymbol{q}} + \mathbf{K}\boldsymbol{q} + \mathbf{B}^{\mathrm{T}}\boldsymbol{\lambda} = \boldsymbol{f} \tag{5.85a}$$

$$\mathbf{B}\boldsymbol{q} = \mathbf{0} \tag{5.85b}$$

Note that in transforming (5.81a)-(5.81c) to (5.85a)-(5.85b), we make use of

$$\mathbf{L}^{\mathrm{T}}\boldsymbol{g} = -\mathbf{L}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{0} \tag{5.86}$$

which is always satisfied due to the null space properties. (5.85a)-(5.85b) can be recast into a matrix format as follows

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \mathbf{K} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \mathbf{0} \end{bmatrix} \tag{5.87}$$

Coupling of reduced component matrices

As outlined with details in Section 5.3, we can perform the reduction on the component level by a reduction matrix. To extend this practice to more components which have symmetric system matrices, a block diagonal basis filled with independent component reduction bases is constructed. Namely,

$$\boldsymbol{q} = \mathbf{R}\boldsymbol{\eta}, \tag{5.88}$$

with

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}^{(1)} \\ \boldsymbol{q}^{(2)} \\ \vdots \\ \boldsymbol{q}^{(k)} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}^{(1)} & & \\ & \ddots & \\ & & \mathbf{R}^{(k)} \end{bmatrix}, \quad \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}^{(1)} \\ \boldsymbol{\eta}^{(2)} \\ \vdots \\ \boldsymbol{\eta}^{(k)} \end{bmatrix} \tag{5.89}$$

In (5.89), $\boldsymbol{q}^{(k)}$, $\mathbf{R}^{(k)}$ and $\boldsymbol{\eta}^{(k)}$ represent the physical degrees of freedom, reduction basis and the generalized degrees of freedom of component k, respectively.

Substituting (5.88) in (5.81a), we find

$$\mathbf{M}\mathbf{R}\ddot{\eta} + \mathbf{K}\mathbf{R}\eta = \boldsymbol{f} + \boldsymbol{g} + \boldsymbol{r} \tag{5.90}$$

Following the analysis performed in Section 5.2.1, we force the residual to be orthogonal to the selected basis, $\mathbf{R}$, namely, $\mathbf{R}^{\mathrm{T}}\boldsymbol{r} = \mathbf{0}$. Then, the reduced decoupled equations of motion of the components can be written as

$$\underbrace{\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}}_{\mathbf{M}_m}\ddot{\eta} + \underbrace{\mathbf{R}^{\mathrm{T}}\mathbf{K}\mathbf{R}}_{\mathbf{K}_m}\eta = \underbrace{\mathbf{R}^{\mathrm{T}}\boldsymbol{f}}_{\boldsymbol{f}_m} + \underbrace{\mathbf{R}^{\mathrm{T}}\boldsymbol{g}}_{\boldsymbol{g}_m} \tag{5.91}$$

where $\mathbf{M}_m$ and $\mathbf{K}_m$ represent the matrices storing the non-assembled reduced component matrices (or, modal matrices) in a block diagonal format. To couple these reduced but decoupled system equations, we also need to impose the compatibility and equilibrium requirements for the generalized coordinates.

The compatibility equation in the generalized coordinates is written by substituting (5.88) in (5.81b) as

$$\mathbf{BR}\boldsymbol{\eta} = \mathbf{0} \equiv \mathbf{B}_m\boldsymbol{\eta} = \mathbf{0}, \quad \text{with} \quad \mathbf{B}_m = \mathbf{BR} \tag{5.92}$$

In analogy to the physical domain, to end up with a unique set of generalized coordinate set, we also have to define the following transformation over the generalized degrees of freedom, namely,

$$\boldsymbol{\eta} = \mathbf{L}_m\boldsymbol{\xi} \tag{5.93}$$

where, after substituting (5.93) in (5.92), one can easily see that

$$\mathbf{B}_m\boldsymbol{\eta} \equiv \mathbf{B}_m\mathbf{L}_m\boldsymbol{\xi} = \mathbf{0}, \tag{5.94}$$

which means that $\mathbf{L}_m$ must span the null space of $\mathbf{B}_m$. If the interface degrees of freedom of the component are kept as generalized degrees of freedom then $\mathbf{B}_m$ is still a Boolean matrix, for instance, when a CB-type reduction basis is used as in this research. After the above discussion, we can also write the following set of equations for the generalized coordinates, which are analogous to (5.81a)-(5.81c), namely,

$$\mathbf{M}_m\ddot{\boldsymbol{\eta}} + \mathbf{K}_m\boldsymbol{\eta} = \boldsymbol{f}_m + \boldsymbol{g}_m \tag{5.95a}$$

$$\mathbf{B}_m\boldsymbol{\eta} = \mathbf{0} \tag{5.95b}$$

$$\mathbf{L}_m^{\mathrm{T}}\boldsymbol{g}_m = \mathbf{0} \tag{5.95c}$$

Primal assembly with reduced components

Substituting (5.93) in (5.95a)-(5.95c), we directly make use of the vector set $\boldsymbol{\xi}$ with a unique set of interface degrees on freedom, namely,

$$\mathbf{M}_m\mathbf{L}_m\ddot{\boldsymbol{\xi}} + \mathbf{K}_m\mathbf{L}_m\boldsymbol{\xi} = \boldsymbol{f}_m + \boldsymbol{g}_m \tag{5.96a}$$

$$\mathbf{B}_m\mathbf{L}_m\boldsymbol{\xi} = \mathbf{0} \tag{5.96b}$$

$$\mathbf{L}_m^{\mathrm{T}}\boldsymbol{g}_m = \mathbf{0} \tag{5.96c}$$

(5.96b) is automatically satisfied since $\mathbf{L}_m = \text{null}\,(\mathbf{B}_m)$. Premultiplication of (5.96a) with $\mathbf{L}_m^{\mathrm{T}}$ and noting that $\mathbf{L}_m^{\mathrm{T}}\boldsymbol{g}_m = \mathbf{0}$ yields the primal assembled system for the coupled structures as

$$\mathbf{m}^g\ddot{\boldsymbol{\xi}} + \mathbf{k}^g\boldsymbol{\xi} = \boldsymbol{f}^g \qquad \text{with} \quad \begin{cases} \mathbf{m}^g & = \mathbf{L}_m^{\mathrm{T}}\mathbf{M}_m\mathbf{L}_m \\ \mathbf{k}^g & = \mathbf{L}_m^{\mathrm{T}}\mathbf{K}_m\mathbf{L}_m \\ \boldsymbol{f}^g & = \mathbf{L}_m^{\mathrm{T}}\boldsymbol{f}_m \end{cases} \tag{5.97}$$

Dual assembly with reduced components

We can write the dually assembled system by imposing the interface equilibrium by choosing the interface forces in generalized coordinates as follows,

$$\boldsymbol{g}_m = -\mathbf{B}_m^{\mathrm{T}}\boldsymbol{\lambda} \tag{5.98}$$

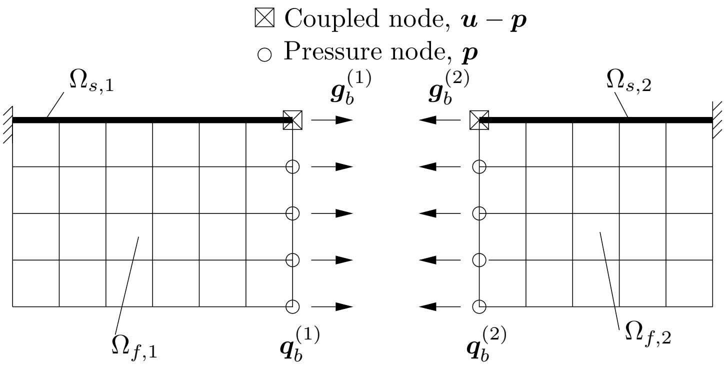


Figure 5.3: Interface variables on a 2D vibro-acoustic domain, $\boldsymbol{g}_b^{(1)}$, $\boldsymbol{g}_b^{(2)}$, $\boldsymbol{q}_b^{(1)}$, $\boldsymbol{q}_b^{(2)}$

where $\boldsymbol{\lambda}$ corresponds to the force intensities. The force equilibrium is now always satisfied due to (5.96c),

$$-\mathbf{L}_m^{\mathrm{T}}\mathbf{B}_m^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{0} \tag{5.99}$$

The equations of motion of the dually assembled system in the generalized coordinates can therefore be written as

$$\begin{bmatrix} \mathbf{M}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}} \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_m & \mathbf{B}_m^{\mathrm{T}} \\ \mathbf{B}_m & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_m \\ \mathbf{0} \end{bmatrix} \tag{5.100}$$

5.4.2. Effect of symmetric reduced component matrices on assembly

In Section 5.3, we provided the details of the reduction performed on symmetric component matrices but did not explicitly mention the effect of creating reduced symmetric component matrices on component coupling. Moreover, in Section 5.3.2, we proposed two options that facilitate to write symmetric component matrices, namely, either by using κ or τ. Also, in Section 5.3.3, we outlined some numerical ways to come up with an efficient framework for the calculations to be performed. Some important additional points will be outlined here to provide a basis to couple the reduced component matrices while using τ-symmetric system matrices on the component level. The common effect of the transformations performed either with τ or κ is that they modify the interface variables. These interface variables are also depicted in Figure 5.3 which shows two vibro-acoustic components with the corresponding coupled fluid and structural domains, Ω_f and Ω_s, respectively.

Explicitly writing the equations of motion for the τ-symmetric system components, we get

$$\mathbf{M}_c^{(1)}\tau^{(1)}\ddot{\tilde{\boldsymbol{q}}}^{(1)} + \mathbf{K}_c^{(1)}\tau^{(1)}\tilde{\boldsymbol{q}}^{(1)} = \boldsymbol{f}^{(1)} + \boldsymbol{g}^{(1)} \tag{5.101a}$$

$$\mathbf{M}_c^{(2)}\tau^{(2)}\ddot{\tilde{\boldsymbol{q}}}^{(2)} + \mathbf{K}_c^{(2)}\tau^{(2)}\tilde{\boldsymbol{q}}^{(2)} = \boldsymbol{f}^{(2)} + \boldsymbol{g}^{(2)} \tag{5.101b}$$

It is important to note that, in (5.101a) and (5.101b), the main variables of the component were transferred from $\boldsymbol{q}$ to $\tilde{\boldsymbol{q}}$ according to the relation in (5.19). As a result, the structural part of the interface variables $\boldsymbol{q}_b^{(1)}$ and $\boldsymbol{q}_b^{(2)}$ have to be updated. However, with this transformation, the interface forces, $\boldsymbol{g}^{(1)}$ and $\boldsymbol{g}^{(2)}$, are not affected. Therefore, with the transformations performed with τ, the first condition of component coupling will be affected, namely, the compatibility condition[12] between the interface degrees of freedom.

Notes for κ-transformation
τ-symmetric system transformations affected the interface degrees of freedom, and, in contrast, the κ-symmetric system transformations affect the interface forces. Namely, we can write the κ-symmetric equations of motions for the two components depicted in Figure 5.3 as follows

$$\kappa^{(1)}\mathbf{M}_c^{(1)}\ddot{\tilde{\boldsymbol{q}}}^{(1)} + \kappa^{(1)}\mathbf{K}_c^{(1)}\boldsymbol{q}^{(1)} = \kappa^{(1)}\boldsymbol{f}^{(1)} + \kappa^{(1)}\boldsymbol{g}^{(1)} \tag{5.102a}$$

$$\kappa^{(2)}\mathbf{M}_c^{(2)}\ddot{\tilde{\boldsymbol{q}}}^{(2)} + \kappa^{(2)}\mathbf{K}_c^{(2)}\boldsymbol{q}^{(2)} = \kappa^{(2)}\boldsymbol{f}^{(2)} + \kappa^{(2)}\boldsymbol{g}^{(2)} \tag{5.102b}$$

In (5.102a) and (5.102b), we can observe the effect of κ-transformation on the interface forces, $\boldsymbol{g}^{(1)}$ and $\boldsymbol{g}^{(2)}$. Namely, the interface forces are modified with the corresponding scaling matrix of the component under consideration, $\kappa^{(1)}\boldsymbol{g}^{(1)}$ and $\kappa^{(2)}\boldsymbol{g}^{(2)}$. In otherwords, second condition for component coupling, mentioned in Section 5.4.1, was affected by these transformations, namely, the force equilibrium between the components on the interface.

5.4.3. Assembly of τ-symmetric reduced vibro-acoustic component matrices

Having pointed out the effects of creating symmetric component matrices, namely, the changes on the interface forces and on the interface degrees of freedom, we are going to continue our discussions with τ-symmetric systems. Namely, the details of the assembly of τ-symmetric reduced components is going to be explained next. The reasons for selecting τ-symmetric systems were detailed in Section 5.3.3 from a numerical perspective and the reduction is performed based on the developed theory therein.

We can write the τ-symmetric equivalent of (5.81a)-(5.81c) by introducing $\boldsymbol{q} = \tau\tilde{\boldsymbol{q}}$ as follows,

$$\mathbf{M}\tau\ddot{\tilde{\boldsymbol{q}}} + \mathbf{K}\tau\tilde{\boldsymbol{q}} = \boldsymbol{f} + \boldsymbol{g} \tag{5.103a}$$

$$\mathbf{B}\tau\tilde{\boldsymbol{q}} = \mathbf{0} \tag{5.103b}$$

$$\mathbf{L}^{\mathrm{T}}\boldsymbol{g} = \mathbf{0} \tag{5.103c}$$

where

$$\tau = \begin{bmatrix} \tau^{(1)} & & \\ & \ddots & \\ & & \tau^{(k)} \end{bmatrix}, \qquad \tilde{\boldsymbol{q}} = \begin{bmatrix} \tilde{\boldsymbol{q}}^{(1)} \\ \vdots \\ \tilde{\boldsymbol{q}}^{(k)} \end{bmatrix} \tag{5.104}$$

---

[12] Matching of the interface degrees of freedom

We have to note that, for the vibro-acoustic system, in (5.103a), $\mathbf{M}$ and $\mathbf{K}$ are constructed with the nonsymmetric component matrices, $\mathbf{M}_c$ and $\mathbf{K}_c$ presented in (5.17a) and (5.17b), respectively. Namely,

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_c^{(1)} & & \\ & \ddots & \\ & & \mathbf{M}_c^{(k)} \end{bmatrix}, \qquad \mathbf{K} = \begin{bmatrix} \mathbf{K}_c^{(1)} & & \\ & \ddots & \\ & & \mathbf{K}_c^{(k)} \end{bmatrix} \tag{5.105}$$

Primal assembly of τ-symmetric reduced matrices

Having introduced the transformation to $\tilde{\boldsymbol{q}}$, the CB transformation is applied on these transformed set of degrees of freedom. Namely,

$$\tilde{\boldsymbol{q}} = \mathbf{R}_{cb}\tilde{\boldsymbol{\eta}} \Longleftrightarrow \begin{bmatrix} \tilde{\boldsymbol{q}}^{(1)} \\ \vdots \\ \tilde{\boldsymbol{q}}^{(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{cb}^{(1)} & & \\ & \ddots & \\ & & \mathbf{R}_{cb}^{(k)} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\eta}}^{(1)} \\ \vdots \\ \tilde{\boldsymbol{\eta}}^{(k)} \end{bmatrix} \tag{5.106}$$

Substituting (5.106) in (5.103a) and premultiplying the equation by $\mathbf{R}_{cb}^{\mathrm{T}}$ results in

$$\mathbf{m}_m\ddot{\tilde{\boldsymbol{\eta}}} + \mathbf{k}_m\tilde{\boldsymbol{\eta}} = \boldsymbol{f}_m + \boldsymbol{g}_m \tag{5.107}$$

where

$$\boldsymbol{f}_m = \mathbf{R}_{cb}^{\mathrm{T}}\boldsymbol{f}, \tag{5.108a}$$

$$\boldsymbol{g}_m = \mathbf{R}_{cb}^{\mathrm{T}}\boldsymbol{g} \tag{5.108b}$$

$$\mathbf{m}_m = \mathbf{R}_{cb}^{\mathrm{T}}\mathbf{M}\tau\mathbf{R}_{cb} \equiv \begin{bmatrix} \mathbf{m}^{(1)} & & \\ & \ddots & \\ & & \mathbf{m}^{(k)} \end{bmatrix} \tag{5.108c}$$

$$\mathbf{k}_m = \mathbf{R}_{cb}^{\mathrm{T}}\mathbf{K}\tau\mathbf{R}_{cb} \equiv \begin{bmatrix} \mathbf{k}^{(1)} & & \\ & \ddots & \\ & & \mathbf{k}^{(k)} \end{bmatrix} \tag{5.108d}$$

It is important to note that, while writing (5.110a), we enforce the condition that $\boldsymbol{R}_{cb}^{\mathrm{T}}\boldsymbol{r} = \mathbf{0}$.

(5.110a) represents the system with the *decoupled* reduced component matrices as shown in (5.108c) and (5.108d). $\boldsymbol{f}_m$ and $\boldsymbol{g}_m$ represent the modal external and internal forces, respectively. These decoupled reduced component matrices have to be coupled to end up with the reduced system level matrices of the problem in the primal assembly framework.

Substituting (5.106) in (5.103b) and arranging, the modified compatibility relation reads as

$$\mathbf{B}\tau\mathbf{R}_{cb}\tilde{\boldsymbol{\eta}} = \mathbf{0} \equiv \mathbf{B}_{m,\tau}\tilde{\boldsymbol{\eta}} = \mathbf{0}, \qquad \text{with} \quad \mathbf{B}_{m,\tau} = \mathbf{B}\tau\mathbf{R}_{cb} \tag{5.109}$$

To conclude, with these transformations, we can write the modal τ-reduced system equations as follows,

$$\mathbf{m}_m\ddot{\tilde{\boldsymbol{\eta}}} + \mathbf{k}_m\tilde{\boldsymbol{\eta}} = \boldsymbol{f}_m + \boldsymbol{g}_m \tag{5.110a}$$

$$\mathbf{B}\tau\mathbf{R}_{cb}\tilde{\boldsymbol{\eta}} = \mathbf{0} \tag{5.110b}$$

$$\mathbf{L}_m^{\mathrm{T}}\boldsymbol{g}_m = \mathbf{0}. \tag{5.110c}$$

To assemble the components in a primal way over these generalized coordinates, we have to follow the same path that was used in Section 5.4.1. Namely, a *Localization matrix* must be constructed. This *Localization matrix* can be written as follows,

$$\tilde{\boldsymbol{\eta}} = \tilde{\mathbf{L}}_m\tilde{\boldsymbol{\eta}}^g \tag{5.111}$$

Using (5.111) in (5.110b) reads as,

$$\mathbf{B}\tau\mathbf{R}_{cb}\tilde{\mathbf{L}}_m\tilde{\boldsymbol{\eta}}^g = \mathbf{0} \tag{5.112}$$

resulting in $\tilde{\mathbf{L}}_m = \mathrm{null}(\mathbf{B}\tau\mathbf{R}_{cb})$. Since we enforced symmetry with a transformation matrix in (5.112), the null space, $\tilde{\mathbf{L}}_m$ is not only computed with $\mathbf{B}\mathbf{R}_{cb}$ anymore as outlined in Section 5.4.1. Namely, it has to be computed for the generalized coordinates in the transformed coordinate space by using the modified compatibility condition.

By using (5.111), we can write (5.110a) in the following form, namely,

$$\mathbf{m}_m\tilde{\mathbf{L}}_m\ddot{\tilde{\boldsymbol{\eta}}}^g + \mathbf{k}_m\tilde{\mathbf{L}}_m\tilde{\boldsymbol{\eta}}^g = \boldsymbol{f}_m + \boldsymbol{g}_m \tag{5.113}$$

However, as stated originally for the primal assembly formulation in Section 5.4.1, we can not directly premultiply (5.113) by $\tilde{\mathbf{L}}_m^{\mathrm{T}}$ since the interface forces will not cancel with this practice, namely, $\tilde{\mathbf{L}}_m^{\mathrm{T}}\boldsymbol{g}_m \neq \mathbf{0}$. And, therefore, it would not be possible to connect the component matrices without violating the force compatibility requirement.

We can prove this by using the definition of the modal interface force vector provided for the dual assembly case, (5.98). Namely,

$$\tilde{\mathbf{L}}_m^{\mathrm{T}}\boldsymbol{g}_m \equiv -\tilde{\mathbf{L}}_m^{\mathrm{T}}\mathbf{B}_m^{\mathrm{T}}\boldsymbol{\lambda} \neq \mathbf{0} \tag{5.114}$$

since, as outlined above, in (5.112), $\tilde{\mathbf{L}}_m$ spans the null space of $\mathbf{B}\tau\mathbf{R}_{cb}$ but not $\mathbf{B}_m$. Eventually, the correct null space which facilitates the assembly of the internal forces, which, as a result, cancels them is the null space of $\mathbf{B}_m$, namely, $\mathbf{L}_m$. This is a logical result since the interface forces did not change with a transformation performed with τ. Besides, the compatibility of interface forces/sources are still represented with the modal compatibility of the original degrees of freedom of the problem.

To summarize and conclude, to correctly assemble the τ-symmetric reduced component matrices in a primal assembly framework, we need to use different left and right null spaces, namely, $\mathbf{L}_m$ and $\tilde{\mathbf{L}}_m$, respectively. The right null space originates from the modified compatibility relations of the modified generalized coordinates. In contrast, the left null space originates from the modal compatibility relations between the interface forces/sources that were not modified.

Eventually, premultiplying (5.113) by $\mathbf{L}_m^{\mathrm{T}}$ results in the primal assembled system resulting in the reduced global system equations, namely,

$$\mathbf{m}^g \ddot{\tilde{\boldsymbol{\eta}}}^g + \mathbf{k}^g \tilde{\boldsymbol{\eta}}^g = \boldsymbol{f}^g \tag{5.115}$$

where

$$\mathbf{m}^g = \mathbf{L}_m^{\mathrm{T}} \mathbf{R}_{cb}^{\mathrm{T}} \mathbf{M} \tau \mathbf{R}_{cb} \tilde{\mathbf{L}}_m \tag{5.116a}$$

$$\mathbf{k}^g = \mathbf{L}_m^{\mathrm{T}} \mathbf{R}_{cb}^{\mathrm{T}} \mathbf{K} \tau \mathbf{R}_{cb} \tilde{\mathbf{L}}_m \tag{5.116b}$$

$$\boldsymbol{f}^g = \mathbf{L}_m^{\mathrm{T}} \mathbf{R}_{cb}^{\mathrm{T}} \boldsymbol{f} \tag{5.116c}$$

It is also important to note that the interface forces/fluxes vanish by the premultiplication with $\mathbf{L}_m^{\mathrm{T}}$, namely, $\mathbf{L}_m^{\mathrm{T}} \mathbf{B}_m^{\mathrm{T}} \boldsymbol{\lambda} = \mathbf{0}$.

After performing the primal assembly, it is important to note that, although, $\mathbf{m}_m$ and $\mathbf{k}_m$ are symmetric matrices, the global assembled matrices $\mathbf{m}^g$ and $\mathbf{k}^g$ are nonsymmetric reduced system matrices which are assembled in the primal way. Here, we end up with nonsymmetric matrices because of the multiplications with different left and right null space matrices, $\mathbf{L}_m$ and $\tilde{\mathbf{L}}_m$[13]. In addition, these matrices are going to be fully populated.

Dual assembly of τ-symmetric reduced matrices

While performing the assembly in a dual way, the modified degrees of freedom on the interfaces or the modified forces acting on the interface degrees of freedom exist for each component independently. So there is no unification on the degrees of freedom. Moreover, the interface forces are also not eliminated but appear as extra degrees of freedom. In contrast to primal assembly, we have to emphasize the following points for dually assembled problems, namely,

- Interface forces are kept as unknowns in the representation so they do not have to cancel during the assembly process. Since they can be kept independently in the dual assembly framework, they can also be modified as given in Section 5.4.2, in case κ is used for the transformation operations,

- Coupled interface degrees are not unique after the assembly of the reduced components so modified interface degrees of freedom could be used directly,

These points might also be considered as analog to coupling of components with *non-matching* meshes on the interfaces where the interface variables do not match explicitly, in a *node to node* sense.

---

[13] Intuitively, we also expect to keep the relations between the off-diagonal coupling blocks on these reduced non-symmetric matrices, but we have not proven it yet. If our intuition is correct, *vibro*-Lanczos can also be used on the reduced problem which might also bring considerable numerical advantage for problems involving large interfaces.

For the vibroacoustic systems considered here, we can assemble two nonsymmetric vibroacoustic components presented in Figure 5.3 in a dual way[14]. Namely,

$$\underbrace{\begin{bmatrix} \mathbf{M}_c^{(1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_c^{(2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{M}_{dual}} \begin{bmatrix} \ddot{\boldsymbol{q}}^{(1)} \\ \ddot{\boldsymbol{q}}^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{K}_c^{(1)} & \mathbf{0} & {\mathbf{B}^{(1)}}^{\mathrm{T}} \\ \mathbf{0} & \mathbf{K}_c^{(2)} & {\mathbf{B}^{(2)}}^{\mathrm{T}} \\ \mathbf{B}^{(1)} & \mathbf{B}^{(2)} & \mathbf{0} \end{bmatrix}}_{\mathbf{K}_{dual}} \begin{bmatrix} \boldsymbol{q}^{(1)} \\ \boldsymbol{q}^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{f}^{(1)} \\ \boldsymbol{f}^{(2)} \\ \mathbf{0} \end{bmatrix}}_{\boldsymbol{f}_{dual}} \tag{5.117}$$

where, in this case, $\mathbf{M}_c^{(s)}$ and $\mathbf{K}_c^{(s)}$ represent the original non-symmetric vibroacoustic component matrices. $\boldsymbol{\lambda}$ represents the intensity of the interface forces for the structural domain and the source terms for the acoustic domains acting on the components. It is important to recall once more that, while writing (5.117), we made use of $\boldsymbol{g}^{(s)} = -{\mathbf{B}^{(s)}}^{\mathrm{T}}\boldsymbol{\lambda}$. As mentioned in Section 5.4.3, $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$ represent the signed Boolean matrices that are used to explicitly write the component coupling on the interfaces.

With the notations and variables introduced in Section 5.3 and thereafter, the reduced system of equations can be explicitly written in a dual framework as

$$\underbrace{\mathbf{R}^{\mathrm{T}}\mathbf{M}_{dual}\tau_{dual}\mathbf{R}}_{\mathbf{m}_{dual}}\ddot{\tilde{\boldsymbol{\eta}}} + \underbrace{\mathbf{R}^{\mathrm{T}}\mathbf{K}_{dual}\tau_{dual}\mathbf{R}}_{\mathbf{k}_{dual}}\tilde{\boldsymbol{\eta}} = \underbrace{\mathbf{R}^{\mathrm{T}}\boldsymbol{f}}_{\boldsymbol{f}_{dual}} \tag{5.118}$$

where τ defined in (5.104) must be updated to conform with the sizes of the matrices used in the dual representation, namely,

$$\tau_{dual} = \begin{bmatrix} \tau^{(1)} & & \\ & \tau^{(2)} & \\ & & \mathbf{I} \end{bmatrix} \tag{5.119}$$

The transformation on the dual coordinates can also be written in a similar fashion to the previous discussions, namely,

$$\boldsymbol{q}_{dual} = \mathbf{R}_{dual}\tilde{\boldsymbol{\eta}}_{dual} \longrightarrow \begin{bmatrix} \tilde{\boldsymbol{q}}^{(1)} \\ \tilde{\boldsymbol{q}}^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{cb}^{(1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{cb}^{(2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\eta}}^{(1)} \\ \tilde{\boldsymbol{\eta}}^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} \tag{5.120a}$$

With the outlined analysis, reduced matrices $\mathbf{m}_{dual}$ and $\mathbf{k}_{dual}$ is written for the dual representation as follows,

$$\mathbf{k}_{dual} = \begin{bmatrix} \mathbf{k}^{(1)} & \mathbf{0} & {\mathbf{B}_m^{(1)}}^{\mathrm{T}} \\ \mathbf{0} & \mathbf{k}^{(2)} & {\mathbf{B}_m^{(2)}}^{\mathrm{T}} \\ \mathbf{B}_{m,\tau}^{(1)} & \mathbf{B}_{m,\tau}^{(2)} & \mathbf{0} \end{bmatrix} \tag{5.121a}$$

$$\mathbf{m}_{dual} = \begin{bmatrix} \mathbf{m}^{(1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{m}^{(2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{5.121b}$$

---

[14] Selection of two substructures is convenient for mathematical representations but the presented theory can be extended to systems having more components without any difficulties.

The reduced component matrices $\mathbf{k}^{(k)}$ and $\mathbf{m}^{(k)}$ in (5.121a) and (5.121b) were defined in (5.40) and (5.60), respectively. By recalling the definition of the modified compatibility relation from (5.109), the Boolean matrices for the modal coordinates are written as follows:

$$\mathbf{B}_m^{(k)\,\mathrm{T}} = \mathbf{R}_{cb}^{(k)\,\mathrm{T}} \mathbf{B}^{(k)\,\mathrm{T}} \tag{5.122a}$$

$$\mathbf{B}_{m,\tau}^{(k)} = \mathbf{B}^{(k)} \mathcal{T}^{(k)} \mathbf{R}_{cb}^{(k)} \tag{5.122b}$$

It is important to note some points on the dually assembled system given in (5.121a)-(5.121b). Looking at (5.121a) and (5.121b), we can observe that the upper left blocks of these reduced matrices keep their symmetric nature in the dual assembly framework because of the symmetric reduced matrices building these blocks. This brings some advantages for the solution techniques to be used with dual formulations. However, because of the duality of the interface degrees of freedom kept for the reduced matrices, the size of the *dual-assembled* reduced problem is larger than its *primal-assembled* counterpart.

Relations between the left and right null spaces from a dual perspective

There is a close relation between the primal and dual assembly options presented in Section 5.4.3. Once more, by explicitly writing the *dual-assembled* reduced problem for two vibroacoustic components, we can clarify these relations. Namely,

$$\begin{bmatrix} \mathbf{m}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\tilde{\boldsymbol{\eta}}} \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \mathbf{k}_m & \mathbf{B}_m^{\mathrm{T}} \\ \mathbf{B}_{m,\tau} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\eta}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_m \\ \mathbf{0} \end{bmatrix} \tag{5.123}$$

with

$$\mathbf{m}_m = \begin{bmatrix} \mathbf{m}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{m}^{(2)} \end{bmatrix} \tag{5.124a}$$

$$\mathbf{k}_m = \begin{bmatrix} \mathbf{k}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{k}^{(2)} \end{bmatrix} \tag{5.124b}$$

$$\mathbf{B}_m^{\mathrm{T}} = \begin{bmatrix} \mathbf{B}_m^{(1)\,\mathrm{T}} \\ \mathbf{B}_m^{(2)\,\mathrm{T}} \end{bmatrix} \tag{5.124c}$$

$$\mathbf{B}_{m,\tau} = \begin{bmatrix} \mathbf{B}_{m,\tau}^{(1)} & \mathbf{B}_{m,\tau}^{(2)} \end{bmatrix} \tag{5.124d}$$

We can separate the block rows of (5.123) and rewrite them as

$$\mathbf{m}_m \ddot{\tilde{\boldsymbol{\eta}}} + \mathbf{k}_m \tilde{\boldsymbol{\eta}} = \boldsymbol{f}_m - \mathbf{B}_m^{\mathrm{T}} \boldsymbol{\lambda} \tag{5.125a}$$

$$\mathbf{B}_{m,\tau} \tilde{\boldsymbol{\eta}} = \mathbf{0} \tag{5.125b}$$

We can think (5.125a) and (5.125b) as the dual counterpart of (5.110a) and (5.110b), respectively. From Section 5.4.3, we know that to perform a primal assembly, we need to transform the degrees of freedom of the reduced components into a unique set and also eliminate the modally reduced interface forces. Applying these requirements on (5.125a) and (5.125b), we can shift from a dually assembled problem to a primal assembled system by

- Finding the null space, $\tilde{\mathbf{L}}_{\mathbf{m}}$, of $\mathbf{B}_{m,\tau}$, namely, $\tilde{\mathbf{L}}_m = \text{null}(\mathbf{B}_{m,\tau})$, which performs the transformation, $\tilde{\boldsymbol{\eta}} = \tilde{\mathbf{L}}_m \tilde{\boldsymbol{\eta}}^g$, and substituting this transformation into (5.125a)

- Premultiplying (5.125a) with $\mathbf{L}_m^{\mathrm{T}}$ which is the null space of $\mathbf{B}_m^{\mathrm{T}}$, namely, $\mathbf{L}_m^{\mathrm{T}} = \text{null}(\mathbf{B}^{\mathrm{T}})$

Eventually, we can conclude that we can convert the dual representation in (5.123) to a primal assembled system. This shows that the primal and dual assembly approaches are equivalent for the τ-symmetric system representation.

5.5. Some notes on implementation and analysis of assembled system

Having assembled the reduced component matrices, we have to perform the analysis on this assembled reduced problem and recover the results back to the physical domain. The details of these important steps are summarized below, namely,

- **Information reuse for computing $\mathbf{B}_{m,\tau}$**, definition of $\mathbf{B}_{m,\tau}$ was provided in (5.109). We can also write $\mathbf{B}_{m,\tau}$ explicitly as follows:

$$\mathbf{B}_{m,\tau} = \begin{bmatrix} \mathbf{B}^{(1)}\tau^{(1)}\mathbf{R}_{cb}^{(1)} & \mathbf{B}^{(2)}\tau^{(2)}\mathbf{R}_{cb}^{(2)} & \dots & \mathbf{B}^{(k)}\tau^{(k)}\mathbf{R}_{cb}^{(k)} \end{bmatrix} \tag{5.126}$$

  which is generalized for k components. Returning back to Section 5.3.4 for the calculation of the reduced component matrices, we can observe that, to compute the reduced symmetric matrices, we need to compute $\tau^{(k)}\mathbf{R}_{cb}^{(k)}$. It is interesting to note that these expressions also appear in (5.126). Therefore, we can store some information from previous steps and make use of that information to construct the blocks in (5.126). It is once more instructive to recall that $\mathbf{B}_{m,\tau}$ is necessary for the primal assembly process to compute $\tilde{\mathbf{L}}_m$. Having pointed out this detail, we are mainly concerned with the calculations performed for the mass matrix. Briefly, $\mathbf{V}$ defined in (5.56) is equivalent to $\tau^{(k)}\mathbf{R}_{cb}^{(k)}$ for the component under consideration. To compute the individual blocks in (5.126), $\mathbf{V}$ must also be multiplied from left with $\mathbf{B}^{(k)}$. It is important to recall that to perform the operations in a CB-scheme, we order our matrices in predefined order. Namely, the Boolean compatibility matrices were defined in (5.72) and we repeat that equation here for convenience, namely,

$$\mathbf{B}^{(k)} = \begin{bmatrix} \mathbf{B}_b^{(k)} & \mathbf{0} \end{bmatrix}$$

  Depending on the coupling on the interfaces, $\mathbf{B}^{(k)}$ can either become $\mathbf{I}$ or $-\mathbf{I}$ or include both if the component has more than one interface. Moreover, the block $\mathbf{0}$ was defined for the internal degrees of freedom. If we carefully investigate $\mathbf{V}$ outlined in (5.56) once more, we notice that, for the operations performed there, it was found in the $\boldsymbol{u}-\boldsymbol{p}$ order of the component. To compute the blocks in (5.126), we have to reorder the rows of $\mathbf{V}$ back to CB-order by using the mapping defined for the component as outlined in Section 5.3.4. By using this convention and after reordering $\mathbf{V}$, one of the blocks in (5.126) can be explicitly written as follows,

$$\mathbf{B}^{(k)}\tau^{(k)}\mathbf{R}^{k}_{cb} = \begin{bmatrix}\mathbf{B}^{(k)}_{b} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{V}^{(k)}_{b}\\ \mathbf{V}^{(k)}_{i}\end{bmatrix} \equiv \mathbf{B}^{(k)}_{b}\mathbf{V}^{(k)}_{b} \tag{5.127}$$

To conclude, we can store $\mathbf{V}$ which was built during the computation of the reduced mass matrix and use it for building the blocks in (5.126) without extra numerical costs except reordering its rows.

- **Computation of the null space, $\mathbf{L}_m$ and $\tilde{\mathbf{L}}_m$**, as we outlined before we have to compute the left and right null spaces, $\mathbf{L}_m$ and $\tilde{\mathbf{L}}_m$ to assemble the problem in a primal way and to compute these null spaces we use the QR decomposition [62] as outlined in [27, 102]. It has been found out there that this practice does not affect the condition number of the matrices involved in the numerical operations. It is also important to note that for simple problems, the null space can also be found by investigating the compatibility matrix. In this practice, one needs to construct a null space matrix so that its multiplication with the Boolean compatibility matrix results in a matrix filled with zeros. However, by using QR decomposition, optimal conditioning is achieved and the process is automated.

- **Reduced system solution and recovery**, after the assembly of the global reduced system matrices presented in (5.115), we have to solve the undamped eigenvalue problem to compute the approximate eigenvalues and eigenvectors. Namely, with

$$\left(\mathbf{k}^{g} - \lambda^{g}\mathbf{m}^{g}\right)\tilde{\boldsymbol{\eta}}^{g} = \mathbf{0} \tag{5.128}$$

It is important to recall that $\mathbf{k}^g$ and $\mathbf{m}^g$ are nonsymmetric reduced system matrices. λ^g represents the approximate eigenvalues found by using the reduced problem.
To obtain the physical mode shape approximations, we have to recover/expand the reduced eigenvector approximations, $\tilde{\boldsymbol{\eta}}^g$, back to full domain. Since, we used a transformation matrix, τ, we perform this recovery as follows:

$$\boldsymbol{q} = \tau\mathbf{R}_{cb}\tilde{\mathbf{L}}_m\tilde{\boldsymbol{\eta}}^{g} \tag{5.129}$$

In (5.129), each component is recovered independently where the resulting vector $\boldsymbol{q}$ reads as,

$$\boldsymbol{q} = \begin{bmatrix}\boldsymbol{q}^{(1)}\\ \boldsymbol{q}^{(2)}\\ \vdots\\ \boldsymbol{q}^{(k)}\end{bmatrix}, \qquad \text{with} \quad \boldsymbol{q}^{(k)} = \begin{bmatrix}\boldsymbol{u}^{(k)}\\ \boldsymbol{p}^{(k)}\end{bmatrix} \tag{5.130}$$

As a last step, the physical vector approximations of each component, $\boldsymbol{q}^{(k)}$, are mapped back onto the global system vector. Following the analysis presented in Appendix E, we can conclude that $\boldsymbol{q}$ is an approximation to the right eigenvectors of the original nonsymmetric problem.

- **Information reuse for vector recovery**: as it was outlined above for the computation of $\mathbf{B}_{m,\tau}$, a similar kind of information reuse is also possible while performing the recovery of the vectors back to the physical domain over a component model. To explain this step, we have to use (5.129). Namely, having solved the reduced eigenvalue problem in (5.128), we can use (5.111) to find the block of generalized coordinates as defined in (5.106), namely,

$$\tilde{\boldsymbol{\eta}} = \tilde{\mathbf{L}}_m \tilde{\boldsymbol{\eta}}^g \tag{5.131}$$

To go back to physical degrees of freedom then we have use (5.129), namely, $\boldsymbol{q} = (\tau \mathbf{R}_{cb})\, \tilde{\boldsymbol{\eta}}$, and use $\tau \mathbf{R}_{cb}$ that was stored using the computations of the mass matrix. Same practice was also employed in (5.127).

As a result, we reused the same information at two points. Namely, the first point is to compute the blocks in (5.126) and the second point is to recover the approximate physical domain vectors on the component level.

5.6. Numerical examples

In this section, we work with a two-dimensional example problem to assess the developed reduction technique. We are going to present two versions of the same model. Namely, we will use an academic model with a moderate size. Subsequently, we are going to show the results on the same model with a finer mesh structure resulting in more degrees of freedom. The two-dimensional domain is divided into two components and the component level reductions were performed according to the theory presented in Section 5.3. Having performed the reduction at the component level, the reduced component matrices are assembled by using the primal assembly outlined in Section 5.4.3 for τ-symmetric system representation. As also outlined in Section 5.4.3, primal assembly results in *nonsymmetric* reduced global nonsymmetric system matrices. The major interest here is on the accuracy of the computed eigenvalues and the eigenvectors since these are of major importance in computing the dynamic response of the coupled problems and structures.

The model considered is a two-dimensional cavity model that is coupled to a structural beam domain. The model was constructed in ANSYS by using the implemented fluid-structure interface. System matrices, node and element tables and the definition of the degrees of freedom are extracted with the developed MATLAB/C++ interface. The rest of the calculations were performed in MATLAB. The information on the interface nodes between coupling components, which are necessary for partitioning and correct coupling on the interfaces, were also extracted from ANSYS in raw text file format.

For this case, the structural part was modelled with beam elements, namely BEAM3 in ANSYS. BEAM3 elements have two translational and one rotational degrees of freedom. The beam that is coupled to the fluid domain is fully constrained

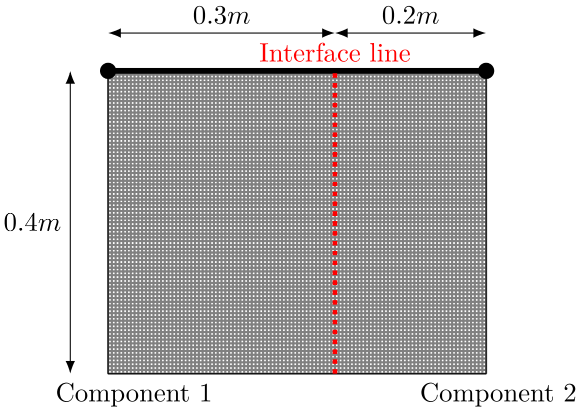


Figure 5.4: 2D vibro-acoustic problem, beam(in bold) coupled to 2D rectangular fluid domain, filled circles represent the fixed boundary conditions on the structure.

at the two ends. It must be noted that, for the composition of the problem we deal here, the inversion of $\mathbf{K}_s$ included in the τ component transformation matrix is not a problem due to the applied constraints on the structural side. The fluid domain was modelled with quadrilateral 2D acoustic fluid elements, namely, FLUID29, which use pressure and the two translations in plane as the main degrees of freedom. The interface coupling information between structural and acoustic elements is implemented in FLUID29 element formulation in ANSYS. Apart from the coupling interface with the beam, the rest of the walls of the model are considered to be rigid walls. The model and the associated dimensions are provided in Figure 5.4. It is important to note once more that the mesh used on the corresponding components exactly match over the interface region. For the numerical experiments conducted, we selected water as the fluid that fills the acoustic cavity. The related properties of water is taken as follows: density, 1000 kg/m^3, and speed of sound, 1500 m/s. The properties of the structural beam is given as follows: density, 2700 kg/m^3, Young's modulus, 71 GPa, and Poisson ratio 0.33, beam height, 5 mm and beam width, 50 mm.

To show the accuracy of the developed technique on the reduced system level, we defined two performance estimators. The first one is on the calculated reduced frequencies and the other one is on the recovered mode shape vectors. Namely,

- The error on the frequencies are calculated with respect to the frequencies calculated by ANSYS, namely,

$$\epsilon_\omega = \frac{f_{red} - f_{ANSYS}}{f_{ANSYS}} \tag{5.132}$$

  where f_{red} represents the reduced frequencies extracted with (5.128) and f_{ANSYS} represents the frequencies calculated with ANSYS.

- The error on the mode shape vectors are calculated with an error measure by using the MAC between the recovered vectors and the mode shape vectors computed with ANSYS. This error expression read as,

$$\epsilon_{\phi} = 1 - \text{MAC} \equiv= 1 - \frac{|\phi_{rec}^{\mathrm{T}} \phi_{ANSYS}|^2}{(\phi_{rec}^{\mathrm{T}} \phi_{rec})(\phi_{ANSYS}^{\mathrm{T}} \phi_{ANSYS})} \tag{5.133}$$

where ϕ_{rec} represents the physically recovered vectors obtained with (5.129).

5.6.1. Test case 1: academic model

For the first test case, the model shown in Figure 5.4 is meshed with an element size of 0.01 m resulting in 2238 coupled degrees of freedom in total. For this model, the fluid domain has 2091 degrees of freedom which is associated with 147 structural degrees of freedom. The number of interface degrees of freedom resulting from the red interface line in Figure 5.4 is 44 for this test case. The overview on the degrees of freedom of the components is provided in Table 5.1 for different fixed interface mode numbers, m. The reduction ratios with respect to the full component sizes are plotted in Figure 5.5.

		Reduced component matrix sizes		
Component	Full size	$m = 10$	$m = 20$	$m = 30$
1	1391	54	64	74
2	921	54	64	74

Table 5.1: Information on component level degrees of freedom

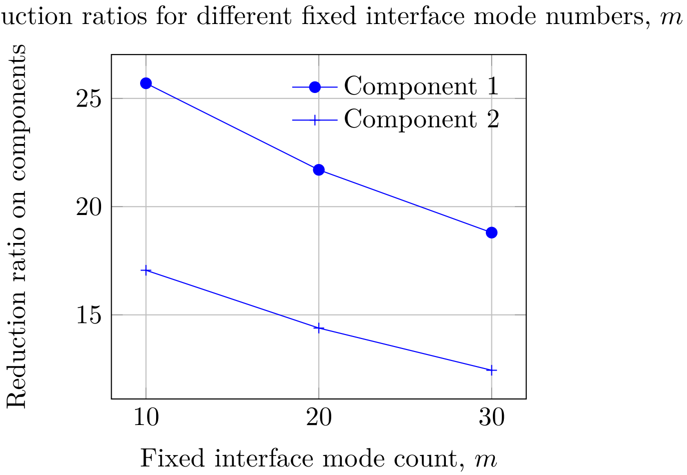


Figure 5.5: Matrix size reduction on components

The results calculated with the error estimators in (5.132) and in (5.133) are plotted in Figures 5.6 and 5.7, respectively. It can be observed from the error plots

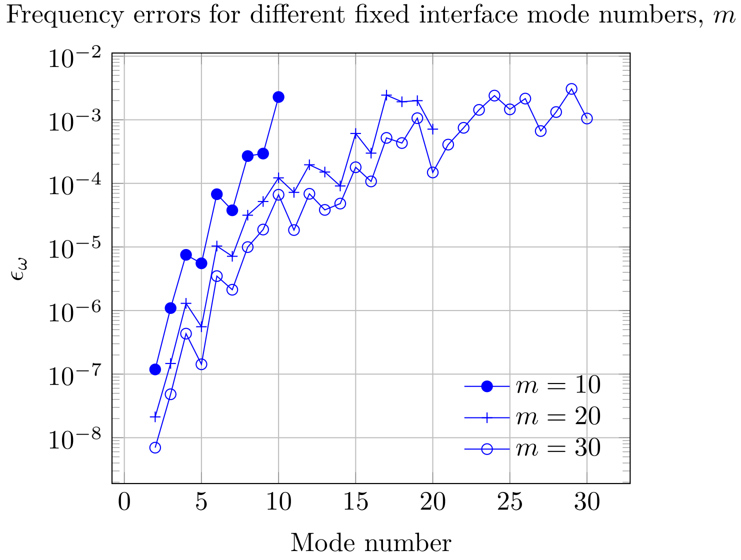


Figure 5.6: Errors on frequencies with respect to ANSYS frequencies

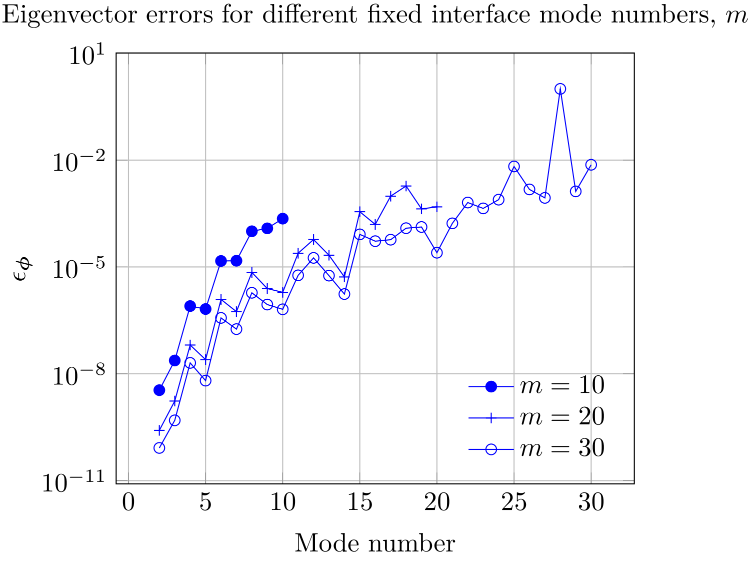


Figure 5.7: Errors on mode shape vectors with respect to ANSYS mode shapes, 1-MAC

that the accuracy of the developed reduction technique is quite high for the selected different fixed interface mode counts.

5.6.2. Test case 2: *pseudo* industrial-sized model

In this section, we show the performance of the technique on a larger version of the same model by decreasing the mesh size significantly. The mesh size used for this model is 0.001 m resulting in 202398 degrees of freedom. The fluid side of the model consists of 200901 degrees of freedom whereas the structural side of the model consists of 1497 degrees of freedom. It is important to note that there is a big difference in the number of degrees of freedom in the fluid and structural domains. Namely, with the pursued two dimensional modelling, the fluid domain is a surface extending over a large area in comparison to the structural domain, which is only modelled as a line for this case.

The interface that divides the full model into components as shown in Figure 5.4 results in 404 interface degrees of freedom. These interface degrees of freedom are kept on the reduced components for component coupling similar to the analysis performed in Section 5.6.1. The overview on the degrees of freedom resulting from the reduction performed is shown in Table 5.2. It is important to note that the reduced components matrix sizes are calculated with the addition of internal mode counts to the *kept interface* degrees of freedom, namely, as $404 + m$, in Table 5.2.

		Reduced component matrix sizes		
Component	Full size	$m = 10$	$m = 20$	$m = 30$
1	121601	414	424	434
2	81201	414	424	434

Table 5.2: Information on component level degrees of freedom

The accuracy achieved with the reduced order model is also quite satisfactory on this large model. We can observe the errors on the frequencies and the mode shapes with respect to ANSYS computed counterparts in Figures 5.9 and 5.10, respectively.

5.7. Summary and conclusions

Based on the well-known Craig-Bampton reduction technique, we proposed a new and numerically efficient reduction technique for coupled vibro-acoustic problems. To the author's knowledge, this technique was not found in the literature before. The inherent nonsymmetry and the ill-conditioned nature of the coupled matrices resulting from the discretization process makes these problems very challenging to solve. Industrial problems in this field require a great deal of computational resources. Moreover, especially if one needs to perform frequency sweeps for these coupled problems to investigate the effect of different parameters, this cost can become quite significant. Frequently, local changes and optimization tests can be applied in some part of the design domain. But, this practice eventually requires a

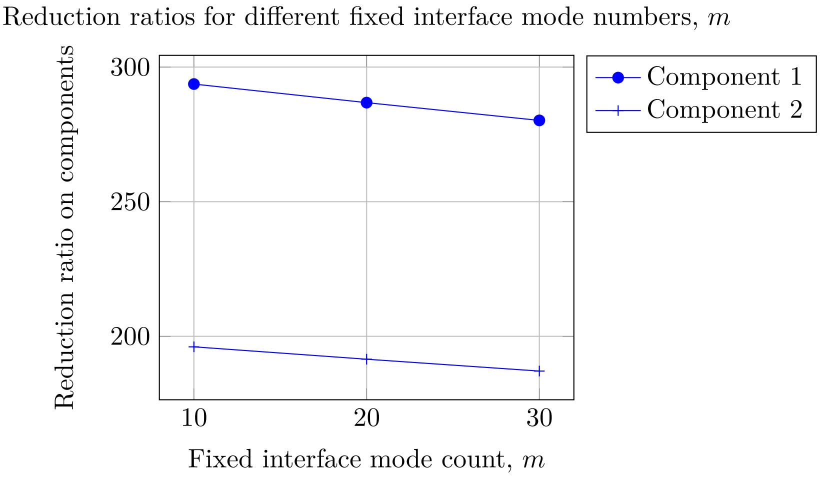


Figure 5.8: Matrix size reduction on components

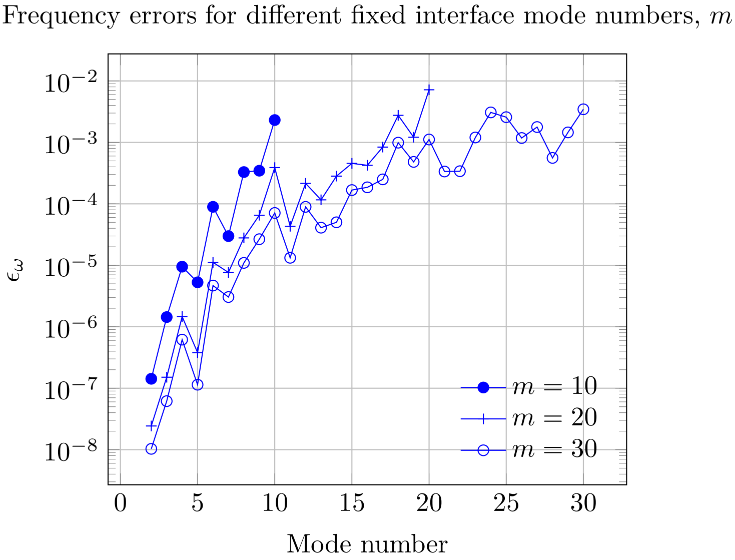


Figure 5.9: Errors on frequencies with respect to ANSYS frequencies

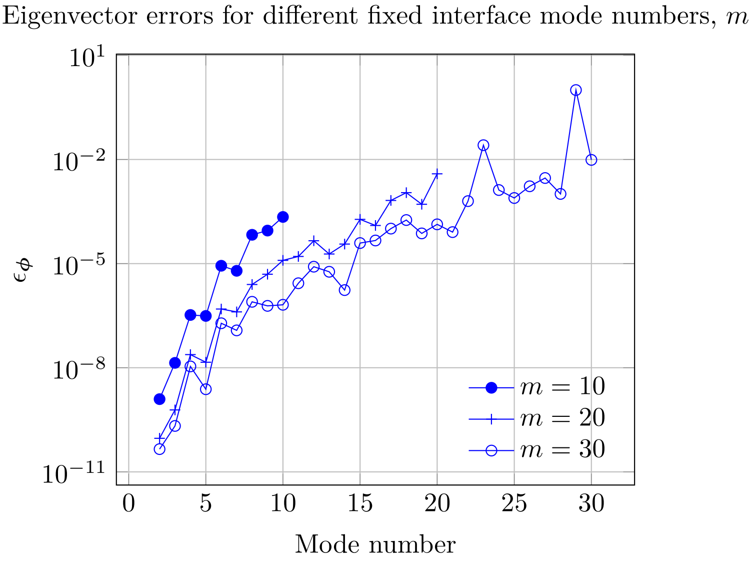


Figure 5.10: Errors on mode shape vectors with respect to ANSYS mode shapes, 1-MAC

new eigenvalue analysis and has to be followed by a new frequency sweep to investigate the effects and/or sensitivity of these changes on the important performance parameters.

The proposed technique offers a new way in reducing the complexity and the computational cost for the above mentioned eigenvalue and frequency sweep analysis. Namely, by partitioning the complete analysis domain into components, one can come up with substantial savings. Besides, different groups can work on different parts of the vibro-acoustic design domain independently and make changes on these component models without affecting the other groups. Eventually, if a design change is performed on some of the components, only these components has to be re-analyzed and reduced with the outlined technique. Subsequently, all the reduced matrices of the components are coupled and the reduced problem is solved for the approximation of its global dynamics with a fraction of the computational resources.

Several advantages of the proposed technique can be summarized as follows, namely,

- With proper numerical arrangements on matrix-vector operations, the ingredients of the Craig-Bampton reduction basis can be computed with a symmetric Lanczos type solver efficiently.
- The use of the symmetric transformation matrix automatically results in proper left and right projection spaces for a Craig & Bampton (CB) type reduction. One way to interpret this is that the symmetric matrices are reduced by using the same trial and test bases[15]. The other interpretation is

---

[15] For instance, in a Galerkin type projection.

that, for the original non-symmetric component matrices, this practice results in different but consistent test and trial bases due to the transformation matrix used[16],

- The symmetric reduced matrices computed with a CB type reduction have some nice sparsity patterns. The technique proposed here also results in symmetric component matrices in the reduced format. And, therefore, the reduced matrices computed with the proposed technique also have these nicely structured reduced matrices as a result of the operations performed on symmetric systems,
- Although, on the component level, we can perform the operations with a symmetric approach and end up with reduced symmetric matrices, it is shown that the scaling operation performed results in a change in the compatibility conditions. This change in the compatibility of interface variables necessitates the use of different left and right null spaces to couple the component matrices in a primal assembly framework as outlined with details. As a result, the assembled global reduced matrices of the system are still nonsymmetric due to the use of different left and right null spaces,
- Eventually, it is also shown that the major dynamic system parameters, namely, the eigenvalues and eigenvectors can be calculated very accurately with the proposed technique.
- Additionally, comparing the technique presented in this chapter with the ones presented in [75, 102], we can list the following advantages:
  - We do not omit any terms in the static representation and guarantee a statically complete representation.
  - Fixed interface modes are computed more accurately.
  - In the above references, they use a decoupled block diagonal projection space for reduction. Here, we perform a more natural projection by the using the transformation matrix which is used to obtain symmetric component matrices.

Therefore, we can conclude that the technique proposed in this chapter can efficiently be used to perform component mode synthesis investigations for problems involving vibro-acoustic coupling.

---

[16] Interpreted as a Petrov-Galerkin type projection.

6

Extending *vibro*-LsCB for Systems with Singularities

In Chapter 5, we presented a new Craig-Bampton type reduction technique for structural-acoustic problems. In this chapter, we provide details for its extension and generalization to systems that have singularities, namely, for components without constraints. In combination with Chapter 5, the contents of this chapter is also a part of the same journal manuscript under preparation.

6.1. Introduction

Up to this point, we always assumed that $\mathbf{K}_s$ is a regular invertible matrix. However, in a more general vibro-acoustic system, some of the components might have no constraints applied on them so that the structural stiffness matrix $\mathbf{K}_s$ becomes singular. In this case, the transformation matrix τ can not be used directly for creating symmetric component matrices. One possible example of this scenario is depicted in Figure 6.1 where a two dimensional domain is divided into 3 components. The bold lines in the figure represent the structural domains which are coupled to the associated fluid domains. In Figure 6.1, we can see the constrained nodes on components 1 and 3 but there are no constraints on component 2. Recalling the transformation matrix, τ, namely,

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{K}_s^{-1}\mathbf{M}_s & \mathbf{K}_s^{-1}\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\tau} \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \boldsymbol{p} \end{bmatrix} \tag{6.1}$$

we can clearly realize that there exists a singularity problem for component 2. Namely, we are not able to use τ transformation matrix directly due to the singular

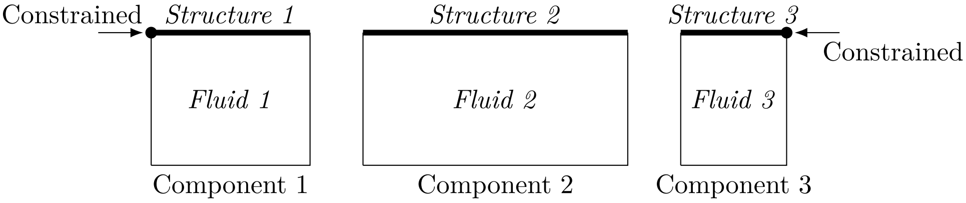


Figure 6.1: Structural-acoustic coupled domain with 3 components, free-free middle component

nature of $\mathbf{K}_s$ since $\mathbf{K}_s^{-1}$ does not exist mathematically for component 2 depicted in Figure 6.1. In this short chapter, we will outline some algebraic manipulations in order to circumvent this problem and represent the local symmetry in the space of elastic models.

In Chapter 5, in Sections 5.3.2 and 5.3.3, we outlined the options to create symmetric component matrices and the numerical details on performing operations with these matrices. It must be noted that the theory presented in this chapter can also be used to perform the equivalent operations when κ matrix is used for component transformations and when the singularity of $\mathbf{K}_f$ has to be tackled.

6.2. Solution with singularities

Since we are going to deal with the solution of singular systems, we provide a short overview here to represent the components used in the representation. Over the structural part of the problem, the most general solution of a singular linear system of equations

$$\mathbf{K}_s \boldsymbol{x}_s = \boldsymbol{b}_s \tag{6.2}$$

is written as [63][1],

$$\boldsymbol{x}_s = \mathbf{K}_s^+ \mathbf{P}_s^{\mathrm{T}} \boldsymbol{b}_s + \mathbf{R}_s \boldsymbol{\alpha}_s \equiv \boldsymbol{x}_{gen} + \mathbf{R}_s \boldsymbol{\alpha}_s \tag{6.3}$$

where $\mathbf{K}_s$ is the stiffness matrix for the structural domain. In (6.3), $\mathbf{K}_s^+$ represent a generalized inverse obtained for instance as follows [63],

$$\mathbf{K}_s^+ = \begin{bmatrix} \mathbf{K}_{s,11}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{6.4}$$

where $\mathbf{K}_{s,11}$ is a nonsingular matrix whose dimension is equal to the rank of $\mathbf{K}_s$, namely $\mathbf{K}_{s,11}$ is the largest nonsingular submatrix of the operator matrix, $\mathbf{K}_s$.

$\mathbf{P}_s^{\mathrm{T}} \boldsymbol{b}_s$ represents a self-equilibrated load vector. In the case of a singular system of equations, we can only find a solution with respect to the arbitrary rigid body vector components *only and only if* the load vector is self-equilibrated [63].

By using (6.3), we can also write the purely elastic component of the solution vector by making the rigid body mode vectors in $\mathbf{R}_s$ and the general solution vector $\boldsymbol{x}$ $\mathbf{M}_s$-orthogonal where $\mathbf{M}_s$ represents the mass matrix of the structural domain. Namely, if we premultiply (6.3) by $\mathbf{R}_s^{\mathrm{T}} \mathbf{M}_s$, we can write,

$$\mathbf{R}_s^{\mathrm{T}} \mathbf{M}_s \boldsymbol{x}_s = \mathbf{R}_s^{\mathrm{T}} \mathbf{M}_s \mathbf{K}_s^+ \mathbf{P}_s^{\mathrm{T}} \boldsymbol{b}_s + \mathbf{R}_s^{\mathrm{T}} \mathbf{M}_s \boldsymbol{R}_s \alpha_s \tag{6.5}$$

Enforcing $\mathbf{R}_s^{\mathrm{T}} \mathbf{M}_s \boldsymbol{x}_s = 0$, we can see that

$$\alpha_s = -\mathbf{R}_s^{\mathrm{T}} \mathbf{M}_s \mathbf{K}_s^+ \mathbf{P}_s^{\mathrm{T}} \boldsymbol{b}_s \tag{6.6}$$

---

[1] Assuming a solution exists

where we assumed that the rigid body mode vectors are mass-normalized, namely, $\mathbf{R}_s^{\mathrm{T}}\mathbf{M}_s\mathbf{R}_s = \mathbf{I}$. Using the definition in (6.6), we can write the general solution vector in (6.3) only in the space of elastic modes as,

$$\begin{aligned}\boldsymbol{x}_e &= \mathbf{K}_s^{+}\mathbf{P}_s^{\mathrm{T}}\boldsymbol{b}_s + \mathbf{R}_s\left(-\mathbf{R}_s^{\mathrm{T}}\mathbf{M}_s\mathbf{K}_s^{+}\mathbf{P}_s^{\mathrm{T}}\boldsymbol{b}_s\right) && (6.7\text{a})\\ &\equiv \left(\mathbf{I} - \mathbf{R}_s\mathbf{R}_s^{\mathrm{T}}\mathbf{M}_s\right)\mathbf{K}_s^{+}\mathbf{P}_s^{\mathrm{T}}\boldsymbol{b}_s && (6.7\text{b})\\ &\equiv \mathbf{P}_s\mathbf{K}_s^{+}\mathbf{P}_s^{\mathrm{T}}\boldsymbol{b}_s \equiv \mathbf{K}^{\oplus}\boldsymbol{b}_s && (6.7\text{c})\end{aligned}$$

where the projector $\mathbf{P}_s$ is mathematically defined as [49],

$$\mathbf{P}_s = \left(\mathbf{I} - \mathbf{R}_s(\mathbf{R}_s^{\mathrm{T}}\mathbf{M}_s\mathbf{R}_s)^{-1}\mathbf{R}_s^{\mathrm{T}}\mathbf{M}_s\right) \xrightarrow{\mathbf{R}_s^{\mathrm{T}}\mathbf{M}_s\mathbf{R}_s=\mathbf{I}} \mathbf{P}_s = \left(\mathbf{I} - \mathbf{R}_s\mathbf{R}_s^{\mathrm{T}}\mathbf{M}_S\right) \quad (6.8)$$

$\boldsymbol{x}_e$ in (6.7c) does not contain any components in the direction of the vectors found in $\mathbf{R}_s$[2]. In other words, we are explicitly subtracting the components of $\mathbf{R}_s$ to find only the elastic part of the solution vector in the $\mathbf{M}_s$-orthogonal complement[3] of the $\mathbf{R}_s$-space.

In the light of the ongoing discussion, without changing the solution vector written in (6.3), we can represent the solution vector as a sum of two vector components, namely, one in the space of elastic modes and the other in the space of $\mathbf{R}_s$. With this practice, we can write,

$$\begin{aligned}\boldsymbol{x}_s &= \boldsymbol{x}_e + \mathbf{R}_s\tilde{\boldsymbol{\alpha}}_s && (6.9\text{a})\\ &= \mathbf{P}_s\boldsymbol{x}_s + \mathbf{R}_s\tilde{\boldsymbol{\alpha}}_s && (6.9\text{b})\end{aligned}$$

where we can easily verify that $\boldsymbol{x}_e^{\mathrm{T}}\mathbf{M}_s\mathbf{R}_s = \mathbf{0}$ meaning that $\boldsymbol{x}_e$ vector is in the $\mathbf{M}_s$-orthogonal complement of $\mathbf{R}_s$. In comparison to (6.3), the intensity of the $\mathbf{R}_s$ component should be different than that of the solution provided in (6.3) because we filtered out the components of $\mathbf{R}_s$ from the first component of the general solution vector, namely, $\mathbf{K}_s^{+}\mathbf{P}_s^{\mathrm{T}}\boldsymbol{b}_s$. For this case, in order to stress the difference with the general solution provided in (6.3), the intensity of the rigid body mode vector is represented with $\tilde{\boldsymbol{\alpha}}_s$ which is still undetermined.

(6.9b) shows that a solution with respect to the rigid body mode vectors found in $\mathbf{R}_s$ can still be found for symmetric problems as soon as we construct a self-equilibrated load with $\mathbf{P}_s^{\mathrm{T}}\boldsymbol{b}$.

Conceptually, we can visualize the geometric interpretation of the solution vector given in either (6.3) or (6.9b) by using a two-dimensional space. Namely, in Figure 6.2, we present a problem which has only one null space vector which is the green line in Figure 6.2. In other words, the green line spans the null space of the considered operator matrix, $\mathbf{K}_s$, for this conceptual case. The solution vector is represented with the black vector as $\boldsymbol{x}_s$. It is clear from Figure 6.2 that we are representing the resulting solution vector with different components in (6.3) or (6.9b), respectively. And these vector sets are represented with the blue and red vectors for (6.3) and (6.9b), respectively. It is once more important to stress out that the intensity of the null space vector is undetermined in these representations.

---

[2] Here, $\mathbf{M}_s$-orthogonality is assumed in all the orthogonality relations.

[3] Space that is $\mathbf{M}_s$-orthogonal to vectors in $\mathbf{R}_s$.

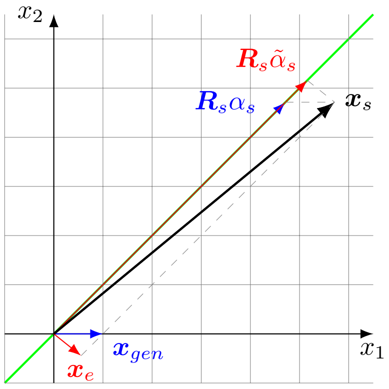


Figure 6.2: Representation of the solution vector with different component vectors

6.3. Representation of symmetric component matrices with singularities

As outlined in Section 6.2, even if singularities exist on the system level, we can still find the components of the solution vectors with respect to the rigid body mode vectors defined in $\mathbf{R}_s$. Following the analysis of Section 6.2, for instance (6.9b), the displacement variable[4] $\tilde{\boldsymbol{u}}$ can be separated into two components, namely,

$$\tilde{\boldsymbol{u}} = \mathbf{P}_s\tilde{\boldsymbol{u}} + \mathbf{R}_s\tilde{\boldsymbol{\alpha}}_s \equiv \tilde{\tilde{\boldsymbol{u}}} + \mathbf{R}_s\tilde{\boldsymbol{\alpha}}_s \tag{6.10}$$

If we consider the degrees of freedom of the system or component, namely, $\tilde{\boldsymbol{u}}$ and $\boldsymbol{p}$, we can write the following transformation on the component level,

$$\begin{bmatrix} \tilde{\boldsymbol{u}} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_s & \mathbf{0} & \mathbf{R}_s \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\tilde{\boldsymbol{u}}} \\ \boldsymbol{p} \\ \tilde{\boldsymbol{\alpha}}_s \end{bmatrix} \leftrightarrow \tilde{\boldsymbol{q}} = \mathbf{T}_s\tilde{\boldsymbol{q}}_t \tag{6.11}$$

where $\tilde{\boldsymbol{q}}_t$ represents the *extended coordinates* of the component under consideration. As outlined above, by using (6.11), only the displacement degrees of freedom are expanded into independent components, namely, in elastic and rigid body mode vectors, whereas the pressure degrees of freedom, $\boldsymbol{p}$, are not affected.

Since we can only find the solution of singular systems under special conditions as outlined in Section 6.2, we first have to project the matrices on $\mathbf{T}_s$ to separate the spaces explicitly and, subsequently, we can use the generalized inverses mentioned in (6.7c) to find the components in the $\mathbf{M}_s$-orthogonal space to $\mathbf{R}_s$ and also in the space of $\mathbf{R}_s$.

To accomplish this, we can rewrite the symmetric system matrices by using the

---

[4] Which is acceleration in this case.

6

τ-transformation matrix, namely,

$$\mathbf{K}_{sym} = \underbrace{\begin{bmatrix} \mathbf{K}_s & -\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}}_{\mathbf{K}_c} \underbrace{\begin{bmatrix} \mathbf{K}_s^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\tau_1} \underbrace{\begin{bmatrix} \mathbf{M}_s & \mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\tau_2} \tag{6.12}$$

for stiffness matrix and

$$\mathbf{M}_{sym} = \underbrace{\begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f \end{bmatrix}}_{\mathbf{M}_c} \underbrace{\begin{bmatrix} \mathbf{K}_s^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\tau_1} \underbrace{\begin{bmatrix} \mathbf{M}_s & \mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\tau_2} \tag{6.13}$$

for mass matrix.

While writing (6.12) and (6.13), we did not pay attention to singularity problem over $\mathbf{K}_s$. To tackle singular cases, we first project $\mathbf{K}_c$, $\mathbf{M}_c$ and τ_2 on $\mathbf{T}_s$ then we can perform the inversion partially by using generalized inverse, $\mathbf{K}_s^{+}$, in the space of elastic modes which are $\mathbf{M}_s$-orthogonal to $\mathbf{R}_s$. Namely,

$$\begin{aligned} \mathbf{K}_{ls} = {} & \overbrace{\begin{bmatrix} \mathbf{P}_s^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{R}_s^{\mathrm{T}} & \mathbf{0} \end{bmatrix}}^{\mathbf{T}_s^{\mathrm{T}}} \underbrace{\begin{bmatrix} \mathbf{K}_s & -\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}}_{\mathbf{K}_c} \overbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{R}_s \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{P}_s & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}}^{\mathbf{T}_s} \begin{bmatrix} \mathbf{K}_s^{+} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \\ & \underbrace{\begin{bmatrix} \mathbf{P}_s^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{R}_s^{\mathrm{T}} & \mathbf{0} \end{bmatrix}}_{\mathbf{T}_s^{\mathrm{T}}} \underbrace{\begin{bmatrix} \mathbf{M}_s & \mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\tau_2} \underbrace{\begin{bmatrix} \mathbf{P}_s & \mathbf{0} & \mathbf{R}_s \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix}}_{\mathbf{T}_s} \end{aligned} \tag{6.14}$$

$$\begin{aligned} \mathbf{M}_{ls} = {} & \overbrace{\begin{bmatrix} \mathbf{P}_s^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{R}_s^{\mathrm{T}} & \mathbf{0} \end{bmatrix}}^{\mathbf{T}_s^{\mathrm{T}}} \underbrace{\begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f \end{bmatrix}}_{\mathbf{M}_c} \overbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{R}_s \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{P}_s & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}}^{\mathbf{T}_s} \begin{bmatrix} \mathbf{K}_s^{+} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \\ & \underbrace{\begin{bmatrix} \mathbf{P}_s^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{R}_s^{\mathrm{T}} & \mathbf{0} \end{bmatrix}}_{\mathbf{T}_s^{\mathrm{T}}} \underbrace{\begin{bmatrix} \mathbf{M}_s & \mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\tau_2} \underbrace{\begin{bmatrix} \mathbf{P}_s & \mathbf{0} & \mathbf{R}_s \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix}}_{\mathbf{T}_s} \end{aligned} \tag{6.15}$$

Performing the multiplications in (6.14) and (6.15), we end up with the matrices which have the local symmetry, namely,

$$\mathbf{K}_{ls} = \begin{bmatrix} \mathbf{M}_s\mathbf{P}_s & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f & \mathbf{0} \\ \mathbf{0} & -\mathbf{R}_s^{\mathrm{T}}\mathbf{K}_{sf} & \mathbf{0} \end{bmatrix} \tag{6.16}$$

$$\mathbf{M}_{ls} = \begin{bmatrix} \mathbf{M}_s\mathbf{K}_s^{\oplus}\mathbf{M}_s & \mathbf{M}_s\mathbf{K}_s^{\oplus}\mathbf{K}_{sf} & \mathbf{0} \\ \mathbf{K}_{sf}^{\mathrm{T}}\mathbf{K}_s^{\oplus}\mathbf{M}_s & \left(\mathbf{M}_f + \mathbf{K}_{sf}^{\mathrm{T}}\mathbf{K}_s^{\oplus}\mathbf{K}_{sf} + \mathbf{K}_{sf}^{\mathrm{T}}\mathbf{R}_s\mathbf{R}_s^{\mathrm{T}}\mathbf{K}_{sf}\right) & \mathbf{K}_{sf}^{\mathrm{T}}\mathbf{R}_s \\ \mathbf{0} & \mathbf{R}_s^{\mathrm{T}}\mathbf{K}_{sf} & \mathbf{I} \end{bmatrix} \tag{6.17}$$

with

$$\mathbf{K}_s^{\oplus} = \mathbf{P}_s\mathbf{K}_s^{+}\mathbf{P}_s^{\mathrm{T}} \tag{6.18}$$

In all the derivations, we also considered that the rigid body vectors are mass-normalized, namely, $\mathbf{R}_s^{\mathrm{T}}\mathbf{M}_s\mathbf{R}_s = \mathbf{I}$. $\mathbf{K}_{ls}$ and $\mathbf{M}_{ls}$ represent the *locally symmetric* component matrices in the extended coordinate space.

With the aforementioned properties, the locally symmetric component stiffness matrix can be partitioned as follows

$$\mathbf{K}_{ls} = \begin{bmatrix} \mathbf{K}_{sym} & \mathbf{0} \\ \mathbf{v}_k & \mathbf{0} \end{bmatrix} \tag{6.19}$$

where

$$\mathbf{K}_{sym} = \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}, \quad \mathbf{v}_k = \begin{bmatrix} \mathbf{0} & -\mathbf{R}_s^{\mathrm{T}}\mathbf{K}_{sf} \end{bmatrix} \tag{6.20}$$

Similarly the component mass matrix reads as

$$\mathbf{M}_{ls} = \begin{bmatrix} \mathbf{M}_{sym} & \mathbf{v}_m^{\mathrm{T}} \\ \mathbf{v}_m & \mathbf{I} \end{bmatrix} \tag{6.21}$$

where

$$\begin{gathered} \mathbf{M}_{sym} = \begin{bmatrix} \mathbf{M}_s\mathbf{K}_s^{\oplus}\mathbf{M}_s & \mathbf{M}_s\mathbf{K}_s^{\oplus}\mathbf{K}_{sf} \\ \mathbf{K}_{sf}^{\mathrm{T}}\mathbf{K}_s^{\oplus}\mathbf{M}_s & \left(\mathbf{M}_f + \mathbf{K}_{sf}^{\mathrm{T}}\mathbf{K}_s^{\oplus}\mathbf{K}_{sf} + \mathbf{K}_{sf}^{\mathrm{T}}\mathbf{R}_s\mathbf{R}_s^{\mathrm{T}}\mathbf{K}_{sf}\right) \end{bmatrix}, \\ \mathbf{v}_m = \begin{bmatrix} \mathbf{0} & \mathbf{R}_s^{\mathrm{T}}\mathbf{K}_{sf} \end{bmatrix} \equiv -\mathbf{v}_k \end{gathered} \tag{6.22}$$

To find the symmetric component matrices presented in (6.19) and (6.21), we projected the necessary matrices on $\mathbf{T}_s$. To complete the presentation on the equations of motion of the component, we also have to project the external and interface forces of the component onto the same space. This practice is the analog of equations presented in Chapter 5, Section 5.4.3. Without going into the details of the compatibility presentation, we can write the equations of motion for component k which has a singular $\mathbf{K}_s$ matrix as follows

$$\mathbf{M}_{ls}^{(k)}\ddot{\tilde{\boldsymbol{q}}}_t^{(k)} + \mathbf{K}_{ls}^{(k)}\tilde{\boldsymbol{q}}_t^{(k)} = {\mathbf{T}_s^{(k)}}^{\mathrm{T}}\boldsymbol{f}^{(k)} + {\mathbf{T}_s^{(k)}}^{\mathrm{T}}\boldsymbol{g}^{(k)} \tag{6.23}$$

It is important to note that (6.23) is an extension of equation (5.103a) for components with singularities.

6.3.1. Format of the reduced component matrices in the extended coordinates

In the form of (6.19) and (6.21), the symmetric parts of the matrices can be reduced in a standard Craig&Bampton (CB) fashion. We can perform the regular projection operations to come up with the reduced system matrices. However, there is an additional point to pay attention to. Namely, we only have to perform the CB reduction in the space of elastic modes and we do not change the system components in the space of $\mathbf{R}_s$. We can write the CB transformation in the extended coordinate space as follows:

$$\begin{bmatrix} \tilde{\boldsymbol{q}} \\ \boldsymbol{\alpha}_s \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{cb} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\xi}} \\ \boldsymbol{\alpha}_s \end{bmatrix} \leftrightarrow \tilde{\boldsymbol{q}}_t = \mathbf{R}_{cb,t} \tilde{\boldsymbol{\eta}}_t \tag{6.24}$$

where $\tilde{\boldsymbol{q}}_t$ represents the transformed coordinates shown in (6.11). It is important to note that to perform the reduction on the component level, in (6.24), $\tilde{\boldsymbol{q}}_t$ must be ordered properly before projection. It is worth mentioning that, in (6.24), there is no transformation for the rigid body mode vector intensities. In this practice, $\mathbf{R}_{cb,t}$ represents the extended CB projection basis where the row size is increased by the size of the null space of the singular operator, $\mathbf{K}_s$.

In closed form, the reduction basis is similar to the one used for nonsingular cases presented in Chapter 5 and reads as,

$$\mathbf{R}_{cb} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \boldsymbol{\Phi}_c & \boldsymbol{\Phi}_{ii} \end{bmatrix} \tag{6.25}$$

with

$$\boldsymbol{\Phi}_c = -\mathbf{K}_{sym,ii}^{-1} \mathbf{K}_{sym,ib} \tag{6.26}$$

Fixed interface modes, $\boldsymbol{\Phi}_{ii}$, of the coupled components result from the solution of the eigenvalue problem with $\mathbf{K}_{sym,ii}$ and $\mathbf{M}_{sym,ii}$ as it was outlined before for regular components.

Using the ordered forms of the matrices $\mathbf{K}_{ls}$ and $\mathbf{M}_{ls}$[5], we can write the reduced form of the component matrices as follows,

$$\mathbf{k}_t = \begin{bmatrix} \mathbf{R}_{cb} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{K}_{sym}^{cb} & \mathbf{0} \\ \mathbf{v}_k^{cb} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{cb} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{cb}^{\mathrm{T}} \mathbf{K}_{sym}^{cb} \mathbf{R}_{cb} & \mathbf{0} \\ \mathbf{v}_k^{cb} \mathbf{R}_{cb} & \mathbf{0} \end{bmatrix} \tag{6.27}$$

for the stiffness matrix and as

$$\mathbf{m}_t = \begin{bmatrix} \mathbf{R}_{cb} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{M}_{sym}^{cb} & \mathbf{0} \\ \mathbf{v}_m^{cb} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{cb} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{cb}^{\mathrm{T}} \mathbf{M}_{sym}^{cb} \mathbf{R}_{cb} & \mathbf{R}_{cb}^{\mathrm{T}} \mathbf{v}_m^{cb\,\mathrm{T}} \\ \mathbf{v}_m^{cb} \mathbf{R}_{cb} & \mathbf{I} \end{bmatrix} \tag{6.28}$$

for the mass matrix.

$\mathbf{k}_t$ and $\mathbf{m}_t$ represent the reduced component matrices in the extended coordinate space which is denoted with the $_t$ subscript. It is also worth mentioning that,

---

[5] To perform the projections, the orders must match.

in (6.27) and (6.28), row and column sizes of $\mathbf{v}_m^{cb}\mathbf{R}^{cb}$ and $\mathbf{v}_k^{cb}\mathbf{R}_{cb}$ equal the size of the null space of the component stiffness matrix $\mathbf{K}_s$. Eventually, the reduced matrices of a component in the extended space includes the null space contributions in addition to the normal reduction terms originating from the projection of the component matrices on the component reduction basis, $\mathbf{R}_{cb}$.

6.3.2. Effect of singularity transformations on compatibility relations of reduced component matrices

Similar to the nonsingular cases, the transformation used in (6.11) also has an effect on the compatibility relations for the components that have a singular $\mathbf{K}_s$ matrix. (6.11) was used to project some associated matrices onto the space where we could perform the solution with the help of the pseudo-inverse, namely, $\mathbf{K}_S^{\oplus}$.

Recalling the discussion in Chapter 5, we already presented the general compatibility relation as follows,

$$\mathbf{B}\boldsymbol{q} = \mathbf{0} \overset{\boldsymbol{q}=\tau\tilde{\boldsymbol{q}}}{\rightarrow} \mathbf{B}\tau\tilde{\boldsymbol{q}} = \mathbf{0} \tag{6.29}$$

or, in matrix format,

$$\begin{bmatrix} \mathbf{B}^{(1)} & \mathbf{B}^{(2)} & \cdots & \mathbf{B}^{(k)} \end{bmatrix} \begin{bmatrix} \tau^{(1)} & & & \\ & \tau^{(2)} & & \\ & & \ddots & \\ & & & \tau^{(k)} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{q}}^{(1)} \\ \tilde{\boldsymbol{q}}^{(2)} \\ \vdots \\ \tilde{\boldsymbol{q}}^{(k)} \end{bmatrix} = \mathbf{0} \tag{6.30}$$

where $\mathbf{B}$ includes the Boolean matrices, $\mathbf{B}^{(k)}$, storing the compatibility relations between the components.

As we have shown in the introduction part of Section 6.3, we still can create a locally symmetric representation for components that have singularities, however, the compatibility relation in (6.30) has to be rewritten by reconsidering the coordinate transformations over the singular components, if any.

Namely, for components with singularities, we have to perform an additional transformation on the Boolean matrices. This additional transformation is necessary due to the explicit representation of the components of the solution vector in the space of rigid body mode vectors and also in the space of elastic mode vectors as it was presented explicitly in (6.11). Since the original variables of the system have been transferred to the new space, namely, $\tilde{\boldsymbol{q}}_t$, we also have to represent the compatibility relations in this new space. However, it is important to note that we only have to perform this transformation on the components with singularities, namely, for components with a singular $\mathbf{K}_s$. For components with a non-singular $\mathbf{K}_s$ operator, we do not need to perform this transformation and the theory presented in Chapter 5 applies directly.

For simplicity and the ease of representations, let us assume that we are performing the component level reductions over the system presented in Figure 6.1 where the middle component has the singularity issue. For this system, the modified compatibility relations can be written in a matrix format as follows,

$$\begin{bmatrix}\mathbf{B}^{(1)} & \mathbf{B}^{(2)}\mathbf{T}_s^{(2)} & \mathbf{B}^{(3)}\end{bmatrix}\begin{bmatrix}\tau^{(1)} & & \\ & \tau^{(2)\oplus} & \\ & & \tau^{(3)}\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{q}}^{(1)} \\ \tilde{\boldsymbol{q}}_t^{(2)} \\ \tilde{\boldsymbol{q}}^{(3)}\end{bmatrix} = \mathbf{0} \tag{6.31}$$

It is important to note that, in (6.31), for component 2, to perform the transformation operations with τ-matrices we have to write the relations in the new coordinate space where the components were explicitly separated into two complementary spaces as outlined before.

We can generalize the above compatibility transformation for a component with singularities and present the closed form expression as,

$$\mathbf{B}^{(k)}\mathbf{T}_s^{(k)}\tau^{(k)\oplus}\tilde{\boldsymbol{q}}_t^{(k)} = \mathbf{0} \tag{6.32}$$

where, for the specific component, $\tau^{(k)\oplus}$, by also referring to (6.14) or (6.15), can be explicitly written as follows,

$$\tau^{(k)\oplus} = \begin{bmatrix}\mathbf{K}_s^{(k)+} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix}\underbrace{\begin{bmatrix}\mathbf{P}_s^{(k)\,\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{R}_s^{(k)\,\mathrm{T}} & \mathbf{0}\end{bmatrix}}_{\mathbf{T}_s^{(k)\,\mathrm{T}}}\underbrace{\begin{bmatrix}\mathbf{M}_s & \mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I}\end{bmatrix}}_{\tau_2}\underbrace{\begin{bmatrix}\mathbf{P}_s & \mathbf{0} & \mathbf{R}_s \\ \mathbf{0} & \mathbf{I} & \mathbf{0}\end{bmatrix}}_{\mathbf{T}_s^{(k)}} \tag{6.33}$$

It is important to note that, in (6.33), to perform the operations, the matrices should be ordered in the CB-order to be able to perform the multiplications in a consistent manner with $\mathbf{B}^{(k)}$. For the above presentation, numerical operations must be performed with the correct order of the involving matrices and details on how to perform these multiplications in an efficient manner were detailed in Chapter 5. Therefore, similar numerical operations must also be followed during the operations performed here.

Recalling the analysis outlined in Section 6.3.1 and the one in Chapter 5 for regular components, the reduced generalized form of the compatibility relation presented in (6.31) reads as,

$$\mathbf{B}_t\tau_t\mathbf{R}_{cb,t}\tilde{\boldsymbol{\eta}}_t = \mathbf{0} \tag{6.34}$$

where

$$\mathbf{B}_t = \begin{bmatrix} \mathbf{B}^{(1)} & \mathbf{B}_t^{(2)} & \cdots & \mathbf{B}^{(k)} \end{bmatrix} \quad \text{with} \quad \mathbf{B}_t^{(k)} = \mathbf{B}^{(k)}\mathbf{T}_s^{(k)} \tag{6.35a}$$

$$\tau_t = \begin{bmatrix} \tau^{(1)} & & & \\ & \tau^{(2)}{}^{\oplus} & & \\ & & \ddots & \\ & & & \tau^{(k)} \end{bmatrix}, \quad \mathbf{R}_{cb,t} = \begin{bmatrix} \mathbf{R}_{cb}^{(1)} & & & \\ & \mathbf{R}_{cb,t}^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{R}_{cb}^{(k)} \end{bmatrix} \tag{6.35b}$$

$$\tilde{\boldsymbol{\eta}}_t = \begin{bmatrix} \tilde{\boldsymbol{\eta}}^{(1)} \\ \tilde{\boldsymbol{\eta}}_t^{(2)} \\ \vdots \\ \tilde{\boldsymbol{\eta}}^{(k)} \end{bmatrix} \tag{6.35c}$$

In (6.35a), (6.35b) and (6.35c), subscript $_t$ is used to represent the components with singularities. And the representation can be generalized for systems having regular and singular components. Eventually, depending on the nature of the components whether having singularities or not, the ingredients of the matrices provided in (6.35a), (6.35b) and (6.35c) have to be arranged and used properly.

6.3.3. Generalized primal assembly

Having provided the effect of the singularity transformation on the compatibility relations, we are going to shortly outline the details of the assembly of the reduced component matrices in a primal way and generalize it. Theoretical framework of the assembly process was provided before for regular components in Chapter 5 and the discussion provided here is built based on the theory and notations used therein.

The modal τ-reduced system equations were written before in Chapter 5 for components with regular $\mathbf{K}_s$ matrices as follows,

$$\mathbf{m}_m\ddot{\tilde{\boldsymbol{\eta}}} + \mathbf{k}_m\tilde{\boldsymbol{\eta}} = \boldsymbol{f}_m + \boldsymbol{g}_m \tag{6.36a}$$

$$\mathbf{B}\tau\mathbf{R}_{cb}\tilde{\boldsymbol{\eta}} = \mathbf{0} \tag{6.36b}$$

$$\mathbf{L}_m^{\mathrm{T}}\boldsymbol{g}_m = \mathbf{0}. \tag{6.36c}$$

where $\mathbf{m}_m$, $\mathbf{k}_m$, $\boldsymbol{f}_m$ and $\boldsymbol{g}_m$ were presented in the modal coordinates also. Here, we show the extension of this system of equations for systems with singularities.

Namely, for systems having the singularity issue, the reduced component mass and stiffness matrices can be generalized as follows,

$$\mathbf{m}_{m,t} = \mathbf{R}_{cb,t}^{\mathrm{T}} \mathbf{M} \mathcal{T}_t \mathbf{R}_{cb,t} \equiv \begin{bmatrix} \mathbf{m}^{(1)} & & & \\ & \mathbf{m}_t^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{m}^{(k)} \end{bmatrix} \tag{6.37a}$$

$$\mathbf{k}_{m,t} = \mathbf{R}_{cb,t}^{\mathrm{T}} \mathbf{K} \mathcal{T}_t \mathbf{R}_{cb,t} \equiv \begin{bmatrix} \mathbf{k}^{(1)} & & & \\ & \mathbf{k}_t^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{k}^{(k)} \end{bmatrix} \tag{6.37b}$$

It is again important to note that, in (6.37a) and (6.37b), subscript $_t$ represents the reduced components with singularities. $\mathbf{M}$ and $\mathbf{K}$ store the original nonsymmetric coupled matrices of the components in a block diagonal format as presented in (5.105). For the general representation, (6.36b) is transformed into (6.34). As a last step, we have to transform (5.110c). In this transformation, we can make use of the representation of the component level equations of motion given in (6.23).

Recalling the discussions on the relations between the Boolean matrices and the interface forces, we wrote before in (5.79), namely,

$$\boldsymbol{g} = -\mathbf{B}^{\mathrm{T}} \boldsymbol{\lambda} \equiv - \begin{bmatrix} \mathbf{B}^{(1)^{\mathrm{T}}} \\ \mathbf{B}^{(2)^{\mathrm{T}}} \\ \vdots \\ \mathbf{B}^{(k)^{\mathrm{T}}} \end{bmatrix} \boldsymbol{\lambda} \tag{6.38}$$

which can be extracted from a common set of interface forces/fluxes acting between the components of the complete vibro-acoustic system.

However, as presented in (6.23), we need to perform the transformation also on the interface forces/fluxes of a component which is necessary to tackle the singularity issue correctly. Namely, assuming that component k has a singular $\mathbf{K}_s$ involved, the transformed interface force/flux vector reads as,

$$\boldsymbol{g}_t^{(k)} = \mathbf{T}_s^{(k)^{\mathrm{T}}} \boldsymbol{g}^{(k)} \tag{6.39}$$

By using (6.38), the interface force/flux vector for component k can be written as $\boldsymbol{g}^{(k)} = -\mathbf{B}^{(k)^{\mathrm{T}}} \boldsymbol{\lambda}$ which filters the related entries from the common vector $\boldsymbol{\lambda}$ (see (6.38)). Using this relation in (6.39), we can write the final form on the component level as follows,

$$\boldsymbol{g}_t^{(k)} = -\mathbf{T}_s^{(k)^{\mathrm{T}}} \mathbf{B}^{(k)^{\mathrm{T}}} \boldsymbol{\lambda} \tag{6.40}$$

(6.40) represents the interface force/flux vector in extended coordinates as it was presented in (6.11).

Having set the basis for the transformations and following (5.79) and (5.108b) from Chapter 5, the most general form of the modal interface force/flux vector can be written as

$$\boldsymbol{g}_{m,t} = -\mathbf{R}_{cb,t}^{\mathrm{T}} \mathbf{N}_s \mathbf{B}^{\mathrm{T}} \boldsymbol{\lambda} \equiv - \begin{bmatrix} \boldsymbol{g}_m^{(1)} \\ \boldsymbol{g}_{m,t}^{(2)} \\ \vdots \\ \boldsymbol{g}_m^{(k)} \end{bmatrix} \tag{6.41}$$

where

$$\mathbf{N}_s = \begin{bmatrix} \mathbf{I}^{(1)} & & & \\ & \mathbf{T}_s^{(2)\,\mathrm{T}} & & \\ & & \ddots & \\ & & & \mathbf{I}^{(k)} \end{bmatrix} \tag{6.42}$$

and $\mathbf{R}_{cb,t}$ was also defined before in (6.33). And, in (6.41) and (6.42), we assumed that component 2 has a singular stiffness matrix therefore the transformations must be applied on this component. However, we can easily generalize the presentation for a system having more components with singular stiffness matrices. For this, proper projections must be applied for the components that has the problem with the singularity of $\mathbf{K}_s$.

To conclude, we can write the generalized system with the reduced component matrices and the compatibility relations as,

$$\mathbf{m}_{m,t}\ddot{\tilde{\boldsymbol{\eta}}}_t + \mathbf{k}_{m,t}\tilde{\boldsymbol{\eta}}_t = \boldsymbol{f}_{m,t} + \boldsymbol{g}_{m,t} \tag{6.43a}$$

$$\mathbf{B}\mathbf{N}_s^{\mathrm{T}}\tau_t\mathbf{R}_{cb,t}\tilde{\boldsymbol{\eta}}_t = \mathbf{0} \tag{6.43b}$$

$$\mathbf{L}_{m,t}^{\mathrm{T}}\boldsymbol{g}_{m,t} = \mathbf{0} \tag{6.43c}$$

It is important to point out that (6.43a) to (6.43c) are counterparts of (5.110a) to (5.110c).

Following the detailed discussion provided in Chapter 5 for the primal assembly of the symmetric reduced system matrices, we follow the same path to perform the assembly for the generalized system outlined in (6.43a) to (6.43c). Namely, we construct a localization matrix [90] in the form of

$$\tilde{\boldsymbol{\eta}}_t = \tilde{\mathbf{L}}_{m,t}\tilde{\boldsymbol{\eta}}_t^g \tag{6.44}$$

where, again, it is important to note that $\tilde{\mathbf{L}}_{m,t}$ transforms the interface degrees of freedom of the reduced component matrices into a set where they are uniquely defined. And, it is this practice that facilitates the component coupling on the interfaces in a primal assembly framework. Similar to the analysis performed in Section 5.4.1, we substitute (6.44) in (6.43b) to get

$$\left(\mathbf{B}\mathbf{N}_s^{\mathrm{T}}\tau_t\mathbf{R}_{cb,t}\right)\tilde{\mathbf{L}}_{m,t}\tilde{\boldsymbol{\eta}}_t^g = \mathbf{0} \tag{6.45}$$

Observing (6.45), we can easily note that $\tilde{\mathbf{L}}_{m,t} = \text{null}(\mathbf{B}\mathbf{N}_s^{\mathrm{T}}\tau_t\mathbf{R}_{cb,t})$ which represents the *right null space* that is used to assemble the reduced components in the *extended coordinate space*.

Referring to Section 5.4.1, to complete the primal assembly of the components, we also have to cancel the interface forces/fluxes, namely, $\boldsymbol{g}_{m,t}$ in (6.43a). This is already presented in (6.43c), and in order to find $\mathbf{L}_{m,t}$, we have to recall the definition in (6.41). Namely, rewriting (6.43c) by substituting (6.41) results in

$$-\mathbf{L}_{m,t}^{\mathrm{T}}\left(\mathbf{R}_{cb,t}^{\mathrm{T}}\mathbf{N}_s\mathbf{B}^{\mathrm{T}}\right)\boldsymbol{\lambda} = \mathbf{0} \tag{6.46}$$

From (6.46), we can also note that $\mathbf{L}_{m,t} = \text{null}\left((\mathbf{R}_{cb,t}^{\mathrm{T}}\mathbf{N}_s\mathbf{B}^{\mathrm{T}})^{\mathrm{T}}\right)$.

As a result, substituting (6.44) in (6.43a) and premultiplying (6.43a) by $\mathbf{L}_{m,t}^{\mathrm{T}}$, we can find the primal assembled system equations in the *extended coordinate space*, namely,

$$\mathbf{m}_t^g\ddot{\tilde{\boldsymbol{\eta}}}_t^g + \mathbf{k}_t^g\tilde{\boldsymbol{\eta}}_t^g = \boldsymbol{f}_t^g \tag{6.47}$$

where

$$\mathbf{m}_t^g = \mathbf{L}_{m,t}^{\mathrm{T}}\mathbf{m}_{m,t}\tilde{\mathbf{L}}_{m,t} \tag{6.48a}$$

$$\mathbf{k}_t^g = \mathbf{L}_{m,t}^{\mathrm{T}}\mathbf{k}_{m,t}\tilde{\mathbf{L}}_{m,t} \tag{6.48b}$$

$$\boldsymbol{f}_t^g = \mathbf{L}_{m,t}^{\mathrm{T}}\boldsymbol{f}_{m,t} \tag{6.48c}$$

Eventually, $\mathbf{m}_t^g$ and $\mathbf{k}_t^g$ represent the system level reduced nonsymmetric matrices which are used to approximate the dynamic response of the full model.

6.4. Numerical example

In this section, we present a two-dimensional academic test case to assess the results found with the generalized representation outlined above. The selected example is similar to the model that is used in Section 5.6 but the dimensions are slightly modified. Namely, a two-dimensional cavity is modelled where the fluid domain is coupled to a structural beam. The finite element model is again constructed in ANSYS by using the implemented structural-acoustic interface. System matrices, node and element tables, the definition of the degrees of freedom, information on the interfaces of the components are also extracted from ANSYS with the developed interface. The length and height of the test domain are 1m and 0.4m, respectively. Test domain is depicted in Figure 6.3. The component domains with the corresponding dimensions are also shown in the same figure. Moreover, we also show the component interfaces with the dashed lines in Figure 6.3.

The structural part is modelled with beam elements, namely, BEAM3 in ANSYS, having two translational and one rotational degrees of freedom. Coupling beam is fully constrained at the two ends. The fluid domain is modelled with quadrilateral 2D acoustic fluid elements, namely, FLUID29, which use pressure and the two translations in plane as the main degrees of freedom. The interface coupling information between structural and acoustic elements is implemented in ANSYS. The rest of the walls of the model are considered to be rigid. We use matching meshes on the interfaces of the components. For the numerical experiments conducted, we selected water as the fluid that fills the acoustic cavity. The related properties of water is taken as follows: density, 1000 kg/m^3, and speed of sound, 1500 m/s. The properties of the structural beam is given as follows: density, 2700 kg/m^3, Young's modulus, 71 GPa, and Poisson ratio 0.33, beam height, 5 mm and beam width, 50 mm.

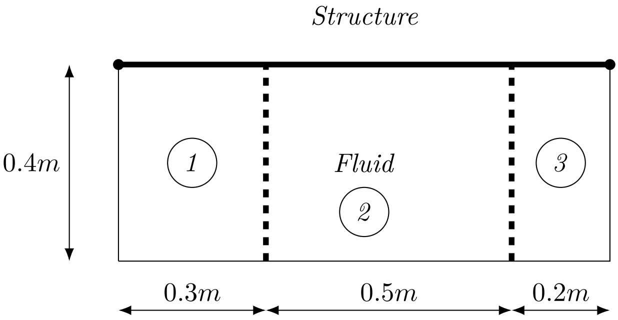


Figure 6.3: 2D structural-acoustic test domain

To show the accuracy of the reduced problem with the generalized technique presented, we use the following error estimator which is similar to the one used in Section 5.6, namely,

$$\epsilon_\omega = \frac{f_{red} - f_{full}}{f_{full}} \tag{6.49}$$

where f_{red} represents the reduced frequencies extracted from the reduced matrices presented in (6.47) and f_{full} represents the frequencies calculated by using the full coupled system matrices.

The model in Figure 6.3 is meshed with an element size of 0.02 m resulting in 1218 coupled degrees of freedom in total. For this model, the fluid domain has 1071 degrees of freedom. The structural beam part has 147 degrees of freedom. The number of interface degrees of freedom resulting from the dashed interface lines in Figure 6.3 is 24 for this model, in which 21 (the number of nodes in y direction) is the number of acoustic degrees of freedom and 3 is the number of structural degrees of freedom (translations in two directions, x and y, and a rotation dof). The overview on the degrees of freedom of the components is provided in Table 6.1 for different fixed interface mode numbers, m. We have to note that the reduced component matrix size of component 2 is slightly increased due to the explicit separation of the rigid body and elastic mode components as outlined in Section 6.3. Namely, the middle component has 3[6] additional rows/columns in comparison to reduced matrices of the other components.

---

[6] The column size of the null space of $\mathbf{K}_s$ on this component is 3.

Component	Full size	Reduced component matrix sizes	
		$m = 8$	$m = 16$
1	381	32	40
2	624	59	67
3	261	32	40

Table 6.1: Information on component level degrees of freedom

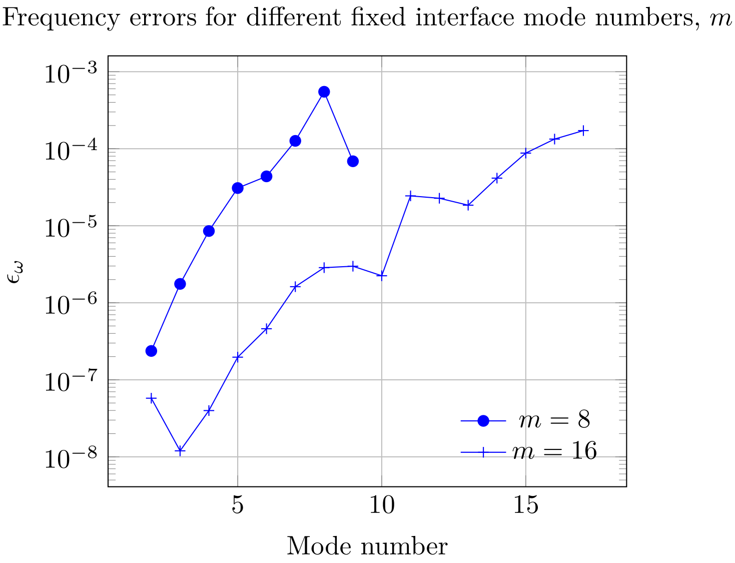


Figure 6.4: Error for different internal mode counts

6.5. Summary and conclusions

In this chapter, we extended the method presented in Chapter 5 and generalized the technique for components including singularities. Namely, the transformation to symmetric format can not be performed explicitly due to the singularity of the structural stiffness matrix, $\mathbf{K}_s$, which is used in the transformation matrix τ.

Nevertheless, by using the generalized inverse concept, we show that it is possible to represent the solution of a singular linear system of equations as a combination of two components: one in the space of the rigid body vectors[7] and the other in the space of elastic modes. With this practice, we can still write the explicit symmetric forms of the component matrices and perform the projections with these matrices. Since we explicitly separate the solution vectors into two spaces, the effect of this operation is also reflected on the structure of the symmetric component matrices. Namely, the resulting matrices have two blocks where one of these blocks is represented in the space of elastic modes and the other block is represented in the space of rigid body vectors. Subsequently, similar to Chapter 5, we show that we can perform a normal CB reduction by performing the projection on the symmetric blocks of the component matrices. And the blocks represented in the space of rigid body vectors are condensed in size.

Last, we show that the compatibility equations must also be arranged according to the transformation to the extended coordinate space. In analogy with Chapter 5, these modified interface compatibility relations are used to formulate a generalized primal coupling approach for the reduced component matrices.

On an academic test case, the reduction performance and the accuracy of the results found with the reduced problem are tested. It is shown that the relative errors on the eigenvalues for different fixed interface mode counts is very small. Although, the model used in this chapter is an academic model resulting in small matrices, we can also conclude that substantial reduction can be achieved both on the component and assembled system level.

---

[7] In otherwords, the vectors spanning the null space of $\mathbf{K}_s$.

7
Conclusions & Recommendations

7.1. Conclusions of the thesis

With the increase in the detail of the models used in engineering simulations, the requirement for efficient simulation tools for large industrial finite element models has increased substantially. This thesis addresses some contributions to the body of the numerical solution techniques available in the literature. Namely, vibro-acoustic problems are investigated in which a structural and acoustic domain are strongly coupled. And the coupling between the domains can not be neglected for realistic predictions most of the time. As stated in the introduction, the goal of the research was

- *To improve and/or propose new numerical solution techniques for the solution of vibro-acoustic problems,*
- *To develop new model order reduction (MOR) techniques for vibro-acoustic problems.*

The research presented in detail in the main matter of the thesis outlines two main contributions with these initially stated objectives: development of eigenvalue solution techniques and development of model order reduction techniques. Moreover, a numerical platform for the implementation of these techniques is also proposed where all the numerical tests were performed. Below, the details of these contributions are outlined.

7.1.1. Developments on eigenvalue solution techniques

One of the main contributions addressed in this thesis is the developments on the eigenvalue solution techniques for vibro-acoustic problems. The main conclusions on this research are summarized below.

Chapter 3 sets out with a modal basis approach which is commonly used for model reduction studies in the literature for weakly coupled problems. But, generally, this approach does not give good results for strongly coupled problems. To alleviate the problems associated with strongly coupled problems, an iterative technique is proposed in Chapter 3. The proposed technique is based on Subspace iteration technique used for symmetric eigenvalue problems. The proposed technique uses the above mentioned decoupled basis vectors as initial vectors to start with. In the course of the iterations, it is shown that the initial decoupled basis vectors are substantially improved. To improve these vectors, at each iteration step, the projection space is enriched with some *dynamic correction vectors represented in the space of the initial decoupled basis vectors*, some *modal truncation vectors* and with *some static residual vectors computed with the coupled stiffness matrix*. By using a transformation matrix, we show that there is a relation between the coupled residual vector and the original Krylov vectors. It is concluded that the use of the residual vectors along with the other components of the reduction basis brings a significant impact on the convergence of the technique, which is due to the fact that the residual vectors include components of the original Krylov vectors as a linear combination. Therefore, it is concluded that the developed technique can be interpretted as an enriched Subspace iteration technique for vibro-acoustic problems.

Chapter 4 addresses the first major scientific contribution of this thesis. Namely, by following and building on the research line of Chapter 3, the Subspace iteration variant presented therein is transformed to a Lanczos type eigenvalue solver operating on symmetric matrices. However, these symmetric matrices are never built explicitly. This practice brings several advantages. The most important advantage over the Subspace iteration variant of Chapter 3 is that the information generated in the previous iteration steps is not discarded but used directly to build a better and enlarging projection space. It is also concluded that the Krylov vectors generated with the proposed symmetric variant span the same space as the ones generated with the Arnoldi type eigenvalue solver. Another advantage is the build up of a symmetric tridiagonal matrix due to the repeated projections onto symmetric system matrices. With this practice, the eigenvalue and eigenvector approaximations can be calculated with a symmetric tridiagonal matrix. Based on the well-known three-term recurrence relations found between the iteration vectors computed during a Lanczos cycle, a numerical cost reduction scheme using partial reorthogonalization is incorporated into the process. The performance of the developed technique is compared with the Two-sided Lanczos solver and the Arnoldi eigenvalue solvers. With the developed numerical framework, numerical performance tests were performed on two 3D test cases by using different fluids. It is shown that the technique gives highly accurate results. It is also shown that the integration of the partial orthogonalization scheme further decreases the overall numerical cost of the operations for large target eigenvalue counts. Namely, the gain in CPU time is on the order of 10% for large target eigenvalue counts.

7.1.2. Developments on model reduction techniques

For efficient simulation and optimization of systems including vibro-acoustic coupling, model reduction techniques are valuable tools to decrease the computational resources and the time used for the analysis. Therefore, in the second part of this thesis, we mainly addressed this problem by developing a Craig & Bampton (CB) type reduction technique for problems including vibro-acoustic coupling. As a result, Chapters 5 and 6, as a whole, outline the second major scientific contribution of this thesis and the main research conclusions on the developed technique are summarized and outlined below.

In Chapter 4, the building blocks of an efficient eigenvalue solver for vibro-acoustic problems is presented. The proposed model reduction technique presented in Chapter 5 extends and builds on the main ideas presented in Chapter 4, namely, creating symmetric reduced component matrices. These symmetric reduced matrices are later on coupled with the other reduced component matrices for the approximation of the global system response. Although the computations can not be carried out in closed form efficiently due to the fully populated nature of the symmetric mass matrix, we have shown that the computations can be arranged in an efficient manner by only using simple ordering operations on the matrices involved. And this practice is used to perform two critical operations. Namely, the first one is the computation of the fixed interface modes of the component. The second one is the computation of the reduced mass matrix by projecting the symmetric mass matrix of the component on the CB basis. By using these arrangements, at first place, we can compute the ingredients of the CB reduction basis efficiently. There after, we show that we can also calculate the reduced mass matrix efficiently without using the fully populated matrices explicitly. Besides, we also show that the reduced symmetric stiffness matrix of the components has a special block diagonal structure. To approximate the dynamic system response, the reduced components must be coupled either in a primal or dual assembly framework. In this thesis, we concentrate on the primal assembly. In the discussions, we show that the primal assembly of the reduced symmetric components needs some attention due to the modified compatibility relations. To accomplish the primal assembly of the reduced component matrices, we provide the mathematical proofs that we need to use different left and right null spaces. Namely, use of different null spaces is necessary because the selected transformation matrix τ results in a change in the interface degrees of freedom and, therefore, the compatibility relations are also modified. In contrast, it is also shown that the compatibility relations between the interface forces are not affected by this transformation. The developed technique was tested with two different 2D finite element models with a coarse and fine mesh, respectively. On these moderate and large size[1] problems, it is concluded that the accuracy of the eigenvalues and eigenvectors computed with the proposed reduction technique is quite high.

Depending on the division of the whole domain of into components and on the boundary conditions applied on these components, Chapter 6 extends the technique and generalizes it for components with singularities. Namely, for free-free components, τ matrix results in a singularity since the inverse of the stiffness matrix of the structure is not defined to perform the transformations to symmetric matrices. However, by using the pseudo-inverse concept, it is shown that it is still possible to perform the operations and come up with locally symmetric representations in closed form. And this is accomplished by representing the main variables of the problem in two spaces, namely, the space of elastic and rigid body modes. It is shown that the symmetric part of the matrices can be used to find the CB basis as a normal regular component without the singularity issue. The theory presented in

---

[1] In terms of the size of the matrices.

Chapter 5 is generalized in Chapter 6 for components which are free-free[2]. Similar to Chapter 5, the compatibility equations are formulated and the mathematical details of component coupling are outlined. On a 2D academic test case, we show that the generalized technique results in highly accurate eigenvalues.

7.2. Recommendations for further investigations

Having outlined the main research contributions, we have to mention that there are still several research topics which can potentially result in new contributions to the field, namely:

- Damping is not explicitly considered in this thesis. Extension of the presented research to systems including structural and/or fluid damping is a prominent research area which can be built on the research presented in this thesis. Moreover, this research should possibly be supported by experimental validations also.

- A natural extension of the research presented on vibro-acoustic component mode synthesis, namely in Chapters 5 and 6, is the investigation of *free-interface* methods. These methods keep the physical nature of the interfaces and, therefore, they are better suited for further experimental validations.

- Increasing the number of components analyzed, the number of interface degrees of freedom also increases substantially. This increase will be even more pronounced for 3D models where the number of kept interface degrees of freedom can become quite high. Therefore, as a follow up, investigation of interface reduction techniques is highly recommended to further decrease the reduced system size and the overall solution time.

- Main theme of the research in this thesis was on the calculation of dynamic response parameters, namely, eigenvalues and eigenvectors, another area which is equally important is the computation of the frequency response functions (FRF) of vibro-acoustic systems. Therefore, by using the experience gained in analyzing and comparing the different techniques from different application fields, namely as presented in Appendix A, it is also recommended to investigate model reduction techniques for approximating the *input-output* behaviour on these systems.

- It was pointed out in Chapter 4 that the round-off related terms were neglected for the analysis on partial re-orthogonalization relations. Therefore, it is recommended to extend the research presented therein by taking these terms into account and investigate their effects on the convergence of the technique.

- In Chapter 5, for the solution of fixed interface modes of a component, it was pointed out that hybrid eigenvalue solvers can be developed. Namely, to circumvent factorization costs and to decrease the numerical cost further, it is

---

[2] Constraint-free

highly recommended to investigate iterative solution techniques for the inverse iteration step of the symmetric Lanczos process.

A

Model Reduction Techniques in Different Application Fields

In this chapter, we reviewed the most commonly used model reduction techniques from different application fields, namely, structural dynamics, numerical mathematics and systems and control community. We briefly outlined the mathematical details of the popular reduction techniques in the above mentioned application areas. Subsequently, we provided a qualitative comparison over the different methods. On a moderately sized industrial problem, we performed a comparison over these model reduction methods from an input-output perspective. We also provided some error measures depending on different input-output scenarios. The results presented in this appendix was published in [23].

A.1. Introduction

An important tool in the design of complex high-tech systems is the numerical simulation of predictive models. However, these dynamical models are typically of high order, i.e. they are described by a large number of ordinary differential equations. This results from either the inherent complexity of the system or the discretization of partial differential equations. Model reduction can be used to find a low-order model that approximates the behavior of the original high-order model, where this low-order approximation facilitates both the computationally efficient analysis and controller design for the system to induce desired behavior.

The earliest methods for model reduction belong to the field of structural dynamics, where the dynamic analysis of structures is of interest. Typical objectives are the identification of eigenfrequencies or the computation of frequency response functions. Besides the mode displacement reduction method and extensions thereof (see e.g. [126, 63]), important techniques are given by component mode synthesis techniques [80, 37], which started to emerge in the 1960s.

The model reduction problem has also been studied in the systems and control community, where the analysis of dynamic systems and the design of feedback controllers are of interest. Some of the most important contributions were made in the 1980s by the development of balanced truncation [104, 44] and optimal Hankel norm approximation [65].

Finally, numerically efficient methods for model reduction have been developed in the field of numerical mathematics in the 1990s. Important techniques are asymptotic waveform evaluation [124], Padé-via-Lanczos [50] and rational interpolation [67]. These methods are often applied in the design and analysis of large electronic circuits.

Despite the fact that the above techniques essentially deal with the same problem of model reduction, the results in the fields of structural dynamics, systems and control and numerical mathematics have largely been developed independently. This chapter aims at providing a thorough comparison between the model reduction techniques from these three fields, facilitating the choice of a suitable reduction procedure for a given reduction problem. To this end, the most popular methods from the fields of structural dynamics, systems and control and mathematics will be reviewed. Then, the properties of these techniques will be compared, where both theoretical and numerical aspects will be discussed. In addition, these differences and commonalities will be illustrated by means of application of the model reduction techniques to a common example.

Reviews of model reduction techniques exist in literature. However, these reviews mainly focus on methods from the individual fields, i.e. they focus on methods from structural dynamics [36, 90], systems and control [69] or numerical mathematics [12, 58] only. Nonetheless, methods from systems and control and numerical mathematics are reviewed and compared in [8, 7, 61], where the comparison is mainly performed by the application of the methods to examples. In the current chapter, we will review the popular model reduction techniques from all the three fields mentioned above. Additionally, both a qualitative and quantitative comparison will be provided. The focus of this chapter is on this comparison. And, therefore, its aim is not to present a full comprehensive historical review of all the methods in these three domains.

In this chapter, the scope will be limited to *model-based* reduction techniques for *linear time-invariant dynamical systems*. Consequently, the data-based model reduction technique of proper orthogonal decomposition [143, 22] will not be discussed. For an overview of proper orthogonal decomposition, see [94, 86]. Also, reduction methods for nonlinear dynamical systems (see e.g. [127, 139]) fall outside the scope of the discussions presented in this chapter.

The outline of this chapter is as follows. First, the most important model reduction techniques from the fields of structural dynamics, numerical mathematics and systems and control will be reviewed in Section A.2. In Section A.3, a qualitative comparison between these methods will be provided, focussing on both theoretical and numerical aspects. This comparison will be illustrated by means of an example problem in Section A.4, which further clarifies differences and commonalities between methods. Finally, conclusions will be stated in Section A.5.

A.2. Review of model reduction techniques

In this section, popular model reduction techniques from different fields are discussed. In Section A.2.1, methods from structural dynamics are discussed, whereas model reduction techniques from the fields of numerical mathematics and systems and control are discussed in Sections A.2.2 and A.2.3, respectively.

A.2.1. Mode displacement methods

In the field of structural dynamics, the design and performance evaluation of mechanical systems is of interest. Herein, the computation of deformations, internal stresses or dynamic properties are subject of analysis. Even though the goal of analysis might differ from one specific application to another, important objectives are the prediction of regions with high stress, prediction of the eigenvalues (related to resonance frequencies) and eigenvecttors (related to structural eigenmodes), the computation of the system's response to a certain excitation in time domain and the fast computation of frequency response functions.

All of the above mentioned goals share a common property. Namely, the models used in the design must, generally, contain detailed information for the precise description of the response properties of the structure. The mathematical models are basically constructed in terms of partial differential equations. These equations might be solved exactly only for simple problems and one has to resort to discretization-based approaches, such as the finite element method (FEM) or the boundary element method (BEM). The techniques discussed in this section are typically used for FEM models, but the same concepts can be applied to BEM.

Model reduction methodologies are efficiently used in the structural field since the 19th century. The most common methods are mode superposition methods [126], in which a limited number of free vibration modes of the structure is used to represent the displacement pattern [35]. There are also improvements of the original mode superposition method by the addition of different vecttors to the expansion procedure, such as the mode acceleration or modal truncation augmentation [126, 157]. Mode superposition methods are generally considered for the complete structure. However, it is common to partition the structure in some components, on which model reduction is performed individually. Then, these reduced-order component models are coupled to represent the global behavior. These methods are all together named component mode synthesis techniques. These methods are extensions of common mode superposition methods to the partition level where the forces on the partition boundaries replace the general forces on the whole structure. In [79] and [80], Hurty provided a general method for component mode synthesis techniques. Craig and Bampton, in [37], used the static deformation shapes of the substructure with respect to its boundary displacements and enriched this space with the internal dynamic mode shape vecttors to increase the accuracy. This method is known as the *fixed-interface reduction method* because the modes of the system are found while all the boundaries are fixed. Later on, the works of MacNeal [101] and Rubin [132] extended these methods to a class of methods known as *free-interface methods*. In these methods, the dynamic mode shape vecttors used in the basis are computed without the application of any restraints on the component boundaries, where in fixed-interface methods the boundary degrees of freedom are all fixed. For an overview of dynamic substructuring methods, see e.g. [36, 90]. Moreover, an automated approach towards substructuring is given by automated multilevel substructuring (AMLS), see [21]. Finally, it is remarked that substructuring can also effectively exploit parallel computing.

Discretization-based methods, such as FEM, analyze complex engineering problems by constructing piece-wise approximation polynomials over the spatial domain and solve for the unknown variables at specific locations of the discretization, known as node points [122]. This representation might already be considered to be a model order reduction process in itself. Namely, the displacement $u(z,t)$, which is dependent on the spatial variable z and time t, is represented by the finite expansion

$$u(z,t) = \sum_{j=1}^{N} \Psi_j(z) q_j(t). \tag{A.1}$$

Herein, $\Psi_j(z)$ are linearly independent functions representing the displacement shape of the structure, where it is noted that they satisfy the essential boundary conditions of the problem. Next, $q_j(t)$ are the unknown functions of time, whereas N represents the number of functions exploited in the representation. Since the representation of a body consists of infinitely many points (and therefore infinitely many degrees of freedom), the finite expansion (A.1) has already accomplished the task of reducing the system to a finite number of degrees of freedom.

The discretization of the differential equations of the problem results in the equations of motion of the system, which are typically of the following form:

$$\mathbf{M}\ddot{\boldsymbol{q}} + \mathbf{K}\boldsymbol{q} = \boldsymbol{f}, \tag{A.2}$$

where $\mathbf{M} \in \mathbb{R}^{N\times N}$ and $\mathbf{K} \in \mathbb{R}^{N\times N}$ represent the mass and stiffness matrices, respectively. Furthermore, $\boldsymbol{q} \in \mathbb{R}^N$ represents the unknown displacements of the structure and $\boldsymbol{f} \in \mathbb{R}^N$ is the externally applied generalized force vecttor. Structural systems possess, most of the time, light damping and the reduction typically is based on the undamped system. Therefore, undamped systems of the form (A.2) are considered in this section. However, it is stressed that this is only suitable when the system is lightly damped and the eigenfrequencies are well separated [63].

In general, a detailed problem representation and the use of a high number of elements in the discretization result in large matrices and, hence, in long computation times. Model order reduction methods are used to efficiently reduce the system size and, as a consequence, achieve acceptable computation times. Reduction methods in structural dynamics may be classified into two classes, namely, methods related to mode superposition and methods related to component mode synthesis techniques. In this section, the context is limited to mode superposition methods, since they apply to the full system. This enables a comparison with methods from the fields of numerical mathematics and systems and control. More information on component mode synthesis can be found in [36].

Mode superposition methods share the common property that they use a small number of free vibration modes to represent the dynamics of the structure with some reduced number of generalized degrees of freedom. With this selection, one represents the solution vecttors as a summation of free vibration modes that form a linearly independent set. This operation therefore reduces the system size to be solved and could result in important computational gains. However, there are some important points to note on the expansion procedures used in practice [117], namely:

1. the used mode shape vecttors do not span the complete space;
2. the computation of eigenvecttors for large systems is very expensive and time consuming;
3. the number of eigenmodes required for satisfactory accuracy is difficult to estimate a priori, which limits the automatic selection of eigenmodes;
4. the eigenbasis ignores important information related to the specific loading characteristics such that the computed eigenvecttors can be nearly orthogonal to the applied loading and therefore do not participate significantly in the solution.

Three different main variants can be considered which are often used in structural dynamics community. These are the mode displacement method, mode acceleration method and modal truncation augmentation method. The latter two methods are enhancements of the mode displacement method with the addition of the contribution of the omitted parts in an expansion process.

Generally, these methods do not propose the computation of an error bound for the response. Consequently, the success of the methods is established on the basis of a posteriori error comparisons. Typically, either the errors on the eigenfrequencies or the errors on the input-output representation are used to show the success of the applied method. In the following sections, the mode displacement method, the mode acceleration methods and the modal truncation augmentation method will be treated in more detail.

Mode displacement method

The equation of motion of the structure (A.2) is recalled:

$$\mathbf{M}\ddot{\boldsymbol{q}} + \mathbf{K}\boldsymbol{q} = \boldsymbol{f}.$$

Then, the mode displacement method is based on the free vibration modes of the structure, which can be found by using a time-harmonic representation for the displacement of the unforced system (i.e. $\boldsymbol{f} = \mathbf{0}$). This leads to the generalized eigenvalue problem

$$\left(\mathbf{K} - \omega_j^2 \mathbf{M}\right) \boldsymbol{\phi}_j = \mathbf{0}, \tag{A.3}$$

where $\boldsymbol{\phi}_j$ is the mode shape vecttor corresponding to the eigenfrequency ω_j, with $j \in \{1, \ldots, N\}$. Using the expansion concept along with the mode shape vecttors $\boldsymbol{\phi}_j$, the displacement can be represented as follows:

$$\boldsymbol{q} = \sum_{j=1}^{N} \boldsymbol{\phi}_j \eta_j, \tag{A.4}$$

where it is recalled that N is the size of the system. Here, η_j is typically referred to as a set of modal coordinates. It is a common practice to mass-normalize the mode shape vecttors, resulting in

$$\begin{aligned} \boldsymbol{\phi}_i^{\mathrm{T}} \mathbf{M} \boldsymbol{\phi}_j &= \delta_{ij}, \\ \boldsymbol{\phi}_i^{\mathrm{T}} \mathbf{K} \boldsymbol{\phi}_j &= \delta_{ij} \omega_j^2, \end{aligned} \tag{A.5}$$

where δ_{ij} denotes the Kronecker delta. These orthogonality relations are used to decouple the coupled equations of motion (A.2). Using (A.5), the decoupled equations are represented in modal coordinates as

$$\ddot{\eta}_j + \omega_j^2 \eta_j = \boldsymbol{\phi}_j^{\mathrm{T}} \boldsymbol{f}, \quad j \in \{1, \dots, N\}. \tag{A.6}$$

An important practical point on the expansion method is related to the computation of the expansion vecttors. The computation of the mode shape vecttors that are used in the mode superposition methods can be an expensive task and, in practice, all the computational methods extract a limited number K of vecttors of the eigenvalue problem. The general idea of the expansion procedure is to keep the first K vecttors in the representation, that correspond to the lowest eigenfrequencies. This results in a truncation, namely,

$$\boldsymbol{q} = \sum_{j=1}^{K} \boldsymbol{\phi}_j \eta_j + \underbrace{\sum_{j_t=K+1}^{N} \boldsymbol{\phi}_{j_t} \eta_{j_t}}_{truncated}, \tag{A.7}$$

where the indices j and j_t represent the kept mode and the truncated mode indices, respectively.

Since the displacement is represented as a linear combination of K linearly independent vecttors, it can also be given in matrix notation, leading to the approximation

$$\boldsymbol{q} = \boldsymbol{\Phi} \boldsymbol{\eta}, \qquad \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \dots & \boldsymbol{\phi}_K \end{bmatrix}. \tag{A.8}$$

Using (A.2) and (A.8) and projecting the resulting equations of motion on the expansion basis $\boldsymbol{\Phi}$ results in the following reduced-order dynamics

$$\mathbf{M}_r \ddot{\boldsymbol{\eta}} + \mathbf{K}_r \boldsymbol{\eta} = \boldsymbol{f}_r, \tag{A.9}$$

where

$$\mathbf{M}_r = \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{M} \boldsymbol{\Phi} = \mathbf{I}, \tag{A.10}$$

$$\mathbf{K}_r = \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{K} \boldsymbol{\Phi} = \mathrm{diag}\{\omega_1^2, \dots, \omega_K^2\}, \tag{A.11}$$

$$\boldsymbol{f}_r = \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{f}. \tag{A.12}$$

In general, the analysts are interested in the response properties of the system for the lower frequency range and therefore, the lowest modes are typically chosen. The reason behind this selection is the fact that most structures are operated at low frequencies.

The importance of a mode is mostly related to two concepts. First, the orthogonality of the mode with respect to the excitation, as given by $\boldsymbol{\phi}_j^{\mathrm{T}} \boldsymbol{f}$, is of importance. Secondly, the closeness of the eigenfrequency of the mode with respect to the excitation spectrum is of interest.

Mode acceleration method

The mode acceleration method is a computational variant of the static correction method. The static correction method aims at taking into account the contribution of the omitted modes. The driving idea of the static correction concept is to be able to include the effects of the truncated modes statically into the summation procedure. Namely, truncated modes have a static contribution on the response for low frequencies. This results in an improvement for the response studies in the lower frequency range. The response might be represented as before but with a correction term $\boldsymbol{q}_{cor}$, such that

$$\boldsymbol{q} = \boldsymbol{\Phi}\boldsymbol{\eta} + \boldsymbol{q}_{cor}. \tag{A.13}$$

To obtain the static correction term $\boldsymbol{q}_{cor}$ (with $\dot{\boldsymbol{q}}_{cor} = \ddot{\boldsymbol{q}}_{cor} = \boldsymbol{0}$), the truncated representation for the acceleration is substituted in the equation of motion (A.2), leading to

$$\mathbf{M}\sum_{j=1}^{K}\boldsymbol{\phi}_j\ddot{\eta}_j + \mathbf{K}\boldsymbol{q} = \boldsymbol{f}. \tag{A.14}$$

Then, the use of the (reduced-order) dynamics in modal coordinates (A.6) leads to

$$\begin{aligned} \boldsymbol{q} &= \mathbf{K}^{-1}\left(\boldsymbol{f} - \mathbf{M}\sum_{j=1}^{K}\boldsymbol{\phi}_j(\boldsymbol{\phi}_j^{\mathrm{T}}\boldsymbol{f} - \omega_j^2\eta_j)\right), \\ &= \sum_{j=1}^{K}\boldsymbol{\phi}_j\eta_j + \left(\mathbf{K}^{-1} - \sum_{j=1}^{K}\frac{\boldsymbol{\phi}_j\boldsymbol{\phi}_j^{\mathrm{T}}}{\omega_j^2}\right)\boldsymbol{f}, \end{aligned} \tag{A.15}$$

where the relation imposed by the eigenvalue problem (A.3) is used in the latter step. When comparing (A.15) to the truncation (A.7), it is observed that the correction term is given as

$$\boldsymbol{q}_{cor} = \left(\mathbf{K}^{-1} - \sum_{j=1}^{K}\frac{\boldsymbol{\phi}_j\boldsymbol{\phi}_j^{\mathrm{T}}}{\omega_j^2}\right)\boldsymbol{f}. \tag{A.16}$$

It is noted that, by using all eigenmodes, the inverse of the stiffness matrix can be represented as [63]

$$\mathbf{K}^{-1} = \sum_{j=1}^{N}\frac{\boldsymbol{\phi}_j\boldsymbol{\phi}_j^{\mathrm{T}}}{\omega_j^2}, \tag{A.17}$$

such that the use of (A.17) in (A.15) results in

$$\boldsymbol{q} = \sum_{j=1}^{K}\boldsymbol{\phi}_j\eta_j + \sum_{j=K+1}^{N}\frac{\boldsymbol{\phi}_j\boldsymbol{\phi}_j^{\mathrm{T}}}{\omega_j^2}\boldsymbol{f}. \tag{A.18}$$

Even though this last form is not applicable in practice, since it requires the computation of all model vecttors, it clearly shows that only the static contribution of the omitted modes $\boldsymbol{\phi}_j$, $j \in \{K+1, \ldots, N\}$, is taken into account in the correction term $\boldsymbol{q}_{cor}$.

Modal truncation augmentation method

The modal truncation augmentation method is an extension of the mode acceleration method. Its main principle depends on the use of the static correction as an additional direction for the truncation expansion [42, 128, 41]. Inclusion of the correction in a modal expansion results in the modal truncation augmentation method, such that $\boldsymbol{q}$ is approximated as

$$\boldsymbol{q} = \sum_{j=1}^{K} \boldsymbol{\phi}_j \eta_j + \boldsymbol{q}_{cor} \xi, \tag{A.19}$$

where $\boldsymbol{q}_{cor}$ is given by the mode acceleration method in (A.18) and ξ is an additional coordinate in the reduced-order system. This correction vecttor is included in the reduction basis, such that the new reduction basis reads

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{q}_{cor} \end{bmatrix}. \tag{A.20}$$

Here, it is noted that $\boldsymbol{\Psi}$ is generally $\mathbf{M}$-orthogonalized.

Modal truncation augmentation methods are mostly used when there are multiple forcing vecttors acting on the system. Therefore, these correction vecttors are not used a posteriori as in the mode acceleration method but they really become a part of the reduction space.

There exist also further extensions of the common mode superposition methods which include higher-order correction vecttors. These methods are outlined in [128] and references therein. Further details can be found in Section A.3.3.

A.2.2. Krylov subspace based model order reduction

Krylov subspace based model order reduction (MOR) methods are methods which reduce a system with many degrees of freedom (i.e. states) to a system with few(er) degrees of freedom but with similar *input-output behavior*. Typical applications are large electronic circuits with large linear subnetworks of components (see e.g. [113, 59]) and micro-electro-mechanical systems (MEMS). For an application in structural vibrations, see e.g. [95]. The main purpose of Krylov methods is the construction of an approximation of the system's transfer function which (accurately enough) describes the dependence between the input and the output of the original system, e.g. in some range of the (input) frequency domain. In particular, a so-called moment expansion of the transfer function is considered and reduction focusses on matching the first coefficient (moments) of this expansion. As this moment matching procedure is related to projections on a Krylov subspace, it provides a computationally cheap approach to model order reduction. Moreover, parallel computing can be effectively exploited. The objective is the derivation of a smaller system with similar input-output behavior and with similar properties such as stability, passivity or a special structure of the matrices in the model description. The quality of the reduced-order approximation can be assessed by studying norms of the difference between the outputs of the unreduced and reduced models applied for the same inputs. Preservation of additional properties is of importance if the reduced system has to exhibit some physical properties of the model; for instance, when the reduced system has to be a (realizable) circuit consisting out of resistors, inductors and capacitors (a RLC network), just as the original system. So far, there have been no proven a priori error-bounds for the Krylov based reduction techniques, see [73] for more details and for alternative approaches to ensure a good (application domain dependent) approximation.

The first reduction method involving the usage of the Krylov subspace, called asymptotic waveform evaluation, was described in 1990, see [124]. However, the main focus of this paper was on finding a Padé approximation of the transfer function rather than on the construction of a Krylov subspace. Later, in 1995, in [50] a method called Padé via Lanczos (PVL) was proposed and the relation between the Padé approximation and Krylov subspace was shown. In 1998, a new reduction technique, PRIMA, was introduced in [113], that uses the Arnoldi algorithm instead of Lanczos to build the reduction bases. These and later developments of Krylov based reduction techniques focus not only on the improvement of the accuracy of the approximation, but also on the preservation of the properties of the system to be reduced.

In this section, the basic ideas of model reduction by projection onto the Krylov subspace are explained and the application of some common reduction techniques based on Arnoldi and Lanczos algorithms (see e.g. [134] for more details) is briefly discussed. Notation-wise, $\mathbb{R}$ represents the field of real numbers, whereas $\mathbb{C}$ represents the field of complex numbers.

Linear time-invariant state-space systems of the form

$$\left\{ \begin{array}{rcl} \mathbf{E}\dot{\boldsymbol{x}} & = & \mathbf{A}\boldsymbol{x} + \boldsymbol{b}u \\ y & = & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}, \end{array} \right. \tag{A.21}$$

are considered, with $\mathbf{E}, \mathbf{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^n$, the input variable $u \in \mathbb{R}$, the output variable $y \in \mathbb{R}$ and $\boldsymbol{x} \in \mathbb{R}^n$ being a vector of the state variables. For the sake of simplicity, SISO systems (with scalar input and scalar output) are considered. However, the methods discussed in this section have been extended to multi-input-multi-output (MIMO) cases (see e.g. [57]).

If the system (A.21) is transformed to the Laplace domain, then, for an arbitrary $s \in \mathbb{C}$, the dependence between its input and its output is given by a transfer function $H(s)$ defined as follows

$$H(s) = \boldsymbol{c}^{\mathrm{T}}(s\mathbf{E} - \mathbf{A})^{-1}\boldsymbol{b}. \tag{A.22}$$

In this section, it is assumed that the pencil $(s\mathbf{E} - \mathbf{A})$ is regular, i.e. it is singular only for a finite number of $s \in \mathbb{C}$. For an arbitrary $s_0 \in \mathbb{C}$, the transfer function

(A.22) may be rewritten in a polynomial form, using a so-called moment expansion:

$$H(s) = \sum_{n=0}^{\infty} (-1)^n M_n(s_0)(s - s_0)^n. \tag{A.23}$$

Here, the coefficients $M_n(s_0)$, called moments of the transfer function, are calculated using the Taylor expansion formula and given by

$$M_n(s_0) = \boldsymbol{c}^{\mathrm{T}}[(s_0\mathbf{E} - \mathbf{A})^{-1}\mathbf{E}]^n (s_0\mathbf{E} - \mathbf{A})^{-1}\boldsymbol{b}. \tag{A.24}$$

Expansion around $s_0 = \infty$ is evaluated based on Laurent series, and the moments then are called Markov parameters. These Markov parameters play an important role in systems theory, see e.g. [8]. The accuracy of the moment expansion depends on the choice of the expansion point s_0. It is also possible to use a multipoint expansion choosing multiple expansion points.

The goal of the Krylov subspace model order reduction is to find a projection-based approximation of the original transfer function, that matches the first k moments of the original transfer function. In other words, the objective is to calculate a reduced-order system with transfer function $\hat{H}(s)$, whose moment expansion is given by

$$\hat{H}(s) = \sum_{n=0}^{\infty} (-1)^n \hat{M}_n(s_0)(s - s_0)^n, \tag{A.25}$$

with

$$\hat{M}_n(s_0) = M_n(s_0), \qquad \text{for} \quad n = 1, \dots, k, \tag{A.26}$$

and $M_n(s_0)$ being the moments of the original transfer function defined in (A.24). This is called the moment matching property of the reduction method.

In case of the reduction methods studied in this section, the reduced-order model is calculated using a projection $\mathbf{\Pi} = \mathbf{V}\mathbf{W}^{\mathrm{T}} \in \mathbb{R}^{n \times n}$, with $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times k}$ being biorthogonal matrices, i.e. $\mathbf{W}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$. Application of the projection $\mathbf{\Pi}$ to the original system (A.21) gives

$$\left\{ \begin{array}{rcl} \mathbf{W}^{\mathrm{T}}\mathbf{E}\mathbf{V}\dot{\hat{\boldsymbol{x}}} & = & \mathbf{W}^{\mathrm{T}}\mathbf{A}\mathbf{V}\hat{\boldsymbol{x}} + \mathbf{W}^{\mathrm{T}}\boldsymbol{b}u, \\ \hat{y} & = & \boldsymbol{c}^{\mathrm{T}}\mathbf{V}\hat{\boldsymbol{x}}, \end{array} \right. \tag{A.27}$$

where the reduced-order state vecttor $\hat{\boldsymbol{x}} \in \mathbb{R}^k$ results from the state transformation

$$\boldsymbol{x} \approx \mathbf{V}\hat{\boldsymbol{x}}. \tag{A.28}$$

The choice of the spaces $\mathbf{V}$ and $\mathbf{W}$ depends on the goal of the reduction procedure. In case of the Krylov subspace based methods, the aim is to approximate the input-output behavior of the system. This is done by matching the moments of the original transfer function. This means that the reduced-order transfer function corresponding to system (A.27), which results from applying matrices $\mathbf{V}$ and $\mathbf{W}$ to the original system matrices, has the property (A.26). To ensure the satisfaction of the moment matching property (A.26), one can choose $\mathbf{V}$ and $\mathbf{W}$ such that the columns of these matrices span so-called Krylov subspaces. The k-th Krylov subspace induced by a matrix $\mathbf{P}$ and a vecttor $\boldsymbol{r}$ is defined as

$$\mathcal{K}_k(\mathbf{P}, \boldsymbol{r}) = \mathrm{span}\{\boldsymbol{r}, \mathbf{P}\boldsymbol{r}, \ldots, \mathbf{P}^{k-1}\boldsymbol{r}\}. \tag{A.29}$$

The choice of the generating matrix $\mathbf{P}$ and the starting vecttor $\boldsymbol{r}$ depends on the value s_0 around which the transfer function should be approximated. If the approximation of the transfer function (A.22) around $s_0 = 0$ is to be found, the matrices $\mathbf{V}$ and $\mathbf{W}$ are chosen as follows:

$$\mathbf{V} \quad \text{is a basis of} \quad \mathcal{K}_{k_1}(\mathbf{A}^{-1}\mathbf{E}, \mathbf{A}^{-1}\boldsymbol{b}), \tag{A.30}$$

$$\mathbf{W} \quad \text{is a basis of} \quad \mathcal{K}_{k_2}(\mathbf{A}^{-\mathrm{T}}\mathbf{E}^{\mathrm{T}}, \mathbf{A}^{-\mathrm{T}}\boldsymbol{c}). \tag{A.31}$$

The sizes of the subspaces, k_1 and k_2, should assure that $\mathbf{V}$ and $\mathbf{W}$ are both of rank k. If $\mathbf{V}$ and $\mathbf{W}$ are built in the way defined in (A.30-A.31), the model reduction method is called a two-sided method. If only one of the projection matrices ($\mathbf{V}$ or $\mathbf{W}$) is built in that way, the method is called one-sided. Application of the two-sided method results in a reduced model that matches the first $2k$ moments of the original transfer function. In case of one-sided methods, k moments are matched.

The general proof of the moment matching property can be found in [67]. To illustrate the idea behind this proof, the matching of the zeroth moment of the system (A.21) for $s_0 = 0$ is shown following [98]. According to the formula (A.24), the zeroth moment for $s_0 = 0$ is equal to

$$M_0(0) = -\boldsymbol{c}^{\mathrm{T}}\mathbf{A}^{-1}\boldsymbol{b}. \tag{A.32}$$

With $\mathbf{V}$ chosen as in (A.30) and the fact that $\mathbf{A}^{-1}\boldsymbol{b}$ belongs to the Krylov subspace $\mathcal{K}_{k_1}(\mathbf{A}^{-1}\mathbf{E}, \mathbf{A}^{-1}\boldsymbol{b})$, one can find a vecttor $\boldsymbol{r}_0$ such that $\mathbf{V}\boldsymbol{r}_0 = \mathbf{A}^{-1}\boldsymbol{b}$. Then, using the reduction procedure defined in (A.27), it can be shown that

$$\begin{aligned}\hat{M}_0(0) = -\boldsymbol{c}^{\mathrm{T}}\mathbf{V}(\mathbf{W}^{\mathrm{T}}\mathbf{A}\mathbf{V})^{-1}\mathbf{W}^{\mathrm{T}}\boldsymbol{b} &= -\boldsymbol{c}^{\mathrm{T}}\mathbf{V}(\mathbf{W}^{\mathrm{T}}\mathbf{A}\mathbf{V})^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{A}\mathbf{V}\boldsymbol{r}_0 \\ &= -\boldsymbol{c}^{\mathrm{T}}\mathbf{V}\boldsymbol{r}_0 = -\boldsymbol{c}^{\mathrm{T}}\mathbf{A}^{-1}\boldsymbol{b} = M_0(0).\end{aligned} \tag{A.33}$$

In case the approximation around $s_0 \neq 0$ or for $s_0 = \infty$ is needed, the starting matrix and vecttor for building the Krylov subspace have to be modified. One can also build a subspace using different values of s_0 at the same time. More details on how to do this and suggestions for starting values for different s_0 can be found in [67].

Besides the difference in the number of moments matched, the choice to use either one- or two-sided methods influences also some other properties of the reduced system. Two-sided methods may lead to better approximations of the output y and deliver a reduced-order model, whose input-output behavior does not depend on the state space realization of the original model. In case of the one-sided techniques with $\mathbf{W} = \mathbf{V}$ and $\mathbf{V}$ defined as in (A.30), for certain original models, one can also prove the preservation of the passivity property. For general models however, it is in general unclear whether matching twice as many moments justifies the additional computational cost of using two-sided methods. Namely, moments only characterize the transfer function in a small neighborhood of the expansion point and do generally not capture global behavior.

The process of constructing the reduction matrices, **V** and **W**, is not straightforward and requires the use of special techniques. Because of round-off errors, the vecttors building a Krylov subspace may quickly become linearly dependent. To avoid this problem, one usually constructs an orthogonal basis of the appropriate Krylov subspace. This can be achieved using e.g. Arnoldi or Lanczos algorithms (explanation of these algorithms and implementation details are given in [134]). The classical Arnoldi algorithm generates a set **V** of orthonormal vecttors, i.e.

$$\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}, \tag{A.34}$$

that form a basis for a given Krylov subspace. The Lanczos algorithm finds two sets of basis vecttors, **V** and **W**, that span an appropriate Krylov subspace and have property

$$\mathbf{W}^{\mathrm{T}}\mathbf{V} = \mathbf{I}. \tag{A.35}$$

Two sets of basis vecttors **V** and **W** for Krylov subspaces may also be computed using a two-sided Arnoldi algorithm (see [98]). In this case, both **V** and **W** are orthonormal,

$$\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}, \qquad \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}. \tag{A.36}$$

As a result, each of the above mentioned techniques generates a Krylov subspace. The choice of the subspace depends on the type of algorithm and the expansion point s_0 around which the approximation is of interest. A more detailed explanation on how to choose the proper subspaces can be found in [67].

The ideas of the Krylov subspace based reduction presented in this section can be further modified, depending on e.g. the application or the specific criteria that the reduced-order model should fulfill. In electronic circuit design, there exist methods especially suited for reducing specific types of systems that exploit the characteristic structure of the underlying matrices, see e.g. [13]. In case of coupled or interconnected systems, the goal may be to preserve the interpretation of the different physical domains. More details on this topic can be found in [59, 155]. There exist also modifications that aim at preserving other properties of the original system, such as stability or passivity. In case of symmetric matrices, the algorithm SyPVL was proposed in [60] that guarantees stability. A stability and passivity preserving technique, PRIMA, is presented in [113].

A.2.3. Balanced truncation

The field of systems and control focusses on the analysis of dynamical systems and design of feedback controllers for these systems. Herein, the objective of controller design is to change the dynamics of the system to induce desired behavior. Typical examples are the stabilization of unstable systems, tracking of a reference trajectory or the rejection of external disturbances on a system.

These control strategies are applied in a broad range of practical engineering problems, such as control of mechanical or electrical systems. These applications have in common that they deal with systems with inputs and outputs. Namely, a dynamical system can often only be influenced by a limited number of actuators, which are represented as inputs, and only a limited number of sensors (i.e. outputs) is typically available in practical engineering systems. For these systems, it is thus particularly relevant to have an accurate model for their input-output behavior. Even though this model does not need to describe the global behavior of a system, complex dynamics can still yield large models of orders up to $\mathcal{O}(10^3)$. To facilitate controller design and/or analysis for these systems, model reduction is needed. Here, it is noted that a controller needs to be implemented in real-time, which also requires a controller realization of relatively low-order. Moreover, the use of these low-order models as substitutes for the original model requires very accurate model reduction techniques, where the preservation of relevant systems properties such as stability is of great importance.

Model reduction procedures in the field of systems and control thus aim at approximating the *input-output behavior* of a high-order model. The quality of the reduced-order model can thus be assessed by comparing the outputs of the high-order and reduced-order models for given inputs, where the magnitude of the output error is measured using some signal norm.

Balanced truncation is the most popular method in systems and control addressing this model reduction problem. It mainly owes its popularity due to the fact that it preserves stability of the high-order model and provides an error bound, which gives a direct measure of the quality of the reduced-order model. The balanced truncation method was first presented by Moore [104], where results of Mullis and Robberts [108] were exploited. Later, the stability preservation property was found by Pernebo and Silverman [121], whereas the error bound was derived by Enns [44] and Glover [65].

Linear dynamical models with inputs and outputs in state-space form

$$\left\{ \begin{array}{rcl} \dot{\boldsymbol{x}} & = & \mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{u} \\ \boldsymbol{y} & = & \mathbf{C}\boldsymbol{x} + \mathbf{D}\boldsymbol{u} \end{array} \right. \tag{A.37}$$

are considered. Here, $\boldsymbol{u} \in \mathbb{R}^m$ denotes the input whereas $\boldsymbol{y} \in \mathbb{R}^p$ represents the output. The internal state is given by $\boldsymbol{x} \in \mathbb{R}^n$ and the system matrices are of corresponding dimensions. Throughout this section, it is assumed that the model (A.37) is asymptotically stable (i.e. all eigenvalues of $\mathbf{A}$ have negative real part) and is a minimal realization, where the latter guarantees that all state components contribute to the input-output behavior. The transfer function of (A.37) is given as

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}, \quad s \in \mathbb{C}. \tag{A.38}$$

In balanced truncation, a reduced-order model is obtained in two steps. First, a so-called balanced realization is found, in which the states are ordered according to their contribution to the input-output behavior. Second, a reduced-order model is obtained on the basis of this balanced realization by discarding the states with the smallest influence.

In order to find the balanced realization, the input-output behavior of the system (A.37) has to be quantified. To this end, the so-called controllability and observability functions are defined. First, the controllability function $\mathcal{E}_c(\boldsymbol{x}_0)$ gives the smallest input energy required to reach the state $\boldsymbol{x}_0$ from the zero state in infinite time, given as

$$\mathcal{E}_c(\boldsymbol{x}_0) = \inf_{\substack{\boldsymbol{u}\in\mathcal{L}_2(-\infty,0)\\ \boldsymbol{x}(-\infty)=\mathbf{0},\ \boldsymbol{x}(0)=\boldsymbol{x}_0}} \int_{-\infty}^{0} \|\boldsymbol{u}(t)\|^2\, dt, \tag{A.39}$$

where $\mathcal{L}_2(-\infty,0)$ denotes the space of square integrable functions, defined on the domain $(-\infty,0)$. Second, the observability function $\mathcal{E}_o(\boldsymbol{x}_0)$ is defined by

$$\mathcal{E}_o(\boldsymbol{x}_0) = \int_{0}^{\infty} \|\boldsymbol{y}(t)\|^2\, dt, \quad \boldsymbol{x}(0)=\boldsymbol{x}_0,\ \boldsymbol{u}(t)=\mathbf{0}\ \ \forall t\in[0,\infty), \tag{A.40}$$

and gives the future output energy of the system when released from an initial condition $\boldsymbol{x}_0$ for zero input. It is well-known (see e.g. [104, 162]) that for linear systems as in (A.37) the controllability and observability functions in (A.39) and (A.43) can be written as the quadratic forms

$$\mathcal{E}_c(\boldsymbol{x}_0) = \boldsymbol{x}_0^{\mathrm{T}}\mathbf{P}^{-1}\boldsymbol{x}_0, \qquad \mathcal{E}_o(\boldsymbol{x}_0) = \boldsymbol{x}_0^{\mathrm{T}}\mathbf{Q}\boldsymbol{x}_0, \tag{A.41}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are the controllability and observability Gramian, given by

$$\mathbf{P} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^{\mathrm{T}}e^{\mathbf{A}^{\mathrm{T}}t}\,\mathrm{d}t \tag{A.42}$$

and

$$\mathbf{Q} = \int_0^\infty e^{\mathbf{A}^{\mathrm{T}}t}\mathbf{C}^{\mathrm{T}}\mathbf{C}e^{\mathbf{A}t}\,\mathrm{d}t, \tag{A.43}$$

respectively. From (A.42) and (A.43), it is easily observed that the controllability and observability Gramians are only finite when the system is asymptotically stable, which explains the assumption stated before. In addition, $\mathbf{P}$ and $\mathbf{Q}$ are symmetric and positive definite, where the latter is guaranteed by the assumption that the system (A.37) is minimal, i.e. controllable and observable. The controllability and observability Gramian can be obtained as the unique positive definite solutions of the respective Lyapunov equations (see e.g. [162])

$$\mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^{\mathrm{T}} + \mathbf{B}\mathbf{B}^{\mathrm{T}} = \mathbf{0} \tag{A.44}$$

and

$$\mathbf{A}^{\mathrm{T}}\mathbf{Q} + \mathbf{Q}\mathbf{A} + \mathbf{C}^{\mathrm{T}}\mathbf{C} = \mathbf{0}, \tag{A.45}$$

which makes balanced truncation computationally feasible. Nonetheless, solving the Lyapunov equations is computationally costly, such that balanced truncation is limited to systems of orders up to $\mathcal{O}(10^3)$.

Since both the controllability and observability Gramian characterize the in- or output energy associated to a state $\boldsymbol{x}_0$, they are dependent on the realization of the system (A.37). Stated differently, a change of coordinates $\bar{\boldsymbol{x}} = \mathbf{T}\boldsymbol{x}$, with $\mathbf{T} \in \mathbb{R}^{n\times n}$ a nonsingular matrix, results in a realization with system matrices

$$\bar{\mathbf{A}} = \mathbf{TAT}^{-1}, \quad \bar{\mathbf{B}} = \mathbf{TB}, \quad \bar{\mathbf{C}} = \mathbf{CT}^{-1}, \quad \bar{\mathbf{D}} = \mathbf{D}. \tag{A.46}$$

Then, the new controllability and observability Gramians are given as

$$\bar{\mathbf{P}} = \mathbf{TPT}^{\mathrm{T}}, \qquad \bar{\mathbf{Q}} = \mathbf{T}^{-\mathrm{T}}\mathbf{QT}^{-1}. \tag{A.47}$$

Nonetheless, the product of $\bar{\mathbf{P}}$ and $\bar{\mathbf{Q}}$ yields

$$\bar{\mathbf{P}}\bar{\mathbf{Q}} = \mathbf{TPQT}^{-1}, \tag{A.48}$$

indicating that the eigenvalues of the product of the controllability and observability Gramian are independent of the set of coordinates and thus system invariants. These eigenvalues equal the (squared) Hankel singular values σ_i [65], such that

$$\sigma_i = \sqrt{\lambda_i(\mathbf{PQ})}, \quad i = 1, \ldots, n, \tag{A.49}$$

where $\lambda_i(\mathbf{X})$ denotes the i-th eigenvalue of the matrix $\mathbf{X}$, ordered as $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n > 0$.

At this point, it is recalled that the observability Gramian $\mathbf{Q}$ characterizes the output energy associated to a given initial state $\boldsymbol{x}_0$ and thus provides a measure of the importance of state components with respect to the output $\boldsymbol{y}$. Hence, states generating high output energy can be considered more important than states generating little output energy, since the former are easy to observe. On the other hand, the controllability Gramian $\mathbf{P}$ gives a measure of the importance of state components $\boldsymbol{x}_0$ with respect to the input $\boldsymbol{u}$, in the sense that states that require little input energy to reach are more relevant than states that require high input energy. States that require little energy to reach are thus easy to control. Clearly, the combination of the controllability and observability Gramians gives a characterization of the importance of state components from an input-output perspective. However, in an arbitrary coordinate system, a state $\bar{\boldsymbol{x}}_0^1$ that requires little energy to reach might also generate little output energy. On the other hand, a different state $\bar{\boldsymbol{x}}_0^2$ might exist that requires a lot of energy to reach, but generates high output energy. In this case, it is not easy to decide which of $\bar{\boldsymbol{x}}_0^1$ and $\bar{\boldsymbol{x}}_0^2$ is the most important component from an input-output perspective. To facilitate this analysis, the balanced realization is introduced. Namely, there exists a state-space realization such that the corresponding controllability and observability Gramians are equal and diagonal, where the entries on the diagonal are given by the Hankel singular values [104]:

$$\bar{\mathbf{P}} = \bar{\mathbf{Q}} = \mathbf{\Sigma} := \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix}. \tag{A.50}$$

In addition, the Hankel singular values are ordered as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$. In this balanced realization, the controllability and observability function are given as

$$\mathcal{E}_c(\bar{\boldsymbol{x}}_0) = \bar{\boldsymbol{x}}_0^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \bar{\boldsymbol{x}}_0, \qquad \mathcal{E}_o(\bar{\boldsymbol{x}}_0) = \bar{\boldsymbol{x}}_0^{\mathrm{T}} \boldsymbol{\Sigma} \bar{\boldsymbol{x}}_0. \tag{A.51}$$

Now, the form (A.51) allows for a clear interpretation. Namely, the realization is balanced in the sense that states that are easy to control are also easily observed. In fact, due to the ordering of the Hankel singular values, the state $\bar{\boldsymbol{x}}_0 = \boldsymbol{e}_1 := [1, 0, \cdots, 0]^{\mathrm{T}}$ requires the least energy to reach ($\mathcal{E}_c(\boldsymbol{e}_1) = \sigma_1^{-1}$ is small) and gives the highest output energy ($\mathcal{E}_o(\boldsymbol{e}_1) = \sigma_1$ is large). Stated differently, this state is easy to control and easy to observe. Hence, $\bar{\boldsymbol{x}}_0 = \boldsymbol{e}_1$ has the largest contribution to the input-output behavior of the system. On the other hand, the state $\bar{\boldsymbol{x}}_0 = \boldsymbol{e}_n := [0, \cdots, 0, 1]^{\mathrm{T}}$ is both difficult to control and difficult to observe, such that it has the smallest contribution to the input-output behavior.

The coordinate transformation $\mathbf{T}$ to obtain the balanced realization can be obtained on the basis of the controllability and observability Gramians (A.42-A.43). Thereto, the Cholesky factor $\mathbf{U}$ of $\mathbf{P}$ is used, as well as the eigenvalue decomposition of $\mathbf{U}^{\mathrm{T}}\mathbf{Q}\mathbf{U}$:

$$\mathbf{P} = \mathbf{U}\mathbf{U}^{\mathrm{T}}, \qquad \mathbf{U}^{\mathrm{T}}\mathbf{Q}\mathbf{U} = \mathbf{K}\mathbf{S}\mathbf{K}^{\mathrm{T}}. \tag{A.52}$$

In the latter, it is noted that $\mathbf{U}^{\mathrm{T}}\mathbf{Q}\mathbf{U}$ is a positive definite symmetric matrix, such that the matrix of eigenvecttors $\mathbf{K}$ is orthonormal. Additionally, the eigenvalues are real and, when ordered, are equal to the squared Hankel singular values such that $\mathbf{S} = \boldsymbol{\Sigma}^2$ with $\boldsymbol{\Sigma}$ as in (A.50). Then, the balancing transformation and its inverse are given as

$$\mathbf{T} = \boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{K}^{\mathrm{T}}\mathbf{U}^{-1}, \qquad \mathbf{T}^{-1} = \mathbf{U}\mathbf{K}\boldsymbol{\Sigma}^{-\frac{1}{2}} \tag{A.53}$$

as can be checked by substitution of (A.53) in (A.47), while using the relations (A.52). An overview of alternative algorithms to obtain the balanced realization can be found in [8].

So far, a balanced realization is found, but no model reduction has been performed yet. However, the balanced realization gives a representation in which the states are ordered according to their contribution to the input-output behavior. Hence, a reduced-order model of order k can be obtained by partitioning the state $\bar{\boldsymbol{x}}$ of the balanced realization as $\bar{\boldsymbol{x}}^1 = [\bar{x}_1, \ldots, \bar{x}_k]^{\mathrm{T}} \in \mathbb{R}^k$ and $\bar{\boldsymbol{x}}^2 = [\bar{x}_{k+1}, \ldots, \bar{x}_n]^{\mathrm{T}} \in \mathbb{R}^{n-k}$, such that $\bar{\boldsymbol{x}}^1$ contains the state components with the largest influence on the input-output behavior. When the system matrices are partitioned accordingly,

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_2 \end{bmatrix}, \quad \bar{\mathbf{A}} = \begin{bmatrix} \bar{\mathbf{A}}_{11} & \bar{\mathbf{A}}_{12} \\ \bar{\mathbf{A}}_{21} & \bar{\mathbf{A}}_{22} \end{bmatrix}, \quad \bar{\mathbf{B}} = \begin{bmatrix} \bar{\mathbf{B}}_1 \\ \bar{\mathbf{B}}_2 \end{bmatrix}, \quad \bar{\mathbf{C}} = \begin{bmatrix} \bar{\mathbf{C}}_1 & \bar{\mathbf{C}}_2 \end{bmatrix}, \quad \bar{\mathbf{D}} = \mathbf{D}, \tag{A.54}$$

a reduced-order system can be obtained by truncation, i.e. by setting $\bar{\boldsymbol{x}}_2 = \mathbf{0}$. The resulting reduced-order model (with $\hat{\boldsymbol{x}} \in \mathbb{R}^k$ an approximation of $\bar{\boldsymbol{x}}^1 \in \mathbb{R}^k$) is given by the state-space realization

$$\left\{\begin{array}{lcl}\dot{\hat{\boldsymbol{x}}} & = & \bar{\mathbf{A}}_{11}\hat{\boldsymbol{x}} + \bar{\mathbf{B}}_1\boldsymbol{u}, \\ \hat{\boldsymbol{y}} & = & \bar{\mathbf{C}}_1\hat{\boldsymbol{x}} + \bar{\mathbf{D}}\boldsymbol{u}.\end{array}\right. \tag{A.55}$$

Here, it can be observed that the reduced-order state-space system (A.55) is itself a balanced realization, with the controllability and observability Gramians given by $\mathbf{\Sigma}_1$ (see [121]). In addition, when $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ have no diagonal entries in common (i.e. when $\sigma_k > \sigma_{k+1}$), the reduced-order system is asymptotically stable [121].

The reduced-order system thus preserves stability of the original model, and its output $\hat{\boldsymbol{y}}$ serves as an approximation for the output of the high-order system $\boldsymbol{y}$. The quality of this approximation can be assessed by means of a bound on the error. Namely, an error bound can be expressed in terms of the discarded Hankel singular values [44, 65] as

$$\|\mathbf{H}(s) - \hat{\mathbf{H}}(s)\|_\infty \leq 2 \sum_{i=k+1}^{n} \sigma_i, \tag{A.56}$$

where $\mathbf{H}(s)$ and $\hat{\mathbf{H}}(s)$ are the transfer functions of the full-order system (A.37) and the reduced-order system (A.55), respectively. Furthermore, $\|\cdot\|_\infty$ denotes the $\mathcal{H}_\infty$ norm defined as

$$\|\mathbf{H}(s)\|_\infty = \sup_{\omega\in\mathbb{R}} \bar{\sigma}(\mathbf{H}(j\omega)), \tag{A.57}$$

with $\bar{\sigma}(\cdot)$ the largest singular value. The error bound (A.56) confirms the intuitive idea that the states corresponding to the largest Hankel singular values are the most important from the perspective of input-output behavior. Namely, a good approximation (i.e. a low error bound) will be obtained when the Hankel singular values in $\mathbf{\Sigma}_2$ are small. Since these Hankel singular values are only dependent on the high-order model (A.37), they can be computed a priori and allow for control over the reduction error by selection of the order k. Finally, it is noted that in (A.56) it is assumed that all Hankel singular values are distinct. When Hankel singular values with multiplicity larger than one occur, they only need to be counted once, leading to a tighter bound (see e.g. [65]).

In the model reduction procedure presented here, a reduced-order system is obtained by truncation (i.e. setting $\bar{\boldsymbol{x}}^2 = 0$) of a balanced realization. An alternative approach is given by singular perturbation [54] of this realization. Herein, it is assumed that the dynamics describing the evolution of $\bar{\boldsymbol{x}}^2$ is very fast (and asymptotically stable). Then, this dynamics can be assumed to be in its equilibrium position at all time, which is obtained by setting $\dot{\bar{\boldsymbol{x}}}^2 = 0$ and solving for $\bar{\boldsymbol{x}}^2$ as a function of $\bar{\boldsymbol{x}}^1$ and $\boldsymbol{u}$. Contrary to balanced truncation, the singular perturbation approach guarantees that the steady-state gains of the high-order system are matched in the reduced-order system. The reduced-order model is controllable, observable, asymptotically stable and the error bound (A.56) also holds [96].

Balanced truncation aims at approximating a high-order system by selecting the state components that have the largest contribution in the input-output behavior, according to the energy in the input and output signals. The entire frequency range is considered in this approach. However, in many practical applications, a good approximation is only required in a specific frequency range. To this end, frequency-weighted balanced truncation can be used [44], which is an extension of the method discussed in the previous paragraphs. In frequency-weighted balanced truncation, the objective is to find a reduced-order system such that the error

$$\|\mathbf{H}_o(s)(\mathbf{H}(s) - \hat{\mathbf{H}}(s))\mathbf{H}_i(s)\|_\infty \tag{A.58}$$

is small, where $\mathbf{H}_i(s)$ and $\mathbf{H}_o(s)$ denote the transfer functions of an input and output frequency weight, respectively. These weights can be designed by the user to emphasize specific regions in the frequency domain. To obtain the frequency-weighted reduced-order model, controllability and observability Gramians are computed on the basis of the frequency weighted high-order system, which are simultaneously diagonalized. Details can be found in [44].

When the original system is asymptotically stable, observable and controllable, and only one-sided weighting is applied (i.e. either $\mathbf{H}_i(s) = \mathbf{I}$ or $\mathbf{H}_o(s) = \mathbf{I}$), asymptotic stability of the reduced-order system is guaranteed. However, in the case of general two-sided weighting, stability of the reduced-order approximant can not be guaranteed. Nonetheless, when the reduced-order model is stable, an error bound is given in [87].

In the preceding paragraphs, the standard balanced truncation technique for asymptotically stable systems as well as an extension to frequency-weighted balanced truncation is presented. Several extensions of balanced truncation exists. For example, balanced truncation of the coprime factorization applies to unstable systems [103, 112], whereas a method preserving passivity is given in [39, 111].

Besides these methods based on balanced truncation, a popular alternative is optimal Hankel norm approximation [65], which is also based on the balanced realization. For an overview of model reduction in systems and control, see e.g. [8, 69].

A.3. Qualitative comparison on model reduction methods

In this section, the methods as discussed in Section A.2 will be compared. First, the common feature of projection is presented in Section A.3.1. Then, a general comparison will be given in Section A.3.2. A close connection between moment matching and modal truncation augmentation is discussed in Section A.3.3. Computational aspects and the preservation of properties will be discussed in Sections A.3.4 and A.3.5, respectively.

A.3.1. Projection

Before discussing differences between the methods from the fields of structural dynamics, numerical mathematics and systems and control, an important similarity is discussed. Namely, the methods discussed in Section A.2 have in common that the reduced-order models are obtained by projection. Here, the reduced-order model is obtained by application of the projection $\mathbf{\Pi} = \boldsymbol{\mathcal{V}}\boldsymbol{\mathcal{W}}^{\mathrm{T}}$ to the original model. In numerical mathematics, the projection matrices might be chosen as $\boldsymbol{\mathcal{V}} = \mathbf{V}$ and $\boldsymbol{\mathcal{W}} = \mathbf{W}$ with $\mathbf{V}$ and $\mathbf{W}$ as in (A.30) and (A.31), respectively. This specific choice ensures moment matching around $s_0 = 0$. For balanced truncation, as used in systems and control, the matrices $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{W}}$ are given as the first k columns of the transformation matrices $\mathbf{T}^{-1}$ and $\mathbf{T}^{\mathrm{T}}$ as in (A.53), respectively. Hence, they project on the subspace of $\mathbf{PQ}$ corresponding to the largest Hankel singular values (see (A.48)), which yields the subspace with the largest contribution in the input-output behavior. Similarly, in the mode superposition techniques in structural dynamics, the projection is given as $\mathbf{\Pi} = \mathbf{M}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}$. Here, the projection basis $\mathbf{\Phi}$ forms a basis for the space spanned by the k most relevant eigenvecttors (see (A.3)), which are typically chosen as the eigenvecttors corresponding to the lowest eigenfrequencies. Here, it is noted that the state-space form is used in the fields of numerical mathematics and systems and control, whereas a second-order form is exploited in structural dynamics when no damping is present or when the damping can be considered as small. More generally, for systems with non-modal viscous damping, a similar reduction procedure can be used but then based on the complex modes of the non-conservative system (see e.g. [63]).

A.3.2. General comparison

Besides the common feature of projection, the reduction techniques in Section A.2 have important differences, as listed below.

- **First-order form versus second-order form**
  The most apparent distinction is the type of model under consideration. In the field of structural dynamics, models of second-order form are usually studied, whereas first-order models are examined in the fields of numerical mathematics and systems and control. Even though the use of this symmetric second-order form seems limiting, it is noted that the mechanical structures studied in this field can indeed be modeled as second-order systems. In addition, these mechanical structures typically have little damping, which motivates the use of undamped vibration modes for model reduction. Here, it is noted that the more general case of modal damping can be handled by such methods as well. Nonetheless, due to the specific structure of these models, the model reduction techniques from structural dynamics can in general not be applied to other application domains. On the other hand, any model that can be written in the first-order form can be handled by the reduction techniques from numerical mathematics and systems and control, although asymptotic stability is assumed in the latter.

- **Input-output behavior versus global behavior**
  A second difference is given by the objective of the approximation. In numerical mathematics and systems and control, a reduced-order model is sought which approximates the input-output behavior of the original system. On the other hand, this input-output behavior is of less relevance in the field of structural dynamics, where the approximation of the global dynamics is of

interest. Again, this results from the specific objectives in structural dynamics. Namely, typical interest is in the identification of the regions where the highest stresses or maximum displacements occur, whose locations are not known beforehand. Hence, the modeling of the global dynamical behavior is the main goal. However, extensions to mode superposition methods (see e.g. Section A.2.1) provide techniques of incorporating the (static) influence of input forces in the reduction basis, partially taking input-output behavior into account.

In numerical mathematics and systems and control, the internal behavior of the model is of little interest. In control design, the system behavior from the control input to the measured outputs is of relevance and this directly forms the basis for the model reduction procedure. In the analysis of large-scale electrical circuits, where moment matching methods from the field of numerical mathematics are typically applied, interest is in the reduction of linear subcircuits. Here, its influence on the total circuit is described by the inputs and outputs, such that the approximation of input-output behavior is of interest. Nonetheless, input-output behavior is only truly taken into account in the latter when two-sided projection techniques are used. Namely, when using one-sided projection techniques, either the input matrix $\mathbf{B}$ or output matrix $\mathbf{C}$ is discarded, such that the focus of the reduction is limited to the state-to-output or input-to-state behavior, respectively. In this case, the number of moments matched is independent of the choice of input or output matrix. Nonetheless, the number of moments matched for a given reduction order k is doubled in the two-sided case, when input-output behavior is fully taken into account.

- **Interpretation of reduction space**
  Model reduction techniques from structural dynamics are largely based on physical properties of mechanical systems. Therefore, the reduction space resulting from modal approximation has a useful engineering interpretation. Namely, it consists of the modes of the system, which represent the typical vibration pattern of a structure at a given eigenfrequency. The most important modes and the corresponding eigenfrequencies are preserved in the reduced-order system. Since these modes are obtained via an eigenvalue decomposition, the system in modal coordinates is in diagonal form, as discussed in Section A.2.1. Here, it is recalled that this only holds when the system is undamped or has proportional (Rayleigh) damping or modal damping. In this diagonal form, the equations describing the dynamics of the modes are uncoupled, which means that no error is introduced in the dynamics of the modes that are kept in the reduced-order model. In fact, the reduction error is due to the deletion of modes, rather than errors in the dynamics of the modes themselves.

  In the reduction techniques from numerical mathematics and systems and control, the reduction space does not have a clear physical interpretation. Of course, this is largely due to the fact that these procedures are not limited to mechanical systems and are based on system-theoretic properties instead, as discussed in Section A.2.3.

- **Global versus local approximation in frequency domain**
  The modal truncation and moment matching model reduction techniques from structural dynamics and numerical mathematics have in common that they can be considered as frequency-domain-based (or Laplace-domain-based) techniques. Therefore, they give a good approximation in some part of the frequency-domain only. This is directly apparent in the modal reduction techniques from structural dynamics, where the modes as used in the reduced-order model are selected by their corresponding eigenfrequency. Here, the modes are typically selected from the lower end of the frequency spectrum. On the other hand, moment matching in numerical mathematics is based on the Taylor series expansion of a transfer function at a specific point (or multiple points) in Laplace domain. Since the moments around this expansion point form the basis for the reduced-order model, this approximation can only be expected to be accurate around the expansion point, leading to a local approximation in frequency domain.

  In balanced truncation, as used in systems and control, the behavior in frequency domain does not form the basis of the model reduction procedure. Instead, the transfer of energy from the input to the output is used as a tool for model reduction, which can be considered as a time-domain approach. Nonetheless, specific regions in frequency domain can be emphasized by the extension to frequency-weighted balanced truncation, as noted in Section A.2.3. An important application of such techniques might be found in vibration problems, where the mesh is typically no longer valid for high frequencies.

- **Automatic versus user-dependent model reduction**
  A final general difference can be found in the level of automation of the model reduction techniques from the different fields. Here, only the balanced truncation method in systems and control is fully automatic when a requirement on the quality of the reduced-order model is given. Namely, the existence of an a priori error bound (A.56) allows for the automatic choice of the reduction order. The methods from structural dynamics and numerical mathematics lack such an error bound.

  Even when the reduction order is chosen beforehand, the methods from structural dynamics and numerical mathematics are heuristic. That is, the mode superposition techniques from structural dynamics are dependent on the frequency range of interest, which needs to be specified a priori. Herein, typically the modes corresponding to the lowest frequencies are chosen. Similarly, the reduction procedure in the moment matching techniques from numerical mathematics is dependent on the choice of expansion points. However, the influence of this choice on the properties of the reduced-order system is largely an open problem and very few guidelines for this selection exists. Therefore, these expansion points are typically chosen as $s_0 = 0$ or $s_0 = \infty$. Of course, the computational procedure in mode superposition and moment matching is fully automatic as soon as a choice is specified for the frequency range of interest and the expansion point, respectively.

A.3.3. Moment matching and model truncation augmentation

A close link exists between modal truncation augmentation techniques used in structural dynamics (Section A.2.1) and the moment matching methods (Section A.2.2). This can be understood by considering the series expansion of the (non-damped) structural equations (A.2) in the Laplace domain for s^2 as

$$\boldsymbol{q} = (\mathbf{K} + s^2\mathbf{M})^{-1}\boldsymbol{f} = \sum_{i=0}^{\infty} \left((\mathbf{K} + s_0^2\mathbf{M})^{-1}\mathbf{M}\right)^i (\mathbf{K} + s_0^2\mathbf{M})^{-1}\boldsymbol{f}(s^2 - s_0^2)^i, \quad \text{(A.59)}$$

where s_0 is a chosen expansion point. Clearly this expansion is similar to the moment matching expansion (A.23) except that here it is written for the second-order form. The reduction basis suggested by this expansion is the Krylov series

$$\mathcal{K}_k\left((\mathbf{K} + s_0^2\mathbf{M})^{-1}\mathbf{M}, (\mathbf{K} + s_0^2\mathbf{M})^{-1}\boldsymbol{f}\right) \quad \text{(A.60)}$$

In the modal truncation augmentation approach the reduction basis consists of some eigenmodes of the system and modal truncation corrections as described in (A.20). Recalling the definition (A.16) of the correction vecttors, it can be seen that the reduction basis (A.20) for the modal truncation augmentation is

$$\text{span}\{\begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{q}_{cor} \end{bmatrix}\} = \text{span}\{\begin{bmatrix} \boldsymbol{\Phi} & \mathbf{K}^{-1}\boldsymbol{f} \end{bmatrix}\}, \quad \text{(A.61)}$$

indicating that it includes the zero-order expansion term around $s_0 = 0$. Thus, it conserves the zero-order moment of the second-order problem around $s_0 = 0$, which is a direct consequence of the fact that the reduction basis includes the exact static solution. Through a similar reasoning one could say that substructuring methods that include the interface static modes (like the Craig-Bampton, the Rubin/MacNeal and the Dual Craig-Bamtpon methods) are matching the zero-order moments for the interface forces.

The modal truncation augmentation form presented in Section A.2.1 includes only the zeroth-order correction as indicated by the basis (A.61). Higher-order corrections as suggested in the Krylov sequence (A.60) can also be included in the reduction space as proposed in [43, 157, 93, 1, 97, 128], which guarantees matching higher-orders moments and thus leads to an approach similar to the moment matching technique. Higher-order correction modes have also been used in the context of substructuring and mode component synthesis [42, 130, 131]. Note that the high-order corrections for structural problems can be obtained as a by-product of the Lanczos algorithm used to compute the eigenmodes [129] and that one can also consider quasi-static corrections (i.e. for $s_0 \neq 0$) in case one is interested in a specific frequency range [160].

It is important to observe that the modal truncation augmentation uses a reduction basis that, in addition to the moments, also includes true eigenmodes of the system. In that sense this method differs from the usual moment matching techniques and it accounts both for the global behavior of the system (through its eigenmodes) and for input-specific components (through the moments).

A.3.4. Computational aspects

A general comparison of properties of model reduction techniques from the fields of structural dynamics, numerical mathematics and systems and control was given in Section A.3.2. Computational aspects are addressed in the current section.

From a computational point of view, the methods from systems and control have the highest cost. In these methods, the computational complexity is mainly due to the solution of two Lyapunov equations (see (A.44) and (A.45)), which are of the size of the original high-order model. This seriously hinders the applicability of balanced truncation to systems of very high order. Moreover, a full coordinate transformation has to be computed, before reduction can be performed by means of truncation. As a result, the total computational cost associated to balanced truncation is high. Some approaches aiming at the reduction of this cost are given by [20] and [70], where iterative methods are discussed for the computation of (low-rank approximations) of the Gramians.

The computational cost for reduction techniques from the fields of structural dynamics and numerical mathematics is significantly lower. First, these methods do not require the computation of a full coordinate transformation. Instead, only the reduction space is computed, which is given by only k basis vecttors. Furthermore, the computations are less costly since the matrix operations that are required are relatively cheap when compared to those needed for the solution of Lyapunov equations. In the mode displacement techniques from structural dynamics, only the most important eigenvalues and eigenvecttors need to be computed. Since the frequency domain of interest is typically known beforehand, efficient iterative methods can be used to find the eigenfrequencies in this range. Here, it is stressed that model reduction is often used as a tool for the fast computation of frequency response functions, which requires such efficient numerical techniques. The Krylov-subspace based moment matching techniques from numerical mathematics also have a small numerical cost. Namely, the application of the Arnoldi or Lanczos methods only requires the solutions of linear sets of equations or matrix-vecttor multiplications. Therefore, moment matching methods by Krylov subspaces can be applied to systems of very high order. Here, it is noted that the cost of two-sided moment matching methods are twice as high as the cost of one-sided methods, as two sets of basis vectors need to be obtained in the former.

Even though the application of balanced truncation seems limited from a computational point of view, it is remarked that the perception of "high-order" differs in the three different fields. Especially, models of very low order (i.e. $\mathcal{O}(10^0 - 10^1)$) are of interest in the field of systems and control. This is mainly due to the fact that controllers have to be implemented in real-time, which provides a limit on the order of the controller. Furthermore, low-order controllers are preferred because of their limited complexity. Hence, even though the computation of Lyapunov equations limits the applicability of balanced truncation to systems of order $\mathcal{O}(10^3)$, it still provides a solution to relevant model reduction problems in practice. On the other hand, the models describing mechanical structures in the field of structural dynamics typically result from finite element procedures, leading to models of orders up to $\mathcal{O}(10^6)$. Similarly, the moment matching techniques from numerical mathematics typically find application in the analysis of large-scale electrical circuits, leading to models of order up to $\mathcal{O}(10^6)$. From these applications, the need for numerically efficient model reduction procedures is clear.

A.3.5. Preservation of properties

In model reduction, the objective is the construction of a reduced-order model that gives a good approximation of the original high-order model. Herein, it is of crucial importance that the reduced-order model preserves some properties of the original system, among which stability is the most important. If the high-order system is asymptotically stable, balanced truncation (see Section A.2.3) indeed preserves this property, which is due to the fact that the (diagonal) Gramians act as Lyapunov equations. The moment matching techniques from Section A.2.2 do not satisfy such a property, such that stability of the reduced-order model can not be guaranteed in general. Nonetheless, methods exist that preserve stability for classes of linear systems (see e.g. [113]).

In the mode superposition techniques outlined in Section A.2.1, stability of the reduced-order model can not be guaranteed when the original high-order system exhibits general damping. However, in the important cases of undamped systems or systems with positive definite symmetric damping matrix (which includes the cases of proportional (Rayleigh) and modal damping), the stability properties are indeed preserved. In fact, since the reduced-order model is based on the computation of the undamped vibration modes, reduction of an undamped system leads to an undamped reduced-order system, where the most important eigenfrequencies are preserved. Stated differently, the pole locations of the most important poles remain unchanged. This property does in general not hold for balanced truncation and moment matching techniques. It is remarked that, in structural dynamics, one of the main objectives for model reduction is the fast computation of frequency response functions rather than performing time simulations. In this case, the preservation of stability properties is of less importance.

Furthermore, it is obvious that modal superposition techniques preserve the second-order form in the reduced-order model. Nonetheless, this is an important feature in the field of structural dynamics since it implies that the kinematic relation between displacement and velocity is preserved. This does not hold for balanced truncation and moment matching, even if the models stem from a second-order form. However, extensions of balanced truncation and moment matching aiming at the preservation of a second-order form exist, see e.g. [29].

Next, it is remarked that the existence of an error bound, as discussed in Section A.3.2, is closely related to stability preservation. Namely, a bound on the difference of solutions from the high-order and reduced-order systems can only be expected to exist when both systems are stable, making stability a prerequisite for the existence of an error bound.

Finally, it is often important to preserve other system properties besides stability. Herein, passivity and bounded realness are the most notable. Even though the methods as discussed in Section A.2 do not generally preserve these properties, it is noted that extensions exist that do. For the different fields, some references to the

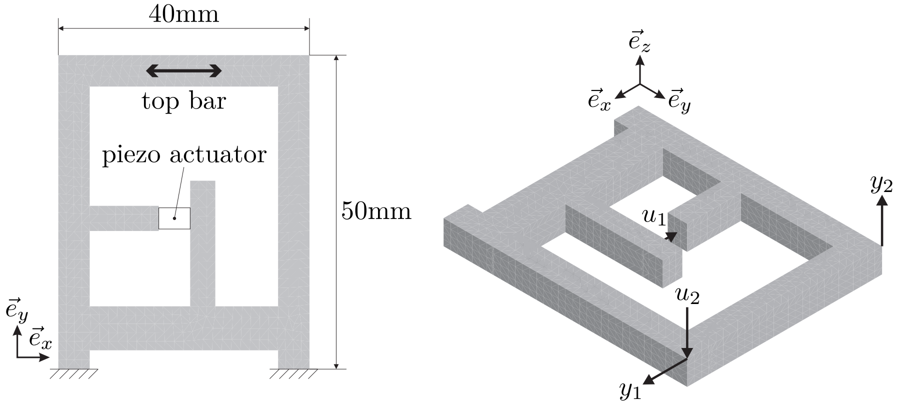


Figure A.1: Actuation frame model

literature are given in the corresponding parts in Section A.2.

A.4. Illustrative example

To illustrate the differences between methods as discussed in Section A.3, the model reduction procedures of Section A.2 are applied to a common benchmark example. Herein, focus is on the properties of the reduced-order models rather than computational aspects, as the implementations of the reduction techniques are not optimized for computational speed. The benchmark example is chosen from the domain of structural dynamics, to allow for application of all model reduction techniques discussed in Section A.2. More specifically, the structure as depicted in Figure A.1 is considered, which represents an actuation frame as applied in high-precision machine components. The frame is made out of steel. Here, a piezo-electric actuator is used to control the displacement in $\boldsymbol{e}_x$-direction of the top bar as depicted in the left panel of Figure A.1. The machine component (without the actuator) is discretized using a finite element approach, leading to a model of the form

$$\mathbf{M}\ddot{\boldsymbol{q}} + \mathbf{D}\dot{\boldsymbol{q}} + \mathbf{K}\boldsymbol{q} = \tilde{\boldsymbol{b}}_1 u_1 + \tilde{\boldsymbol{b}}_2 u_2, \tag{A.62}$$

$$y_1 = \tilde{\boldsymbol{c}}_1^{\mathrm{T}} \boldsymbol{q}, \tag{A.63}$$

$$y_2 = \tilde{\boldsymbol{c}}_2^{\mathrm{T}} \boldsymbol{q}, \tag{A.64}$$

where $\boldsymbol{q} \in \mathbb{R}^N$ is the vecttor of the displacements and rotations of the nodes, with $N = 8730$, and $\mathbf{M}$ and $\mathbf{K}$ represent the mass- and stiffness matrices, respectively, resulting from the finite-element discretization[1]. In (A.62), the input u_1 represents the actuator force (in $\boldsymbol{e}_x$-direction), whereas u_2 is a disturbance in $\boldsymbol{e}_z$-direction ($\tilde{\boldsymbol{b}}_1$ and $\tilde{\boldsymbol{b}}_2$ are the generalised force direction related to the forces u_1 and u_2, respectively). Moreover, the outputs y_1 and y_2 give the displacement of certain corner points of the actuator frame as depicted in the right panel of Figure A.1, and $\tilde{\boldsymbol{c}}_1^{\mathrm{T}}$

---

[1] The matrices are available for interested readers. Please contact one of the authors.

and $\tilde{\boldsymbol{c}}_2^{\mathrm{T}}$ represent the corresponding output matrices. Finally, the actuator frame is lightly damped, which is modeled using Rayleigh damping in the damping matrix $\mathbf{D}$.

Due to the high order of the model (A.62), model reduction is required to enable efficient time-simulations of the model. Herein, the objective of reduction is to obtain a reduced-order model which accurately captures the input-output behavior from input u_1 to output y_1.

In order to apply model reduction techniques from the fields of numerical mathematics and systems and control, the actuation frame model (A.62) has to be written in state-space form. By choosing the state vecttor as $\boldsymbol{x}^{\mathrm{T}} = [\boldsymbol{q}^{\mathrm{T}}\ \dot{\boldsymbol{q}}^{\mathrm{T}}]$, the dynamics is given by

$$\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}_1 u_1 + \boldsymbol{b}_2 u_2, \tag{A.65}$$

$$y_i = \boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{x}, \quad i \in \{1,2\}, \tag{A.66}$$

where it is noted that $\boldsymbol{x} \in \mathbb{R}^n$ with $n = 2N = 17460$. The system matrices in (A.65-A.66) read

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix}, \quad \boldsymbol{b}_i = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\tilde{\boldsymbol{b}}_i \end{bmatrix}, \quad \boldsymbol{c}_i = \begin{bmatrix} \tilde{\boldsymbol{c}}_i \\ \mathbf{0} \end{bmatrix}, \quad i \in \{1,2\}. \tag{A.67}$$

Alternatively, the moment matching methods discussed in Section A.2.2 can also be applied using system descriptions of the form

$$\mathbf{E}\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}_1 u_1 + \boldsymbol{b}_2 u_2, \tag{A.68}$$

$$y_i = \boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{x}, \quad i \in \{1,2\}, \tag{A.69}$$

with matrices

$$\mathbf{E} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\mathbf{D} \end{bmatrix}, \quad \boldsymbol{b}_i = \begin{bmatrix} \mathbf{0} \\ \tilde{\boldsymbol{b}}_i \end{bmatrix}, \quad i \in \{1,2\}, \tag{A.70}$$

which avoids the need for inversion of the matrix $\mathbf{M}$. In this form, the output vecttors $\boldsymbol{c}_i$ remain unchanged. Furthermore, since the matrices $\mathbf{M}$, $\mathbf{D}$ and $\mathbf{K}$ are symmetric

The model reduction techniques from Section A.2 are applied to this example. From the field of structural dynamics, the mode displacement method is used. Here, it is recalled that this method is based on the undamped system (i.e. the projection basis is computed for $\mathbf{D} = \mathbf{0}$) and the location of the inputs and outputs (i.e. knowledge on $\tilde{\boldsymbol{b}}_i$ and $\tilde{\boldsymbol{c}}_i$) is not taken into account. Nonetheless, the (static) influence of the locations of the inputs can be taken into account by the extensions given by mode acceleration and modal truncation augmentation (see Section A.2.1). On the other hand, model reduction with respect to input u_1 and output y_1 is performed using moment matching and balanced truncation. Since these methods are based on the state-space form (A.65) (or (A.68)), damping can be included. In moment matching, the expansion point is chosen as $s_0 = 0$ and a one-sided projection is used, based on the input only (see (A.30)). In reduction, the reduced-order size of the first-order models k is chosen as $2K$, with K the number of modes taken into account in the mode displacement methods. This choice is motivated by the fact that the representation of a second-order model in first-order form doubles the number of equations. Hereby, the reduced-order models in first-order and second-order form have comparable complexity. Choosing $K = 10$ ($k = 20$) leads to the frequency response functions, with input u_1 and output y_1, as depicted in Figures A.2 and A.3. Here, H_{11} and $\hat{H}_{11}$ denote the transfer functions from input u_1 to output y_1 of the high-order and reduced-order models, respectively.

From Figure A.2 it is clear that all reduction methods provide a good reduced-order model for the frequency range up to about 10^4Hz. For higher frequencies, the balanced truncation (BT) method yields a better improvement than the mode displacement method (MD) and the moment matching technique (MM). This can be understood be recalling that the moment matching method chooses the eigenvecttors corresponding to the K lowest eigenfrequencies as a reduction basis, where the deflection shapes corresponding to the first three eigenvecttors are depicted in Figure A.4. On the other hand, balanced truncation takes the location of the input and output into account. From Figure A.4 it can be seen that the first mode is a pure bending mode, leading to displacements in $\boldsymbol{e}_z$-direction. Thus, this mode does not influence the input-output behavior from input u_1 to output y_1, as they are both taken in $\boldsymbol{e}_x$-direction. A similar statement can be made about the third mode and some other higher modes. As a result, these modes are not included in the reduction basis for balanced truncation and modes corresponding to higher frequencies are taken instead, explaining the better result (when compared to the mode displacement technique) for high frequencies. Finally, the moment matching technique (MM) gives a good approximation at low frequencies, which originates from the choice of the expansion point as $s_0 = 0$. Therefore, the lowest resonance peaks are accurately captured, where it is recalled that the location of the input u_1 is taken into account in the construction of the reduced-order model. Consequently, the non-exited modes do not appear in reduced-order model obtained by moment matching. The performance of the moment matching technique is particularly apparent in the error graph in Figure A.3, where it gives the best approximation for low frequencies. However, moment matching gives the largest $\mathcal{H}_\infty$ norm of the error system, see Figure A.3. The lowest norm is obtained by balanced truncation, due to the good match for higher frequencies as discussed before.

To illustrate the influence of the locations of the inputs and outputs on the reduced-order model, the frequency response functions for input u_2 and output y_2 are depicted in Figure A.5, whereas the corresponding error is given in Figure A.6. Here, the same reduction bases were used as in Figure A.2. Hence, the input u_2 and output y_2 are not taken into account in the model reduction procedure. Since the mode displacement method is based on the global dynamics rather than specific inputs and outputs, it also gives a good approximation for these new inputs. On the contrary, the reduced-order models obtained by balanced truncation and moment matching are dependent on the inputs and outputs taken into account in the reduction, where it is recalled that reduction was based on input u_1 and output y_1. Therefore, they

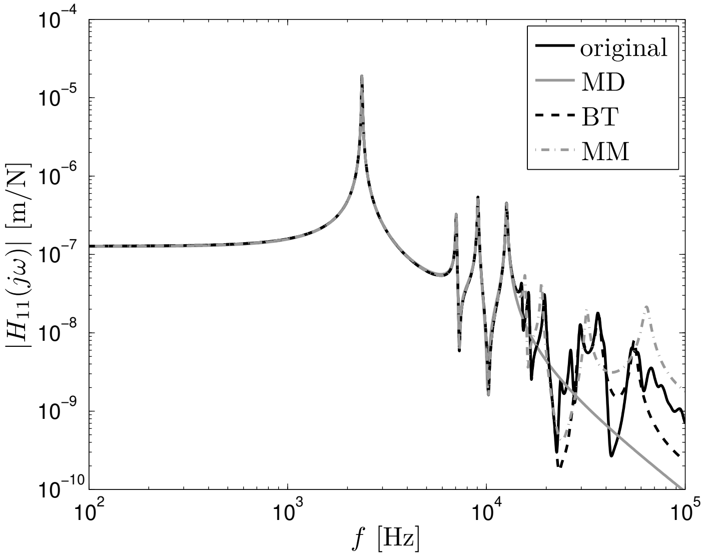


Figure A.2: Comparison of the modal displacement method (MD), balanced truncation (BT) and moment matching (MM) for reduction to $K = 10$ ($k = 20$): magnitude of the frequency response function for input u_1 and output y_1.

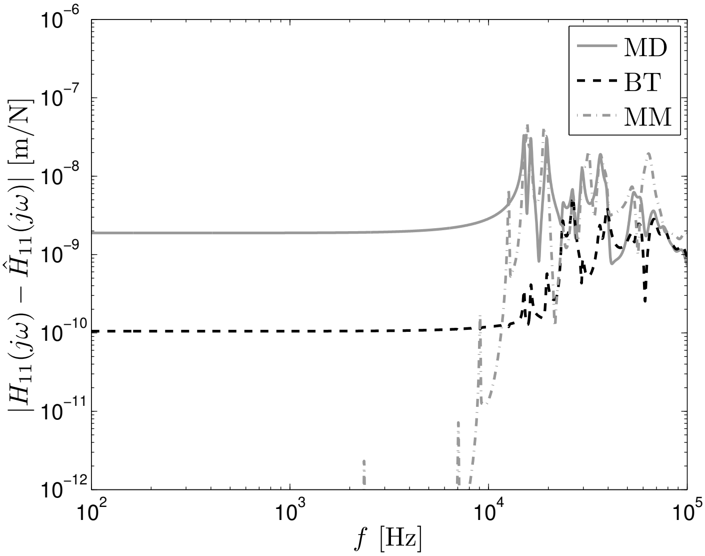


Figure A.3: Magnitude of the error for reduction to $K = 10$ ($k = 20$) for input u_1 and output y_1. Line styles as in Figure A.2.

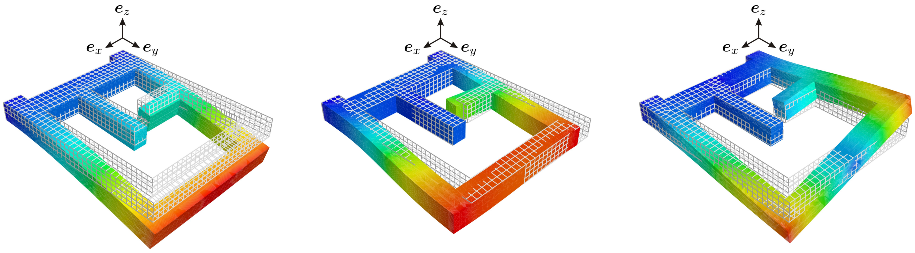


Figure A.4: Deflection shapes of the first three modes with eigenfrequencies $1.20 \cdot 10^3$, $2.66 \cdot 10^3$ and $2.83 \cdot 10^3$Hz.

do not give a good approximation for the input-output behavior from input u_2 to output y_2, as is clear from the large errors in Figure A.6. As an example, the first mode (see Figure A.4), which was not important for the input-output behavior from input u_1 to output y_1 as in Figure A.2, forms an important resonance peak in Figure A.5 and is not captured by either balanced truncation or the moment matching technique.

Finally, stability of the reduced-order models is checked. Since the truss frame system exhibits modal damping, the reduced-order model obtained by the mode displacement technique is guaranteed to be stable. Stability is also guaranteed in the case of balanced truncation. For moment matching, stability can not be guaranteed a priori. In fact, for $k = 20$, the reduced-order model obtained by moment matching is unstable, as follows from an a posteriori check.

A.5. Conclusions

In this appendix chapter, an overview and comparison of popular model reduction methods from the fields of structural dynamics, numerical mathematics and systems and control are provided. A detailed review is given on mode displacement techniques, moment matching methods and balanced truncation, whereas important extensions are outlined briefly.

The differences and similarities between presented methods are discussed, both qualitatively and quantitatively. Here, an important difference is the fact that the global dynamics is taken into account in the mode displacement methods, whereas moment matching and balanced truncation aim at the approximation of input-output behavior. Moreover, the computational cost of the methods differs, which limits the application of balanced truncation to systems of moderate size. On the other hand, balanced truncation has an a priori error bound, which is not the case for the mode displacement and moment matching techniques. Also, balanced truncation and the mode displacement method preserve stability of the high-order model, whereas stability is not guaranteed when applying moment matching.

The overview of the differences and commonalities between the different reduction methods facilitates the choice of the reduction technique with the desirable properties for a given reduction problem.

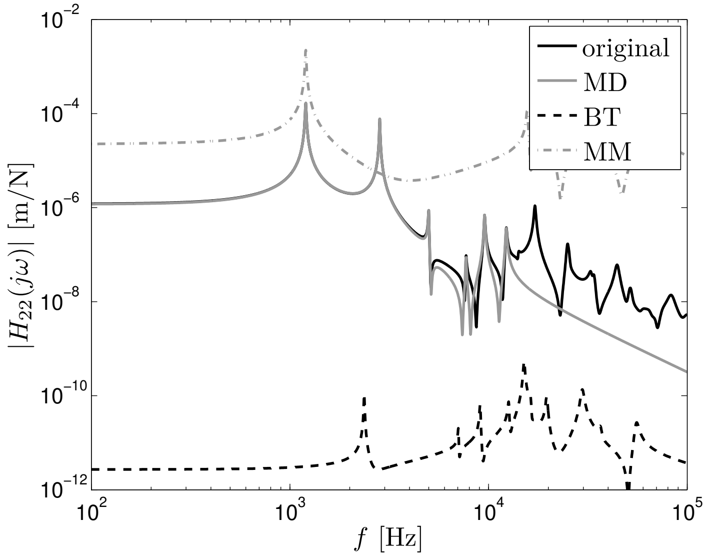


Figure A.5: Comparison of the modal displacement method (MD), balanced truncation (BT) and moment matching (MM) for reduction to $K = 10$ ($k = 20$): magnitude of the frequency response function for input u_2 and output y_2. For balanced truncation and moment matching, the reduced-order model is based on input u_1 and output y_1.

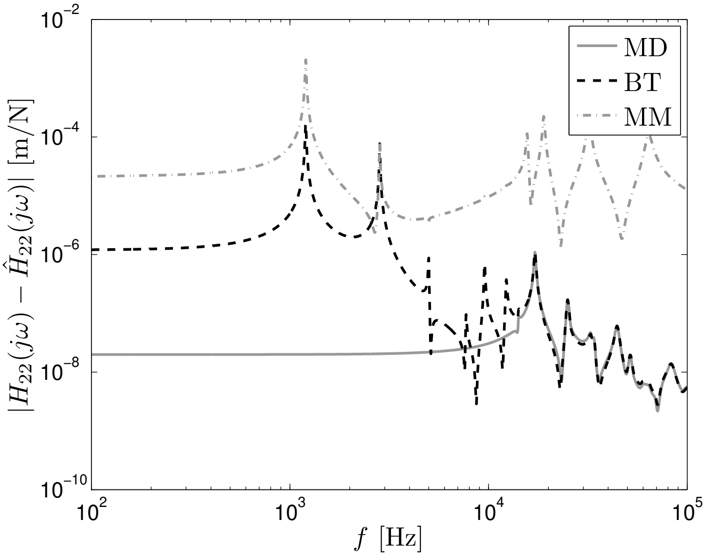


Figure A.6: Magnitude of the error for reduction to $K = 10$ ($k = 20$) for input u_2 and output y_2. For balanced truncation and moment matching, the reduced-order model is based on input u_1 and output y_1.

Finally, these differences are illustrated by means of application of the different methods to a common benchmark example.

B

Pre- and Post-multiplication Matrices for Creating Symmetric Vibro-acoustic Matrices

In this appendix, the pre- and post-multiplication matrices which can be used to end up with symmetric system matrices is outlined based on [52]. In [52], two pre- and two post-multiplication matrices were presented. We briefly outline these matrices for convenience below for the presentation. Namely, we first present the pre-multiplication matrices and then we present the post-multiplication matrices along with the resulting symmetric matrices.

B.1. Pre-multiplication matrices

The first pre-multiplication matrix resulting in a symmetric format is

$$\kappa_1 = \begin{bmatrix} \mathbf{K}_s \mathbf{M}_s^{-1} & \mathbf{0} \\ -\mathbf{K}_{sf}^{\mathrm{T}} \mathbf{M}_s^{-1} & \mathbf{I} \end{bmatrix} \tag{B.1}$$

and premultiplying (2.29) with κ_1, the symmetric matrices read as,

$$\mathbf{M}_{sym} = \begin{bmatrix} \mathbf{K}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_f \end{bmatrix}, \quad \mathbf{K}_{sym} = \begin{bmatrix} \mathbf{K}_s \mathbf{M}_s^{-1} \mathbf{K}_s & -\mathbf{K}_s \mathbf{M}_s^{-1} \mathbf{K}_{sf} \\ \mathbf{K}_{sf}^{\mathrm{T}} \mathbf{M}_s^{-1} \mathbf{K}_s & (\mathbf{K}_f + \mathbf{K}_{sf}^{\mathrm{T}} \mathbf{M}_s^{-1} \mathbf{K}_{sf}) \end{bmatrix} \tag{B.2}$$

The second pre-multiplication matrix resulting in a symmetric format is

$$\kappa_2 = \begin{bmatrix} \mathbf{I} & \mathbf{K}_{sf} \mathbf{K}_f^{-1} \\ \mathbf{0} & \mathbf{M}_f \mathbf{K}_f^{-1} \end{bmatrix} \tag{B.3}$$

and premultiplying (2.29) with κ_2, the symmetric matrices read as,

$$\mathbf{M}_{sym} = \begin{bmatrix} (\mathbf{M}_s + \mathbf{K}_{sf} \mathbf{K}_f^{-1} \mathbf{K}_{sf}^{\mathrm{T}}) & \mathbf{K}_{sf} \mathbf{K}_f^{-1} \mathbf{M}_f \\ \mathbf{M}_f \mathbf{K}_f^{-1} \mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f \mathbf{K}_f^{-1} \mathbf{M}_f \end{bmatrix}, \quad \mathbf{K}_{sym} = \begin{bmatrix} \mathbf{K}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_f \end{bmatrix} \tag{B.4}$$

B.2. Post-multiplication matrices

The first post-multiplication matrix resulting in a symmetric format is written as,

$$\tau_1 = \begin{bmatrix} \mathbf{K}_s^{-1}\mathbf{M}_s & \mathbf{K}_s^{-1}\mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{B.5}$$

and post-multiplying (2.29) with τ_1, the symmetric matrices read as,

$$\mathbf{M}_{sym} = \begin{bmatrix} \mathbf{M}_s\mathbf{K}_s^{-1}\mathbf{M}_s & \mathbf{M}_s\mathbf{K}_s^{-1}\mathbf{K}_{sf} \\ \mathbf{K}_{sf}^{\mathrm{T}}\mathbf{K}_s^{-1}\mathbf{M}_s & (\mathbf{M}_f + \mathbf{K}_{sf}^{\mathrm{T}}\mathbf{K}_s^{-1}\mathbf{K}_{sf}) \end{bmatrix}, \quad \mathbf{K}_{sym} = \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix} \tag{B.6}$$

It is important to note that the transformations with the post-multipliers result in a change of basis where the main variables of the domain are transformed to a new set. This was also shortly outlined in (4.23).

The second post-multiplication matrix resulting in a symmetric format is written as,

$$\tau_2 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{M}_f^{-1}\mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{M}_f^{-1}\mathbf{K}_f \end{bmatrix} \tag{B.7}$$

and post-multiplying (2.29) with τ_2, the symmetric matrices read as,

$$\mathbf{M}_{sym} = \begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix}, \quad \mathbf{K}_{sym} = \begin{bmatrix} (\mathbf{K}_s + \mathbf{K}_{sf}\mathbf{M}_f^{-1}\mathbf{K}_{sf}^{\mathrm{T}}) & -\mathbf{K}_{sf}\mathbf{M}_f^{-1}\mathbf{K}_f \\ -\mathbf{K}_f\mathbf{M}_f^{-1}\mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{K}_f\mathbf{M}_f^{-1}\mathbf{K}_f \end{bmatrix} \tag{B.8}$$

C

Convergence Proofs on Subspace Iteration

In this part, we would like to provide the mathematical convergence proofs behind the inverse iteration and the Subspace iteration. The presentation is mainly based on [63],[110],[18].

C.1. Refresher on inverse iteration and its convergence

The generalized eigenvalue problem is given as

$$(\mathbf{K} - \lambda \mathbf{M})\boldsymbol{X} = 0. \tag{C.1}$$

To solve the eigenvalue problem in a power iteration like scheme, one can convert (C.1)into standard eigenvalue problem, namely,

$$(\mathbf{K}^{-1}\mathbf{M})\boldsymbol{X} = \frac{1}{\lambda}\boldsymbol{X} \longrightarrow \mathbf{A}\boldsymbol{X} = \tilde{\lambda}\boldsymbol{X}, \tag{C.2}$$

with $\tilde{\lambda} = 1/\lambda$ and $\mathbf{A} = \mathbf{K}^{-1}\mathbf{M}$ which is the dynamic matrix.

With this transformation, one could conceptually[1] use the power iteration with the dynamic matrix. Power iteration locates the maximum of the eigenvalues of the system in an absolute sense therefore we can understand that the above transformation, (C.2), helps to find the lowest eigenvalue of the original problem due to the reciprocal relation $\tilde{\lambda} = 1/\lambda$. With a small arrangement, the power iterations can be put in a more efficient framework where the dynamic matrix, $\mathbf{K}^{-1}\mathbf{M}$, is never formed explicitly. Namely, for step k, the operations are performed in two steps,

$$\boldsymbol{y}_k = \mathbf{M}\boldsymbol{q}_k \tag{C.3a}$$

$$\boldsymbol{q}_{k+1} = \mathbf{K}^{-1}\boldsymbol{y}_k \tag{C.3b}$$

where $\boldsymbol{q}_{k+1}$ represents the next iteration vector in the sequence. And, this practice is called the inverse iteration approach and detailed in[63],[110].

Since the eigenvectors $\mathbf{X}$ of (C.1) form an orthogonal set of linearly independent vectors, any vector $\boldsymbol{v}$ which is not a null-vector, can be represented as a linear

---

[1] Since it is not practically possible to construct $\mathbf{K}^{-1}\mathbf{M}$ due to its fully populated nature

combination of the eigenvectors of the problem, namely,

$$\boldsymbol{v} = \boldsymbol{X}_1\eta_1 + \boldsymbol{X}_2\eta_2 + \boldsymbol{X}_3\eta_3 + \cdots + \boldsymbol{X}_m\eta_m \longrightarrow \boldsymbol{v} = \mathbf{X}\boldsymbol{\eta}, \tag{C.4}$$

where m represents the column size of the original matrices $\mathbf{K}$ and $\mathbf{M}$. Taking a closer look at (C.3b), one can write that in the format of successive linear solves, namely,

$$\mathbf{K}\boldsymbol{q}_{k+1} = \mathbf{M}\boldsymbol{q}_k, \tag{C.5}$$

If one writes (C.5) for $k = 1$, it reads as

$$\mathbf{K}\boldsymbol{q}_2 = \mathbf{M}\boldsymbol{q}_1, \tag{C.6}$$

by using the fact that $\boldsymbol{q}_2$ and $\boldsymbol{q}_1$ can be represented in the space of the eigenvectors, namely,

$$\boldsymbol{q}_1 = \mathbf{X}\boldsymbol{\eta}_1, \quad \boldsymbol{q}_2 = \mathbf{X}\boldsymbol{\eta}_2, \tag{C.7}$$

and substituting these expressions for $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ in (C.6) and premultiplying by $\mathbf{X}^{\mathrm{T}}$, the equation reads as

$$\mathbf{X}^{\mathrm{T}}\mathbf{K}\mathbf{X}\boldsymbol{\eta}_2 = \mathbf{X}^{\mathrm{T}}\mathbf{M}\mathbf{X}\boldsymbol{\eta}_1. \tag{C.8}$$

Assuming that the modes are mass normalized, namely,

$$\mathbf{X}^{\mathrm{T}}\mathbf{K}\mathbf{X} = \boldsymbol{\Omega} \equiv \mathbf{diag}(\lambda_1, \lambda_2, ..., \lambda_k), \quad \mathbf{X}^{\mathrm{T}}\mathbf{M}\mathbf{X} = \mathbf{I}. \tag{C.9}$$

(C.8) can be written as,

$$\boldsymbol{\eta}_2 = \boldsymbol{\Omega}^{-1}\boldsymbol{\eta}_1, \tag{C.10}$$

Repeating the process for k=2, results in $\boldsymbol{\eta}_3 = \boldsymbol{\Omega}^{-1}\boldsymbol{\eta}_2 \equiv \boldsymbol{\Omega}^{-2}\boldsymbol{\eta}_1$, and, for higher modes, this recursion results in,

$$\boldsymbol{q}_{k+1} = \mathbf{X}\boldsymbol{\Omega}^{-k}\boldsymbol{\eta}_1, \quad \boldsymbol{\eta}_1 = [\eta_1^1, \eta_2^1, ..., \eta_m^1] \tag{C.11}$$

Since $\boldsymbol{\Omega}$ is a diagonal matrix due to the orthogonality properties of the eigenvectors, (C.11) can be expanded and arranged as,

$$\boldsymbol{q}_{k+1} = \frac{1}{\lambda_1^k}[\boldsymbol{X}_1\eta_1^1 + \boldsymbol{X}_2\eta_1^2(\frac{\lambda_1}{\lambda_2})^k + \cdots + \boldsymbol{X}_k\eta_1^m(\frac{\lambda_1}{\lambda_m})^k], \tag{C.12}$$

assuming that $c_1 \neq 0$ and the eigenvalues are distinct, as k increases, $(\frac{\lambda_1}{\lambda_i})^k \to 0$, then we are left with

$$\boldsymbol{q}_{k+1} = \frac{\boldsymbol{X}_1\eta_1^1}{\lambda_1^k} + \text{small-order terms}, \tag{C.13}$$

which indicates that, in the limit, $\boldsymbol{q}_{k+1}$ converges to a multiple of $\boldsymbol{X}_1$ which is the fundamental eigenvector of the original problem.

C.2. Convergence of Subspace iteration

With the inverse iteration technique outlined in Section C.1, we are iterating with only one vector at a time. And to find higher order modes, at each iteration, we have to start from scratch discarding the information generated in the previous steps. Subspace iteration can be thought as an improvement on the original inverse iteration scheme in which the iterations are performed in blocks. In order to prove the convergence of Subspace iteration, as for the case of inverse iteration, we start with the block version of (C.3b), namely, (3.2).

We can start by rewriting the eigenvector block as follows,

$$\mathbf{X} = [\mathbf{X}_1 \quad \mathbf{X}_2]\,, \tag{C.14}$$

and similarly this could also be done for $\boldsymbol{\Omega}$, namely

$$\boldsymbol{\Omega} = \mathbf{diag}(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) \tag{C.15}$$

where $\mathbf{X}_1$ is an $n \times p$ block matrix and $\mathbf{X}_2$ is an $n \times (n-p)$ block matrix. Similarly, $\boldsymbol{\Omega}_1$ is a $p \times p$ diagonal matrix and $\boldsymbol{\Omega}_2$ is an $(n-p) \times (n-p)$ diagonal matrix. Using (C.14) and (C.15), the generalized eigenvalue problem can be written as,

$$\mathbf{K}\,[\mathbf{X}_1 \quad \mathbf{X}_2] = \mathbf{M}\,[\mathbf{X}_1 \quad \mathbf{X}_2] \begin{bmatrix} \boldsymbol{\Omega}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega}_2 \end{bmatrix}, \tag{C.16}$$

resulting in the submatrix equations

$$\mathbf{K}\mathbf{X}_1 = \mathbf{M}\mathbf{X}_1\boldsymbol{\Omega}_1, \tag{C.17}$$
$$\mathbf{K}\mathbf{X}_2 = \mathbf{M}\mathbf{X}_2\boldsymbol{\Omega}_2. \tag{C.18}$$

In the Subspace iteration method, we iterate simultaneously on p trial vectors with

$$\mathbf{K}\mathbf{Q}_{k+1} = \mathbf{M}\mathbf{Q}_k \tag{C.19}$$

which is the block analog of (C.3b). Similar to the expansion performed for the inverse iterations, the trial vectors, $\mathbf{Q}_k$, might be represented as a linear combination of the eigenvectors, namely,

$$\mathbf{Q}_k = \mathbf{X}_1\boldsymbol{\eta}_1 + \mathbf{X}_2\boldsymbol{\eta}_2, \tag{C.20}$$

where $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ are the coefficient matrices of order $p \times p$ and $(n-p) \times p$. Substituting (C.20) in (C.19) and writing it explicitly for step $k = 1$ gives

$$\mathbf{K}\mathbf{Q}_2 = \mathbf{M}\left(\mathbf{X}_1\boldsymbol{\eta}_1 + \mathbf{X}_2\boldsymbol{\eta}_2\right) \longrightarrow \mathbf{Q}_2 = \mathbf{K}^{-1}\mathbf{M}\mathbf{X}_1\boldsymbol{\eta}_1 + \mathbf{K}^{-1}\mathbf{M}\mathbf{X}_2\boldsymbol{\eta}_2, \tag{C.21}$$

by using (C.17) and (C.18), (C.21) can be written generally as,

$$\mathbf{Q}_2 = \mathbf{X}_1\boldsymbol{\Omega}_1^{-1}\boldsymbol{\eta}_1 + \mathbf{X}_2\boldsymbol{\Omega}_2^{-1}\boldsymbol{\eta}_2, \tag{C.22}$$

Using the inverse iteration relation of the Subspace algorithm repeatedly results in a general expression for the next set of iteration vectors at step k, namely,

$$\mathbf{X}_{k+1} = \mathbf{X}_1\boldsymbol{\Omega}_1^{-k}\boldsymbol{\eta}_1 + \mathbf{X}_2\boldsymbol{\Omega}_2^{-k}\boldsymbol{\eta}_2, \tag{C.23}$$

Due to the repeated application of the inverse iteration relation, the second term in (C.23), will decrease much faster in comparison to the first term. Thus if k increases sufficiently, we can safely write

$$\mathbf{X}_{k+1} \simeq \mathbf{X}_1 \boldsymbol{\Omega}_1^{-k} \boldsymbol{\eta}_1, \tag{C.24}$$

indicating that the contribution of the vectors in $\mathbf{X}_2$ vanishes as a result of the inverse iteration relation used in the Subspace iteration algorithm.

The next cycle in the Subspace iteration is the solution of the $p \times p$ reduced eigenvalue problem. This is a direct result of the minimization of the p simultaneous Rayleigh quotients with a set of assumed admissible displacement functions. In the case of Subspace iteration, these functions are represented in the space of the computed vectors $\tilde{\mathbf{X}}_{k+1}$ as,

$$\mathbf{V}_{k+1} = \mathbf{X}_{k+1} \mathbf{E}_{k+1}, \tag{C.25}$$

where the $p \times p$ matrix, $\mathbf{E}_{k+1}$ contains the Ritz coordinates when combined as linear combination with the vectors in $\mathbf{X}_{k+1}$ will minimize the Rayleigh quotient[110].

To conclude, the convergence behaviour of the Subspace iteration depends mainly on two facts, namely,

1. Representation of the main variables in the space of the orthogonal eigenvectors and the use of their **K** and **M**-orthogonality properties.
2. The use of the inverse iteration relation, which by the use of the orthogonality properties of the modes, is a proof for the convergence of the method.

D

Proofs on 3-term Recurrence Relations for Partial Reorthogonalization

The analysis presented in this appendix is mainly based on [110],[118],[142].

Base of the recurrence relation
Starting from the relation provided in blocks at step i of the Lanczos process, namely,

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_i - \mathbf{Q}_i\mathbf{T}_i = \beta_{i+1}\boldsymbol{q}_{i+1}\boldsymbol{e}_i^{\mathrm{T}} \tag{D.1}$$

Multiplying (D.1) from left by $\boldsymbol{e}_i$, we can write

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_i\boldsymbol{e}_i - \mathbf{Q}_i\mathbf{T}_i\boldsymbol{e}_i = \beta_{i+1}\boldsymbol{q}_{i+1} \tag{D.2}$$

At this point, we can multiply (D.2) by $\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}$ from left in order to write,

$$\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_i\boldsymbol{e}_i - \mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\mathbf{Q}_i\mathbf{T}_i\boldsymbol{e}_i = \beta_{i+1}\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_{i+1} \tag{D.3}$$

At this point, we can reshape (D.1) into

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_i = \mathbf{Q}_i\mathbf{T}_i + \beta_{i+1}\boldsymbol{q}_{i+1}\boldsymbol{e}_i^{\mathrm{T}} \tag{D.4}$$

Taking the transpose of (D.5), we can write,

$$(\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_i)^{\mathrm{T}} = \mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\mathbf{K}^{-\mathrm{T}} \equiv \boldsymbol{e}_i\boldsymbol{q}_{i+1}^{\mathrm{T}}\beta_{i+1} + \mathbf{T}_i^{\mathrm{T}}\mathbf{Q}_i^{\mathrm{T}} \tag{D.5}$$

By carefully looking at (D.3), it is possible to use (D.5) in that relation. Specifically, we have to replace [1]$\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\mathbf{K}^{-1}$ by (D.5) resulting in

$$(\boldsymbol{e}_i\boldsymbol{q}_{i+1}^{\mathrm{T}}\beta_{i+1} + \mathbf{T}_i^{\mathrm{T}}\mathbf{Q}_i^{\mathrm{T}})\mathbf{M}\mathbf{Q}_i\boldsymbol{e}_i - \mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\mathbf{Q}_i\mathbf{T}_i\boldsymbol{e}_i = \beta_{i+1}\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_{i+1}\mathbf{T}_i\boldsymbol{e}_i \tag{D.6}$$

resulting in[2]

$$\mathbf{T}_i\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\mathbf{Q}_i\boldsymbol{e}_i + \beta_{i+1}\boldsymbol{e}_i\boldsymbol{q}_{i+1}^{\mathrm{T}}\mathbf{M}\mathbf{Q}_i\boldsymbol{e}_i - \mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\mathbf{Q}_i\mathbf{T}_i\boldsymbol{e}_i = \beta_{i+1}\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_{i+1} \tag{D.7}$$

by grouping related terms, we can write (D.7) in a different format as follows:

$$(\mathbf{T}_i\mathbf{Q}_i^{\mathrm{T}} + \beta_{i+1}\boldsymbol{e}_i\boldsymbol{q}_{i+1}^{\mathrm{T}})\mathbf{M}\mathbf{Q}_i\boldsymbol{e}_i - \mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\mathbf{Q}_i\mathbf{T}_i\boldsymbol{e}_i = \beta_{i+1}\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_{i+1} \tag{D.8}$$

(D.8) is the main equation that we are going to use in order to express the recurrence relation for the partial orthogonalization operations.

---

[1] Due to the symmetry of K, tranpose operation has no effect.
[2] Since $\mathbf{T}_i$ is also symmetric

Use of the three-term recurrence to derive the partial reorthogonalization relations
Going back to (D.2) and writing that in another format as,

$$\mathbf{Q}_i\mathbf{T}_i\boldsymbol{e}_i = \mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_i\boldsymbol{e}_i - \beta_{i+1}\boldsymbol{q}_{i+1} \tag{D.9}$$

considering $\boldsymbol{q}_i = \mathbf{Q}_i\boldsymbol{e}_i$ in (D.2) with the three-term recurrence relation

$$\beta_{i+1}\boldsymbol{q}_{i+1} = \mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}_i - \alpha_i\boldsymbol{q}_i - \beta_i\boldsymbol{q}_{i-1} \tag{D.10}$$

or

$$\mathbf{K}^{-1}\mathbf{M}\boldsymbol{q}_i = \beta_{i+1}\boldsymbol{q}_{i+1} + \alpha_i\boldsymbol{q}_i + \beta_i\boldsymbol{q}_{i-1} \tag{D.11}$$

Using (D.11), we can write (D.9) as follows:

$$\mathbf{Q}_i\mathbf{T}_i\boldsymbol{e}_i = \alpha_i\boldsymbol{q}_i + \beta_i\boldsymbol{q}_{i-1} \tag{D.12}$$

Derivation of the complete relation
We can now substitute (D.12) in the second term on the left side of the equality in (D.8), namely,

$$(\mathbf{T}_i\mathbf{Q}_i^{\mathrm{T}} + \beta_{i+1}\boldsymbol{e}_i\boldsymbol{q}_{i+1}^{\mathrm{T}})\mathbf{M}\mathbf{Q}_i\boldsymbol{e}_i - \mathbf{Q}_i^{\mathrm{T}}\mathbf{M}(\alpha_i\boldsymbol{q}_i + \beta_i\boldsymbol{q}_{i-1}) = \beta_{i+1}\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_{i+1} \tag{D.13}$$

which can also be written as

$$(\mathbf{T}_i\mathbf{Q}_i^{\mathrm{T}} + \beta_{i+1}\boldsymbol{e}_i\boldsymbol{q}_{i+1}^{\mathrm{T}})\mathbf{M}\mathbf{Q}_i\boldsymbol{e}_i - \alpha_i\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_i - \beta_i\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_{i-1} = \beta_{i+1}\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_{i+1} \tag{D.14}$$

By grouping the first and third terms on the left side in (D.14), we can write

$$\beta_{i+1}(\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_{i+1} - \boldsymbol{e}_i\boldsymbol{q}_{i+1}^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_i) = (\mathbf{T}_i - \alpha_i\mathbf{I})\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_i - \beta_i\mathbf{Q}_i^{\mathrm{T}}\mathbf{M}\boldsymbol{q}_{i-1} \tag{D.15}$$

which is valid for $i \geq 2$.

E

Relation of Left and Right Eigenvector with τ

By using τ from (5.19), we can write the symmetric vibro-acoustic eigenvalue problem and define its eigenvectors by

$$(\mathbf{K}_c - \omega^2 \mathbf{M}_c)\tau \boldsymbol{\phi}_{sym} = \mathbf{0} \longleftrightarrow (\mathbf{K}_c - \omega^2 \mathbf{M}_c)\boldsymbol{\phi}_R = \mathbf{0} \tag{E.1}$$

Investigating (E.1), we can conclude the following algebraic result on the right eigenvectors originating from the original nonsymmetric eigenvalue problem. Namely, the right eigenvectors read as,

$$\boldsymbol{\phi}_R = \tau \boldsymbol{\phi}_{sym} \tag{E.2}$$

The left eigenvector of this problem is, by symmetry, identical to $\boldsymbol{\phi}_{sym}$, namely $\boldsymbol{\phi}_L = \boldsymbol{\phi}_{sym}$. Expanding (E.2) by using the definition of τ, we can write

$$\boldsymbol{\phi}_R = \tau \boldsymbol{\phi}_{sym} \longleftrightarrow \boldsymbol{\phi}_R = \tau \begin{bmatrix} \boldsymbol{\phi}_{s,L} \\ \boldsymbol{\phi}_{f,L} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{K}_s^{-1}\mathbf{M}_s \boldsymbol{\phi}_{s,L} + \mathbf{K}_s^{-1}\mathbf{K}_{sf}\boldsymbol{\phi}_{f,L} \\ \boldsymbol{\phi}_{f,L} \end{bmatrix} \tag{E.3}$$

The left eigenvalue problem can be written as

$$\left(\mathbf{K}_c^{\mathrm{T}} - \omega^2 \mathbf{M}_c^{\mathrm{T}}\right) \boldsymbol{\phi}_L = \mathbf{0} \rightarrow \left(\begin{bmatrix} \mathbf{K}_s & \mathbf{0} \\ -\mathbf{K}_{sf}^{\mathrm{T}} & \mathbf{K}_f \end{bmatrix} - \omega^2 \begin{bmatrix} \mathbf{M}_s & \mathbf{K}_{sf} \\ \mathbf{0} & \mathbf{M}_f \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{\phi}_{s,L} \\ \boldsymbol{\phi}_{f,L} \end{bmatrix} = \mathbf{0} \tag{E.4}$$

The first line in (E.4) yields

$$\boldsymbol{\phi}_{s,L} = \omega^2 \left(\mathbf{K}_s^{-1}\mathbf{M}_s \boldsymbol{\phi}_{s,L} + \mathbf{K}_s^{-1}\mathbf{K}_{sf}\boldsymbol{\phi}_{f,L} \right) \tag{E.5}$$

Substituting this relation in (E.3) one finds

$$\boldsymbol{\phi}_R = \begin{bmatrix} \frac{1}{\omega^2}\boldsymbol{\phi}_{s,L} \\ \boldsymbol{\phi}_{f,L} \end{bmatrix} \qquad \text{or} \qquad \boldsymbol{\phi}_L = \begin{bmatrix} \omega^2 \boldsymbol{\phi}_{s,R} \\ \boldsymbol{\phi}_{f,R} \end{bmatrix} \tag{E.6}$$

which is inline with the proofs in [99] and [161].

Bibliography

[1] M. Akgün. "A new family of mode-superposition methods for response calculations". In: *Journal of Sound and Vibration* 167.2 (1993), pp. 289–302 (cit. on p. 156).

[2] N. Akkas, H. Akay, and C. Yilmaz. "Applicability of general-purpose finite element programs in solid-fluid interaction problems". In: *Computers & Structures* 10.5 (1979), pp. 773–783 (cit. on p. 7).

[3] A. Alia, N. Aquelet, and M. Souli. "Symmetrization Procedure for Structure-Fluid Cavity System Involving Pseudostatic Correction". In: *39th AIAA Fluid Dynamics Conference*. 2009 (cit. on p. 22).

[4] A. Alia and M. Souli. "Symmetrization Procedure for Structure–Fluid Cavity System Involving Pseudostatic Correction". In: *AIAA Journal* 48.10 (2010), pp. 2196–2205 (cit. on p. 22).

[5] R. Allemang and D. Brown. "A correlation coefficient for Modal Vector Analysis". In: *Proceedings of the International Modal Analysis Conference, Society for Experimental Mechanics, Bethel, CT*. 1982, pp. 110–116 (cit. on p. 61).

[6] ANSYS. "Academic Research". Release 18 (cit. on pp. 16, 38, 60).

[7] A. Antoulas. "An overview of approximation methods for large-scale dynamical systems". In: *Annual Reviews in Control* 29.2 (2005), pp. 181–190 (cit. on p. 136).

[8] A. Antoulas. *Approximation of large-scale dynamical systems*. SIAM, Philadelphia, USA, 2005 (cit. on pp. 136, 144, 150, 152).

[9] S. A. Arjmandi and V. Lotfi. "Comparison of Three Efficient Methods for Computing Mode Shapes of Fluid-Structure Interaction Systems". In: *Arab J. Sci. Eng.* 38 (2013), pp. 787–803 (cit. on p. 44).

[10] W. Arnoldi. "The principle of minimized iterations in the solution of the matrix eigenvalue problem". In: *Quarterly of Applied Mathematics* 9 (1951), pp. 17–29 (cit. on pp. 44, 47).

[11] N. Atalla and R. J. Bernhard. "Review of Numerical Solutions for Low-Frequency Structural-Acoustic Problems". In: *Applied Acoustics* 43 (1994), pp. 271–294 (cit. on p. 7).

[12] Z. Bai. "Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems". In: *Applied Numerical Mathematics* 43.1-2 (2002), pp. 9–44 (cit. on p. 136).

[13] Z. Bai, R. Li, and Y. Su. “A unified Krylov projection framework for structure-preserving model reduction”. In: *Model Order Reduction: Theory, Research Aspects and Applications*. Ed. by W. Schilders, H. van der Vorst, and J. Rommes. Vol. 13. Mathematics in Industry. Springer-Verlag Berlin Heidelberg, 2008, pp. 75–93 (cit. on p. 146).

[14] E. Balmès. “Optimal Ritz Vectors for Component Mode Synthesis Using the Singular Value Decomposition”. In: *AIAA Journal* 34.6 (1996), pp. 1256–1260 (cit. on p. 32).

[15] E. Balmès. “Modes and Regular Shapes. How to Extend Component Mode Synthesis Theory”. In: *Proceedings of IMAC-XXVI: Conference & Exposition on Structural Dynamics*. 2008 (cit. on pp. 31, 32).

[16] P. K. Banerjee and R. Butterfield. *Boundary element methods in engineering science*. Vol. 17. McGraw-Hill London, 1981 (cit. on p. 7).

[17] K. J. Bathe. *Finite Element Procedures*. Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1995 (cit. on pp. 2, 10, 31, 60).

[18] K. Bathe. “Convergence of subspace iteration”. In: *Formulations and computational algorithms in finite element analysis* (1977), pp. 575–598 (cit. on pp. 36, 169).

[19] K.-J. Bathe. *Solution methods for large generalized eigenvalue problems in structural engineering*. Tech. rep. Structural Engineering Laboratory, University of California, Berkeley, 1971 (cit. on p. 22).

[20] P. Benner, J.-R. Li, and T. Penzl. “Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems”. In: *Numerical Linear Algebra with Applications* 15.9 (2008), pp. 755–777. ISSN: 1099-1506. DOI: `10.1002/nla.622`. URL: `http://dx.doi.org/10.1002/nla.622` (cit. on p. 157).

[21] J. Bennighof and R. Lehoucq. “An automated multilevel substructuring method for eigenspace computation in linear elastodynamics”. In: *SIAM Journal on Scientific Computing* 25.6 (2004), pp. 2084–2106 (cit. on p. 137).

[22] G. Berkooz, P. Holmes, and J. Lumley. “The proper orthogonal decomposition in the analysis of turbulent flows”. In: *Annual Review of Fluid Mechanics* 25 (1993), pp. 539–575 (cit. on p. 136).

[23] B. Besselink, U. Tabak, A. Lutowska, N. Van de Wouw, H Nijmeijer, D. J. Rixen, M. Hochstenbach, and W. Schilders. “A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control”. In: *Journal of Sound and Vibration* 332.19 (2013), pp. 4403–4422 (cit. on pp. 5, 71, 135).

[24] A. Bobillot and E. Balmès. “Iterative techniques for eigenvalue solutions of damped structures coupled with fluids”. In: *43rd AIAAASMEASCEAHSASC Structures, Structural Dynamics, and Materials Conference, Denver, Colorado*. 2002 (cit. on pp. 22, 29).

[25] C. A. Brebbia. *The boundary element method for engineers.* BOOK. Pentech press, 1980 (cit. on p. 2).

[26] C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel. *Boundary element techniques: theory and applications in engineering.* Springer Science & Business Media, 2012 (cit. on pp. 2, 7).

[27] E. Brechlin and L. Gaul. "Two methodological improvements for component mode synthesis". In: *25. International Seminar on Modal Analysis, Leuven.* 2000 (cit. on p. 103).

[28] Bruel&Kjaer. *Vehicle noise vibration and harshness development.* `https://www.bksv.com/en/Applications/vehicle-noise-vibration-and-harshness-development`. Accessed on 15-11-2019. 2019 (cit. on p. 1).

[29] Y. Chahlaoui, K. Gallivan, A. Vandendorpe, and P. Van Dooren. "Model reduction of second-order systems". In: *Dimension Reduction of Large-Scale Systems.* Ed. by P. Benner, V. Mehrmann, and D. Sorensen. Vol. 45. Lecture Notes in Computational Science and Engineering. Springer-Verlag, Berlin Heidelberg, Germany, 2005, pp. 149–172 (cit. on p. 158).

[30] H. C. Chen and R. L. Taylor. "Vibration analysis of fluid–solid systems using a finite element displacement formulation". In: *International Journal for Numerical Methods in Engineering* 29.4 (1990), pp. 683–698 (cit. on p. 7).

[31] R. D. Ciskowski and C. A. Brebbia. *Boundary element methods in acoustics.* Springer, 1991 (cit. on p. 7).

[32] C.Lanczos. "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators". In: *Journal of Research of the National Bureau of Standards* 45(4) (1950), pp. 255–282 (cit. on p. 46).

[33] R. D. Cook, M. E. Plesha, D. S. Malkus, and R. J. Witt. *Concepts and applications of finite element analysis.* John Wiley and Sons, 2002 (cit. on p. 11).

[34] A. Craggs. "The Transient Response of A Coupled Plate-Acoustic System Using Plate and Acoustic Finite Elements". In: *Journal of Sound and Vibration* 15(4) (1971), pp. 509–528 (cit. on pp. 2, 7, 8).

[35] R. Craig and A. Kurdila. *Fundamentals of structural dynamics.* 2nd. John Wiley & Sons, Hoboken, USA, 2006 (cit. on pp. 72, 90, 137).

[36] R. Craig Jr. "Coupling of substructures for dynamic analysis: an overview". In: *Proceedings of the 41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Atlanta, USA.* 2000 (cit. on pp. 53, 136–138).

[37] R. Craig Jr. and M. Bampton. "Coupling of substructures for dynamic analyses". In: *AIAA Journal* 6.7 (1968), pp. 1313–1319 (cit. on pp. 3, 71, 74, 75, 82, 135, 137).

[38] J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide.* Society for Industrial and Applied Mathematics, 2000 (cit. on p. 44).

[39] U. Desai and D. Pal. “A transformation approach to stochastic model reduction”. In: *IEEE Transactions on Automatic Control* AC-29.12 (1984), pp. 1097–1100 (cit. on p. 152).

[40] W. Desmet and D. Vandepitte. *Advanced Numerical Acoustics*. Course notes ISAAC 2005: Advanced Numerical Acoustics. 2005 (cit. on p. 7).

[41] J. Dickens, J. Nakagawa, and M. Wittbrodt. “A critique of mode acceleration and modal truncation augmentation methods for modal response analysis”. In: *Computers & Structures* 62.6 (1997), pp. 985–998 (cit. on pp. 27, 28, 142).

[42] J. Dickens and A. Stroeve. “Modal truncation vectors for reduced dynamic substructure models”. In: *Proceedings of the 41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Atlanta, USA*. 2000 (cit. on pp. 28, 142, 156).

[43] J. Dickens and E. Wilson. *Numerical Methods for Dynamic Substructure Analysis*. Tech. rep. UBC/EERC-80/120. Earthquake Eng. Res. Center, University of California, Berkeley, 1980 (cit. on p. 156).

[44] D. Enns. “Model reduction with balanced realizations: an error bound and a frequency weighted generalization”. In: *Proceedings of the 23rd IEEE Conference on Decision and Control, Las Vegas, USA*. 1984, pp. 127–132 (cit. on pp. 135, 147, 151, 152).

[45] T. Ericsson and A. Ruhe. “The Spectral Transformation Lanczos Method for the Numerical Solution of Large Sparse Generalized Symmetric Eigenvalue Problems”. In: *Mathematics of Computation* Vol. 35, No. 152 (1980), pp. 1251–1268 (cit. on p. 59).

[46] G. C. Everstine. “A symmetric potential formulation for fluid-structure interaction”. In: *Journal of Sound and Vibration* 79(1) (1981), pp. 157–160 (cit. on pp. 15, 44).

[47] D. J. Ewins. *Modal Testing: Theory, Practice and Application*. 2nd. Research Studies Press, 2000 (cit. on p. 39).

[48] F. J. Fahy and P. Gardonio. *Sound and structural vibration: radiation, transmission and response*. Elsevier, 2007 (cit. on pp. 12, 21, 24).

[49] C. Farhat and D. Rixen. “Linear Algebra”. In: *Encyclopedia of Vibration*. Ed. by S. G. Braun. Elsevier, 2001, pp. 710–720 (cit. on pp. 30, 49, 55, 115).

[50] P. Feldmann and R. Freund. “Efficient linear circuit analysis by Pade approximation via the Lanczos process”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 14.5 (1995), pp. 639–649 (cit. on pp. 136, 143).

[51] C. Felippa and R. Ohayon. “Mixed variational formulation of finite element analysis of acoustoelastic/slosh fluid-structure interaction”. In: *Journal of Fluids and Structures* 4(1) (1990), pp. 35–57 (cit. on p. 81).

[52] C. A. Felippa. “Symmetrization of the contained compressible-fluid vibration eigenproblem”. In: *Communications in Applied Numerical Methods* 1 (1985), pp. 241–247 (cit. on pp. 34, 42–44, 49, 81, 167).

[53] C. A. Felippa. “Symmetrization of coupled eigenproblems by eigenvector augmentation”. In: *Communications in Applied Numerical Methods* 4(4) (1988), pp. 561–563 (cit. on pp. 34, 49, 81).

[54] K. Fernando and H. Nicholson. “Singular perturbational model reduction of balanced systems”. In: *IEEE Transactions on Automatic Control* AC-27.2 (1982), pp. 466–468 (cit. on p. 151).

[55] J. Fish and T. Belytschko. *A first course in finite elements*. John Wiley & Sons New York, 2007 (cit. on pp. 2, 8, 10–12, 91).

[56] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal. “An implementation of the look-ahead Lanczos algorithm for non-hermetian matrices”. In: *Siam J. Sci. Comput.* 14(1) (1993), pp. 137–158 (cit. on pp. 44, 47).

[57] R. Freund. “Krylov-subspace methods for reduced-order modeling in circuit simulation”. In: *Journal of Computational and Applied Mathematics* 123.1-2 (2000), pp. 395–421 (cit. on p. 143).

[58] R. Freund. “Model reduction methods based on Krylov subspaces”. In: *Acta Numerica* 12 (2003), pp. 267–319 (cit. on p. 136).

[59] R. Freund. “SPRIM: structure-preserving reduced-order interconnect macromodeling”. In: *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*. 2004, pp. 80–87 (cit. on pp. 142, 146).

[60] R. Freund and P. Feldmann. “Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm”. In: *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, San Jose, USA*. 1996, pp. 280–287 (cit. on p. 146).

[61] K. Gallivan, E. Grimme, and P. Van Dooren. “Model reduction of large-scale systems: rational Krylov versus balancing techniques”. In: *Error Control and Adaptivity in Scientific Computing*. Ed. by H. Bulgak and C. Zenger. Kluwer Academic, 1999, pp. 177–190 (cit. on p. 136).

[62] C. F.V. L. Gene H. Golub. *Matrix Computations, 3rd Edition*. The John Hopkins University Press, 1996 (cit. on pp. 3, 56, 103).

[63] M. Géradin and D. J. Rixen. *Mechanical Vibrations, Theory and Application to Structural Dynamics*. Third. Chichester: Wiley, 2015 (cit. on pp. 16, 22, 25, 30, 36, 42, 47, 51, 53, 56, 72, 79, 83, 84, 86, 114, 135, 138, 141, 153, 169).

[64] G. Gladwell. “A finite element method for acoustics(Acoustic problems formulated and solved by finite element method of variational calculus, using both force and displacement procedures)”. In: *CONGRES INTERNATIONAL D’ACOUSTIQUE, 5 TH, LIEGE, BELGIUM*. Vol. 7. 14. 1965 (cit. on p. 7).

[65] K. Glover. “All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds”. In: *International Journal of Control* 39.6 (1984), pp. 1115–1193 (cit. on pp. 135, 147, 149, 151, 152).

[66] R. Grimes, J. Lewis, and H. Simon. “A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems”. In: *SIAM Journal on Matrix Analysis and Applications* 15.1 (1994), pp. 228–272 (cit. on p. 58).

[67] E. Grimme. “Krylov projection methods for model reduction”. PhD thesis. University of Illinois at Urbana-Champaign, USA, 1997 (cit. on pp. 136, 145, 146).

[68] P. G.Sandberg. “A symmetric finite element formulation for acoustic fluid-structure interaction analysis”. In: *Journal of Sound and Vibration* 123(3) (1988), pp. 507–515 (cit. on pp. 8, 15).

[69] S. Gugercin and A. Antoulas. “A survey of model reduction by balanced truncation and some new results”. In: *International Journal of Control* 77.8 (2004), pp. 748–766 (cit. on pp. 136, 152).

[70] S. Gugercin, D. Sorensen, and A. Antoulas. “A modified low-rank Smith method for large-scale Lyapunov equations”. In: *Numerical Algorithms* 32 (1 2003), pp. 27–55. ISSN: 1017-1398. URL: `http://dx.doi.org/10.1023/A:1022205420182` (cit. on p. 157).

[71] M. H. Gutknecht. “A completed theory of the unsymmetric Lanczos process and related algorithms, part II”. In: *Siam J. Matrix Anal. Appl.* 15(1) (1994), pp. 15–58 (cit. on p. 47).

[72] R. Guyan. “Reduction of mass and stiffness matrices”. In: *AIAA Journal* 3(2) (1965), p. 380 (cit. on p. 75).

[73] P. Heres. “Robust and efficient Krylov subspace methods for model order reduction”. PhD thesis. Eindhoven University of Technology, the Netherlands, 2005 (cit. on p. 143).

[74] J Herrmann, M Junge, and L Gaul. “Vibroacoustic Response of Flexible Car Components”. In: *Computer Modeling in Engineering & Sciences(CMES)* 86.6 (2012), pp. 487–504 (cit. on p. 72).

[75] J. Herrmann, M. Maess, and L. Gaul. “Substructuring including interface reduction for efficient vibro-acoustic simulation of fluid filled piping systems”. In: *Mechanical Systems and Signal Processing* 24 (2010), pp. 153–163 (cit. on pp. 72, 79, 111).

[76] M. R. Hestenes and E. Stiefel. “Methods of Conjugate Gradients for Solving Linear Systems”. In: *J. Res. Natl. Bur. Stand.* 49 (1952), pp. 409–436 (cit. on p. 83).

[77] *HSL(2013). A collection of Fortran codes for large scale scientific computation. http://www.hsl.rl.ac.uk* (cit. on p. 60).

[78] T. J. R. Hughes. *The finite element method : linear static and dynamic finite element analysis.* Englewood Cliffs, N.J. Prentice-Hall International, 1987 (cit. on p. 2).

[79] W. Hurty. “Vibrations of structural systems by component mode synthesis”. In: *ASCE Journal of the Engineering Dynamics Division* 86.EM4 (1960), pp. 51–69 (cit. on pp. 71, 137).

[80] W. Hurty. “Dynamic analysis of structural systems using component modes”. In: *AIAA Journal* 3.4 (1965), pp. 678–685 (cit. on pp. 71, 135, 137).

[81] U. Ingard. *Notes on acoustics*. Jones & Barlett Learning, 2008 (cit. on p. 11).

[82] *Intel(R) Fortran Compiler XE 14.0 Update 1 for Linux** (cit. on p. 60).

[83] *Intel(R) Math Kernel Library 11.1 Update 1 for Linux** (cit. on p. 60).

[84] B. M. Irons. “Role of part inversion in fluid-structure problems with mixed variables”. In: *AIAA J* 8(3) (1969), p. 568 (cit. on p. 44).

[85] M. Junge. “Model Reduction Methods for FE-BE Coupling Applied to Vibro-Acoustic Simulations and Experimental Validation”. PhD thesis. University of Stuttgart, Institute of Applied and Experimental Mechanics, 2009 (cit. on pp. 32, 34).

[86] G. Kerschen, J.-C. Golinval, A. Vakakis, and L. Bergman. “The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview”. In: *Nonlinear Dynamics* 41.1 (2005), pp. 147–169 (cit. on p. 136).

[87] S. Kim, B. Anderson, and A. Madievski. “Error bound for transfer function order reduction using frequency weighted balanced truncation”. In: *Systems & Control Letters* 24.3 (1995), pp. 183–192 (cit. on p. 152).

[88] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders. *Fundamentals of Acoustics*. 4th edition. Wiley, 1999 (cit. on p. 11).

[89] K.J.Bathe. “The subspace iteration method - Revisited”. In: *Computers & Structures* (2012) (cit. on p. 36).

[90] D. de Klerk D. Rixen S. Voormeeren. “General Framework for Dynamic Substructuring: History, Review and Classification of Techniques”. In: *AIAA Journal* 46, no.5 (2008), pp. 1169–1181 (cit. on pp. 3, 88, 90, 92, 124, 136, 137).

[91] D. C. Lay, S. R. Lay, and J. McDonald. *Linear Algebra and Its Applications*. 5th Edition. Pearson Education Limited, 2016 (cit. on p. 16).

[92] R. B. Lehoucq and D. C. Sorensen. “Deflation Techniques for an Implicitly Restarted Arnoldi Iteration”. In: *SIAM. J. Matrix Anal. & Appl.* 17(4) (1996), pp. 789–821 (cit. on pp. 44, 48).

[93] Y. Leung. “Fast reponse method for undamped structures”. In: *Engineering Structures* 5.2 (1983), pp. 141–149 (cit. on p. 156).

[94] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu. “Proper orthogonal decomposition and its applications - Part I: Theory”. In: *Journal of Sound and Vibration* 252.3 (2002), pp. 527–544 (cit. on p. 136).

[95] H.-L. Liew and P. Pinsky. “Matrix-Padé via Lanczos solutions for vibrations of fluid-structure interaction”. In: *International Journal for Numerical Methods in Engineering* 84.10 (2010), pp. 1183–1204. ISSN: 1097-0207. DOI: `10.1002/nme.2936`. URL: `http://dx.doi.org/10.1002/nme.2936` (cit. on p. 142).

[96] Y. Liu and B. Anderson. “Singular perturbation approximation of balanced systems”. In: *International Journal of Control* 50.4 (1989), pp. 1379–1405 (cit. on p. 151).

[97] Z.-S. Liu, S.-H. Chen, and Y.-Q. Liu W.-T.and Zhao. “An accurate modal method for computing response to harmonic excitation”. In: *Modal Analysis: The International Journal of Analytical and Experimental Modal Analysis* 9.1 (1994), pp. 1–13 (cit. on p. 156).

[98] B. Lohmann and B. Salimbahrami. “Introduction to Krylov subspace methods in model order reduction”. In: *Methods and Applications in Automation.* Ed. by B. Lohmann and A. Gräser. Shaker Verlag, Aachen, 2003, pp. 1–13 (cit. on pp. 145, 146).

[99] J. Luo and H. C. Gea. “Modal Sensitivity Analysis of Coupled Acoustic-Structural Systems”. In: *Journal of Vibration and Acoustics* 119 (1997), pp. 545–550 (cit. on p. 175).

[100] R. MacNeal. “A Hybrid Method of Component Mode Synthesis”. In: *Computers&Structures* 1.4 (1971), pp. 581–601 (cit. on p. 3).

[101] R. MacNeal. “A hybrid method of component mode synthesis”. In: *Computers & Structures* 1.4 (1971), pp. 581–601 (cit. on pp. 71, 74, 137).

[102] M. Maess and L. Gaul. “Substructuring and model reduction of pipe components interacting with acoustic fluids”. In: *Mechanical Systems and Signal Processing* 20 (2006), pp. 45–64 (cit. on pp. 15, 72, 79, 103, 111).

[103] D. Meyer. “A fractional approach to model reduction”. In: *Proceedings of the American Control Conference, Atlanta, USA*. 1988, pp. 1041–1047 (cit. on p. 152).

[104] B. Moore. “Principal component analysis in linear systems - controllability, observability, and model reduction”. In: *IEEE Transactions on Automatic Control* AC-26.1 (1981), pp. 17–32 (cit. on pp. 135, 147–149).

[105] H. J. P. Morand and R. Ohayon. *Fluid structure interaction.* John Wiley and Sons, Masson, 1995 (cit. on pp. 2, 3, 7, 15, 44, 45).

[106] P. M. Morse and K. U. Ingard. *Theoretical acoustics.* Princeton university press, 1986 (cit. on p. 11).

[107] M.Petyt and S.P.Lim. “Finite element analysis of the noise inside a mechanically excited cylinder”. In: *International Journal for Numerical Methods in Engineering* 13 (1978), pp. 109–122 (cit. on p. 21).

[108] C. Mullis and R. Roberts. “Synthesis of minimum roundoff noise fixed point digital filters”. In: *IEEE Transactions on Circuits and Systems* CAS-23.9 (1976), pp. 551–562 (cit. on p. 147).

[109] D. J. Nefske and L. J. Howell. “Automobile interior noise reduction using finite element methods”. In: *SAE Transactions* (1978), pp. 1727–1737 (cit. on p. 7).

[110] N.S.Sehmi. *Large Order Structural Eigenanalysis techniques: algorithms for finite element systems*. Ed. by G.M.Bell. Ellis Horwood Series in Mathematics and its Applications, 1989 (cit. on pp. 22, 57, 58, 169, 172, 173).

[111] R. Ober. "Balanced parametrization of classes of linear systems". In: *SIAM Journal on Control and Optimization* 29.6 (1991), pp. 1251–1287 (cit. on p. 152).

[112] R. Ober and D. McFarlane. "Balanced canonical forms for minimal systems: A normalized coprime factor approach". In: *Linear Algebra and its Applications* 122-124 (1989), pp. 23–64 (cit. on p. 152).

[113] A. Odabasioglu, M. Celik, and L. Pileggi. "PRIMA: passive reduced-order interconnect macromodeling algorithm". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 17.8 (1998), pp. 645–654 (cit. on pp. 142, 143, 146, 158).

[114] R. Ohayon. "Reduced models for fluid-structure interaction problems". In: *International Journal for Numerical Methods In Engineering* 60 (2004), pp. 139–152 (cit. on p. 81).

[115] K.-J. Olson Lorraine G.and Bathe. "Analysis of Fluid-Structure interactions, a direct symmetric coupled formulation based on the fluid velocity potential". In: *Computers & Structures* 21(1/2) (1985), pp. 21–32 (cit. on p. 44).

[116] N. S. Ottosen and H. Petersson. *Introduction to the Finite Element Method*. eng. Prentice-Hall, 1992, p. 422. ISBN: 0-13-473877-2 (cit. on pp. 8, 11).

[117] M. Papadrakakis. *Solving large-scale problems in mechanics*. John Wiley & Sons, New York, USA, 1993 (cit. on p. 138).

[118] B. N. Parlett. *The Symmetric Eigenvalue Problem*. SIAM, 1998 (cit. on pp. 47, 49, 51, 56, 57, 173).

[119] B. N. Parlett. Personal Communication (cit. on p. 56).

[120] B. N. Parlett. "Reduction to tridiagonal form and minimal realizations". In: *Siam J. Matrix Anal. Appl.* 13(2) (1992), pp. 567–593 (cit. on p. 47).

[121] L. Pernebo and L. Silverman. "Model reduction via balanced state space representations". In: *IEEE Transactions on Automatic Control* AC-27.2 (1982), pp. 382–387 (cit. on pp. 147, 151).

[122] M. Petyt. *Introduction to finite element vibration analysis*. Cambridge University Press, Cambridge, United Kingdom, 1990 (cit. on p. 138).

[123] M. Petyt, J. Lea, and G. Koopmann. "A finite element method for determining the acoustic modes of irregular shaped cavities". In: *Journal of Sound and Vibration* 45.4 (1976), pp. 495–502 (cit. on pp. 7, 8).

[124] L. Pillage and R. Rohrer. "Asymptotic waveform evaluation for timing analysis". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 9.4 (1990), pp. 352–366 (cit. on pp. 136, 143).

[125] C. Rajakumar and C. R. Rogers. “The Lanczos Algorithm Applied To Unsymmetric Generalized Eigenvalue Problem”. In: *International Journal for Numerical Methods In Engineering* 32 (1991), pp. 1009–1026 (cit. on pp. 44, 45).

[126] J. W. S. Rayleigh. *The Theory of Sound*. Dover Publications, 1945 (cit. on pp. 135, 137).

[127] M. Rewieński and J. White. “A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 22.2 (2003), pp. 155–170 (cit. on p. 136).

[128] D. Rixen. “Generalized mode acceleration methods and modal truncation augmentation”. In: *Proceedings of the 42rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Seattle, USA*. 2001 (cit. on pp. 142, 156).

[129] D. Rixen. “A Lanczos procedure for efficient mode superposition in dynamic analysis”. In: *Proceedings of the 43st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Denver, USA*. 2002 (cit. on p. 156).

[130] D. Rixen. “High order static correction modes for component mode synthesis”. In: *Proceedings of the 5th World Congress on Computational Mechanics, Vienna, Austria*. 2002 (cit. on p. 156).

[131] D. Rixen. “Dual Craig-Bampton with enrichment to avoid spurious modes”. In: *Proceedings of the IMAC-XXVII Conference & Exposition on Structural Dynamics, Orlando, USA*. 2009 (cit. on p. 156).

[132] S. Rubin. “Improved component-mode representation for structural dynamic analysis”. In: *AIAA Journal* 13.8 (1975), pp. 995–1006 (cit. on pp. 3, 71, 74, 137).

[133] Y. Saad. “The Lanczos Biorthogonalization Algorithm and Other Oblique Projection Methods for Solving Large Unsymmetric Systems”. In: *SIAM J. Numer. Anal.* 19 (1982), pp. 485–506 (cit. on p. 45).

[134] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, Philadelphia, USA, 2003 (cit. on pp. 3, 143, 146).

[135] Y. Saad. “Variations on Arnoldi’s method for computing the eigenelements of large unsymmetric matrices”. In: *Linear Algebra and its Applications* 34 (1980), pp. 269–295 (cit. on p. 48).

[136] Y. Saad. *Numerical Methods for Large Eigenvalue Problems, Revised Edition*. SIAM, 2011 (cit. on pp. 3, 44, 48, 50, 63, 68).

[137] G. Sandberg. “A new strategy for solving fluid-structure problems”. In: *International Journal for Numerical Methods in Engineering* 38 (1995), pp. 357–370 (cit. on p. 22).

[138] G. Sandberg and R. Ohayon, eds. *Computational aspects of structural acoustics and vibration.* CISM, Udine: International Centre for Mechanical Sciences, CISM Courses and Lectures 505, 2008 (cit. on p. 2).

[139] J. Scherpen. “Balancing for nonlinear systems”. In: *Systems & Control Letters* 21 (1993), pp. 143–153 (cit. on p. 136).

[140] J.-F. Sigrist and S. Garreau. “Dynamic analysis of fluid–structure interaction problems with modal methods using pressure-based fluid finite elements”. In: *Finite Elements in analysis and design* 43.4 (2007), pp. 287–300 (cit. on p. 15).

[141] H. D. Simon. “Analysis of the Symmetric Lanczos Algorithm with Reorthogonalization Methods”. In: *Linear Algebra and its Applications* 61 (1984), pp. 101–131 (cit. on pp. 56, 57).

[142] H. D. Simon. “The Lanczos Algorithm with Partial Reorthogonalization”. In: *Mathematics of Computation* 42.165 (1984), pp. 115–142 (cit. on pp. 56–58, 66, 173).

[143] L. Sirovich. “Turbulence and the dynamics of coherent structures, part I: Coherent structures”. In: *Quarterly of Applied Mathematics* 45.3 (1987), pp. 561–571 (cit. on p. 136).

[144] B. B. Smida, R. Majed, N. Bouhaddi, and M. Ouisse. “Investigations for a model reduction technique of fluid–structure coupled systems”. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 226(1) (2012) (cit. on pp. 22, 29).

[145] B. Stroustrup. *The C++ programming language.* Pearson Education, 2013 (cit. on pp. 17, 38, 60).

[146] F. C. Sylvain Boily. “The vibroacoustic response of a cylindrical shell structure with viscoelastic and poroelastic materials”. In: *Applied Acoustics* 58 (1999), pp. 131–152 (cit. on p. 28).

[147] D. Systemes. *ABAQUS Inc. 6.14, Theory Guide* (cit. on p. 44).

[148] U. Tabak and D. Rixen. “Reduced Iterative Correction Algorithm for Coupled Vibro-acoustic Problems”. In: *International Conference on Noise and Vibration Engineering, ISMA 2010, Leuven, Belgium.* 2010, pp. 4685–4695 (cit. on pp. 4, 21).

[149] U. Tabak and D. J. Rixen. “Globally enriched substructuring techniques for vibro-acoustic simulation”. In: *Linking Models and Experiments, Volume 2. Conference Proceedings of the Society for Experimental Mechanics Series.* Ed. by T. Proulx. Springer, New York, NY, 2011 (cit. on p. 71).

[150] U. Tabak and D. J. Rixen. “Spectrally Preconditioned Iterative Reduced Correction Algorithm for Vibro-acoustic Problems, SPIRCA.” In: ed. by R. Allemang, J. De Clerck, C. Niezrecki, and J Blough. Topics in Modal Analysis II, Volume 6. Conference Proceedings of the Society for Experimental Mechanics Series. Springer, New York, NY, 2012 (cit. on pp. 4, 21, 29).

[151] U. Tabak and D. J. Rixen. “vibro-Lanczos, a symmetric Lanczos solver for vibro-acoustic simulations”. In: *International Journal for Numerical Methods in Engineering* 107.4 (2016), pp. 290–311 (cit. on pp. 5, 43, 79, 80).

[152] M. Tournour and N. Atalla. “Pseudostatic corrections for the forced vibroacoustic response of a structure-cavity system”. In: *Journal of the Acoustical Society of America* 107(5) (2000), pp. 2379–2386 (cit. on pp. 22, 28, 38).

[153] Q. H. Tran, M. Ouisse, and N. Bouhaddi. “A robust component mode synthesis method for stochastic damped vibroacoustics”. In: *Mechanical Systems and Signal Processing* 24(1) (2010), pp. 164–181 (cit. on p. 29).

[154] A. C. Ugural and S. K. Fenster. *Advanced mechanics of materials and applied elasticity*. Pearson Education, 2011 (cit. on pp. 9, 10).

[155] A. Vandendorpe and P. Van Dooren. “Model reduction of interconnected systems”. In: *Model Order Reduction: Theory, Research Aspects and Applications*. Ed. by W. Schilders, H. van der Vorst, and J. Rommes. Vol. 13. Mathematics in Industry. Springer-Verlag Berlin Heidelberg, 2008, pp. 305–321 (cit. on p. 146).

[156] H. I. ven der Veen and K. Vuik. “Bi-Lanczos with Partial Orthogonalization”. In: *Computers & Structures* 7 (1995), pp. 605–613 (cit. on p. 66).

[157] E. Wilson, M.-W. Yuan, and J. Dickens. “Dynamic analysis by direct superposition of Ritz vectors”. In: *Earthquake Engineering & Structural Dynamics* 10.6 (1982), pp. 813–821 (cit. on pp. 137, 156).

[158] J. A. Wolf Jr. “Modal Synthesis for Combined Structural-Acoustic Systems”. In: *AIAA* 15.5 (1977), pp. 741–745 (cit. on p. 21).

[159] Q. Ye. “A breakdown-free variation of the nonsymmetric Lanczos algorithms”. In: *Mathematics of Computation* 62(205) (1994), pp. 179–207 (cit. on p. 47).

[160] Y.-Q. Zhao, S.-H. Chen, S. Chai, and Q.-W. Qu. “An improved modal truncation method for responses to harmonic excitation”. In: *Computers & Structures* 80.1 (2002), pp. 99–103 (cit. on p. 156).

[161] I. H. Zheng-Dong Ma. “Sensitivitity Analysis Methods for coupled structural-acoustic systems part I : Modal sensitivities”. In: *AIAA Journal* 29(10) (1991), pp. 1787–1795 (cit. on pp. 16, 30, 47, 175).

[162] K. Zhou, J. Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall, Upper Saddle River, USA, 1996 (cit. on p. 148).

Curriculum Vitae

Umut TABAK

08-06-1979	Born in Izmir, Turkey.

Education

1997–2001	BSc. Mechanical Engineering, Summa cum laude, Dokuz Eylul University, Izmir, Turkey
2001–2005	MSc. Mechanical Engineering, Teaching Assistant Dokuz Eylul University, Izmir, Turkey
2008–2020	PhD. Mechanical Engineering, 2013–Present, guest researcher Delft University of Technology, Delft, The Netherlands *Thesis:* Methods for Efficient Analysis of Vibro-Acoustic Problems *Promotors:* Prof. dr. D. J. Rixen and Prof. dr. P. G. Steeneken

Awards

2001	Graduation with distinction, 1/149 in class, 3/694 at faculty
2005	Marie Curie Early Stage Research Scholarship Katholieke Universiteit Leuven, Belgium

Work Experience

2005–2008	Research Engineer, Dept. of Mechanical Engineering Katholieke Universiteit Leuven, Belgium
2013–present	Senior Product Development Engineer, Sonion, The Netherlands

List of Publications

Journal Articles

3. **Umut Tabak**, Daniel J. Rixen, *Locally symmetric Craig-Bampton Technique for Vibro-Acoustic Problems*, In preparation for Int. J. Numer. Meth. Engng.

2. **Umut Tabak**, Daniel J. Rixen, *vibro-Lanczos, A symmetric Lanczos solver for vibro-acoustic simulations*, Int. J. Numer. Meth. Engng; 107:290-311, 2016.

1. B. Besselink, **U. Tabak**, A. Lutowska, N. van de Wouw, H. Nijmeijer, D. J. Rixen, W. H. A. Schilders, *A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control*, Journal of Sound and Vibration, 332(19):4403-4422, 2013.

Conference Papers

3. **U. Tabak**, D. J. Rixen, *A Spectrally Preconditioned Iterative Reduced Correction Algorithm for Vibro-acoustic Problems*, Proceedings of International Modal Analysis Conference, IMAC XXX, Jacksonville, USA, 2012.

2. **U. Tabak**, D. J. Rixen, *Globally enriched substructuring techniques for vibro-acoustic simulation*, Proceedings of International Modal Analysis Conference, IMAC XXIX, Jacksonville, USA, 2011.

1. **U. Tabak**, D. J. Rixen, *Reduced iterative correction algorithm for coupled interior vibro-acoustic problems*, Proceedings of International Conference on Noise and Vibration engineering, ISMA 2010, p 4685-4695, Leuven, Belgium.

Acknowledgements

After a long struggle with many ups and downs of a long PhD. period coupled with a full-time job aside, it is a pleasure that I could write this section eventually.

First of all, I would like to express my deepest appreciation for my family. During the long years I spent away from home, they always supported me to mentally stay strong and focus on my goals in life, even if, they missed me a lot back at home. Their presence and support is more valuable to me than anything in this world.

I believe I have had a rewarding life-time experience and learned a lot during my PhD. I became a numerical expert and independent researcher. Having said this, I have not come to this point alone. At the forefront, I would like to express my sincere gratitude for my promotor Daniel J. Rixen. His continuous support and challenging supervision changed my academic vision. Even if he had to leave permanently to Germany, he did not lose interest in my research and supported me in his busy schedule. I call him the *swiss-army-knife* professor where he could, almost always, find solutions to difficult problems I have encountered during my research. I also admire his optimism, energy and hard-working spirit. I also deeply hope that our research collaboration will last after my PhD. So, thank you Daniel, for the support and feedback during all these years.

The other important driving force that pushed me to the finish line came from Prof. Peter G. Steeneken. I will never forget the day we met and discussed on possible ways how I can still finish my PhD. If his push and support was not there, it might have been that I would not be able to write these lines. Therefore, I also would like to thank him and will always appreciate his time and effort.

I also would like to deeply thank Prof. Just Herder, Gaby Offermans and Lisette Janse for giving me the chance to use the department facilities by extending my *guest-researcher* status several times and allowing me to use the computational cluster of the faculty. Without their support, it would also not be possible to come this far.

Having a job aside and trying to finish a PhD. is a difficult task. During these years, I had a friend Nishant with whom we had carpooling for almost 4.5 years. He was also trying to finish his PhD. at EWI like me. And, we were supporting each other on busy times and after sleepless nights by driving to work alternately. We overcame many difficulties in time and drove almost 70000 km together on A4 between Delft and Hoofddorp. With his driving support, I could get a bit more sleep in the mornings and in the evenings on the way back home. Thanks a lot Nishant, you are a real friend to me.

The last but not the least, Mehmet Selman Tamer, with whom we met far from homeland and became very good friends. The dinners prepared by his wife Merve and the times we spent together after the dinners were always very entertaining and refreshing. I would like to thank you both for inviting me frequently so that I can have a *home-taste*.

As a last word, I would also like to thank all the members of the Precision and Microsystems Engineering Department (PME) at TUDelft.
Umut Tabak
Delft, February 2020