Dynamic Substructuring Methodologies for Integrated Dynamic Analysis of Wind Turbines

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op woensdag 7 november 2012 om 12:30 uur
door

Sven Niels VOORMEEREN

werktuigkundig ingenieur
geboren te Hellevoetsluis.

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. D.J. Rixen

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. D.J. Rixen,	Technische Universiteit Delft / TU München, promotor
Prof. dr. E. Balmès,	Arts et Métiers ParisTech, Frankrijk
Prof. dr. ir. A. de Boer,	Universiteit Twente
Prof. dr. ir. W. Desmet,	Katholieke Universiteit Leuven, België
Prof. Dr.–Ing. habil. L. Gaul,	Universität Stuttgart, Duitsland
Prof. dr. ir. L.J. Sluys,	Technische Universiteit Delft
Dr. ir. D-P. Molenaar,	Siemens Wind Power
Prof. dr. ir. A. van Keulen,	Technische Universiteit Delft, reservelid

Copyright © 2012 by Sven Voormeeren

– All rights reserved – No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without the prior permission of the author.

ISBN 978-94-91104-10-7

Printed by: Uitgeverij BOXPress

Author email: sven.voormeeren@siemens.com

Abstract

One of the great questions society is faced with today is how to sustain the current level of well-being for future generations. The tremendous increase in prosperity that mankind has seen since the industrial revolution has been mainly fueled by the abundant availability of cheap energy, in the form of coal and oil. As maintaining our current lifestyle would deplete these precious fossil resources within the reasonably near future, sustainable alternatives need to be developed sooner rather than later.

Over the last decades wind energy has manifested itself as one of the serious alternatives. With exponential growth rates seen in both the installed wind energy capacity and wind turbine size, the wind industry has achieved impressive cost reduction of wind generated electricity. In fact, for some for favorable locations on land wind energy is cost-competitive with conventional energy sources. However, subsidies are often still required to make wind energy economically viable, which is especially true for offshore wind energy. Hence, further cost reductions are required to truly benefit from the huge potential wind energy offers to generate “green” electricity on a large scale.

Since wind turbines are very complex machines that operate under continuous and intricate environmental dynamic loading, insight in the turbine’s structural dynamic behavior is essential in their design and operation. Traditionally, these dynamics are analyzed using aero-elastic simulation models. Such models account for all relevant dynamic phenomena (structural, aero-, hydro- and controller dynamics), but the underlying structural models are very simple. Loads obtained from these simulations are subsequently applied statically to detailed component finite element models, thereby assuming that these components only contribute quasi-statically to the global dynamic behavior.

In the quest for further cost reductions, lots of efforts are spent in optimizing wind turbine designs. These optimized designs generally tend to introduce more structural flexibility. As a result, dynamic effects get more pronounced such that local dynamic behavior becomes relevant. Due to their simplified structural models these dynamics cannot be predicted by the aero-elastic simulations. Hence, the current sequential analysis procedure breaks down.

To overcome this, a methodology is developed in this thesis that enables integrated dynamic analysis of wind turbines with a high level of structural detail. This methodology is based on the concept of dynamic substructuring (DS), which consists in analyzing a system’s structural dynamics in a componentwise fashion. Doing so has many advantages: models can be tailored to the characteristics of the components, modeled and measured parts can be combined, the total model can be efficiently re-analyzed if some components are changed, and so on.

In part I of this thesis, several theoretical methods are developed to enhance the practical useability of the DS methodology. The cornerstones of the DS approach are component model reduction techniques, which express detailed dynamic models in terms of their most dominant behavior. This leads to compact yet accurate component models, thereby greatly improving the computational efficiency of subsequent simulations. These methods are investigated in detail and a generalization of existing methods is proposed. Various assembly techniques are thereafter developed in order to obtain the reduced dynamic model of the total structure, regardless of the interface representations of its components. If necessary, this model can be further compacted through interface reduction, which is also addressed. An important issue that is treated thereafter is the accuracy of this reduced model. Using error estimation methods, it is shown how the accuracy can be estimated without analyzing the unreduced model. Furthermore, these error estimates enable adaptive model reduction. Finally, part I is concluded by an investigation on how to efficiently update reduced component models that are subjected to (parametric) design modifications. Preconditioned iterative algorithms are proposed to make optimal use of the available information from the nominal model.

Part II of this thesis applies the methods developed in part I to representative wind turbine engineering problems in the form of three case studies. Firstly, it is shown how (interface) model reduction and assembly techniques can be used to obtain a very compact model of a yaw system of a multi-megawatt wind turbine. The resulting model proves to be an accurate representation of the full model at only a fraction of its original size and computational cost. Secondly, a reduced model of a wind turbine bedframe structure is subjected to parametric design changes. It is demonstrated how the model can be efficiently updated, with a performance gain of around a factor three with respect to recomputation for typical modifications. As such it can be used effectively in a practical design setting. Finally, the modeling of an offshore wind turbine on a complex support structure is addressed. Using adaptive model reduction and error estimation, optimal models are created that can be used in aero-elastic simulations. For selected quantities, these models provide up to an order of magnitude better accuracy at the same model size as their uniformly reduced counterparts.

In conclusion, a modeling framework is developed based on the dynamic substructuring methodology, that is a suitable for performing detailed integrated dynamic analysis of wind turbines. It is a flexible and versatile methodology, that is able to provide thorough insight in the system's dynamics without compromising too much on computational efficiency. Therefore, the developed framework could prove a useful contribution to the design tools needed for developing innovative next-generation wind turbines: a small yet essential step in the elaborate path towards cost competitiveness of wind energy.

Samenvatting

Eén van de grootste uitdagingen van de moderne maatschappij is het in stand houden van het huidige niveau van welzijn voor toekomstige generaties. De ongekende groei in rijkdom die de mensheid heeft doorgemaakt sinds de industriële revolutie, is voornamelijk mogelijk gemaakt door de overvloedige beschikbaarheid van goedkope energie in de vorm van kolen en olie. Nu door het handhaven van onze levensstijl deze waardevolle fossiele grondstoffen binnen afzienbare tijd dreigen op te raken, is het van groot belang dat op korte termijn duurzame alternatieven worden gevonden.

Gedurende de afgelopen decennia heeft windenergie zich gemanifesteerd als één van de serieuze alternatieven. Substantiële kostenverlagingen zijn gerealiseerd door met name de exponentiële groei van zowel de geïnstalleerde windenergiecapaciteit als de grootte van windturbines. Op gunstige locaties op land zijn de kosten van windenergie daarom al concurrerend met die van conventionele energiecentrales. Echter, in veel gevallen zijn nog altijd subsidies benodigd om windenergie economisch haalbaar te maken. Dit is in het bijzonder het geval voor offshore windenergie. Verdere kostenreducties zijn daarom noodzakelijk om op grote schaal te profiteren van het gigantische potentieel dat de wind biedt tot het opwekken van "groene" energie.

Aangezien windturbines zeer complexe machines zijn die onder voortdurende dynamische belasting moeten functioneren, is inzicht in het structurele dynamische gedrag cruciaal voor het ontwerp en de werking van deze machines. Traditiegetrouw wordt deze dynamica geanalyseerd met behulp van zogenaamde aero-elastische simulatiemodellen. Deze modellen houden rekening met alle relevante dynamische fenomenen (structuur-, aero-, hydro- en regelaardynamica), maar zijn gebaseerd op zeer grove structuurmodellen. De belastingen die met deze simulaties worden berekend worden daarna op een statische manier toegepast op gedetailleerde eindige elementen componentmodellen, waarbij wordt aangenomen dat deze componenten slechts quasi-statisch bijdragen aan het globale dynamische gedrag.

In de voortdurende zoektocht naar kostenverlagingen wordt gestreefd naar verdere optimalisatie van het windturbine ontwerp. Deze geoptimaliseerde ontwerpen zorgen in het algemeen voor meer structurele flexibiliteit, waardoor dynamische effecten versterken en lokaal dynamisch gedrag kan optreden. Door de simpele structurele modellen die worden gebruikt in de aero-elastische simulaties, kan deze dynamica echter niet voorspeld worden in de huidige belastingsberekeningen. Hierdoor is de tot nu toe gebruikte stapsgewijze structurele analyseprocedure niet langer geldig.

Om dit probleem op te lossen is in dit proefschrift een methodologie ontwikkeld die het mogelijk maakt geïntegreerde dynamische analyse uit te voeren met een hoge mate van structurele gedetailleerdheid. Deze methodologie is gebaseerd op het concept van "dynamisch substructureren" (DS), wat inhoudt dat de dynamica van het systeem geanalyseerd wordt op basis van de dynamica van haar componenten. Deze aanpak heeft vele voordelen: modellen kunnen aangepast worden aan de eigenschappen van de componenten, experimenteel en

numeriek bepaalde componentmodellen kunnen worden gecombineerd, heranalyse van het totale model kan efficiënt worden uitgevoerd wanneer enkele componenten worden veranderd, enzovoorts.

In deel I van dit proefschrift worden verschillende theoretische methoden ontwikkeld om de praktische bruikbaarheid van de DS methodologie te verbeteren. De hoekstenen van het DS concept worden gevormd door component modelreductie technieken, waarmee gedetailleerde dynamische modellen beschreven kunnen worden in termen van hun meest dominante gedrag. Dit leidt tot compacte en nauwkeurige componentmodellen die veel sneller kunnen worden doorgerekend. Deze methoden worden in detail onderzocht en een generalisatie van bestaande methoden wordt voorgesteld. Verschillende assemblagetechnieken worden daarna beschouwd om zo het gereduceerde dynamische model van het totale systeem op te bouwen, onafhankelijk van de interface representatie van de componentmodellen. Ook wordt besproken hoe, indien nodig, dit model verder kan worden verkleind door het gebruik van interface reductietechnieken. Een belangrijk aspect dat daarna wordt behandeld is de nauwkeurigheid van dit gereduceerde model. Met behulp van foutschattingsmethoden wordt laten zien hoe de nauwkeurigheid van het gereduceerde model kan worden geschat zonder het volledige model door te rekenen. Deze foutschattingen maken het bovendien mogelijk om modelreductie op een adaptieve manier toe te passen. Tenslotte wordt deel I afgesloten met een onderzoek naar het efficiënt updaten van gereduceerde modellen die onderhevig zijn aan parametrische ontwerpveranderingen. Hiertoe worden iteratieve oplossingsmethoden ontwikkeld om zo optimaal mogelijk gebruik te maken van de beschikbare informatie van het nominale model.

Deel II van dit proefschrift behandelt de toepassing van de methodologie van deel I op ontwerpproblemen uit de windturbine praktijk. Dit wordt gedaan in de vorm van drie casestudies. Ten eerste wordt geïllustreerd hoe (interface) modelreductie en assemblagetechnieken toegepast kunnen worden om een zeer compact model te creëren van een kruisysteem van een moderne windturbine. Dit model blijkt een nauwkeurige representatie van het volledige model, voor slechts een fractie van de originele modelgrootte en rekentijd. Ten tweede wordt een gereduceerd model van een windturbine bedplaat onderworpen aan parametrische ontwerpmodificaties. Er wordt gedemonstreerd hoe dit model efficiënt kan worden geupdate; voor realischtische modificaties wordt een prestatieverbetering behaald ten opzichte van volledige herberekening van ongeveer een factor drie. Hierdoor kunnen gereduceerde modellen effectief gebruikt worden in een praktische ontwerpomgeving. Tenslotte wordt gekeken naar de modellering van een offshore windturbine op een complexe fundatie. Door het gebruik van foutschattingsmethoden en adaptieve modelreductie worden van deze turbine optimale gereduceerde modellen gemaakt voor gebruik in belastingsberekeningen. Voor de geselecteerde variabelen is de nauwkeurigheid van deze modellen tot een ordegrootte beter dan die van uniform gereduceerde modellen.

Al met al kan worden geconcludeerd dat in dit proefschrift een modelleringsraamwerk is ontwikkeld op basis van dynamisch substructureren, dat gedetailleerde en geïntegreerde dynamische analyse van windturbines mogelijk maakt. Het ontwikkelde raamwerk vormt daarmee een nuttige bijdrage aan de ontwerpgereedschappen die nodig zijn om de volgende generatie windturbines te ontwikkelen: een kleine doch essentiële stap richting rendabele windenergie.

Acknowledgements

Over the last four years I had the opportunity to perform my PhD research project in cooperation with Siemens Wind Power, investigating methods to predict vibrations of large wind turbines. These turned out to be four very exciting years, with a unique combination of scientific depth and practical application in the dynamic world of wind turbine engineering. Now the thesis is completed, the time has come to reflect and express my appreciation to those who supported this work along the way.

First of all I would like to thank prof. Daniel Rixen, who convinced me in 2007 to take the leap of faith and write our own PhD project proposal for the NWO Toptalent program. Although our subsidy request got rejected, I never regretted taking this step as it initiated the fruitful cooperation with Siemens. Daniel, you have been a source of inspiration throughout the project and I admire your passion for teaching, professional knowledge and positive spirit.

Secondly I would like to thank Siemens Wind Power for their financial support of this project and letting me perform this research in their company. In particular I am thankful to David Molenaar, who has been a great yet critical supporter of this project, always asking the right questions and safeguarding the balance between scientific content and practical applicability. David, the way you manage to involve (and be involved with) everyone in your department, both on a personal and professional level, is truly remarkable.

Furthermore I want to thank the colleagues at the university for the pleasant working environment, especially our engineering dynamics colleagues Corinne, Paul, Paolo and Dennis. Many thanks also go to my fellow PhDs for the great times we had at the conferences and mountainbike trips: Alexander, Caspar, Chris, Edwin, Kelvin, Maarten, Michael, Nico, Paul, Stephan, thanks!

Appreciation also goes out to the colleagues at Siemens Wind Power in Denmark for their support of and interest in the project. Furthermore, although they are now too many to name individually, my sincere gratitude is expressed to the colleagues at Siemens in The Hague. You provided an exceptional working atmosphere that is both fun and inspiring, and I am proud to remain part of your team in the future.

Special thanks must go out to “my” MSc students. Paul, Diederik and Bas: without your contributions this thesis would not have been what it is today. It has been a pleasure to work with you and I am thankful for the joy we had doing it, both during and outside office hours.

Last but certainly not least I would like to thank my family and friends. Pap en mam, if it weren’t for your continuous support and all the opportunities you gave me, I could have never gotten to this point. Finally I thank Nina for her loving and understanding, but most of all for simply always being there and making life pleasant.

Sven Voormeeren
The Hague, September 2012

Contents

Abstract iii

Samenvatting v

Acknowledgements vii

Nomenclature xi

1 Introduction 1
- 1.1 Wind Energy: For Free and Still Too Expensive? 1
- 1.2 Cost Reduction and the Role of Structural Dynamics 4
- 1.3 State-of-the-Art in Wind Turbine Dynamic Analysis 6
- 1.4 Towards Integrated Dynamic Analysis of Wind Turbines 7
- 1.5 Dynamic Substructuring 7
- 1.6 Thesis Objectives & Outline 8

PART I – Theory 13

2 Dynamic Substructuring & Component Model Reduction 15
- 2.1 Introduction to Dynamic Substructuring 15
- 2.2 Generic Formulation of Component Model Reduction 18
- 2.3 Reduction Basis Ingredients – Static Modes 21
- 2.4 Reduction Basis Ingredients – Vibration Modes 25
- 2.5 Classic Component Reduction Methods 28
- 2.6 The Mixed Craig-Bampton Method 40
- 2.7 Modal Truncation Augmentation 44
- 2.8 Summary 53

3 Assembly of Component Models 55
- 3.1 Introduction 55
- 3.2 General Framework for Component Assembly 56
- 3.3 Stiffness Assembly 56
- 3.4 Flexibility Assembly 62
- 3.5 Mixed Assembly 64
- 3.6 Assembly of Non-Conforming Meshes 68
- 3.7 Assembly with Additional Interface Physics 72
- 3.8 Decoupling of Component Models 77
- 3.9 Summary 83

4 Interface Reduction & Assembled System Analysis 85
- 4.1 Introduction 85

4.2 Options for Interface Reduction 86
4.3 Interface Rigidification 88
4.4 Modal Reduction of Interface Displacements 89
4.5 Modal Reduction of Interface Forces 92
4.6 Analysis of Assembled Model 94
4.7 Comparison & Correlation Methods 97
4.8 Summary 101

5 Error Estimation & Adaptive Model Reduction 103
5.1 Introduction 103
5.2 Aspects of Error Estimation 104
5.3 Alternative Reduced System Description 106
5.4 Error Estimation for Harmonic Solutions 109
5.5 Error Estimation for Time-Stepping Solutions 115
5.6 Error Estimation for Global Eigensolutions 117
5.7 Refinement Schemes for Adaptive Model Reduction 122
5.8 Summary 125

6 Updating of Component Reduction Bases 127
6.1 Introduction 127
6.2 Updating of Static Modes 130
6.3 Updating of Vibration Modes 140
6.4 Sequential Updating 155
6.5 Summary 157

PART II – Application to Wind Turbine Engineering 159

7 Dynamic Substructuring Analysis of a Yaw System 161
7.1 Introduction 161
7.2 System Description 162
7.3 Component Modeling – Bedplate 165
7.4 Component Modeling – Yaw Pads 167
7.5 Component Modeling – Tower Top & Yaw Ring 168
7.6 Component Modeling – Yaw Gearbox 169
7.7 Assembled Models & Analysis Results 173
7.8 Summary 182

8 Design Modification of a Bedframe Structure 185
8.1 Introduction 185
8.2 Structural Model & Design Modifications 186
8.3 Efficiency of Updating of Static Modes 189
8.4 Efficiency of Updating of Vibration Modes 194
8.5 Overall Efficiency of Basis Updating 200
8.6 Sequential Updating 201
8.7 Effectiveness of Different Reduction Bases 205
8.8 Summary 208

9 Optimal Reduced Models of an Offshore Wind Turbine 211
9.1 Introduction 211
9.2 System Description 214
9.3 Adaptive Reduction Applied to Offshore Wind Turbine Model 219
9.4 Modal Truncation Augmentation & Time Simulation 229
9.5 Summary 234

PART III – Conclusions & Recommendations 239

10 Conclusions & Recommendations 241
10.1 Conclusions 241
10.2 Recommendations 243

Bibiliography 247

PART IV – Appendices 265

A The Dynamic Substructuring Toolbox 267
A.1 Preparation Tool 268
A.2 Assembly Tool 269
A.3 Postprocessing Tool 270

B Component Validation Measurements 273
B.1 Experimental Modal Analysis of the Bedplate 273
B.2 Validation Measurements on the Yaw Gearbox 276
B.3 Validation of the Assembly of Bedplate & Yaw Gearboxes 279

C Detailed Derivation of Error Estimates for Global Eigensolutions 281
C.1 Error Estimates for Global Eigenfrequency 282
C.2 Error Estimates for Global Eigenmode 283

D The Proper Orthogonal Decomposition Method 285

Curriculum Vitae 287

Index 289

Nomenclature

Symbols

General meaning of often used symbols, unless otherwise noted in context:

$\boldsymbol{A}$	Boolean localization matrix	$\boldsymbol{L}$	Boolean assembly matrix
$\boldsymbol{a}$	Adjoint selection vector	$\boldsymbol{M}$	Mass matrix
$\boldsymbol{B}$	Signed Boolean assembly matrix	$\boldsymbol{m}$	Error indicator vector
$\boldsymbol{C}$	Damping matrix	$\boldsymbol{P}$	Matrix with CG iterates
$\boldsymbol{D}$	Collocation assembly matrix	$\boldsymbol{q}$	Generalized DoF vector
$\boldsymbol{d}$	Adjoint solution vector	$\boldsymbol{r}$	Residual vector
$\boldsymbol{e}$	Error vector	$\boldsymbol{R}$	Reduction matrix
$\boldsymbol{f}$	External force vector	$\boldsymbol{S}$	Preconditioning matrix
$\boldsymbol{F}$	Matrix of external force vectors	$\boldsymbol{T}$	Transformation matrix
$\boldsymbol{g}$	Connection force vector	$\boldsymbol{u}$	Displacement DoF vector
$\boldsymbol{G}$	Flexibility matrix	$\boldsymbol{u}_\gamma$	Unique interface displacement DoF
$\boldsymbol{g}_b$	Interface force DoF vector	$\boldsymbol{W}$	Matrix with CG iterates
$J(\star)$	Target functional	$\boldsymbol{Y}$	Receptance matrix
$\boldsymbol{K}$	Stiffness matrix	$\boldsymbol{Z}$	Dynamic stiffness matrix

Greek symbols:

$\boldsymbol{\eta}$	Modal DoF vector	ω	Eigenfrequency, circular frequency
$\boldsymbol{\phi}$	Vibration mode shape	$\boldsymbol{\lambda}$	Lagrange multipliers
$\boldsymbol{\Phi}$	Set of vibration mode shapes	$\boldsymbol{\Omega}$	Diagonal eigenfrequency matrix
$\boldsymbol{\psi}$	Static deformation shape	$\rho(\star)$	Rayleigh quotient
$\boldsymbol{\Psi}$	Set of static deformation shapes		

General meaning of sub- and superscripts, unless otherwise noted in context:

$_a$	Attachment mode	$_k$	Iteration index
$_b$	Boundary	^{c}	Residual mode / rigid body mode
$_{\vert b}$	Trace on the boundary	$^{(0)}$	Belonging to nominal system
$_c$	Constraint mode	$^{(1)}$	Belonging to modified system
$_f$	Free interface mode	$^{+}$	Generalized (pseudo) inverse
$_i$	Internal	$^{(s)}$	Belonging to substructure s
$_j$	Iteration index		

Notation Conventions

A number of typographical conventions are used in this thesis. Scalar variables are indicated by normal lowercase symbols, for instance f, while vectors are denoted by bold lowercase symbols, e.g. $\boldsymbol{f}$. Uppercase bold symbols represent matrices, for instance $\boldsymbol{K}$. Block diagonal matrices are indicated by diag$(\star, \star)$, for instance:

$$\mathrm{diag}(\boldsymbol{X}, \boldsymbol{Y}) = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Y} \end{bmatrix}$$

Stacked column vectors are denoted by col$(\star; \star)$, e.g.:

$$\mathrm{col}(\boldsymbol{x}; \boldsymbol{y}) = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}$$

Furthermore, total time derivatives are indicated by a dot over a variable, e.g.

$$\dot{x} = \frac{dx}{dt} \quad \text{and} \quad \ddot{x} = \frac{d^2x}{dt^2},$$

where x is some time dependent variable and t denotes time.

Abbreviations

The following abbreviations are used throughout this thesis:

AEP	Annual electricity production
CAD	Computer aided design
CAPEX	Captial expenditures
CB	Craig-Bampton
CG	Conjugate gradient
CMS	Component mode synthesis
DCB	Dual Craig-Bampton
DoF	Degrees of freedom
DS	Dynamic substructuring
DWR	Dual-weighted residual
FBS	Frequency based substructuring
FE	Finite element
FRF	Frequency response function
HAWT	Horizontal axis wind turbine
IBS	Impulse based substructuring
IFPKS	Inverse-free Krylov subspace
IRA	Implicit restarted Arnoldi
IRF	Impulse response function
LCOE	Levelized cost of energy
MAC	Modal assurance criterion
MCB	Mixed Craig-Bampton
MTA	Modal truncation augmentation
NREL	National Renewable Energy Lab
OPEX	Operational expenditures
OWT	Offshore wind turbine
POD	Proper orthogonal decomposition
POM	Proper orthogonal mode
POV	Proper orthogonal value
RBM	Rigid body modes
RNA	Rotor-nacelle assembly
SUMAC	Substructure MAC
SVD	Singular value decomposition
SWT	Siemens wind turbine
TRAC	Time response assurance criterion
VAWT	Vertical axis wind turbine
WTG	Wind turbine generator

1

Introduction

When the wind of change blows,
some build walls, others build windmills.

(Chines proverb)

1.1 Wind Energy: For Free and Still Too Expensive?

Wind energy is the kinetic energy contained in the mass of flowing air that occurs due to the temperature differences on the planet, caused by the energy of the sun absorbed by the earth. Wind energy is abundant, inexhaustible and, most importantly, for free. Despite this, its conversion to electrical energy is still too expensive, as will be explained in this section.

For centuries mankind has harnessed the power of the wind. From the first Persian vertical-axis windmills used to grind corn in the ninth century, to the fifteenth century Dutch windmills that helped to reclaim land from the sea: wind energy has long been instrumental to the development of mankind. In fact, its importance to our energy supply has only recently been diminished by the industrial revolution, when fossil fuels became the preferred source of energy. Still, the development of wind power persisted and the end of the nineteenth century marked the transition from traditional windmills to wind turbines. That is, machines that convert wind energy into electrical instead of mechanical energy. These machines formed the basis for modern day wind turbines.

It took until the 1970's for wind energy to regain public attention, when the first oil crisis triggered investigations of renewable energy sources. Especially in the United States this led to the development of the first multi-megawatt wind turbines. However, a steep decline in oil prices in the mid 1980's rendered many of these initiatives uneconomical and hence interest in wind energy eroded. In Denmark however, development of wind turbines had steadily continued throughout the twentieth century, mainly driven by a desire for self-sufficiency during the country's decentralized electrification. This resulted in the so-called *Danish design* that turned out to be the blueprint for modern commercial wind turbines: a horizontal-axis, upwind, three-bladed rotor supported by a tubular tower.

Fueled by concerns over energy security, global warming and depletion of fossil energy sources, the wind industry finally regained momentum in the beginning of this century. Over last decade the installed wind power capacity has been consistently growing by 20 to 30% annually, this exponential growth is shown in figure 1.1. With the market entry of some big multinational companies, such as Siemens, General Electric, Alstom and Mitsubishi Heavy Industries, the wind industry today has become a global 50-billion euro business [76]. At the end of 2011 the global capacity reached 239 Gigawatts, generating around 520 Terawatthours of electrical energy per year [225]. This is sufficient to cover about 3% of the worldwide electricity demand.

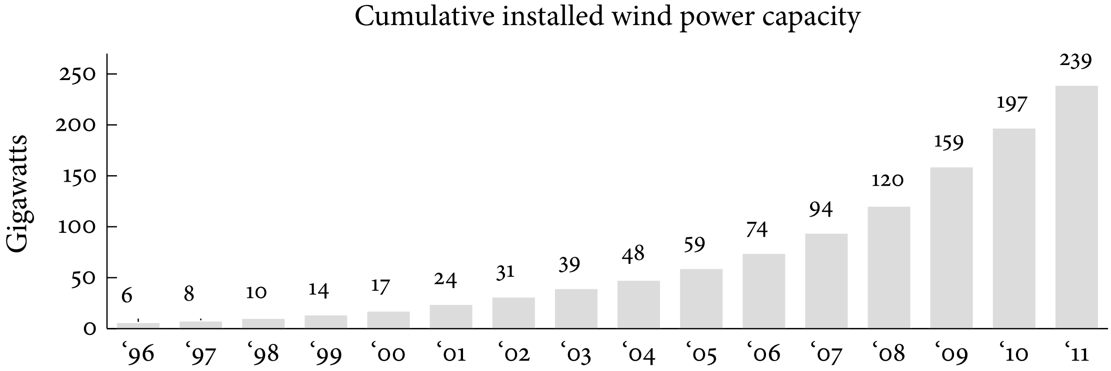


Figure 1.1: Global installed wind power capacity over the last decades, data taken from [76].

Enabling this exponential growth of installed capacity was the incredible pace with which wind turbine technology evolved. In turn, these developments were driven by the desire to bring down the costs of wind generated electricity. Given that the amount of energy extracted from the wind is proportional to the cube of the wind speed and the square of the rotor diameter, and wind speeds generally increase with elevation from the ground, a natural consequence was to construct ever larger and taller wind turbines. Whereas in the mid 1980's the first commercial wind turbines had a rated power output of 50 kilowatts with a rotor diameter of 15 meters, their modern successors come as large as 7.5 megawatts and rotor diameters up to 154 meters. With hub heights (the hub is the center of the rotor) of around 130 meters, this gives a total blade tip height of over 200 meters. For the sake of comparison, the rotor swept area of these machines is close to two hectares, while their rotor diameters are almost twice the wing span of the Airbus A380 jumbo jet (80 meters).

Despite its success over the last few decades, wind energy has not been without controversy. Opponents point at the intermittency and variability of wind generated electricity, also noise and "landscape pollution" are often heard complaints. Furthermore, the natural habitat of some wildlife species tends to be disturbed by the the installation of wind turbines. In an attempt to mitigate these negative effects, developers have turned to a new place to install wind turbines: the sea. Offshore, average wind speeds are higher, acoustic and visual nuisance are hardly relevant and space constraints are absent. Hence, offshore wind energy offers a huge potential to generate "green" electricity on a large scale.

Again Denmark took a pioneering role with the installation of the first offshore wind farm in 1991 at Vindeby. A few years later other countries followed and installed wind turbines in

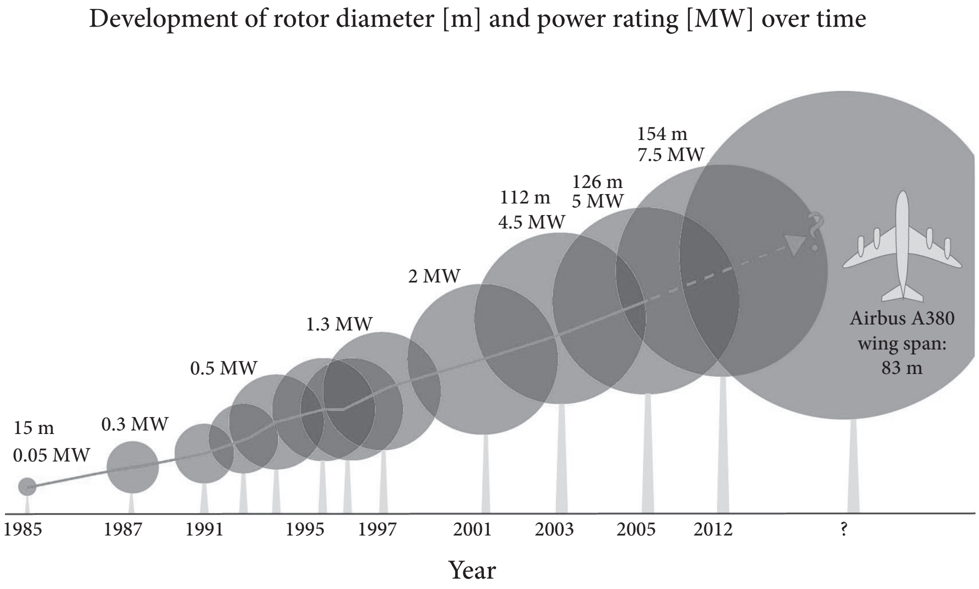


Figure 1.2: Sizes of generations of commercial wind turbines.

their own coastal waters, such as the Netherlands, Sweden, the United Kingdom and Germany. During the last years numerous new offshore wind farms have been installed while many more are currently planned or under construction, as indicated by the numbers in table 1.1.

	2009	2010	2011	Under construction	Planned
Number of turbines installed	201	308	235	-	-
Total capacity installed [MW]	584	883	866	2375	2910
Cumulative capacity [MW]	2063	2946	3813	6188	9098

Table 1.1: European offshore wind industry statistics, source EWEA [61, 62].

The true debate however centers around the economics of wind generated electricity. Although at some favorable (coastal) onshore sites certain wind turbines can compete with conventional energy sources in terms of price per kilowatthour of electricity, general consensus is that wind energy is still too expensive. As a result subsidies are required to make the installation of wind turbines economically viable, which is especially true at offshore locations. This explains the paradoxical title of this section, that was borrowed from Molenaar [137]: even though wind energy itself is for free, harnessing it cost-effectively is still a challenging task.

Nonetheless, wind energy is currently the most promising of all renewable energy sources, both in terms of cost and potential. According to a recent report of the US Energy Information Administration, of all sources of renewable energy, wind is closest to reaching cost-competitiveness with conventional sources under unsubsidized conditions [204]. Furthermore, in 2009 a study by a team of Harvard scientists showed that the global potential for wind energy was estimated at more than forty times the current global electricity consumption and over five times the total energy consumption [128]. Provided that its costs are further reduced, wind energy thus has the potential to play a major role in a sustainable future energy supply.

1.2 Cost Reduction and the Role of Structural Dynamics

In general, the lifetime averaged cost of one kilowatt-hour (kWh) of wind generated electricity, also referred to as the levelized cost of energy (LCOE), is governed by three factors:

- Capital expenditures (CAPEX): the cost of the investment of the wind turbine and all secondary items, such as (offshore) foundation, cabling, etc.
- Operational expenditures (OPEX): the cost of operating the wind turbine during its lifetime, e.g. scheduled maintenance and repair costs.
- Lifecycle energy output: the amount of electricity produced by the wind turbine is the product of its annual electricity production (AEP) and technical lifetime.

Bringing down the cost of wind energy can hence be achieved by reducing the CAPEX and OPEX while increasing the lifecycle energy output. Many ways can be imagined to reach these goals. For instance, an important way of lowering the CAPEX is by reducing the total top mass of the turbine. Thereby not only less material is used, but also more efficient support structures can be designed, transport and installation is simplified, etc. On the OPEX side, cost reductions may be achieved by designing a more reliable turbine that is less prone to failures. Finally, the energy output might be increased by applying a larger rotor diameter and increasing the design lifetime from the common twenty years to for instance twenty-five years.[1]

However simple this may seem, designing a cost-effective multi-megawatt wind turbine is a delicate task. Indeed, a modern turbine combines very slender and flexible structures (blades and tower) with high-tech machinery (drivetrain, yaw system) and electronics in the nacelle, all subjected to intricate and continuous dynamic loading. In addition to the aerodynamic, and for an offshore turbine, hydrodynamic loading, the structure is also excited by each blade passing the tower. This effect is known as tower shadow and occurs with multiples of the rotor rotation frequency P (i.e. for a three-bladed rotor 1P, 3P, 9P, etc.). Furthermore, complex internal excitations are generated for instance in the gearbox and bearings.

Therefore, a crucial aspect in designing a wind turbine is thorough understanding of its structural dynamic behavior. The field of structural dynamics is concerned with the determination of an engineering structure's dynamic behavior, for instance its free vibration frequencies (i.e. eigenfrequencies) and its response to time varying loading. Indeed, a lack of knowledge of the wind turbine's structural dynamics has in the past caused problems, ranging from cracking blades, breaking gearboxes to "singing" towers. To achieve the necessary cost reductions the wind turbine design gets pushed more and more to the limit, such that structural dynamic analysis becomes even more instrumental to the development of wind turbines.

Structural dynamics hence is a fundamental engineering discipline that greatly influences performance (AEP), reliability (OPEX) and cost (CAPEX) of wind turbines and in fact many engineering systems. In order to gain insight in these effects, various analysis tools have been developed over the years. These tools can in general be divided into two classes: simulation versus measurement based methods. Both will be very briefly described next.

---

[1] Of course the CAPEX, OPEX and AEP can be strongly coupled, both in a positive and negative way. For instance using a cheaper, lower quality gearbox reduces CAPEX but increases OPEX, since this gearbox is more likely to need replacement during the turbine's lifetime. Hence, as in the design of any complex machine, the challenge is to find the best compromises.

1.2.1 Simulation: the Finite Element Method

The most notable simulation tool in structural dynamic analysis is the *finite element* (FE) method. The origins of the finite element method date back to the 1940's [94, 35], although its current name first emerged in 1960 [34]. Given its importance today, the basic idea of the FE method is deceivingly simple. It consists in dividing a complex domain into a finite number of smaller domains (elements) which have assumed approximation functions (shape functions). In this way, the continuous partial differential equations describing the physical phenomena, which usually have no closed-form solution, are transformed to a discrete set of ordinary differential equations. By solving these discretized equations, approximate numerical solutions are obtained to the underlying continuous problem.

After the invention of the microprocessor in the 1970's the FE method quickly evolved to become an indispensable tool in the analysis of a large variety of engineering problems. In structural dynamics, the FE method is often employed for the purpose of modal analysis (i.e. extracting the eigenfrequencies and mode shapes of a structure), harmonic analysis, time simulation, etc. Although it was already established some 70 years ago, the research field of finite element methods, in all its facets, is nowadays more active than ever. It is therefore not intended to give an overview here; instead, the reader is referred to some of the standard works for details [19, 95, 73, 44].

Despite tremendous advances in computer power over the last decades, the practical use of high-fidelity FE models is sometimes still hindered by excessive computational cost. This is especially true in the design phase, where different variants must be quickly analyzed and long computation times become prohibitive. Another difficulty is that the FE method only gives reliable results if all relevant physical phenomena are included in the model. Obviously it is far from trivial to guarantee that all physics of the real-world system are included in its simulation model, especially when a design only exists in the virtual world.

1.2.2 Measurements: Experimental Dynamic Analysis

Another way to characterize a system's structural dynamic behavior is by using measured data. This approach can be chosen to circumvent the issues faced when modeling a (newly designed) system or, equally important, to validate the outcomes of a simulation model.

Several types of experimental dynamic analyses can be performed. The most common approach is to measure the dynamic response of a structure to some excitation (e.g. hammer impact, sine sweep, random noise) over a range of frequencies, thereby obtaining *frequency response functions* (FRFs). These can be used directly in order to judge the system's response to certain excitation frequencies, or they can be further processed to obtain the structure's eigenfrequencies and modes. The latter process is called experimental modal analysis. A relatively new field of research is the area of output only dynamic testing, i.e. measuring the structure's dynamic behavior due to ambient and/or operational excitation. The process of determining the eigenmodes in this way is referred to as operational modal analysis. Again, experimental dynamic techniques span a vast and active field of research, and it is not intended to give an overview here. Some classic references can be consulted for details [64, 130].

Like simulation methods, experimental methods for structural dynamic analysis are not without difficulty. The biggest obstacle in performing dynamic testing is the amount of resources needed, both in terms of (expensive) measurement equipment and man hours. Furthermore, dynamic testing is sometimes plagued by a number of practical issues, such as noisy signals, the difficulty of measuring rotational motions, sensitivity to sensor and actuator placement, imposing of the right boundary conditions, and so on.

1.3 State-of-the-Art in Wind Turbine Dynamic Analysis

As explained in above, structural dynamic behavior is an intrinsic aspect in the design and operation of a modern wind turbine. Due to the costs and practical challenges associated with measurements on the huge wind turbine (components), insight in this behavior is in practice mainly gained through simulations. Of course, the simulation results are regularly validated using measured data of (prototypes of) real turbines, but compared to for instance the aerospace industry the amount of dynamic testing is rather limited.

Wind turbine manufacturers, research institutes and universities have therefore developed a specialized class of simulation tools, so-called aero-elastic codes [27, 148]. Based on the finite element method, these specialized programs are developed to analyze the global dynamic behavior of a wind turbine by taking into account all relevant phenomena:

- Aero-elasticity, i.e. coupling of structural deformation and aerodynamic loads
- Rotational effects caused by the spinning of the rotor
- The wind turbine controller's dynamics
- Hydrodynamic loads for offshore turbines
- Soil-structure interaction for offshore turbines

Not only are these codes generally non-linear, also thousands of load cases need to be evaluated for certification purposes. These simulations are needed to quantify the loading on the turbine during its lifetime; to gather statistically relevant data for all of the turbine's different operational regimes (normal operation, extreme wind speeds, shutdown, idling, etc.) many simulations are required. In order to keep the computation times at acceptable levels the models must therefore be as compact as possible. Hence a very limited number of elements is available to describe the wind turbine, resulting in strongly simplified geometries. For instance, many components in the nacelle are represented as equivalent mass or stiffness with only a few degrees of freedom (DoF) and complex frame-like structures are modeled by simple beam elements. Typically, these simulation models have around 300-400 DoF in total. As explained before, these aero-elastic models are validated using measurement data and updated if needed.

As a result of the global dynamic simulations, often called "load calculations" in practice, one obtains both the ultimate and fatigue loads that the turbine is to withstand during its design lifetime. Subsequently, these loads are statically applied to detailed finite element models of the wind turbine components to assess whether their load bearing capacity is sufficient. Such finite element models can easily consist of hundreds of thousands or even millions of DoF.

This sequential approach to structural dynamic analysis of a wind turbine implicitly assumes that components only contribute in a quasi-static way to the global dynamics. As explained next, this might not always be true.

1.4 Towards Integrated Dynamic Analysis of Wind Turbines

It was explained in section 1.2 that in the quest to drive down the costs of wind energy, the wind turbine industry is spending a lot of research and development efforts in optimizing their products. Such optimized turbine designs generally tend to introduce more structural flexibility.

As a result, local component dynamic behavior becomes relevant, especially since interaction with the global dynamics can occur (e.g. resonance of a single component due to some global excitation). This can lead to increased component loading up to a level that results in failure, compromising the overall reliability of the wind turbine. Thorough understanding of the coupled dynamics is therefore essential to enable further cost reduction of wind turbines.

However, due to their relatively few degrees of freedom and geometric simplifications, the existing aero-elastic models are often not capable of predicting these coupled global/local dynamic effects. In this case, the validity of the current sequential approach to structural analysis breaks down and an integrated method should be chosen instead.

Hence it can be concluded that currently in wind turbine engineering a need exists for a more detailed, integrated structural dynamic analysis tool, without losing generality and computational efficiency. The development of such a tool is the topic of this thesis and it is proposed to base such a tool on the paradigm of *dynamic substructuring*. Before formulating the thesis objective, the next section therefore first gives a brief introduction to this methodology.

1.5 Dynamic Substructuring

It was the roman emperor Julius Caesar who in ancient times introduced the principle of "divide and conquer". Ever since, this tactic has been successfully applied in a range of domains: from economics to warfare and from computer science to politics. Probably this inspired scientists and engineers in the previous century to apply the same paradigm to structural dynamic analysis. In those days the available resources, in terms of computing power and measurement hardware, were very limited and global analysis of large and complex structures was infeasible. Therefore they decomposed the structure into several smaller, simpler substructures (or components) of which the dynamic behavior is generally easier to determine. Thereafter they assembled these component dynamic models to obtain the dynamic model of the total system, an approach that is nowadays known as dynamic substructuring (DS). Even today this approach remains highly relevant, as the huge advances in modeling capabilities and measurement hardware are balanced by increasingly complex engineering structures and ever shorter design cycles.

Analyzing a system's structural dynamics in such a componentwise fashion has proven to have some important advantages over global methods where the entire problem is handled at once:

- It allows evaluating the dynamic behavior of structures that are too large or complex to be analyzed as a whole.
- By analyzing the subsystems, local dynamic behavior can be recognized more easily than when the entire system is analyzed.

- When a single component is changed only that component needs to be reanalyzed; the total system can be analyzed at low additional cost.
- It gives the possibility to combine modeled parts and experimentally identified components. This is especially useful for components that are very difficult to model correctly.
- When a structure consists of several identical parts, dynamic substructuring allows these parts to be taken into account very efficiently in the total model.
- It enables sharing and combining substructures from different project groups.
- The level of detail of the component models can be matched with the intended usage of the model.

In short, dynamic substructuring offers a flexible and efficient approach to structural dynamic analysis. Three classes of dynamic substructuring methods consist, which will be treated in more detail in the next chapter.

1.6 Thesis Objectives & Outline

Despite its benefits dynamic substructuring has not yet become a standard tool for the structural dynamic engineer, for various reasons that will be explained in the next chapter. Therefore, the objective of the work presented in this thesis can be formulated as follows:

> *"Develop a practical modeling framework based on the concept of dynamic substructuring that enables detailed, integrated structural dynamic analysis of wind turbines without compromising on computational efficiency."*

In the true spirit of componentwise analysis, this main objective can be broken down into two subproblems. The first can be stated as:

> *"1. Further develop the DS methodology through theoretical extensions that enhance its practical usability."*

The second subproblem can be formulated as:

> *"2. Implement the methodology in the wind turbine engineering practice and demonstrate its potential through realistic case studies."*

Since the majority of wind turbine dynamic analysis is performed through modeling and simulation, the developments in this thesis are focused on numerical substructuring methods. As will be explained in the next chapter, these methods are best suited to bridge the gap between the existing aero-elastic and detailed FE simulation environments, thereby enabling true integrated structural dynamic analysis. Furthermore, the work in this thesis is restricted to *linear* structures. That is, it is assumed that structural deformations remain relatively small such that the models can be described in their undeformed state. In vibration analysis this is generally a valid assumption.

According to the two objectives stated above, the main contents of this thesis is divided into two parts. Part I consists of chapters 2 to 6 and covers all theory, while in part II (chapters

7 to 9) the methods are applied to various wind turbine analysis examples. This division is further motivated by the fact that the theoretical methods in this thesis are of general nature and can be applied to all sorts of engineering problems.[2] Part III presents the conclusions and recommendations following from this work, while part IV comprises the appendices. The structure of parts I and II will be elaborated next.

1.6.1 Part I – Theory

Part I of this thesis is structured according to the typical workflow followed when applying dynamic substructuring analysis in a practical design process. In figure 1.3 this workflow is schematically shown and the chapters associated with each topic are indicated. This process, which is continued until some satisfactory analysis result is obtained, can be described in further detail as follows:

1. Component modeling. The first step in any DS analysis is to decompose the system into components. Each component is thereafter modeled in a suitable way. The actual component modeling is not of primary interest in this work, it is simply assumed that some discrete dynamic model is created using finite elements or some other method.
2. Component validation. To assess the accuracy of the component models they should be validated against component measurements. This topic will be briefly touched upon in appendix B, but is not truly in the scope of this work.
3. Component reduction. In practice component models can have a very large number of DoF. To obtain compact representations of these models, component reduction techniques are applied. Chapter 2 addresses the details of a variety of reduction methods.
4. Assembly. With all components modeled and reduced, they need to be assembled to obtain the dynamic model of the complete system. During assembly, different choices exist as to how to treat the interfaces between components. Chapter 3 elaborates on these assembly techniques. Furthermore, it is sometimes required to decouple models in order to obtain an isolated component model; this is also discussed in chapter 3.
5. Interface reduction. If the assembled model still has too many DoF, it can be further compacted using interface reduction. These methods are outlined in chapter 4.
6. Error estimation. The total model is built from reduced component models which approximate their full counterparts. It is important to have an estimate of the accuracy of the assembled model; chapter 5 therefore presents error estimation methods. Furthermore, refinement strategies are discussed that can be employed in case the accuracy is insufficient. Steps 3-5 have to be redone in that case.
7. Model analysis. Now the assembled model has sufficient accuracy it can be subjected to a dynamic analysis. This could for instance be a modal analysis or time simulation. These analysis methods, as well as techniques to compare the obtained results, are briefly discussed in chapter 4.

---

[2] As a result of this structure, readers mainly interested in the wind turbine application are advised to simply read the summaries at the end of each chapter in Part I, and thereafter proceed to Part II.

8. Assembly validation. To ensure that the assembled model resembles the true structure, validation measurements are required on the total system. In case the model turns out to be inaccurate, the interface models must be updated, since the component models have already been validated in step 2. However, this topic is out of the scope of this thesis.

9. Reduction basis updating. Based on the analysis of the total model, it can be judged whether the design is adequate. In practice the design is rarely "first time right" and often some incremental design change is needed. If so, an updated reduced component model must be created. A method to handle this efficiently is presented in chapter 6.

1.6.2 Part II – Application to Wind Turbine Engineering

In part II the theoretical methods presented in part I are applied to various representative case studies. The purpose of this part is to demonstrate the potential of the developed modeling framework for wind turbine engineering problems. It consists of the following chapters:

- In chapter 7, a detailed yet compact model of a wind turbine yaw system is created and analyzed. Thereby, it makes use of the theory presented in chapters 2 to 4, as indicated by the dotted lines in the flowchart in figure 1.3.
- Chapter 8 treats the iterative design modification of a wind turbine bedframe, making use of the theoretical methods presented in chapter 6.
- In chapter 9 an optimal reduced model of an offshore wind turbine is created and analyzed. Again, the dotted lines in figure 1.3 indicate the related theoretical chapters, namely chapters 2 to 5.

1.6.3 Thesis Contributions

In view of the objectives formulated above, this thesis presents a number of scientific and practical contributions. The proposed developments of the research reported here are:

- Existing component reduction and assembly techniques are summarized and cast in common notation framework, showing their similarities and differences in a transparent way.
- In the field of component model reduction, a generalization of the existing Craig-Bampton and Dual Craig-Bampton methods is presented. This method, dubbed the "Mixed Craig-Bampton method", can be used to overcome the issue of choosing fixed or free interface mode shapes for the component reduction. See section 2.6 and [217].
- To enable assembly of all types of reduced component models, a general framework for substructure assembly is developed. Thereby reduced component models can be created independently in order to increase the modularity of the DS methodology. See chapter 3 and [219]. Furthermore, a simple method is proposed to incorporate additional physics arising from the interface, see section 3.7 and [218].

- Although not in the main scope of this thesis, a substructure *decoupling* method is proposed. Based on a dual assembly formulation, different decoupling methods are derived and classified with the aim to reduce the sensitivity of such methods to measurement errors. See section 3.8 and [212, 213].

- To gain insight in the accuracy of assembled reduced models, without knowledge of the full solution, existing error estimation methods are analyzed and recast into an algebraic CMS framework. Adaptive model reduction schemes are proposed using these error estimates. See chapter 5.

- An updating method is developed to efficiently obtain the reduction bases of component models subjected to design modifications. Thereby the need for (expensive) recomputation of these bases is avoided. See chapter 6 and [214, 215].

- A Matlab toolbox has been developed to facilitate substructuring analysis: the *Dynamic Substructuring Toolbox* (or DS Toolbox). Through a graphical user interface and a dedicated data format, component models can be imported, reduced, assembled and analyzed in a consistent way. The DS Toolbox automates the "bookkeeping" involved in the assembly of numerical models and most of the methods presented in this work are implemented in these tools. See appendix A and [210] for details.

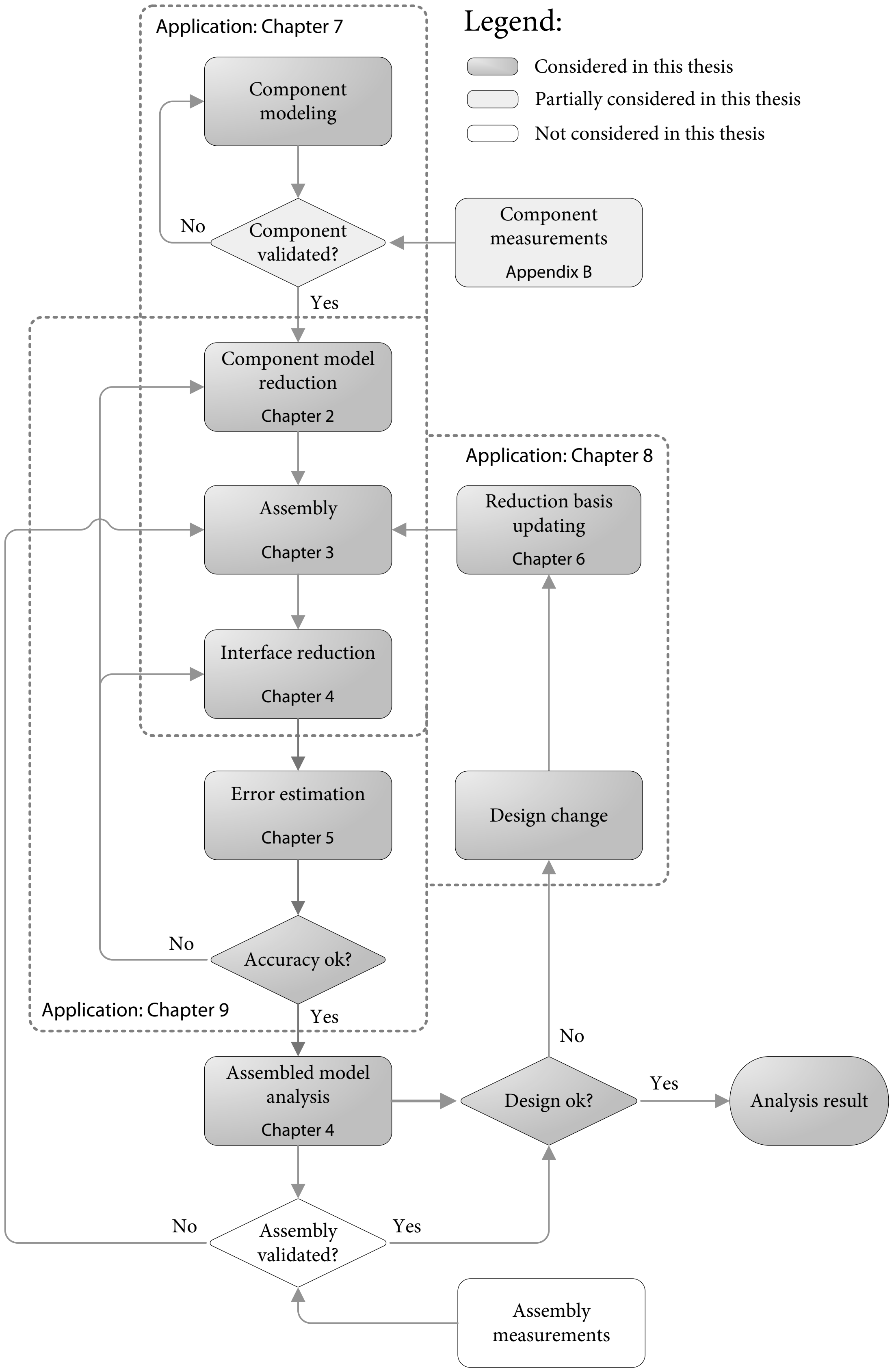
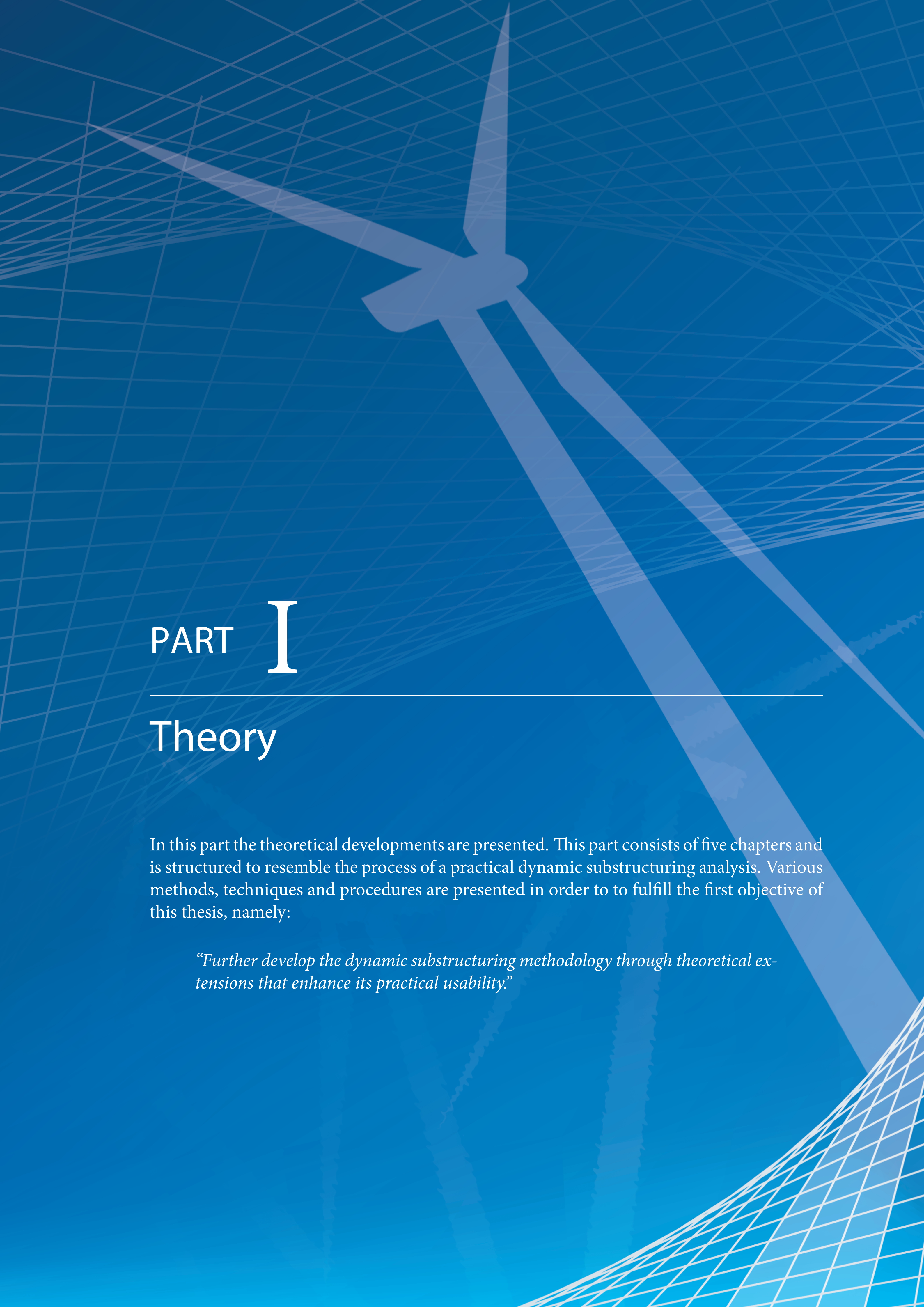


Figure 1.3: Schematic representation of thesis content in a flowchart.



PART I

Theory

In this part the theoretical developments are presented. This part consists of five chapters and is structured to resemble the process of a practical dynamic substructuring analysis. Various methods, techniques and procedures are presented in order to to fulfill the first objective of this thesis, namely:

> *"Further develop the dynamic substructuring methodology through theoretical extensions that enhance its practical usability."*

2

Dynamic Substructuring & Component Model Reduction

Each model is useful for different things, but a 'model' that tried to describe the design, the engineering, the ecology and the economics would be no simpler than reality itself and so would add nothing to our understanding.

(Tim Harford in "The Undercover Economist")

2.1 Introduction to Dynamic Substructuring

In the previous chapter the basic concept of dynamic substructuring was explained, as well as a number of advantages of this componentwise approach. In this thesis the vision is adopted that dynamic substructuring is any technique that tries to find the structural dynamics of a system through assembly of the dynamics of its components. From this perspective dynamic substructuring techniques can be subdivided into three classes of methods. Before diving into the theory, these methods and their historic roots will be treated in a general way in subsections 2.1.1 to 2.1.3. Thereafter subsection 2.1.4 outlines the structure of the remainder of this chapter.

2.1.1 Frequency Based Substructuring

Frequency (or FRF) based substructuring (FBS) is the oldest class of DS methods, with the first contributions in the literature dating back some seven decades [60, 194]. The basic concept of frequency based substructuring is to assemble component FRFs to obtain the FRFs of the total system. Usually these FRFs are obtained from measurements, but they can also be calculated from numerical models. When experimental and numerical models are combined this is sometimes referred to as "hybrid analysis".

After the early contributions cited above, it took until the 1980's for the field of FBS to further develop. Due to ever more accurate and faster measurement equipment, researchers could now quite successfully apply FBS methods for the purpose of structural dynamic modification. The aim of such applications was to alter the dynamic behavior of a base structure by coupling a "modification" structure (usually lumped masses or springs). Although structural modification techniques were at that time not considered as substructuring techniques, the two concepts are in fact identical as observed in a number of publications [42, 10, 50].

One of the first steps towards "true" frequency based coupling techniques was marked by the work of Crowley et al., who proposed a structural modification method called SMURF (Structural Modification Using experimental frequency Response Functions) [46]. However, this method failed to gain popularity in a broad public. A few years later Jetmundsen et al. formulated the classical FBS method [106], which was more efficient and more accurate then the existing FBS methods at that time. The original formulation of the method was generalized in the following years by a number of authors [82, 81, 158].

While the basic theory of FBS was well established in the mid 1990's, successful application of the method was often troubled by all sorts of experimental difficulties. Combined with the realization that some structures (e.g. a car body and its interior) are simply too complex to model numerically, the mid 2000's therefore saw a renewed interest in this field. Since then, a good amount of research has been devoted to developing methods that increase the robustness of FBS analyses. The main goal thereby is to decrease the extreme sensitivity of FBS results to measurement inaccuracies. Some examples include the work in [52, 51, 191, 190].

Finally, a subclass of FBS methods deals with the issue of substructure decoupling, or "inverse substructuring". This problem arises when the FRFs of a component need to be found from the assembled system, which is relevant when the component cannot be measured separately but only when coupled to neighboring substructure(s). Such problems turn out to be even more sensitive to measurement inaccuracies; recently quite some research has therefore been performed to develop methods which alleviate this sensitivity. See for instance [99, 192, 47, 49].

2.1.2 Impulse Based Substructuring

The youngest family of substructuring methods is that of *impulse based substructuring* (IBS) techniques, the first work on this topic was published only two years ago [169]. The concept of IBS is to assemble component *impulse response functions* (IRFs) to obtain the time response of the assembled system. By the use of the convolution product this time response can be arbitrary, but the method is most effective to predict the response to impact or shock-like loading. Indeed, in those cases the high-frequency dynamics of the system are excited, rendering the modal truncation based approach of CMS ineffective. As such, this method is the time domain equivalent of the FBS approach.

Similar to FBS, the IRFs can be obtained either experimentally from (hammer) impact tests or from time integration of numerical component models. Furthermore, since it is a time domain method, IBS also allows straightforward incorporation of non-linear component models [207].

However, like other DS methods there are some difficulties associated with the practical application of IBS. Firstly, it was found that accurate predictions with experimental IRFs is troubled by measurement noise and the fact that a measurement impact is not a perfect Dirac pulse [168]. Secondly, the computational cost of the method escalates for longer simulation times. Therefore, truncation and windowing strategies are currently under development [173].

2.1.3 Component Mode Synthesis

Perhaps the most well known dynamic substructuring methods are those in the class of *component mode synthesis* (CMS), which combines the concepts of componentwise analysis and model reduction techniques. Its basic idea is simple: instead of describing a numerical model at all its nodal DoF, it is described in terms of its most dominant dynamic behavior as expressed in mode shape vectors (or simply "modes"). In this way very compact descriptions of component finite element models are obtained. These models are subsequently assembled to obtain the numerical model of the total system, which can be analyzed at low computational cost to obtain the global dynamic behavior.

The development of these methods was triggered by the initial ideas of Hurty in 1960 [97] which were further worked out in [96]. At the same period a method using a branched vision of the organization of substructures was proposed by Gladwell [75]. Most probably these ideas came forth from the very limited computation power in those days, making the reduction of finite element models a bare necessity to analyze any realistic structure. Rapidly the scientific and engineering communities discovered the benefits of dynamic substructuring and component mode synthesis became an important research topic in the field of structural dynamics. Some major developments followed shortly, resulting in the classic methods by Guyan [85], Craig [37], Rubin [176] and MacNeal [129] in the late 1960's and 1970's. Nowadays these numerical CMS methods are often used in the structural dynamics community; especially the Guyan and Craig-Bampton methods are integrated in many FE software packages. Still, widespread use of these methods in practical design processes can be hindered by high computational cost. Further challenges remain the (estimation of) the accuracy of the reduced models, handling of non-linearities and time-variant (e.g. rotating) systems, and so on.

Around the same time when the numerical CMS methods were developed, already the first attempts were made to perform CMS using measured component models [115]. Such a hybrid approach can be very useful for structures that are difficult to be properly modeled. This is however not a trivial task, since for accurate results so-called static modes are required that are very hard to determine experimentally. Research is therefore still ongoing to improve the results of these hybrid analyses, see e.g. [4, 6, 141].

2.1.4 Chapter Outline

As mentioned in the previous chapter, the focus of this thesis lies on numerical substructuring methods and more specifically methods of the component mode synthesis type. The motivation therefore is twofold. Firstly, performing experimental substructuring on wind turbine structures involves many practical difficulties, rendering such an approach inefficient in a design stage. Secondly, in the wind industry extensive finite element (FE) analysis is already performed for stress analysis purposes. Therefore CMS based dynamic substructuring is chosen to fulfill the objectives formulated in 1.6.

The current chapter focuses on the first step in the CMS based substructuring process, namely reduction of component models, and is related to the other topics of part I of this thesis as illustrated in figure 2.1. This chapter is organized as follows. In the next section, component model reduction methods are formulated in a general way. Thereafter, the most commonly used ingredients used in such methods will be treated: the static modes in section 2.3 and the vibration modes in section 2.4. Section 2.5 then describes the classic component reduction methods that are obtained by combining different ingredients. In section 2.6 a new reduction method is proposed, which is in fact a generalization of two existing methods. Finally, section 2.7 addresses an augmentation method that improves the forced response of reduced models. The chapter is ended by a summary in section 2.8.

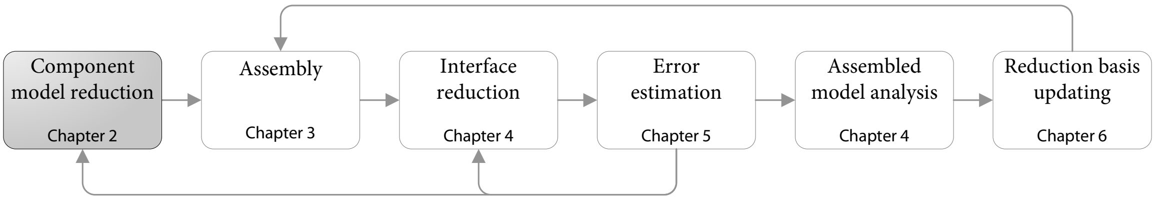


Figure 2.1: Current chapter in relation to other chapters in part I of this thesis.

2.2 Generic Formulation of Component Model Reduction

The creation of a substructured dynamic model starts with the decomposition of the structure into non-overlapping components. For each of these components a finite element (FE) model is then created. In practice, these component FE models are also used for detailed static stress analysis. Since the small geometric details often determine the stress concentrations this requires very fine meshes: models with millions of DoF are no exception. For a few static analysis runs the computational cost associated with these models is still manageable. However, structural dynamic analysis generally requires solving many static-like problems and working with such refined models quickly becomes infeasible. Furthermore, coarser models are usually sufficient since the dynamic behavior is governed by the structure's global mass and stiffness distributions. The engineer thus faces a choice: either use the very fine model or create a coarser mesh. Both options are not very time efficient, as creating a good mesh can be a time consuming process too.

A more elegant approach would therefore be to decrease the number of DoF without modifying the FE mesh. Such methods exist and are known as *model reduction* methods. In structural dynamics these methods consist in replacing the physical DoF by a much smaller set of *generalized* DoF, using the principles of modal superposition and truncation.[1] In other words, instead of describing the structure's dynamic behavior at a very large number of points

---

[1] Model reduction techniques are also employed in many other fields, such as computational mathematics and control engineering. In those fields the reduction methods usually not derive from physical principles but from optimizing certain input/output relations. The main difference is that in those cases the quantity of interest is typically known (e.g. the bandwidth of a certain transfer function) whereas in mechanics/dynamics this is usually not the case (e.g. the location of a maximum stress). For further discussion see e.g. [7, 152, 181].

along the structure, the behavior is expressed in terms of a limited number of deformation shapes ("modes" in general) with associated amplitudes. As such, these reduced models allow to obtain a compact description of the component's structural dynamics while maintaining the level of detail of the original FE model. When the models are reduced for the purpose of subsequent assembly to neighboring substructure models, the process is referred to as component model reduction.

Hence, the starting point for the derivation of component reduction methods is the set of undamped, linear equations of motion of some component *s*:

$$\boldsymbol{M}^{(s)}\ddot{\boldsymbol{q}}^{(s)}(t) + \boldsymbol{K}^{(s)}\boldsymbol{q}^{(s)}(t) = \boldsymbol{f}^{(s)}(t) + \boldsymbol{g}^{(s)}(t) \tag{2.1}$$

Here $\boldsymbol{M}$ is the component's mass matrix, $\boldsymbol{K}$ its stiffness matrix, $\boldsymbol{q}$ the vector of degrees of freedom (DoF), $\boldsymbol{f}$ the external excitation vector and $\boldsymbol{g}$ the excitation felt from neighboring substructures. Equation (2.1) expresses the component's dynamic equilibrium, by stating that the sum of inertia and elastic forces equals the sum of the applied forces and reaction forces from adjacent substructures. In general, the DoF vector can be split into some set of displacement DoF $\boldsymbol{u}_u$ and a set of boundary (or interface) DoF $\boldsymbol{q}_b$:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{u}_u \\ \boldsymbol{q}_b \end{bmatrix} \tag{2.2}$$

Here the explicit time dependence and component identifier *s* are omitted for clarity; in the remainder of this chapter all equations are on a substructure basis, unless otherwise noted. Following this partitioning of the DoF vector, it can be realized that, by definition, the reaction excitation in $\boldsymbol{g}$ is only non-zero at the boundary DoF:

$$\boldsymbol{g} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_b \end{bmatrix} \tag{2.3}$$

As will be further explained in the coming sections, different choices exist for both $\boldsymbol{u}_u$ and $\boldsymbol{q}_b$ depending on the reduction method:

- In a fixed interface method, $\boldsymbol{u}_u$ is the set of internal displacement DoF and $\boldsymbol{q}_b$ are the boundary displacement DoF. In that case $\boldsymbol{g}_b$ contains boundary connection forces.
- In a free interface method, $\boldsymbol{u}_u$ is the complete set of substructure displacement DoF and $\boldsymbol{q}_b$ are the boundary force DoF. In this case $\boldsymbol{g}_b$ contains boundary connection displacements.

Next, a *reduction basis* is computed for the component such that the vector $\boldsymbol{u}_u$ containing n_u physical DoF is replaced by a reduced set of *generalized* DoF. The interface DoF are generally retained for the sake of assembly with neighboring (reduced) component models. A reduction basis usually consists of two types of "ingredients": vibration modes to account for the component's dynamics and static modes to describe the interaction with neighboring substructures. These ingredients can be derived, for the general case, by splitting the response into a static and a dynamic part:

$$\boldsymbol{u}_u = \boldsymbol{u}_{u,\text{stat}} + \boldsymbol{u}_{u,\text{dyn}} \tag{2.4}$$

Firstly, the static response is obtained by setting the accelerations to zero in the equation of motion in eq. (2.1). Using the DoF partitioning of eq. (2.2) one obtains:

$$\begin{bmatrix} \boldsymbol{K}_{uu} & \boldsymbol{K}_{ub} \\ \boldsymbol{K}_{bu} & \boldsymbol{K}_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_u \\ \boldsymbol{q}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_b \end{bmatrix} \tag{2.5}$$

Assuming no external excitation at the DoF in $\boldsymbol{u}_u$ (i.e. $\boldsymbol{f}_u = \boldsymbol{0}$), the static response is found in terms of the boundary DoF:

$$\boldsymbol{u}_{u,\text{stat}} = -\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{ub}\boldsymbol{q}_b = \boldsymbol{\Psi}\boldsymbol{q}_b \tag{2.6}$$

Here $\boldsymbol{\Psi}$ are the static modes, which describe the static deformation of the DoF in $\boldsymbol{u}_u$ as a result of displacement or force (depending on the reduction method) excitation at the boundary DoF. Note that instead of static modes one can also use quasi-static modes, obtained from solving the same equations for a dynamic stiffness matrix obtained by shifting around a central frequency. See for instance [124, 187].

Secondly, the dynamic response is approximated by a set of m vibration modes using modal superposition:

$$\boldsymbol{u}_{u,\text{dyn}} \approx \sum_{j=1}^{m} \boldsymbol{\phi}_j \eta_j = \boldsymbol{\Phi}\boldsymbol{\eta} \tag{2.7}$$

The actual reduction is achieved by taking not all n_u vibration modes into account but by truncating the response. Usually the lowest m modes are used as these capture most of the structure's mass and stiffness; for effective reduction it should hold that $m \ll n_u$. The vibration modes are in general obtained from solving the eigenvalue problem for the DoF in $\boldsymbol{u}$:

$$\left(\boldsymbol{K}_{uu} - \omega_j^2 \boldsymbol{M}_{uu}\right)\boldsymbol{\phi}_j = \boldsymbol{0} \tag{2.8}$$

Hence the approximation for $\boldsymbol{u}_u$ becomes:

$$\boldsymbol{u}_u \approx \boldsymbol{\Psi}\boldsymbol{q}_b + \boldsymbol{\Phi}\boldsymbol{\eta} \tag{2.9}$$

Retaining the interface DoF and writing in matrix/vector notation leads to the following general reduction basis:

$$\boldsymbol{q} \approx \boldsymbol{R}\tilde{\boldsymbol{q}} \quad \rightarrow \quad \begin{bmatrix} \boldsymbol{u}_u \\ \boldsymbol{q}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{\Psi} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{q}_b \end{bmatrix} \tag{2.10}$$

The DoF vector $\tilde{\boldsymbol{q}}$ is often referred to as the set of *generalized* DoF. The above transformation can be inserted in the component's equations of motion, giving:

$$\boldsymbol{M}\boldsymbol{R}\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}\boldsymbol{R}\tilde{\boldsymbol{q}} = \boldsymbol{f} + \boldsymbol{g} + \boldsymbol{r} \tag{2.11}$$

In this equation $\boldsymbol{r}$ is a residual force introduced due to the fact that the approximation given in the reduced basis cannot usually represent the exact solution. The reduced equilibrium equations are then obtained by imposing that the residual force must be zero in the reduction space, namely, $\boldsymbol{R}^T\boldsymbol{r} = \boldsymbol{0}$. In mathematics this is known as Galerkin[2] projection, but in mechanics this often referred to as a Rayleigh-Ritz procedure. After projection by the reduction basis one thus finds the reduced equations of motion of the component as:

$$\tilde{\boldsymbol{M}}\ddot{\tilde{\boldsymbol{q}}} + \tilde{\boldsymbol{K}}\tilde{\boldsymbol{q}} = \tilde{\boldsymbol{f}} + \tilde{\boldsymbol{g}}, \tag{2.12}$$

with:

$$\begin{aligned}
\tilde{\boldsymbol{M}} &= \boldsymbol{R}^T\boldsymbol{M}\boldsymbol{R}\\
\tilde{\boldsymbol{K}} &= \boldsymbol{R}^T\boldsymbol{K}\boldsymbol{R}\\
\tilde{\boldsymbol{f}} &= \boldsymbol{R}^T\boldsymbol{f}\\
\tilde{\boldsymbol{g}} &= \boldsymbol{R}^T\boldsymbol{g}
\end{aligned}$$

Note that due to the reduction of the number of degrees of freedom to deform in, the reduced component model will be stiffer than the full model and consequently has higher eigenfrequencies. This effect is similar to the stiffening caused by the discretization of the underlying continuous equations to a discrete finite element model. After performing some analysis on the reduced model, the solution for the original DoF can be recovered by expansion of the generalized DoF using the reduction matrix in eq. (2.10).

In structural dynamics many component model reduction methods exist; they differ only in the "ingredients" that constitute the reduction basis $\boldsymbol{R}$. The most commonly used ingredients will be treated in the next two sections.

2.3 Reduction Basis Ingredients – Static Modes

As explained above, static modes are an essential part of a component reduction basis. Not only do they make sure that response of the reduced model is statically correct, but more importantly they allow the compatibility condition to be satisfied after assembly (see chapter 3). In the next three subsections three types of commonly used static modes are discussed.

2.3.1 Constraint Modes

Static *constraint modes* are used in fixed interface reduction methods (see section 2.5) and can be derived by partitioning a component's displacement DoF into internal $_i$ and boundary $_b$ DoF. The general DoF vector of eq. (2.2) in this case becomes:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} \tag{2.13}$$

Using the same procedure as in the previous section, the static constraint modes can be computed by neglecting the inertia forces and condensing the remaining static on the boundary DoF $\boldsymbol{u}_b$. This gives:

$$\boldsymbol{u}_{i,\text{stat}} = -\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}\boldsymbol{u}_b = \boldsymbol{\Psi}_c\boldsymbol{u}_b \tag{2.14}$$

---

[2] If a different basis is chosen for the projection than for the reduction this is known as a Petrov-Galerkin method.

The term $-\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}$ is referred to as the static condensation matrix, whose columns contain the static constraint modes. Using these modes, the original set of degrees of freedom $\boldsymbol{u}$ are reduced to the set of boundary DoF $\boldsymbol{u}_b$, as:

$$\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Psi}_c \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{u}_b \tag{2.15}$$

Physically, these modes represent the static deformation shape due to a unit displacement applied to one of the boundary DoF, while the remaining boundary DoF are constrained and no forces are applied at the internal DoF. The constraint modes thus contain the substructure's static response to applied interface displacements. An example of a static constraint mode is illustrated in figure 2.2.

Finally, as was noted in the previous section for the general case, the constraint modes can also be computed at a central frequency ω_s, giving rise to a set of quasi-static constraint modes [124]. These are computed from the dynamic stiffness matrix, i.e.:

$$-\omega_s^2\boldsymbol{M}\boldsymbol{u} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{Z}(\omega_s)\boldsymbol{u} = \boldsymbol{f}(\omega_s) \tag{2.16}$$

This leads to the following shifted constraint modes:

$$\hat{\boldsymbol{u}}_{i,\text{stat}} = -\boldsymbol{Z}_{ii}^{-1}\boldsymbol{Z}_{ib}\boldsymbol{u}_b = \hat{\boldsymbol{\Psi}}_c\boldsymbol{u}_b \tag{2.17}$$

In this case the internal inertia forces are no longer neglected but taken into account at the frequency ω_s.

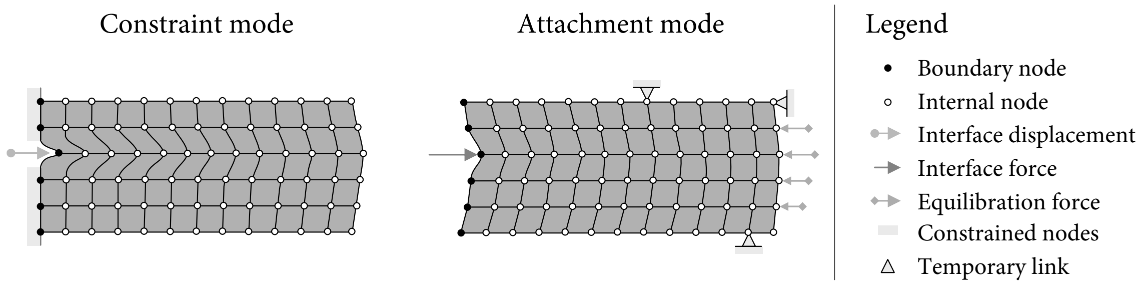


Figure 2.2: Illustration of the types of static modes.

2.3.2 Attachment Modes

A different type of static mode is the so-called *attachment mode*. These modes find their origin in experimental CMS since, in contrast to the static constraint modes, they can be obtained relatively easily from experiments (see e.g. [132, 59]). In numerical CMS however, these modes are barely used due to the more advantageous properties of the residual attachment modes introduced in the next subsection. Still, it is explained in this section how these modes should be calculated, as they are a prerequisite for the computation of the residual attachment modes.

The (residual) attachment modes are normally used in free interface reduction methods (see section 2.5) in which the displacement DoF are not partitioned, but the complete displacement field is approximated. In order to allow assembly to neighboring components, interface force DoF are included (see chapter 3). This gives the following DoF vector:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{g}_b \end{bmatrix}, \tag{2.18}$$

with $\boldsymbol{g}_b$ the interface connection forces from neighboring components. The static problem now writes:

$$\begin{bmatrix} \boldsymbol{K} & -\boldsymbol{A}^T \\ -\boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} \tag{2.19}$$

Here $\boldsymbol{A}$ is a local Boolean matrix selecting the interface DoF of the substructure. This Boolean matrix has size $n_b \times n_u$, with n_b the number of boundary DoF for the component. Note that the second equation seems redundant, but is added for symmetry and will be used to enforce compatibility during assembly. The latter is explained in the next chapter 3. Again assuming no external excitation ($\boldsymbol{f} = \boldsymbol{0}$) the static response is simply found as:

$$\boldsymbol{u}_{\text{stat}} = \boldsymbol{K}^{+}\boldsymbol{A}^T\boldsymbol{g}_b = \boldsymbol{\Psi}_a\boldsymbol{g}_b \tag{2.20}$$

Here $\boldsymbol{\Psi}_a$ are the attachment modes while $\boldsymbol{K}^{+}$ is the generalized (or pseudo) inverse of the stiffness matrix, which is in fact a flexibility matrix that will be subsequently denoted by $\boldsymbol{G}$. Hence, the attachment modes physically correspond to the static flexible response to unit interface forces applied at the interface DoF. If the substructure is constrained such that no rigid body modes exist then $\boldsymbol{G} = \boldsymbol{K}^{+} = \boldsymbol{K}^{-1}$ and computation of the attachment modes is very straightforward. However, in case the structure is free floating their computation is considerably more complex and a number of additional steps must be taken, namely:

1. Obtain the rigid body modes $\boldsymbol{\Phi}_r$, this is treated in detail in section 2.4.2.

2. Compute a generalized inverse of $\boldsymbol{K}$. This generalized inverse is not unique. One way to obtain a generalized inverse is to constrain the system at some DoF $\boldsymbol{u}_c$, referred to as isostatic constraints or temporary links, while leaving the remaining $\boldsymbol{u}_u$ free:

   $$\begin{bmatrix} \boldsymbol{K}_{cc} & \boldsymbol{K}_{cu} \\ \boldsymbol{K}_{uc} & \boldsymbol{K}_{uu} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_c \\ \boldsymbol{A}_u^T \end{bmatrix} \tag{2.21}$$

   In [206] an algorithm is outlined for finding the optimal isostatic constraints, in the sense that they result in minimal constraint forces $\boldsymbol{f}_c$. This is important in order to get the best possible estimate of the component's static deformation due to unit interface forces. With the isostatic constraints applied the remaining stiffness matrix is just rendered positive definite, allowing to compute a generalized inverse:

   $$\boldsymbol{K}^{+} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{uu}^{-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_{uu} \end{bmatrix} = \boldsymbol{G}_c \tag{2.22}$$

   The matrix $\boldsymbol{G}_c$ is called the constrained flexibility matrix.

3. Determine self-equilibrated "force" vectors in $\boldsymbol{A}_{eq}^T$. In the presence of rigid body modes there is no solution to eq. (2.20), unless the applied forces are self-equilibrating. If this would not be the case, the structure would undergo infinite accelerations, since

the mass is not taken into account, and undetermined deformations. To obtain self-equilibrated interface forces one can orthogonalize them with respect to the component's rigid body modes $\boldsymbol{\Phi}_r$. This is achieved through projection as follows:

$$\boldsymbol{A}_{eq}^T = \left(\boldsymbol{I} - \boldsymbol{M}\boldsymbol{\Phi}_r\boldsymbol{\Phi}_r^T\right)\boldsymbol{A}^T = \boldsymbol{P}\boldsymbol{A}^T, \tag{2.23}$$

where mass normalized rigid body modes are assumed. The projection matrix $\boldsymbol{P}$ is termed the inertia-relief projection matrix; for a detailed derivation see [36]. Now the attachment modes are found as:

$$\tilde{\boldsymbol{\Psi}}_a = \boldsymbol{G}_c\boldsymbol{A}_{eq}^T \tag{2.24}$$

An illustration of the attachment modes and a self-equilibrated interface force is given in figure 2.2.

4. Finally, the attachment modes $\tilde{\boldsymbol{\Psi}}_a$ need to be mass-orthogonalized with respect to the rigid body modes. This is needed to ensure that the attachment modes do not contain any rigid body contribution themselves and only describe the static deformation of the structure. In order to achieve this, they are pre-multiplied by the projection matrix $\boldsymbol{P}$:

$$\boldsymbol{\Psi}_a = \boldsymbol{P}^T\tilde{\boldsymbol{\Psi}}_a = \boldsymbol{P}^T\boldsymbol{G}_c\boldsymbol{P}\boldsymbol{A}^T = \boldsymbol{G}_f\boldsymbol{A}^T \tag{2.25}$$

Here the matrix $\boldsymbol{G}_f$ is referred to as the elastic flexibility matrix, which can be shown to have the following properties:

$$\begin{aligned} &\boldsymbol{G}_f^T = \boldsymbol{G}_f \\ &\boldsymbol{G}_f^T\boldsymbol{K}\boldsymbol{G}_f = \boldsymbol{G}_f \\ &\boldsymbol{\Phi}_r^T\boldsymbol{M}\boldsymbol{G}_f = \boldsymbol{0} \end{aligned} \tag{2.26}$$

Hence, the attachment modes are in fact the columns of the elastic flexibility matrix associated to the boundary DoF.

Note that the attachment modes can also be computed at certain frequency of interest ω_s to obtain the quasi-static attachment modes, as proposed in [188]. Similar to the constraint modes, these are computed from the dynamic stiffness matrix, as:

$$\hat{\boldsymbol{u}}_{\text{stat}} = \left(\boldsymbol{K} - \omega_s^2\boldsymbol{M}\right)^{-1}\boldsymbol{A}^T\boldsymbol{g}_b = \boldsymbol{Z}^{-1}\boldsymbol{A}^T\boldsymbol{g}_b = \hat{\boldsymbol{\Psi}}_a\boldsymbol{g}_b \tag{2.27}$$

The so obtained modes represent the structure's deformation resulting from harmonic forces applied at the boundary DoF with frequency ω_s. In this case the presence of rigid body modes does not require any special computational procedure, since the dynamic stiffness matrix is positive definite for any $\omega_s > 0$.

2.3.3 Residual Attachment Modes

The attachment modes derived in the previous section were historically derived for use in experimental CMS analysis. However, when used in a numerical setting in combination with free interface vibration modes $\boldsymbol{\Phi}_f$ (see section 2.4.1), the attachment modes can be further processed to obtain the *residual attachment modes* which have better computational properties. This can be understood from the spectral expansion of the flexibility matrix is [74]:

$$\boldsymbol{G}_f = \sum_{j=n_r+1}^{n} \frac{\boldsymbol{\phi}_j \boldsymbol{\phi}_j^T}{\omega_j^2} \tag{2.28}$$

Here n_r is the number of rigid body modes. This spectral expansion shows that the flexibility associated to the vibration modes is implicitly accounted for twice. In order to simplify the expressions of the reduced system and, more importantly, obtain an $\boldsymbol{M}$- and $\boldsymbol{K}$-orthogonal basis, one could therefore subtract the flexibility that is already accounted for by the m free vibration modes in the reduction basis. As a result, the residual flexibility matrix is obtained:

$$\boldsymbol{G}_r = \boldsymbol{G}_f - \sum_{j=n_r+1}^{m} \frac{\boldsymbol{\phi}_j \boldsymbol{\phi}_j^T}{\omega_j^2} \tag{2.29}$$

This matrix contains the flexibility contained in the $d = n - m$ discarded modes. Similar to the attachment modes, the *residual attachment modes* are now simply found by picking the columns associated to the boundary DoF. This gives:

$$\boldsymbol{\Psi}_r = \boldsymbol{G}_r \boldsymbol{A}^T \tag{2.30}$$

Due to their construction the residual attachment modes are not only mass and stiffness orthogonal with respect to the rigid body modes, like the “normal” attachment modes, but also with respect to the retained free interface vibration modes.

2.4 Reduction Basis Ingredients – Vibration Modes

As was explained in section 2.2, the vibration modes are included in the reduction basis to account for the dynamic behavior of the component. By including only a limited number of these modes (i.e. applying modal truncation), the component model is reduced. Similar to the static modes, different types of vibration modes exist. Next the three most commonly used types are addressed.

2.4.1 Free Interface Modes

The *free interface modes* are the most natural type of vibration mode. They are the deformation shapes of the component when it is allowed to vibrate freely. These modes can be computed from the component’s equation of motion by including all displacement DoF and assuming no external excitation:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{0} \tag{2.31}$$

A solution to this equation can be sought by separating the response of the internal DoF in a space-dependent and time-dependent part, by assuming:

$$\boldsymbol{u} = \boldsymbol{x} e^{i\omega t} \tag{2.32}$$

Substitution in equation (2.31) then gives:

$$(\boldsymbol{K} - \omega^2 \boldsymbol{M})\, \boldsymbol{x} e^{i\omega t} = \boldsymbol{0} \tag{2.33}$$

Hence, in order to obtain a non-trivial solution the following eigenvalue problem needs to be solved:

$$\left(\boldsymbol{K} - \omega_{f,j}^2 \boldsymbol{M}\right) \boldsymbol{\phi}_{f,j} = \boldsymbol{0} \tag{2.34}$$

Here $\omega_{f,j}^2$ is the j$^{\text{th}}$ free interface eigenfrequency and $\boldsymbol{\phi}_{f,j}$ its associated eigenmode. A set of these free interface vibration modes is denoted by $\boldsymbol{\Phi}_f$; an illustration of a free interface vibration mode is given in figure 2.3. Note that in practice the eigenvalue problem in eq. (2.34) can be very large and is hence computationally expensive to solve; algorithms for eigenvalue problems are treated in section 6.3.1. In this work it is assumed that the free interface vibration modes are mass normalized (i.e. unity modal mass).

Since it is not required to apply any boundary conditions, free interface vibration modes can also be obtained quite easily from measurements. These can either be used directly in an experimental CMS analysis , or could be used to validated the modes obtained from a numerical model. A method that combines these approaches is the *System Equivalent Reduction Expansion Process* (SEREP) [144], in which the measured mode shapes are expanded to the full space of the numerical model through a least squares fit.

2.4.2 Rigid Body Modes

A special type of vibration modes are *rigid body modes*. In these modes the substructure shows displacements without deformations, that is, it is undergoing rigid body motion. Rigid body modes (RBM) exist if the substructure is (partially) unconstrained. In a substructuring analysis this is often the case: even when the total structure is constrained this is probably not the case for many of its components. In other words, the components' rigid body modes only vanish after assembly.

Given the definition of rigid body modes as displacement shapes that do not cause any deformation, their basic equation reads:

$$\boldsymbol{K}\boldsymbol{\Phi}_r = \boldsymbol{0} \tag{2.35}$$

In addition to the "global" rigid body modes, six for an unconstrained three dimensional model, a component could also contain mechanisms (e.g. hinges) which allow for parts of the substructure to displace without introducing any deformations. Such displacements are also rigid body modes and satisfy eq. (2.35).

The actual computation of the rigid body modes can be achieved in a number of ways. They can be obtained by solving eq. (2.35) (i.e. computing the nullspace of the stiffness matrix) or equivalently as the zero frequency modes obtained from the eigenvalue analysis performed to obtain the free interface mode shapes (i.e. from (2.34)). Both approaches are however computationally expensive and, provided the component is free of mechanisms, a much cheaper alternative exists by obtaining the rigid body modes geometrically. This will be explained next.

In general, the rigid body modes can be represented on a per node basis as:

$$\boldsymbol{\Phi}_r = \left[\boldsymbol{\Phi}_{r|1}^T \; \dots \; \boldsymbol{\Phi}_{r|j}^T \; \dots \; \boldsymbol{\Phi}_{r|n_n}^T\right]^T \tag{2.36}$$

Here $\boldsymbol{\Phi}_{r|j}$ is the trace of the rigid body modes on the DoF of node j, with $j = 1 \dots n_n$ and n_n the number of nodes in the component model. For a finite element mesh consisting of elements with six DoF per node, $\boldsymbol{\Phi}_{r|j}$ is obtained from the geometry of the structural model as follows:

$$\boldsymbol{\Phi}_{r|j}^{\text{geo}} = \left[\boldsymbol{\Phi}_{r(\boldsymbol{x},j)}^{\text{geo}} \quad \boldsymbol{\Phi}_{r(\boldsymbol{\theta},j)}^{\text{geo}}\right] = \begin{bmatrix} \boldsymbol{I}_{33} & \boldsymbol{R}_{\boldsymbol{\theta\theta}(j)} \\ \boldsymbol{0}_{33} & \boldsymbol{I}_{33} \end{bmatrix} \tag{2.37}$$

Here the subscript $_{\boldsymbol{x}}$ is used to indicate the translational rigid body modes and the subscript $_{\boldsymbol{\theta}}$ for the rotational ones, while the superscript $^{\text{geo}}$ denotes the fact that the rotational rigid body modes are obtained geometrically. Hence, it is assumed that the nodal DoF are ordered according to translational DoF and rotational DoF:

$$\boldsymbol{u}_{(j)} = \left[\boldsymbol{u}_{(\boldsymbol{x},j)}^T \quad \boldsymbol{u}_{(\boldsymbol{\theta},j)}^T\right]^T = \left[x_{(j)} \quad y_{(j)} \quad z_{(j)} \quad \theta_x(j) \quad \theta_y(j) \quad \theta_z(j)\right]^T \tag{2.38}$$

Using the FE model geometry the term $\boldsymbol{R}_{\boldsymbol{\theta\theta}(j)}$, describing the displacements due to rotational rigid body motions, can be computed as:

$$\boldsymbol{R}_{\boldsymbol{\theta\theta}(j)} = \left[\boldsymbol{e}_1 \times \boldsymbol{d}_j \quad \boldsymbol{e}_2 \times \boldsymbol{d}_j \quad \boldsymbol{e}_3 \times \boldsymbol{d}_j\right] = \begin{bmatrix} 0 & d_{z(j)} & -d_{y(j)} \\ -d_{z(j)} & 0 & d_{x(j)} \\ d_{y(j)} & -d_{x(j)} & 0 \end{bmatrix} \tag{2.39}$$

In this expression the vector $\boldsymbol{d}_j$ is the position vector from some reference node in the FE model, which can be chosen arbitrarily, to the current node j. For a finite element model built from elements with only translational DoF (e.g. using tetrahedron elements), the bottom three lines in eq. (4.3) can be discarded and $\boldsymbol{\Phi}_{r(j)}^{\text{geo}}$ becomes a matrix of size 3 × 6 instead of 6 × 6.

Note that by construction the above obtained translational rigid body modes are mass orthogonal, i.e.:

$$\boldsymbol{\Phi}_{r(\boldsymbol{x})}^{\text{geo}^T} \boldsymbol{M} \boldsymbol{\Phi}_{r(\boldsymbol{x})}^{\text{geo}} = \text{diag}\left(\mu_x, \mu_y, \mu_z\right) \tag{2.40}$$

Here $\mu_\star$ are the modal masses of the translational rigid body modes, which in fact correspond to the global masses of the structure and hence serve as a simple check of the model.

However, the above orthogonality does not hold for the rotational rigid body modes. Since the reference node is chosen randomly, it is likely not at the center of mass and hence the rotational displacement vectors $\boldsymbol{\Phi}_{r(\boldsymbol{\theta})}^{\text{geo}}$ are a combination of a translational and rotational rigid body mode. In order to orthogonalize the rigid body modes, the translational modes have to be projected out of the space spanned by the rotational rigid body vectors.[3] This projection is similar to the equilibration process in section 2.3.2 and is achieved by:

$$\tilde{\boldsymbol{\Phi}}_{r(\boldsymbol{\theta})}^{\text{geo}} = \left(\boldsymbol{I} - \boldsymbol{\Phi}_{r(\boldsymbol{x})}^{\text{geo}} \left(\boldsymbol{\Phi}_{r(\boldsymbol{x})}^{\text{geo}^T} \boldsymbol{M} \boldsymbol{\Phi}_{r(\boldsymbol{x})}^{\text{geo}}\right)^{-1} \boldsymbol{\Phi}_{r(\boldsymbol{x})}^{\text{geo}^T} \boldsymbol{M}\right) \boldsymbol{\Phi}_{r(\boldsymbol{\theta})}^{\text{geo}} = \boldsymbol{P}_{r(\boldsymbol{x})} \boldsymbol{\Phi}_{r(\boldsymbol{\theta})}^{\text{geo}} \tag{2.41}$$

---

[3] Alternatively, one could choose to solve the eigenproblem on the subspace of the rigid body modes in order to orthogonalize the rigid body modes with respect to the mass matrix.

After this projection the rigid body modes are mass and (by definition) stiffness orthogonal. Finally, these modes need to be mass normalized to obtain the orthonormalized rigid body modes for use in component reduction bases:

$$\boldsymbol{\Phi}_r = \begin{bmatrix} \boldsymbol{\Phi}^{\text{geo}}_{r(\boldsymbol{x})} & \tilde{\boldsymbol{\Phi}}^{\text{geo}}_{r(\boldsymbol{\theta})} \end{bmatrix} \boldsymbol{\mu}^{-1} \quad \text{with} \quad \boldsymbol{\mu} = \text{diag}\left(\mu_x, \mu_y, \mu_z, \mu_{\theta_x}, \mu_{\theta_y}, \mu_{\theta_z}\right) \tag{2.42}$$

Since in this approach the RBM are identified with respect to a chosen reference node, rigid motions due to mechanisms cannot be detected. One therefore has to ensure the structure has no unconstrained mechanisms, otherwise the set of rigid body modes obtained from this method will not be complete. An illustration of a rigid body mode is given in figure 2.3.

2.4.3 Fixed Interface Modes

The *fixed interface modes* are the vibration shapes of the component when it is constrained at the interface DoF. Similar to the constraint modes, the first step in their computation is the partitioning of the displacement DoF vector into internal and boundary DoF. This gives the following partitioned equations of motion:

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{ib} \\ \boldsymbol{M}_{bi} & \boldsymbol{M}_{bb} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_i \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib} \\ \boldsymbol{K}_{bi} & \boldsymbol{K}_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_b \end{bmatrix} \tag{2.43}$$

Next, the boundary DoF are constrained and it is assumed there is no external excitation on the internal DoF (i.e. $\boldsymbol{f}_i = \boldsymbol{0}$). The equation of motion is thus reduced to only the internal DoF:

$$\boldsymbol{M}_{ii}\ddot{\boldsymbol{u}}_i + \boldsymbol{K}_{ii}\boldsymbol{u}_i = \boldsymbol{0} \tag{2.44}$$

Analogous to the free vibration modes a solution can be assumed that separates the time and space dependence of $\boldsymbol{u}_i$, giving rise to internal the eigenvalue problem:

$$\left(\boldsymbol{K}_{ii} - \omega^2_{i,j}\boldsymbol{M}_{ii}\right)\boldsymbol{\phi}_{i,j} = \boldsymbol{0} \tag{2.45}$$

Here $\omega^2_{i,j}$ is the j$^{\text{th}}$ fixed interface eigenfrequency and $\boldsymbol{\phi}_{i,j}$ its associated mode shape; a set of fixed interface vibration modes is represented by $\boldsymbol{\Phi}_i$. When a free floating component model has six or more interface DoF, which is generally the case, the fixed interface modes do not contain any rigid body displacements. Figure 2.3 gives an illustration of a fixed interface vibration mode. Note that it is assumed that the fixed interface vibration modes are mass normalized (i.e. unity modal mass).

Note that the fixed interface vibration modes, like the static constraint modes, are hard to obtain experimentally. This is due to the fact that the interface DoF need to be constrained, a condition that is very difficult to realize in practice.

2.5 Classic Component Reduction Methods

In the previous two sections the most common types of modes used in component reduction methods have been outlined. This section therefore addresses the actual reduction methods that are obtained by mixing the ingredients provided in the previous sections.

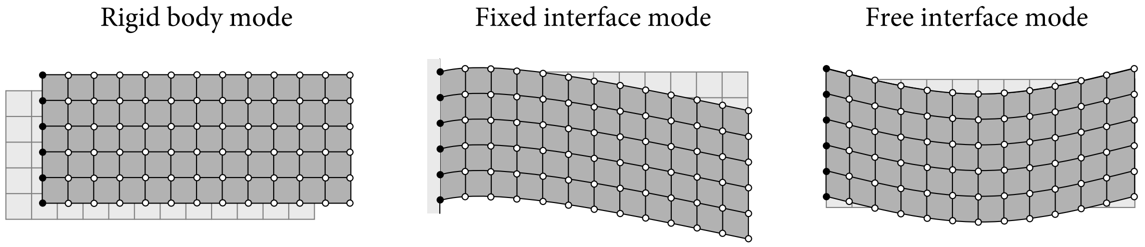


Figure 2.3: Illustration of the types of vibration modes.

2.5.1 Guyan Method

The oldest of the component reduction methods discussed in this section is the method proposed by Guyan. In his classic half page long paper [85], Guyan proposed in 1965 an idea that is still actively used in structural dynamics today. Note that the method is sometimes also referred to as the Guyan-Irons method, after Irons who proposed the same idea some five years later [101], or as static condensation (see section 2.3).

Starting from the component equations of motion in eq. (2.1), the set of displacement DoF are partitioned into boundary DoF $\boldsymbol{u}_b$ and internal DoF $\boldsymbol{u}_i$. The Guyan method then condenses the internal DoF to the boundary DoF using the static constraint modes derived in section 2.3.1:

$$\boldsymbol{u}_i = \boldsymbol{\Psi}_c \boldsymbol{u}_b, \tag{2.46}$$

while the boundary DoF are retained. This gives the Guyan reduction basis:

$$\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Psi}_c \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{u}_b = \boldsymbol{R}_G \boldsymbol{q}_G \tag{2.47}$$

Hence, the reduction basis consists of only the static constraint modes and the generalized DoF are simply the boundary DoF. Applying this reduction basis to the equations of motion as in eq. (2.12) results in the reduced system of equations:

$$\tilde{\boldsymbol{M}} \ddot{\boldsymbol{u}}_b + \tilde{\boldsymbol{K}} \boldsymbol{u}_b = \tilde{\boldsymbol{f}} + \boldsymbol{g}_b \tag{2.48}$$

Where the reduced matrices are found as:

$$\begin{aligned}
\tilde{\boldsymbol{K}} &= \boldsymbol{K}_{bb} - \boldsymbol{K}_{bi} \boldsymbol{K}_{ii}^{-1} \boldsymbol{K}_{ib} \\
\tilde{\boldsymbol{M}} &= \boldsymbol{M}_{bb} - \boldsymbol{M}_{bi} \boldsymbol{K}_{ii}^{-1} \boldsymbol{K}_{ib} - \boldsymbol{K}_{bi} \boldsymbol{K}_{ii}^{-1} \boldsymbol{M}_{ib} + \boldsymbol{K}_{bi} \boldsymbol{K}_{ii}^{-1} \boldsymbol{M}_{ii} \boldsymbol{K}_{ii}^{-1} \boldsymbol{K}_{ib} \\
\tilde{\boldsymbol{f}} &= \boldsymbol{f}_b - \boldsymbol{K}_{bi} \boldsymbol{K}_{ii}^{-1} \boldsymbol{f}_i
\end{aligned} \tag{2.49}$$

In the derivation of the constraint modes it is assumed that there is no excitation at the internal DoF and the internal inertia forces can be neglected (see section 2.3.1). Whereas the former assumption is only rarely a limitation for the usability of the Guyan method (methods exist to overcome this, see section 2.7), the latter is all the more restrictive. Due to the fact that the internal inertia forces are statically condensed to the interface they only contribute in a quasi-static manner, meaning that the dynamic behavior is strongly approximated. The accuracy of this approximation is only reasonable as long as the highest eigenfrequency one wants to compute for the entire structure is much lower than the lowest eigenfrequency of the substructure clamped at its interface.

Note that the Guyan method can also be applied around a central frequency by using the quasi-static constraint modes for reduction (see section 2.3.1). In this accurate results are found around this shifting frequency.

Since the interface DoF of the substructure model are retained, assembly to other (reduced) models is very straightforward (see chapter 3). Due to this property, such reduced models are referred to as *superelements*.

Finally it is noted that in case the substructure has rigid body modes, these are retained in the reduced model. This can be easily shown as follows. Recall the definition of the rigid body modes from equation (2.35), which can be partitioned as:

$$\begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib} \\ \boldsymbol{K}_{bi} & \boldsymbol{K}_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_{r|i} \\ \boldsymbol{\Phi}_{r|b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{2.50}$$

Since only the boundary DoF of the component are retained after reduction, a rigid body displacement imposed on these DoF should not generate any elastic forces. So the following condition should hold:

$$\tilde{\boldsymbol{K}} \boldsymbol{\Phi}_{r|b} = \left(\boldsymbol{K}_{bb} - \boldsymbol{K}_{bi} \boldsymbol{K}_{ii}^{-1} \boldsymbol{K}_{ib} \right) \boldsymbol{\Phi}_{r|b} = \boldsymbol{0} \tag{2.51}$$

From the second equation in the definition of the rigid body modes in (2.50) it follows that:

$$\boldsymbol{K}_{ib} \boldsymbol{\Phi}_{r|b} = -\boldsymbol{K}_{ii} \boldsymbol{\Phi}_{r|i} \tag{2.52}$$

Such that after substitution one finds:

$$\tilde{\boldsymbol{K}} \boldsymbol{\Phi}_{r|b} = \boldsymbol{K}_{bb} \boldsymbol{\Phi}_{r|b} + \boldsymbol{K}_{bi} \boldsymbol{K}_{ii}^{-1} \boldsymbol{K}_{ii} \boldsymbol{\Phi}_{r|i} = \boldsymbol{0} \tag{2.53}$$

Which is indeed in accordance with the first equation in (2.50). Hence, even though the rigid body modes are not explicitly contained in the static constraint modes, these modes are consistent with the RBM such that the reduced model can still predict rigid motions. This is an important notion when the static modes are calculated with only a limited accuracy, as will be discussed in section 6.2.

2.5.2 Craig-Bampton Method

To overcome the main drawback of the Guyan method, the classic Craig-Bampton method proposes an expansion of this method by including information on the internal dynamics in the reduced component model [37]. This is achieved by expanding the reduction basis with fixed interface vibration modes, thereby obtaining a more complete and versatile basis to describe the component's dynamic behavior.

Again starting from the equations of motion partitioned into internal and boundary displacement DoF, the internal DoF are approximated as follows:

$$\boldsymbol{u}_i \approx \boldsymbol{\Psi}_c \boldsymbol{u}_b + \boldsymbol{\Phi}_i \boldsymbol{\eta}_i \tag{2.54}$$

Here $\boldsymbol{\Phi}_i$ is a reduced set of $m \ll n_i$ fixed interface vibration modes, obtained as outlined in section 2.4.3. Like in the Guyan method the boundary DoF are retained, leading to the following Craig-Bampton reduction basis:

$$\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi}_i & \boldsymbol{\Psi}_c \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{u}_b \end{bmatrix} = \boldsymbol{R}_{CB} \boldsymbol{q}_{CB} \tag{2.55}$$

Application of the reduction basis in the usual fashion gives the reduced equations of motion:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{M}_{\phi b} \\ \boldsymbol{M}_{b\phi} & \tilde{\boldsymbol{M}}_{bb} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_i \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Omega}_i^2 & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{f}}_i \\ \tilde{\boldsymbol{f}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_b \end{bmatrix} \tag{2.56}$$

With:

$$\begin{aligned}
\tilde{\boldsymbol{K}}_{bb} &= \boldsymbol{K}_{bb} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} \\
\tilde{\boldsymbol{M}}_{bb} &= \boldsymbol{M}_{bb} - \boldsymbol{M}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{M}_{ib} + \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{M}_{ii}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} \\
\boldsymbol{M}_{\phi b} &= \boldsymbol{M}_{b\phi}^T = \boldsymbol{\Phi}_i^T \left(\boldsymbol{M}_{ib} - \boldsymbol{M}_{ii}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}\right) \\
\tilde{\boldsymbol{f}}_i &= \boldsymbol{\Phi}_i^T \boldsymbol{f}_i \\
\tilde{\boldsymbol{f}}_b &= \boldsymbol{f}_b - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{f}_i
\end{aligned} \tag{2.57}$$

Here $\boldsymbol{\Omega}_i^2$ is a diagonal matrix containing the fixed interface vibration frequencies $\omega_{i,j}^2$.

One of the strengths of the Craig-Bampton method is the straightforward calculation of its reduction basis. Secondly, like in the Guyan method, the physical interface DoF $\boldsymbol{u}_b$ are retained in the reduced model, which facilitates easy assembly of the reduced substructure as a superelement in common FE codes.

A major drawback of the Craig-Bampton method is that if the substructure interface is changed, for instance when connecting more components, the entire reduction basis needs to be recomputed. Another issue is that, due to the type of ingredients used in the basis, the Craig-Bampton reduction basis can in practice not be obtained experimentally.

2.5.3 Rubin and MacNeal Methods

After Craig and Bampton published their fixed interface CMS method in 1968, researchers were soon focussing on using free interface vibration modes in the reduction basis. This seems natural, as these are the "true" modes of the system and can be experimentally determined by testing the component in free-free conditions. These modes were first used in a CMS method by MacNeal in 1971 [129] and later by Rubin in 1975 [176]. These methods, and their relations with other component reduction methods proposed at that time, were further analyzed by Craig and Chang [38, 39].[4]

To derive the Rubin and MacNeal methods, recall the unreduced component equations:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} + \boldsymbol{g}$$

---

[4] In the literature one sometimes encounters the "Craig-Chang" method, which is in fact an alternative, more systematic derivation of the Rubin method. In their paper [39], Craig and Chang show that the Rubin method indeed is a Rayleigh-Ritz procedure, while Rubin originally claimed this not to be the case. Both derivations lead to the same reduced system as obtained in section 2.5.3.2.

Next, the original set of substructure displacement DoF is written as:

$$\boldsymbol{u} = \boldsymbol{u}_{\text{stat}} + \sum_{j=n_r+1}^{n} \boldsymbol{\phi}_{f,j}\eta_{f,j} \tag{2.58}$$

where the total response of the substructure is represented in terms of the free vibration modes of the substructure and a static solution. Again n_r is the number of rigid body modes of the substructure. The static response can be expressed as:

$$\boldsymbol{u}_{\text{stat}} = \boldsymbol{\Psi}_r\boldsymbol{g}_b + \boldsymbol{\Phi}_r\boldsymbol{\eta}_r \tag{2.59}$$

In this expression the first term describes the static flexible response to the interface forces, using the residual attachment modes described in section 2.3.3. The second term gives the rigid body mode contribution (see section 2.4.2 for their computation) with $\boldsymbol{\eta}_r$ the RBM amplitudes. The reduction is now obtained by approximating the transformation in eq. (2.58) by taking only the first $m \ll n$ free interface vibration modes:[5]

$$\boldsymbol{u} \approx \boldsymbol{\Psi}_r\boldsymbol{g}_b + \boldsymbol{\Phi}_r\boldsymbol{\eta}_r + \boldsymbol{\Phi}_f\boldsymbol{\eta}_f \tag{2.60}$$

This transformation can now be used to reduce the component equations of motion. Using the orthogonality properties of the residual attachment modes (see section 2.3.3) the reduced equations of motion become:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}_{r,bb} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_r \\ \ddot{\boldsymbol{\eta}}_f \\ \ddot{\boldsymbol{g}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Omega}_f^2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{G}_{r,bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_r^T \\ \boldsymbol{\Phi}_f^T \\ \boldsymbol{\Psi}_r^T \end{bmatrix} \boldsymbol{f} + \begin{bmatrix} \boldsymbol{\Phi}_r^T \\ \boldsymbol{\Phi}_f^T \\ \boldsymbol{\Psi}_r \end{bmatrix} \boldsymbol{g} \tag{2.61}$$

with:

$$\begin{aligned} \boldsymbol{G}_{r,bb} &= \boldsymbol{\Psi}_r^T\boldsymbol{K}\boldsymbol{\Psi}_r = \boldsymbol{A}\boldsymbol{G}_r\boldsymbol{A}^T \\ \boldsymbol{M}_{r,bb} &= \boldsymbol{\Psi}_r^T\boldsymbol{M}\boldsymbol{\Psi}_r \end{aligned} \tag{2.62}$$

Here $\boldsymbol{G}_{r,bb}$ is the residual flexibility on the interface, while $\boldsymbol{M}_{r,bb}$ is the interface inertia associated to the residual flexibility modes. Hence this term will be referred to as the "residual mass".

The reduced equations in eq. (2.61) give rise to a so-called "dual" system (see chapter 3), where the generalized DoF vector contains the modal amplitudes $\boldsymbol{\eta}$ and the interface force DoF $\boldsymbol{g}_b$. In order to facilitate assembly of the reduced model in existing finite element software as a superelement, both MacNeal and Rubin chose to apply a second transformation to the equations of motion to obtain interface displacement DoF instead of force DoF. Such a transformation is found by pre-multiplying eq. (2.60) by the Boolean matrix $\boldsymbol{A}$ which selects the component's interface DoF (see section 2.3.2):

$$\boldsymbol{u}_b = \boldsymbol{A}\boldsymbol{u} = \boldsymbol{A}\left(\boldsymbol{\Psi}_r\boldsymbol{g}_b + \boldsymbol{\Phi}_r\boldsymbol{\eta}_r + \boldsymbol{\Phi}_f\boldsymbol{\eta}_f\right) = \boldsymbol{G}_{r,bb}\boldsymbol{g}_b + \boldsymbol{\Phi}_{r|b}\boldsymbol{\eta}_r + \boldsymbol{\Phi}_{f|b}\boldsymbol{\eta}_f \tag{2.63}$$

---

[5] Note that the residual attachment modes can only be calculated after the number of modes m has been chosen. In practice one therefore first computes the attachment modes and subsequently subtracts the contribution of the free vibration modes once these are available.

Here the subscript $_{|b}$ indicates the trace on the boundary DoF. From this equation the interface force DoF can solved as:

$$\boldsymbol{g}_b = \boldsymbol{K}_{r,bb}\left(\boldsymbol{u}_b - \boldsymbol{\Phi}_{r|b}\boldsymbol{\eta}_r - \boldsymbol{\Phi}_{f|b}\boldsymbol{\eta}_f\right) \quad \text{with} \quad \boldsymbol{K}_{r,bb} = \boldsymbol{G}^{-1}_{r,bb} \tag{2.64}$$

It is observed that $\boldsymbol{K}_{r,bb}$ is in fact the boundary residual stiffness associated to all discarded modes. The transformation from interface force DoF to displacement DoF is thus found as:

$$\begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{K}_{r,bb}\boldsymbol{\Phi}_{r|b} & -\boldsymbol{K}_{r,bb}\boldsymbol{\Phi}_{f|b} & \boldsymbol{K}_{r,bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{u}_b \end{bmatrix} \tag{2.65}$$

Up to this point the MacNeal and Rubin methods are identical. However, hereafter the two methods differ in the way the reduced mass matrix is treated. These differences will be highlighted next.

2.5.3.1 MacNeal's method

In MacNeal's method, the residual mass term $\boldsymbol{M}_{r,bb}$ in eq. (2.61) is neglected. This can be achieved in two ways. In a first approach, one simply sets $\boldsymbol{M}_{r,bb} = \boldsymbol{0}$ in eq. (2.61) and pre- and post-multiplies the resulting equation by the transformation in (2.65). In a second approach, the stiffness matrix is also obtained from transformation of the stiffness matrix in eq. (2.65), but a different reduction basis, namely employing only the rigid body and vibration modes, is used to reduce the mass matrix.

Either way, the following reduced equations of motion are obtained:

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_r \\ \ddot{\boldsymbol{\eta}}_f \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \dots \\ &\begin{bmatrix} \boldsymbol{\Phi}_r^T\bar{\boldsymbol{K}}_r\boldsymbol{\Phi}_r & \boldsymbol{\Phi}_r^T\bar{\boldsymbol{K}}_r\boldsymbol{\Phi}_f & -\boldsymbol{\Phi}_{r|b}^T\boldsymbol{K}_{r,bb} \\ \boldsymbol{\Phi}_f^T\bar{\boldsymbol{K}}_r\boldsymbol{\Phi}_r & \boldsymbol{\Omega}_f^2 + \boldsymbol{\Phi}_f^T\bar{\boldsymbol{K}}_r\boldsymbol{\Phi}_f & -\boldsymbol{\Phi}_{f|b}^T\boldsymbol{K}_{r,bb} \\ -\boldsymbol{K}_{r,bb}\boldsymbol{\Phi}_{r|b} & -\boldsymbol{K}_{r,bb}\boldsymbol{\Phi}_{f|b} & \boldsymbol{K}_{r,bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{f}}_r \\ \tilde{\boldsymbol{f}}_f \\ \tilde{\boldsymbol{f}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{g}_b \end{bmatrix} \end{aligned} \tag{2.66}$$

Here $\boldsymbol{\Omega}_f^2$ is again a diagonal matrix containing the free eigenfrequencies, and:

$$\begin{aligned} \bar{\boldsymbol{K}}_r &= \boldsymbol{A}^T\boldsymbol{K}_{r,bb}\boldsymbol{A} \\ \tilde{\boldsymbol{f}}_r &= \left(\boldsymbol{\Phi}_r^T - \boldsymbol{\Phi}_{r|b}^T\boldsymbol{K}_{r,bb}\boldsymbol{\Psi}_r^T\right)\boldsymbol{f} \\ \tilde{\boldsymbol{f}}_f &= \left(\boldsymbol{\Phi}_f^T - \boldsymbol{\Phi}_{f|b}^T\boldsymbol{K}_{r,bb}\boldsymbol{\Psi}_r^T\right)\boldsymbol{f} \\ \tilde{\boldsymbol{f}}_b &= \boldsymbol{K}_{r,bb}\boldsymbol{\Psi}_r^T\boldsymbol{f} \end{aligned} \tag{2.67}$$

Although the component is reduced using free vibration modes and residual flexibility modes, the reduction process still leads to a superelement since the physical boundary DoF $\boldsymbol{u}_b$ are reintroduced in the generalized DoF.

However, the equations of motion obtained by MacNeal's method are in fact inconsistent. This is due to the fact that the mass and stiffness matrices are not reduced with the same basis [36, 73]: whereas the stiffness matrix is reduced using both the free vibration and residual attachment modes, the mass matrix is only reduced using the free vibration modes. As a result, there is no inertia coupling between the modal DoF and the boundary DoF. Hence, the MacNeal method is not a true Rayleigh-Ritz method in the sense that the reduced equations are not obtained from a consistent reduction and projection.

2.5.3.2 Rubin's Method

In Rubin's method the equations of motion are reduced in a consistent way. That is, equation (2.61), including the term $\boldsymbol{M}_{r,bb}$, is transformed using (2.65) to find the reduced equations of motion:

$$\begin{bmatrix} \boldsymbol{I} + \boldsymbol{\Phi}_r^T \bar{\boldsymbol{M}}_r \boldsymbol{\Phi}_r & \boldsymbol{\Phi}_r^T \bar{\boldsymbol{M}}_r \boldsymbol{\Phi}_f & -\boldsymbol{\Phi}_{r|b}^T \bar{\boldsymbol{M}}_r \\ \boldsymbol{\Phi}_f^T \bar{\boldsymbol{M}}_r \boldsymbol{\Phi}_r & \boldsymbol{I} + \boldsymbol{\Phi}_f^T \bar{\boldsymbol{M}}_r \boldsymbol{\Phi}_f & -\boldsymbol{\Phi}_{f|b}^T \bar{\boldsymbol{M}}_r \\ -\bar{\boldsymbol{M}}_r \boldsymbol{\Phi}_{r|b} & -\bar{\boldsymbol{M}}_r \boldsymbol{\Phi}_{f|b} & \boldsymbol{M}_{r,bb} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_r \\ \ddot{\boldsymbol{\eta}}_f \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \dots$$
$$\begin{bmatrix} \boldsymbol{\Phi}_r^T \bar{\boldsymbol{K}}_r \boldsymbol{\Phi}_r & \boldsymbol{\Phi}_r^T \bar{\boldsymbol{K}}_r \boldsymbol{\Phi}_f & -\boldsymbol{\Phi}_{r|b}^T \boldsymbol{K}_{r,bb} \\ \boldsymbol{\Phi}_f^T \bar{\boldsymbol{K}}_r \boldsymbol{\Phi}_r & \boldsymbol{\Omega}_f^2 + \boldsymbol{\Phi}_f^T \bar{\boldsymbol{K}}_r \boldsymbol{\Phi}_f & -\boldsymbol{\Phi}_{f|b}^T \boldsymbol{K}_{r,bb} \\ -\boldsymbol{K}_{r,bb} \boldsymbol{\Phi}_{r|b} & -\boldsymbol{K}_{r,bb} \boldsymbol{\Phi}_{f|b} & \boldsymbol{K}_{r,bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{f}}_r \\ \tilde{\boldsymbol{f}}_f \\ \tilde{\boldsymbol{f}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{g}_b \end{bmatrix} \tag{2.68}$$

where the reduced force vectors are stated in eq. (2.67) and the term $\tilde{\boldsymbol{M}}_r$ is defined as:

$$\bar{\boldsymbol{M}}_r = \boldsymbol{A}^T \boldsymbol{K}_{r,bb} \boldsymbol{M}_{r,bb} \boldsymbol{K}_{r,bb} \boldsymbol{A} \tag{2.69}$$

Clearly, the reduced stiffness matrices of the MacNeal and Rubin methods are identical but the mass matrices are different. Indeed, the Rubin reduced mass matrix is a consistent, fully coupled matrix.

Similar to MacNeal's method, the reduced model obtained from Rubin's method can be simply assembled to other component models as a superelement. Note that the approximation in eq. (2.60) and transformation (2.65) can be combined to obtain the Rubin reduction matrix:

$$\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi}_{r|i} - \boldsymbol{\Psi}_{r|i} \boldsymbol{K}_{r,bb} \boldsymbol{\Phi}_{r|b} & \boldsymbol{\Phi}_{f|i} - \boldsymbol{\Psi}_{r|i} \boldsymbol{K}_{r,bb} \boldsymbol{\Phi}_{f|b} & \boldsymbol{\Psi}_{r|i} \boldsymbol{K}_{r,bb} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{u}_b \end{bmatrix} = \boldsymbol{R}_R \boldsymbol{q}_R \tag{2.70}$$

This reduction matrix can be directly applied to the component equations of motion to obtain the same result as in eq. (2.68). This shows that, in contrast to MacNeal's method, the Rubin method is a true Rayleigh-Ritz method.

An advantage of both the Rubin and MacNeal methods over the Craig-Bampton method is that the reduction basis only partially changes if the interface DoF set is altered. The free interface modes and associated parts of the reduced matrices remain unchanged. Only the residual attachment modes have to be recomputed, which amounts to reselecting columns from the residual flexibility matrix, and the associated parts of the reduced matrices have to be updated. In contrast to the Craig-Bampton method, this involves very little computational effort. Furthermore, experimental results can be used to validate the reduced model, since experimentally determined mode shapes are usually free interface modes.

A weakness of these methods is the rather cumbersome expression of their reduced matrices which are no longer sparse. Such full reduced matrices increase the computational cost of analyzing the reduced models. Although these are usually compact, they can still be relatively large if the component has many interface DoF. This is a general problem in component model reduction, see chapter 4, but becomes worse when the reduced matrices are full. Note that this is less of an issue for the MacNeal method with its diagonal mass matrix, but that comes at the price of decreased accuracy with respect to Rubin's method.

2.5.4 Dual Craig-Bampton Method

2.5.4.1 Original Formulation

A more recent component model reduction method is the Dual Craig-Bampton method [172], that is inspired by both the Craig-Bampton and the Rubin method. In fact, it uses the same approximation for the component's displacements as the latter (eq. (2.60)), namely:

$$\boldsymbol{u} \approx \boldsymbol{\Psi}_r \boldsymbol{g}_b + \boldsymbol{\Phi}_r \boldsymbol{\eta}_r + \boldsymbol{\Phi}_f \boldsymbol{\eta}_f \tag{2.71}$$

However, where the Rubin method transforms the interface forces back to interface displacement DoF to enable easy assembly of reduced structures, the Dual Craig-Bampton method explicitly keeps the interface forces as part of the generalized DoF. This means that no longer a true superelement is obtained and a special assembly procedure must be employed, called "dual" assembly (hence the name of the method). See section 3.4. To this end, the reduction basis in eq. (2.71) can be written in matrix-vector form as:

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{g}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi}_r & \boldsymbol{\Phi}_f & \boldsymbol{\Psi}_r \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{g}_b \end{bmatrix} = \boldsymbol{R}_{DCB} \boldsymbol{q}_{DCB} \tag{2.72}$$

In addition to the original substructure displacements $\boldsymbol{u}$ the interface forces $\boldsymbol{g}_b$ are included in the new set of DoF. The substructure's equation of motion can thus be written as:

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}} \\ \ddot{\boldsymbol{g}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{K} & -\boldsymbol{A}^T \\ -\boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{u}_b \end{bmatrix} \tag{2.73}$$

The second equation seems redundant, but is added for symmetry and is used to enforce compatibility during assembly. By substituting and projecting these equations onto the reduction basis and using the properties of the residual flexibility matrix, the following reduced equations of motion are found:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}_{r,bb} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_r \\ \ddot{\boldsymbol{\eta}}_f \\ \ddot{\boldsymbol{g}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{\Phi}_{r|b}^T \\ \boldsymbol{0} & \boldsymbol{\Omega}_f^2 & -\boldsymbol{\Phi}_{f|b}^T \\ -\boldsymbol{\Phi}_{r|b} & -\boldsymbol{\Phi}_{f|b} & -\boldsymbol{G}_{r,bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_r^T \boldsymbol{f} \\ \boldsymbol{\Phi}_f^T \boldsymbol{f} \\ \boldsymbol{\Psi}_r^T \boldsymbol{f} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{u}_b \end{bmatrix} \tag{2.74}$$

Here the terms $\boldsymbol{M}_{r,bb}$, $\boldsymbol{G}_{r,bb}$ and $\boldsymbol{\Phi}_{|b}$ are used as defined in the previous section. Assembly of substructures reduced in this way involves "coupling" of the interface forces, thereby resulting in a true dual system. This will be described in section 3.4.

The Dual Craig-Bampton (DCB) method shares many of the positive features of the MacNeal/Rubin methods, namely easy adaptation of the reduction basis to changing interface

DoF and the possibility to obtain or validate the reduction basis ingredients experimentally. In addition, the DCB method gives rise to sparse reduced matrices like in the regular Craig-Bampton, overcoming one of the drawbacks of the MacNeal/Rubin methods. On the downside, the DCB method does not lead to a classic superelement and hence assembly is less straightforward.

The main difference between the MacNeal/Rubin and DCB methods is that in the latter not only the dynamic equilibrium is reduced, but also the interface compatibility condition is approximated (i.e. the second line in equation (2.73)). This leads to a weakening of the compatibility condition, which can be understood from the last line in eq. (2.74). Assuming no external force, this reduced compatibility condition writes:

$$\boldsymbol{\Phi}_{r|b}\boldsymbol{\eta}_r + \boldsymbol{\Phi}_{f|b}\boldsymbol{\eta}_f + \boldsymbol{G}_{r,bb}\boldsymbol{g}_b = \boldsymbol{u}_b - \boldsymbol{M}_{r,bb}\ddot{\boldsymbol{g}}_b \quad \rightarrow \quad \boldsymbol{A}\boldsymbol{u} = \boldsymbol{u}_b + \boldsymbol{\epsilon} \tag{2.75}$$

Hence, the presence of the last term allows a small error on the compatibility equation. This observation can be interpreted as follows. The last line in (2.74) can be seen as obtained from:

$$\begin{bmatrix} \boldsymbol{\Psi}_r^T & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \text{Reduced dynamic equilibrium} \\ \text{Compatibility condition} \end{bmatrix}$$

Due to the reduction residual forces $\boldsymbol{r}$ arise in the substructure dynamic equilibrium. If those forces would be statically applied to the substructure through the residual flexibility, a residual displacement $\boldsymbol{u}_r = \boldsymbol{G}_r\boldsymbol{r}$ would result. The weak compatibility then states that the compatibility error can be as big as the trace of displacements $\boldsymbol{u}_r$ on the boundary, i.e. $\boldsymbol{u}_{r|b} = \boldsymbol{A}\boldsymbol{u}_r = \boldsymbol{\Psi}_r^T\boldsymbol{r}$.

Since the incompatibility allows motions which are physically impossible (e.g. overlapping of interface meshes), so-called spurious modes can be introduced in the frequency range of interest if the reduction basis is too poor. Two ways exist to shift these spurious modes to higher frequencies: enrich the reduction basis with more normal modes[6] and/or enrich the basis with higher order residual modes. The latter will be described in section 2.7.

Finally, note that if the last term in equation (2.75) is neglected (i.e. $\boldsymbol{M}_{r,bb} = \boldsymbol{0}$), exact compatibility is required as in the MacNeal method. In this case, the DCB and MacNeal methods are in fact equivalent. This can be easily shown by transforming the interface force DoF back to displacement DoF using eq. (2.75) with $\boldsymbol{M}_{r,bb} = \boldsymbol{0}$:

$$\boldsymbol{g}_b = \boldsymbol{K}_{r,bb}\left(\boldsymbol{u}_b - \boldsymbol{\Phi}_{r|b}\boldsymbol{\eta}_r - \boldsymbol{\Phi}_{f|b}\boldsymbol{\eta}_f\right) \tag{2.76}$$

Indeed, this is the same transformation of eq. (2.64) used by MacNeal/Rubin, and substitution in eq. (2.74) with $\boldsymbol{M}_{r,bb} = \boldsymbol{0}$ directly gives MacNeal's reduced equations of motion.

2.5.4.2 Alternative Formulation Using Attachment Modes

In its original formulation, the Dual Craig-Bampton method employs residual attachment modes to describe the static response of the component to unit boundary forces. This results

---

[6] Note that if one would retain all the eigenmodes in the reduction basis, both $\boldsymbol{M}_{r,bb}$ and $\boldsymbol{G}_{r,bb}$ would vanish and exact compatibility is satisfied.

in the reduced matrices in eq. (2.74), where the mass matrix is (block) diagonal and the coupling between the modal and interface force DoF occurs in the stiffness matrix. However, one can also choose to use attachment modes in the basis which, as will be shown below, give rise to reduced equations that are mainly coupled in the mass matrix. This is beneficial for the Mixed Craig-Bampton method derived in section 2.6 and allows to further reduce the number of DoF using interface reduction, as explained in section 4.5.

With attachment modes the alternative DCB reduction basis becomes:

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{g}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi}_r & \boldsymbol{\Phi}_f & \boldsymbol{\Psi}_a \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{g}_b \end{bmatrix} = \boldsymbol{R}_{DCBa} \boldsymbol{q}_{DCB} \tag{2.77}$$

Note that this alternative formulation does not alter the space spanned by the reduction basis; the same space is just represented using slightly different vectors.

In the same way as for the normal DCB the reduced equations of motion are obtained by reducing both the dynamic equilibrium and compatibility equations. Using the orthogonality properties of the attachment modes and flexibility matrix, as discussed in section 2.3.2, this yield the alternative DCB reduced system as:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{M}_{\phi\psi} \\ \boldsymbol{0} & \boldsymbol{M}_{\psi\phi} & \boldsymbol{M}_{f,bb} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_r \\ \ddot{\boldsymbol{\eta}}_f \\ \ddot{\boldsymbol{g}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{\Phi}_{r|b}^T \\ \boldsymbol{0} & \boldsymbol{\Omega}_f^2 & \boldsymbol{0} \\ -\boldsymbol{\Phi}_{r|b} & \boldsymbol{0} & -\boldsymbol{G}_{f,bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_r^T \boldsymbol{f} \\ \boldsymbol{\Phi}_f^T \boldsymbol{f} \\ \boldsymbol{\Psi}_a^T \boldsymbol{f} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{u}_b \end{bmatrix} \tag{2.78}$$

Where:

$$\begin{aligned} \boldsymbol{M}_{f,bb} &= \boldsymbol{\Psi}_a^T \boldsymbol{M} \boldsymbol{\Psi}_a = \boldsymbol{A} \boldsymbol{G}_f \boldsymbol{M} \boldsymbol{G}_f \boldsymbol{A}^T \\ \boldsymbol{M}_{\phi\psi} &= \boldsymbol{M}_{\psi\phi}^T = \boldsymbol{\Phi}_f^T \boldsymbol{M} \boldsymbol{\Psi}_a \\ \boldsymbol{G}_{f,bb} &= \boldsymbol{A} \boldsymbol{G}_f \boldsymbol{A}^T \end{aligned} \tag{2.79}$$

The reduced equations are now mainly coupled in the mass matrix instead of the in the stiffness matrix as in equation (2.74). In the stiffness matrix one still finds coupling terms between the rigid body modes and boundary force DoF. This cannot be avoided since the rigid motions must be considered as part of the static solution, as in eq. (2.59). In case no rigid body modes exist, the reduced stiffness matrix is (block) diagonal.

2.5.5 Remarks, Variations and Other Methods

Regarding the component reduction methods treated in this section, a number of general remarks can be made:

- Although theoretically the boundaries of the component models may be chosen arbitrarily, it is in practice more convenient to adopt the physical component boundaries. The latter is the approach taken in this work.

- In a practical implementation of the reduction methods, it is not advisable to obtain the reduced matrices through pre- and post-multiplication by the reduction basis. Due to numerical round-off errors, non-zero off-diagonal terms will appear that should in fact be zero, leading to ill-conditioned matrices and additional computational cost. Instead, the reduced matrices should be directly computed from their detailed expressions.

- All reduction methods treated in this section are implemented in the Dynamic Substructuring Toolbox (see appendix A). This is also true for the Mixed Craig-Bampton method presented in the next section as well as the Modal Truncation Augmentation technique treated in section 2.7.

Furthermore, in addition to the well-established methods described in the previous subsections, many variations have been proposed over the years. Some of the important contributions are briefly outlined below. This overview and the references mentioned herein are however by no means exhaustive.

- Many authors have proposed different modes to build a reduction basis that better approximates the substructures' dynamics:
  - Already at the time when Hurty published the idea of substructuring, the branch method [75] was proposed where the reduction modes are evaluated by taking into account the influence of neighboring substructures.
  - As indicated before, several authors have proposed to replace the static modes in the reduction basis by quasi-static modes related to a dynamic stiffness matrix obtained by shifting around a central frequency [124, 188, 187].
  - Some authors have proposed to add masses to the interface when measuring or computing the vibration modes in order to account for the inertia loading of the neighboring substructures [107, 31]. A direct extension of these concepts is the "transmission simulator" method [6]. In this method a fixture is used to condition the interfaces of experimentally obtained substructure models, in order to improve the results of hybrid substructuring analyses.
  - Instead of using vibration modes, which represent the general dynamics of the components, one can also use the so-called Krylov vectors related to the interface loading [43] which in fact arise from the concept of the load-dependent vectors proposed by Wilson for model reduction [222].
  - A way to include both true eigenmodes and Krylov vectors consists in adding to the reduction basis so-called Modal Truncation Augmentation vectors. Such vectors can be used to improve the response of the reduced component model to loading at its internal DoF. These methods will discussed in section 2.7.
- Model reduction for non-linear systems is a research field that is gaining more and more attention. For such systems the challenge is to find an a priori reduction basis that properly captures all of the structure's relevant non-linear behavior. Various approaches have been proposed over the years to do so. One approach is based on the use of the proper orthogonal decomposition (POD) method, where a reduction basis is obtained by extracting the dominant deformation shapes from representative non-linear response data [112, 126, 84].[7] Another approach is to use so-called *non-linear normal modes* to reduce the non-linear model. These modes are a generalization of the classic linear vibration modes and can be computed analytically/numerically but can also be determined from experiments. See for instance [113, 151, 150, 5]. Other

---

[7] This method is also used in this work to create load dependent reduction vectors, see section 2.7 and appendix D for details.

approaches for non-linear model reduction include the use of so-called modal derivatives also known as higher order modes [98, 182, 198, 136], or the empirical approach proposed in the DEIM (Discrete Emperical Interpolation Method) technique [33, 92].

- A special situation is encountered when components are modeled that undergo large motions, for instance the blades of a wind turbine. When formulating the equations of motion for such systems non-linear coupling terms arise between the local deformations and global motions, even when the structural deformations remain small. Various methods can be found in the literature to incorporate reduced component models in such large motion formulations, e.g. [73, 186, 93]. Note that other approaches have also been developed where instead of individual component reduction, model reduction is applied to the assembled multi-body system [25, 26, 89, 88]

  The non-linear terms associated with these large motion formulations greatly increase the computational cost of analyzing the reduced model. In the context of this PhD project, a study has therefore been performed to identify which of the non-linear terms may be simplified or linearized when reduced component modeling is to be used to represent a wind turbine rotor (e.g. blades, hub, pitch system, etc.). Based on the *floating frame of reference* formulation [186] and a simple yet representative wind turbine model, various simplifications of the non-linear formulation have been proposed and evaluated for different load cases.

  In short, it was concluded that for operational load cases, where the rotational speeds of the rotor are more or less constant, the simplified formulations could accurately predict the response of the turbine at much lower computational cost than the full non-linear formulation. For highly transient load cases such as wind gusts and turbine shutdowns however, most simplifications gave rise to inaccuracies and only minor simplifications of the non-linear terms could be justified. The results of this work are reported in [55]; a detailed treatment is however out of the scope of this thesis.

- For damped systems, classical reduction bases can lead to full reduced damping matrices [156] or might not be adequate for an efficient reduction. In that case some authors have proposed to consider the state-space form of the equation and use the associated modes and Krylov vectors [41, 45, 196]. Also, various other approaches have been proposed to handle for instance issues such as frequency dependent damping [16], tuning of damping of global modes based on the substructure contributions [175], as well as methods to find equivalent viscous damping models to account for hysteretic structural damping [69, 9].

- Multilevel substructuring consists in performing nested decomposition of the system and successively reducing the components starting at the lowest level. Any reduction technique can be applied in a multi-level manner. One popular technique is the Automated Multilevel Substructuring method or AMLS [20, 21].

- Some publications have addressed the issue of reducing the interface problem [40, 23, 8]. This is an important issue since in many practical applications the number of degrees of freedom on the interface is still unnecessarily high. A second reduction step for the interface DoF can then be performed. Interface reduction methods will be discussed in detail in section 4.2.

- A number of authors have proposed to iteratively construct or improve the reduction basis, using the error residual from the previous iteration [13, 155, 24]. Closely related to this work is the issue of error estimation and adaptive refinement of reduced models, which is the topic of chapter 5.

- Building substructured models that undergo parametric modifications (in design optimization procedures or for model updating) is an active research area. Several techniques have been proposed to update or enrich the reduction basis and to reduce families of models [14, 56, 133, 30]. In chapter 6 this problem will be addressed in detail.

2.6 The Mixed Craig-Bampton Method

In the previous section the most common component reduction methods were reviewed. These methods can be classified according to many criteria, but one classification often seen in the literature (e.g. in [36]) is the distinction between "fixed interface" and "free interface" methods, referring to the type of vibration mode employed in the reduction basis. The most well known variant of the former class is the Craig-Bampton method, while the Rubin/MacNeal and Dual Craig-Bampton methods are examples of the latter.

From this classification a question that naturally arises is: when should one use fixed interface methods and when are free interface methods to be preferred? This question is relevant when a situation is imagined where two components are assembled: component one is stiff and/or heavy whereas substructure two is flexible and/or lightweight. One can imagine that after assembly the stiff structure will behave as if its interface is still quasi-free, thereby largely dictating the motion of the flexible structure through its interface. Ideally, one would thus reduce substructure one using a free interface mode and component two using fixed interface modes. However, this can still lead to difficulties where one of the substructures is for example very stiff in one direction or at one location but relatively flexible in another direction/location. In this case one would want to employ a mix of both methods per substructure. In other words, some interface degrees of freedom should be fixed in the model reduction while others are left free: a mixed boundary approach to model reduction.

In this section the Mixed Craig-Bampton (MCB) method is proposed to handle this problem. As will be proven later, this method is a natural mixed boundary generalization of the existing CB and DCB methods. In general, the MCB method comprises the following steps:

1. For all components in the assembly define the interfaces
2. Using some criterion, choose boundary conditions for all pairs of interface DoF
3. Calculate the reduction bases for the components and compute the reduced matrices
4. Assemble the reduced components in the correct manner

The first step is no different than for other CMS methods. The subsequent steps however are non-standard and will be discussed next. It should be noted that the reduction of substructures using the Mixed Craig-Bampton method cannot be performed separately, since the reduction basis of one substructure is dependent on the properties of its neighboring components.

2.6.1 Selection of Free or Fixed Interface Vibration Modes

In order to select fixed or free modes for the reduction basis of the components, some criterion must be established. To this end, an *a priori* estimate is needed of the components' behavior; ideally the responses of all components to a unit load or displacement at all their interface DoF. This is however computationally inefficient, so an approximation can be made by estimating the substructure behavior simply from the value on the diagonal of the stiffness matrix corresponding to the interface DoF. Then, three cases can be distinguished and the following selection scheme is proposed:

1. In the first case, subsystem 1 is much stiffer than subsystem 2:

   $$K_{ii}^{(1)}/K_{ii}^{(2)} > 10^{c}$$

   Here c is some constant that can be chosen to suit the problem at hand. In this case, subsystem 1 will feel some connection forces through its interface but will not be influenced much by the presence of its neighboring substructure, behaving nearly as if it were free. The motion of component 2 will however be largely dictated through its interface with component 1. The natural choice would thus be to let free the interface DoF of component 1 in the reduction, while the corresponding interface DoF of substructure 2 should be fixed. The DoF that remain free will be denoted as "dual" DoF, while fixed DoF are called "primal".

2. In the second case, the stiffness at the interface DoF of both subsystems is of approximately the same order of magnitude, i.e.:

   $$10^{-c} \leq K_{ii}^{(1)}/K_{ii}^{(2)} \leq 10^{c}$$

   In this case, both interface DoF can be reduced with either fixed or free interface modes. The choice for fixed or free modes can be made per set of interface DoF, although a consistent choice for the complete assembly leads to a simpler assembly procedure.

3. In the third case, subsystem 2 is much stiffer then 1:

   $$K_{ii}^{(1)}/K_{ii}^{(2)} < 10^{-c}$$

   Using the same reasoning as before, the natural choice is to reduce subsystem 2 with free interface modes and system 1 with fixed interface modes.

So, the goal of the selection scheme is to use some (simple) knowledge of the the assembled system in order to construct a better reduction basis for the components. As such, the method is somewhat similar to the application of Robin-type interface conditions (see e.g. [105]), where stiffness and mass of the neighboring substructures is in some form taken into account in the component reduction. Thereby the modes used in the component reduction bases more closely represent the assembled mode shapes.

Using the above selection scheme, the substructure DoF vector $\boldsymbol{u}^{(s)}$ can be partitioned into internal DoF $\boldsymbol{u}_i^{(s)}$, "dual" DoF $\boldsymbol{u}_d^{(s)}$ and "primal" DoF $\boldsymbol{u}_p^{(s)}$, as:

$$\boldsymbol{u}^{(s)} = \begin{bmatrix} \boldsymbol{u}_i^{(s)} & \boldsymbol{u}_d^{(s)} & \boldsymbol{u}_p^{(s)} \end{bmatrix}^T \tag{2.80}$$

Now the DoF set $\boldsymbol{u}_m^{(s)}$ is introduced, which comprises the set of internal plus "dual" DoF, to denote the DoF that will be replaced by generalized DoF in the reduction:

$$\boldsymbol{u}_m^{(s)} = \begin{bmatrix} \boldsymbol{u}_i^{(s)} & \boldsymbol{u}_d^{(s)} \end{bmatrix}^T \tag{2.81}$$

The above division of DoF will be used in the next subsection to find the reduction basis. It should be remarked that the above proposed selection method can be easily automated but is useful only if the component interface coincides with the material interface. If this is not the case, the values on the diagonal of the stiffness matrix not truly reflect the "global" stiffness of the system (imagine for instance a rubber bushing with a metal core). In such situations one should use some other criterion for selecting fixed/free modes or resort to "engineering judgement".

2.6.2 Mixed Component Reduction

Given the partitioning of substructure DoF introduced above, the partitioned equations of motion become (again the substructure denotation $^{(s)}$ is omitted for simplicity):

$$\begin{bmatrix} \boldsymbol{M}_{mm} & \boldsymbol{M}_{mp} \\ \boldsymbol{M}_{pm} & \boldsymbol{M}_{pp} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_m \\ \ddot{\boldsymbol{u}}_p \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{mm} & \boldsymbol{K}_{mp} \\ \boldsymbol{K}_{pm} & \boldsymbol{K}_{pp} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_m \\ \boldsymbol{u}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_m \\ \boldsymbol{f}_p \end{bmatrix} + \begin{bmatrix} \boldsymbol{g}_m \\ \boldsymbol{g}_p \end{bmatrix} \tag{2.82}$$

It should now be realized that the DoF in $\boldsymbol{u}_m$ will be reduced and assembly of the $\boldsymbol{u}_d$ in this DoF set will be performed using interface forces. Hence, similar to the Dual Craig-Bampton method, the interface forces $\boldsymbol{g}_d$ need to be included in the DoF vector. Furthermore, the compatibility equation is added to both ensure symmetry of the equations and allow weakening of this condition to avoid interface locking. This gives:

$$\begin{bmatrix} \boldsymbol{M}_{mm} & \boldsymbol{M}_{mp} & \boldsymbol{0} \\ \boldsymbol{M}_{pm} & \boldsymbol{M}_{pp} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_m \\ \ddot{\boldsymbol{u}}_p \\ \ddot{\boldsymbol{g}}_d \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{mm} & \boldsymbol{K}_{mp} & -\boldsymbol{A}_m^T \\ \boldsymbol{K}_{pm} & \boldsymbol{K}_{pp} & \boldsymbol{0} \\ -\boldsymbol{A}_m & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_m \\ \boldsymbol{u}_p \\ \boldsymbol{g}_d \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_m \\ \boldsymbol{f}_p \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_p \\ -\boldsymbol{u}_d \end{bmatrix} \tag{2.83}$$

Here $\boldsymbol{A}_m$ is the Boolean matrix localizing the DoF $\boldsymbol{u}_d$ in $\boldsymbol{u}_m$. As a consequence of the partitioning of DoF, the Mixed Craig-Bampton reduction basis in general consists of three ingredients, namely:

- Static constraint modes $\boldsymbol{\Psi}_c$ associated to the interface DoF $\boldsymbol{u}_p$ that are fixed.
- Attachment modes $\boldsymbol{\Psi}_a$ associated to the interface DoF $\boldsymbol{u}_d$ that will be left free.[8]
- A truncated set of fixed/free vibration modes $\boldsymbol{\Phi}_m$ of the structure.

Hence the following approximation is obtained:

$$\boldsymbol{u}_m \approx \boldsymbol{\Phi}_m \boldsymbol{\eta}_m + \boldsymbol{\Psi}_r \boldsymbol{g}_d + \boldsymbol{\Psi}_c \boldsymbol{u}_p \tag{2.84}$$

---

[8] Originally, the MCB method was derived with *residual* attachment modes [217, 219], leading to coupling terms between the modal and interface DoF in both the mass matrix (like in the CB method) and the stiffness matrix (like the DCB method). However, a formulation with attachment modes only gives these coupling terms in the mass matrix and is hence preferred here. Note that coupling terms between both sets of interface DoF ($\boldsymbol{u}_p$ and $\boldsymbol{g}_d$) cannot be avoided.

The ingredients can be computed according to the procedures outlined in the preceding sections but will be briefly reiterated here. Firstly, the mixed vibration modes result from solving the fixed/free eigenproblem with the DoF in $\boldsymbol{u}_m$ free and the $\boldsymbol{u}_p$ fixed, so:

$$\left(\boldsymbol{K}_{mm} - \omega^2 \boldsymbol{M}_{mm}\right) \boldsymbol{\Phi}_m = \boldsymbol{0} \tag{2.85}$$

Note that in case the fixed DoF in $\boldsymbol{u}_p$ do not fully constrain the system, $\boldsymbol{\Phi}_m$ also contains the remaining rigid body modes. Secondly, the constraint modes can be computed by condensing the stiffness matrix to the "primal" DoF, as:

$$\boldsymbol{\Psi}_c = -\boldsymbol{K}_{mm}^{-1} \boldsymbol{K}_{mp} \tag{2.86}$$

Note again that in case a set of primal interface DoF is chosen that does not constrain possible rigid body modes of the substructure, the inverse becomes a pseudo-inverse $^+$. Finally, the attachment modes can be found from:

$$\boldsymbol{\Psi}_a = \boldsymbol{K}_{mm}^{-1} \boldsymbol{A}_m^T \tag{2.87}$$

Next, the Mixed Craig-Bampton reduction matrix can be put in matrix form as:

$$\begin{bmatrix} \boldsymbol{u}_m \\ \boldsymbol{u}_p \\ \boldsymbol{g}_d \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi}_m & \boldsymbol{\Psi}_c & \boldsymbol{\Psi}_a \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_m \\ \boldsymbol{u}_p \\ \boldsymbol{g}_d \end{bmatrix} = \boldsymbol{R}_{MCB} \boldsymbol{q}_{MCB} \tag{2.88}$$

As usual, the reduced equations of motion can be computed through substitution and projection of the reduction basis in the partitioned equations of motion of eq. (2.83). This gives:

$$\begin{gathered} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{M}_{\phi m} & \boldsymbol{M}_{\phi \psi} \\ \boldsymbol{M}_{m\phi} & \tilde{\boldsymbol{M}}_{pp} & \boldsymbol{M}_{m\psi} \\ \boldsymbol{M}_{\psi\phi} & \boldsymbol{M}_{\psi m} & \boldsymbol{M}_{f,dd} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_m \\ \ddot{\boldsymbol{u}}_p \\ \ddot{\boldsymbol{g}}_d \end{bmatrix} + \ldots \\ \begin{bmatrix} \boldsymbol{\Omega}_m^2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}_{pp} & -\boldsymbol{\Psi}_{c|m}^T \\ \boldsymbol{0} & -\boldsymbol{\Psi}_{c|m} & -\boldsymbol{G}_{f,dd} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_m \\ \boldsymbol{u}_p \\ \boldsymbol{g}_d \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{f}}_m \\ \tilde{\boldsymbol{f}}_p \\ \tilde{\boldsymbol{f}}_d \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_p \\ -\boldsymbol{u}_d \end{bmatrix} \end{gathered} \tag{2.89}$$

Here the terms in the reduced matrices are defined as:

$$\begin{aligned} \boldsymbol{M}_{\phi m} &= \boldsymbol{M}_{m\phi}^T = \boldsymbol{\Phi}_m^T \left(\boldsymbol{M}_{mp} - \boldsymbol{M}_{mm} \boldsymbol{K}_{mm}^{-1} \boldsymbol{K}_{mp}\right) & \boldsymbol{G}_{f,dd} &= \boldsymbol{A}_m \boldsymbol{K}_{mm}^{-1} \boldsymbol{A}_m^T \\ \boldsymbol{M}_{\phi\psi} &= \boldsymbol{M}_{\psi\phi}^T = \boldsymbol{\Phi}_m^T \left(\boldsymbol{M}_{mp} - \boldsymbol{M}_{mm} \boldsymbol{K}_{mm}^{-1} \boldsymbol{K}_{mp}\right) & \boldsymbol{\Psi}_{c|m} &= \boldsymbol{A}_m \boldsymbol{\Psi}_c \\ \boldsymbol{M}_{\psi m} &= \boldsymbol{M}_{m\psi}^T = \boldsymbol{\Psi}_a^T \left(\boldsymbol{M}_{mp} - \boldsymbol{M}_{mm} \boldsymbol{K}_{mm}^{-1} \boldsymbol{K}_{mp}\right) & \tilde{\boldsymbol{f}}_m &= \boldsymbol{\Phi}_m^T \boldsymbol{f}_m \\ \tilde{\boldsymbol{M}}_{pp} &= \boldsymbol{M}_{pp} + \boldsymbol{\Psi}_c^T \boldsymbol{M}_{mm} \boldsymbol{\Psi}_c - \boldsymbol{M}_{pm} \boldsymbol{\Psi}_c - \boldsymbol{\Psi}_c^T \boldsymbol{M}_{mp} & \tilde{\boldsymbol{f}}_p &= \boldsymbol{\Psi}_c^T \boldsymbol{f}_m + \boldsymbol{f}_p \\ \boldsymbol{M}_{f,dd} &= \boldsymbol{\Psi}_a^T \boldsymbol{M}_{mm} \boldsymbol{\Psi}_a = \boldsymbol{A}_m \boldsymbol{K}_{mm}^{-1} \boldsymbol{M} \boldsymbol{K}_{mm}^{-1} \boldsymbol{A}_m^T & \tilde{\boldsymbol{f}}_d &= \boldsymbol{\Psi}_a^T \boldsymbol{f}_m \\ \tilde{\boldsymbol{K}}_{pp} &= \boldsymbol{K}_{pp} - \boldsymbol{K}_{pm} \boldsymbol{K}_{mm}^{-1} \boldsymbol{K}_{mp} \end{aligned} \tag{2.90}$$

and $\boldsymbol{\Omega}_m^2$ is again a diagonal matrix containing the squares of the fixed/free eigenfrequencies of the component. Assembly of systems reduced using the Mixed Craig-Bampton method requires a special assembly procedure, which is developed in section 3.5.

From the above equation one can clearly see that the Mixed Craig-Bampton method is a true generalization of the original Craig-Bampton and Dual Craig-Bampton methods; if there are no “dual” DoF (i.e. $\boldsymbol{u}_d$ is empty) the reduced matrices are exactly equal to those found with the Craig-Bampton method whereas in the absence of “primal” DoF (i.e. $\boldsymbol{u}_p$ is empty) the matrices of the Dual Craig-Bampton method, formulated with attachment modes, are found. Practically, one can implement this by taking a high value for the parameter c in the selection scheme. The scheme will then consider the stiffness of all connected pairs of interface DoF to be of the same magnitude and hence an equal boundary condition will be chosen on both sides.

Note that the reduction basis in (2.88) was already proposed in [200], where a mixed assembly was also proposed but was not general and limited to decompositions where no more then two substructures connect on an interface. Also this basis was later proposed in [131], but there, very much like in the MacNeal and Rubin method, the authors eliminated the interface forces from the basis, leading to full and intricate reduced matrices.

2.7 Modal Truncation Augmentation

The total models obtained from assembly of the reduced component models are used for structure dynamic analysis. Amongst others, this can be modal analysis of the system or, as often encountered in wind turbine engineering, a forced response time simulation. Since modal superposition is used to obtain the reduced component models, the component model reduction methods treated in this chapter are particularly suited to efficiently approximate the total system’s modal properties. However, for the same reason the models are less effective in the case of forced response analysis. This is due to the difference between spectral and spatial convergence, as explained next.

In the case of forced response analysis, the approximation made by the component reduction basis is generally accurate as long as the modes discarded during reduction are not excited by the external loading. This is governed by two aspects, namely:

- *Spectral convergence.* For accurate response predictions the frequency content of the reduced models should be higher than the highest frequency present in the external loading. An often used rule of thumb is to include all substructure modes up to 1.5 times the highest frequency of interest; in chapter 5 a more systematic approach will be taken based on error estimation theory.

- *Spatial convergence.* In order to accurately capture the response, the modes should also accurately represent the (response to) the spatial distribution of the external loading. In other words, the external load should be orthogonal to the discarded modes. In contrast to convergence of the spectral type, spatial convergence is often overlooked. One way to improve the spatial convergence is by retaining more modes in the reduction basis, but the convergence rate is dependent on the correlation between the mode shapes and spatial load vectors. Especially for localized loading this rate can be very poor.

Several methods have been devised to improve the forced response of reduced models. One of these methods is the classic *mode acceleration* method which a posteriori corrects the solution by adding the static response of the discarded modes [157, 74, 67]. Another method, which is an extension of the mode acceleration method, is called the *modal truncation augmentation* (MTA) method [57, 163] and can be used to compute load dependent vectors (or pseudo-modes) which capture the spatial part of the force response not captured by the retained modes. These pseudo-modes can be used to a priori augment the component reduction bases, leading to improved response predictions without the need to postprocess the solution.

In the remainder of this section the MTA method is first explained in the general case. Thereafter some specific details are addressed when augmentation is applied to both fixed and free interface vibration modes, in subsections 2.7.2 and 2.7.3 respectively. Subsection 2.7.4 briefly discusses some practical details.

2.7.1 General Concept

In order to demonstrate the general concept of the MTA vectors, a single system without interfaces is considered that is assumed to be constrained such that no rigid body modes exist. Its equations of motion are simply:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} \tag{2.91}$$

Here the time dependence of $\boldsymbol{u}$ and $\boldsymbol{f}$ is omitted for compactness. Similar to equation (2.4), the response $\boldsymbol{u}$ can be split into a static and dynamic part:

$$\boldsymbol{u} = \boldsymbol{u}_{\text{stat}} + \boldsymbol{u}_{\text{dyn}} \tag{2.92}$$

The static response is obtained simply by assuming zero accelerations, such that $\boldsymbol{u}_{\text{stat}} = \boldsymbol{K}^{-1}\boldsymbol{f}$. Equation (2.92) can then be substituted into (2.91) and rewritten for the dynamic part of the response, this gives:

$$\boldsymbol{M}\ddot{\boldsymbol{u}}_{\text{dyn}} + \boldsymbol{K}\boldsymbol{u}_{\text{dyn}} = -\boldsymbol{M}\boldsymbol{K}^{-1}\ddot{\boldsymbol{f}} \tag{2.93}$$

This equation can now be treated similarly to (2.91), by separating again the solution for $\boldsymbol{u}_{\text{dyn}}$ in a quasi-static solution and a dynamic solution relative to this quasi-static part:

$$\boldsymbol{u}_{\text{dyn}} = \boldsymbol{y}_{\text{stat}} + \boldsymbol{y}_{\text{dyn}} \tag{2.94}$$

The quasi-static solution $\boldsymbol{y}_{\text{stat}}$ can be found by setting $\ddot{\boldsymbol{u}}_{\text{dyn}}$ in (2.93) to zero:

$$\boldsymbol{y}_{\text{stat}} = \boldsymbol{K}^{-1}\left(-\boldsymbol{M}\boldsymbol{K}^{-1}\right)\ddot{\boldsymbol{f}} \tag{2.95}$$

If equation (2.94) is substituted in (2.93) and rewritten for the dynamic part $\boldsymbol{y}_{\text{dyn}}$ the following is obtained:

$$\boldsymbol{M}\ddot{\boldsymbol{y}}_{\text{dyn}} + \boldsymbol{K}\boldsymbol{y}_{\text{dyn}} = \left(-\boldsymbol{M}\boldsymbol{K}^{-1}\right)^{2}\frac{d^{4}\boldsymbol{f}}{dt^{4}} \tag{2.96}$$

Once again, the obtained result can be treated as (2.93) by representing the solution for $\boldsymbol{y}_{\text{dyn}}$ by a quasi-static and a dynamic solution. Repeating this procedure gives rise to the following sequence:

$$\begin{aligned}\boldsymbol{u} &= \boldsymbol{u}_{\text{stat}} + \boldsymbol{u}_{\text{dyn}}\\ &= \boldsymbol{u}_{\text{stat}} + \boldsymbol{y}_{\text{stat}} + \boldsymbol{y}_{\text{dyn}}\\ &= \boldsymbol{u}_{\text{stat}} + \boldsymbol{y}_{\text{stat}} + \boldsymbol{z}_{\text{stat}} + \ldots + \boldsymbol{w}_{\text{dyn}}\end{aligned} \tag{2.97}$$

When the results obtained for the quasi-static solutions are substituted in (2.97), the following expression is therefore obtained for the response:

$$\boldsymbol{u} = \sum_{j=1}^{k} \boldsymbol{K}^{-1}(-\boldsymbol{M}\boldsymbol{K}^{-1})^{j-1} \frac{d^{2(j-1)}\boldsymbol{f}}{dt^{2(j-1)}} + \boldsymbol{w}_{\text{dyn}} \tag{2.98}$$

This expression describes the system's response as a sequence of quasi-static solutions up to order k. The remaining dynamic solution $\boldsymbol{w}_{\text{dyn}}$ is approximated by the usual truncated set of retained (indicated by $_r$) vibration modes $\boldsymbol{\Phi}_r$.

In order to use eq. (2.98) to compute basis vectors that improve the spatial convergence of the modal superposition, the external loading $\boldsymbol{f}(t)$ can be decomposed in a spatial and spectral (time dependent) part:

$$\boldsymbol{f}(t) = \sum_{j=1}^{p} \boldsymbol{f}_j \alpha_j(t) = \boldsymbol{F}\boldsymbol{\alpha}(t) \tag{2.99}$$

Here $\boldsymbol{f}_j$ is the j^{th} spatial force vector which is modulated by its corresponding time function $\alpha_j(t)$. It is assumed that the external force can be represented by a limited number p of such spatial force distributions, which are collected in $\boldsymbol{F}$ and $\boldsymbol{\alpha}(t)$ contains their corresponding time functions. These load distribution vectors can represent all types of forces such as point forces, surface forces (e.g. wind or wave loads) and body forces (e.g. gravity). Since the aim is to improve the spatial convergence, only the force distributions in $\boldsymbol{F}$ are taken into account and the time dependent part is discarded.

Note that this procedure is only applied for the sake of computing the MTA vectors; the actual response analysis is carried out with the original loading $\boldsymbol{f}(t)$. Hence, in practice one would preferably obtain these spatial force distributions from a limited yet representative set of force data, especially if a large number of load cases need to be evaluated. One way to achieve this is through the use of the powerful *proper orthogonal decomposition* (POD), this method is explained in appendix D.

Using the force distribution vectors, the first term on the right hand side of (2.98) the *modal truncation augmentation* vectors can now be calculated as:

$$\tilde{\boldsymbol{\Phi}}_{M,j} = \boldsymbol{K}^{-1}(\boldsymbol{M}\boldsymbol{K}^{-1})^{j-1}\boldsymbol{F} \quad \text{for} \quad j = 1 \ldots k \tag{2.100}$$

Here $\tilde{\boldsymbol{\Phi}}_{M,j}$ is a set of MTA vectors (or MTAs in short) of order j; the number of vectors depends on the number of force distributions p used in $\boldsymbol{F}$. From the above expression it is clear why MTAs are often referred to as load dependent vectors. Also, as pointed out in [167], a clear link can be observed to moment-matching techniques used for model reduction in control and electrical engineering [70, 127].

Physically, the MTAs can be interpreted as follows. The first order MTAs ($j = 1$) simply correspond to the static response of the structure to the external loading in $\boldsymbol{F}$. For the second order vectors ($j = 2$) the static displacements due to these forces are converted into inertia forces via the mass matrix, which in turn are converted by the inverse of the stiffness matrix to quasi-static displacements resulting from these inertia forces. Hence they provide a load dependent quasi-static correction to the solution. Higher order of MTA vectors provide increasingly higher order load dependent correction modes. Furthermore, it can be seen from the above expression that the MTA vectors are in fact forming a so-called *Krylov sequence*. In section 2.7.4 it will be explained how this can be exploited in order to efficiently compute the MTAs.

Both the retained vibration modes and the MTA vectors are used to capture the response of the system to the external loading. Similar to the residual attachment modes in section 2.3.3, this can lead to an "information overlap", meaning that the vibration modes and MTAs partly span the same space. This not only causes a non-sparse reduced model, but could in extreme cases lead to a linearly dependent reduction basis which jeopardizes numerical stability. To overcome this issue the contribution of the vibration modes should be subtracted. Using the spectral expansion of the inverse stiffness matrix, see eq. (2.28), this is achieved by:

$$\begin{aligned}\tilde{\boldsymbol{\Phi}}_{M,j} &= \left(\boldsymbol{K}^{-1} - \boldsymbol{\Phi}_r\boldsymbol{\Omega}_r^{-2}\boldsymbol{\Phi}_r^T\right)\left(\boldsymbol{M}\boldsymbol{K}^{-1}\right)^j\boldsymbol{F}\\ &= \left(\boldsymbol{I} - \boldsymbol{\Phi}_r\boldsymbol{\Phi}_r^T\boldsymbol{M}\right)\left(\boldsymbol{K}^{-1}\boldsymbol{M}\right)^j\boldsymbol{K}^{-1}\boldsymbol{F}\\ &= \boldsymbol{P}\left(\boldsymbol{K}^{-1}\boldsymbol{M}\right)^j\boldsymbol{K}^{-1}\boldsymbol{F}\end{aligned}$$

Here $\boldsymbol{P}$ is an orthogonal projector that is very similar to the one used in eq. (2.23); the resulting MTAs are both $\boldsymbol{K}$- and $\boldsymbol{M}$-orthogonalized with respect to the retained vibration modes. Next, the different orders of MTAs can be collected as follows:

$$\tilde{\boldsymbol{\Phi}}_M = \left[\tilde{\boldsymbol{\Phi}}_{M,1} \ \dots \ \tilde{\boldsymbol{\Phi}}_{M,j} \ \dots \ \tilde{\boldsymbol{\Phi}}_{M,k}\right] \tag{2.101}$$

The value of k specifies the highest order of the MTAs and hence the total number of MTA vectors. These correction vectors are orthogonal to the free vibration modes, but not mutually orthogonal. To preserve the sparsity of the reduced matrices and improve numerical robustness the MTA vectors can be orthonormalized by solving a reduced eigenvalue problem in the space of the MTAs. This is known as the *interaction problem* (for further details see section 6.3.1):

$$\left(\tilde{\boldsymbol{\Phi}}_M^T\boldsymbol{K}\tilde{\boldsymbol{\Phi}}_M\right)\boldsymbol{y} = \boldsymbol{\sigma}^2\left(\tilde{\boldsymbol{\Phi}}_M^T\boldsymbol{M}\tilde{\boldsymbol{\Phi}}_M\right)\boldsymbol{y} \tag{2.102}$$

Here $\boldsymbol{\sigma}^2$ is a diagonal matrix containing the pseudo-frequencies belonging to the MTA vectors, while $\boldsymbol{y}$ are the eigenvectors. The orthonomalized MTAs are found from expansion of the reduced eigenmodes as:

$$\boldsymbol{\Phi}_M = \tilde{\boldsymbol{\Phi}}_M\boldsymbol{y} \tag{2.103}$$

Finally, the MTAs can be mass normalized such that:

$$\begin{aligned}\boldsymbol{\Phi}_M^T\boldsymbol{M}\boldsymbol{\Phi}_M &= \boldsymbol{I}\\ \boldsymbol{\Phi}_M^T\boldsymbol{K}\boldsymbol{\Phi}_M &= \boldsymbol{\sigma}^2\end{aligned} \tag{2.104}$$

The MTA vectors are often referred to as "pseudo-modes", since they share the orthogonality properties of the vibration modes but are not fundamental properties of the system, that is, they are not solutions to the system's eigenproblem. Note that the MTA pseudo-frequencies in $\boldsymbol{\sigma}^2$ are always higher than those of the retained vibration modes $\boldsymbol{\Omega}_r^2$. This is because, by construction, the MTAs take into account the relevant contributions of the discarded vibration modes. Hence the addition of these vectors to the reduction basis not only improves the quasi-static results but also provides a dynamic correction. An indication up to what frequency this correction is provided is given by the pseudo-frequencies.

2.7.2 Augmentation of Fixed Interface Vibration Modes

Whereas in the previous subsection the derivation of MTAs for the general case was considered, this subsection treats some specific details when MTAs are used in conjunction with fixed interface vibration modes to reduce a component model. Indeed, the MTAs can then be used to enrich the Craig-Bampton reduction basis.

The starting point for the derivation of this type of MTAs are partitioned component equations of motion in eq. (2.43). Taking the first equation and separating the terms related to the internal and boundary DoF gives:

$$\boldsymbol{M}_{ii}\ddot{\boldsymbol{u}}_i + \boldsymbol{K}_{ii}\boldsymbol{u}_i = -\boldsymbol{M}_{ib}\ddot{\boldsymbol{u}}_b - \boldsymbol{K}_{ib}\boldsymbol{u}_b + \boldsymbol{f}_i \tag{2.105}$$

Similar to the derivation of the MTAs for the general case, the first step is to split the response of the internal DoF in a static and dynamic part:

$$\boldsymbol{u}_i = \boldsymbol{u}_{i,\text{stat}} + \boldsymbol{u}_{i,\text{dyn}} \tag{2.106}$$

In a similar fashion as in the previous section, the static response is now found from (2.105) by neglecting the inertia forces, i.e.:

$$\boldsymbol{u}_{i,\text{stat}} = \boldsymbol{K}_{ii}^{-1}\boldsymbol{f}_i - \boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}\boldsymbol{u}_b = \boldsymbol{K}_{ii}^{-1}\boldsymbol{f}_i + \boldsymbol{\Psi}_c\boldsymbol{u}_b \tag{2.107}$$

In this expression the second term on the right hand side can be recognized as the static constraint modes derived in section 2.3.1. In the derivation of the Craig-Bampton method, and in fact all the classic component reduction methods treated in this chapter, the first term on the right hand side is assumed to be zero. That is, no excitation is assumed on the internal DoF. The MTA vectors specifically take this term into account to improve the spatial convergence of the reduced model. Next, the above expression is substituted in (2.106) and subsequently into the equations of motion (2.105). Rewriting for the dynamic solution $\boldsymbol{u}_{i,\text{dyn}}$ then results in the following:

$$\boldsymbol{M}_{ii}\ddot{\boldsymbol{u}}_{i,\text{dyn}} + \boldsymbol{K}_{ii}\boldsymbol{u}_{i,\text{dyn}} = -\left(\boldsymbol{M}_{ii}\boldsymbol{\Psi}_c + \boldsymbol{M}_{ib}\right)\ddot{\boldsymbol{u}}_b - \boldsymbol{M}_{ii}\boldsymbol{K}_{ii}^{-1}\ddot{\boldsymbol{f}}_i \tag{2.108}$$

The columns of the bracketed term on the right hand side can be interpreted as load vectors associated to the interface accelerations. For ease of notation it is now defined that:

$$\boldsymbol{Y}_i = \boldsymbol{K}_{ib} - \boldsymbol{K}_{ii}\boldsymbol{M}_{ii}^{-1}\boldsymbol{M}_{ib}, \tag{2.109}$$

which allows to write the previous equation as:

$$\boldsymbol{M}_{ii}\ddot{\boldsymbol{u}}_{i,\text{dyn}} + \boldsymbol{K}_{ii}\boldsymbol{u}_{i,\text{dyn}} = -\boldsymbol{M}_{ii}\boldsymbol{K}_{ii}^{-1}\left(\ddot{\boldsymbol{f}}_i - \boldsymbol{Y}_i\ddot{\boldsymbol{u}}_b\right) \tag{2.110}$$

Next, the remaining dynamic solution $\boldsymbol{u}_{i,\text{dyn}}$ is again split into a quasi-static solution $\boldsymbol{y}_{i,\text{stat}}$ and a relative dynamic solution $\boldsymbol{y}_{i,\text{dyn}}$. By subsequently repeating this process a sequence is obtained similar to eq. (2.97), now for the response of the internal DoF:

$$\boldsymbol{u}_i = \boldsymbol{u}_{i,\text{stat}} + \boldsymbol{y}_{i,\text{stat}} + \boldsymbol{z}_{i,\text{stat}} + \ldots + \boldsymbol{w}_{i,\text{dyn}} \tag{2.111}$$

After substitution of the results for the (quasi-)static solutions the following is obtained:

$$\boldsymbol{u}_i = \boldsymbol{\Psi}_c\boldsymbol{u}_b + \boldsymbol{K}_{ii}^{-1}\boldsymbol{f}_i - \boldsymbol{K}_{ii}^{-1}\boldsymbol{M}_{ii}\boldsymbol{K}_{ii}^{-1}\left(\ddot{\boldsymbol{f}}_i - \boldsymbol{Y}_i\ddot{\boldsymbol{u}}_b\right) + \left(\boldsymbol{K}_{ii}^{-1}\boldsymbol{M}_{ii}\right)^2\boldsymbol{K}_{ii}^{-1}\frac{d^4}{dt^4}\left(\boldsymbol{f}_i - \boldsymbol{Y}_i\boldsymbol{u}_b\right) + \cdots + \boldsymbol{w}_{i,dyn}$$

(2.112)

In the Craig-Bampton method the first term on the right hand side, the static constraint modes, are already included in the reduction basis. The dynamic solution $\boldsymbol{w}_{i,\text{dyn}}$ is approximated by a superposition of the fixed-interface vibration modes $\boldsymbol{\Phi}_i$. The remaining terms in the expression can be sorted into a group involving the external force on the internal DoF and another involving the interface excitation. Similar to the previous section it is assumed that the spatial content of the external loading $\boldsymbol{f}_i$ can be described by the force distributions in $\boldsymbol{F}_i$; the time dependent part is again discarded. This allows to compute each order of the fixed interface MTAs as:

$$\tilde{\boldsymbol{\Phi}}_{Mi,j} = \begin{cases} \boldsymbol{P}_i \boldsymbol{K}_{ii}^{-1} \boldsymbol{F}_i & \text{for} \quad j = 1 \\ \boldsymbol{P}_i (\boldsymbol{K}_{ii}^{-1} \boldsymbol{M}_{ii})^{j-1} \boldsymbol{K}_{ii}^{-1} [\boldsymbol{F}_i \quad \boldsymbol{Y}_i] & \text{for} \quad j = 2 \dots k \end{cases} \tag{2.113}$$

In this expression the projection step for orthogonalization with the fixed interface modes is already incorporated. The projection matrix in this case writes:

$$\boldsymbol{P}_i = \boldsymbol{I} - \boldsymbol{\Phi}_i \boldsymbol{\Phi}_i^T \boldsymbol{M}_{ii} \tag{2.114}$$

It is important to note that the number of MTA vectors per order now equals the number of interface DoF plus the number of load vectors. Hence, the so-obtained MTAs improve the spatial convergence in response to both external loading and excitation from neighboring components through the interface DoF. Next, the MTAs up to the desired order k are collected in the matrix $\boldsymbol{\Phi}_{Mi}$, and are subsequently $\boldsymbol{M}_{ii}$– and $\boldsymbol{K}_{ii}$-orthogonalized by solving the interaction problem similar to eq. (2.102). Mass normalization finally gives the same properties as in eq. (2.104).

Including the MTA vectors, the response of the internal DoF can now be represented by:

$$\boldsymbol{u}_i \approx \boldsymbol{\Psi}_c \boldsymbol{u}_b + \boldsymbol{\Phi}_i \boldsymbol{\eta}_i + \boldsymbol{\Phi}_{Mi} \boldsymbol{\zeta}_i \tag{2.115}$$

Here $\boldsymbol{\zeta}$ are the modal amplitudes of the MTAs. The above approximation gives rise to the augmented Craig-Bampton reduction basis as:

$$\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi}_i & \boldsymbol{\Phi}_{Mi} & \boldsymbol{\Psi}_c \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{\zeta}_i \\ \boldsymbol{u}_b \end{bmatrix} = \boldsymbol{R}_{ACB} \boldsymbol{q}_{ACB} \tag{2.116}$$

Substitution and projection using this basis gives the reduced equations of motion:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{M}_{\phi b} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{M}_{\zeta b} \\ \boldsymbol{M}_{b\phi} & \boldsymbol{M}_{b\zeta} & \tilde{\boldsymbol{M}}_{bb} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_i \\ \ddot{\boldsymbol{\zeta}}_i \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Omega}_i^2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\sigma}^2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \tilde{\boldsymbol{K}}_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{\zeta}_i \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{f}}_i \\ \tilde{\boldsymbol{f}}_M \\ \tilde{\boldsymbol{f}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{g}_b \end{bmatrix} \tag{2.117}$$

In addition to the terms in eq. (2.57), the terms in the reduced matrices are defined as:

$$\begin{aligned} \boldsymbol{M}_{\zeta b} &= \boldsymbol{M}_{b\zeta}^T = \boldsymbol{\Phi}_{Mi}^T \left(\boldsymbol{M}_{ib} - \boldsymbol{M}_{ii} \boldsymbol{K}_{ii}^{-1} \boldsymbol{K}_{ib} \right) \\ \tilde{\boldsymbol{f}}_M &= \boldsymbol{\Phi}_{Mi}^T \boldsymbol{f}_i \end{aligned} \tag{2.118}$$

By comparing equations (2.56) and (2.117), it can be observed that the properties of the Craig-Bampton reduced matrices remain the same after augmentation with MTA vectors. This is due to the "pseudo-mode" character of the MTAs. Since all interface DoF are retained, the augmented Craig-Bampton reduced model can be assembled as a regular superelement. Due to the addition of MTA vectors to the basis the augmented superelement has much better spatial convergence properties, enabling more accurate predictions of the total system. Since the MTAs improve the response to both external loading and loading from adjacent components through the interface, the augmented reduced model will not only provide a more accurate forced response but will also improve the modal analysis results of the full system.

2.7.3 Augmentation of Free Interface Vibration Modes

In this subsection MTAs are derived that can be used in a component reduction basis in combination with free interface vibration modes. The starting point is again the unreduced equations of motion of a substructure excited by an external force.

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} + \boldsymbol{A}^T\boldsymbol{g}_b \tag{2.119}$$

As before, $\boldsymbol{g}_b$ are the connection forces at the interface from neighboring substructures, $\boldsymbol{A}$ the local Boolean matrix and $\boldsymbol{f}$ is an externally applied force. Since the component model can be free-floating, these forces need to be split into a self-equilibrating part and a remainder, according to the procedure outlined for the attachment modes in section 2.3.2. This is done using the projection matrix of eq. (2.23) which contains the mass normalized rigid body modes $\boldsymbol{\Phi}_r$, giving:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{P}_r\left(\boldsymbol{f} + \boldsymbol{A}^T\boldsymbol{g}_b\right) + \left(\boldsymbol{I} - \boldsymbol{P}_r\right)\left(\boldsymbol{f} + \boldsymbol{A}^T\boldsymbol{g}_b\right) \tag{2.120}$$

Since the aim is to augment the free vibration modes, here no partitioning of the DoF vector $\boldsymbol{u}$ is applied. One directly seeks a solution for $\boldsymbol{u}$ where, as before, the response is split into a quasi-static part and a dynamic part. Solving the static problem results in:

$$\boldsymbol{u} = \boldsymbol{u}_{\text{stat}} + \boldsymbol{u}_{\text{dyn}} = \boldsymbol{G}_f\left(\boldsymbol{f} + \boldsymbol{A}^T\boldsymbol{g}_b\right) + \boldsymbol{u}_{\text{dyn}} = \boldsymbol{G}_f\boldsymbol{f} + \boldsymbol{\Psi}_a\boldsymbol{g}_b + \boldsymbol{u}_{\text{dyn}} \tag{2.121}$$

Here $\boldsymbol{G}_f$ is the elastic flexibility matrix from section 2.3.2 and $\boldsymbol{\Psi}_a$ are the attachment modes derived therein. Remark again that the first term on the right hand side is normally neglected in the derivation of free interface component reduction methods. As usual, the above expression is substituted into (2.120) and rewritten for the dynamic solution. This dynamic solution can in turn be split in a quasi-static and relative dynamic solution. Repeating this process like in the previous sections and making use of the orthogonality between $\boldsymbol{G}_f$ and $\boldsymbol{\Phi}_r$, gives rise to the following sequence:

$$\boldsymbol{u} = \boldsymbol{\Psi}_a\boldsymbol{g}_b + \boldsymbol{G}_f\boldsymbol{f} - \left(\boldsymbol{G}_f\boldsymbol{M}\right)\boldsymbol{G}_f\left(\ddot{\boldsymbol{f}} + \boldsymbol{A}^T\ddot{\boldsymbol{g}}_b\right) + \left(\boldsymbol{G}_f\boldsymbol{M}\right)^2\boldsymbol{G}_f\frac{d^4}{dt^4}\left(\boldsymbol{f} + \boldsymbol{A}^T\boldsymbol{g}_b\right) + \ldots + \boldsymbol{w}_{\text{dyn}} \tag{2.122}$$

In this expression $\boldsymbol{w}_{\text{dyn}}$ is again the remaining dynamic solution which is approximated by the rigid body modes $\boldsymbol{\Phi}_r$ and a truncated set of free vibration modes $\boldsymbol{\Phi}_f$. Similar to the previous section, the attachment modes in first term (or their residual attachment siblings) are usually already present in the component reduction basis. Assuming again that the external force can be represented by p spatial distribution vectors and discarding the time dependency, the free interface MTAs can thus be computed as:

$$\tilde{\boldsymbol{\Phi}}_{M,j} = \begin{cases} \boldsymbol{P}_f \boldsymbol{G}_f \boldsymbol{F} & \text{for} \quad j = 1 \\ \boldsymbol{P}_f \left(-\boldsymbol{G}_f \boldsymbol{M}\right)^{j-1} \boldsymbol{G}_f \begin{bmatrix} \boldsymbol{F} & \boldsymbol{A}^T \end{bmatrix} & \text{for} \quad j = 2 \dots k \end{cases} \tag{2.123}$$

By construction the MTAs are orthogonal to the rigid body modes; in the above expression they are also orthogonalized with respect to the free vibration modes using the project matrix:

$$\boldsymbol{P}_f = \boldsymbol{I} - \boldsymbol{\Phi}_f \boldsymbol{\Phi}_f^T \boldsymbol{M} \tag{2.124}$$

Hereafter the vectors are again mutually orthogonalized with respect to $\boldsymbol{M}$ and $\boldsymbol{K}$ and mass normalized, the resulting properties are given in (2.104). Assuming that the MTAs are used in a reduction basis that employs residual attachment modes, one can finally choose to further orthogonalize the mode sets $\boldsymbol{\Phi}_M$ and $\boldsymbol{\Psi}_r$ since both contribute to the static solution. One option is to modify $\boldsymbol{\Psi}_r$ by subtracting the component already present in the MTAs, i.e:

$$\boldsymbol{\Psi}_r \leftarrow \left(\boldsymbol{\Psi}_r - \boldsymbol{\Phi}_M \boldsymbol{\sigma}^{-2} \boldsymbol{\Phi}_M^T \boldsymbol{A}^T\right) \tag{2.125}$$

Note that this is similar to the $\boldsymbol{K}$-orthogonalization of the attachment modes to obtain the residual attachment modes in section 2.3.3. However, since the MTAs are only pseudo-modes, they will not be $\boldsymbol{M}$-orthogonal to the residual attachment modes (see [166]). Another option is to make the residual modes $\boldsymbol{M}$-orthogonal to the MTAs as follows:

$$\boldsymbol{\Psi}_r \leftarrow \left(\boldsymbol{I} - \boldsymbol{\Phi}_M \boldsymbol{\Phi}_M^T \boldsymbol{M}\right) \boldsymbol{\Psi}_r \tag{2.126}$$

When using the above procedure the MTAs and residual attachment modes are not $\boldsymbol{K}$-orthogonal. Either option can be used as the choice does not influence the solution obtained by the reduced system.

Using the MTAs to augment the free interface mode reduction basis, the system's response can now be approximated by:

$$\boldsymbol{u} \approx \boldsymbol{\Phi} \boldsymbol{\eta} + \boldsymbol{\Phi}_M \boldsymbol{\zeta} - \boldsymbol{\Psi}_r \boldsymbol{g}_b \tag{2.127}$$

For compactness, the rigid body modes and free interface vibration modes are here collected in one matrix, i.e. $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_r & \boldsymbol{\Phi}_f \end{bmatrix}$. Indeed, this reduction basis is used in both the Rubin and Dual Craig-Bampton methods. Next it will be briefly shown how the augmentation affects the reduced matrices obtained from these methods.

As was discussed in section 2.5.3, the Rubin reduction matrix is obtained as the product of the reduction basis in (2.127) and a subsequent transformation to backsubstitute the boundary displacement DoF $\boldsymbol{u}_b$. Including the MTAs, the augmented Rubin basis thus becomes:

$$\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi}_{|i} - \boldsymbol{\Psi}_{r|i} \boldsymbol{K}_{r,bb} \boldsymbol{\Phi}_{|b} & \boldsymbol{\Phi}_{M|i} - \boldsymbol{\Psi}_{r|i} \boldsymbol{K}_{r,bb} \boldsymbol{\Phi}_{M|b} & \boldsymbol{\Psi}_{r|i} \boldsymbol{K}_{r,bb} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\zeta} \\ \boldsymbol{u}_b \end{bmatrix} = \boldsymbol{R}_{AR} \boldsymbol{q}_{AR} \tag{2.128}$$

Application of this reduction basis gives reduced equations of motion that are similar to those obtained in eq. (2.68); due to the transformation to interface displacements the matrices are again full. As outlined in section 2.5.4, the difference between the Rubin and Dual Craig-Bampton reduction is that in the latter also the compatibility condition is reduced. To this end, the interface forces are added to the DoF vector and the augmented Dual Craig-Bampton reduction basis becomes:

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{g}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{\Phi}_M & \boldsymbol{\Psi}_r \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\zeta} \\ \boldsymbol{g}_b \end{bmatrix} = \boldsymbol{R}_{ADCB} \boldsymbol{q}_{ADCB} \tag{2.129}$$

Substitution of this basis in the equations of motion in (2.73) and subsequent projection leads to the following reduced system:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{M}_{\phi\psi} \\ \boldsymbol{0} & \boldsymbol{M}_{\psi\phi} & \boldsymbol{M}_{r,bb} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}} \\ \ddot{\boldsymbol{\zeta}} \\ \ddot{\boldsymbol{g}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Omega}^2 & \boldsymbol{0} & -\boldsymbol{\Phi}_{|b}^T \\ \boldsymbol{0} & \boldsymbol{\sigma}^2 & -\boldsymbol{\Phi}_{M|b}^T \\ -\boldsymbol{\Phi}_{|b} & -\boldsymbol{\Phi}_{M|b} & -\boldsymbol{G}_{r,bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\zeta} \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^T \boldsymbol{f} \\ \boldsymbol{\Phi}_M^T \boldsymbol{f} \\ \boldsymbol{\Psi}_r^T \boldsymbol{f} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{u}_b \end{bmatrix} \tag{2.130}$$

Where in addition to the terms previously introduced, it is defined that:

$$\begin{aligned} \boldsymbol{M}_{\phi\psi} &= \boldsymbol{M}_{\psi\phi}^T = \boldsymbol{\Phi}_M^T \boldsymbol{M} \boldsymbol{\Psi}_r \\ \boldsymbol{\Phi}_{M|b} &= \boldsymbol{\Phi}_{M|b}^T = \boldsymbol{A} \boldsymbol{\Phi}_M \end{aligned} \tag{2.131}$$

By comparing the above reduced equations with eq. (2.74) it can be observed that the topology of the Dual Craig-Bampton reduced matrices is retained after augmentation.

2.7.4 Practical Details & Further Remarks

Having derived in the previous subsections MTA vectors for the augmentation of fixed and free interface vibration modes, this subsection briefly addresses a few practical details regarding their use and implementation:

- In section 2.7.1 it was remarked that the different orders of MTAs form a *Krylov sequence*. This sequence is similar to those generated by inverse iteration eigensolvers, a feature can be exploited to efficiently compute the MTAs. In fact, the MTAs can be obtained as a by-product of the popular (block) Lanczos eigensolver used to compute the substructure vibration modes. In order to do so the external force vectors, both from the interface and external loading, need to be chosen as the initial guess for the eigensolver. The MTAs can then easily be obtained by postprocessing the Lanczos iteration vectors. Hence, the MTAs can be obtained at very little additional cost which greatly enhances their practical value. For more details on the Lanczos eigensolver in general see section 6.3.1, the use of this solver to compute MTAs is addressed in [164].
- When MTAs arising from interface excitation are used to augment the reduction basis, the number of augmentation vectors is dependent on the desired order and the number of interface DoF. In case a substructure has many interface DoF this number

can hence grow rapidly, limiting the effectiveness of the model reduction. To overcome this limitation, interface reduction techniques can be applied. Various interface reduction methods exist, both on substructure and system level; some techniques will be outlined in chapter 4. Note that these techniques serve two purpose, namely 1) to reduce the actual set of interface DoF and 2) to obtain a smaller set of MTA vectors to more efficiently augment the reduction basis. The interface reduction methods can be applied to independently achieve either or both of these goals.

- In addition to interface reduction techniques, other ways exist to reduce the number of MTA vectors. One very simple method may be to select the MTAs based on their pseudo-frequencies in $\boldsymbol{\sigma}^2$. Similar to the selection of vibration modes, some cutoff frequency maybe defined (e.g. two or three times the targeted maximum frequency) and all MTAs below that frequency can be included. This means that all MTAs still need to be computed and the most suitable ones are chosen a posteriori. Another, more elaborate, method is to use the effective modal mass associated with the MTAs, see [68] and [142] for details.

- Like the static modes, the MTA vectors can be frequency shifted. This allows the load dependent vectors to be created around a central frequency ω_s, which is beneficial if a clear source of harmonic excitation is known in advance (e.g. an operational rotation frequency). However, this incurs additional computational cost since the dynamic stiffness $(\boldsymbol{K} - \omega_s^2\boldsymbol{M})$ matrix needs to be factorized, unless this shifted matrix was already used to compute the static and/or vibration modes.

- As treated in the previous subsection, the free interface MTAs originating from interface excitation can be used to augment the Dual Craig-Bampton reduction basis. In [166] it was shown that these MTAs help to avoid so-called "spurious peaks" caused by incompatible motions between the substructures.

- In practice one finds that a combination of vibration modes and MTAs in the reduction basis gives the best results (see e.g. [163, 165, 142]). The reason is that the MTAs are well suited to improve the response to interface excitations whereas the vibration modes are important ingredients to represent the internal dynamics. Both aspects are important to accurately represent the dynamics of the total system.

2.8 Summary

This chapter was concerned with component model reduction techniques, which are used to obtain a compact model of a component's structural dynamics. After a general introduction to dynamic substructuring in section 2.1, these techniques were derived in section 2.2 in general terms. This showed that a component reduction basis usually consists of a set of static modes and truncated set of vibration modes. The former account for the interaction with neighboring components while the latter describe the component's dynamic behavior. Different types of static and vibration modes were derived respectively in sections 2.3 and 2.4, forming the ingredients for the component reduction methods.

By combining these ingredients the most common reduction methods can be obtained, as discussed in 2.5. These methods generally fall into two classes: fixed and free interface modes, referring to the type of vibration modes used in the basis. It was observed that no clear criteria exist as to when to applied each type of method. Therefore, section 2.6 introduced a mixed boundary component reduction method, called the Mixed Craig-Bampton method. It was shown that this method is a natural generalization of the existing regular and Dual Craig-Bampton methods. Table 2.1 summarizes all component reduction methods treated in this chapter in terms of ingredients and properties.

		Guyan	CB	Rubin/MacNeal	Dual CB	Mixed CB
Static modes						
Constraint	$\boldsymbol{\Psi}_c$	✓	✓			✓
Attachment (res.)	$\boldsymbol{\Psi}_r$			✓	✓	✓
Vibration modes						
Rigid body	$\boldsymbol{\Phi}_r$			✓	✓	?
Fixed interface	$\boldsymbol{\Phi}_i$		✓			✓
Free interface	$\boldsymbol{\Phi}_f$			✓	✓	✓
Properties						
Adaptiveness		–	–	+/–	++	+
Sparsity		+	+	–	+	+
Implementation		++	++	+	+/–	–
Experimental		–	–	+	+	+/–
Superelement?		✓	✓	✓	✗	✗

Table 2.1: Overview of component reduction methods, “CB” stands from Craig-Bampton.

The final section 2.7 of this chapter addressed the topic of modal truncation augmentation (MTA), a technique that aims to improve the spatial convergence of reduced component models. Especially for accurate forced response predictions this is relevant, since the external forces are usually neglected in the derivation of component reduction methods. The so-called MTA vectors were derived, which can be used in conjunction with both fixed and free interface vibration modes to enrich the reduction basis, thereby improving the spatial convergence of the reduced component.

3

Assembly of Component Models

Nothing is particularly hard if you divide it into small jobs.

(Henry Ford)

3.1 Introduction

After the reduced component models are created according to the methods treated in the previous chapter, or perhaps obtained from experiments, the next step is to assemble these models to obtain the structural dynamic model of the total system. The assembly techniques required to achieve this are an important aspect of dynamic substructuring analysis. In the majority of the literature on dynamic substructuring however, assembly is treated as an integral aspect of the reduction method and one simply assembles the components as superelements, i.e. like regular finite elements. In this chapter component assembly is considered from a wider perspective, with the aim to develop generic assembly procedures that allow (reduced) component modeling and assembly to be treated separately.

To this end, this chapter is organized as follows. The first part of this chapter is devoted to a general framework for component assembly. Section 3.2 describes the three possible assembly cases, for which assembly procedures are subsequently derived in sections 3.3, 3.4 and 3.5. The developed framework thereby enables assembly of all types of reduced substructures.

Thereafter, section 3.6 addresses the issue of non-conforming interface discretizations. Section 3.7 treats a method to account for additional physical effects arising from the interface. Finally, a sidestep is made in section 3.8, which addresses the decoupling (or disassembly) of components obtained from measurements of assembled systems. A brief summary is provided in section 3.9. Note that although the component reduction methods discussed in the previous chapter considered linear systems only, the assembly techniques presented here are equally valid for non-linear models. The relation between this and other chapters of part I is shown in figure 3.1.

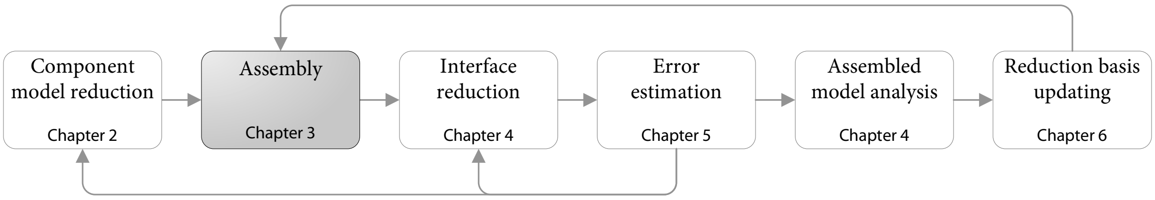


Figure 3.1: Current chapter in relation to other chapters in part I of this thesis.

3.2 General Framework for Component Assembly

In general, structural dynamic component models can be expressed either in terms of stiffness or flexibility, or a mix of both, at their interface degrees of freedom. This holds both for reduced and unreduced component models as well as for numerically and experimentally obtained models. Examples of substructure models expressed in terms of interface stiffness are regular (full) finite element models as well as models reduced using classical component reduction methods, such as the Craig-Bampton and MacNeal/Rubin methods. Hence the latter are often referred to as superelements. Substructure models having an interface flexibility representation arise from *dual* or *mixed* reduction techniques such as the Dual and Mixed Craig-Bampton methods, or from measured models.

Hence, the substructure DoF vector either contains some set of internal DoF and *interface displacements* or some set of internal DoF and *interface forces*. Given the different representations of the substructure models on the interface, three assembly cases can be distinguished:

1. Assembly of interface displacements to interface displacements: “stiffness assembly”.
2. Assembly of interface forces to interface forces: “flexibility assembly”.
3. Assembly of interface displacements to interface forces: “mixed assembly”.

These three cases are illustrated in figure 3.2 and will be treated in detail in the three subsequent sections, where a so called three-field variational formulation will be used to derive the required assembly procedures. Such an approach is needed to handle the mixed assembly problem; for the more straightforward cases of stiffness and flexibility assembly a two-field formulation is already sufficient (see [53]). Note that in the subsequent discussion in sections (3.3 to 3.5) it is assumed that the substructure interfaces are *conforming*, that is, their nodes are collocated and element shape functions are matching. Section 3.6 outlines a method that can be used in case the interfaces are non-conforming.

The notations used in this chapter are similar to those of the previous chapter, namely vectors $\boldsymbol{u}$ denote physical displacements, while a vector $\boldsymbol{q}$ denotes a set of generalized DoF. A superscript $\bar{\star}$ denotes properties belonging to a reduced substructure with force interface DoF (i.e. flexibility interface representation). Assembled matrices are indicated by $\hat{\star}$.

3.3 Stiffness Assembly

This section treats the assembly of components expressed in terms of stiffness at the structural interface. As indicated before, this can either be a full FE model, or a model reduced

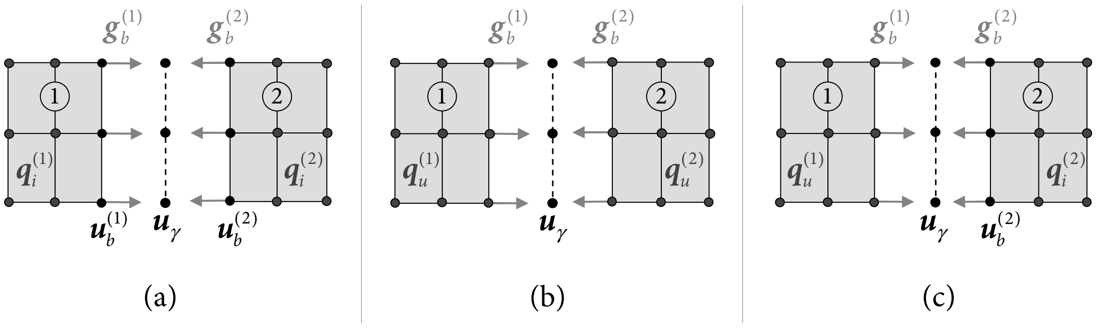


Figure 3.2: Three different cases encountered in assembly of reduced component models: "stiffness" assembly (a), "flexibility" assembly (b) and "mixed" assembly (c).

using any of the classic superelement methods (i.e. Craig-Bampton, Rubin or MacNeal, see section 2.5). The starting point is again the set of linear, discretized and possibly reduced equations of motion of a substructure s connected to other substructures:

$$\boldsymbol{M}^{(s)}\ddot{\boldsymbol{q}}^{(s)} + \boldsymbol{K}^{(s)}\boldsymbol{q}^{(s)} = \boldsymbol{f}^{(s)} + \boldsymbol{g}^{(s)} \tag{3.1}$$

Here $\boldsymbol{M}^{(s)}$ denotes the substructure's mass matrix, $\boldsymbol{K}^{(s)}$ is the stiffness matrix, $\boldsymbol{q}^{(s)}$ some vector of generalized degrees of freedom, $\boldsymbol{f}^{(s)}$ the external excitation vector and $\boldsymbol{g}^{(s)}$ the vector of connection forces felt from connected substructures. After partitioning the subsystem DoF (and the associated matrices) in an internal $_i$ and boundary $_b$ part the generalized DoF vector and vector of connection forces can be written as:

$$\boldsymbol{q}^{(s)} = \begin{bmatrix} \boldsymbol{q}_i^{(s)} \\ \boldsymbol{u}_b^{(s)} \end{bmatrix} \quad , \quad \boldsymbol{g}^{(s)} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_b^{(s)} \end{bmatrix} \tag{3.2}$$

Here $\boldsymbol{q}_i$ are the generalized internal DoF (for non-reduced systems these are simply the nodal displacements $\boldsymbol{u}_i$) and $\boldsymbol{u}_b$ the physical boundary displacement DoF. By definition the vector of connection forces $\boldsymbol{g}$ is zero at the internal subsystem DoF, while $\boldsymbol{g}_b$ is similar to the local Lagrange multipliers used for instance in the algebraic FETI method [146]. Between the substructures one can define the intermediate interface displacement field $\boldsymbol{u}_\gamma$, to govern the compatibility of substructural displacements at the interface. This compatibility condition ensures no relative motion between the boundaries of connected substructures and writes:

$$\boldsymbol{u}_b^{(s)} - \boldsymbol{L}_b^{(s)}\boldsymbol{u}_\gamma = \boldsymbol{0} \tag{3.3}$$

Here $\boldsymbol{L}_b^{(s)}$ is a Boolean matrix of size $n_b^{(s)} \times n_\gamma$, with $n_b^{(s)}$ the number of boundary DoF of the substructure and n_γ the number of unique boundary DoF of the total system, and localizes the substructure boundary DoF from the global intermediate displacement field. Its construction is treated in [53]. As a result, one ends up with a three-field formulation of the substructuring problem, having as independent unknowns the substructure DoF field $\boldsymbol{q}$, the field of interface connection forces $\boldsymbol{g}_b$ and the intermediate interface displacement field $\boldsymbol{u}_\gamma$. This situation is depicted for the assembly of two components in figure 3.2 (a). Taking a variational approach one can now obtain the assembled equations. To this end, the Lagrangian of this problem is set up as:

$$\mathcal{L}\left(\boldsymbol{q}^{(s)}, \boldsymbol{g}_b^{(s)}, \boldsymbol{u}_\gamma\right) = \sum_{s=1}^{n}\left(\frac{1}{2}\boldsymbol{q}^{(s)^T}\boldsymbol{K}^{(s)}\boldsymbol{q}^{(s)} - \boldsymbol{f}^{(s)^T}\boldsymbol{q}^{(s)} + \boldsymbol{g}_b^{(s)^T}\left(\boldsymbol{L}_b^{(s)}\boldsymbol{u}_\gamma - \boldsymbol{u}_b^{(s)}\right)\right) \tag{3.4}$$

Note that the above considers the static problem only. To include the inertia terms, one needs to go back to the variational principle in terms of the Hamiltonian [74]; these details will not be treated here. However, since in eq. (3.3) it was chosen to express the compatibility condition in terms of displacements (instead of accelerations) the coupling between substructure will occur in the assembled stiffness matrix. The assembled mass matrix, and possibly damping matrix, is hence not of primary interest. To find the assembled equations the variation is taken with respect to the free variables to find the equations of motion of the assembled system:

$$\begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{K}_{bi} & \boldsymbol{K}_{bb} & -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{L}_b \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}_b^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \\ \boldsymbol{g}_b \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_b \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{3.5}$$

Here the following block diagonal matrices have been defined:[1]

$$\begin{aligned} \boldsymbol{K}_{ii} &= \operatorname{diag}\left(\boldsymbol{K}_{ii}^{(1)}, \dots, \boldsymbol{K}_{ii}^{(n)}\right) \\ \boldsymbol{K}_{ib} &= \boldsymbol{K}_{bi}^T = \operatorname{diag}\left(\boldsymbol{K}_{ib}^{(1)}, \dots, \boldsymbol{K}_{ib}^{(n)}\right) \\ \boldsymbol{K}_{bb} &= \operatorname{diag}\left(\boldsymbol{K}_{bb}^{(1)}, \dots, \boldsymbol{K}_{bb}^{(n)}\right) \end{aligned} \tag{3.6}$$

Details on the content of these matrices for specific reduction methods are found in section 2.5. Furthermore, the vectors and Boolean matrix $\boldsymbol{L}_b$ are written in block form as:

$$\begin{aligned} \boldsymbol{q}_i &= \begin{bmatrix} \boldsymbol{q}_i^{(1)} \\ \vdots \\ \boldsymbol{q}_i^{(n)} \end{bmatrix} \triangleq \operatorname{col}\left(\boldsymbol{q}_i^{(1)}; \dots; \boldsymbol{q}_i^{(n)}\right) \\ \boldsymbol{u}_b &= \operatorname{col}\left(\boldsymbol{u}_b^{(1)}; \dots; \boldsymbol{u}_b^{(n)}\right) \\ \boldsymbol{f}_i &= \operatorname{col}\left(\boldsymbol{f}_i^{(1)}; \dots; \boldsymbol{f}_i^{(n)}\right) \\ \boldsymbol{f}_b &= \operatorname{col}\left(\boldsymbol{f}_b^{(1)}; \dots; \boldsymbol{f}_b^{(n)}\right) \\ \boldsymbol{g}_b &= \operatorname{col}\left(\boldsymbol{g}_b^{(1)}; \dots; \boldsymbol{g}_b^{(n)}\right) \\ \boldsymbol{L}_b &= \operatorname{col}\left(\boldsymbol{L}_b^{(1)}; \dots; \boldsymbol{L}_b^{(n)}\right) \end{aligned} \tag{3.7}$$

In the above equation 3.5 one can recognize the third equation as the compatibility conditions, governing the compatibility between the $\boldsymbol{u}_b$ and $\boldsymbol{u}_\gamma$. The last row is the equilibrium condition on the interface, stating that the sum of the substructure connection forces must be zero.

For completeness, the assembled mass matrix is shown below; it is a simple matrix containing the block diagonal component submatrices.

$$\hat{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{ib} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{M}_{bi} & \boldsymbol{M}_{bb} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{3.8}$$

---

[1] This block diagonal notation is used for explanatory purposes only. Obviously, the sparse band-diagonality of the component matrices is lost due to this ordering of the equations, which is undesirable from a computational perspective. In a practical implementation of the assembly procedures, an additional DoF reordering of the transformation matrices (i.e. eqs. (3.10), (3.13) and (3.15)) is advised.

With the mass matrix, the complete dynamic equations of the assembled system are found. In case damping is formulated on a substructure level, the assembled damping matrix has the same topology. However, these assembled equations of motion still contain the full three-fields, which is for most analyses inefficient from a computational point of view. Therefore, it is desired to simplify the equations. In essence two ways exist to do this, namely so called *primal* or *dual* assembly, as discussed next.

3.3.1 Dual Assembly

In dual assembly, one eliminates the interface connection force fields by realizing that the interface forces should be equal and opposite to satisfy the interface equilibrium. To *a priori* satisfy this condition a unique field of interface forces $\boldsymbol{\lambda}$ is introduced, as follows:

$$\boldsymbol{g}_b = -\boldsymbol{B}_b^T \boldsymbol{\lambda}$$

Here $\boldsymbol{B}_b$ is a signed Boolean matrix acting on the substructure interface DoF and $\boldsymbol{\lambda}$ corresponds physically to the interface force intensities. Note that the minus sign is chosen to stress the fact that whereas $\boldsymbol{g}_b$ was seen as an external force for the substructure, $\boldsymbol{\lambda}$ is considered an internal force. Due to the construction of the Boolean matrices it holds that [53]:

$$\boldsymbol{B}_b \boldsymbol{L}_b = \boldsymbol{0} \tag{3.9}$$

Hence this choice for the interface connection forces satisfies the interface equilibrium for any $\boldsymbol{\lambda}$. This choice gives rise to the following transformation:

$$\begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \\ \boldsymbol{g}_b \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{B}_b^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \\ \boldsymbol{\lambda} \\ \boldsymbol{u}_\gamma \end{bmatrix} \tag{3.10}$$

Substituting this transformation in the three-field assembled equations of motion (3.5) replaces the local connection forces by the unique global field $\boldsymbol{\lambda}$, satisfying the interface equilibrium condition.[2] To end up with a symmetric system one can use the above transformation to subsequently pre-multiply the equations. This eliminates the intermediate displacement field $\boldsymbol{u}_\gamma$, due to the relation between the Boolean matrices in (3.9). The procedure is illustrated in figure 3.3 and the simplified assembled equations become:

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{ib} & \boldsymbol{0} \\ \boldsymbol{M}_{bi} & \boldsymbol{M}_{bb} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{u}}_b \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib} & \boldsymbol{0} \\ \boldsymbol{K}_{bi} & \boldsymbol{K}_{bb} & \boldsymbol{B}_b^T \\ \boldsymbol{0} & \boldsymbol{B}_b & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_b \\ \boldsymbol{0} \end{bmatrix} \tag{3.11}$$

The above system is called the *dual* assembled system, since the unknowns defining the interface problem are forces which are mathematically dual to the original displacement unknowns. As a result, the compatibility condition is present explicitly in the assembled equations of motion, i.e. the last row in eq. (3.11). Note that dual assembly approaches were

---

[2] It should be noted that in this work, Lagrange multipliers are used to enforce compatibility on the interface. In the same framework one can also use additional Lagrange multipliers to enforce Dirichlet boundary conditions imposed on substructure DoF that are not on the interface.

already considered in the early days of finite element theory, but only became popular in the 1990's as a way to implement efficient solvers on parallel processing computers. This led to the family of parallel solvers known as FETI (Finite Elements Tearing and Interconnecting) [65].

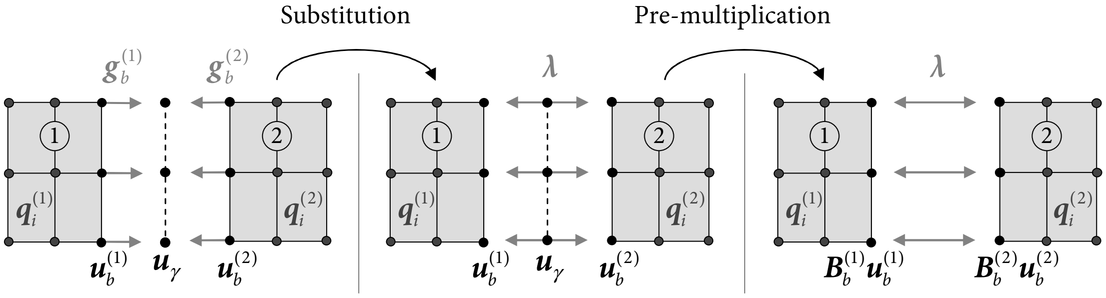


Figure 3.3: Simplification of the three-field formulation for stiffness assembly through a dual approach.

3.3.2 Primal Assembly

An even further simplified expression for the assembled system can be obtained by realizing that the compatibility condition can be *a priori* satisfied by choosing a unique set of substructure interface DoF as:

$$\boldsymbol{u}_b = \boldsymbol{L}_b \boldsymbol{u}_\gamma \tag{3.12}$$

This choice gives the following transformation:

$$\begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \\ \boldsymbol{g}_b \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}_b \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{g}_b \\ \boldsymbol{u}_\gamma \end{bmatrix} \tag{3.13}$$

Again, this transformation is substituted in (3.5) thereby eliminating the substructure boundary DoF sets $\boldsymbol{u}_b$ and satisfying the interface compatibility condition. Pre-multiplication is then needed to obtain a symmetric system of equations. Using again the relation between the Boolean matrices $\boldsymbol{L}_b$ and $\boldsymbol{B}_b$ in eq. (3.9) the equilibrium condition is also satisfied and drops out of the equation. The procedure is illustrated in figure 3.4 and results in the following expression for the assembled system:

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{ib}\boldsymbol{L}_b \\ \boldsymbol{L}_b^T\boldsymbol{M}_{bi} & \boldsymbol{L}_b^T\boldsymbol{M}_{bb}\boldsymbol{L}_b \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib}\boldsymbol{L}_b \\ \boldsymbol{L}_b^T\boldsymbol{K}_{bi} & \boldsymbol{L}_b^T\boldsymbol{K}_{bb}\boldsymbol{L}_b \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{L}_b^T\boldsymbol{f}_b \end{bmatrix} \tag{3.14}$$

The above equations form the so called *primal* assembled system; the most compact form of the assembled equations of motion using a minimum number of DoF. Note that this type of assembly is the way individual elements are classically assembled in a finite element method. Furthermore primal assembly is almost always used in component mode synthesis methods.

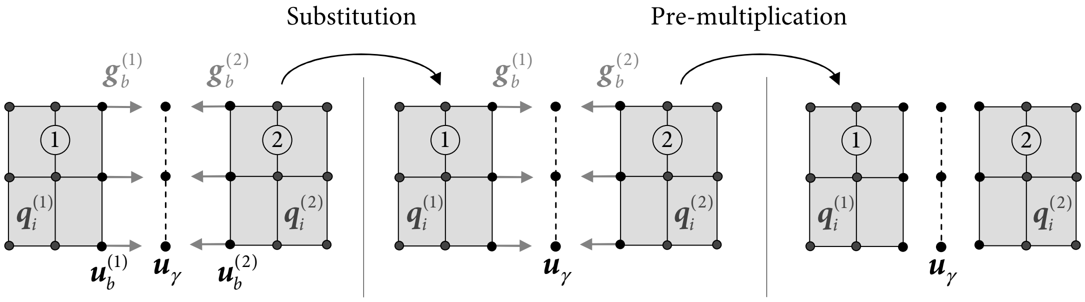


Figure 3.4: Simplification of the three-field formulation for stiffness assembly by a primal approach.

3.3.3 A Mix of Both: Dirichlet-Neumann Assembly

In addition to the primal and dual assembly methods treated above, the assembled equations in (3.5) can also be simplified by combining both methods, that is, by choosing both a unique set of interface DoF and interface forces. Thereby, both the equilibrium and compatibility condition on the interface are satisfied *a priori*. This gives rise to the following transformation:

$$\begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \\ \boldsymbol{g}_b \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}_b \\ \boldsymbol{0} & -\boldsymbol{B}_b^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\lambda} \\ \boldsymbol{u}_\gamma \end{bmatrix} \tag{3.15}$$

Substitution of this transformation in the three-field assembled equations of motion in (3.5) simultaneously introduces the interface force field $\boldsymbol{\lambda}$ and eliminates the substructure boundary DoF $\boldsymbol{u}_b$, as illustrated in figure 3.5. This gives:

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{0} & \boldsymbol{M}_{ib}\boldsymbol{L}_b \\ \boldsymbol{M}_{bi} & \boldsymbol{0} & \boldsymbol{M}_{bb}\boldsymbol{L}_b \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{\lambda}} \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{0} & \boldsymbol{K}_{ib}\boldsymbol{L}_b \\ \boldsymbol{K}_{bi} & \boldsymbol{B}_b^T & \boldsymbol{K}_{bb}\boldsymbol{L}_b \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\lambda} \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_b \end{bmatrix} \tag{3.16}$$

Note that two lines of zeros have dropped out of the equation such that, although not directly clear from the above expression, a square system of equations is obtained. As can be seen this expression for the assembled system is non symmetric, therefore in practice this form of the assembled equations is often not very useful. Pre-multiplication with this transformation matrix does solve this, as it eliminates the Lagrange multipliers and results in the primal assembled system of eq. (3.14). The reason this type of assembly is still shown is that a similar transformation is used in the case of mixed assembly. Note that the name of this type of assembly refers to the way the assembled system can be solved, namely using the Gauss-Seidel method. This leads to so-called Dirichlet-Neumann iterations known from domain decomposition theory [199].

3.4 Flexibility Assembly

This section will address the assembly of subsystems reduced such that they are expressed in terms of flexibility at their interface, resulting for example from a Dual Craig-Bampton

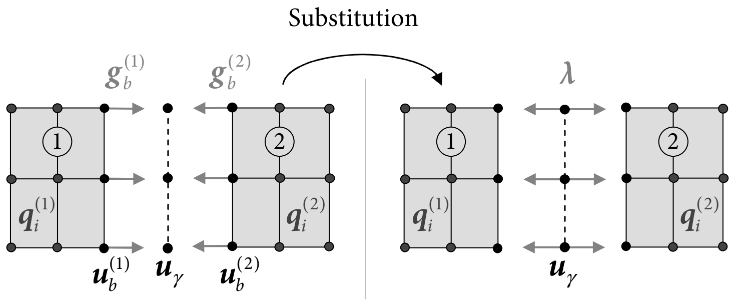


Figure 3.5: Simplification of the stiffness three-field formulation by a Dirichlet-Neumann approach.

reduction (see section 2.5.4). For such systems, the original displacement field $\boldsymbol{u}^{(s)}$ of the substructure is approximated by some set of generalized DoF and interface forces, as:

$$\boldsymbol{u}^{(s)} \approx \bar{\boldsymbol{R}}^{(s)}\bar{\boldsymbol{q}}^{(s)} \quad \rightarrow \quad \begin{bmatrix} \boldsymbol{u}_i^{(s)} \\ \boldsymbol{u}_b^{(s)} \end{bmatrix} \approx \begin{bmatrix} \bar{\boldsymbol{R}}_{ii}^{(s)} & \bar{\boldsymbol{R}}_{ib}^{(s)} \\ \bar{\boldsymbol{R}}_{bi}^{(s)} & \bar{\boldsymbol{R}}_{bb}^{(s)} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{q}}_i^{(s)} \\ \boldsymbol{g}_b^{(s)} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{R}}_i^{(s)} & \bar{\boldsymbol{R}}_b^{(s)} \end{bmatrix} \bar{\boldsymbol{q}}^{(s)} \tag{3.17}$$

To come to the assembled equations of motion for multiple components one needs to formulate again the Lagrangian of the substructuring problem. However due to the reduction, the boundary DoF are no longer explicitly present in the reduced substructure descriptions. Hence they can be recovered from the reduction basis by

$$\boldsymbol{u}_b^{(s)} = \begin{bmatrix} \bar{\boldsymbol{R}}_{bi}^{(s)} & \bar{\boldsymbol{R}}_{bb}^{(s)} \end{bmatrix} \bar{\boldsymbol{q}}^{(s)}, \tag{3.18}$$

where $\bar{\boldsymbol{R}}_b^{(s)}$ is the part of the reduction matrix associated with the boundary DoF. This then allows to set up the Lagrangian of the flexibility assembly problem as:

$$\begin{aligned} \mathcal{L}\left(\bar{\boldsymbol{q}}^{(s)}, \boldsymbol{g}_b^{(s)}, \boldsymbol{u}_\gamma\right) = \sum_{s=1}^{n} \Bigg(& \frac{1}{2}\bar{\boldsymbol{q}}^{(s)^T}\bar{\boldsymbol{K}}^{(s)}\bar{\boldsymbol{q}}^{(s)} - \bar{\boldsymbol{f}}^{(s)^T}\bar{\boldsymbol{q}}^{(s)} \dots \\ & + \boldsymbol{g}_b^{(s)^T}\left(\boldsymbol{L}_b^{(s)}\boldsymbol{u}_\gamma - \begin{bmatrix} \bar{\boldsymbol{R}}_{bi}^{(s)} & \bar{\boldsymbol{R}}_{bb}^{(s)} \end{bmatrix} \bar{\boldsymbol{q}}^{(s)}\right)\Bigg) \end{aligned} \tag{3.19}$$

where:

$$\bar{\boldsymbol{K}}^{(s)} = \bar{\boldsymbol{R}}^{(s)^T}\boldsymbol{K}^{(s)}\bar{\boldsymbol{R}}^{(s)} \quad \text{and} \quad \bar{\boldsymbol{f}}^{(s)} = \bar{\boldsymbol{R}}^{(s)^T}\boldsymbol{f}^{(s)}. \tag{3.20}$$

This assembly case is illustrated for two subsystems in figure 3.2 (b). Taking again the variation of this expression to the free variables, one obtains the assembled equations of motion as:

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{ii} & \bar{\boldsymbol{M}}_{ib} & \boldsymbol{0} \\ \bar{\boldsymbol{M}}_{bi} & \bar{\boldsymbol{M}}_{bb} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\bar{\boldsymbol{q}}}_i \\ \ddot{\boldsymbol{g}}_b \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{K}}_{ii} & \bar{\boldsymbol{K}}_{ib} & \boldsymbol{0} \\ \bar{\boldsymbol{K}}_{bi} & \bar{\boldsymbol{K}}_{bb} & \boldsymbol{L}_b \\ \boldsymbol{0} & \boldsymbol{L}_b^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{q}}_i \\ \boldsymbol{g}_b \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_i \\ \bar{\boldsymbol{f}}_b \\ \boldsymbol{0} \end{bmatrix} \tag{3.21}$$

Analogous to the previous section, the following (reduced) block diagonal matrices have been defined:

$$\begin{aligned} \bar{\boldsymbol{K}}_{ii} &= \operatorname{diag}\left(\bar{\boldsymbol{K}}_{ii}^{(1)}, \dots, \bar{\boldsymbol{K}}_{ii}^{(n)}\right) && \text{with} \quad \bar{\boldsymbol{K}}_{ii}^{(s)} = \bar{\boldsymbol{R}}_i^{(s)^T}\boldsymbol{K}^{(s)}\bar{\boldsymbol{R}}_i^{(s)} \quad \text{for} \quad s = 1\dots n \\ \bar{\boldsymbol{K}}_{ib} &= \bar{\boldsymbol{K}}_{bi}^T = \operatorname{diag}\left(\bar{\boldsymbol{K}}_{ib}^{(1)}, \dots, \bar{\boldsymbol{K}}_{ib}^{(n)}\right) && \text{with} \quad \bar{\boldsymbol{K}}_{ib}^{(s)} = \bar{\boldsymbol{R}}_i^{(s)^T}\boldsymbol{K}^{(s)}\bar{\boldsymbol{R}}_b^{(s)} - \bar{\boldsymbol{R}}_{bi}^{(s)^T} \\ \bar{\boldsymbol{K}}_{bb} &= \operatorname{diag}\left(\bar{\boldsymbol{K}}_{bb}^{(1)}, \dots, \bar{\boldsymbol{K}}_{bb}^{(n)}\right) && \text{with} \quad \bar{\boldsymbol{K}}_{bb}^{(s)} = \bar{\boldsymbol{R}}_b^{(s)^T}\boldsymbol{K}^{(s)}\bar{\boldsymbol{R}}_b^{(s)} - \bar{\boldsymbol{R}}_{bb}^{(s)} - \bar{\boldsymbol{R}}_{bb}^{(s)^T} \end{aligned} \tag{3.22}$$

The mass terms are defined similarly, but obviously do not contain any coupling terms:

$$\begin{aligned}
\bar{\boldsymbol{M}}_{ii} &= \operatorname{diag}\left(\bar{\boldsymbol{M}}_{ii}^{(1)},\ldots,\bar{\boldsymbol{M}}_{ii}^{(n)}\right) &&\text{with}\quad \bar{\boldsymbol{M}}_{ii}^{(s)} = \bar{\boldsymbol{R}}_{i}^{(s)^T}\boldsymbol{M}^{(s)}\bar{\boldsymbol{R}}_{i}^{(s)}\\
\bar{\boldsymbol{M}}_{ib} = \bar{\boldsymbol{M}}_{bi}^{T} &= \operatorname{diag}\left(\bar{\boldsymbol{M}}_{ib}^{(1)},\ldots,\bar{\boldsymbol{M}}_{ib}^{(n)}\right) &&\text{with}\quad \bar{\boldsymbol{M}}_{ib}^{(s)} = \bar{\boldsymbol{R}}_{i}^{(s)^T}\boldsymbol{M}^{(s)}\bar{\boldsymbol{R}}_{b}^{(s)}\\
\bar{\boldsymbol{M}}_{bb} &= \operatorname{diag}\left(\bar{\boldsymbol{M}}_{bb}^{(1)},\ldots,\bar{\boldsymbol{M}}_{bb}^{(n)}\right) &&\text{with}\quad \bar{\boldsymbol{M}}_{bb}^{(s)} = \bar{\boldsymbol{R}}_{b}^{(s)^T}\boldsymbol{M}^{(s)}\bar{\boldsymbol{R}}_{b}^{(s)}
\end{aligned} \tag{3.23}$$

Furthermore, the vector $\boldsymbol{g}_b$ and Boolean matrix $\boldsymbol{L}_b$ are as defined in the previous section, while the remaining vectors are defined as:

$$\begin{aligned}
\bar{\boldsymbol{q}}_i &= \operatorname{col}\left(\bar{\boldsymbol{q}}_i^{(1)};\ldots;\bar{\boldsymbol{q}}_i^{(n)}\right) && \\
\bar{\boldsymbol{f}}_i &= \operatorname{col}\left(\bar{\boldsymbol{f}}_i^{(1)};\ldots;\bar{\boldsymbol{f}}_i^{(n)}\right) &&\text{with}\quad \bar{\boldsymbol{f}}_i^{(s)} = \bar{\boldsymbol{R}}_i^{(s)^T}\boldsymbol{f}^{(s)}\\
\bar{\boldsymbol{f}}_b &= \operatorname{col}\left(\bar{\boldsymbol{f}}_b^{(1)};\ldots;\bar{\boldsymbol{f}}_b^{(n)}\right) &&\text{with}\quad \bar{\boldsymbol{f}}_b^{(s)} = \bar{\boldsymbol{R}}_b^{(s)^T}\boldsymbol{f}^{(s)}
\end{aligned} \tag{3.24}$$

Note that the part $\bar{\boldsymbol{K}}_{bb}$ of the stiffness matrix acting on the interface DoF is in fact a true flexibility matrix and $\bar{\boldsymbol{f}}_b$ are in fact interface displacements, namely the interface displacements due to the applied external forces $\boldsymbol{f}$. In section 2.5.4 the full expressions of these submatrices are derived for the case of the Dual Craig-Bampton method.

Although the starting point was a three-field formulation, the equation (3.21) is actually the dual assembled form for interface flexibility type of substructures, as can be seen by comparison with eq. (3.11). In the dual assembly of the interface flexibility, the intermediate displacement $\boldsymbol{u}_\gamma$ plays the role of the Lagrange multipliers and enforces the equilibrium of the interface force DoF. Due to the reduction, the compatibility conditions for the components are stated in the second row in a weakened form; see section 2.5.4. The third row can be recognized as the equilibrium condition.

From the above set of assembled equations, the only simplification to be made is therefore to go to a true primal system as in the previous section. As in the dual assembly of the previous section, the interface forces are again chosen in the form:

$$\boldsymbol{g}_b = -\boldsymbol{B}_b^T\boldsymbol{\lambda} \tag{3.25}$$

As before, this gives the following transformation:

$$\begin{bmatrix}\bar{\boldsymbol{q}}_i\\ \boldsymbol{g}_b\\ \boldsymbol{u}_\gamma\end{bmatrix} = \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & -\boldsymbol{B}_b^T & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\bar{\boldsymbol{q}}_i\\ \boldsymbol{\lambda}\\ \boldsymbol{u}_\gamma\end{bmatrix} \tag{3.26}$$

First, the above transformation is again substituted in eq. (3.21). This introduces the unique interface force field $\boldsymbol{\lambda}$ and eliminates the $\boldsymbol{g}_b$, hence the equilibrium condition is satisfied. Pre-multiplication then eliminates the interface displacement field (illustrated in figure 3.6) and gives the primal assembled system as:

$$\begin{bmatrix}\bar{\boldsymbol{M}}_{ii} & -\bar{\boldsymbol{M}}_{ib}\boldsymbol{B}_b^T\\ -\boldsymbol{B}_b\bar{\boldsymbol{M}}_{bi} & \boldsymbol{B}_b\bar{\boldsymbol{M}}_{bb}\boldsymbol{B}_b^T\end{bmatrix}\begin{bmatrix}\ddot{\bar{\boldsymbol{q}}}_i\\ \ddot{\boldsymbol{\lambda}}\end{bmatrix} + \begin{bmatrix}\bar{\boldsymbol{K}}_{ii} & -\bar{\boldsymbol{K}}_{ib}\boldsymbol{B}_b^T\\ -\boldsymbol{B}_b\bar{\boldsymbol{K}}_{bi} & \boldsymbol{B}_b\bar{\boldsymbol{K}}_{bb}\boldsymbol{B}_b^T\end{bmatrix}\begin{bmatrix}\bar{\boldsymbol{q}}_i\\ \boldsymbol{\lambda}\end{bmatrix} = \begin{bmatrix}\bar{\boldsymbol{f}}_i\\ -\boldsymbol{B}_b\bar{\boldsymbol{f}}_b\end{bmatrix} \tag{3.27}$$

As can be seen, this primal form of the equations of motion is very similar to that of the previous section, consisting only of the generalized substructure DoF and one unique interface field. Due to the flexibility expression of the substructure interfaces the unique interface field here is the set of interface forces, whereas in the previous section it was the unique set of interface displacements.

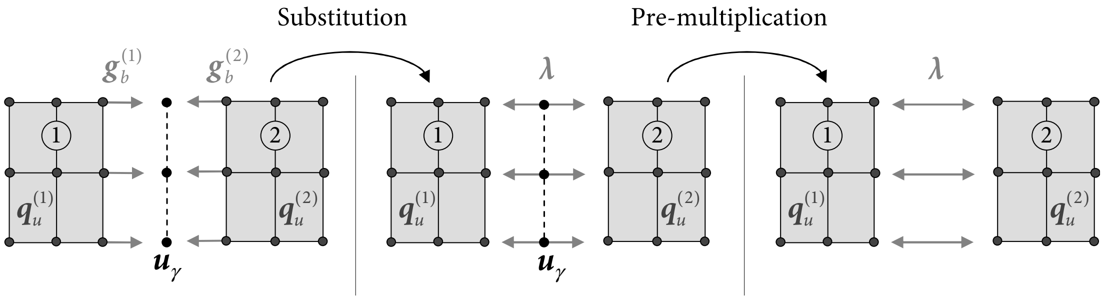


Figure 3.6: Simplification of the three-field formulation for the flexibility assembly case.

3.5 Mixed Assembly

In this section the case of mixed assembly is addressed, where some subsystems are expressed in terms of stiffness at their interfaces while the other interfaces are expressed in terms of flexibility. This is for instance the case when assembling a combination of Craig-Bampton and Dual Craig-Bampton reduced systems, or when assembling Mixed Craig-Bampton reduced systems (see section 2.6). To derive the assembled equations of motion in this case, one simply combines the Lagrangians found earlier for the stiffness and flexibility assembly cases in eqs. (3.4) and (3.19) and takes their variation.

Suppose that some n_f substructures, expressed in flexibility at their interfaces, are to be assembled with n_s components which have an interface stiffness representation. This is schematically shown in figure 3.2 (c) for two substructures. The following three-field assembled equations are then found from a variational approach:

$$\cdots + \begin{bmatrix} \bar{K}_{ii} & \bar{K}_{ib} & 0 & 0 & 0 & 0 \\ \bar{K}_{bi} & \bar{K}_{bb} & 0 & 0 & 0 & L_b^f \\ 0 & 0 & K_{ii} & K_{ib} & 0 & 0 \\ 0 & 0 & K_{bi} & K_{bb} & -I & 0 \\ 0 & 0 & 0 & -I & 0 & L_b^s \\ 0 & L_b^{f^T} & 0 & 0 & L_b^{s^T} & 0 \end{bmatrix} \begin{bmatrix} \bar{q}_i \\ g_b^f \\ q_i \\ u_b^s \\ g_b^s \\ u_\gamma \end{bmatrix} = \begin{bmatrix} \bar{f}_i \\ \bar{f}_b^f \\ f_i \\ f_b^s \\ 0 \\ 0 \end{bmatrix} \tag{3.28}$$

Similar to the previous sections, the submatrices are defined as the block diagonals of the component submatrices; the superscripts f and s denote "flexibility" and "stiffness", respectively. In this equation, the second row is the weak compatibility condition for the n_f "flexibility components" while the fifth row can be recognized as the compatibility condition for the n_s "stiffness components". The sixth row constitutes the equilibrium condition. The mass matrix is not shown for compactness. The above equations can be rearranged by grouping similar terms:

$$\cdots + \left[\begin{array}{cc|cc|c|c} \bar{K}_{ii} & 0 & \bar{K}_{ib} & 0 & 0 & 0 \\ 0 & K_{ii} & 0 & 0 & K_{ib} & 0 \\ \hline \bar{K}_{bi} & 0 & \bar{K}_{bb} & 0 & 0 & L_b^f \\ 0 & 0 & 0 & 0 & -I & L_b^s \\ \hline 0 & K_{bi} & 0 & -I & K_{bb} & 0 \\ \hline 0 & 0 & L_b^{f^T} & L_b^{s^T} & 0 & 0 \end{array}\right] \left[\begin{array}{c} \bar{q}_i \\ q_i \\ \hline g_b^f \\ g_b^s \\ \hline u_b^s \\ \hline u_\gamma \end{array}\right] = \left[\begin{array}{c} \bar{f}_i \\ f_i \\ \hline \bar{f}_b^f \\ 0 \\ \hline f_b^s \\ \hline 0 \end{array}\right] \tag{3.29}$$

This allows a more compact notation, as:

$$\begin{bmatrix} \boldsymbol{M}_{qq} & \boldsymbol{M}_{qg} & \boldsymbol{M}_{qb} & \boldsymbol{0} \\ \boldsymbol{M}_{gq} & \boldsymbol{M}_{gg} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{M}_{bq} & \boldsymbol{0} & \boldsymbol{M}_{bb} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \ddot{\boldsymbol{g}}_b \\ \ddot{\boldsymbol{u}}_b^s \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{qq} & \boldsymbol{K}_{qg} & \boldsymbol{K}_{qb} & \boldsymbol{0} \\ \boldsymbol{K}_{gq} & \boldsymbol{K}_{gg} & -\boldsymbol{A}_s^T & \boldsymbol{L}_b \\ \boldsymbol{K}_{bq} & -\boldsymbol{A}_s & \boldsymbol{K}_{bb} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}_b^T & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{g}_b \\ \boldsymbol{u}_b^s \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_q \\ \bar{\boldsymbol{f}}_b \\ \boldsymbol{f}_b^s \\ \boldsymbol{0} \end{bmatrix} \tag{3.30}$$

As in the previous sections, the three-field assembled equations of motion can be simplified to find a more efficient expression for the assembled system. However, since mixed assembly is considered, one cannot simply apply the primal or dual assembly methods of the previous sections. Instead, the transformations should also be mixed. First it should be realized that in the case of mixed assembly both a unique interface displacement $\boldsymbol{u}_\gamma$ field and unique interface force field $\boldsymbol{\lambda}$ are needed to facilitate the interaction between the force and displacement interface DoF of both substructures. One can then devise a transformation in the form:

$$\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{g}_b \\ \boldsymbol{u}_b^s \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{B}_b^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}_b^s \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\lambda} \\ \boldsymbol{u}_\gamma \end{bmatrix} \tag{3.31}$$

This transformation corresponds to primal assembly for the "flexibility" structures as in the previous section, whereas the "stiffness" components are subject to Dirichlet-Neumann assembly as outlined in section 3.3.3. Substitution of this transformation in eq. (3.28) introduces both the unique interface force field and eliminates $\boldsymbol{u}_b^s$, as shown in figure 3.7. Premultiplication is needed only for the sake of symmetry and gives the mixed assembled system:

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{M}_{qq} & -\boldsymbol{M}_{qg}\boldsymbol{B}_b^T & \boldsymbol{M}_{qb}\boldsymbol{L}_b^s \\ -\boldsymbol{B}_b\boldsymbol{M}_{gq} & \boldsymbol{B}_b\boldsymbol{M}_{gg}\boldsymbol{B}_b^T & \boldsymbol{0} \\ \boldsymbol{L}_b^{s^T}\boldsymbol{M}_{bq} & \boldsymbol{0} & \boldsymbol{L}_b^{s^T}\boldsymbol{M}_{bb}\boldsymbol{L}_b^s \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \ddot{\boldsymbol{\lambda}} \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \cdots \\ &\qquad \begin{bmatrix} \boldsymbol{K}_{qq} & -\boldsymbol{K}_{qg}\boldsymbol{B}_b^T & \boldsymbol{K}_{qb}\boldsymbol{L}_b^s \\ -\boldsymbol{B}_b\boldsymbol{K}_{gq} & \boldsymbol{B}_b\boldsymbol{K}_{gg}\boldsymbol{B}_b^T & -\boldsymbol{B}_b^f\boldsymbol{L}_b^f \\ \boldsymbol{L}_b^{s^T}\boldsymbol{K}_{bq} & -\boldsymbol{L}_b^{f^T}\boldsymbol{B}_b^{f^T} & \boldsymbol{L}_b^{s^T}\boldsymbol{K}_{bb}\boldsymbol{L}_b^s \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\lambda} \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_q \\ -\boldsymbol{B}_b\bar{\boldsymbol{f}}_b \\ \boldsymbol{L}_b^{s^T}\boldsymbol{f}_b \end{bmatrix} \end{aligned} \tag{3.32}$$

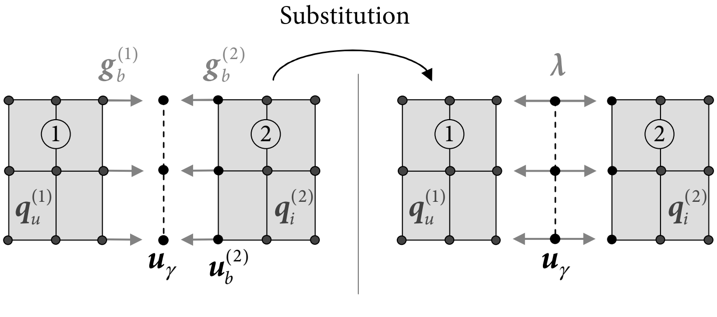


Figure 3.7: Simplification of the three-field formulation for the mixed assembly case.

For the sake of illustration, the above assembled equations are now worked out for the case depicted in figure 3.2 (c), namely the assembly of two components, with substructure 1 has force interface DoF and substructure 2 displacement DoF. Furthermore, it is assumed that the boundary DoF of both substructures are ordered equally.

In this case $\boldsymbol{L}_b^f$, $\boldsymbol{L}_b^s$ and $\boldsymbol{B}_b^s$ are identity matrices, while $\boldsymbol{B}_b^f$ is minus identity. If the block matrices are expanded and the DoF sets are reordered such that the Lagrange multipliers are associated to component 1 and the interface displacement field is associated to component 2, one can write the assembled equations of motion according to:

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{ii}^{(1)} & \bar{\boldsymbol{M}}_{ib}^{(1)} & \boldsymbol{0} & \boldsymbol{0} \\ \bar{\boldsymbol{M}}_{bi}^{(1)} & \bar{\boldsymbol{M}}_{bb}^{(1)} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}_{ii}^{(2)} & \boldsymbol{M}_{ib}^{(2)} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}_{bi}^{(2)} & \boldsymbol{M}_{bb}^{(2)} \end{bmatrix} \begin{bmatrix} \ddot{\bar{\boldsymbol{q}}}_i^{(1)} \\ \ddot{\boldsymbol{\lambda}} \\ \ddot{\boldsymbol{q}}_i^{(2)} \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{K}}_{ii}^{(1)} & \bar{\boldsymbol{K}}_{ib}^{(1)} & \boldsymbol{0} & \boldsymbol{0} \\ \bar{\boldsymbol{K}}_{bi}^{(1)} & \bar{\boldsymbol{K}}_{bb}^{(1)} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{ii}^{(2)} & \boldsymbol{K}_{ib}^{(2)} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{K}_{bi}^{(2)} & \boldsymbol{K}_{bb}^{(2)} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{q}}_i^{(1)} \\ \boldsymbol{\lambda} \\ \boldsymbol{q}_i^{(2)} \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_i^{(1)} \\ \bar{\boldsymbol{f}}_b^{(1)} \\ \boldsymbol{f}_i^{(1)} \\ \boldsymbol{f}_b^{(2)} \end{bmatrix} \tag{3.33}$$

From the above equation one can clearly see the way the two systems interact. In addition to the external excitations, component 1 is excited by interface displacements from component 2 through its boundary DoF, while component 2 feels additional forces from component 1 through its interface.

The general framework for substructure assembly presented in the last three sections is summarized in table 3.1. This table shows the DoF sets involved with the three-field formulation as well as the possible dual, primal and Dirichlet-Neumann simplifications and the resulting DoF sets. Furthermore, the total number of DoF n of the assembled system is indicated, where n_i is the sum of all the substructure internal/generalized DoF and n_b the number of unique boundary DoF in the total system.

	Three-field	Dual	Primal	Dirichlet-Neumann
Stiffness	✓ Start: $\begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \\ \boldsymbol{g}_b \\ \boldsymbol{u}_\gamma \end{bmatrix}$ $n = n_i + 5n_b$	✓ $\xrightarrow{\boldsymbol{g}_b = -\boldsymbol{B}_b^T \boldsymbol{\lambda}}$ $\begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \\ \boldsymbol{\lambda} \end{bmatrix}$ $n = n_i + 3n_b$	✓ $\xrightarrow{\boldsymbol{u}_b = \boldsymbol{L}_b \boldsymbol{u}_\gamma}$ $\begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_\gamma \end{bmatrix}$ $n = n_i + n_b$	✓ $\xrightarrow{\substack{\boldsymbol{g}_b = -\boldsymbol{B}_b^T \boldsymbol{\lambda} \\ \boldsymbol{u}_b = \boldsymbol{L}_b \boldsymbol{u}_\gamma}}$ $\begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\lambda} \\ \boldsymbol{u}_\gamma \end{bmatrix}$ $n = n_i + 2n_b$
Flexibility	✗	✓ Start: $\begin{bmatrix} \bar{\boldsymbol{q}}_i \\ \boldsymbol{g}_b \\ \boldsymbol{u}_\gamma \end{bmatrix}$ $n = n_i + 3n_b$	✓ $\xrightarrow{\boldsymbol{g}_b = -\boldsymbol{B}_b^T \boldsymbol{\lambda}}$ $\begin{bmatrix} \bar{\boldsymbol{q}}_i \\ \boldsymbol{\lambda} \end{bmatrix}$ $n = n_i + n_b$	✗
Mixed	✓ Start: $\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{g}_b \\ \boldsymbol{u}_b^s \\ \boldsymbol{u}_\gamma \end{bmatrix}$ $n = n_i + 4n_b$	✗	✗	✓ $\xrightarrow{\substack{\boldsymbol{g}_b = -\boldsymbol{B}_b^T \boldsymbol{\lambda} \\ \boldsymbol{u}_b = \boldsymbol{L}_b \boldsymbol{u}_\gamma}}$ $\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{\lambda} \\ \boldsymbol{u}_\gamma \end{bmatrix}$ $n = n_i + 2n_b$

Table 3.1: Simplification of the three-field formulation for the mixed assembly case.

Intermezzo: Relations Between Reduction Methods and Assembly Techniques

From the previous chapter it became clear that basically three types of component model reduction methods exist: those employing fixed, free or mixed interface vibration modes in the reduction basis. In the preceding sections it was shown that also three types of assembly exist: stiffness, flexibility and mixed assembly. A strong relation exists between these assembly methods and reduction techniques. Indeed, the ingredients used in the reduction basis govern whether the interface degrees of freedom are displacement DoF, force DoF, or a mixed of both. This in turn determines the kind of assembly procedure to be used.

In the case of fixed interface modes, the boundary DoF are automatically retained and a stiffness representation at the interface is obtained, like in the Craig-Bampton method. The substructures are then naturally assembled using the stiffness assembly procedures outlined in section 3.3. Similarly, reduction of a model using free interface vibration modes leads to force interface DoF. The most natural assembly procedure is hence the flexibility assembly of section 3.4, like is done in the Dual Craig-Bampton method. In the same fashion the logical consequence of reduction using the Mixed Craig-Bampton method is to employ the mixed assembly method of the previous section.

In the Rubin and MacNeal methods however, the interface force DoF are transformed back to displacement DoF in order to enable stiffness assembly of the reduced model. This could be regarded as an artificial approach, which is confirmed by the fact that the resulting matrices lose their sparsity. Analogously, one could devise a reduction method that employs fixed interface vibration modes in which the natural interface displacement DoF are transformed to force DoF such that flexibility assembly can be employed. Obviously, like the Rubin and MacNeal method, this is unnatural and would lead to cumbersome reduced matrices.

The above discussion is summarized in figure 3.8, where the methods in light-grey circles indicate the natural combination of reduction basis ingredient and assembly procedure. The dark circles represent combinations that can be regarded as artificial.

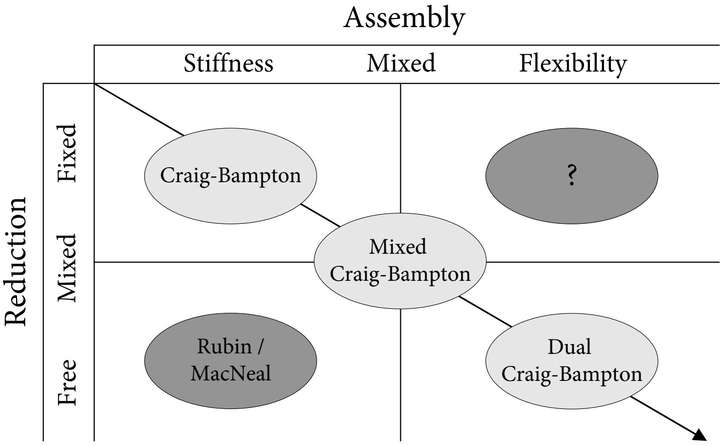


Figure 3.8: Relations between different reduction methods and assembly techniques.

3.6 Assembly of Non-Conforming Meshes

One of the benefits of dynamic substructuring is that it allows to combine substructure models created by different engineering groups. These models are often created without any knowledge of, or consideration for, the neighboring substructures. Since the models are meshed independently, it is likely that their meshes are incompatible. This means that the nodes at both sides of the interface are not collocated (i.e. at the same geometric position) and/or the models are meshed with different types of elements having non-matching shape functions. If this is the case, the meshes are said to be *non-conforming* and their assembly requires special procedures.

One approach would be to remesh the component models such that they become compatible. However, as explained in section 2.2, remeshing can be a tedious job that compromises the overall effectiveness of the dynamic substructuring strategy. Furthermore, sometimes a specific element type is needed to properly describe the structure and remeshing with different elements is not advisable.

A more efficient approach is therefore needed. One option is to use the interpolation functions of the interface elements in order to enable assembly of non-conforming substructure meshes. Based on the discussion in [171], the next subsections therefore treat the simple but effective *node collocation method* and its least squares variant.[3] Subsection 3.6.3 thereafter briefly addresses how to apply this method in the assembly procedures of the previous sections.

3.6.1 Node Collocation

For the sake of illustration suppose that two substructures are to be assembled, but the interfaces are not matching as depicted in figure 3.9. Obviously, due to the unequal number of non-collocated boundary nodes the the compatibility condition can in this case no longer be expressed using the Boolean matrices in eq. (3.3). Instead, a set of constraints is needed for each substructure that equals its number of boundary DoF. This can be achieved by using the intermediate interface field $\boldsymbol{u}_\gamma$ and its associated shape functions to interpolate and connect the nodes to the intermediate field at some n_γ DoF. This is illustrated in figure 3.9. The compatibility condition can now be expressed as:

$$\boldsymbol{u}_b^{(s)} = \boldsymbol{D}_b^{(s)} \boldsymbol{u}_\gamma \quad \text{for} \quad s = 1, 2 \tag{3.34}$$

where $\boldsymbol{D}_b^{(s)}$ is the substructure "collocation" matrix of size $n_b^{(s)} \times n_\gamma$. This collocation matrix is obtained by evaluating the interface field shape functions at the coordinates of each of the substructure boundary nodes. In order to uniquely define the displacements of the interface field by the substructure displacements, it is assumed that its number of DoF is lower or equal to the minimum number of substructure boundary DoF, that is:

$$n_\gamma \leq \min\left(n_b^{(1)}, n_b^{(2)}\right) \tag{3.35}$$

---

[3] Note that in the last two decades, the assembly of structural models with non-conforming discretizations has become a research field on its own. An important contribution is the so-called Mortar element method, see e.g. [22]. However, it is out of the scope of this work to treat these advanced methods.

However, this approach is cumbersome in practice as specific shape functions have to be defined for the interface field. Hence it would be desirable to express the interface field in terms of the existing boundary DoF fields, thereby eliminating the intermediate interface field.[4] One way to achieve this is by choosing the interface field $\boldsymbol{u}_\gamma$ equal to the coarsest substructure boundary field, i.e. the substructure with the minimum of the number of nodes on each interface. Note that $\boldsymbol{u}_\gamma$ could also be chosen equal to the finest interface but this would leave some DoF on this side unconnected, which might be undesirable in practice.

From figure 3.9 it becomes clear that in this case $\boldsymbol{u}_b^{(2)}$ are the *master* interface DoF while $\boldsymbol{u}_b^{(1)}$ are the *slave* interface DoF. As a result, $\boldsymbol{D}_b^{(2)}$ becomes an identity matrix and only the collocation matrix of substructure 1 needs to be computed, such that the compatibility condition can be written as:

$$\begin{aligned} \boldsymbol{u}_b^{(2)} &= \boldsymbol{u}_\gamma \\ \boldsymbol{u}_b^{(1)} &= \boldsymbol{D}_b^{(1)} \boldsymbol{u}_\gamma = \boldsymbol{D}_b^{(1)} \boldsymbol{u}_b^{(2)} \end{aligned} \tag{3.36}$$

The collocation matrix $\boldsymbol{D}_b^{(1)}$ contains the values of the boundary shape functions on the interface of substructure 2 at the positions of the interface nodes of substructure 1. This condition imposes that the nodes of substructure 1 remain on the interface of substructure 2, as illustrated in the bottom right figure in 3.9.

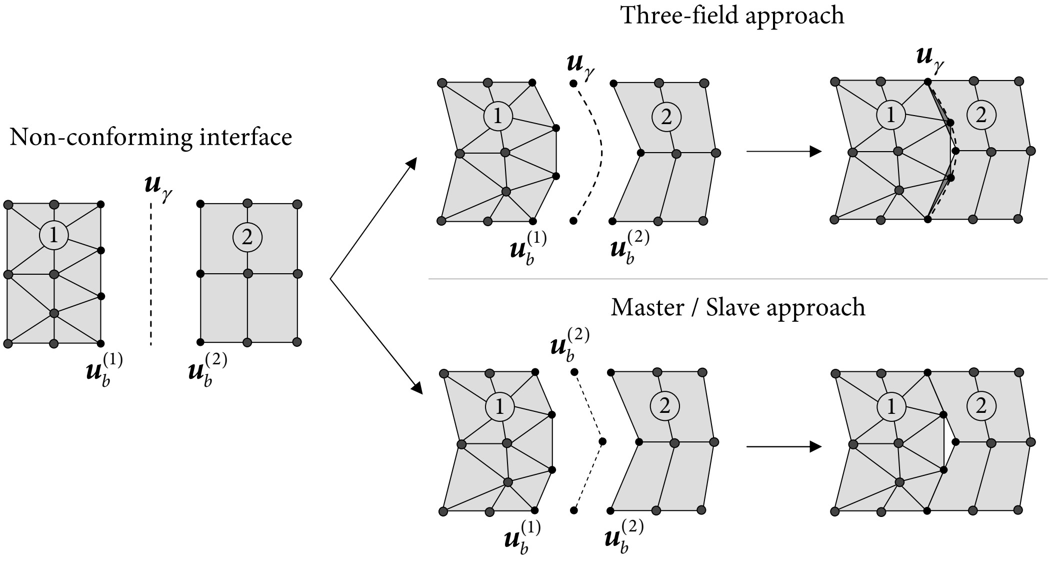


Figure 3.9: Two node-collocation approaches to handle non-conforming meshes on the interface.

However, both the general node collocation (eq. (3.34)) and the specific case where the interface field is chosen as the coarsest of the substructure interfaces (eq. (3.36)) restrict the number of interface DoF of the finest side, thereby stiffening its behavior. This might be undesirable, for instance if there is a large difference in the number of DoF at both sides of the interface. It might therefore be beneficial to render some flexibility to the interface; one way to achieve this is to relax the collocation constraints as explained next.

---

[4] The three-field formulation might however have some advantages for instance in formulating contact problems between non-matching grids [189].

3.6.2 Discrete Least Squares Compatibility

Suppose now that eq. (3.34) is not directly imposed, but some interface incompatibility is allowed. This means that eq. (3.34) can be used to compute the substructure displacements such that the incompatibility between both fields is minimized in terms of the L^2-norm. Hence it is wished to find:

$$\min_{\boldsymbol{u}_\gamma}\left(\left(\boldsymbol{u}^{(s)}-\boldsymbol{D}_b^{(s)}\boldsymbol{u}_\gamma\right)^T\left(\boldsymbol{u}^{(s)}-\boldsymbol{D}_b^{(s)}\boldsymbol{u}_\gamma\right)\right) \quad \text{for} \quad s=1,2, \tag{3.37}$$

which corresponds to

$$\frac{\partial}{\partial \boldsymbol{u}_\gamma}\left(\left(\boldsymbol{u}_b^{(s)}-\boldsymbol{D}_b^{(s)}\boldsymbol{u}_\gamma\right)^T\left(\boldsymbol{u}_b^{(s)}-\boldsymbol{D}_b^{(s)}\boldsymbol{u}_\gamma\right)\right)=0 \quad \text{for} \quad s=1,2 \tag{3.38}$$

Obviously, given condition (3.35), this is an overdetermined problem that can be solved in a least square sense. This gives results in:

$$\boldsymbol{u}_\gamma=\left(\boldsymbol{D}_b^{(s)^T}\boldsymbol{D}_b^{(s)}\right)^{-1}\boldsymbol{D}_b^{(s)^T}\boldsymbol{u}_b^{(s)}=\tilde{\boldsymbol{D}}_b^{(s)}\boldsymbol{u}_b^{(s)} \tag{3.39}$$

This expression gives the discrete least squares compatibility condition between the substructure boundary DoF and the intermediate interface field. As in case of the normal node collocation, the intermediate field can again be eliminated by choosing the coarsest interface as the master interface. This leads to the following compatibility condition between substructures 1 and 2:

$$\begin{aligned}&\boldsymbol{u}_\gamma=\boldsymbol{u}_b^{(2)}\\&\boldsymbol{u}_\gamma=\left(\boldsymbol{D}_b^{(1)^T}\boldsymbol{D}_b^{(1)}\right)^{-1}\boldsymbol{D}_b^{(1)^T}\boldsymbol{u}_b^{(1)} \quad \rightarrow \quad \boldsymbol{u}_b^{(2)}=\tilde{\boldsymbol{D}}_b^{(1)}\boldsymbol{u}_b^{(1)}\end{aligned} \tag{3.40}$$

where use was made of the fact that in this case $\boldsymbol{D}_b^{(2)}=\tilde{\boldsymbol{D}}_b^{(2)}=\boldsymbol{I}$. Clearly the discrete least squares approach gives less rigid connections between the substructure DoF, as the number of constraints imposed by (3.40) is now equal to the number of DoF on the coarsest interface, instead of the number of DoF of the finest side as in eq. (3.36). The compatibility condition in (3.40) therefore leads to a "best fit" in the least squares sense, by minimizing the interface incompatibility.

It should be noted that in both the node collocation and the discrete least square method only nodal compatibility is considered. Hence the substructures are only point-wise connected and the compatibility error along the interface between the nodes is disregarded. This could lead to bad overall compatibility for non-uniform and highly incompatible meshes and detailed results (e.g. stress concentrations) at the interface cannot be accurately predicted.

Alternatively, continuous compatibility methods could be used. These methods can be seen as evaluating eq. (3.37) by an integral over the boundary; the resulting $\boldsymbol{D}^{(s)}$ have the same form but are obtained by integration of the product of shape function of the substructure displacement and $\boldsymbol{u}_\gamma$ on the interface. In terms of subsequent algebraic treatment it therefore makes no difference whether discrete or continuous compatibility is considered. The reader is referred to [171] for a detailed discussion on continuous compatibility methods.

Nonetheless, discrete compatibility methods are still often used (also in many commercial software packages) since they are easy to implement and, following Saint-Venant's principle, will in general not significantly alter the global dynamic behavior.

3.6.3 Implementation in Assembly Procedures

The node collocation and discrete least squares compatibility can be implemented in a straightforward manner in the assembly procedures treated in the previous sections. For the stiffness and flexibility assembly cases this is done as described below, for the case of mixed assembly both approaches are simply combined.

Stiffness assembly In the case of stiffness assembly, the methods are easily implemented by replacing the local Boolean matrices $\boldsymbol{L}_b^{(s)}$ by the collocation matrices $\boldsymbol{D}_b^{(s)}$ or $\tilde{\boldsymbol{D}}_b^{(s)}$ in case least squares collocation is used. In both cases the collocation matrices can either be obtained using a general intermediate field as in eqs. (3.34) and (3.39), or by selecting a master interface as in eqs. (3.36) and (3.40). In the latter case, some of the collocation matrices are equal to identity.

Collecting the substructure collocation matrices $\boldsymbol{D}_b^{(s)}$ (or equivalently $\tilde{\boldsymbol{D}}_b^{(s)}$) in a block form as

$$\boldsymbol{D}_b = \mathrm{col}\left(\boldsymbol{D}_b^{(1)}; \ldots ; \boldsymbol{D}_b^{(n)}\right), \tag{3.41}$$

leads to a three-field formulation similar to eq. (3.5), namely:

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{ib} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{M}_{bi} & \boldsymbol{M}_{bb} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{u}}_b \\ \ddot{\boldsymbol{g}}_b \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{K}_{bi} & \boldsymbol{K}_{bb} & -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{D}_b \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D}_b^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \\ \boldsymbol{g}_b \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_b \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{3.42}$$

One can now proceed as in section 3.3 by simplifying the equations through dual or primal assembly. This requires a new signed matrix that is acting on the interface DoF $\bar{\boldsymbol{B}}_b$ which, like $\boldsymbol{D}_b$, is no longer Boolean. However, the relation between these matrices is still according to eq. (3.9) and hence $\bar{\boldsymbol{B}}_b$ can be computed as:

$$\bar{\boldsymbol{B}}_b = \mathrm{null}\left(\boldsymbol{D}_b\right)$$

The dual assembled equations are now obtained by choosing

$$\boldsymbol{g}_b = -\bar{\boldsymbol{B}}_b^T \boldsymbol{\lambda}$$

There are now as many Lagrange multipliers as there are DoF in $\boldsymbol{u}_\gamma$, hence one Lagrange multipliers can act between multiple boundary DoF of the same substructure. The intensities are however different since $\bar{\boldsymbol{B}}_b$ is no longer Boolean. The obtained equations of motion have the same form as eq. (3.11). In a similar fashion as in section 3.3 the primal system can be obtained, by choosing:

$$\boldsymbol{u}_b = \boldsymbol{D}_b \boldsymbol{u}_\gamma$$

Again, the obtained equations of motion look like eq. (3.14), with $\boldsymbol{L}_b$ replaced by $\boldsymbol{D}_b$.

Flexibility assembly Implementation of the nodal collocation methods in the case of flexibility assembly is equally straightforward. In the first step, the Boolean matrices are replaced by the collocation matrices to find the following equivalent of eq. (3.21):

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{ii} & \bar{\boldsymbol{M}}_{ib} & \boldsymbol{0} \\ \bar{\boldsymbol{M}}_{bi} & \bar{\boldsymbol{M}}_{bb} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\bar{\boldsymbol{q}}}_i \\ \ddot{\boldsymbol{g}}_b \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{K}}_{ii} & \bar{\boldsymbol{K}}_{ib} & \boldsymbol{0} \\ \bar{\boldsymbol{K}}_{bi} & \bar{\boldsymbol{K}}_{bb} & \boldsymbol{D}_b \\ \boldsymbol{0} & \boldsymbol{D}_b^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{q}}_i \\ \boldsymbol{g}_b \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_i \\ \bar{\boldsymbol{f}}_b \\ \boldsymbol{0} \end{bmatrix} \tag{3.43}$$

As explained in section 3.4, these equations can only be further simplified in a primal manner. This is again done by choosing the interface forces as:

$$\boldsymbol{g}_b = -\bar{\boldsymbol{B}}_b^T \boldsymbol{\lambda},$$

and assembled equations are found that are similar to eq. (3.27).

3.7 Assembly with Additional Interface Physics

The assembly procedures outlined above all assume a perfect connection between the substructure boundary DoF. For many substructures this however does not hold true. Consider for example two components that are connected by a bolt, a situation encountered very often in practice. Due to this connection some flexibility and/or damping is introduced on the interface that is not present in the separate components. Many other examples of connections are imaginable where some physics are added to the system simply through the coupling of components. Often these interface effects are simply neglected. However, in case the interface physics introduce significant flexibility or inertia to the system this simplification cannot be made.

In this section a method is therefore proposed which allows easy incorporation of these interface effects in the assembly procedure by changing the imposed interface conditions. Next, it will be explained how additional interface stiffness or inertia can be accounted for in both the stiffness and flexibility cases. In order to incorporate interface physics in the case of mixed assembly, both approaches can be simply combined.

3.7.1 Stiffness Assembly with Additional Interface Physics

3.7.1.1 Stiffness Assembly with Interface Stiffness

In order to explain the treatment of additional interface physics for the case of stiffness assembly, first recall the three-field formulation of section 3.3:

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{ib} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{M}_{bi} & \boldsymbol{M}_{bb} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{u}}_b \\ \ddot{\boldsymbol{g}}_b \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{K}_{bi} & \boldsymbol{K}_{bb} & -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{L}_b \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}_b^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \\ \boldsymbol{g}_b \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_b \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \qquad (3.5)$$

Suppose now that the boundary DoF are not perfectly connected, but assembled through some interface stiffness $\boldsymbol{\kappa}_{bb}$. This situation is illustrated in figure 3.10(a) for the assembly of two components. Due to the presence of this interface stiffness the interface DoF are free to have a relative displacement and the compatibility condition between these DoF no longer holds. Hence, the third equation in the above expression cancels which eliminates the intermediate field $\boldsymbol{u}_\gamma$. Due to the construction of the Boolean matrix $\boldsymbol{B}_b$, the relative displacements of the interface DoF can be expressed as:

$$\Delta \boldsymbol{u}_b = \boldsymbol{B}_b \boldsymbol{u}_b \qquad (3.44)$$

Furthermore, the interface forces associated to the "flexible" boundary DoF can be chosen to be expressed as:

$$\boldsymbol{g}_b = -\boldsymbol{B}_b^T \boldsymbol{\lambda} \tag{3.45}$$

In this way the interface forces in $\boldsymbol{g}_b$ are chosen such that, due to the construction of the Boolean matrix, the interface forces are always equal and opposite. As before, the Lagrange multipliers $\boldsymbol{\lambda}$ are the force intensities. Indeed, due to the properties of the Boolean matrices the equilibrium condition is always satisfied:

$$\boldsymbol{L}_b^T \boldsymbol{g}_b = -\boldsymbol{L}_b^T \boldsymbol{B}_b^T \boldsymbol{\lambda} = \boldsymbol{0} \tag{3.46}$$

Next it should be realized that the interface forces result from the deformation of the interface stiffness $\boldsymbol{\kappa}_{bb}$. Hence the interface force intensities can be expressed as:

$$\boldsymbol{\lambda} = \boldsymbol{\kappa}_{bb} \Delta \boldsymbol{u}_b = \boldsymbol{\kappa}_{bb} \boldsymbol{B}_b \boldsymbol{u}_b \tag{3.47}$$

The interface connection forces are therefore found as:

$$\boldsymbol{g}_b = -\boldsymbol{B}_b^T \boldsymbol{\kappa}_{bb} \boldsymbol{B}_b \boldsymbol{u}_b \tag{3.48}$$

Note that the interface stiffness $\boldsymbol{\kappa}_{bb}$ contains the stiffness of the interface springs and is in general a diagonal matrix. As such, $\boldsymbol{\kappa}_{bb}$ is not like a regular stiffness matrix, but becomes one after pre- and postmultiplication by $\boldsymbol{B}_b$. This can be illustrated by the simplest example where two boundary DoF $u_b^{(1)}$ and $u_b^{(2)}$ are connected through stiffness k_b. In that case one finds:

$$\boldsymbol{B}_b = \begin{bmatrix} 1 & -1 \end{bmatrix} \quad \rightarrow \quad \boldsymbol{B}_b^T k_b \boldsymbol{B}_b \boldsymbol{u}_b = \begin{bmatrix} k_b & -k_b \\ -k_b & k_b \end{bmatrix} \begin{bmatrix} u_b^{(1)} \\ u_b^{(2)} \end{bmatrix}$$

With the expression for the connection forces in eq. (3.48), the following assembled equations of motion are found:[5]

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{ib} \\ \boldsymbol{M}_{bi} & \boldsymbol{M}_{bb} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib} \\ \boldsymbol{K}_{bi} & \boldsymbol{K}_{bb} + \boldsymbol{B}_b^T \boldsymbol{\kappa}_{bb} \boldsymbol{B}_b \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_b \end{bmatrix} \tag{3.49}$$

A few remarks are in place regarding these assembled equations:

- Due to the action of the interface stiffness the substructures are automatically assembled. In other words, there is no longer any choice whether to perform the assembly in a dual or primal way.
- Interface damping effects can be taken into account in the exact same manner. In that case, the expression for the interface connection forces becomes

  $$\boldsymbol{g}_b = -\boldsymbol{B}_b^T \boldsymbol{\kappa}_{bb} \boldsymbol{B}_b \boldsymbol{u}_b - \boldsymbol{B}_b^T \boldsymbol{\delta}_{bb} \boldsymbol{B}_b \dot{\boldsymbol{u}}_b, \tag{3.50}$$

  with $\boldsymbol{\delta}_{bb}$ the interface damping matrix, and the assembled damping matrix has the same topology as the stiffness matrix. In [159], this formulation was applied to include non-linear interface damping arising from friction effects.

---

[5] Note that this result can also be obtained directly from the Lagrangian in eq. (3.4), by disregarding the intermediate interface field $\boldsymbol{u}_\gamma$ and including the interface stiffness $\boldsymbol{\kappa}_{bb}$ and taking the variation.

- In (3.47), by stating that $\boldsymbol{\kappa}_{bb}$ is a diagonal matrix, it is implicitly assumed that the force expressed by a Lagrange multiplier is related only to the interface incompatibility seen by that multiplier. So, in other words, one can create coupling only between corresponding DoF across the interface, and not represent a coupled stiffness/damping network on the interface. In the latter case the interface physics would need to be incorporated as a true "third component" between two substructure interfaces. However, as long as the interface springs/dampers are connected in the same way as the interface DoF, the formulation in this section is considered a simpler and more elegant approach.

- In case the interface includes dynamics, it is no longer correct to assume that the forces on each side of the interface are equal as in eq. (3.45). In the next subsection it is therefore explained how to include additional interface inertia.

- In the above discussion, interface stiffness was assumed to act between all boundary DoF. In case not all boundary DoF are affected by the interface stiffness but are perfectly connected instead, one can a) increase the stiffness for those DoF in $\boldsymbol{\kappa}_{bb}$ to a value that is orders of magnitude higher than the corresponding entries in $\boldsymbol{K}_{bb}$ or b) partition the DoF set into a set of "perfect" boundary DoF $\boldsymbol{u}_b$ and a set of "flexible" DoF $\boldsymbol{u}_f$. As the former approach leads to bad matrix conditioning the latter approach is preferable, and is illustrated in figure 3.10 (a). Partitioning the boundary DoF and applying the procedure in this section to only the "flexible" DoF then gives the following assembled matrices:

$$\cdots + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ip} & \boldsymbol{K}_{if} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{K}_{pi} & \boldsymbol{K}_{pp} & \boldsymbol{K}_{pf} & -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{K}_{fi} & \boldsymbol{K}_{fp} & \boldsymbol{K}_{ff} + \boldsymbol{B}_f^T \boldsymbol{\kappa}_{bb} \boldsymbol{B}_f & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}_p \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}_p^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_p \\ \boldsymbol{u}_f \\ \boldsymbol{g}_p \\ \boldsymbol{u}_{y,p} \end{bmatrix} \tag{3.51}$$

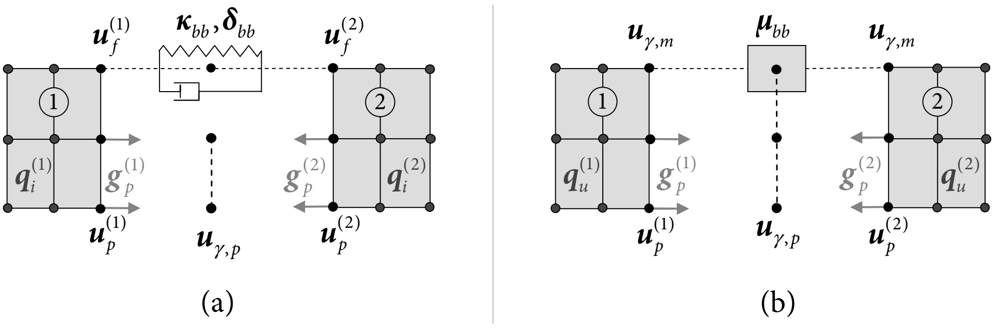
  Here the mass matrix is omitted for compactness. Note that for the "perfect" DoF $\boldsymbol{u}_p$ a choice needs still to be made as to assemble them in a dual or primal fashion, as described in sections 3.3.1 and 3.3.2.

- The assembly of systems with perfect connections can be regarded as a special case of the above situation, namely when $\|\boldsymbol{\kappa}_{bb}\| \rightarrow \infty$. Then $\Delta \boldsymbol{u}_b = \boldsymbol{B}_b \boldsymbol{u}_b \rightarrow \boldsymbol{0}$ and the compatibility condition indeed holds.

- The same formulation is found when one wants to enforce compatibility with a penalty method; the interface stiffness is then the penalty. See for instance [171].

3.7.1.2 Stiffness Assembly with Interface Inertia

In a way similar to the previous subsection, additional interface inertia $\boldsymbol{\mu}_{bb}$ can be incorporated in the assembled equations of motion. This is illustrated in figure 3.10 (b). In this case the reasoning is as follows. Due to the presence of additional inertia the force equilibrium



(a) (b)

Figure 3.10: Assembly of components with additional interface physics in the form of stiffness and damping (a) and additional inertia (b).

between the interface DoF no longer holds. Instead, a resulting force exists on the interface which can be expressed as:

$$\Delta \boldsymbol{g}_b = \boldsymbol{L}_b^T \boldsymbol{g}_b \tag{3.52}$$

This force leads to acceleration of the additional inertia at the intermediate interface DoF:

$$\boldsymbol{\mu}_{bb} \ddot{\boldsymbol{u}}_\gamma = -\Delta \boldsymbol{g}_b = -\boldsymbol{L}_b^T \boldsymbol{g}_b \tag{3.53}$$

The minus sign originates from the fact that the $\boldsymbol{g}_b$ are the forces applied on the substructures, hence the additional interface inertia experiences these forces in the opposite direction. Next it is realized that, since only inertia is added between the components, the compatibility condition still holds:

$$\boldsymbol{u}_b = \boldsymbol{L}_b \boldsymbol{u}_\gamma \tag{3.54}$$

Substitution of expressions (3.53) and (3.54) in the three-field formulation of eq. (3.5), and considering the primal assembly process that led to (3.14), leads to the following assembled equations:

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{ib}\boldsymbol{L}_b \\ \boldsymbol{L}_b^T \boldsymbol{M}_{bi} & \boldsymbol{L}_b^T \boldsymbol{M}_{bb} \boldsymbol{L}_b + \boldsymbol{\mu}_{bb} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib}\boldsymbol{L}_b \\ \boldsymbol{L}_b^T \boldsymbol{K}_{bi} & \boldsymbol{L}_b^T \boldsymbol{K}_{bb} \boldsymbol{L}_b \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{L}_b^T \boldsymbol{f}_b \end{bmatrix} \tag{3.55}$$

Clearly, the interface inertia is added to the sum of the substructure inertia's at the interface. Note that in this case one automatically ends up with a primal assembled system. The additional inertia violates the equilibrium condition, such that dual assembly using Lagrange multipliers becomes impossible. However, since the compatibility condition still holds the redundant interface DoF can be eliminated through primal assembly. As was remarked for the case of additional interface stiffness, the interface DoF $\boldsymbol{u}_b$ can easily be split into a part $\boldsymbol{u}_m$ where additional inertia is present and a part $\boldsymbol{u}_p$ that are perfectly connected. For the latter, one still has a choice for further simplification through dual or primal assembly.

In case both interface stiffness and inertia need to be taken into account, two situations can be encountered:

- When the interface stiffness and inertia act at disjoint sets of boundary DoF (i.e. an interface DoF feels either additional stiffness or mass), the approach of this and the previous subsection can be individually applied to the associated DoF sets.

- In case interface stiffness and inertia act simultaneously between some interface DoF, both the compatibility and equilibrium condition can no longer be satisfied at these DoF. Indeed, this amounts to assembly of the interface physics as an intermediate substructure.

3.7.2 Flexibility Assembly with Additional Interface Physics

3.7.2.1 Flexibility Assembly with Interface Stiffness

To account for the presence of additional interface stiffness in the case of flexibility assembly, recall the assembled system obtained in section 3.4:

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{ii} & \bar{\boldsymbol{M}}_{ib} & \boldsymbol{0} \\ \bar{\boldsymbol{M}}_{bi} & \bar{\boldsymbol{M}}_{bb} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\bar{\boldsymbol{q}}}_i \\ \ddot{\boldsymbol{g}}_b \\ \ddot{\boldsymbol{u}}_y \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{K}}_{ii} & \bar{\boldsymbol{K}}_{ib} & \boldsymbol{0} \\ \bar{\boldsymbol{K}}_{bi} & \bar{\boldsymbol{K}}_{bb} & \boldsymbol{L}_b \\ \boldsymbol{0} & \boldsymbol{L}_b^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{q}}_i \\ \boldsymbol{g}_b \\ \boldsymbol{u}_y \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_i \\ \bar{\boldsymbol{f}}_b \\ \boldsymbol{0} \end{bmatrix} \tag{3.21}$$

Again it is assumed that the boundary DoF are not perfectly connected, but are coupled through some interface stiffness $\boldsymbol{\kappa}_{bb}$. To account for this stiffness it should again be realized that the compatibility condition no longer holds. As a result, it is no longer correct to state that there exists a unique set of interface DoF such that $\boldsymbol{u}_b = \boldsymbol{L}_b \boldsymbol{u}_y$, but one must keep all $\boldsymbol{u}_b$ as independent variables. The above expression thus becomes:

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{ii} & \bar{\boldsymbol{M}}_{ib} & \boldsymbol{0} \\ \bar{\boldsymbol{M}}_{bi} & \bar{\boldsymbol{M}}_{bb} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\bar{\boldsymbol{q}}}_i \\ \ddot{\boldsymbol{g}}_b \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{K}}_{ii} & \bar{\boldsymbol{K}}_{ib} & \boldsymbol{0} \\ \bar{\boldsymbol{K}}_{bi} & \bar{\boldsymbol{K}}_{bb} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{L}_b^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{q}}_i \\ \boldsymbol{g}_b \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_i \\ \bar{\boldsymbol{f}}_b \\ \boldsymbol{0} \end{bmatrix} \tag{3.56}$$

Next, following the reasoning of the previous section, it is observed that the equilibrium condition can still be satisfied by choosing:

$$\boldsymbol{g}_b = -\boldsymbol{B}_b^T \boldsymbol{\lambda} \tag{3.57}$$

Introducing this relation in (3.56) cancels the third line and multiplying the second line by $-\boldsymbol{B}_b$, similar to what was done in the dual assembly to obtain (3.27), then gives:

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{ii} & -\bar{\boldsymbol{M}}_{ib}\boldsymbol{B}_b^T \\ -\boldsymbol{B}_b\bar{\boldsymbol{M}}_{bi} & \boldsymbol{B}_b\bar{\boldsymbol{M}}_{bb}\boldsymbol{B}_b^T \end{bmatrix} \begin{bmatrix} \ddot{\bar{\boldsymbol{q}}}_i \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{K}}_{ii} & -\bar{\boldsymbol{K}}_{ib}\boldsymbol{B}_b^T \\ -\boldsymbol{B}_b\bar{\boldsymbol{K}}_{bi} & \boldsymbol{B}_b\bar{\boldsymbol{K}}_{bb}\boldsymbol{B}_b^T \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{q}}_i \\ \boldsymbol{\lambda} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{B}_b\boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_i \\ -\boldsymbol{B}_b\bar{\boldsymbol{f}}_b \end{bmatrix} \tag{3.58}$$

Like in the previous section for the stiffness assembly, the interface forces $\boldsymbol{\lambda}$ are caused by the relative interface displacements $\Delta\boldsymbol{u}_b = \boldsymbol{B}_b\boldsymbol{u}_b$ through the stiffness between every pair of DoF related by the Lagrange multipliers:

$$\boldsymbol{\lambda} = \boldsymbol{\kappa}_{bb}\boldsymbol{B}_b\boldsymbol{u}_b \quad \text{or} \quad \boldsymbol{B}_b\boldsymbol{u}_b = \boldsymbol{\kappa}_{bb}^{-1}\boldsymbol{\lambda} \tag{3.59}$$

This indicates that the interface gap is generated by the $\boldsymbol{\lambda}$ over the interface *flexibility*. Replacing this relation in (3.58) yields the final result

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{ii} & -\bar{\boldsymbol{M}}_{ib}\boldsymbol{B}_b^T \\ -\boldsymbol{B}_b\bar{\boldsymbol{M}}_{bi} & \boldsymbol{B}_b\bar{\boldsymbol{M}}_{bb}\boldsymbol{B}_b^T \end{bmatrix} \begin{bmatrix} \ddot{\bar{\boldsymbol{q}}}_i \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{K}}_{ii} & -\bar{\boldsymbol{K}}_{ib}\boldsymbol{B}_b^T \\ -\boldsymbol{B}_b\bar{\boldsymbol{K}}_{bi} & \boldsymbol{B}_b\bar{\boldsymbol{K}}_{bb}\boldsymbol{B}_b^T + \boldsymbol{\kappa}_{bb}^{-1} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{q}}_i \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_i \\ -\boldsymbol{B}_b\bar{\boldsymbol{f}}_b \end{bmatrix} \tag{3.60}$$

So, the additional interface flexibility is simply added to the assembled flexibility of the substructures at the interface DoF (recall that $\bar{\boldsymbol{K}}_{bb}$ is an interface flexibility matrix). Note the duality with adding the interface inertia in the stiffness assembly in (3.55); in both cases one ends up with a primal assembled set of equations.

Of course, as remarked before, the above technique can also be applied to a subset of interface DoF in case additional stiffness acts only on part of the interface(s). Furthermore it is remarked that it is probably possible to also include additional interface damping in the case of flexibility assembly, although it is not directly clear how. This is beyond the scope of this work.

3.7.2.2 Flexibility Assembly with Interface Inertia

Finally, this subsection considers the incorporation of additional interface inertia in the case of flexibility assembly. As before, due to the presence of the interface inertia, the interface forces $\boldsymbol{g}_b$ are no longer satisfying the action-reaction equilibrium, but a resulting force will be created as expressed by eq. (3.53):

$$\boldsymbol{\mu}_{bb}\ddot{\boldsymbol{u}}_\gamma = -\Delta\boldsymbol{g}_b = -\boldsymbol{L}_b^T\boldsymbol{g}_b \tag{3.53}$$

Replacing then the last line of eq. (3.21) by this dynamic equilibrium directly yields the equations for the assembled system with additional interface inertia:

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{ii} & \bar{\boldsymbol{M}}_{ib} & \boldsymbol{0} \\ \bar{\boldsymbol{M}}_{bi} & \bar{\boldsymbol{M}}_{bb} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mu}_{bb} \end{bmatrix} \begin{bmatrix} \ddot{\bar{\boldsymbol{q}}}_i \\ \ddot{\boldsymbol{g}}_b \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{K}}_{ii} & \bar{\boldsymbol{K}}_{ib} & \boldsymbol{0} \\ \bar{\boldsymbol{K}}_{bi} & \bar{\boldsymbol{K}}_{bb} & \boldsymbol{L}_b \\ \boldsymbol{0} & \boldsymbol{L}_b^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{q}}_i \\ \boldsymbol{g}_b \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_i \\ \bar{\boldsymbol{f}}_b \\ \boldsymbol{0} \end{bmatrix} \tag{3.61}$$

3.8 Decoupling of Component Models

So far, this chapter considered techniques to *couple* substructure models to obtain the total system model. However, sometimes one has to consider the reverse problem, namely how a substructure model can be found from the assembled system by *decoupling*. Obviously this is only relevant for components that cannot be modeled numerically, for instance due to viscoelasticity (i.e. frequency dependent behavior), high damping, too complex geometry, high modal density, etc. In that case an experimental model can be created from measurements. Sometimes a component cannot be tested separately, but only when coupled to their neighboring substructure(s) (e.g. a fixture needed for testing). This can for example be the case for subsystems that are very delicate or in operational conditions [99]. In these cases decoupling of the component from neighboring substructures becomes a relevant issue.

Although substructure decoupling techniques are not the main focus of this thesis, this section takes a sidestep and briefly addresses a number of developments in this field. To this end, the next subsection briefly describes the decoupling problem, which will be formalized in subsection 3.8.2 using a dual (dis)assembly approach. Thereafter, subsection 3.8.3 presents a framework for decoupling methods. Note that the discussion in this section is a concise version of the work presented in [211, 212, 213], more details can be found in these publications.

3.8.1 Problem Description

To illustrate the problem at hand, consider subsystems A and C shown in figure 3.11 (a); when assembled they form assembly AC. In a dynamic substructuring analysis, the dynamics of AC are obtained by coupling the dynamic models of A and C. In substructure decoupling, the reverse problem is solved. In this case, it is assumed that the models of the assembly AC and the substructure A are known. Based on this information, the aim is to find the dynamics of component C as a "stand alone" component, that is, completely decoupled from A.

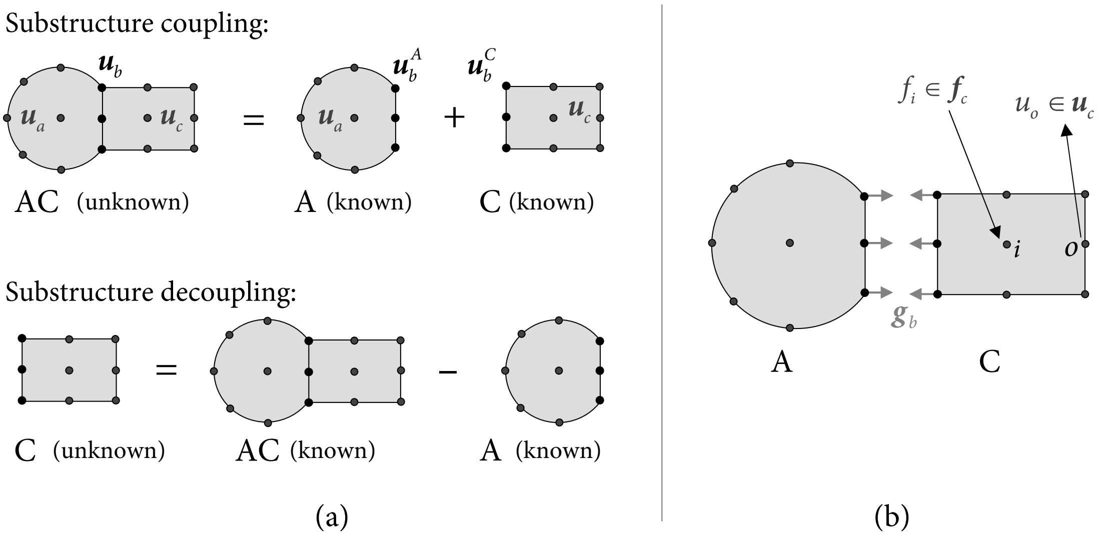


Figure 3.11: Substructure coupling vs. decoupling (a); finding the uncoupled response of C (b).

Practical applications of substructure decoupling can be imagined in structural monitoring and vibration control for critical components in an assembly. However, as outlined in [47], a number of challenges remain in the practical implementation of decoupling techniques. One important issue is the sensitivity of decoupling techniques to small (measurement) errors, especially around the anti-resonances of the known component [192].[6] In this thesis the decoupling problem is formulated in a general framework, using a dual (dis)assembly approach. This framework can then be used to show the relations between existing decoupling techniques and allows deriving new, possibly more robust techniques.

To illustrate the problem of substructure decoupling more thoroughly, the situation depicted in figure 3.11 (b) is considered. For the sake of illustration, suppose one is interested in the response u_o of component C at degree of freedom (DoF) o, due to excitation f_i at DoF i, without the influence of neighboring subsystem A. Both DoF are internal to subsystem C and hence part of $\boldsymbol{u}_c$. In general, the decoupling problem can now be described as follows:

- The force f_i excites the system AC at DoF i internal to component C. As a result, the assembled system AC shows a response $\boldsymbol{u}^{AC}$.

---

[6] In substructure decoupling the sensitivity is highest around the *anti-resonance* frequencies of the known subsystem(s), while in substructure coupling the sensitivities are highest around the subsystems *resonances* [209].

- Now take only the part of the response of *AC* associated to component *C* and realize that in addition to the excitation force f_i, subsystem *C* in the assembly *AC* is also subjected to connection forces of component *A*.
- Additional forces opposing these connection forces should thus be applied to the assembly *AC* in order to let *C* behave without "feeling" the influence of *A*.
- Using the dynamic model of uncoupled system *A*, one can determine these connection forces loads by imposing to the uncoupled model the coupled responses of subsystem *A*.

Summarizing, one can formulate the decoupling problem as finding the behavior of substructure *C* as part of the assembled system *AC* when additional forces are applied at the interface such that substructure *C* experiences no connection forces from subsystem *A*. Hence, substructure *C* behaves as if it were decoupled from *A*.

3.8.2 Dual Formulation of the Decoupling Problem

The decoupling procedure outlined in the previous section will be formalized mathematically in this section. As mentioned above, the dynamics of the assembled system *AC* and component *A* are known and assumed to be expressed in the frequency domain. A systematic approach can be taken when starting from a dynamic stiffness representation of the subsystems. In general, the dynamic stiffness matrix can be obtained from the structural matrices as follows:

$$\boldsymbol{Z}(\omega) = \boldsymbol{K} + i\omega^2 \boldsymbol{C} - \omega^2 \boldsymbol{M} \tag{3.62}$$

Here ω denotes the frequency dependence. Hence, the assembled system *AC* is described as

$$\begin{aligned} \boldsymbol{Z}^{AC}\boldsymbol{u}^{AC} &= \boldsymbol{f}^{AC} + \boldsymbol{g}^{AC} \\ \begin{bmatrix} \boldsymbol{Z}^A_{aa} & \boldsymbol{Z}^A_{ab} & \boldsymbol{0} \\ \boldsymbol{Z}^A_{ba} & \boldsymbol{Z}^A_{bb} + \boldsymbol{Z}^B_{bb} & \boldsymbol{Z}^C_{bc} \\ \boldsymbol{0} & \boldsymbol{Z}^C_{cb} & \boldsymbol{Z}^C_{cc} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}_b \\ \boldsymbol{u}_c \end{bmatrix} &= \begin{bmatrix} \boldsymbol{f}_a \\ \boldsymbol{f}_b \\ \boldsymbol{f}_c \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_b \\ \boldsymbol{0} \end{bmatrix} \end{aligned} \tag{3.63}$$

For the sake of illustration it is assumed that $\boldsymbol{Z}^{AC}$ originates from a primal assembly of the substructures, as can be seen from comparison with eq. (3.14) (with identity Boolean matrices). Similarly the dynamic stiffness of subsystem *A* writes:

$$\begin{aligned} \boldsymbol{Z}^{A}\boldsymbol{u}^{A} &= \boldsymbol{f}^{A} - \boldsymbol{g}^{A} \\ \begin{bmatrix} \boldsymbol{Z}^A_{aa} & \boldsymbol{Z}^A_{ab} \\ \boldsymbol{Z}^A_{ba} & \boldsymbol{Z}^A_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}^A_c \end{bmatrix} &= \begin{bmatrix} \boldsymbol{f}_a \\ \boldsymbol{f}_c \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}^A_c \end{bmatrix}. \end{aligned} \tag{3.64}$$

Here $\boldsymbol{g}^{\star}$ now represents the *disconnection* forces felt from the coupling/decoupling of the neighboring components. Since the aim is to decouple component *A* from *AC*, the signs of these connection forces should be opposite as explained in section 3.8.1. The subscripts $_a$, $_c$ and $_b$ denote "internal to subsystem *A*", internal to subsystem *C* and "boundary", respectively;

the superscripts A, C and AC denote the two subsystems and the assembled system. The explicit frequency dependence is omitted for clarity.

Just as in the case of substructure coupling two interface conditions must be satisfied when decoupling, namely the compatibility and equilibrium conditions. Firstly, the compatibility condition governs the compatibility of of connected interface DoF. In this case:

$$\boldsymbol{u}_b = \boldsymbol{u}_b^A \tag{3.65}$$

Secondly, the equilibrium condition imposes that the connection forces between the substructures should be in equilibrium, i.e.:

$$\boldsymbol{g}_b + \boldsymbol{g}_b^A = \boldsymbol{0} \tag{3.66}$$

Similar to the procedures in sections 3.3 to 3.5, two Boolean matrices (assuming the interfaces are conforming) can be introduced to allow a systematic description of the problem. The first is the signed Boolean matrix $\boldsymbol{B}$, operating on the substructure interface degrees of freedom.[7] Using this Boolean matrix, the compatibility condition can be conveniently expressed as

$$\boldsymbol{B}\boldsymbol{u} = \begin{bmatrix} \boldsymbol{B}^{AC} & \boldsymbol{B}^{A} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{AC} \\ \boldsymbol{u}^{A} \end{bmatrix} = \boldsymbol{u}_b - \boldsymbol{u}_b^A = \boldsymbol{0}. \tag{3.67}$$

The second Boolean matrix $\boldsymbol{L}$ localizes the interface DoF of the substructures in the global set of DoF and is similar to the localization matrices used in the assembly of individual elements in finite element models. In this case, the equilibrium condition can be stated as:[8]

$$\boldsymbol{L}^T\boldsymbol{g} = \begin{bmatrix} \boldsymbol{L}^{AC^T} & \boldsymbol{L}^{A^T} \end{bmatrix} \begin{bmatrix} \boldsymbol{g}^{AC} \\ \boldsymbol{g}^{A} \end{bmatrix} = \boldsymbol{0} \quad \rightarrow \quad \boldsymbol{g}_b + \boldsymbol{g}_b^A = \boldsymbol{0} \tag{3.68}$$

As was noted in section 3.3 in eq. (3.9), $\boldsymbol{L}$ actually represents the nullspace of $\boldsymbol{B}$ or vice versa.

Equations (3.63), (3.64), (3.67) and (3.68) now fully describe the decoupling problem:

$$\begin{cases} \begin{bmatrix} \boldsymbol{Z}^{AC} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}^{A} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{AC} \\ \boldsymbol{u}^{A} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{AC} \\ \boldsymbol{f}^{A} \end{bmatrix} + \begin{bmatrix} \boldsymbol{g}^{AC} \\ -\boldsymbol{g}^{A} \end{bmatrix} \\ \begin{bmatrix} \boldsymbol{B}^{AC} & \boldsymbol{B}^{A} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{AC} \\ \boldsymbol{u}^{A} \end{bmatrix} = \boldsymbol{0} \\ \begin{bmatrix} \boldsymbol{L}^{AC^T} & \boldsymbol{L}^{A^T} \end{bmatrix} \begin{bmatrix} \boldsymbol{g}^{AC} \\ \boldsymbol{g}^{A} \end{bmatrix} = \boldsymbol{0} \end{cases} \tag{3.69}$$

These equations are the two-field formulation of the decoupling problem, which bears strong resemblance to the three-field form of the coupling problem of eq. (3.5). The main difference is that in the above equations the intermediate field $\boldsymbol{u}_\gamma$ is absent, which can likewise be eliminated from eq. (3.5) by multiplying its third row by $\boldsymbol{B}_b$.[9]

---

[7] Whereas in the previous sections this Boolean matrix was defined on the interface DoF only ($\boldsymbol{B}_b$), here $\boldsymbol{B}$ is defined to act on all substructure DoF. Both Booleans are simply related as $\boldsymbol{B} = [\boldsymbol{0} \quad \boldsymbol{B}_b]$, were the zeros correspond to the substructure's internal DoF.

[8] Similar to the $\boldsymbol{B}$ matrix, the localization matrix $\boldsymbol{L}$ is in this section defined to act on all substructure DoF in contrast to the previously used $\boldsymbol{L}_b$. These matrices are simply related through $\boldsymbol{L} = [\boldsymbol{I} \quad \boldsymbol{L}_b]$, with the identity matrix corresponding to the internal DoF.

[9] The decoupling problem can also be formulated in a three-field form. Like in the case of substructure coupling, this is however only useful for *mixed decoupling* problems, i.e. when a component expressed in stiffness terms is subtracted from an assembly expressed in flexibility or vice versa. This is not very relevant since measured systems are generally in terms of flexibility matrices and hence the mixed decoupling problem is not considered here.

Starting from equation (3.69), one can proceed to solve the decoupling problem in a primal or dual manner exactly like in section 3.3. In this section a dual approach will be taken, since it was shown in [52, 53] to be the most natural when dealing with experimental data and avoids double inversion of the (measured) FRF matrices as encountered in primal assembly. As in section 3.3.1, dual assembly is achieved by choosing the interface forces in the form:

$$\boldsymbol{g} = -\boldsymbol{B}^T\boldsymbol{\lambda} \tag{3.70}$$

Here, $\boldsymbol{\lambda}$ are again Lagrange multipliers corresponding to the interface force intensities. Due to the construction of the Boolean matrices, this choice satisfies the equilibrium condition and the decoupling problem can be formulated in a dual way as:

$$\begin{cases} \boldsymbol{Z}^{AC}\boldsymbol{u}^{AC} + \boldsymbol{B}^{AC^T}\boldsymbol{\lambda} = \boldsymbol{f}^{AC} \\ \boldsymbol{Z}^{A}\boldsymbol{u}^{A} - \boldsymbol{B}^{A^T}\boldsymbol{\lambda} = \boldsymbol{f}^{A} \\ \boldsymbol{B}^{AC}\boldsymbol{u}^{AC} + \boldsymbol{B}^{A}\boldsymbol{u}^{A} = \boldsymbol{0} \end{cases} \tag{3.71}$$

One can transform this to matrix vector notation as:

$$\begin{bmatrix} \boldsymbol{Z}^{AC} & \boldsymbol{0} & \boldsymbol{B}^{AC^T} \\ \boldsymbol{0} & -\boldsymbol{Z}^{A} & \boldsymbol{B}^{A^T} \\ \boldsymbol{B}^{AC} & \boldsymbol{B}^{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{AC} \\ \boldsymbol{u}^{A} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{AC} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{3.72}$$

The above equation has been symmetrized by multiplying the second equation by minus 1. Note that since component A is not of particular interest here it is assumed that, for the sake of simplicity, $\boldsymbol{f}^A = \boldsymbol{0}$. This last relation clearly shows that the decoupling of a subsystem is equivalent to a dual assembly of a negative dynamic stiffness for the substructure that one wants to subtract (here substructure A). Note that this is sometimes referred to as “fictitious domain substructuring” [52, 48]. The actual uncoupled FRFs of C can now be found by eliminating the Lagrange multipliers. At first, start by writing explicitly the substructure DoF as:

$$\boldsymbol{u}^{AC} = \boldsymbol{Z}^{AC^{-1}}\left(\boldsymbol{f}^{AC} - \boldsymbol{B}^{AC^T}\boldsymbol{\lambda}\right) \tag{3.73}$$

$$\boldsymbol{u}^{A} = \boldsymbol{Z}^{A^{-1}}\boldsymbol{B}^{A^T}\boldsymbol{\lambda} \tag{3.74}$$

Substitution in the compatibility condition and solving for $\boldsymbol{\lambda}$ gives:

$$\boldsymbol{\lambda} = \left(\boldsymbol{B}^{AC}\boldsymbol{Z}^{AC^{-1}}\boldsymbol{B}^{AC^T} - \boldsymbol{B}^{A}\boldsymbol{Z}^{A^{-1}}\boldsymbol{B}^{A^T}\right)^{-1}\boldsymbol{B}^{AC}\boldsymbol{Z}^{AC^{-1}}\boldsymbol{f}^{AC} \tag{3.75}$$

Substitution in the expression for $\boldsymbol{u}^{AC}$ (eq. (3.73)) gives the decoupled responses:

$$\boldsymbol{u}^{AC} = \left(\boldsymbol{Z}^{AC^{-1}} - \boldsymbol{Z}^{AC^{-1}}\boldsymbol{B}^{AC^T}\left(\boldsymbol{B}^{AC}\boldsymbol{Z}^{AC^{-1}}\boldsymbol{B}^{AC^T} - \boldsymbol{B}^{A}\boldsymbol{Z}^{A^{-1}}\boldsymbol{B}^{A^T}\right)^{-1}\boldsymbol{B}^{AC}\boldsymbol{Z}^{AC^{-1}}\right)\boldsymbol{f}^{AC} \tag{3.76}$$

It should now be realized that the dynamic stiffness matrices are the inverse of the receptance matrices, where the latter are usually obtained from dynamic experiments. , which might be obtained from a (reduced) numerical model, So:

$$\boldsymbol{Z}^{AC^{-1}} = \boldsymbol{Y}^{AC} \quad \text{and} \quad \boldsymbol{Z}^{A^{-1}} = \boldsymbol{Y}^{A}. \tag{3.77}$$

Hence one can write:

$$\begin{aligned}\boldsymbol{u}^{AC} &= \left(\boldsymbol{Y}^{AC} - \boldsymbol{Y}^{AC}\boldsymbol{B}^{AC^T}\left(\boldsymbol{B}^{AC}\boldsymbol{Y}^{AC}\boldsymbol{B}^{AC^T} - \boldsymbol{B}^{A}\boldsymbol{Y}^{A}\boldsymbol{B}^{A^T}\right)^{-1}\boldsymbol{B}^{AC}\boldsymbol{Y}^{AC}\right)\boldsymbol{f}^{AC} \\ &= \boldsymbol{Y}^{AC}\boldsymbol{f}^{AC} - \boldsymbol{Y}^{AC}\boldsymbol{B}^{AC^T}\boldsymbol{Z}^{\text{int}}\boldsymbol{u}^{\text{int}}\end{aligned} \tag{3.78}$$

where

$$\begin{aligned}\boldsymbol{Z}^{\text{int}} &= \left(\boldsymbol{B}^{AC}\boldsymbol{Y}^{AC}\boldsymbol{B}^{AC^T} - \boldsymbol{B}^{A}\boldsymbol{Y}^{A}\boldsymbol{B}^{A}\right)^{-1} \\ \boldsymbol{u}^{\text{int}} &= \boldsymbol{B}^{AC}\boldsymbol{Y}^{AC}\boldsymbol{f}^{AC}\end{aligned} \tag{3.79}$$

A clear physical interpretation can be given to this form of the decoupling problem as follows:

- The term $\boldsymbol{Y}^{AC}\boldsymbol{f}^{AC}$ represents the response of assembly *AC* to external excitation $\boldsymbol{f}^{AC}$
- This leads to interface displacements $\boldsymbol{u}^{\text{int}}$
- However, these interface displacements are due to the combined stiffness of *A* and *C*. Therefore, a corrected interface stiffness $\boldsymbol{Z}^{\text{int}}$ must be calculated to eliminate the influence of substructure *A*
- The adjusted interface stiffness times the interface displacements ($\boldsymbol{Z}^{\text{int}}\boldsymbol{u}^{\text{int}}$) leads to a correction force at the interface DoF
- This force correction is spread to the other DoF through multiplication by $\boldsymbol{Y}^{AC}\boldsymbol{B}^{AC^T}$

Next, the expressions for the receptance matrices of systems *A* and *AC* can be inserted in (3.78) and the products with the Boolean matrices calculated. This gives: Extracting the third row then gives the expression for the uncoupled FRFs $\boldsymbol{Y}_{cc}$ of *C*:

$$\boldsymbol{u}_c = \boldsymbol{Y}^{B}_{cc}\boldsymbol{f}_c = \left(\boldsymbol{Y}^{AC}_{cc} - \boldsymbol{Y}^{AC}_{cb}\left(\boldsymbol{Y}^{AC}_{bb} - \boldsymbol{Y}^{A}_{bb}\right)^{-1}\boldsymbol{Y}^{AC}_{bc}\right)\boldsymbol{f}_c \tag{3.80}$$

Note that this expression for $\boldsymbol{Y}^{B}_{cc}$ is exactly equal to what would be found if a primal approach was taken, as shown in [212]. The procedure of dual decoupling is illustrated in figure 3.12.

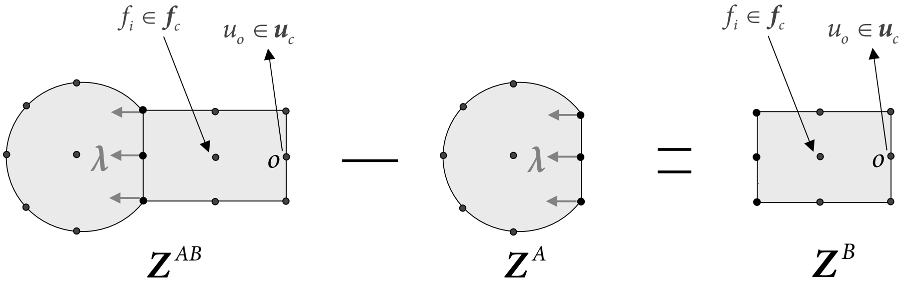

$f_i \in \boldsymbol{f}_c$
$u_o \in \boldsymbol{u}_c$
λ
o
$\boldsymbol{Z}^{AB}$
λ
$\boldsymbol{Z}^{A}$
$f_i \in \boldsymbol{f}_c$
$u_o \in \boldsymbol{u}_c$
o
$\boldsymbol{Z}^{B}$

Figure 3.12: Substructure decoupling from a dual perspective.

3.8.3 Framework for Substructure Decoupling

Based on the dual formulation of the previous section, the approach can be generalized to find a framework from which several decoupling techniques can be derived. This framework is based on the idea that in decoupling problems, in contrast to substructure coupling, a certain freedom exists in the choice of DoF on which to enforce the compatibility and equilibrium conditions. In other words, it is not required to enforce compatibility and equilibrium on the same (number of) DoF. Physically this translates to applying forces at some set of DoF in order to satisfy compatibility at another set of DoF. This idea can be expressed in equations by taking different Boolean matrices for the compatibility and equilibrium conditions, as:

$$\begin{bmatrix} \boldsymbol{Z}^{AC} & \boldsymbol{0} & \boldsymbol{E}^{AC^T} \\ \boldsymbol{0} & -\boldsymbol{Z}^{A} & \boldsymbol{E}^{A^T} \\ \boldsymbol{C}^{AC} & \boldsymbol{C}^{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{AC} \\ \boldsymbol{u}^{A} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{AC} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{3.81}$$

Here $\boldsymbol{E}^{\star}$ are the Boolean matrices defining the location of the uncoupling forces while $\boldsymbol{C}^{\star}$ are the matrices enforcing compatibility. One way to understand the value of this formulation is by realizing that assembly *AC* itself can be written as a dual assembly of components *A* and *C*, so:

$$\boldsymbol{Z}^{AC}\boldsymbol{u}^{AC} = \boldsymbol{f}^{AC} \quad \leftrightarrow \quad \begin{bmatrix} \boldsymbol{Z}^{A} & \boldsymbol{0} & \boldsymbol{B}^{A^T} \\ \boldsymbol{0} & \boldsymbol{Z}^{B} & \boldsymbol{B}^{B^T} \\ \boldsymbol{B}^{A} & \boldsymbol{B}^{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{AC}_{A} \\ \boldsymbol{u}^{AC}_{B} \\ \boldsymbol{\lambda}^{AC} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{f}^{AC}_{B} \\ \boldsymbol{0} \end{bmatrix} \tag{3.82}$$

Here it was again assumed that $\boldsymbol{f}^{A} = \boldsymbol{0}$. This dual form can be inserted in eq. (3.81) to find:

$$\left[\begin{array}{ccc|cc} \boldsymbol{Z}^{A} & \boldsymbol{0} & \boldsymbol{B}^{A^T} & \boldsymbol{0} & -\boldsymbol{E}^{A^T} \\ \boldsymbol{0} & \boldsymbol{Z}^{B} & \boldsymbol{B}^{B^T} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{B}^{A} & \boldsymbol{B}^{B} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\tilde{\boldsymbol{Z}}^{A} & \boldsymbol{E}^{A^T} \\ -\boldsymbol{C}^{A} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{C}^{A} & \boldsymbol{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{u}^{AC}_{A} \\ \boldsymbol{u}^{AC}_{B} \\ \boldsymbol{\lambda}^{AC} \\ \hline \boldsymbol{u}^{A} \\ \boldsymbol{\lambda} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{0} \\ \boldsymbol{f}^{AC}_{B} \\ \boldsymbol{0} \\ \hline \boldsymbol{0} \\ \boldsymbol{0} \end{array}\right] \tag{3.83}$$

Here the dynamic stiffness matrix of the separate component *A* is denoted by $\tilde{\boldsymbol{Z}}^{A}$, while the dynamic stiffness of *A* embedded in the total system *AC* is denoted by $\boldsymbol{Z}^{A}$. Theoretically these should be identical, but due to measurement errors and test procedures they will not be exactly equal in practice so that $\tilde{\boldsymbol{Z}}^{A} = \boldsymbol{Z}^{A} + \boldsymbol{\Delta}$. Writing the decoupling problem in this form allows to derive different decoupling methods, depending on the choices for $\boldsymbol{C}^{\star}$ and $\boldsymbol{E}^{\star}$. In essence these conditions can be anything, as long as they enforce interface compatibility and lead to a solvable interface problem. The goal thereby is to choose these conditions such that the effect of the error $\boldsymbol{\Delta}$ on the decoupled FRFs is minimized.

For the sake of clarity, the expression for $\boldsymbol{Z}^{AC}$ is compacted again to write:

$$\begin{bmatrix} \boldsymbol{Z}^{AC} & \boldsymbol{0} & \boldsymbol{E}^{AC^T} \\ \boldsymbol{0} & -\tilde{\boldsymbol{Z}}^{A} & \boldsymbol{E}^{A^T} \\ \boldsymbol{C}^{AC} & \boldsymbol{C}^{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{AC} \\ \boldsymbol{u}^{A} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{AC} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{3.84}$$

Taking the above expression as the general formulation of the decoupling problem, the Lagrange multipliers can be eliminated and solved for the uncoupled responses as before. This gives

$$\boldsymbol{u}^{AC} = \left(\boldsymbol{Y}^{AC} - \boldsymbol{Y}^{AC}\boldsymbol{E}^{AC^T}\left(\boldsymbol{C}^{AC}\boldsymbol{Y}^{AC}\boldsymbol{E}^{AC^T} - \boldsymbol{C}^{A}\tilde{\boldsymbol{Y}}^{A}\boldsymbol{E}^{A^T}\right)^{+}\boldsymbol{C}^{AC}\boldsymbol{Y}^{AC}\right)\boldsymbol{f}^{AC}, \tag{3.85}$$

where $^{+}$ denotes the (Moore-Penrose) pseudo-inverse, since the middle bracketed term is now no longer necessarily a square matrix.

On the basis of equations (3.84) and (3.85) a family of decoupling techniques can be derived and existing techniques can be classified. These methods are however not in the main focus of this thesis, the reader is therefore referred to [213] for further details.

3.9 Summary

Substructure assembly techniques were the topic of this chapter. In section 3.2 a general framework was sketched for substructure assembly by identifying the three possible assembly cases. Based on a three-field formulation the "stiffness", "flexibility" and "mixed" assembly cases have been worked out in sections 3.3, 3.4 and 3.5 respectively, thereby enabling assembly of all types of (reduced) component models. Hence, all types of reduced models can now be treated as superelements, even those obtained from the Dual and Mixed Craig-Bampton methods (see sections 2.5.4 and 2.6). As such, this framework allows reduced models to be created independently and used as building blocks for the total model.

Section 3.6 addressed the issue of non-conforming interface meshes, a situation regularly encountered if substructure models are created separately. The simple but effective node collocation method was outlined, as well as its least squares variant. These methods allow to enforce point-wise compatibility between the substructures and can be directly incorporated in the assembly framework. Thereafter, section 3.7 treated the situation where additional physics arise from the interface. Such effects are often overlooked or neglected in substructure assembly. It was outlined how these additional physics can be accounted for in a straightforward manner, without the need to model them explicitly as an intermediate substructure. Note that all assembly techniques have been incorporated in the DS Toolbox, and more specifically the AssemblyTool (see appendix A).

In the final section of this chapter a small sidestep was made by considering the reverse problem, namely the decoupling of a component model from an assembled system. This problem is relevant for subsystems that cannot be properly modeled and can only be characterized experimentally when attached to some neighboring substructure(s). Based on a dual formulation, it was shown that the decoupling problem amounts to assembly of a negative dynamic stiffness. Furthermore, the dual disassembly approach allows freedom in the choice for DoF on which to enforce the compatibility and equilibrium conditions. This freedom is unique to decoupling problems and allows to derive new, possibly more robust techniques.

4

Interface Reduction & Assembled System Analysis

Essentially, all models are wrong,
but some are useful.

(George Box)

4.1 Introduction

In the previous to chapters it was explained how component models can be reduced and subsequently assembled to obtain compact representations of detailed structural dynamic models. Sometimes however, the component models are connected through large and complex interfaces leading to high numbers of interface DoF. Obviously, this limits the achievable reduction of the component models and hence decreases the efficiency of the dynamic substructuring approach.

To overcome this problem *interface reduction* can be applied such that truly compact models can be obtained. In this second reduction step, the number of interface DoF is decreased while the overall accuracy remains at an acceptable level. A number of methods exist to achieve this, which will be addressed in the first part of this chapter. First, section 4.2 explains the issue associated with and different options for reducing the boundary DoF. Section 4.3 thereafter outlines a simple yet effective reduction technique. The more general modal truncation based interface reduction methods are subsequently addressed in sections 4.4 and 4.5, respectively for interfaces expressed in terms of displacement and force DoF.

Once a satisfactory assembled model is obtained, it can be used for the actual structural dynamic analysis of the modeled system. This is the topic of the second part of this chapter. For instance, the model can be subjected to a modal or harmonic analysis, or a time simulation. Although such analysis methods span a research area in itself, section 4.6 briefly addresses

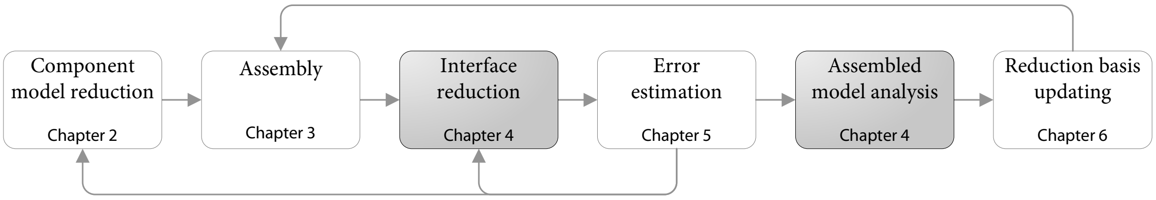


Figure 4.1: Current chapter in relation to other chapters in part I of this thesis.

these for illustrative purposes. Thereafter, section 4.7 outlines different comparison and correlation techniques that can be used judge the results obtained from the analyses. As usual, the chapter is concluded with a brief summary in section 4.8 and the relation between the topics of this and other chapters is illustrated in figure 4.1.

4.2 Options for Interface Reduction

Complex engineering structures, such as a modern wind turbine, commonly consist of a large number of (structural) components. Consequently, a large number of interfaces between these components exist. Furthermore, these interfaces may be extensive and complex. When a structural dynamic model of such a structure is created in a componentwise fashion using the reduction techniques of chapter 2, the ratio of interface over generalized DoF becomes unacceptably high. Consequently, the achievable reduction is limited and the memory requirements for handling the full (instead of sparse) component reduction matrices become excessive. This difficulty can be overcome using interface reduction techniques. Similar to component reduction techniques, these consist in replacing the original interface DoF by some set of deformation shapes, called interface modes, and associated generalized DoF. As such, they constitute a second reduction step which leads to more compact equations of motion as well as smaller sized component reduction matrices.

In principle, interface reduction can be applied both on substructure level and on assembly level. Since the interface behavior is dependent on all components that it connects, it is obvious that a priori interface reduction on component level in general gives far less accurate results than when information of the assembled structure is used to reduce the interface DoF. Even worse, component level interface reduction gives rise to non-conforming interfaces, similar to non-matching element shape functions, which in turn can cause so-called interface locking if the reduction bases are poor. Hence, component level reduction of the interface DoF can only be successful when the interface DoF of all components connected through one interface are reduced using the same basis, and the deformation shapes in this basis are representative for the actual interface behavior.

Having outlined the general idea behind interface reduction techniques, the actual reduction can be achieved in a number of ways:

- The simplest method to decrease the number of interface DoF is to assume that the interface behavior can be described by local rigid motions. This "interface rigidification" hence neglects local deformation of the interface, allowing each substructure interface to be described by only six DoF. The process of interface rigidification will be elaborated in section 4.3.

- In many cases however, interfaces cannot be assumed to behave rigidly. A more accurate method, which can in fact be seen as a generalization of the interface rigidification, is then to apply modal reduction to the interface DoF. Similar to the component reduction techniques of chapter 2, a modal basis is computed for the boundary DoF by solving an eigenvalue problem. By truncating the interface modes, interface reduction is achieved and the physical boundary DoF are replaced by generalized DoF. In sections 4.4 and 4.5 the procedures are explained in detail for the case where the interface is expressed in terms of displacement DoF and force DoF, respectively. For systems with mixed boundary DoF, such as described in section 3.5, interface reduction can be applied by combining both approaches.

- A number of other interface reduction methods have been proposed over the years. Some notable recent contributions are:
    - The work in [201] considers model reduction of cyclic symmetric structures. By exploiting the symmetry only a small portion of the total structure needs to be modeled, but as a result this model has many interface DoF. An interface reduction method is proposed to overcome this. The work is extended in [202] for the case when only part of the interface DoF needs to be reduced and other boundary DoF are retained.
    - In [223, 224] an interface between two substructures is considered internal to the structure instead of considering the associated DoF as boundary DoF. Using the equilibrium condition, so-called *joint interface modes* are then computed which are subsequently used to reduce the (double set of) joint DoF.
    - An extension of the technique presented in section 4.4 is proposed in [109], for assembled systems obtained from Rubin reduced component models. The authors developed an iterative scheme in which (residual) attachment modes are only added to the reduction basis if they satisfy a certain orthogonality criterion. In this way, only the most important static modes are taken into account, leading to a much smaller reduction basis.

It is noted that all interface reduction methods discussed in this chapter can be applied on component as well as on assembly level. For the reasons explained above, the interface rigidification method of the next section is well suited to be applied on component level, whereas the general modal methods of sections 4.4 and 4.5 usually give much better results when applied on assembly level. One exception is the case when modal truncation augmentation is applied to improve the dynamic response to excitation coming from neighboring components through the interface DoF (see section 2.7). In that case, substructure level interface reduction may be a good way to decrease the number of additional MTA vectors. Indeed, the accuracy of this interface reduction is less critical since the MTAs are only higher order correction vectors that augment the basis; the full set of boundary DoF remains.

Finally, note that due to the interface reduction the total number of DoF to deform in is further decreased which, as in any reduction method, leads to stiffening of the overall structure.

4.3 Interface Rigidification

In case an interface is located on a stiff part of the substructure, or is relatively small compared to the total substructure, one could approximate the behavior of the interface by a local rigid region. This assumption allows to reduce the interface DoF to only six rigid motions, namely three translations and three rotations. Hence a very compact representation of the substructure interface is obtained. Note that rigid interface modeling is a straightforward way of enabling assembly of substructures with non-conforming meshes, eliminating the need for more complex node collocation techniques (see section 3.6).

The rigidification process is basically a projection of the original interface DoF on the local rigid body modes of the interface. Since these rigid body modes only depend on the geometry of the interface, the rigidification can be performed both on substructure and assembly level. As long as the same reference point on the interface is chosen, identical results will be obtained in both cases. In practice it is however more efficient to perform the rigidification a priori on substructure level, as this avoids the computation of the many static modes associated to the original set of substructure boundary DoF.

Suppose now that a substructure's interface DoF $\boldsymbol{u}_b$ can be partitioned into a set of DoF belonging to interfaces that need to be rigidified $\boldsymbol{u}_{b,r}$ and a set that is retained $\boldsymbol{u}_{b,b}$, for instance because the associated interfaces are too large or flexible for rigidification. For each of the g physical interfaces in $\boldsymbol{u}_{b,r}$ that need to be rigidified a projection matrix needs to be constructed, such that:

$$\boldsymbol{u}_{b,r}^j = \boldsymbol{T}_r^j \boldsymbol{q}_r^j \quad \text{for} \quad j = 1 \dots g \tag{4.1}$$

Here $\boldsymbol{q}_r^j$ are the six rigid body coordinates of interface j:

$$\boldsymbol{q}_r^j = \begin{bmatrix} x^j & y^j & z^j & \theta_x^j & \theta_y^j & \theta_z^j \end{bmatrix}^T \tag{4.2}$$

Note that the origin of the rigid body coordinates can be chosen arbitrarily, either coinciding with some existing FE node or some other convenient location. One however needs to ensure that the interface rigid body DoF on the to-be connected substructure(s) are defined with respect to the same global geometric position. Next, the interface projection matrix is constructed on a per-node basis as:

$$\boldsymbol{T}_r^j = \text{col}\left(\boldsymbol{T}_{r(1)}^j; \dots; \boldsymbol{T}_{r(n_n)}^j\right) \tag{4.3}$$

Here n_n is the number of nodes associated with interface j. The nodal projection matrix for node k is formed as:

$$\boldsymbol{T}_{r(k)}^j = \begin{bmatrix} \boldsymbol{I}_{33} & \boldsymbol{T}_{\boldsymbol{\theta\theta}(k)} \\ \boldsymbol{0}_{33} & \boldsymbol{I}_{33} \end{bmatrix} \tag{4.4}$$

with:

$$\boldsymbol{T}_{\boldsymbol{\theta\theta}(k)} = \begin{bmatrix} \boldsymbol{e}_1 \times \boldsymbol{d}_k & \boldsymbol{e}_2 \times \boldsymbol{d}_k & \boldsymbol{e}_3 \times \boldsymbol{d}_k \end{bmatrix} = \begin{bmatrix} 0 & d_{z(k)} & -d_{y(k)} \\ -d_{z(k)} & 0 & d_{x(k)} \\ d_{y(k)} & -d_{x(k)} & 0 \end{bmatrix} \tag{4.5}$$

Here $\boldsymbol{d}_k$ is the position vector from the chosen location of the rigid body coordinates to the current node k. For a finite element model built from elements with only translational DoF (e.g. using tetrahedron elements), the bottom three lines in eq. (4.4) can be discarded and $\boldsymbol{T}^j_{r(k)}$ becomes a matrix of size 3×6 instead of 6×6. This approach is similar to the procedure for obtaining the global rigid body modes in section 2.4.2.

When the transformation matrices are collected for all nodes of all g interfaces in $\boldsymbol{u}_{b,r}$, one obtains the following transformation:

$$\begin{bmatrix} \boldsymbol{u}_{b,r} \\ \boldsymbol{u}_{b,b} \\ \boldsymbol{u}_i \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{T}_r & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_r \\ \boldsymbol{u}_{b,b} \\ \boldsymbol{u}_i \end{bmatrix} = \boldsymbol{R}_r \begin{bmatrix} \boldsymbol{q}_r \\ \boldsymbol{u}_{b,b} \\ \boldsymbol{u}_i \end{bmatrix} \tag{4.6}$$

By substitution and projection of $\boldsymbol{R}_r$ on the original substructure equations of motion, the stiffness and mass of the interfaces in $\boldsymbol{u}_{b,r}$ are condensed onto the reference interface nodes. The so obtained locally rigidified component equations may be further reduced using any of the component reduction techniques outlined in chapter 2.

Since interface rigidification locally creates an infinitely stiff section, one can imagine that this will mostly affect the mode shapes in which this rigid section would previously deform. This leads to higher eigenfrequencies for these modes after rigidification. If a substructure has many interfaces and/or interfaces take up a large portion of the substructure's surface this approach is therefore not advisable, as it would substantially increase the stiffness of the entire structure. One way to check whether rigidification of an interface is "allowed" is by calculating its original rigidity. Suppose that some deformation vector, such as a mode shape or operational deflection shape, is known for the component. For this deformation vector the interface rigidity can be quantified by projecting the interface deformation on the local rigid modes $\boldsymbol{T}_r$ and recovering the deformations in a least squares sense [54]:

$$\text{rigidity} = \frac{\left\| \boldsymbol{T}_r \left(\boldsymbol{T}_r^T \boldsymbol{T}_r \right)^{-1} \boldsymbol{T}_r^T \boldsymbol{u}_{b,r} \right\|}{\left\| \boldsymbol{u}_{b,r} \right\|} \cdot 100\% \tag{4.7}$$

Note that in order to obtain a complete picture of the interface rigidity, this check should be performed for a representative set of deformation vectors. Hence, in practice this check is usually carried out a posteriori.

This rigidification could also be extended by including local interface modes to the basis in (4.6) to account for some interface flexibility [134, 4, 54]. In this case, this approach can be regarded as the substructure level equivalent of the modal interface reduction method outlined next.

4.4 Modal Reduction of Interface Displacements

Although the rigidification technique of the previous section is useful for some structures, the rigid interface approximation is often too crude. In that case the more general modal truncation based method discussed in this section can be applied. This method is based on the observation that determining the interface behavior generally does not require detailed insight in the components' dynamic behavior; an accurate representation of the static behavior at the interface is often sufficient. Hence, interface modes are computed from the eigenvalue problem of the substructures statically condensed to the interface. This idea was first proposed in [40] and was later further worked out in [15, 201]. In this section the method will be detailed for systems with displacement interface DoF.

As a starting point for the interface reduction method, recall the primal form of the stiffness assembly derived in section 3.3.2:

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{ib}\boldsymbol{L}_b \\ \boldsymbol{L}_b^T\boldsymbol{M}_{bi} & \boldsymbol{L}_b^T\boldsymbol{M}_{bb}\boldsymbol{L}_b \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib}\boldsymbol{L}_b \\ \boldsymbol{L}_b^T\boldsymbol{K}_{bi} & \boldsymbol{L}_b^T\boldsymbol{K}_{bb}\boldsymbol{L}_b \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{L}_b^T\boldsymbol{f}_b \end{bmatrix} \tag{3.14}$$

Taking only the boundary portion of the assembled equations of motion, one finds:

$$\hat{\boldsymbol{M}}_{bb}\ddot{\boldsymbol{u}}_\gamma + \hat{\boldsymbol{K}}_{bb}\boldsymbol{u}_\gamma = \hat{\boldsymbol{f}}_b \tag{4.8}$$

Following from the definitions of section 3.3 one has:

$$\begin{aligned} \hat{\boldsymbol{M}}_{bb} &= \boldsymbol{L}_b^T\boldsymbol{M}_{bb}\boldsymbol{L}_b = \sum_{s=1}^{n} \boldsymbol{L}_b^{(s)^T}\boldsymbol{M}_{bb}^{(s)}\boldsymbol{L}_b^{(s)} \\ \hat{\boldsymbol{K}}_{bb} &= \boldsymbol{L}_b^T\boldsymbol{K}_{bb}\boldsymbol{L}_b = \sum_{s=1}^{n} \boldsymbol{L}_b^{(s)^T}\boldsymbol{K}_{bb}^{(s)}\boldsymbol{L}_b^{(s)} \\ \hat{\boldsymbol{f}}_{bb} &= \boldsymbol{L}_b^T\boldsymbol{f}_b = \sum_{s=1}^{n} \boldsymbol{L}_b^{(s)^T}\boldsymbol{f}_b^{(s)} \end{aligned} \tag{4.9}$$

Here it is assumed that the assembly consists of reduced component models obtained from either the Craig-Bampton or Rubin method (see section 2.5). In this case, equation (4.8) in fact constitutes a static condensation of all the substructures to the interface DoF. For the Craig-Bampton method this is obvious, given the definitions of $\tilde{\boldsymbol{K}}_{bb}^{(s)}$ and $\tilde{\boldsymbol{M}}_{bb}^{(s)}$ in eq. (2.57). Although less obvious, this is also true for Rubin reduced components. As long as no rigid body modes are present, it can be shown that the attachment modes in the Rubin basis span the same space as the constraint modes and hence the terms $\boldsymbol{K}_{r,bb}$ and $\boldsymbol{M}_{r,bb}$ in eq. (2.68) are equivalent to their Craig-Bampton counterparts. If the substructure is unconstrained, one can alter the reduction basis to include the constraint modes to enable static condensation; this is described in [201].

Since the mass and stiffness of the substructures is condensed to the interface, equation 4.8 can be considered as the assembled system obtained when no vibration modes are included in the component reduction bases. This equation can therefore be used to find the interface behavior. Assuming the external forces to be zero and finding a non-trivial solution then leads to boundary eigenvalue problem:

$$\left(\hat{\boldsymbol{K}}_{bb} - \omega_{\gamma,j}^2\hat{\boldsymbol{M}}_{bb}\right)\boldsymbol{\phi}_{\gamma,j} = \boldsymbol{0} \tag{4.10}$$

Here $\boldsymbol{\phi}_{\gamma,j}$ is the j^{th} *interface displacement mode* with $\omega_{\gamma,j}^2$ its associated frequency. As usual, the reduction is obtained by truncating the number of modes. Taking the n_γ interface modes with the lowest frequency, the interface DoF are approximated by:

$$\boldsymbol{u}_\gamma \approx \sum_{j=1}^{n_\gamma} \boldsymbol{\phi}_{\gamma,j}\eta_{\gamma,j} = \boldsymbol{\Phi}_\gamma\boldsymbol{\eta}_\gamma \tag{4.11}$$

The interface reduction basis thus becomes:

$$\begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_\gamma \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}_\gamma \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\eta}_\gamma \end{bmatrix} = \boldsymbol{R}_\gamma \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\eta}_\gamma \end{bmatrix} \tag{4.12}$$

Application of this reduction basis to the assembled system of eq. (3.14) then gives the final reduced equations of motion, where the physical boundary DoF are replaced by generalized DoF. For the sake of illustration, the approach is shown here for the case of an assembled system consisting of Craig-Bampton reduced components. Using the reduced matrix expressions from section 2.5.2 and the block notation of section 3.3, the assembled system can be written as:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{M}_{\phi b}\boldsymbol{L}_b \\ \boldsymbol{L}_b^T\boldsymbol{M}_{b\phi} & \boldsymbol{L}_b^T\tilde{\boldsymbol{M}}_{bb}\boldsymbol{L}_b \end{bmatrix}\begin{bmatrix} \ddot{\boldsymbol{\eta}}_i \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Omega}_i^2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}_b^T\tilde{\boldsymbol{K}}_{bb}\boldsymbol{L}_b \end{bmatrix}\begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{f}}_i \\ \boldsymbol{L}_b^T\tilde{\boldsymbol{f}}_b \end{bmatrix} \tag{4.13}$$

Here all submatrices are block diagonal containing the reduced substructure submatrices; the vectors are also put in block form, e.g.:

$$\begin{aligned} \tilde{\boldsymbol{K}}_{bb} &= \operatorname{diag}\left(\tilde{\boldsymbol{K}}_{bb}^{(1)},\ldots,\tilde{\boldsymbol{K}}_{bb}^{(n)}\right) \\ \boldsymbol{\eta}_i &= \operatorname{col}\left(\boldsymbol{\eta}_i^{(1)};\ldots;\boldsymbol{\eta}_i^{(n)}\right) \end{aligned} \tag{4.14}$$

Application of the interface reduction basis in eq. (4.12) to the assembly of Craig-Bampton reduced components in (4.13) then gives:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{M}_{\phi b}\boldsymbol{L}_b\boldsymbol{\Phi}_\gamma \\ \boldsymbol{\Phi}_\gamma^T\boldsymbol{L}_b^T\boldsymbol{M}_{b\phi} & \boldsymbol{I} \end{bmatrix}\begin{bmatrix} \ddot{\boldsymbol{\eta}}_i \\ \ddot{\boldsymbol{\eta}}_\gamma \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Omega}_i^2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Omega}_\gamma^2 \end{bmatrix}\begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{\eta}_\gamma \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{f}}_i \\ \boldsymbol{\Phi}_\gamma^T\boldsymbol{L}_b^T\tilde{\boldsymbol{f}}_b \end{bmatrix} \tag{4.15}$$

Here the interface displacement modes are assumed to be mass normalized. As can be seen, the assembled stiffness matrix is now fully diagonal. Note that for Rubin reduced systems this is not the case due to the presence of off-diagonal terms. Furthermore, since all components are reduced with the same set of interface displacement modes, the substructures remain compatible in terms of their interface discretizations.

In addition to reducing the total number of DoF of the assembled model, interface reduction can also be used to decrease the size of the component reduction matrices. Thereby, handling of the component reduction matrices, which can easily take up many gigabytes of memory and storage space, is greatly simplified. This can be achieved by simply combining the subsequent coordinate transformations that lead to equation (4.13). Indeed, three transformation steps are performed to arrive at this equation, which using the block notation can be written as:

1. Reduction of the components using the Craig-Bampton reduction bases:

$$\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi}_i & \boldsymbol{\Psi}_c \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}\begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{u}_b \end{bmatrix} = \boldsymbol{R}_{CB}\begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{u}_b \end{bmatrix} \tag{4.16}$$

2. Primal assembly is performed by pre- and post-multiplication by the assembly matrix, as explained in section 3.3.2:

$$\begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}_b \end{bmatrix}\begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{u}_\gamma \end{bmatrix} = \boldsymbol{L}\begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{u}_\gamma \end{bmatrix} \tag{4.17}$$

3. Application of the interface reduction basis in equation (4.12).

By far the largest amount of memory space is required by the first transformation; the three transformations can be combined to decrease the size of this matrix. This gives rise to the following total reduction matrix:

$$\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi}_i & \boldsymbol{\Psi}_c \boldsymbol{L}_b \boldsymbol{\Phi}_\gamma \\ \boldsymbol{0} & \boldsymbol{L}_b \boldsymbol{\Phi}_\gamma \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{\eta}_\gamma \end{bmatrix} = \boldsymbol{R}_{CB} \boldsymbol{L} \boldsymbol{R}_\gamma \begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{\eta}_\gamma \end{bmatrix} = \boldsymbol{R}_{\text{tot}} \begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{\eta}_\gamma \end{bmatrix} \tag{4.18}$$

Due to the interface reduction, the size of $\boldsymbol{R}_{\text{tot}}$ is much smaller than that of $\boldsymbol{R}_{CB}$. Therefore, handling the assembled reduced system becomes much easier, while the substructure nodal results can still be easily obtained through expansion with $\boldsymbol{R}_{\text{tot}}$.

4.5 Modal Reduction of Interface Forces

In a similar fashion as in the previous section, interface reduction can also be applied to systems described in terms of interface force DoF. To this end, recall the primal assembled equations derived in section 3.4:

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{ii} & -\bar{\boldsymbol{M}}_{ib} \boldsymbol{B}_b^T \\ -\boldsymbol{B}_b \bar{\boldsymbol{M}}_{bi} & \boldsymbol{B}_b \bar{\boldsymbol{M}}_{bb} \boldsymbol{B}_b^T \end{bmatrix} \begin{bmatrix} \ddot{\bar{\boldsymbol{q}}}_i \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{K}}_{ii} & -\bar{\boldsymbol{K}}_{ib} \boldsymbol{B}_b^T \\ -\boldsymbol{B}_b \bar{\boldsymbol{K}}_{bi} & \boldsymbol{B}_b \bar{\boldsymbol{K}}_{bb} \boldsymbol{B}_b^T \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{q}}_i \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{f}}_i \\ -\boldsymbol{B}_b \bar{\boldsymbol{f}}_b \end{bmatrix} \tag{3.27}$$

Next, a static condensation of the substructures to the boundary DoF needs again to be obtained. In contrast to the previous section, this cannot be found by simply taking the boundary portion of the assembled equations since in this case rigid body modes may contribute to the static solution. Hence, these must be explicitly accounted for in the static condensation.

This is best illustrated by considering the Dual Craig-Bampton version of the assembled equations. Here, the alternative formulation of the Dual Craig-Bampton method, using attachment instead of residual attachment modes, will be used; the reason for doing so will be explained below. Combining the associated expressions for the reduced component matrices from section 2.5.4 with the assembled form shown above, one then obtains the assembled model consisting of Dual Craig-Bampton component models as:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & -\boldsymbol{M}_{\phi\psi} \boldsymbol{B}_b^T \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{B}_b \boldsymbol{M}_{\psi\phi} & \boldsymbol{0} & \boldsymbol{B}_b \boldsymbol{M}_{f,bb} \boldsymbol{B}_b^T \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_f \\ \ddot{\boldsymbol{\eta}}_r \\ \ddot{\boldsymbol{g}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Omega}_f^2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Phi}_{r|b}^T \boldsymbol{B}_b^T \\ \boldsymbol{0} & \boldsymbol{B}_b \boldsymbol{\Phi}_{r|b} & -\boldsymbol{B}_b \boldsymbol{G}_{f,bb} \boldsymbol{B}_b^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_r \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_f^T \boldsymbol{f} \\ \boldsymbol{\Phi}_r^T \boldsymbol{f} \\ -\boldsymbol{B}_b \boldsymbol{\Psi}_a^T \boldsymbol{f} \end{bmatrix} \tag{4.19}$$

Similar to the previous section, block notation is applied here. The submatrices are block diagonal matrices containing the component submatrices, while the vectors are also stacked in block form, e.g.:

$$\begin{aligned} \boldsymbol{G}_{f,bb} &= \operatorname{diag}\left(\boldsymbol{G}_{f,bb}^{(1)}, \dots, \boldsymbol{G}_{f,bb}^{(n)}\right) \\ \boldsymbol{\eta}_f &= \operatorname{col}\left(\boldsymbol{\eta}_f^{(1)}; \dots; \boldsymbol{\eta}_f^{(n)}\right) \end{aligned} \tag{4.20}$$

Taking no free interface vibration modes in the reduction basis allows to obtain the system statically condensed to the interface DoF as:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_b \boldsymbol{M}_{f,bb} \boldsymbol{B}_b^T \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_r \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Phi}_{r|b}^T \boldsymbol{B}_b^T \\ \boldsymbol{B}_b \boldsymbol{\Phi}_{r|b} & -\boldsymbol{B}_b \boldsymbol{G}_{f,bb} \boldsymbol{B}_b^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_r^T \boldsymbol{f} \\ -\boldsymbol{B}_b \boldsymbol{\Psi}_a^T \boldsymbol{f} \end{bmatrix} \tag{4.21}$$

Note that equation (4.19) corresponds to the alternative formulation of the Dual Craig-Bampton method using attachment modes, while the original formulation employs the residual attachment modes (see section 2.5.4). Due to the absence of vibration modes in the static interface condensation, eq. (4.21) is independent of the type of attachment. However, there is an important practical advantage of using the alternative formulation.

Namely, in case one follows the approach of this section using the original Dual Craig-Bampton formulation, the residual interface flexibility matrix $\boldsymbol{G}_{r,bb}$ (see section 2.5.4) needs to be supplemented by the retained free interface vibration modes to obtain the (statically complete) interface flexibility matrix $\boldsymbol{G}_{f,bb}$. One thus needs to reload the substructure reduction bases and apply (2.29) in a backward fashion; this additional computational effort can be avoided by using attachment modes.

Next, the interface modes are obtained by solving the statically condensed eigenproblem:

$$\left(\begin{bmatrix} \boldsymbol{0} & \boldsymbol{\Phi}_{r|b}^T \boldsymbol{B}_b^T \\ \boldsymbol{B}_b \boldsymbol{\Phi}_{r|b} & -\boldsymbol{B}_b \boldsymbol{G}_{f,bb} \boldsymbol{B}_b^T \end{bmatrix} - \omega_{\lambda,j}^2 \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_b \boldsymbol{M}_{f,bb} \boldsymbol{B}_b^T \end{bmatrix}\right) \begin{bmatrix} \boldsymbol{\phi}_{\lambda,j}^r \\ \boldsymbol{\phi}_{\lambda,j}^\lambda \end{bmatrix} = \boldsymbol{0} \tag{4.22}$$

Here $\boldsymbol{\phi}_{\lambda,j}$ is the j^{th} *interface force mode* with $\omega_{\lambda,j}^2$ the corresponding eigenvalues. It was observed in [170] that these eigenvalues can be both positive and negative. The former are associated with the rigid body modes while the latter correspond to the Lagrange multipliers. Interface reduction is now achieved by choosing the n_γ modes with the *lowest absolute eigenvalues* [170], resulting in the following interface reduction basis:

$$\begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_r \\ \boldsymbol{\lambda} \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}_\lambda^r \\ \boldsymbol{0} & \boldsymbol{\Phi}_\lambda^\lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_\lambda \end{bmatrix} = \boldsymbol{R}_\lambda \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_\lambda \end{bmatrix} \tag{4.23}$$

Applying this reduction to the assembled equations in (4.19) leads to:

$$\begin{bmatrix} \boldsymbol{I} & -\boldsymbol{M}_{\phi\psi} \boldsymbol{B}_b^T \boldsymbol{\Phi}_\lambda^\lambda \\ -\boldsymbol{\Phi}_\lambda^{\lambda^T} \boldsymbol{B}_b \boldsymbol{M}_{\psi\phi} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_f \\ \ddot{\boldsymbol{\eta}}_\lambda \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Omega}_f^2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Omega}_\lambda^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_\lambda \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_f^T \boldsymbol{f} \\ \boldsymbol{f}_\lambda \end{bmatrix} \tag{4.24}$$

with:

$$\boldsymbol{f}_\lambda = \left(\boldsymbol{\Phi}_\lambda^{r^T} \boldsymbol{\Phi}_r^T + \boldsymbol{\Phi}_\lambda^{\lambda^T} \boldsymbol{B}_b \boldsymbol{\Psi}_a^T\right) \boldsymbol{f} \tag{4.25}$$

After interface reduction one again obtains a diagonal stiffness matrix whereas coupling terms are found in the mass matrix.

In a similar fashion as in the previous section, the interface reduction basis can be used to decrease the size of the component reduction bases. Again, the reduced assembled equations in (4.24) are obtained in three transformation steps. Using the same block notation as before, these transformations are:

1. Component reduction using the Dual Craig-Bampton reduction bases of eq. (2.72):

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{g}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi}_f & \boldsymbol{\Phi}_r & \boldsymbol{\Psi}_a \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_f \\ \boldsymbol{g}_b \end{bmatrix} = \boldsymbol{R}_{DCB} \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_f \\ \boldsymbol{g}_b \end{bmatrix} \tag{4.26}$$

2. Assembly of components by the primal transformation derived in section 3.4:

$$\begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_f \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{B}_b^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_f \\ \lambda \end{bmatrix} = \boldsymbol{B} \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_f \\ \lambda \end{bmatrix} \tag{4.27}$$

3. Interface reduction of the assembled system by applying the basis $\boldsymbol{R}_\lambda$ in (4.23).

Combining these transformations allows to obtain a compact form of the total reduction matrix. As explained before, this can limits the memory required for handling and storing the reduction basis, which is very useful in practice. The total reduction matrix is in this case found as:

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{g}_b \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi}_f & \boldsymbol{\Phi}_r \boldsymbol{\Phi}_\lambda^r - \boldsymbol{\Psi}_a \boldsymbol{B}_b^T \boldsymbol{\Phi}_\lambda^\lambda \\ \boldsymbol{0} & -\boldsymbol{B}_b^T \boldsymbol{\Phi}_\lambda^\lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_\gamma \end{bmatrix} = \boldsymbol{R}_{DCB} \boldsymbol{B} \boldsymbol{R}_\lambda \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_\gamma \end{bmatrix} = \boldsymbol{R}_{\text{tot}} \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_\gamma \end{bmatrix} \tag{4.28}$$

4.6 Analysis of Assembled Model

After the assembled model is obtained it is used to perform the actual dynamic analysis of the modeled system. To this end the model can be subjected to variety of analysis types. In this section three types of analysis often encountered in practice are briefly addressed, namely harmonic analysis, modal analysis and time integration. In order to simplify the notations, the equations of motion of the reduced assembled model are written in general as:

$$\tilde{\boldsymbol{M}} \ddot{\boldsymbol{q}}(t) + \tilde{\boldsymbol{C}} \dot{\boldsymbol{q}}(t) + \tilde{\boldsymbol{K}} \boldsymbol{q}(t) = \tilde{\boldsymbol{f}}(t) \tag{4.29}$$

Here the damping matrix is assumed to derive from some simple damping model, such as the Rayleigh or modal damping assumption.

4.6.1 Harmonic Analysis

When performing an harmonic analysis the aim is to determine the steady-state response of the structure to a certain harmonic external loading. This can for instance be relevant for rotating machinery, such as a wind turbine, which are subjected to excitation at (multiples of) the rotation frequency, or other structures exposed to loading with a distinct frequency.

Suppose the external excitation can be written as some harmonic with frequency ω_e:

$$\tilde{\boldsymbol{f}}(t) = \tilde{\boldsymbol{p}} e^{i\omega_e t} \tag{4.30}$$

Here $\tilde{\boldsymbol{p}}$ is the spatial force distribution vector. A steady-state solution can now be found by assuming that the response of the system is synchronous with the excitation, i.e.:

$$\boldsymbol{q}(t) = \tilde{\boldsymbol{x}} e^{i\omega_e t}, \tag{4.31}$$

with $\tilde{\boldsymbol{x}}$ the unknown harmonic deformation. Inserting the harmonic response in the assembled equations of motion (4.29) and discarding the time dependent part, one obtains the harmonic problem as:

$$\left(\tilde{\boldsymbol{K}} + i\omega_e \tilde{\boldsymbol{C}} - \omega_e^2 \tilde{\boldsymbol{M}}\right) \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{Z}} \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{p}} \tag{4.32}$$

The matrix $\tilde{\boldsymbol{Z}}$ is known as the (reduced) dynamic stiffness matrix of the system. Note that when the external excitation frequency ω_e reduces to zero, the harmonic problem becomes a static problem. In general, the harmonic problem is solved by *factorizing* (i.e. triangularization of) the dynamic stiffness matrix, using direct solvers based on the classic elimination techniques by Gauss. Since the reduced harmonic problem is small, this can be done very efficiently. Alternatively, iterative solvers such as the conjugate gradient method can be employed (see section 6.2.1 for details).

Finally, note that the obtained deformation $\tilde{\boldsymbol{x}}$ is in terms of the generalized DoF. Using the component and possibly interface reduction bases, this vector can be expanded to obtain the structure's response at the physical degrees of freedom:

$$\boldsymbol{x} = \boldsymbol{R}\tilde{\boldsymbol{x}} \tag{4.33}$$

The vector $\boldsymbol{x}$ is known as the operational deflection shape (ODS). When the external excitation coincides with one of the system's eigenfrequencies, the ODS has the same shape as the associated eigenmode.

4.6.2 Modal Analysis

To determine the eigenmodes and -frequencies of the assembled system, a modal analysis can be performed. Such an analysis is useful when the external excitation is still unknown and provides insight in the intrinsic dynamic properties of the structure, namely its free vibration behavior in terms of eigenfrequencies and mode shapes. Modal analysis thus amounts to finding a solution to the equations of motion with no external force applied:

$$\tilde{\boldsymbol{M}}\ddot{\boldsymbol{q}}(t) + \tilde{\boldsymbol{K}}\boldsymbol{q}(t) = \boldsymbol{0} \tag{4.34}$$

Here damping is assumed to be small, such that it can be neglected during the modal analysis. If needed a correction can be computed to obtain the damped eigensolutions, see [74]. A non-trivial solution can be found by separating the response of the internal DoF in a space-dependent and time-dependent part, i.e.:

$$\boldsymbol{q} = \tilde{\boldsymbol{x}}e^{i\omega t} \tag{4.35}$$

Substitution in the free vibration equations then leads to the eigenproblem:

$$\left(\tilde{\boldsymbol{K}} - \omega_j^2\tilde{\boldsymbol{M}}\right)\tilde{\boldsymbol{\phi}}_j = \boldsymbol{0}, \tag{4.36}$$

with ω_j^2 the eigenfrequency of mode j and $\tilde{\boldsymbol{\phi}}_j$ the associated eigenmode (or eigenvector or -shape). Many types of solvers exist to handle the eigenproblem, both direct (such as the Jacobi and Householder methods) and iterative (such as the Lanczos and Arnoldi methods). Solvers of the latter type are treated in more detail the next chapter, section 6.3.1. However, since the eigenproblem of the reduced assembled system is again rather small, the computational cost associated to solving its eigenproblem is rather low and the choice of eigensolver is not critical.

Once the eigensolutions of the reduced assembly are obtained, they can be judged against some criterion. For instance, the first eigenfrequency should be above a certain threshold or within a certain margin from experimental results in order to validate the model. Furthermore, the eigenmodes can again be expanded to obtain the modes at the physical DoF. These modes can be used to judge the dynamic behavior of the structure and also serve as a means for correlating the model with experimental results for validation purposes. The next section outlines comparison criteria that could be used to do so.

Finally, the eigensolutions could be used to synthesize the frequency response function (FRF) matrix of the structure. This is done as follows:

$$\boldsymbol{Y}(\omega) = \sum_{j=1}^{n_a} \frac{\boldsymbol{\phi}_j \boldsymbol{\phi}_j^T}{(\omega_j^2 - \omega^2) + 2i\zeta_j \omega_j \omega} \tag{4.37}$$

In this expression n_a is the number of DoF of the (unreduced) assembled system and ζ_j the modal damping ratio if damping is taken into account. Also, mass normalized modes $\boldsymbol{\phi}_j$ are assumed. Since a different FRF matrix exists for all frequencies ω, one usually only synthesizes the matrix for a limited number of DoF of interest in order to limit the required storage space. The so obtained FRFs may for instance be used to compare directly with measured FRFs or to use in a frequency based substructuring analysis in which the current assembly is coupled to some experimentally obtained component(s).

4.6.3 Time Integration

One analysis type where the use of reduced models especially pays off, is when performing time integration. In time integration, the transient time response of the system to some external excitation or initial condition is computed. This process is also referred to as time simulation or transient analysis.

Various techniques exist for performing the actual numerical (time) integration. Since the equations to be solved are second order ordinary differential equations, one option is to transform them to a state-space formulation to obtain a first order system of equations. This first order form could for instance be:

$$\begin{bmatrix} \tilde{\boldsymbol{M}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}(t) \\ \dot{\boldsymbol{q}}(t) \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{C}} & \tilde{\boldsymbol{K}} \\ -\boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}(t) \\ \boldsymbol{q}(t) \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{f}}(t) \\ \boldsymbol{0} \end{bmatrix} \quad \leftrightarrow \quad \boldsymbol{A}\dot{\boldsymbol{z}}(t) + \boldsymbol{B}\boldsymbol{z}(t) = \boldsymbol{p}(t) \tag{4.38}$$

Popular methods such as the Runge-Kutta schemes can then be employed to integrate the above equations of motion. An important disadvantage is that in this case the number of DoF doubles, which strongly increases computational cost.

Methods exist that are tailored for solving second order differential equations. One such method especially developed for structural dynamics is the Newmark method [140]. Although first published in 1959, this method is still often used today. Without going into the details of its derivation, this method can be summarized as follows. At time step $k+1$ in the integration process, the method first computes the so-called predictors based on the results from the previous time step k:

$$\begin{aligned} \hat{\boldsymbol{q}}_{k+1} &= \boldsymbol{q}_k + h\dot{\boldsymbol{q}}_k + (1/2 - \beta)h^2\ddot{\boldsymbol{q}}_k \\ \dot{\hat{\boldsymbol{q}}}_{k+1} &= \dot{\boldsymbol{q}}_k + (1-\gamma)h\ddot{\boldsymbol{q}}_k \end{aligned} \tag{4.39}$$

Here h is the time step and $\gamma \geq 0$ and $\beta \geq 0$ are parameters that can be chosen freely, leading to different stability and accuracy properties for the method. For details, see for instance [74]. Secondly, the accelerations for the current time step are computed by:

$$\boldsymbol{S}\ddot{\boldsymbol{q}}_{k+1} = \tilde{\boldsymbol{f}}_{k+1} - \tilde{\boldsymbol{C}}\hat{\dot{\boldsymbol{q}}}_{k} - \tilde{\boldsymbol{K}}\hat{\boldsymbol{q}}_{k+1}, \tag{4.40}$$

with:

$$\boldsymbol{S} = \tilde{\boldsymbol{M}} + h\gamma\tilde{\boldsymbol{C}} + h^2\beta\tilde{\boldsymbol{K}} \tag{4.41}$$

The matrix $\boldsymbol{S}$ is often referred to as the time-stepping matrix. Finally, the predictors are updated in a correction step as follows:

$$\begin{aligned} \boldsymbol{q}_{k+1} &= \hat{\boldsymbol{q}}_{k+1} + h^2\beta\ddot{\boldsymbol{q}}_{k+1} \\ \dot{\boldsymbol{q}}_{k+1} &= \hat{\dot{\boldsymbol{q}}}_{k+1} + h\gamma\ddot{\boldsymbol{q}}_{k+1} \end{aligned} \tag{4.42}$$

The above process is repeated for the desired number of time steps, that is, the desired duration of the time simulation divided by the step size. Different variations of the Newmark scheme have been developed, the most notable being the Hilber-Hughes-Taylor (HHT) method [91]. Through s slight modification of the Newmark scheme this method introduces numerical damping into the simulation for the high frequency range, while adding only very little damping in the lower frequencies, thereby preserving the accuracy. This numerical damping helps to suppress the response of eigenmodes with infinite or negative eigenfrequencies. The latter may be encountered in Dual Craig-Bampton reduced models due to so-called spurious modes (see section 2.5.4).

Note that although not relevant in this work, the Newmark method can easily be extended to non-linear systems. Furthermore, the last decade saw the development of new time integration methods, most notably methods with energy and momentum conservation built in the integration scheme [120, 117, 118].

When the time integration is completed, regardless of the method used, the results in terms of the generalized DoF can again be expanded to obtain time responses for the structure's physical DoF. However, if the simulated time series becomes too long, it might be advisable to expand only the results at certain DoF of interest to limit the required space for storing and handling the data.

4.7 Comparison & Correlation Methods

After the assembled system has been analyzed, the results are to be interpreted. In addition to "engineering judgement" and simple rules of thumb, it is useful to establish more objective criteria. Such methods are especially valuable when comparing the results from one model to another, or when correlating analysis outcomes to experimental results. In this section some basic comparison and correlation metrics are therefore outlined.

4.7.1 Eigenfrequency Comparison

Suppose that one has performed a modal analysis on the assembled system (system a) and one wishes to compare the results to some set of reference results (system b), for instance obtained from the full model or from measurements. The most basic comparison one can perform is to simply compare the eigenfrequencies of ω_a and ω_b of both systems.

In practice one usually compares the relative frequency error, i.e.:

$$\Delta\omega = \frac{|\omega_b - \omega_a|}{\omega_b} \cdot 100\% \tag{4.43}$$

For matching mode shapes, some maximum error can be defined (e.g. 5%). In some industries standards are established which state maximum frequency errors between models and measurements, for instance standards defined for spacecraft by NASA [139]. Should some modes in the frequency range of interest not satisfy this criterion, one either needs to enrich the reduced models or update the underlying FE models.

4.7.2 Modal Assurance Criterion

An eigenfrequency comparison is usually carried out in conjunction with a comparison of the associated mode shapes. Indeed, only when the mode shapes correspond reasonably well it makes sense to compare the difference in frequency of these modes, and vice versa. One often used method for comparing mode shape vectors is the *modal assurance criterion* (MAC) [3], which computes the vector correlation between a pair of mode shapes from systems a and b as follows:

$$\text{MAC} = \frac{|\boldsymbol{\phi}_a^T \boldsymbol{\phi}_b|^2}{\left(\boldsymbol{\phi}_a^T \boldsymbol{\phi}_a\right)\left(\boldsymbol{\phi}_b^T \boldsymbol{\phi}_b\right)} \tag{4.44}$$

Due to the scaling by the vector norms, the MAC value is a scalar in the range of zero to one. A MAC value of zero indicates no correlation (i.e. the vectors are orthogonal), while a value of one indicates perfect correlation (i.e. the vectors are parallel). Ideally, matching mode shapes should thus have a MAC value close to one, whereas cross-correlating different mode shapes should give a value close to zero. Furthermore, a number of remarks are in place regarding the MAC:

- Note that the MAC computes the true orthogonality between two mode shape vectors, while in theory they are in fact mass- and stiffness orthogonal. When a mass matrix is available, it is therefore more appropriate to use the mass-weighted MAC, sometimes referred to as MAC-M.

- When comparing multiple mode shape vectors for two systems, a MAC matrix is obtained. Along the diagonal one then finds the matching modes, while the off-diagonal terms show the correlation between non-matching modes.

- A MAC analysis can be used to correlate any type of deformation shapes as long as they are of the same dimension. Hence, not only mode shape vectors can be compared but the MAC can also be used to correlate static, harmonic and time response vectors.

- Historically, the MAC was derived to correlate modes obtained from a finite element model to experimentally identified mode shapes. In that case the mode shapes generally do not have the same dimension; two ways exist to overcome this. Firstly, the response of the finite element model can be extracted at the measurement locations, and the MAC is computed only at this limited number of DoF and no weighting with the mass matrix is applied. Secondly, experimental mode shapes can be expanded using the SEREP method [144] to allow correlation at all FE DoF, as well as weighting with the FE mass matrix.

- Similar to eigenfrequency comparisons, in some industries standards have been established for correlating experiments and models in terms of the MAC (for an example see again [139]). In the frequency range of interest, these standards dictate a certain minimum MAC value along the diagonal (e.g. 0.95) and a maximum value for the cross-terms (e.g. 0.1).
- Furthermore, the MAC can be used to compare mode shapes between reduced and full models. In this case one usually finds that the lowest modes are very well correlated while higher modes gradually show less and less correlation. It might therefore make sense to examine the 1-MAC value on a logarithmic scale, in order to distinguish the results for the highly correlated modes and clearly identify a drop in MAC value for the higher modes.

4.7.3 Substructure Modal Assurance Criterion

Over the years many variants of the modal assurance criterion have been proposed, an overview is given in [2]. In a dynamic substructuring analysis, one might be interested in which particular substructure modes are dominant in a certain global mode. In other words, one would like to somehow correlate the substructure modes and the global system modes. Therefore yet another variant of the MAC is proposed here, namely the *substructure modal assurance criterion* (SUMAC).

The SUMAC is a tool which can be used to check the modal contribution of a certain substructure to the global mode shapes, and is calculated as:

$$\text{SUMAC} = \frac{|\boldsymbol{\phi}_g^{(s)^T} \boldsymbol{\phi}_l^{(s)}|^2}{\left(\boldsymbol{\phi}_g^{(s)^T} \boldsymbol{\phi}_g^{(s)}\right)\left(\boldsymbol{\phi}_l^{(s)^T} \boldsymbol{\phi}_l^{(s)}\right)} \tag{4.45}$$

Here, $\boldsymbol{\phi}_g^{(s)}$ is the trace of the global mode shape on the DoF of substructure s and $\boldsymbol{\phi}_l^{(s)}$ is the local, uncoupled mode shape of that substructure. The trace of the global mode is obtained by localization of the substructure DoF using the assembly matrix $\boldsymbol{L}$ or $\boldsymbol{B}$, depending on the type of reduced model used.

A SUMAC value close to one indicates a strong contribution of the local mode to the global mode shape, meaning that the local substructure dynamics are important in that global mode shape. On the other hand, a SUMAC value close to zero indicates that the substructure is only quasi-statically participating in the global mode shape. One can imagine that the accuracy of the local mode with a high contribution in the global mode, has a significant effect on the accuracy of the global mode. Hence, the SUMAC can, in an a posteriori fashion, help identifying the dominant substructure modes in the global dynamic behavior, enabling selection of the relevant modes in the component reduction basis and possibly discarding less relevant ones.

Conversely, when dominant substructure modes are not, or only with low accuracy, contained in the reduced substructure model, this will incur errors in the global dynamic behavior. Obviously, this cannot be identified with the SUMAC. Instead, the error estimation methods developed in the next chapter should be applied.

4.7.4 Comparison of Time Integration Results

Finally, this subsection addresses the comparison of time integration results, for instance obtained from a full and reduced model. As will be discussed next, this is always a difficult task and different comparison criteria can be applied. A number of options are outlined below.

MAC analysis at each time step One idea for correlating time integration results is to simply perform a MAC analysis at every time step. In case both time responses are very similar, the deformation shapes at each time step should correlate well. However, small periodicity errors are introduced by the time integration scheme and in addition frequency differences usually exist between a full and reduced model. Over time, these frequency errors accumulate leading phase shifts between the different time signals. Analyzing these data sets using the MAC then easily leads to false conclusions on their accuracy.

Time-response assurance criterion One way to overcome this issue is as follows. Instead of computing the MAC value at every time step for all DoF simultaneously, one could also compute a MAC between the complete time series for one single DoF. Hence, one compares two time responses of the same DoF as if they were shape vectors. This method is called the *time response assurance criterion* (TRAC) [11]. The idea behind this method is that although the phase shifts may change the signal, its shape remains similar. However, it creates a correlation value per DoF and should be properly weighted to produce a meaningful statement on the accuracy of the global response.

POD analysis of time series A more consistent comparison of time series can be made using the *proper orthogonal decomposition* (POD) method. The POD is a mathematical data analysis method for efficient analysis of complex data and is explained in more detail in appendix D. In short, using the POD a time signal can be decomposed into *proper orthogonal modes* (POMs) and *proper orthogonal values*, which respectively describe the dominant modes in the signal and their energy level. The POMs and POVs can be treated similarly to the eigenmodes and eigenfrequencies of the system: POMs can be compared through correlation via the MAC, while the POVs are compared by computing the relative error.[1] Close correspondence in the POMs and POVs obtained from two sets of time series then indicates good correlation between those time series.

Energy based comparison Finally another comparison criterion, that is slightly different from the above methods, is to compare energies in the system. One could for instance track in time the deformation energy, kinetic energy, energy dissipated by damping (if

---

[1] Note it is advisable to scale the relative error on the POVs by the relative energy captured by the corresponding POM, to take into account that the relative energy captured by the POMs, and so their importance, diminishes quickly for higher POMs.

applicable) or the total energy in the system. By doing so, a global comparison criterion is obtained that can be easily interpreted. Indeed, when comparing time responses of reduced and full models, it is expected that the deformation energy in the reduced model is always less than in the full model due to the stiffening induced by the model reduction. Furthermore, energies in the total system can easily be broken down into substructure contributions, yielding insight in which component causes the largest discrepancies.

4.8 Summary

Two different topics were addressed in this chapter: sections 4.2 to 4.5 were concerned with the issue of interface reduction, while sections 4.6 and 4.7 considered the analysis of the assembled system.

Interface reduction was introduced as a method to further decrease the number of DoF of assembled system, which is relevant for systems consisting of many components and consequently carry many boundary DoF in comparison to generalized internal DoF. Interface reduction may then be applied to obtain a truly compact set of equations and maximize computational efficiency; section 4.2 outlined different options for doing so.

The most straightforward approach was to simply assume the interface to behave as a locally rigid region. As was shown in section, this interface rigidification can be applied a priori on a substructure level. Generalizations of this method were introduced in sections 4.4 and 4.5, where modal reduction of the interface was treated for displacement and force interface DoF, respectively. In those methods, the internal substructure DoF were statically condensed to the interface such that interface modes could be calculated. A truncated set of interface modes was subsequently used to reduce the original interface DoF. Furthermore, it was shown how these interface modes could be used to compact the component reduction bases to minimize memory and storage requirements.

Section 4.6 briefly treated different types of analysis that can be applied to the assembled reduced model, namely harmonic and modal analysis as well as time integration. Thereafter, it was explained in section 4.7 how the results of such analyses can be correlated between different (reduced) models and/or between models and experiments. Most notably, the modal assurance criterion was discussed, which is a simple but powerful method for correlating (mode) shape vectors. Also, a variant of the MAC was proposed, termed the substructure modal assurance criterion, with the purpose of correlating substructure modes with global system modes. Finally, the non-trivial issue of comparing time series from different models was addressed. A number of options were discussed, each with their own pros and cons.

5

Error Estimation & Adaptive Model Reduction

Make everything as simple as possible,
but not simpler.

(Albert Einstein)

5.1 Introduction

In the previous chapters it has been outlined how to obtain reduced component models, assemble these, and subsequently further reduce their size using interface reduction. Through this approach, compact models can be created for structural dynamic analysis of complex built-up structures. One important question that up to now however remained unaddressed is how accurate these reduced models actually are. Indeed, to confidently use these reduced models in practice requires insight in their accuracy.

Both the component and interface reduction incur an error on the total model. In these procedures, physical DoF are replaced by generalized DoF through a truncated set of deformation shapes, leading to an approximation of the dynamic behavior. The accuracy of this approximation depends on the number of deformation shapes, or modes in short, used to generate the model: the more modes the higher the accuracy. The computational efficiency of the reduced model is inversely related to the number of modes: more modes means more DoF and less compact models.

Finding the right balance between accuracy and efficiency is not trivial and is in practice often done using simple rules of thumb. One often applied method is frequency selection where all modes up to a certain frequency are included. In order to test the accuracy of the so obtained reduced system, its response can be compared to that of the unreduced system. When the accuracy proves to be unsatisfactory, more modes can be iteratively included.

However, two difficulties are associated with this approach. Firstly, obtaining the reference solution from the unreduced system is in general (very) computationally expensive. Secondly, no insight is gained in which components to enrich in order to improve the accuracy in the most effective way.

In this chapter a more systematic approach for quantification of the accuracy of the reduced model is therefore developed. This is done on the basis of *error estimation* methods. Thereby it is assumed that the full finite element model is an accurate representation of the actual structure, for instance through validation with dynamic measurements (see appendix B for an example). Hence, the goal is to quantify the error arising from the component and interface reduction, based on which effective enrichment schemes can be derived. As such, this chapter relates to the other chapters of part I as indicated in figure 5.1.

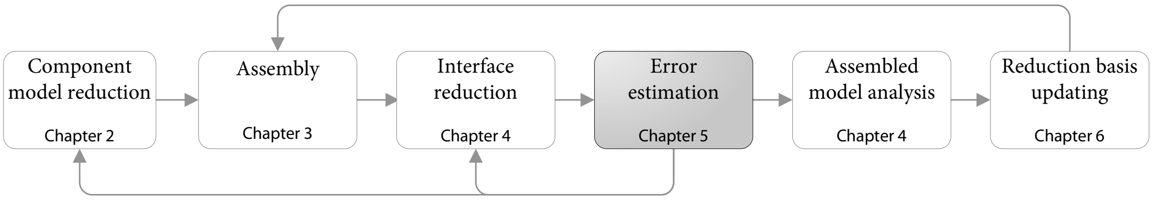


Figure 5.1: Current chapter in relation to other chapters in part I of this thesis.

5.2 Aspects of Error Estimation

Before diving into the theory of error estimation methods for reduction methods, this section briefly discusses two important aspects: firstly the similarity between errors arising from discretization and reduction, and secondly the notions of a priori and a posteriori error estimation. Thereafter, an outline for the remainder of this chapter is sketched.

5.2.1 Discretization & Reduction Errors

The problem of determining the accuracy of a reduced model is very similar to the problem that arises one step earlier in the modeling process, namely in the construction of an adequate finite element model. Due to the discretization of the continuous problem a difference between the analytical and discrete solutions is found, called the *discretization error*. This is illustrated in figure 5.2. Assuming an appropriate element type is chosen, this error is governed by the mesh size. Refining the mesh enhances the accuracy, or in other words reduces the error, but also increases the computational effort and required storage space.

Various error estimation techniques were developed in the field of finite element modeling to predict and control the discretization error, based on which (local) mesh refinement strategies have been be derived. See for instance [28, 18]. An important requirement for these methods is that they provide a conservative estimate of the error, i.e. the error should never be underestimated. In this context one therefore usually speaks of error bounds instead of the actual error, since the latter generally remains unknown.

Similar to the discretization error is the *reduction error* which arises due to the reduction of the FE model by truncating the number of modes. Analogous to mesh refinement of

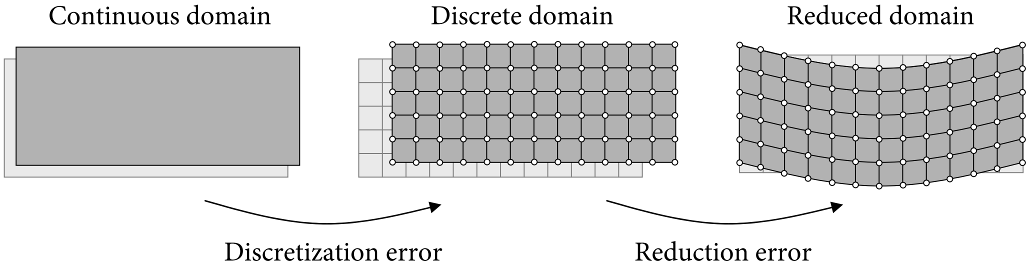


Figure 5.2: Sources of error in a substructured reduced model.

the FE model, the reduction error can be controlled through refinement of the reduction basis by adding modes. As outlined above, in order to do this effectively requires both a conservation error estimate as well as quantitative information on its distribution across the different component models in the assembly. Two types of error estimation exist, namely a priori and a posteriori estimation; both are briefly described in the next subsection.

5.2.2 A Priori Versus A Posteriori Error Estimates

Two classes of error estimation methods are defined in the literature: *a priori* and *a posteriori* methods, referring to when in the modeling and analysis process the error estimation is performed. Both are briefly described next.

A priori error estimation is used to predict a certain error before the actual computation is performed. The error however is expressed in terms of constants which depend on the exact solution. The resulting estimates do not quantify error bounds but only provide information on the convergence and the stability properties of the approximate model and the asymptotic behavior of the error if the refinement parameters are changed [18].

On the other hand, a posteriori error estimation is used to give both qualitative and quantitative measures of the error after the actual computation, using only the approximate solutions obtained from the reduced system. This type of estimation is able to provide both global and local information. An example of the former is an error bound expressed in terms of some global norm (e.g. energy), while local information can for instance be the error in displacements or stresses at a specific target location in the model. From the advantages described above it is understood that a posteriori estimation has become a popular method in error estimate and refinement strategies [28].

5.2.3 Chapter Outline

Given the discussion above, the focus of this chapter is the development of an a posteriori error estimation method for reduced substructured models. Indeed, error estimation techniques used for mesh refinement are seldom applied to component mode synthesis problems; in the literature only a few examples were found [102, 103, 104, 203]. The main ideas put forward in this chapter are based on these publications, but the applied mathematics are adapted and simplified. More specifically, the original functional analysis notation is translated to the algebraic notation as used throughout this thesis.

Using the proposed concepts, error estimates can be derived for the different types of dynamic analysis discussed in the previous chapter, section 4.6. Based on these error estimates, refinement strategies can be formulated. These allow to iteratively refine the component reduction bases until a satisfactory result is obtained (i.e. the error estimate drops below a certain threshold), a process that will be referred to as *adaptive model reduction*.

The remainder of this chapter is structured in the following way. Next, section 5.3 casts the expressions for the reduced system in the format needed to enable error estimation. The error estimates for systems excited by an external harmonic force are derived in section 5.4. Based on these results, section 5.5 describes an error estimate for the very similar time-stepping problem encountered in time integration schemes. Thereafter, error estimates for global eigensolution approximations, i.e. approximations of eigenfrequencies and -modes of the assembled system, are developed in section 5.6. Refinement schemes based on the error estimates are proposed in section 5.7, while section 5.8 finally provides a summary of the methods.

5.3 Alternative Reduced System Description

The error estimation methods in this chapter are developed for an assembled system built from Craig-Bampton (CB) reduced component models with additional reduction of the interface DoF. In this section, the expressions of the assembled system, as developed in the previous chapters, are rewritten in order to enable a posteriori error estimation.

To summarize, an assembly of CB reduced components was in the previous chapters obtained as follows. First, the component models are reduced using using the CB reduction basis in eq. (2.55) obtained in section 2.5.2. Thereafter, they are collected in a block diagonal form and assembled using the primal transformation (3.14) derived in section 3.3.2. Finally, interface reduction is applied as described in section 4.4 using the interface reduction basis in eq. (4.12).

In this section, the order of these operations is slightly altered, namely:

1. First, the component models are transformed using only the static constraint modes of the CB reduction basis. This step is no reduction, but just a transformation where the substructure internal DoF are transformed to DoF relative to the static modes associated to the interface DoF.
2. Thereafter, the transformed component models are collected in a block diagonal form and assembled using the primal transformation.
3. Finally, the reduction of the substructure internal DoF is combined with the reduction of the global interface DoF, leading to a single reduction step.

In short, the main difference with the previous formulation of the assembled model is that here the application of the CB reduction basis is performed into two separate steps. To obtain the associated equations, the first step is to collect all substructure models in a (non-assembled) block form, as:

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{ib} \\ \boldsymbol{M}_{bi} & \boldsymbol{M}_{bb} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_i \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib} \\ \boldsymbol{K}_{bi} & \boldsymbol{K}_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_b \end{bmatrix} \tag{5.1}$$

Here the same block notation is used as in chapters 3 and 4, i.e. the submatrices are block diagonal containing the component submatrices. Next, the equations are transformed using the component static constraint modes. These are again gathered in a block form, leading to the following transformation:

$$\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Psi}_c \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{u}}_i \\ \boldsymbol{u}_b \end{bmatrix} \tag{5.2}$$

As a result the following equations of motion are obtained:

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \bar{\boldsymbol{M}}_{ib} \\ \bar{\boldsymbol{M}}_{bi} & \bar{\boldsymbol{M}}_{bb} \end{bmatrix} \begin{bmatrix} \ddot{\bar{\boldsymbol{u}}}_i \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{0} \\ \boldsymbol{0} & \bar{\boldsymbol{K}}_{bb} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{u}}_i \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \bar{\boldsymbol{f}}_b \end{bmatrix}, \tag{5.3}$$

with the (block diagonal) submatrices defined as:[1]

$$\begin{aligned} \bar{\boldsymbol{K}}_{bb} &= \boldsymbol{K}_{bb} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} \\ \bar{\boldsymbol{M}}_{ib} &= \tilde{\boldsymbol{M}}_{bi}^T = \boldsymbol{M}_{ii}\boldsymbol{\Psi}_c + \boldsymbol{M}_{ib} \\ \bar{\boldsymbol{M}}_{bb} &= \boldsymbol{M}_{bb} + \boldsymbol{\Psi}_c^T\boldsymbol{M}_{ii}\boldsymbol{\Psi}_c + \boldsymbol{\Psi}_c^T\boldsymbol{M}_{ib} + \boldsymbol{M}_{bi}\boldsymbol{\Psi}_c \\ \bar{\boldsymbol{f}}_b &= \boldsymbol{f}_b + \boldsymbol{\Psi}_c^T\boldsymbol{f}_i \end{aligned} \tag{5.4}$$

Note that the $\bar{\boldsymbol{u}}_i$ correspond to the dynamic response of the internal DoF relative to the static displacements caused by static constraint modes; furthermore since all static modes are used the above transformation is exact. The transformed equations of motion are still unassembled, assembly is achieved by applying the primal transformation derived in section 3.3.2:

$$\begin{bmatrix} \bar{\boldsymbol{u}}_i \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}_b \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{u}}_i \\ \boldsymbol{u}_\gamma \end{bmatrix}, \tag{5.5}$$

leading to an assembled, unreduced set of equations with a block diagonal stiffness matrix:

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \bar{\boldsymbol{M}}_{ib}\boldsymbol{L}_b \\ \boldsymbol{L}_b^T\bar{\boldsymbol{M}}_{bi} & \boldsymbol{L}_b^T\bar{\boldsymbol{M}}_{bb}\boldsymbol{L}_b \end{bmatrix} \begin{bmatrix} \ddot{\bar{\boldsymbol{u}}}_i \\ \ddot{\boldsymbol{u}}_\gamma \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}_b^T\bar{\boldsymbol{K}}_{bb}\boldsymbol{L}_b \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{u}}_i \\ \boldsymbol{u}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{L}_b^T\bar{\boldsymbol{f}}_b \end{bmatrix} \tag{5.6}$$

In the derivation of the error estimation methods, the above assembled equations will be considered as the reference system. The benefit of this formulation is that the stiffness matrix is block diagonal, a property that will be used in subsequent derivations. As a result, the interface can be considered as an additional domain to the internal component domains.

Whereas in dynamic substructuring one usually considers a physical component (internal plus boundary DoF) as one domain, here only its internal DoF are defined as one domain. An additional domain is thereby defined, namely the global interface domain (i.e. the "sum" of all interfaces). As a result, an assembly of n components can now be seen as consisting of $n+1$ domains, which is illustrated by expanding the block notation of the stiffness matrix:

$$\bar{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{K}_{ii}^{(1)} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{ii}^{(2)} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \ddots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{ii}^{(n)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}_b^T\bar{\boldsymbol{K}}_{bb}\boldsymbol{L}_b \end{bmatrix} \quad \text{and} \quad \bar{\boldsymbol{u}} = \begin{bmatrix} \bar{\boldsymbol{u}}_i^{(1)} \\ \bar{\boldsymbol{u}}_i^{(2)} \\ \vdots \\ \bar{\boldsymbol{u}}_i^{(n)} \\ \boldsymbol{u}_\gamma \end{bmatrix} \tag{5.7}$$

---

[1] Note that these matrices correspond to the reduced matrices of the Guyan approach.

The interface is simply regarded as domain $n+1$. Similarly, the mass matrix can be expanded to reveal the coupling terms between the internal and boundary domains:

$$\bar{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{M}_{ii}^{(1)} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \bar{\boldsymbol{M}}_{ib}^{(1)}\boldsymbol{L}_b^{(1)} \\ \boldsymbol{0} & \boldsymbol{M}_{ii}^{(2)} & \boldsymbol{0} & \boldsymbol{0} & \bar{\boldsymbol{M}}_{ib}^{(2)}\boldsymbol{L}_b^{(2)} \\ \boldsymbol{0} & \boldsymbol{0} & \ddots & \boldsymbol{0} & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}_{ii}^{(n)} & \bar{\boldsymbol{M}}_{ib}^{(n)}\boldsymbol{L}_b^{(n)} \\ \boldsymbol{L}_b^{(1)^T}\bar{\boldsymbol{M}}_{bi}^{(1)} & \boldsymbol{L}_b^{(2)^T}\bar{\boldsymbol{M}}_{bi}^{(2)} & \cdots & \boldsymbol{L}_b^{(n)^T}\bar{\boldsymbol{M}}_{bi}^{(n)} & \boldsymbol{L}_b^T\bar{\boldsymbol{M}}_{bb}\boldsymbol{L}_b \end{bmatrix} \tag{5.8}$$

In compact notation this transformed version of the equations of motion will be written as:

$$\bar{\boldsymbol{M}}\ddot{\bar{\boldsymbol{u}}} + \bar{\boldsymbol{K}}\bar{\boldsymbol{u}} = \bar{\boldsymbol{f}} \tag{5.9}$$

In the last step the actual reduction is performed, both of the internal domains as the boundary DoF. The former are reduced using the CB fixed interface modes of section 2.4.3, while the latter are approximated using the interface displacement modes obtained in section 4.4. Combining these mode sets in a block diagonal matrix gives the following reduction basis:

$$\begin{bmatrix} \bar{\boldsymbol{u}}_i \\ \boldsymbol{u}_\gamma \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{\Phi}_i & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}_\gamma \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{\eta}_\gamma \end{bmatrix} = \boldsymbol{R}\boldsymbol{q} \tag{5.10}$$

Application of this reduction finally leads to the Craig-Bampton reduced assembled system:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Phi}_i^T\bar{\boldsymbol{M}}_{ib}\boldsymbol{L}_b\boldsymbol{\Phi}_\gamma \\ \boldsymbol{\Phi}_\gamma^T\boldsymbol{L}_b^T\bar{\boldsymbol{M}}_{bi}\boldsymbol{\Phi}_i & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_i \\ \ddot{\boldsymbol{\eta}}_\gamma \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Omega}_i^2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Omega}_\gamma^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{\eta}_\gamma \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_i^T\boldsymbol{f}_i \\ \boldsymbol{\Phi}_\gamma^T\boldsymbol{L}_b^T\tilde{\boldsymbol{f}}_b \end{bmatrix} \tag{5.11}$$

Since the ingredients in the reduction bases have not changed, these equations are exactly equal to those obtained in section 4.4, equation (4.15). The above equation can be cast in a more compact notation as:

$$\tilde{\boldsymbol{M}}\ddot{\boldsymbol{q}} + \tilde{\boldsymbol{K}}\boldsymbol{q} = \tilde{\boldsymbol{f}} \tag{5.12}$$

Finally it is noted that if one chooses to leave the interface unreduced, i.e. $\boldsymbol{\Phi}_\gamma = \boldsymbol{I}$, the system matrices in (5.11) are equal to those obtained using the standard CB method.

After performing any type of structural dynamic analysis on the assembled reduced system of equation (5.12), the solution can be expanded to the space of the physical DoF using the reduction basis in (5.10). Since the reduction is an approximation the dynamic equilibrium is never satisfied exactly, giving rise to a force residual. This force residual can be calculated by substitution of the expanded reduced solution into the unreduced, but transformed equations of motion in (5.9):

$$\boldsymbol{r} = \bar{\boldsymbol{f}} - \bar{\boldsymbol{M}}\boldsymbol{R}\ddot{\boldsymbol{q}} - \bar{\boldsymbol{K}}\boldsymbol{R}\boldsymbol{q} \tag{5.13}$$

Physically, the force residual can be interpreted as the constraining forces needed to restrict the system to its reduced displacement space. The magnitude of the residual depends on how well the contained modes describe the full system response.

5.4 Error Estimation for Harmonic Solutions

One type of analysis that is regularly encountered in structural dynamics is the harmonic analysis. As explained in section 4.6.1, for a system excited by a harmonic force it holds for the steady state solution that $\ddot{\boldsymbol{q}} = -\omega_e^2 \boldsymbol{q}$. Hence, one can define the dynamic stiffness of the unreduced system as

$$\bar{\boldsymbol{Z}} = \bar{\boldsymbol{K}} - \omega_e^2 \bar{\boldsymbol{M}}, \tag{5.14}$$

such that the force residual can be written as:

$$\boldsymbol{r} = \bar{\boldsymbol{f}} - \bar{\boldsymbol{Z}} \boldsymbol{R} \boldsymbol{q} \tag{5.15}$$

This residual can be allocated to the different domains:

$$\boldsymbol{r} = \mathrm{col}\left(\boldsymbol{r}_i^{(1)}; \ldots; \boldsymbol{r}_i^{(n)}; \boldsymbol{r}_\gamma\right) \tag{5.16}$$

An important property of this residual is its orthogonality with respect to the reduction basis, both on local (domain) and global level:

$$\boldsymbol{R}^T \boldsymbol{r} = \boldsymbol{0} \quad \text{and} \quad \boldsymbol{R}^{(s)^T} \boldsymbol{r}^{(s)} = \boldsymbol{0} \tag{5.17}$$

This property is often referred to as the Galerkin orthogonality principle, which simply means that the force residual exists only in the space spanned by the discarded modes. By definition the residual will therefore be zero if all the modes are retained; in that case $\boldsymbol{R}$ is merely a transformation. The fact that the residual vector can be split into domain contributions hints that it might be used as an indicator for the relative error produced by each domain. However, knowledge of the domain force residual does not directly relate to the size of the domain error.

The error is defined as the difference between the full and expanded reduced solution:

$$\boldsymbol{e} = \bar{\boldsymbol{u}} - \boldsymbol{R}\boldsymbol{q} = \bar{\boldsymbol{u}} - \tilde{\boldsymbol{u}} \tag{5.18}$$

Since the full solution creates no force residual, the residual of the reduced system can also be written as:

$$\boldsymbol{r} = \bar{\boldsymbol{Z}}\bar{\boldsymbol{u}} - \bar{\boldsymbol{Z}}\boldsymbol{R}\boldsymbol{q} = \bar{\boldsymbol{Z}}\left(\bar{\boldsymbol{u}} - \tilde{\boldsymbol{u}}\right) = \bar{\boldsymbol{Z}}\boldsymbol{e} \tag{5.19}$$

Clearly, there is a relation between the error and the residual; to compute the error however still requires the full solution. In the remainder of this section an upper bound will therefore be derived for the error on reduced solutions for the harmonic problem. To this end, the next subsection introduces the Dual Weighted Residual (DWR) method, which enables error estimates for quantities of interest. This approach is generalized in subsection 5.4.2 to obtain a global error bound in terms of the energy norm.

5.4.1 Goal Oriented Error Estimates – the Dual Weighted Residual Method

The dual-weight-residual (DWR) method is often used to obtain error estimates in the fields of structural optimization and parameter sensitivity analysis. It is also a common method for estimating the error resulting from discretization of a continuous domain by means of finite elements, which is then used as input for mesh refinement strategies. Less common is its use in model reduction, for one of the few examples in the literature see [135]. Here the DWR method will be employed to estimate the reduction error by finding a relation between the known force residual $\boldsymbol{r}$ and the unknown error $\boldsymbol{e}$.

When performing structural dynamic analyses one is often interested in a certain output quantity. In wind turbine engineering, this could for instance be the amplitude of the tip displacement of a wind turbine blade, or the stress at a particular location in the support structure. To measure the contribution of the reduction error to these quantities one can define the so-called linear goal oriented function, or target functional, as:

$$J(\bar{\boldsymbol{u}}) = \boldsymbol{a}^T \bar{\boldsymbol{u}} \tag{5.20}$$

Vector $\boldsymbol{a}$ is a selection vector which selects the predefined quantities of interest. Equally, this goal oriented function can be evaluated for the approximate reduced solution:

$$J(\tilde{\boldsymbol{u}}) = \boldsymbol{a}^T \tilde{\boldsymbol{u}} \tag{5.21}$$

Since the target functional is linear, subtracting the two gives the linear target functional of the error, also called goal oriented error:

$$J(\boldsymbol{e}) = J(\bar{\boldsymbol{u}}) - J(\tilde{\boldsymbol{u}}) = \boldsymbol{a}^T \left(\bar{\boldsymbol{u}} - \tilde{\boldsymbol{u}}\right) = \boldsymbol{a}^T \boldsymbol{e} \tag{5.22}$$

This goal oriented error cannot be evaluated without knowledge of the full solution. Recalling that $\bar{\boldsymbol{u}}$ is the solution to the harmonic problem with the dynamic stiffness matrix $\bar{\boldsymbol{Z}}$, one may rewrite the goal oriented error by:

$$\begin{aligned} J(\boldsymbol{e}) &= \boldsymbol{a}^T \left(\bar{\boldsymbol{u}} - \tilde{\boldsymbol{u}}\right) \\ &= \boldsymbol{a}^T \bar{\boldsymbol{Z}}^{-1} \bar{\boldsymbol{Z}} \left(\bar{\boldsymbol{u}} - \tilde{\boldsymbol{u}}\right) \\ &= \left(\bar{\boldsymbol{Z}}^{-1} \boldsymbol{a}\right)^T \boldsymbol{r} \\ &= \boldsymbol{d}^T \boldsymbol{r} \end{aligned} \tag{5.23}$$

Here use was made of the symmetry of the dynamic stiffness matrix and $\boldsymbol{d}$ is referred to as the solution of the *adjoint problem*:[2]

$$\bar{\boldsymbol{Z}} \boldsymbol{d} = \boldsymbol{a} \tag{5.24}$$

The adjoint solution $\boldsymbol{d}$ can be interpreted as the response of the system excited at the quantities of interest, defined in vector $\boldsymbol{a}$. Since the goal oriented error in the last line of eq. (5.23) is a vector product, it may be expressed as a summation over the individual domains:

$$J(\boldsymbol{e}) = \sum_{s=1}^{n+1} \boldsymbol{d}^{(s)^T} \boldsymbol{r}^{(s)} \quad \text{denoted by} \quad e_{\Sigma} \tag{5.25}$$

Here $\boldsymbol{r}^{(s)}$ is the force residual of domain s (note that for $s = n+1$ one has $\boldsymbol{r}_{\gamma}$), weighting by the trace of the adjoint solution $\boldsymbol{d}^{(s)}$ on the same domain this gives the domain error; their

---

[2] In the literature this problem is also referred to as the *dual problem*, but this terminology is not used here to avoid confusion with dual assembly techniques

summation yields the total goal oriented error e_Σ. Hence, the value of the domain error gives an indication of its relative contribution to the total goal oriented error.

The domain errors can be positive and negative, such that in the summation the error in one domain can counteract the error in another. An upper bound for the error can therefore be obtained by using the triangle inequality:

$$|J(\boldsymbol{e})| \leq \sum_{s=1}^{n+1} \left| \boldsymbol{d}^{(s)^T} \boldsymbol{r}^{(s)} \right| \quad \text{denoted by} \quad e_{\leq\Sigma} \tag{5.26}$$

Here the notation $e_{\leq\Sigma}$ is introduced to denote the upper bound for the goal oriented error.

However, to evaluate the goal oriented error one needs the solution to the adjoint problem. From the definition of the adjoint problem (5.24) it can be seen that this requires solving the full system. Obviously this would take similar computational effort as solving for the full solution $\bar{\boldsymbol{u}}$ which would allow the error $\boldsymbol{e}$ to be computed directly. Still, using an adjoint problem formulation is beneficial in case many load cases need to be solved. One would then want a reduced model to run all the load cases, but one could afford a full solution for the adjoint problem to estimate the accuracy.

Furthermore, in practice the adjoint solution is often approximated using a reduced model that is more refined than the one used to obtain $\tilde{\boldsymbol{u}}$. As a result the inequality in expression (5.26) becomes an approximation:

$$|J(\boldsymbol{e})| \leq \sum_{s=1}^{n+1} \left| \boldsymbol{d}^{(s)^T} \boldsymbol{r}^{(s)} \right| \approx \sum_{s=1}^{n+1} \left| \tilde{\boldsymbol{d}}^{(s)^T} \boldsymbol{r}^{(s)} \right| = \sum_{s=1}^{n+1} m_J^{(s)} \quad \text{denoted by} \quad e_{\approx\Sigma} \tag{5.27}$$

In this expression $\tilde{\boldsymbol{d}}$ is the approximate adjoint solution and $m_J^{(s)}$ an error indicator, collected in the vector $\boldsymbol{m}_J$, that is introduced to unify notations in the coming sections. It is emphasized that the approximate adjoint solution must be computed using a more refined model than the "nominal" model used to compute $\tilde{\boldsymbol{u}}$. Indeed, both $\tilde{\boldsymbol{u}}^{(s)}$ and $\tilde{\boldsymbol{d}}^{(s)}$ can be written as a modal superposition of the modes contained in the domain reduction basis. Due to the Galerkin orthogonality, the domain residual is orthogonal to the part of $\tilde{\boldsymbol{d}}^{(s)}$ that lives in the space of nominal reduction basis. In order to obtain meaningful results, the reduction bases used to computed the approximate adjoint solution should therefore contain additional modes.

5.4.2 Global Energy Norm Based Error Estimate

The main difficulty associated with the goal oriented error estimates derived above is that they require solving the adjoint problem. In this subsection the DWR method is therefore generalized to obtain a global error estimate that does not require any adjoint solution. In order to do so, first the following terms are defined to unify the notation for the internal and interface domains:

$$\boldsymbol{\Phi}_l^{(s)} = \begin{cases} \boldsymbol{\Phi}_i^{(s)} & \text{for} \quad s = 1 \dots n \\ \boldsymbol{\Phi}_\gamma & \text{for} \quad s = n+1 \end{cases} \quad \text{and} \quad \boldsymbol{M}^{(s)} = \begin{cases} \boldsymbol{M}_{ii} & \text{for} \quad s = 1 \dots n \\ \boldsymbol{L}_b^T \bar{\boldsymbol{M}}_{bb} \boldsymbol{L}_b & \text{for} \quad s = n+1 \end{cases} \tag{5.28}$$

Here it is assumed that $\boldsymbol{\Phi}_l^{(s)}$ contains the k_s lowest eigenmodes of each domain and normalized with respect to their corresponding mass matrix $\boldsymbol{M}^{(s)}$. Note that $\boldsymbol{\Phi}_l^{(s)}$ are the eigenmodes of $\left(\boldsymbol{K}^{(s)}, \boldsymbol{M}^{(s)}\right)$ as used earlier for the Craig-Bampton method and interface reduction. Using these definitions, an operator $\boldsymbol{P}^{(s)}$ can be defined for each domain as:

$$\boldsymbol{P}^{(s)} = \boldsymbol{\Phi}_l^{(s)} \boldsymbol{\Phi}_l^{(s)^T} \boldsymbol{M}^{(s)} \tag{5.29}$$

Next, use is made of the fact that the domain residual is orthogonal to the associated domain reduction basis (Galerkin orthogonality), such that the goal oriented error in eq. (5.26) can be formulated as:

$$|J(\boldsymbol{e})| \leq \sum_{s=1}^{n+1} \left| \boldsymbol{d}^{(s)^T} \boldsymbol{r}^{(s)} \right| = \sum_{s=1}^{n+1} \left| \left(\left(\boldsymbol{I} - \boldsymbol{P}^{(s)} \right) \boldsymbol{d}^{(s)} \right)^T \boldsymbol{r}^{(s)} \right| \tag{5.30}$$

The above expression can be rewritten using the so-called Cauchy-Schwarz inequality. For two arbitrary vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ this inequality states:

$$\left| \boldsymbol{x}^T \boldsymbol{y} \right| \leq \sqrt{\boldsymbol{x}^T \boldsymbol{x}} \sqrt{\boldsymbol{y}^T \boldsymbol{y}} \tag{5.31}$$

In this case, one can define that:

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{M}^{(s)^{1/2}} \left(\boldsymbol{I} - \boldsymbol{P}^{(s)} \right) \boldsymbol{d}^{(s)} \\ \boldsymbol{y} &= \boldsymbol{M}^{(s)^{-1/2}} \boldsymbol{r}^{(s)} \end{aligned} \tag{5.32}$$

It therefore follows that:

$$\begin{aligned} \sqrt{\boldsymbol{x}^T \boldsymbol{x}} &= \sqrt{\left(\left(\boldsymbol{I} - \boldsymbol{P}^{(s)} \right) \boldsymbol{d}^{(s)} \right)^T \boldsymbol{M}^{(s)} \left(\boldsymbol{I} - \boldsymbol{P}^{(s)} \right) \boldsymbol{d}^{(s)}} = \left\| \left(\boldsymbol{I} - \boldsymbol{P}^{(s)} \right) \boldsymbol{d}^{(s)} \right\| \\ \sqrt{\boldsymbol{y}^T \boldsymbol{y}} &= \sqrt{\boldsymbol{r}^{(s)^T} \boldsymbol{M}^{(s)^{-1}} \boldsymbol{r}^{(s)}} = \left\| \hat{\boldsymbol{r}}^{(s)} \right\| \end{aligned} \tag{5.33}$$

Here $\| \star \|$ is the L^2 norm $\sqrt{\star^T \boldsymbol{M}^{(s)} \star}$, and it has been defined that $\hat{\boldsymbol{r}}^{(s)} = \boldsymbol{M}^{(s)^{-1}} \boldsymbol{r}^{(s)}$. Hence, the force residual is translated to an acceleration residual by pre-multiplication with the inverse mass matrix. Note that in practice it might be undesirable to factorize the domain mass matrix, so instead one could employ either of the two following methods to approximate the acceleration residual. Firstly, one can use a lumped mass approximation, leading to diagonal domain mass matrices. It is speculated that for the internal domains this might be a reasonable approximation as the domain mass matrices are diagonally dominant. For the interface domain however, where coupling terms with the other domains are present, this approximation might be too crude. Secondly, one might consider evaluating the acceleration residual using a more refined reduced model, similar to the approximate evaluation of the goal oriented error in the previous section.

Using the above definitions, one can write the goal oriented error in eq. (5.30) as follows:

$$|J(\boldsymbol{e})| \leq \sum_{s=1}^{n+1} \left\| \left(\boldsymbol{I} - \boldsymbol{P}^{(s)} \right) \boldsymbol{d}^{(s)} \right\| \left\| \hat{\boldsymbol{r}}^{(s)} \right\| \tag{5.34}$$

Next, the projector in the first term on the right hand side is rewritten using the spectral expansion of the identity matrix (see e.g. [74] for the derivation of the spectral expansions):

$$\boldsymbol{I} - \boldsymbol{P}^{(s)} = \left(\boldsymbol{\Phi}_l^{(s)} \boldsymbol{\Phi}_l^{(s)^T} + \boldsymbol{\Phi}_h^{(s)} \boldsymbol{\Phi}_h^{(s)^T} \right) \boldsymbol{M}^{(s)} - \boldsymbol{\Phi}_l^{(s)} \boldsymbol{\Phi}_l^{(s)^T} \boldsymbol{M}^{(s)} = \boldsymbol{\Phi}_h^{(s)} \boldsymbol{\Phi}_h^{(s)^T} \boldsymbol{M}^{(s)} \tag{5.35}$$

Here $\boldsymbol{\Phi}_h^{(s)}$ contains the domain's higher (i.e. discarded) modes; substitution of this expansion in (5.34) and applying the definition of the L^2 norm then gives:

$$|J(\boldsymbol{e})| \leq \sum_{s=1}^{n+1} \left(\boldsymbol{d}^{(s)^T} \boldsymbol{M}^{(s)} \boldsymbol{\Phi}_h^{(s)} \boldsymbol{\Phi}_h^{(s)^T} \boldsymbol{M}^{(s)} \boldsymbol{d}^{(s)} \right)^{1/2} \left\| \hat{\boldsymbol{r}}^{(s)} \right\| \tag{5.36}$$

To obtain this result it was assumed that the domain eigenfrequencies are ordered in a monotonically increasing sequence

$$0 \leq \omega_1^{(s)} \leq \ldots \leq \omega_{n_s}^{(s)}, \tag{5.37}$$

with n_s the total number of DoF in domain s. Hence one can conservatively scale the goal oriented error using the first discarded eigenvalue as follows:

$$|J(\boldsymbol{e})| \leq \sum_{s=1}^{n+1} \left(\boldsymbol{d}^{(s)^T} \boldsymbol{M}^{(s)} \boldsymbol{\Phi}_h^{(s)} \boldsymbol{\Omega}_h^{(s)^2} \boldsymbol{\Phi}_h^{(s)^T} \boldsymbol{M}^{(s)} \boldsymbol{d}^{(s)} \right)^{1/2} \frac{1}{\omega_{k_s+1}^{(s)}} \left\| \hat{\boldsymbol{r}}^{(s)} \right\| \tag{5.38}$$

Here $\boldsymbol{\Omega}_h$ is a diagonal matrix containing the eigenfrequencies associated to the discarded modes in $\boldsymbol{\Phi}_h$. Furthermore, for the bracketed term in (5.38) it is found that:

$$\begin{aligned} \boldsymbol{d}^{(s)^T} \boldsymbol{M}^{(s)} \boldsymbol{\Phi}_h^{(s)} \boldsymbol{\Omega}_h^{(s)^2} \boldsymbol{\Phi}_h^{(s)^T} \boldsymbol{M}^{(s)} \boldsymbol{d}^{(s)} &\leq \boldsymbol{d}^{(s)^T} \boldsymbol{M}^{(s)} \boldsymbol{\Phi}^{(s)} \boldsymbol{\Omega}^{(s)^2} \boldsymbol{\Phi}^{(s)^T} \boldsymbol{M}^{(s)} \boldsymbol{d}^{(s)} \\ &\leq \boldsymbol{d}^{(s)^T} \boldsymbol{K}^{(s)} \boldsymbol{d}^{(s)} = \left\| \boldsymbol{d}^{(s)} \right\|_K^2 \end{aligned} \tag{5.39}$$

Here $\boldsymbol{\Phi}^{(s)}$ contains all the domain modes and the spectral expansion of the domain stiffness matrix has been used in the second step. Hence one obtains:

$$|J(\boldsymbol{e})| \leq \sum_{s=1}^{n+1} \left\| \boldsymbol{d}^{(s)} \right\|_K \frac{1}{\omega_{k_s+1}^{(s)}} \left\| \hat{\boldsymbol{r}}^{(s)} \right\| \tag{5.40}$$

Once more the Cauchy-Schwarz and triangle inequalities can be used to separate the sum of products, to give:

$$|J(\boldsymbol{e})| \leq \left(\sum_{s=1}^{n+1} \left\| \boldsymbol{d}^{(s)} \right\|_K^2 \right)^{1/2} \left(\sum_{s=1}^{n+1} \frac{1}{\omega_{k_s+1}^{(s)^2}} \left\| \hat{\boldsymbol{r}}^{(s)} \right\|^2 \right)^{1/2} \tag{5.41}$$

Due to the block-diagonality (or decoupling) of the stiffness matrix of the transformed unreduced system, see eq. (5.6), it holds that:

$$\sum_{s=1}^{n+1} \left\| \boldsymbol{d}^{(s)} \right\|_K^2 = \|\boldsymbol{d}\|_K^2 \tag{5.42}$$

As a result, the goal oriented error is found as:

$$|J(\boldsymbol{e})| \leq \|\boldsymbol{d}\|_K \left(\sum_{s=1}^{n+1} \frac{1}{\omega_{k_s+1}^{(s)^2}} \left\| \hat{\boldsymbol{r}}^{(s)} \right\|^2 \right)^{1/2} \tag{5.43}$$

In order to obtain an estimate for the global error $\boldsymbol{e}$, the selection vector for the adjoint problem is chosen as $\boldsymbol{a} = \bar{\boldsymbol{K}}\boldsymbol{e}$. Due to this choice, the goal oriented error $J(\boldsymbol{e})$ becomes a global error estimate in terms of the energy norm:

$$|J(\boldsymbol{e})| = \left|\boldsymbol{a}^T \boldsymbol{e}\right| = \left|\boldsymbol{e}^T \bar{\boldsymbol{K}} \boldsymbol{e}\right| = \|\boldsymbol{e}\|_K^2 \tag{5.44}$$

Consequently applying eq. (5.43), the global error estimate is found as:

$$\|\boldsymbol{e}\|_K^2 \leq \|\boldsymbol{d}\|_K \left(\sum_{s=1}^{n+1} \frac{1}{\omega_{k_s+1}^{(s)^2}} \left\|\hat{\boldsymbol{r}}^{(s)}\right\|^2 \right)^{1/2} \tag{5.45}$$

Here $\boldsymbol{d}$ is now the solution to the adjoint problem $\bar{\boldsymbol{Z}}\boldsymbol{d} = \bar{\boldsymbol{K}}\boldsymbol{e}$. However, this estimate still depends on the full adjoint solution. The final step in this derivation is therefore to obtain a bound for the energy norm of the adjoint solution $\|\boldsymbol{d}\|_K$ in terms of the error $\boldsymbol{e}$. Therefore, the adjoint solution computed from eq. (5.24) is written using the spectral expansion of the global matrices:

$$\boldsymbol{d} = \bar{\boldsymbol{Z}}^{-1}\bar{\boldsymbol{K}}\boldsymbol{e} = \sum_{j=1}^{n_a} \left(\omega_j^2 - \omega_e^2\right)^{-1} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{K}}\boldsymbol{e} = \sum_{j=1}^{n_a} \omega_j^2 \left(\omega_j^2 - \omega_e^2\right)^{-1} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{M}}\boldsymbol{e} \tag{5.46}$$

In this expression n_a is the number of DoF of the assembled, unreduced model; *the eigenfrequencies ω_j and mode shapes $\boldsymbol{\phi}_j$ now also refer to this total model.* Next, the above expression is rewritten as follows:

$$\begin{aligned} \boldsymbol{d} &= \sum_{j=1}^{n_a} \left(\omega_j^2 - \omega_e^2\right)^{-1} \left(\omega_j^2 - \omega_e^2 + \omega_e^2\right) \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{M}}\boldsymbol{e} \\ &= \boldsymbol{e} + \sum_{j=1}^{n_a} \omega_e^2 \left(\omega_j^2 - \omega_e^2\right)^{-1} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{M}}\boldsymbol{e} = \boldsymbol{e} + \boldsymbol{d}_1 \end{aligned} \tag{5.47}$$

In the second step the spectral expansion of the identity matrix was used. Using the above result and the triangle inequality, the energy norm of the adjoint solution can now be expressed as:

$$\|\boldsymbol{d}\|_K \leq \|\boldsymbol{e}\|_K + \|\boldsymbol{d}_1\|_K \tag{5.48}$$

The final step is to obtain a bound for the last term. This can be achieved by writing the energy norm of this term and maximizing the result:

$$\begin{aligned} \|\boldsymbol{d}_1\|_K^2 &= \boldsymbol{e}^T \left(\sum_{j=1}^{n_a} \omega_e^2 \left(\omega_j^2 - \omega_e^2\right)^{-1} \bar{\boldsymbol{M}}\boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \right) \bar{\boldsymbol{K}} \left(\sum_{k=1}^{n_a} \omega_e^2 \left(\omega_k^2 - \omega_e^2\right)^{-1} \boldsymbol{\phi}_k \boldsymbol{\phi}_k^T \bar{\boldsymbol{M}} \right) \boldsymbol{e} \\ &= \boldsymbol{e}^T \left(\sum_{j=1}^{n_a} \omega_e^4 \left(\omega_j^2 - \omega_e^2\right)^{-2} \omega_j^2 \bar{\boldsymbol{M}}\boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{M}} \right) \boldsymbol{e} \\ &\leq \max_{\omega_j} \left(\frac{\omega_e^2}{\omega_j^2 - \omega_e^2} \right)^2 \boldsymbol{e}^T \left(\sum_{j=1}^{n_a} \omega_j^2 \bar{\boldsymbol{M}}\boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{M}} \right) \boldsymbol{e} = \max_{\omega_j} \left(\frac{\omega_e^2}{\omega_j^2 - \omega_e^2} \right)^2 \|\boldsymbol{e}\|_K^2 \end{aligned} \tag{5.49}$$

With this result the bound for the energy norm of the adjoint solution can be expressed as:

$$\|\boldsymbol{d}\|_K \leq (1 + S_\omega) \|\boldsymbol{e}\|_K \quad \text{with} \quad S_\omega = \max_{\omega_j} \left(\frac{\omega_e^2}{\omega_j^2 - \omega_e^2} \right) \tag{5.50}$$

Here S_ω is the so-called stability factor, i.e. an additional factor on the inequality in order for it to hold true. This stability factor requires knowing the eigenfrequency of the total system, that is closest to the excitation frequency. Finally, combining this result and equation (5.45), the error estimate in the energy norm becomes:

$$\|\boldsymbol{e}\|_K \leq (1 + S_\omega)\left(\sum_{s=1}^{n+1} \frac{1}{\omega_{k_s+1}^{(s)^2}} \left\|\hat{\boldsymbol{r}}^{(s)}\right\|^2\right)^{1/2} \quad \text{denoted by} \quad e_{\leq K} \tag{5.51}$$

To determine the relative domain contributions for use in refinement schemes, the following error indicators are introduced based on this estimate:

$$m_K^{(s)} = \frac{1}{\omega_{k_s+1}^{(s)^2}} \left\|\hat{\boldsymbol{r}}^{(s)}\right\|^2 \tag{5.52}$$

For convenience all domain indicators are collected in the vector $\boldsymbol{m}_K$. The error estimate $e_{\leq K}$ in eq. (5.51) can be interpreted as follows.

The energy norm represents the total elastic energy in the system caused by the displacement error. The above equation finds an upper bound for it by conservatively scaling the (acceleration) residual using the eigenfrequencies of the first modes outside the space in which the approximate solution is obtained. This scaling of the residual can be interpreted as taking into account that part of the inverse stiffness matrix which is outside the reduced space. Multiplication by the stability factor then gives the estimate for the error in the energy norm.

The stability factor can be recognized as the dynamic amplification factor of the harmonic response due to the global eigenfrequency that is closest to the excitation frequency. Obviously, this eigenfrequency is not calculated from the full model but is also approximated by the reduced model. In order for the error estimate to remain an upper bound, it is therefore important that this frequency is accurately predicted by the reduced model. Should this not be the case, for instance when the reduction bases are still very poor at the start of a refinement algorithm (see section 5.7), the error estimate no longer holds quantitatively but can still be used to estimate the relative domain contributions.

Finally, note that the error estimate may be scaled by the energy norm of the (approximate) solution to obtain relative energy norms of the error.

5.5 Error Estimation for Time-Stepping Solutions

In this section, an error estimate will be derived for the time-stepping problem. As the name suggests, this problem is solved in each time step of a time integration scheme, see section 4.6.3. Indeed, this problem is very similar to the harmonic problem and as a result, most of the developments of the previous section can be directly applied to the time-stepping problem.

The time-stepping problem for the popular Newmark time-integration scheme can be obtained by combining the equations for the prediction, acceleration and correction steps (i.e. eqs. (4.39), (4.40) and (4.42) in section 4.6.3). Discarding damping and, for the sake of illustration, assuming that the system was at rest at the previous time step, one finds that the problem to be solved has the following form:

$$\hat{\boldsymbol{S}}\boldsymbol{q}_k = \bar{\boldsymbol{f}}_k \tag{5.53}$$

Where:

$$\hat{\boldsymbol{S}} = \bar{\boldsymbol{K}} + \tau \bar{\boldsymbol{M}} \quad \text{with} \quad \tau = \frac{1}{h^2 \beta} \tag{5.54}$$

Hence, the time-stepping problem is very similar to the harmonic problem. Consequently, in order to derive an error estimate the same line of reasoning can be followed for both the goal oriented and the global error estimates. In fact, the goal error estimates can be one-to-one applied to the time-stepping problem. For the global error estimate in terms of the energy norm, the derivation in section 5.4.2 can be followed up to eq. (5.45):

$$\|\boldsymbol{e}\|_K^2 \leq \|\boldsymbol{d}\|_K \left(\sum_{s=1}^{n+1} \frac{1}{\omega_{k_s+1}^{(s)2}} \left\|\hat{\boldsymbol{r}}^{(s)}\right\|^2 \right)^{1/2} \tag{5.45}$$

Here $\boldsymbol{d}$ is again the unknown adjoint solution while $\hat{\boldsymbol{r}}$ now is the (acceleration) residual resulting from the time-stepping computation with the reduced model. To arrive at a global error estimate for the time-stepping problem the adjoint solution should again be eliminated. In this case, this can be easily achieved by proving that $\|\boldsymbol{e}\|_K \leq \|\boldsymbol{d}\|_K$ which is done by noting that the adjoint solution is now obtained from:

$$\hat{\boldsymbol{S}}\boldsymbol{d} = \bar{\boldsymbol{K}}\boldsymbol{e} \tag{5.55}$$

Hence, using spectral expansion and the definition of $\hat{\boldsymbol{S}}$, the energy norm of the adjoint solution can be written as:

$$\begin{aligned}
\|\boldsymbol{d}\|_K &= \boldsymbol{e}^T \bar{\boldsymbol{K}} \hat{\boldsymbol{S}}^{-1} \bar{\boldsymbol{K}} \hat{\boldsymbol{S}}^{-1} \bar{\boldsymbol{K}} \boldsymbol{e} \\
&= \boldsymbol{e}^T \bar{\boldsymbol{K}} \left(\sum_{j=1}^{na} \left(\omega_j^2 + \tau\right)^{-1} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \right) \bar{\boldsymbol{K}} \left(\sum_{k=1}^{na} \left(\omega_k^2 + \tau\right)^{-1} \boldsymbol{\phi}_k \boldsymbol{\phi}_k^T \right) \bar{\boldsymbol{K}} \boldsymbol{e} \\
&= \boldsymbol{e}^T \bar{\boldsymbol{K}} \left(\sum_{j=1}^{na} \omega_j^2 \left(\omega_j^2 + \tau\right)^{-2} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \right) \bar{\boldsymbol{K}} \boldsymbol{e} \\
&= \boldsymbol{e}^T \bar{\boldsymbol{K}} \left(\sum_{j=1}^{na} \omega_j^4 \left(\omega_j^2 + \tau\right)^{-2} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{M}} \right) \boldsymbol{e} \leq \|\boldsymbol{e}\|_K
\end{aligned} \tag{5.56}$$

In the last step it was used that $\tau > 0$, which can be seen from its definition in eq. (5.54). As a result, the error estimate for the time-stepping problem can be expressed as:

$$\|\boldsymbol{e}\|_K \leq \left(\sum_{s=1}^{n+1} \frac{1}{\omega_{k_s+1}^{(s)2}} \left\|\hat{\boldsymbol{r}}^{(s)}\right\|^2 \right)^{1/2} \tag{5.57}$$

Again the following domain error indicators are defined and gathered in the vector $\boldsymbol{m}_K$:

$$m_K^{(s)} = \frac{1}{\omega_{k_s+1}^{(s)2}} \left\|\hat{\boldsymbol{r}}^{(s)}\right\|^2 \tag{5.58}$$

Note that apart from the stability factor the obtained error bound is exactly similar to the one found for harmonic problems. This is expected as both problems only differ in the fact that the mass matrix is multiplied by a factor $\tau > 0$ in the time-stepping problem whereas it is multiplied by $-\omega_e^2 \leq 0$ in the harmonic problem. Due to this difference, the stability factor can be omitted in the error estimate for the former.

Finally, it is pointed out that the error bound in (5.57) gives an estimation of the reduction error incurred in a single time step. A propagation analysis is required in order to quantify the total reduction error that is accumulated during a time integration with multiple time steps. This is a complex problem that is out of the scope of this work.

5.6 Error Estimation for Global Eigensolutions

In addition to the common harmonic analysis and time integration, in practice an often seen type of dynamic analysis is the modal analysis (see section 4.6.2). By extending the theory of the previous sections, an error estimate can be derived for a global eigensolution (eigenfrequency and -mode of the assembled system) obtained from the reduced model without knowledge of the solution for the full model. The next subsection therefore first defines the error on the eigensolution, thereafter the actual estimates are derived in subsections 5.6.2 and 5.6.3. Throughout this section references will be made to appendix C where some steps in the derivation of the error estimates are worked out in detail. This is done to avoid too lengthy derivations in the main text.

5.6.1 Error & Force Residual

In section 5.4 the goal was to create a reduced model whose harmonic response converges towards that of the unreduced model. Hence the error was defined as the difference between those two responses. Here the goal is to iteratively improve a reduced model having an approximate eigensolution that should converge towards a selected/target eigensolution of the unreduced model. Since an eigensolution consists of an eigenfrequency and an eigenmode, two errors need to be defined.

The error on the eigenfrequency of eigensolution j, denoted by e_ω, is simply defined as the absolute difference between the exact eigenfrequency ω_j obtained from the unreduced model and the approximate one $\tilde{\omega}$ obtained from the reduced system:

$$e_\omega = \left|\tilde{\omega}^2 - \omega_j^2\right| \tag{5.59}$$

The error on the eigenmode is defined some what differently, namely as the part of the approximate eigenmode $\tilde{\boldsymbol{\phi}}$ that is (mass) orthogonal to the space of its exact counterpart $\boldsymbol{\phi}_j$, i.e.:

$$\boldsymbol{e}_\phi = \tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}} = \left(\boldsymbol{I} - \boldsymbol{P}_j\right) \tilde{\boldsymbol{\phi}} \tag{5.60}$$

Here mass normalized modes are assumed and $\boldsymbol{P}_j$ is an operator on the global level, similar to the projector used on the domain level in section 5.4.2. Note that in practice one might include multiple eigenmodes in this projector, for instance when several modes are very closely spaced. Note that the error estimates that will be derived in this section are equally valid in that case.

Since the eigensolution of the reduced system is an approximation of the exact solution it does not satisfy the eigenproblem of the full system. Hence, when the approximate eigensolution is substituted in the full eigenproblem a force residual is found:

$$\boldsymbol{r} = \bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} - \tilde{\omega}^2\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \tag{5.61}$$

Note that the force residual can again be split into domain contributions similar to eq. (5.16). Furthermore, in order to converge to the correct global eigensolution, both the error on the eigenvalue and -vector need to be below a certain tolerance.

For the derivation of the error estimates for the approximate eigensolution, the starting point is again the target functional introduced in section 5.4.1. This target functional can be evaluated for the approximate mode $\tilde{\boldsymbol{\phi}}$ as:

$$\begin{aligned} J\left(\tilde{\boldsymbol{\phi}}\right) = \boldsymbol{a}^T\tilde{\boldsymbol{\phi}} &= \boldsymbol{d}^T\left(\bar{\boldsymbol{K}} - \omega_j^2\bar{\boldsymbol{M}}\right)\tilde{\boldsymbol{\phi}} \\ &= \boldsymbol{d}^T\bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} - \omega_j^2\boldsymbol{d}^T\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} = \boldsymbol{d}^T\boldsymbol{r} \end{aligned} \tag{5.62}$$

In this expression ω_j^2 is the exact eigenfrequency of the unreduced assembled system, and $\boldsymbol{d}$ is again the solution to the adjoint problem which for the eigenproblem becomes:

$$\left(\bar{\boldsymbol{K}} - \omega_j^2\bar{\boldsymbol{M}}\right)\boldsymbol{d} = \boldsymbol{a} \tag{5.63}$$

Next, the projector $\boldsymbol{P}$ is defined as the block diagonal equivalent of the domain projectors $\boldsymbol{P}^{(s)}$ defined in eq. (5.29). This allows to split the adjoint solution $\boldsymbol{d}$ in a part that lives in the space of the domain modes and an orthogonal part. One can apply this split to the first term eq. (5.62) to write the goal oriented error as:

$$J\left(\tilde{\boldsymbol{\phi}}\right) = \left(\left(\boldsymbol{I} - \boldsymbol{P}\right)\boldsymbol{d}\right)^T\bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} + \left(\boldsymbol{P}\boldsymbol{d}\right)^T\bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} - \omega_j^2\boldsymbol{d}^T\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \tag{5.64}$$

After a number of manipulations, which are detailed in appendix C, this equation can be rewritten to:

$$J\left(\tilde{\boldsymbol{\phi}}\right) = \left(\left(\boldsymbol{I} - \boldsymbol{P}\right)\boldsymbol{d}\right)^T\left(\bar{\boldsymbol{K}} - \tilde{\omega}^2\bar{\boldsymbol{M}}\right)\tilde{\boldsymbol{\phi}} + \left(\tilde{\omega}^2 - \omega_j^2\right)\boldsymbol{d}^T\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \tag{5.65}$$

This expression for the goal oriented error forms the basis for the error estimates for the approximate eigensolution. In the next two subsections, these estimates will be derived for the eigenfrequency and eigenmode, respectively.

5.6.2 Error Estimate for Global Eigenfrequency

To come to an error estimate for an approximate global eigenfrequency $\tilde{\omega}^2$, the first step is to choose the selection vector in the adjoint problem simply as $\boldsymbol{a} = \boldsymbol{0}$. With no goal quantity selected, the goal oriented error in eq. (5.65) reduces to zero:

$$\boldsymbol{0} = \left(\left(\boldsymbol{I} - \boldsymbol{P}\right)\boldsymbol{d}\right)^T\left(\bar{\boldsymbol{K}} - \tilde{\omega}^2\bar{\boldsymbol{M}}\right)\tilde{\boldsymbol{\phi}} + \left(\tilde{\omega}^2 - \omega_j^2\right)\boldsymbol{d}^T\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}}, \tag{5.66}$$

while the adjoint problem in eq. (5.63) now amounts to:

$$\bar{\boldsymbol{K}}\boldsymbol{d} - \omega_j^2\bar{\boldsymbol{M}}\boldsymbol{d} = \boldsymbol{0} \tag{5.67}$$

Hence, a non-trivial adjoint solution can be found as:

$$\boldsymbol{d} = \boldsymbol{P}_j \tilde{\boldsymbol{\phi}} \tag{5.68}$$

This solution is simply verified with the definition of the orthogonal projector $\boldsymbol{P}_j$ in eq. (5.60):

$$\bar{\boldsymbol{K}} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}} - \omega_j^2 \bar{\boldsymbol{M}} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}} = \omega_j^2 \bar{\boldsymbol{M}} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}} - \omega_j^2 \bar{\boldsymbol{M}} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}} = \boldsymbol{0} \tag{5.69}$$

Using this adjoint solution, the last term $\boldsymbol{d}^T \boldsymbol{M} \tilde{\boldsymbol{\phi}}$ on the right hand side of eq. (5.66) can be written as:

$$\boldsymbol{d}^T \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}} \geq 1 - \delta \tag{5.70}$$

Here $0 \leq \delta < 1$ is a so-called safety factor which is often introduced in the literature on error estimation to ensure the conservatism of error bounds [104, 122]. Details of this step are shown in appendix C.1. Substitution of this result in (5.66) then leads to the following:

$$\left(\tilde{\omega}^2 - \omega_j^2\right)(1 - \delta) \leq -\left((\boldsymbol{I} - \boldsymbol{P})\,\boldsymbol{d}\right)^T \left(\bar{\boldsymbol{K}} - \tilde{\omega}^2 \bar{\boldsymbol{M}}\right) \tilde{\boldsymbol{\phi}} \tag{5.71}$$

This expression can be transformed to an error estimate on the approximate eigenfrequency by:

$$e_\omega \leq \frac{1}{1 - \delta} \left| \left((\boldsymbol{I} - \boldsymbol{P})\,\boldsymbol{d}\right)^T \left(\bar{\boldsymbol{K}} - \tilde{\omega}^2 \bar{\boldsymbol{M}}\right) \tilde{\boldsymbol{\phi}} \right| \tag{5.72}$$

Next, using the definition of the residual in eq. (5.61), one can write:

$$e_\omega \leq \frac{1}{1 - \delta} \left| \left((\boldsymbol{I} - \boldsymbol{P})\,\boldsymbol{d}\right)^T \boldsymbol{r} \right| \tag{5.73}$$

As before, the adjoint solution and residual can be written as a summation of the domain contributions. As a result, the eigenfrequency error estimate becomes:

$$e_\omega \leq \frac{1}{1 - \delta} \sum_{s=1}^{n+1} \left| \left(\left(\boldsymbol{I} - \boldsymbol{P}^{(s)}\right) \boldsymbol{d}^{(s)}\right)^T \boldsymbol{r}^{(s)} \right| \tag{5.74}$$

Indeed, apart from the safety factor δ, this expression is exactly similar to eq. (5.30) obtained in the derivation of the error estimates for harmonic problems. Hence, one can proceed the derivation exactly as in section 5.4.2 to obtain the equivalent of eq. (5.45):

$$e_\omega \leq \frac{1}{1 - \delta} \left\| \boldsymbol{d} \right\|_K \left(\sum_{s=1}^{n+1} \frac{1}{\omega_{k_s+1}^{(s)^2}} \left\| \hat{\boldsymbol{r}}^{(s)} \right\| \right) \tag{5.75}$$

The final step in obtaining the error estimate for the eigenfrequency is to find a bound for the energy norm of the adjoint solution $\|\boldsymbol{d}\|_K$. Recalling the adjoint solution found with $\boldsymbol{a} = \boldsymbol{0}$ from eq. (5.68), one can write this energy norm as:

$$\begin{aligned} \|\boldsymbol{d}\|_K &= \left(\tilde{\boldsymbol{\phi}}^T \boldsymbol{P}_j^T \boldsymbol{K} \boldsymbol{P}_j \tilde{\boldsymbol{\phi}}\right)^{1/2} \\ &= \left(\tilde{\boldsymbol{\phi}}^T \bar{\boldsymbol{M}} \boldsymbol{\phi}_j^T \boldsymbol{\phi}_j \bar{\boldsymbol{K}} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}}\right)^{1/2} \\ &= \omega_j \left(\tilde{\boldsymbol{\phi}}^T \boldsymbol{M} \boldsymbol{P}_j \tilde{\boldsymbol{\phi}}\right)^{1/2} \leq \omega_j \end{aligned} \tag{5.76}$$

In the last step it was used that $\tilde{\boldsymbol{\phi}}^T \boldsymbol{M} \boldsymbol{P}_j \tilde{\boldsymbol{\phi}} \leq 1$, see appendix C.1. Finally, the error estimate for the approximate eigenfrequency is thus obtained as:

$$e_\omega \leq \frac{\omega_j}{1-\delta} \left(\sum_{s=1}^{n+1} \frac{1}{\omega_{k_s+1}^{(s)^2}} \left\| \hat{\boldsymbol{r}}^{(s)} \right\| \right)^{1/2} = \frac{\omega_j}{1-\delta} \left(\sum_{s=1}^{n+1} m_K^{(s)} \right)^{1/2} \quad \text{denoted by} \quad e_{\leq\omega} \tag{5.77}$$

As before, the notation $e_{\leq\omega}$ is used to indicate the error estimate and distinguish it from the actual error e_ω. Furthermore, m_K are the same domain indicators as defined previously (eq. (5.52)), and can be used in adaptive refinement schemes such that the error on the eigenfrequency is decreased most efficiently. Note that a relative frequency error can be found by normalizing $e_{\leq\omega}$ by the (approximate) target eigenfrequency.

Quite paradoxically, the error estimate for the approximate global eigenfrequency $\tilde{\omega}$ contains the exact eigenfrequency ω_j for which an error bound is sought. In practice ω_j is obviously not calculated but simply replaced by the approximate frequency obtained from the reduced model. Indeed, due to the reduction this frequency will always exceed the exact frequency of the unreduced model, such that the error estimate remains a conservative bound. When applying the error estimate in an adaptive reduction scheme it could therefore be that at first this frequency is poorly approximated, leading to a (too) high error estimate, but as the refinement progresses the eigenfrequency will converge and the error bound will become more and more accurate.

5.6.3 Error Estimate for Global Eigenmode

In order to arrive at an error estimate for the approximate eigenmode $\tilde{\boldsymbol{\phi}}$, one chooses the selection vector in the goal oriented error similarly as in section 5.4.2, namely $\boldsymbol{a} = \bar{\boldsymbol{K}} \boldsymbol{e}_\phi$. As a result, the goal oriented error again becomes a global energy norm of the error on the eigenmode as defined in eq. (5.60):

$$J\left(\tilde{\boldsymbol{\phi}}\right) = \boldsymbol{a}^T \tilde{\boldsymbol{\phi}} = \boldsymbol{e}_\phi^T \bar{\boldsymbol{K}} \tilde{\boldsymbol{\phi}} = \left\| \boldsymbol{e}_\phi \right\|_K^2 \tag{5.78}$$

Details on how this result is obtained are treated in appendix C.2. With this choice, the goal oriented error from eq. (5.65) becomes the following global error estimate:

$$\left\| \boldsymbol{e}_\phi \right\|_K^2 = \left(\left(\boldsymbol{I} - \boldsymbol{P} \right) \boldsymbol{d} \right)^T \left(\bar{\boldsymbol{K}} - \tilde{\omega}^2 \bar{\boldsymbol{M}} \right) \tilde{\boldsymbol{\phi}} + \left(\tilde{\omega}^2 - \omega_j^2 \right) \boldsymbol{d}^T \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}} \tag{5.79}$$

As in the previous subsection, the last term $\left(\tilde{\omega}^2 - \omega_j^2 \right) \boldsymbol{d}^T \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}}$ in the expression is first analyzed. As is shown in appendix C.2, the following bound for this term is found:

$$\left(\tilde{\omega}^2 - \omega_j^2 \right) \boldsymbol{d}^T \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}} \leq \delta \left\| \boldsymbol{e}_\phi \right\|_K^2 \tag{5.80}$$

In this case the safety factor δ, with $0 \leq \delta < 1$, has the following property (see appendix C.2 for details):

$$\max_{i \neq j} \left(\frac{\tilde{\omega}^2 - \omega_j^2}{\omega_i^2 - \omega_j^2} \right) \leq \delta \tag{5.81}$$

Such that, after substitution in eq. (5.79), the error estimate becomes:

$$\left\| \boldsymbol{e}_\phi \right\|_K^2 \leq \frac{1}{1-\delta} \left(\left(\left(\boldsymbol{I} - \boldsymbol{P} \right) \boldsymbol{d} \right)^T \left(\bar{\boldsymbol{K}} \tilde{\boldsymbol{\phi}} - \tilde{\omega}^2 \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}} \right) \right) \tag{5.82}$$

From this point onwards, the same steps can be followed as in the previous subsection. By doing so one obtains:

$$\left\|\boldsymbol{e}_{\phi}\right\|_{K}^{2} \leq \frac{1}{1-\delta}\left\|\boldsymbol{d}\right\|_{K}\left(\sum_{s=1}^{n+1} \frac{1}{\omega_{k_s+1}^{(s)^2}}\left\|\hat{\boldsymbol{r}}^{(s)}\right\|\right)^{1/2} \tag{5.83}$$

Again, the final step in the derivation of the error estimate is to establish a bound for the energy norm of the adjoint solution. Indeed, for the current adjoint solution this problem is very similar to that of section 5.4.2. Applying the same reasoning as therein thus leads to the following bound:

$$\left\|\boldsymbol{d}\right\|_{K} \leq \left(1+S_{\omega}\right)\left\|\boldsymbol{e}_{\phi}\right\|_{K} \quad \text{with} \quad S_{\omega} = \max_{i \neq j}\left(\frac{\omega_j^2}{\omega_i^2-\omega_j^2}\right) \tag{5.84}$$

Here S_{ω} is again a stability factor that is similar to that for the harmonic problem, except that the frequency of external excitation ω_e is replaced by the frequency of the target eigensolution. With this bound for the adjoint solution, the error estimate $e_{\leq\phi}$ for the approximate eigenmode finally becomes:

$$\left\|\boldsymbol{e}_{\phi}\right\|_{K} \leq \frac{1+S_{\omega}}{1-\delta}\left(\sum_{s=1}^{n+1} \frac{1}{\omega_{k_s+1}^{(s)^2}}\left\|\hat{\boldsymbol{r}}^{(s)}\right\|\right)^{1/2} = \frac{1+S_{\omega}}{1-\delta}\left(\sum_{s=1}^{n+1} m_K^{(s)}\right)^{1/2} \quad \text{denoted by} \quad e_{\leq\phi} \tag{5.85}$$

Similar to what was remarked before regarding the stability factor for the harmonic problem, it is noted here that the above stability factor depends on the (exact) global eigenfrequency that is closest to the target eigenfrequency. Again, these frequencies are not calculated from the full model but are approximated by the reduced model. Therefore, when the stability factor is poorly predicted, the error estimate no longer holds quantitatively but can still be used to estimate the relative domain contributions. The further the domain reduction bases are refined, the better the error estimate becomes.

Intermezzo: Error Estimation for DCB and MTA Methods

The error estimates of the previous sections were derived with respect to the Craig-Bampton reduced model formulation. Although the CB method is often regarded as the most popular component reduction technique, the practical value of the error estimation methods would be further enhanced if it could also be used to obtain error estimates for models reduced with other methods, such as the Rubin, Dual Craig-Bampton and Mixed Craig-Bampton.

To derive the error estimates in this chapter, use was made of the fact that an uncoupled stiffness matrix can be obtained through a transformation with the static constraint modes. The Rubin methods employs static residual attachment modes combined with an additional transformation to obtain displacement interface DoF, leading to coupling between the static and vibration modes (and hence the internal and boundary DoF, see section 2.5.3). It will hence be impossible to obtain such a block-diagonal stiffness matrix for the Rubin method. This is not to say that no error estimates can be derived for this method, but at least not using the current approach.

In the Dual CB method the same modes are used as in the Rubin method, but as explained in section 2.5.4 there is no back-transformation to interface displacement DoF. Therefore, when performing a transformation similar to that of section 5.3, one also obtains a block-diagonal stiffness matrix but with the original interface DoF $\boldsymbol{u}_b$ now replaced by the force DoF $\boldsymbol{g}_b$. However, due to the presence of these force DoF, the DCB equations of motion describe a saddle point problem in contrast to the convex problem described by the CB equations. The derivations in terms of the energy norm therefore no longer hold and a different type of norm should be applied instead. Further research is needed to investigate this.

Another valuable extension of the error estimation methods would be to apply them to reduced models with MTA vectors in the basis. These vectors, as described in section 2.7, are "pseudo-modes" which adhere to all the properties of true eigenmodes except that they do not originate from an eigenproblem. In fact, in this work it was attempted to generalize the error estimates from section 5.4.2 to such augmented reduction bases. However, the presence of the pseudo modes causes problems in the derivation step in eq. (5.39), where both mass- and stiffness orthogonality with respect to the domain's discarded vibration modes is required. This cannot be achieved for the MTAs (otherwise they would be true eigenmodes). Again, further research is needed to see if and how this can be overcome.

5.7 Refinement Schemes for Adaptive Model Reduction

In the previous sections error estimates were presented that give quantitative measures of the reduced model's accuracy for different types of dynamic analysis. All estimates are based on the summation of domain error contributions and can therefore be used in refinement schemes. These refinement schemes are applied in an iterative way to perform adaptive model reduction, that is, the reduced model is adapted through refinement until it has sufficient accuracy for the given analysis. This process is illustrated for the general case in figure 5.3. In the next subsection two adaptivity schemes are discussed that are suitable for respectively goal oriented and global error estimates. Thereafter two basic refinement strategies are proposed in subsection 5.7.2 for use in these adaptivity schemes.

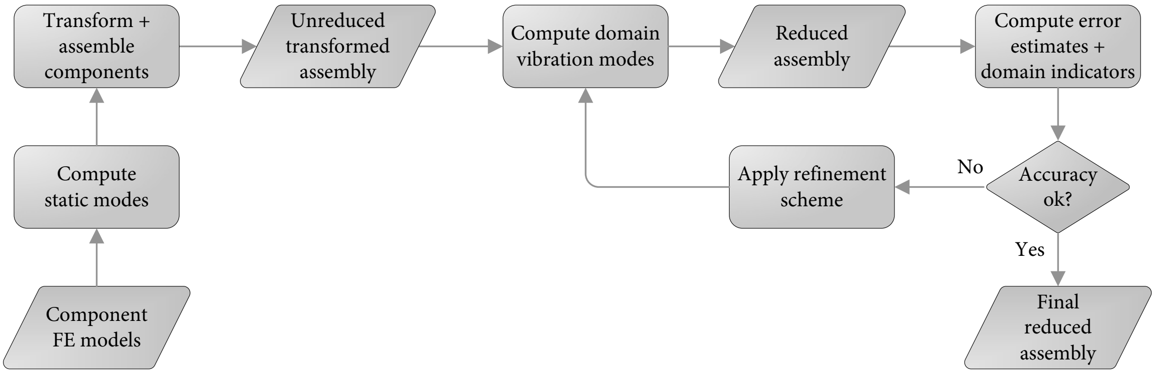

Transform + assemble components
Unreduced transformed assembly
Compute domain vibration modes
Reduced assembly
Compute error estimates + domain indicators
Compute static modes
Apply refinement scheme
No
Accuracy ok?
Yes
Component FE models
Final reduced assembly

Figure 5.3: Schematic representation of adaptive model reduction procedure using error estimation.

5.7.1 Adaptivity Schemes

Based on the error estimates two adaptive model reduction algorithms are defined; one for goal oriented error estimation (algorithm 5.1) and another for global error estimates (algorithm 5.2). Both algorithms consist of a few basic steps. First one defines either an error tolerance ε or a maximum number of DoF of the total reduced model, or both. Then the internal and interface domains are reduced with an initial number of modes k_s. From the resulting force residual the error estimates are evaluated to obtain the domain contributions to the overall error.

In algorithm 5.1 for goal oriented errors the next step is to solve the adjoint problem using a more refined reduced system, where each domain is reduced with k_s^+ modes. Based on the values of the domain error indicators in $\boldsymbol{m}_J$, the domains can be enriched according to some refinement scheme. Two examples of such refinement schemes are proposed in the next subsection. This process is repeated until the error estimate $e_{\approx\Sigma}$ is smaller than a predefined tolerance and/or a maximum size of the assembled reduced model is reached.

Algorithm 5.1 Adaptivity scheme for goal oriented error

```
Define tolerance $\varepsilon$ or maximum size of model $k_{\max}$
Set initial number of modes $k_s$
while $e_{\approx\Sigma} > \varepsilon$ or $\sum_s k_s < k_{\max}$ do
    Reduce domains with number of modes $k_s$
    Solve the reduced system, obtain $\tilde{\boldsymbol{u}}$
    Solve the adjoint problem, using $k_s^+$ modes, obtain $\tilde{\boldsymbol{d}}$
    Compute error indicator $\boldsymbol{m}_J$ from (5.27)
    Apply refinement scheme such that $k_s = k_s + k_{\text{add}}$
end while
```

The second adaptive model reduction algorithm is applicable for global norm-based error estimates. As treated in the previous sections these can be obtained for harmonic analysis, time-stepping problems and global eigensolution convergence. The resulting algorithm 5.2 is very similar to algorithm 5.1, expect that in this case a different error estimate is calculated as well as different domain error indicators. Furthermore, no longer any adjoint solution needs to be solved which is beneficial in terms of computational efficiency.

Algorithm 5.2 Adaptivity scheme for global error estimates

```
Define tolerance $\varepsilon$ or maximum size of model $k_{\max}$
Set initial number of modes $k_s$
while $e_{\leq\star} > \varepsilon$ or $\sum_s k_s < k_{\max}$ do
    Reduce domains with number of modes $k_s$
    Solve the reduced system, obtain $\tilde{\boldsymbol{u}}$ or $\left(\tilde{\lambda}, \tilde{\boldsymbol{\phi}}\right)$
    Compute error indicator $\boldsymbol{m}_K$ from eq. (5.52)
    Apply refinement scheme such that $k_s = k_s + k_{\text{add}}$
end while
```

In the adaptive model reduction schemes the domain reduction bases are iteratively refined with additional eigenmodes. Hence, from a numerical implementation perspective a suitable algorithm should be used for obtaining these eigensolutions. Specifically, the numerical method should be capable of efficiently handling the computation of a limited number of eigenmodes when a number of lower eigenmodes has already been computed. Hence, a good choice would be a restarted iterative algorithm like the implicitly restarted Arnoldi (IRA) method [195, 123] or the implicitly restarted Lanczos method [29].

5.7.2 Refinement Schemes

Based on the knowledge of the domain contribution to the total error, in terms of the error indicators, two refinement strategies are presented here for use in the adaptive model reduction algorithms. Note that here only two basic refinement schemes are proposed and that other, more elaborate heuristics can be imagined.

The first strategy is to obtain the maximum domain contribution and add a fixed number of modes k_{add} to all domains that contribute more than a certain fraction c of this maximum. This scheme will be referred to as "threshold refinement". The parameter c is to be chosen beforehand with a value from zero to one. Using the generic notation $\boldsymbol{m}$ for the domain indicators, which depending on the type of estimate (goal oriented or global) that is used can either be $\boldsymbol{m}_J$ or $\boldsymbol{m}_K$, this refinement scheme can be expressed as in algorithm 5.3.

Algorithm 5.3 Refinement scheme 1 – threshold

```
Set threshold value for refinement $0 \leq c \leq 1$
Select $m_{\max} = \max(\boldsymbol{m})$
for domain $s = 1, \dots, n+1$ do
    if $m^{(s)} \geq c \cdot m_{\max}$ then
        $k_s = k_s + k_{\text{add}}$
    end if
end for
```

In case one chooses $c = 0$ this scheme enriches all domains such that it effectively becomes a uniform refinement scheme, whereas with $c = 1$ only the most inaccurate domain is enriched.

The second refinement strategy computes a relative contribution for each domain based on the total error. This normalized distribution is then used to determine which share of the available enrichment modes k_{add} are to be added to each domain. Of course these values have to be rounded to the nearest integer. This refinement scheme is called "linear refinement" and is shown in algorithm 5.4.

Algorithm 5.4 Refinement scheme 2 – linear

```
Compute $m_{\text{tot}} = \sum_{s=1}^{n+1} m^{(s)}$
for domain $s = 1, \dots, n+1$ do
    Compute $\hat{m}^{(s)} = m^{(s)} / m_{\text{tot}}$
    $k_s = k_s + \hat{m}^{(s)} \cdot k_{\text{add}}$
end for
```

For both of the above refinement schemes it holds that they are easy to compute and implement. The benefit of the "linear" scheme is that the growth of the reduced model per iteration is known in advance and therefore can be better controlled. On the other hand, the "threshold" scheme offers more flexibility by adaptively choosing the value for c in the iterative model reduction algorithm (i.e. algorithm 5.1 or 5.2). For instance one can apply quasi-uniform refinement (i.e. $c \to 0$) when the error is large, whereas targeted refinement steps can be taken when the error gets close to the tolerance ε ($c \to 1$).

5.8 Summary

This chapter was concerned with error estimation for assembled reduced models. This was motivated by the fact that the reduced models only approximate the dynamic behavior of the full system; to have confidence in the outcomes of these models it is important to gain insight into their accuracy. Preferably, this should be achieved without knowledge of the full solution, which can be very computationally expensive, while at the same time providing information on the contribution of each component to the total error.

To this end, a posteriori error estimation methods were considered. It was explained in section 5.2 that these methods were initially developed in the field of finite element analysis to estimate the error arising from discretization of the continuous domain, but have recently been applied for the first time to the similar problem of estimating the reduction error. Based on that work, error estimates were derived in this chapter in an algebraic model reduction setting.

In order to do so, the reduced equations of motion were first cast in an alternative format in section 5.3. This resulted in a transformed system description with an uncoupled, block diagonal stiffness matrix; an essential property for the derivation of the error estimates. Section 5.4 thereafter treated the derivation of error estimates for approximate solutions to harmonic problems obtained from the reduced model. Goal oriented errors were considered first, that is, errors in quantities of interest. Through the definition of an adjoint problem, that is also solved approximately but from a more refined reduced model, error estimates could be derived that express the total error as a summation of component error contributions. The goal oriented error estimate was subsequently generalized to a global error estimate in terms of the energy norm by a specific choice of the selection vector.

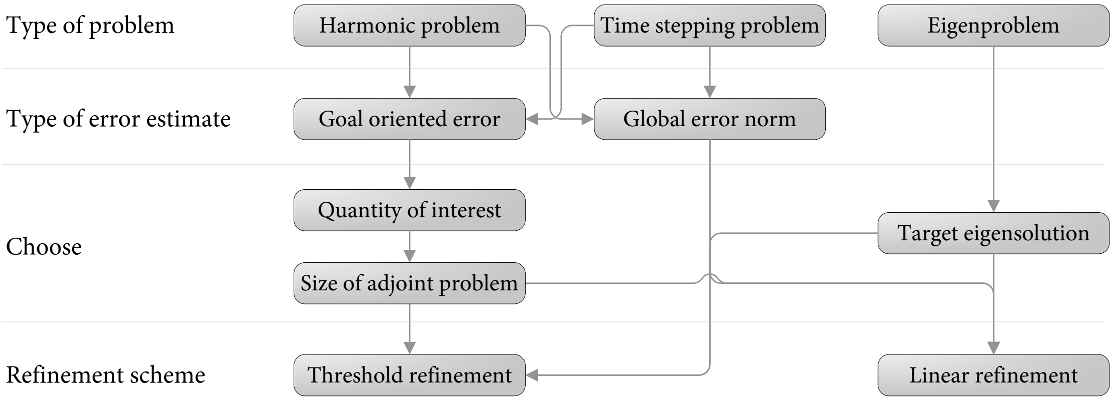


Figure 5.4: Different options and choices for adaptive model reduction using error estimation.

Thereafter, section 5.5 treated error estimates for the time stepping problem that is solved within time integration schemes. As this problem is very similar to the harmonic problem, so are its error estimates. Finally, in section 5.6 the problem of estimating the accuracy of approximate global eigensolutions was addressed. Using the same reasoning as for the harmonic solution, error estimates for an approximate eigenfrequency and -mode could be derived.

In section 5.7 it was outlined how the error estimates may be used in an adaptive model reduction scheme. These schemes aim to reduce the reduction error as efficiently as possible by refining those components that contribute most to the total error. Two simple refinement schemes were proposed to do so.

Summarizing once more, error estimates have been derived for three types of problems. For harmonic and time stepping problems, one can further choose a goal oriented error or global error norm. In the former case, one additionally needs to choose a quantity of interest and the size of the approximate adjoint problem. Finally, one can choose one of the two proposed refinement schemes. These different options and choices are indicated in the flowchart in figure 5.4.

6

Updating of Component Reduction Bases

> *Fast cars, fast women, fast algorithms…*
> *what more could a man want?*
>
> (Joe Mattis)

6.1 Introduction

In a dynamic substructuring analysis, the main computational cost lies in the computation of the component reduction bases. As explained in chapter 2, this reduction basis in general consists of a set of static modes and a truncated set of vibration modes. Hence, obtaining the basis involves solving both a linear system of equations for the static modes and an eigenproblem to obtain the vibration modes. This initial investment pays off when the assembled system is analyzed; since it is built up from reduced component models it is very compact and can be analyzed efficiently. However, in the practical setting of an iterative design process, one or more of the component models may be subject to (parametric) modifications. This means that for every design change these components need to be remodeled and their reduction basis recomputed. Thereby the efficiency of the componentwise reduced approach is largely undone.

However, the design changes made to the components are in practice often rather small. It is therefore unlikely that the modified component will exhibit completely different behavior in terms of eigenfrequencies and mode shapes. The question thus arises whether one actually needs to recalculate the complete component reduction basis or if maybe the knowledge of the nominal system could be used to efficiently update its basis to suit the modified component. In this chapter a method is developed to achieve this, the interaction with the topics presented in other chapters is shown in figure 6.1.

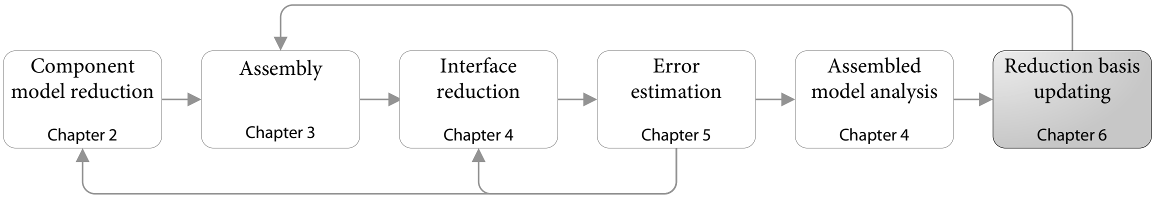


Figure 6.1: Current chapter in relation to other chapters in part I of this thesis.

6.1.1 Existing Methods

In the literature, different approaches have been described to efficiently update component reduction bases. For the static modes, one approach is to simply assume that the nominal static modes can also be used to reduce the modified component [133]. In another publication it is proposed to completely recompute them based on the new model [30]. Finally, the static modes can be updated using the Combined Approximations (CA) technique [114], as proposed in [1].

The vibration modes can also be updated using the Combined Approximations method, as was suggested in [30]. Another approach is to enrich the reduction basis using sensitivity modes as described in [133, 1]. Such sensitivity modes represent the change of the vibration mode shapes around the nominal design point with respect to the design change. Furthermore, methods exist where an (approximate) extended reduction basis is created for a parametric family of models [14, 13]. This requires a priori knowledge of the design modifications.

The main difficulties with the existing methods is that either the approximate reduction basis fails to accurately capture the response of the modified system or, in the case of multiple design parameters, the extended basis grows larger and larger, eventually losing its effectiveness. In this work a different approach is therefore taken. Here the aim is not to approximate or enrich the reduction basis but truly update the static and vibration modes to suit the modified component model, thereby reusing as much information as possible from the nominal system. To this end, *iterative solution techniques* will be used in combination with effective *preconditioning*. These concepts will be elaborated further in the following sections, after explaining the starting points for this work:

- In line with the work referred to earlier [1, 133, 30], it is assumed that the component FE mesh topology and connectivity are not altered. Nodal positions may be changed (e.g. elongation or thickness changes) as long as this does not result in element shape violations.

- The goal of the methodology presented here is to update the reduction basis. This means the reduced models remain of the same size, which implicitly assumes that the same number of modes can properly represent the structural response after modification.

- The stiffness matrix of the nominal system is factorized when setting up the nominal reduced model. This factorization is a significant part of the computational cost of the reduction, it is therefore assumed that this factorization is stored for further use.

- The methods developed in this chapter should be universally applicable to all types of component reduction bases. To this end, the generic notation from section 2.2 will be adopted; details on how specific types of modes are calculated were addressed in sections 2.3 and 2.4 for respectively the static and vibration modes.
- Parametrization of a component FE model is out of the scope of this chapter. Instead, it is simply assumed that some design change is performed which is translated into a modified FE model. In chapter 8 an example is presented where a component FE mesh is parameterized for simple design modifications and the structural matrices updated accordingly.

6.1.2 Direct Versus Iterative Solvers

As explained in chapter 2, the computation of a component reduction basis requires solving both a static and an eigenvalue problem. The former is associated with the computation of the p static modes $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1 \; \dots \; \boldsymbol{\psi}_p]$ and generally is of the form (see section 2.3)

$$\boldsymbol{K}\boldsymbol{\psi}_j = \boldsymbol{f}_j, \tag{6.1}$$

while the latter is solved to find the m vibration modes $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \; \dots \; \boldsymbol{\phi}_m]$ and takes the form (see section 2.4)

$$\boldsymbol{K}\boldsymbol{\phi}_j = \omega_j^2 \boldsymbol{M}\boldsymbol{\phi}_j \tag{6.2}$$

Here $\boldsymbol{f}$ is some right hand side equivalent to an applied force. The vector $\boldsymbol{\psi}_j$ is some required static mode and $\boldsymbol{\phi}_j$ the vibration mode with ω_j its associated eigenfrequency. Note that in case one is interested in solutions around a certain frequency of interest ω_s, one can also use modes of the shifted problem. In that case $\boldsymbol{K}$ should be replaced by $(\boldsymbol{K} - \omega_s^2 \boldsymbol{M})$; the methods discussed in this chapter can be used without modification in case shifting is applied.

The static problem is usually solved using direct techniques. These are basically variants of the factorization techniques by Gauss, such as LDLT decomposition for symmetric matrices or Cholesky decomposition for symmetric positive matrices, which are very robust and efficient. Direct solvers are therefore the standard choice for most static calculations. In this case, the factorization of the stiffness matrix can be stored for future computations.

On the other hand, eigenvalue problems in dynamics are generally solved using iterative techniques such as the Lanczos method [121]. Due to the sparsity of the matrices and the fact that one is usually only interested in a few of the structure's (lowest) eigenfrequencies, iterative methods are clearly the most efficient choice. The Lanczos method, and related methods such as the Block Lanczos, Arnoldi and Implicitly Restarted Arnoldi [195], require the factorized stiffness matrix in their iterations. When the static problem is solved using a direct solver, this factorization is readily available.

The most time consuming steps in computing the reduction basis are the factorization of the stiffness matrix and the eigensolver iterations. Hence, when a component is modified one would ideally like to avoid these computationally intensive steps. Indeed, it was noted earlier that when the modified system is "close" to the nominal one, its dynamic behavior will probably not change dramatically. Therefore, it is proposed not to start the factorization and eigensolution for the modified component from scratch but instead use iterative methods. The aim is to start from the solutions of the nominal system, reuse as much of the available information as possible and quickly converge to the solutions for the modified system.

6.1.3 Chapter Outline

The updating strategy proposed in this work builds on preconditioned iterative methods. Similar to the calculation of the nominal reduction basis, the updating strategy requires the static and eigenproblem to be solved sequentially. The procedure for updating the static modes will be based upon the *preconditioned conjugate gradient* method and is elaborated in section 6.2. Thereafter, the updating scheme for the vibration modes is explained in section 6.3 and achieved using the *inverse-free preconditioned Krylov subspace* method. Section 6.4 then briefly covers the issue of sequential updating, that is, when a component is subjected to a sequence of design modifications. A summary is provided in 6.5.

Regarding the notations in this chapter: a subscript $\star_{(0)}$ refers to the nominal component, while $\star_{(1)}$ designates the modified component. The component denotation $\star^{(s)}$ used in the previous chapters will be omitted here for compactness; all expressions regard component properties unless otherwise noted. Furthermore, the symbol $\Delta\star$ is used to indicate the difference between the nominal and modified system. Subscripts $_i$ and $_k$ refer to iteration indices.

6.2 Updating of Static Modes

In this section an approach will be presented for updating the static modes in the component reduction basis. The starting point for this updating problem is as follows. For the nominal component, the static problem has been solved:

$$\boldsymbol{\Psi}_{(0)} = \boldsymbol{K}_{(0)}^{-1}\boldsymbol{F}_{(0)} \tag{6.3}$$

Subsequently a design modification is made to the component, which leads to the new static problem:

$$\boldsymbol{K}_{(1)}\boldsymbol{\Psi}_{(1)} = \boldsymbol{F}_{(1)} \tag{6.4}$$

The goal is now to efficiently find $\boldsymbol{\Psi}_{(1)}$ by reusing $\boldsymbol{\Psi}_{(0)}$ and $\boldsymbol{K}_{(0)}^{-1}$, while avoiding the factorization of $\boldsymbol{K}_{(1)}$. This section is therefore organized as follows. First the general concept of the CG method is outlined, thereafter subsection 6.2.2 addresses its convergence as well as how to improve it using preconditioning. The issue of solving for multiple right hand sides is addressed in subsection 6.2.3 while subsection 6.2.4 finally returns to the updating problem and shows why the approach might be useful.

6.2.1 The Conjugate Gradient Method

In addition to direct solvers, iterative solvers can also be employed to solve static problems. Most iterative methods for static problems are based on the *conjugate gradient* (CG) algorithm proposed by Hestenes and Stiefel [90] in 1952. These methods initially did not gain popularity as their convergence was slow and they were lacking robustness compared to direct techniques. However, due to their sequential nature, the direct solvers turned out to be ill-suited for parallel computing. The upcoming of parallel computing in the 1980's therefore boosted the development of iterative solvers and as a result, they became a serious alternative to their direct counterparts. Since the CG method will be heavily used in this chapter, it is summarized hereafter.

The conjugate gradient (CG) method is an iterative algorithm for solving systems of linear equations with symmetric and positive-definite operators, such as the problem for a static mode:

$$\boldsymbol{K}\boldsymbol{\psi} = \boldsymbol{f} \tag{6.5}$$

Solving this linear system is equal to minimization of the quadratic problem:

$$\mathcal{L}(\boldsymbol{\psi}) = \frac{1}{2}\boldsymbol{\psi}^T\boldsymbol{K}\boldsymbol{\psi} - \boldsymbol{\psi}^T\boldsymbol{f} \tag{6.6}$$

Suppose that one has some initial guess $\boldsymbol{\psi}_0$ of the solution ($\boldsymbol{\psi}_0$ can be equal to $\boldsymbol{0}$) and wishes to iteratively refine it by adding some improvement: $\boldsymbol{\psi}_1 = \boldsymbol{\psi}_0 + \Delta\boldsymbol{\psi}$. The question now is how to choose $\Delta\boldsymbol{\psi}$. Taking the negative gradient of the quadratic problem around the initial guess, one finds:

$$-\left.\frac{\partial\mathcal{L}}{\partial\boldsymbol{\psi}}\right|_{\boldsymbol{\psi}_0} = \boldsymbol{f} - \boldsymbol{K}\boldsymbol{\psi}_0 = \boldsymbol{r}_0 \tag{6.7}$$

In fact, $\boldsymbol{r}_0$ is the residual force vector for the initial guess $\boldsymbol{u}_0$ and corresponds to the steepest descent direction around $\boldsymbol{\psi}_0$, that is, the direction in which $\mathcal{L}$ is reduced quickest. Hence the new approximation $\boldsymbol{\psi}_1$ is written as

$$\boldsymbol{\psi}_1 = \boldsymbol{\psi}_0 + \eta_0\boldsymbol{r}_0, \tag{6.8}$$

where η_0 is the unknown amplitude in the direction of the residual $\boldsymbol{r}_0$. Inserting this in the linear problem yields

$$\boldsymbol{K}\boldsymbol{\psi}_1 = \boldsymbol{K}(\boldsymbol{\psi}_0 + \eta_0\boldsymbol{r}_0) = \boldsymbol{f} - \boldsymbol{r}_1 \tag{6.9}$$

To find the "optimal" value for η, the new residual is required to be zero in the space of the current approximation, i.e. $\boldsymbol{r}_0^T\boldsymbol{r}_1 = 0$. This gives

$$\boldsymbol{r}_0^T\boldsymbol{K}(\boldsymbol{\psi}_0 + \eta_0\boldsymbol{r}_0) = \boldsymbol{r}_0^T\boldsymbol{f}, \tag{6.10}$$

and using the expression for $\boldsymbol{r}_0$ then allows to find η_0:

$$\eta_0 = \frac{\boldsymbol{r}_0^T\boldsymbol{r}_0}{\boldsymbol{r}_0^T\boldsymbol{K}\boldsymbol{r}_0} \tag{6.11}$$

Note that this corresponds to a "line search" minimization step. Subsequently one can calculate $\boldsymbol{\psi}_1$ and the associated residual $\boldsymbol{r}_1$, compute the next correction for η_1, and so on. In this sequence, the current residual is each time minimized in the space of only the previous residual. This means that the same direction might appear several times in the residual and as a result, the solution is not guaranteed to be found in a finite number of steps.

To overcome this, Hestenes and Stiefel proposed to use *conjugate* gradients, by adding an orthogonalization step to the process. In this way, the search direction at iteration k is no longer directly the residual $\boldsymbol{r}_k$, but is first $\boldsymbol{K}$-orthogonalized to all the previous search directions. In theory, the new search direction needs to be orthogonalized only with respect to the previous one, but due to numerical round-off errors the preceding directions will progressively reappear. Hence a full orthogonalization is needed, as follows:

$$\boldsymbol{p}_k = \boldsymbol{r}_k - \sum_{i=0}^{k-1}\boldsymbol{p}_i\beta_i \tag{6.12}$$

where k is the index of the current iteration and i the index of the previous directions. Here $\boldsymbol{p}$ is used to denote the search direction (i.e. orthogonalized residual) as opposed to the true residual $\boldsymbol{r}$. Realizing that after orthogonalization it should hold that $\boldsymbol{p}_i^T\boldsymbol{K}\boldsymbol{p}_k = 0$, and after scaling the search directions such that $\boldsymbol{p}_i^T\boldsymbol{K}\boldsymbol{p}_i = 1$, the factor β_i is found as:

$$\beta_i = \boldsymbol{r}_k^T\boldsymbol{K}\boldsymbol{p}_i \tag{6.13}$$

Since the search directions obtained in this way span a $\boldsymbol{K}$-orthogonal space, the algorithm is guaranteed to converge in n iterations, where n is the size of the problem. The basic ingredients of the CG method have now been treated and can be summarized in the form of algorithm 6.1.

Algorithm 6.1 Basic conjugate gradient method

```
Input: $\boldsymbol{K}, \boldsymbol{f}, \boldsymbol{\psi}_0$
$\boldsymbol{r}_0 = \boldsymbol{f} - \boldsymbol{K}\boldsymbol{\psi}_0,\ k = 0$
while $||\boldsymbol{r}_k|| > \varepsilon||\boldsymbol{f}||$ do
    $\boldsymbol{p}_k = \boldsymbol{r}_k$
    – Orthogonalization with previous directions –
    for $i = 0 \dots k-1$ do
        $\beta_i = \boldsymbol{r}_k^T\boldsymbol{K}\boldsymbol{p}_i$
        $\boldsymbol{p}_k = \boldsymbol{p}_k - \boldsymbol{p}_i\beta_i$
    end for
    $\boldsymbol{p}_k = \frac{\boldsymbol{p}_k}{\sqrt{\boldsymbol{p}_k^T\boldsymbol{K}\boldsymbol{p}_k}}$
    – Minimization and updating –
    $\eta_k = \boldsymbol{p}_k^T\boldsymbol{r}_k$
    $\boldsymbol{\psi}_{k+1} = \boldsymbol{\psi}_k + \eta_k\boldsymbol{p}_k$
    $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \eta_k\boldsymbol{K}\boldsymbol{p}_k = \boldsymbol{f} - \boldsymbol{K}\boldsymbol{\psi}_{k+1}$
    $k = k + 1$
end while
```

Some remarks are in place regarding the algorithm 6.1 and the CG method in general:

- The main cost of the algorithm is in the matrix-vector multiplication $\boldsymbol{K}\boldsymbol{p}_k$. Hence one can optimize the algorithm by storing this product as $\boldsymbol{w}_k$, so that it needs to be calculated only once per iteration.

- The calculation of the new residual can be done recursively or directly. For numerical stability it is advisable to explicitly evaluate the residual once every few iterations (for instance $\sqrt{n}$); see [193] for a discussion.

- The convergence criterion is to compare the norm of the residual with the norm of the force (right hand side) and define an iteration tolerance ε for their ratio.

- The normal CG method makes use of the symmetry of the operator $\boldsymbol{K}$ in the conjugation step. However, for non-symmetric operators similar algorithms can be devised such as the bi-CG or GMRES methods [179, 178].

- The sequence of search directions spanning the solution space is in fact the Krylov space $\mathcal{K}_k$ spanned by $\boldsymbol{K}$ starting from $\boldsymbol{r}_0$ [77]:

$$\text{span}\left\{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_k\right\} = \text{span}\left\{\boldsymbol{r}_0, \boldsymbol{K}\boldsymbol{r}_0, ..., \boldsymbol{K}^{k-1}\boldsymbol{r}_0\right\} = \mathcal{K}_k\left(\boldsymbol{K}, \boldsymbol{r}_0\right) \tag{6.14}$$

6.2.2 Convergence & Preconditioning

In the original version of the CG method outlined above, the convergence of the algorithm is usually very slow. To illustrate this, a lower bound for the convergence rate of the conjugate gradient method can be expressed as a function of the condition number κ of the operator $\boldsymbol{K}$ [77]:

$$\left\|\boldsymbol{\psi} - \boldsymbol{\psi}_k\right\|_K \leq 2 \left\|\boldsymbol{\psi} - \boldsymbol{\psi}_0\right\|_K \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^k \tag{6.15}$$

Here $\|\star\|_K$ is the energy norm defined as $\sqrt{\star^T \boldsymbol{K} \star}$, see section 5.4. This indicates that when $\kappa(\boldsymbol{K}) \approx 1$ the convergence of the CG algorithm is very fast. Most structural problems however, suffer from bad mathematical conditioning for instance arising from stiffness changes in the structure, mixing of translation and rotational DoF, etc. In those cases $\kappa(\boldsymbol{K}) \gg 1$ and the convergence rate is very slow. An important way to increase the convergence rate is thus to improve the condition number of the operator. This can be done by the transformation

$$\hat{\boldsymbol{K}} = \boldsymbol{S}^{-1} \boldsymbol{K}, \tag{6.16}$$

with $\boldsymbol{S}$ chosen such that the condition number of $\hat{\boldsymbol{K}}$ is lower than that of $\boldsymbol{K}$. This concept is known as *preconditioning* and was the driving force behind the rapid development of the CG type methods some three decades ago. Note that $\boldsymbol{S}$ must be a full rank matrix in order to retain the full possible solution space. In the iterative scheme of the CG method the preconditioning step can be implemented by taking

$$\boldsymbol{p}_k = \boldsymbol{S}^{-1} \boldsymbol{r}_k, \tag{6.17}$$

such that the Krylov subspace generated by the CG algorithm becomes:

$$\mathcal{K}_k\left(\hat{\boldsymbol{K}}, \boldsymbol{r}_0\right) = \operatorname{span}\left\{\boldsymbol{r}_0, \hat{\boldsymbol{K}} \boldsymbol{r}_0, \ldots, \hat{\boldsymbol{K}}^{k-1} \boldsymbol{r}_0\right\} \tag{6.18}$$

Ideally one would take $\mathbf{S}^{-1} = \boldsymbol{K}^{-1}$, so that $\kappa(\hat{\boldsymbol{K}}) = 1$. In that case the search direction becomes

$$\boldsymbol{p}_k = \boldsymbol{K}^{-1} \boldsymbol{r}_k, \tag{6.19}$$

and the exact solution would be found in one iteration. This is of course the paradox of the method: to achieve this the inverse of $\boldsymbol{K}$ is needed from a factorization. The updating step is then nothing more than a forward/backward substitution and the CG method has in fact become a direct solver. The trick is therefore to come up with some approximation $\hat{\boldsymbol{K}}^{-1}$ for $\boldsymbol{K}^{-1}$ which is good enough to seriously speed up the calculations but is not too costly in its construction.

A physical interpretation of the slow convergence without preconditioning is that the displacement solution one seeks is iteratively updated using forces (the orthogonalized residual vectors). This seems inconsistent from a mechanical point of view and one should in fact translate the force errors to a displacement correction, using preconditioning. The better the preconditioner approximates the inverse of $\boldsymbol{K}$, the closer the correction is to a true displacement correction.[1]

---

[1] Another interpretation of the slow convergence is that the structural response is dominated by the lowest modes of $\boldsymbol{K}$. Since one applies direct iteration on $\boldsymbol{K}$, as opposed to inverse iteration, one first has to go through the high spectrum of $\boldsymbol{K}$ in order to arrive at the lowest modes.

The simplest and computationally cheapest preconditioning is the Jacobi preconditioner

$$\boldsymbol{S} = \operatorname{diag}(\boldsymbol{K}), \tag{6.20}$$

which corresponds to simple scaling of $\boldsymbol{K}$. The issue of choosing an optimal preconditioner has been studied intensively over the years and as a result, many other preconditioners exist. This is however not of interest here, as will be shown in subsection 6.2.4.

6.2.3 Multiple Right Hand Sides

In the above discussion the CG method was considered for solving a static problem with a single right hand side:

$$\boldsymbol{K}\boldsymbol{\psi}_j = \boldsymbol{f}_j \tag{6.21}$$

In practice however, a component has very rarely only a single interface degree of freedom and hence usually multiple static modes need to be calculated, i.e.

$$\boldsymbol{K}\boldsymbol{\Psi} = \boldsymbol{F}, \tag{6.22}$$

where $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1 \quad \dots \quad \boldsymbol{\psi}_p]$ and $\boldsymbol{F} = [\boldsymbol{f}_1 \quad \dots \quad \boldsymbol{f}_p]$. Using a direct solver this is very easy; since the factorization is already available only a forward/backward substitution is needed to solve for the new right hand sides.

The question thus arises if something similar can be done for the CG method or whether one needs to start from scratch. In fact, two ways exist to handle multiple right hand sides in the CG method, namely 1) performing subsequent iterations making use of the previous results through projection and reconjugation, and 2) simultaneous iteration on a block of vectors. Both will be treated next.

6.2.3.1 Projection and Reconjugation

Suppose that the first static mode has been computed with a (preconditioned) CG solver:

$$\boldsymbol{K}\boldsymbol{\psi}_1 = \boldsymbol{f}_1 \tag{6.23}$$

In the process the following sequences were generated:

$$\boldsymbol{P}_1 = [\boldsymbol{p}_1, \boldsymbol{p}_2, \dots], \quad \boldsymbol{W}_1 = [\boldsymbol{K}\boldsymbol{p}_1, \boldsymbol{K}\boldsymbol{p}_2, \dots], \tag{6.24}$$

such that by construction one has a $\boldsymbol{K}$-orthonormal basis:

$$\boldsymbol{P}_1^T \boldsymbol{K} \boldsymbol{P}_1 = \boldsymbol{P}_1^T \boldsymbol{W}_1 = \boldsymbol{I} \tag{6.25}$$

Hence, if a new right hand side is considered in order to find the next static mode, one can first search for (an estimate of) the solution in the existing space. So suppose it is tried to solve

$$\boldsymbol{K}\boldsymbol{\psi}_2 = \boldsymbol{f}_2 \tag{6.26}$$

and an initial guess $\boldsymbol{\psi}_{2,0}$ for the solution already exists (the following also holds if $\boldsymbol{\psi}_{2,0} = \boldsymbol{0}$). Instead of directly starting the CG iterations, the solution can first be improved by searching for a correction in the existing space:

$$\boldsymbol{\psi}_2 = \boldsymbol{\psi}_{2,0} + \boldsymbol{P}_1\boldsymbol{\alpha}_1 \tag{6.27}$$

Inserting this in the static mode problem, premultiplying with $\boldsymbol{P}_1^T$ and solving for $\boldsymbol{\alpha}_1$ gives:

$$\boldsymbol{\alpha}_1 = \boldsymbol{P}_1^T\left(\boldsymbol{f}_2 - \boldsymbol{K}\boldsymbol{\psi}_{2,0}\right) \tag{6.28}$$

The improved solution for the second static mode is thus found as

$$\boldsymbol{\psi}_2 = \boldsymbol{\psi}_{2,0} + \boldsymbol{P}_1\boldsymbol{P}_1^T\left(\boldsymbol{f}_2 - \boldsymbol{K}\boldsymbol{\psi}_{2,0}\right) = \boldsymbol{P}_1\boldsymbol{P}_1^T\boldsymbol{f}_2 + \left(\boldsymbol{I} - \boldsymbol{P}_1\boldsymbol{P}_1^T\boldsymbol{K}\right)\boldsymbol{\psi}_{2,0}, \tag{6.29}$$

where the last term can be recognized as projection of the $\boldsymbol{\psi}_{2,0}$ on the space $\boldsymbol{K}$-orthogonal to the existing space $\boldsymbol{P}_1$. By doing so the solution is first computed in the space $\boldsymbol{P}_1$ and subsequently improved by CG iterations. To that end, the associated residual can be calculated as:

$$\begin{aligned}\boldsymbol{r} &= \boldsymbol{f}_2 - \boldsymbol{K}\boldsymbol{\psi}_2\\ &= \left(\boldsymbol{I} - \boldsymbol{W}_1\boldsymbol{P}_1^T\right)\boldsymbol{f}_2 - \left(\boldsymbol{K} - \boldsymbol{W}_1\boldsymbol{W}_1^T\right)\boldsymbol{\psi}_{2,0}\end{aligned} \tag{6.30}$$

This forms the starting point for the CG iterations for the new static mode. To avoid searching in the existing subspace, the new search directions should not only be mutually orthogonalized but should also be orthogonalized with respect to $\boldsymbol{P}_1$ (this is sometimes called *re-conjugation*). This process can be repeated for all right hand sides and the search space $\boldsymbol{P}$ is continuously enriched. When this space is rich enough, the solution for a new right hand side may be found with very few iterations. For more details see for instance [177, 58].

6.2.3.2 Block Conjugate Gradient

Another way to treat multiple right hand sides in a CG solver is by solving them all simultaneously (i.e. solving directly eq. (6.22)). This can be achieved by iterating on a block of vectors [145].[2] In order to adapt the CG algorithm to block computations, the only notable difference with the single vector algorithm is instead of simple scaling one needs to perform a $\boldsymbol{K}$-orthonormalization of the vectors in the block of search directions $\boldsymbol{P}$, such that $\boldsymbol{P}^T\boldsymbol{K}\boldsymbol{P} = \boldsymbol{P}^T\boldsymbol{W} = \boldsymbol{I}$. This can be accomplished for example through a modified Gram-Schmidt process [77]. Using this orthonormality, the orthogonalization step of the block of vectors with respect to the previous directions is straightforward:

$$\boldsymbol{P}_k = \boldsymbol{R}_k - \sum_{i=1}^{k-1}\boldsymbol{P}_i\boldsymbol{\beta}_i \quad \text{with} \quad \boldsymbol{\beta}_i = \boldsymbol{W}_i^T\boldsymbol{R}_k. \tag{6.31}$$

Similarly, the minimization step in the block algorithm is also easily solved as:

$$\boldsymbol{\Psi}_{k+1} = \boldsymbol{\Psi}_k + \boldsymbol{P}_k\boldsymbol{\eta}_k \quad \text{with} \quad \boldsymbol{\eta}_k = \boldsymbol{P}_k^T\boldsymbol{R}_k. \tag{6.32}$$

---

[2] Obviously, the block CG method can only be applied if all right hand sides are known in advance, i.e. if a right hand side is not computed based on the outcome of previous right hand side(s). This is usually the case for component static modes.

Since in the block implementation the residuals are minimized simultaneously in all search directions of the set of residuals, the convergence of the method is very fast in terms of number of iterations. However, the total number of search directions needed for convergence (i.e. block size times number of iterations) is approximately equal for the block and the single vector algorithms.[3] Still the block CG method is an interesting option since modern CPUs can run more efficiently when iterations are performed on a number of vectors instead of a single vector.

However, when a large number of static modes need to be computed (i.e. when the component has many interface DoF), the efficiency of the block algorithm breaks down due to its memory requirements and the need to orthonormalize $\boldsymbol{P}_k$. To overcome this, the projection and reconjugation approach for multiple right hand sides can be combined with the block CG algorithm. This allows one to choose the block size such that the processor is used most efficiently while at the same time making use of the previous iterates for finding the solutions for the other block right hand sides. So, the static problem of eq. (6.22) is split in a number of block equations

$$\boldsymbol{K}\boldsymbol{\Psi}_j = \boldsymbol{F}_j, \tag{6.33}$$

and subsequently solved using a combination of the projection/reconjugation and block approaches. This variation of the CG method is outlined in algorithm 6.2.

6.2.4 Application to Updating of Static Modes

6.2.4.1 Preconditioning

In the previous subsections the basics of the CG method were outlined and it became clear that the convergence rate algorithm is highly dependent on the effectiveness of the preconditioner. Next it will be shown why the CG algorithm is potentially very attractive for the updating of static modes in a component reduction basis.

Suppose that the static modes $\boldsymbol{\Psi}_{(0)}$ of the nominal component have been calculated and the factorization of $\boldsymbol{K}_{(0)}$ is stored. Next the static modes problem for the modified component is to be solved

$$\boldsymbol{K}_{(1)}\boldsymbol{\Psi}_{(1)} = \boldsymbol{F}_{(1)}, \tag{6.34}$$

using the CG method from algorithm 6.2. As initial guess for the solution the static modes of the nominal system are used and, more importantly, the factorization of the nominal stiffness matrix is used for preconditioning. Now recall that the convergence rate of the CG method is given by the bound in (6.15) and is dependent on the conditioning of the preconditioned stiffness matrix. After preconditioning one thus has the following operator in the CG algorithm:

$$\hat{\boldsymbol{K}} = \boldsymbol{K}_{(0)}^{-1}\boldsymbol{K}_{(1)} = \boldsymbol{K}_{(0)}^{-1}\left(\boldsymbol{K}_{(0)} + \Delta\boldsymbol{K}\right) = \boldsymbol{I} + \boldsymbol{K}_{(0)}^{-1}\Delta\boldsymbol{K} \tag{6.35}$$

Clearly, the closer $\boldsymbol{K}_{(0)}^{-1}\Delta\boldsymbol{K}$ is to some factor times identity, the closer the condition number of the preconditioned stiffness matrix will be to unity. Indeed, in case $\Delta\boldsymbol{K} = \alpha\boldsymbol{K}_{(0)}$, then the

---

[3] The total number of vectors needed in the block algorithm is usually slightly higher than for the single vector algorithm. This is due to the fact that all vectors in a block come from the same Krylov order.

Algorithm 6.2 Preconditioned block conjugate gradient with multiple right hand sides

```
Input: $\boldsymbol{K}, \boldsymbol{F}, \boldsymbol{\Psi}_0, \boldsymbol{S}^{-1}$
$k = 0, j = 1$
$\boldsymbol{R}_0 = \boldsymbol{F}_1 - \boldsymbol{K}\boldsymbol{\Psi}_{1,0}$
for $j = 1 \dots p$ do
    – Initial guess for new right hand sides –
    if $j > 1$ then
        $\boldsymbol{Q} = \boldsymbol{K}\boldsymbol{\Psi}_{j,0}$
        $\boldsymbol{\alpha} = \boldsymbol{P}^T \left(\boldsymbol{F}_j - \boldsymbol{Q}\right)$
        $\boldsymbol{R}_k = \boldsymbol{F}_j - \boldsymbol{W}\boldsymbol{\alpha} - \boldsymbol{Q}$
        $\boldsymbol{\Psi}_{j,0} = \boldsymbol{\Psi}_{j,0} + \boldsymbol{P}\boldsymbol{\alpha}$
    end if
    while $||\boldsymbol{r}_{m,k}|| > \varepsilon||\boldsymbol{f}_{m,j}||$ do
        $\boldsymbol{P}_k = \boldsymbol{S}^{-1}\boldsymbol{R}_k$
        – Orthogonalization with previous directions –
        for $i = 1 \dots k-1$ do
            $\boldsymbol{\beta}_i = \boldsymbol{W}_i^T\boldsymbol{P}_k$
            $\boldsymbol{P}_k = \boldsymbol{P}_k - \boldsymbol{P}_i\boldsymbol{\beta}_i$
        end for
        $\boldsymbol{W}_k = \boldsymbol{K}\boldsymbol{P}_k$
        Orthonormalize $\boldsymbol{P}_k : \boldsymbol{P}_k^T\boldsymbol{W}_k = \boldsymbol{I}$
        – Minimization and updating –
        $\boldsymbol{\eta}_k = \boldsymbol{P}_k^T\boldsymbol{R}_k$
        $\boldsymbol{\Psi}_{j,k+1} = \boldsymbol{\Psi}_{j,k} + \boldsymbol{P}_k\boldsymbol{\eta}_k$
        $\boldsymbol{R}_{k+1} = \boldsymbol{F}_j - \boldsymbol{K}\boldsymbol{\Psi}_{j,k+1} = \boldsymbol{R}_k - \boldsymbol{W}_k\boldsymbol{\eta}_k$
        $k = k + 1$
    end while
end for
```

condition number would be one and the solution is obtained in a single iteration. Hence, a very good preconditioner is available "for free" if $\boldsymbol{K}_{(0)}$ was factorized for the nominal system. The closer the modified component is to the nominal one (i.e. the smaller the design change), the better this preconditioner becomes and the faster the CG algorithm will converge.

Note that when dealing with a floating subsystem the stiffness matrix $\boldsymbol{K}_{(0)}$ is singular and its inverse will be a generalized inverse $\boldsymbol{K}_{(0)}^+$.[4] This generalized inverse can for instance be obtained through projecting $\boldsymbol{K}_{(0)}$ out of the space spanned by the rigid body modes. However, the rigid body modes may change with a design modification, or in other words the nullspace of $\boldsymbol{K}_{(0)}$ is not equal to the nullspace of $\boldsymbol{K}_{(1)}$. The new rigid body modes may be obtained geometrically or as a by-product of a CG solver [162]. Hence applying $\boldsymbol{K}_{(0)}^+$ as a preconditioner for $\boldsymbol{K}_{(1)}$ requires special attention to make sure that one retains the full solution space.

---

[4] Note that in general this is only relevant for the Rubin/MacNeal and Dual Craig-Bampton methods. In the Craig-Bampton method the constraining of the interface DoF usually eliminates any rigid body modes. For more details refer to section 2.5.

6.2.4.2 Implementation Issues of Block Conjugate Gradient Method

As outlined in the introduction, a component's static modes correspond physically to the static deformation shapes in response to an excitation (either displacement or force) at one of its boundary DoF. In practice, where the boundary DoF may be located close to each other, this means that some of the static modes may be very similar in shape. Mathematically speaking, such vectors are (nearly) linearly dependent.

In those cases, solving the updating problem with a block CG method might become problematic. Indeed, when starting the CG iterations with $\boldsymbol{\Psi}_{(0)}$ having linearly dependent columns, the initial residual will not have full rank. As a result, the orthonormalization process of the vectors in the block $\boldsymbol{P}_0$ will break down and the CG algorithm is no longer guaranteed to converge. To overcome this, a check on the rank of the initial residual needs to be performed when initializing the block CG algorithm [66]. If the block of vectors is rank deficient, they need to be orthogonalized such that:

$$\boldsymbol{R}_0 = \boldsymbol{F}_{(1)} - \boldsymbol{K}_{(1)}\boldsymbol{\Psi}_{(0)} = \bar{\boldsymbol{R}}_0\boldsymbol{G} \tag{6.36}$$

Here $\bar{\boldsymbol{R}}_0$ has size $n \times r$ and $\boldsymbol{G}$ has size $r \times p$, with r the rank of $\boldsymbol{R}_0$ and p the original number of vectors in the block. The original problem is then transformed to:

$$\boldsymbol{K}_{(1)}\bar{\boldsymbol{\Psi}}_{(1)} = \bar{\boldsymbol{R}}_0 \tag{6.37}$$

Having solved the transformed problem, the solutions to the original static updating problem can be recovered by:

$$\boldsymbol{\Psi}_{(1)} = \boldsymbol{\Psi}_{(0)} + \bar{\boldsymbol{\Psi}}_{(1)}\boldsymbol{G} \tag{6.38}$$

Another issue encountered in practice is that not all vectors in the block converge at the same rate. Some vectors may thus meet the convergence criterion much sooner than others. In order to minimize the computational effort and, more importantly, to avoid numerical instabilities in the minimization step it is therefore advisable to remove converged vectors from the iteration block.

6.2.4.3 Low Tolerance Updating

One possibility that arises when iteratively updating the static modes is to perform the updating only up to a relatively low tolerance. This idea is motivated by the fact that due to the reduction one already makes an approximation and hence the accuracy of the updated modes needs not to be higher than the accuracy that the reduction imposes on the component model. In case one applies low tolerance updating to the static modes for a fixed interface reduction method (e.g. the Guyan or Craig-Bampton method, see section 2.5), one has to make sure that possible rigid body modes of the structure are properly described. A procedure to achieve this is outlined below.

The rigid body modes of the structure are denoted by $\boldsymbol{\Phi}_r^T = \begin{bmatrix}\boldsymbol{\Phi}_{r|u}^T & \boldsymbol{\Phi}_{r|b}^T\end{bmatrix}$, such that by definition they do not produce any elastic forces:

$$\boldsymbol{K}\begin{bmatrix}\boldsymbol{\Phi}_{r|u}\\ \boldsymbol{\Phi}_{r|b}\end{bmatrix} = \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{0}\end{bmatrix} \tag{6.39}$$

Here $\boldsymbol{\Phi}_r$ has size $n \times r$, with r between 1 and 6 when the structure is assumed to be free of mechanisms. In case the mesh of the model has changed due to the design modification the new rigid body modes may be obtained geometrically which, as explained in section 2.4.2, is computationally cheap. Suppose now that, for the sake of illustration, a component model is reduced using only static modes:

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{q}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Psi} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{q}_b = \boldsymbol{R}_s \boldsymbol{q}_b \tag{6.40}$$

Hence, only the boundary DoF are retained. If a rigid body displacement is imposed on these DoF, the reduced model should not generate any elastic forces such that neighboring components do not experience any reaction forces. This can be interpreted as a patch test for a reduced component model (superelement), similar to a regular finite element. Therefore one should have that:

$$\boldsymbol{R}_s^T \boldsymbol{K} \boldsymbol{R}_s \boldsymbol{\Phi}_{r|b} = \boldsymbol{0} \tag{6.41}$$

One way to satisfy this condition is to impose that:

$$\boldsymbol{R}_s \boldsymbol{\Phi}_{r|b} = \boldsymbol{\Phi}_r \quad \leftrightarrow \quad \boldsymbol{\Psi} \boldsymbol{\Phi}_{r|b} = \boldsymbol{\Phi}_{r|u} \tag{6.42}$$

Since the static modes have been updated with a limited tolerance, this condition does not hold in general. Instead, one has:

$$\boldsymbol{\Psi} \boldsymbol{\Phi}_{r|b} = \boldsymbol{\Phi}_{r|u} + \boldsymbol{\epsilon} \tag{6.43}$$

In order to cancel the error $\boldsymbol{\epsilon}$ and satisfy the rigid body condition in (6.42), a correction is added to the static modes:

$$\boldsymbol{\Psi} = \tilde{\boldsymbol{\Psi}} + \boldsymbol{\Delta} \tag{6.44}$$

The correction can be found from substitution in (6.42), hence

$$\boldsymbol{\Delta} = \left(\boldsymbol{\Phi}_{r|u} - \tilde{\boldsymbol{\Psi}} \boldsymbol{\Phi}_{r|b} \right) \left(\boldsymbol{\Phi}_{r|b}^T \boldsymbol{\Phi}_{r|b} \right)^{-1} \boldsymbol{\Phi}_{r|b}^T, \tag{6.45}$$

where it was assumed that there are more boundary DoF than rigid body modes (i.e. $\boldsymbol{\Phi}_{r|b}$ is a tall matrix). Note that by adding this correction to the updated static modes the approximate CG solution to the original static problem in eq. (6.4) is changed. This could lead to a larger residual, meaning that due to the correction the part of the static modes describing the deformations is less accurately captured. However, this should not be of great concern since one has chosen to compute the static modes with a low tolerance in the first place. Furthermore, correctly describing the rigid body modes is deemed more important in a substructuring analysis.

Finally, it is noted that the (domain) error estimates derived in the previous chapter could be used as a guideline for determining the updating tolerance required for the static modes (as well as for the vibration modes).

6.3 Updating of Vibration Modes

After outlining the updating procedure for static modes in the previous section, this section addresses the updating of that other important ingredient in a component reduction basis: the vibration modes. This updating problem is formulated as follows: suppose that the vibration modes $\boldsymbol{\Phi}_{(0)} = [\boldsymbol{\phi}_{(0),1} \quad \dots \quad \boldsymbol{\phi}_{(0),m}]$ of the nominal component have been obtained. After a design modification the new vibration modes must be found by solving

$$\boldsymbol{K}_{(1)}\boldsymbol{\phi}_{(1),j} = \omega^2_{(1),j}\boldsymbol{M}_{(1)}\boldsymbol{\phi}_{(1),j} \qquad \text{for} \qquad j = 1\dots m \tag{6.46}$$

in an efficient way and without factorizing $\boldsymbol{K}_{(1)}$. To present the proposed strategy, subsection 6.3.1 treats structural dynamic eigensolvers in general and explains their shortcomings for updating purposes. Subsection 6.3.3 therefore addresses an eigensolver that is suited for the updating problem. Some details for further improvement of the updating strategy for the vibration modes are finally treated in subsection 6.3.4.

6.3.1 Eigensolvers in Structural Dynamics

The most important ingredient of any reduction basis is the set of vibration modes $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \quad \dots \quad \boldsymbol{\phi}_m]$. To find these modes requires solving an eigenvalue problem of the form

$$\boldsymbol{K}\boldsymbol{\phi}_j = \omega_j^2\boldsymbol{M}\boldsymbol{\phi}_j. \tag{6.47}$$

Since one is usually interested in only the lowest $m \ll n$ modes and generally the $\boldsymbol{K}$ and $\boldsymbol{M}$ matrices are symmetric, (semi-)positive definite and very sparse, structural dynamic eigenproblems are usually solved using iterative methods. The basis for these methods is the concept of *inverse iteration*, which will be briefly explained next.

6.3.1.1 Inverse Iteration Method

Suppose one wants to find the first eigenmode associated to the eigenproblem in eq. (6.47) and have some arbitrary starting vector $\boldsymbol{x}_0$. Using modal superposition this vector can be written as a combination of eigenmodes:

$$\boldsymbol{x}_0 = \sum_{i=1}^{n} \alpha_i \boldsymbol{\phi}_i \tag{6.48}$$

Next, the initial eigenvalue problem can be written as

$$\sigma_j \boldsymbol{\phi}_j = \boldsymbol{K}^{-1}\boldsymbol{M}\boldsymbol{\phi}_j, \tag{6.49}$$

where $\sigma_j = 1/\omega_j^2$ and consequently $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$. The solution for the first mode $\boldsymbol{\phi}_1$ can be found iteratively by starting from $\boldsymbol{x}_0$ and forming the iterates $\boldsymbol{x}_k$ by:

$$\boldsymbol{x}_{k+1} = \boldsymbol{K}^{-1}\boldsymbol{M}\boldsymbol{x}_k = \boldsymbol{D}\boldsymbol{x}_k \tag{6.50}$$

This method is known as *power iteration*, where the matrix $\boldsymbol{D}$ is referred to as the iteration matrix. After inserting the modal expansion for $\boldsymbol{x}_0$ and k successive applications of the power iteration one thus has:

$$\boldsymbol{x}_k = \boldsymbol{D}^k\boldsymbol{x}_0 = \boldsymbol{D}^k \sum_{i=1}^{n} \alpha_i \boldsymbol{\phi}_i = \sum_{i=1}^{n} \sigma_i^k \alpha_i \boldsymbol{\phi}_i \tag{6.51}$$

This can also be written as:

$$\boldsymbol{x}_k = \sigma_1^k \left(\alpha_1 \boldsymbol{\phi}_1 + \sum_{i=2}^{n} \left(\frac{\sigma_i}{\sigma_1} \right)^k \alpha_i \boldsymbol{\phi}_i \right) \tag{6.52}$$

Since $\sigma_1 \geq \sigma_i$, the ratios $(\sigma_i/\sigma_1)^k$ tend to zero with increasing k and hence $\boldsymbol{x}_k$ converges to $\boldsymbol{\phi}_1$, provided that $\alpha_1 \neq 0$. The convergence rate is governed by the ratio (σ_2/σ_1) and is thus independent of the system size, which can be very attractive for large systems. However, explicitly forming the iteration matrix $\boldsymbol{D}$ becomes infeasible for such systems. Therefore, the algorithm can be split up into two operations, giving the *inverse iteration* scheme:

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{M}\boldsymbol{x}_k \\ \boldsymbol{x}_{k+1} &= \boldsymbol{K}^{-1}\boldsymbol{y}_k \end{aligned} \tag{6.53}$$

The first step is a simple multiplication by the mass matrix, while the section operation corresponds to solving a static problem. Convergence can be monitored by calculating the residual at iteration k:

$$\boldsymbol{r}_k = \boldsymbol{K}^{-1}\boldsymbol{M}\boldsymbol{x}_k - \sigma_k \boldsymbol{x}_k \tag{6.54}$$

The algorithm can be terminated when $\|\boldsymbol{r}_k\| < \varepsilon \|\boldsymbol{K}\boldsymbol{x}_k\|$, where ε is some preset tolerance. Since in structural dynamics one is often interested in the convergence of the eigenfrequency, an alternative convergence criterion can be based on the *Rayleigh quotient*:

$$\rho_k = \frac{\boldsymbol{x}_k^T \boldsymbol{K} \boldsymbol{x}_k}{\boldsymbol{x}_k^T \boldsymbol{M} \boldsymbol{x}_k} \tag{6.55}$$

such that the algorithm is stopped when $\|\rho_{k+1} - \rho_k\| < \varepsilon \|\rho_k\|$, where ε is now a tolerance on the frequency. The eigenfrequency is then found as $\omega = \sqrt{\rho_k}$. A straightforward relation exists between the residual criterion and the frequency criterion. Knowing that at iteration k the mode approximation has converged up to a tolerance ε, one can write:

$$\boldsymbol{x}_k = \boldsymbol{\phi}_1 + \varepsilon \boldsymbol{y} = \boldsymbol{\phi}_1 + \varepsilon \sum_{i=2}^{n} \alpha_i \boldsymbol{\phi}_i \tag{6.56}$$

Computing the Rayleigh quotient for this mode approximation, assuming mass normalized modes and using mode orthogonality, it is found that:

$$\rho_k = \frac{(\boldsymbol{\phi}_1 + \varepsilon \boldsymbol{y})^T \boldsymbol{K} (\boldsymbol{\phi}_1 + \varepsilon \boldsymbol{y})}{(\boldsymbol{\phi}_1 + \varepsilon \boldsymbol{y})^T \boldsymbol{M} (\boldsymbol{\phi}_1 + \varepsilon \boldsymbol{y})} = \frac{\omega_1^2 + \varepsilon^2 \boldsymbol{y}^T \boldsymbol{K} \boldsymbol{y}}{1 + \varepsilon^2} \approx \omega_1^2 + \varepsilon^2 \sum_{i=2}^{n} \alpha_i^2 \omega_i^2 \tag{6.57}$$

This shows that when a tolerance ε is requested for the residual, the associated frequency has converged with an accuracy in order of approximately ε^2.

After the first eigenmode is found, higher modes can be found by orthogonal deflation. Furthermore, in case one is interested in the eigenmodes around a certain target frequency μ, one can apply spectral shifting and solve the following problem:

$$(\boldsymbol{K} - \mu \boldsymbol{M}) \boldsymbol{\phi}_j = \left(\omega_j^2 - \mu \right) \boldsymbol{M} \boldsymbol{\phi}_j. \tag{6.58}$$

This requires the factorization of the shifted stiffness matrix (also known as the dynamic stiffness) so that the inverse iteration step becomes:

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{M}\boldsymbol{x}_k \\ \boldsymbol{x}_{k+1} &= \left(\boldsymbol{K} - \mu\boldsymbol{M}\right)^{-1}\boldsymbol{y}_k \end{aligned} \tag{6.59}$$

Using the same reasoning as before, one can easily show that this inverse iteration sequence converges to the eigenmode closest to the target frequency μ. Finally, one can also choose to integrate the shifting in the iterations. This means that at every iteration, one calculates the eigenfrequency estimate (i.e. the Rayleigh quotient) and shifts the problem accordingly. This is called Rayleigh quotient iteration [77] and the inverse iteration step becomes:

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{M}\boldsymbol{x}_k \\ \boldsymbol{x}_{k+1} &= \left(\boldsymbol{K} - \rho_k\boldsymbol{M}\right)^{-1}\boldsymbol{y}_k \end{aligned} \tag{6.60}$$

This iteration scheme gives an eigensolver with extremely fast convergence, which comes at the high cost of having to factorize a different linear system at every iteration. The approach is therefore only feasible for relatively small systems, as the cost of the factorization becomes prohibitive for larger systems. For more details regarding power and inverse iteration methods see for instance [77, 74].

6.3.1.2 Subspace Iteration Method

Using the inverse iteration method to find the vibration modes of a structure has some shortcomings. Most notably, each eigensolution is calculated from scratch and converges without knowledge of higher eigensolutions. This means that information from previous solutions is thrown away and the convergence is slow when eigenvalues are closely spaced. Much like the CG algorithm in the previous section, the aim is to modify the method such that these drawbacks are eliminated. The concept of inverse iteration can therefore be generalized such that iterations are performed on several vectors simultaneously, leading to the well known *subspace iteration* method.

Instead of iterating on a single vector, the subspace method iterates on a block of m vectors. The iteration step in the algorithm thus becomes:

$$\begin{aligned} \boldsymbol{Y}_k &= \boldsymbol{M}\boldsymbol{X}_k \\ \boldsymbol{X}_{k+1} &= \boldsymbol{K}^{-1}\boldsymbol{Y}_k \end{aligned} \tag{6.61}$$

To avoid that all m vectors in $\boldsymbol{X}_k$ converge towards the first eigenmode, an orthogonalization step is needed at every iteration to make sure that the columns of $\boldsymbol{X}_k$ span a m-dimensional subspace. This orthogonalization can be accomplished by using a Gram-Schmidt procedure or by solving the *interaction problem*. The interaction problem amounts to finding the best estimate for the eigensolutions in the space spanned by $\boldsymbol{X}_k$. Hence one forms the reduced matrices:

$$\begin{aligned} \tilde{\boldsymbol{K}} &= \boldsymbol{X}_k^T\boldsymbol{K}\boldsymbol{X}_k \\ \tilde{\boldsymbol{M}} &= \boldsymbol{X}_k^T\boldsymbol{M}\boldsymbol{X}_k \end{aligned} \tag{6.62}$$

and subsequently solves the reduced eigenproblem:

$$\tilde{\boldsymbol{K}}\boldsymbol{V}_k = \tilde{\boldsymbol{M}}\boldsymbol{V}_k\boldsymbol{\rho}_k \tag{6.63}$$

Since it is very small, this eigenproblem can be solved efficiently by a direct method. The resulting eigenvalue estimates are $\boldsymbol{\rho}_k = \operatorname{diag}\left(\rho_{k,1} \dots \rho_{k,m}\right)$ and the eigenmode estimates can be found from expansion of the reduced modes:

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k \boldsymbol{V}_k \tag{6.64}$$

These eigenmode estimates are $\boldsymbol{K}$- and $\boldsymbol{M}$-orthogonal. As a convergence criterion one can again choose to either iterate until the residual drops below a certain threshold

$$\left\|\boldsymbol{R}_k\right\| = \left\|\boldsymbol{K}\boldsymbol{X}_k - \boldsymbol{M}\boldsymbol{X}_k\boldsymbol{\rho}_k\right\| < \varepsilon \left\|\boldsymbol{K}\boldsymbol{X}_k\right\|, \tag{6.65}$$

or monitor the difference between subsequent eigenfrequency estimates:

$$\left\|\boldsymbol{\rho}_k - \boldsymbol{\rho}_{k-1}\right\| < \varepsilon \left\|\boldsymbol{\rho}_k\right\| \tag{6.66}$$

Similar to the inverse iteration method, the convergence rate of the subspace iteration method for mode j is driven by the ratio $\left(\omega_j^2/\omega_{m+1}^2\right)$. Therefore, in order to increase the convergence rate towards the highest required eigenvalue ω_m^2 it might be interesting to add so-called *buffer vectors* to the iteration block $\boldsymbol{X}$. The goal is then to choose the number of buffer vectors such that the convergence speeds up while at the same time the total number of operations is not increased by too much. As a rule of thumb one usually chooses the total number of iteration vectors according to $\min\left\{2m, m+8\right\}$. The subspace iteration method is summarized in algorithm 6.3; more details can be found in [19].

Algorithm 6.3 Subspace iteration method

```
Input: $\boldsymbol{K}, \boldsymbol{M}, \boldsymbol{K}^{-1}, \boldsymbol{X}_0 : \boldsymbol{X}_0^T\boldsymbol{M}\boldsymbol{X}_0 = \boldsymbol{I}$
$\boldsymbol{\rho}_0 = \operatorname{diag}\left(\boldsymbol{X}_0^T\boldsymbol{K}\boldsymbol{X}_0\right),\ k = 1$
while $\|\boldsymbol{\rho}_k - \boldsymbol{\rho}_{k-1}\| > \varepsilon\|\boldsymbol{\rho}_k\|$ do
    – Inverse iteration –
    $\boldsymbol{Y}_k = \boldsymbol{M}\boldsymbol{X}_{k-1}$
    $\boldsymbol{X}_k = \boldsymbol{K}^{-1}\boldsymbol{Y}_k$
    – Interaction analysis & orthogonalization –
    $\tilde{\boldsymbol{K}} = \boldsymbol{X}_k^T\boldsymbol{K}\boldsymbol{X}_k$
    $\tilde{\boldsymbol{M}} = \boldsymbol{X}_k^T\boldsymbol{M}\boldsymbol{X}_k$
    Solve: $\tilde{\boldsymbol{K}}\boldsymbol{V}_k = \tilde{\boldsymbol{M}}\boldsymbol{V}_k\boldsymbol{\rho}_k$
    $\boldsymbol{X}_{k+1} = \boldsymbol{X}_k\boldsymbol{V}_k$
    $k = k + 1$
end while
$\boldsymbol{\Phi} = \boldsymbol{X}_k$
$\boldsymbol{\Omega}^2 = \boldsymbol{\rho}_k$
```

6.3.1.3 Lanczos Method

Although the subspace iteration method is an efficient and robust method for computing the lowest few eigenmodes of a structure, it still has some flaws. Most importantly, at every iteration the information from previous steps is disregarded. Intuitively, this is not the most efficient way of converging to the solution and in fact one would like to use as much of that previous information as possible. The Lanczos method, first published in 1950 [121], does exactly that and will be briefly outlined next.

Conceptually the Lanczos method for solving eigenproblems is very similar to the CG method for static problems. The Lanczos method also generates a growing, orthogonal subspace and seeks for the eigensolutions in this space. Starting from an initial vector $\boldsymbol{z}_0$ that is mass normalized (i.e. $\boldsymbol{z}_0^T \boldsymbol{M} \boldsymbol{z}_0 = 1$), the first step in the Lanczos algorithm is the inverse iteration to find

$$\tilde{\boldsymbol{z}}_1 = \boldsymbol{K}^{-1} \boldsymbol{M} \boldsymbol{z}_0 \tag{6.67}$$

From the inverse iteration method it is known that if this process is continued the subsequent iterates will all converge towards the first mode. To avoid this, the following iterate is $\boldsymbol{M}$-orthogonalized to the previous one, much like the $\boldsymbol{K}$-orthogonality of the CG iterates:

$$\begin{aligned} \alpha_0 &= \boldsymbol{z}_0^T \boldsymbol{M} \tilde{\boldsymbol{z}}_1 \\ \hat{\boldsymbol{z}}_1 &= \tilde{\boldsymbol{z}}_1 - \boldsymbol{z}_0 \alpha_0 \end{aligned} \tag{6.68}$$

Hereafter $\hat{\boldsymbol{z}}_1$ is mass normalized according to:

$$\begin{aligned} \gamma_1 &= \sqrt{\hat{\boldsymbol{z}}_1^T \boldsymbol{M} \hat{\boldsymbol{z}}_1} \\ \boldsymbol{z}_1 &= \hat{\boldsymbol{z}}_1 / \gamma_1 \end{aligned} \tag{6.69}$$

This process is repeated, so that after k iterations the following Krylov sequence is created:

$$\mathcal{K}_k \left(\boldsymbol{K}^{-1} \boldsymbol{M}, \boldsymbol{z}_0 \right) = \text{span} \left\{ \boldsymbol{z}_0, \boldsymbol{K}^{-1} \boldsymbol{M} \boldsymbol{z}_0, \left(\boldsymbol{K}^{-1} \boldsymbol{M} \right)^2 \boldsymbol{z}_0, \ldots, \left(\boldsymbol{K}^{-1} \boldsymbol{M} \right)^{k-1} \boldsymbol{z}_0 \right\} \tag{6.70}$$

The eigensolutions of the problem are searched for in this growing Krylov subspace. Similar to the subspace iteration method, one therefore needs to solve the interaction problem to find the approximate first m eigensolutions. After $k \geq m$ iterations, this interaction problem turns out to take the following form (see [121, 77, 74] for more details):

$$\boldsymbol{T} \boldsymbol{V}_k = \boldsymbol{\sigma}_k \boldsymbol{V}_k \tag{6.71}$$

Here $\boldsymbol{\sigma}_k = \text{diag}(\sigma_1 \ldots \sigma_k)$ contains the reciprocals of the eigenvalue estimates ($\sigma_j = 1/\rho_j^2$); the eigenmode approximations are again found by expansion, i.e. $\boldsymbol{X}_k = \boldsymbol{Z}_k \boldsymbol{V}_k$. The matrix $\boldsymbol{T}$ is built from the coefficients $\alpha_\star$ and $\gamma_\star$ and it can be shown that, due to orthogonality and symmetry of $\boldsymbol{K}$ and $\boldsymbol{M}$, this matrix is symmetric and tri-diagonal:

$$\boldsymbol{T} = \begin{bmatrix} \alpha_0 & \gamma_1 & & \boldsymbol{0} \\ \gamma_1 & \alpha_1 & \ddots & \\ & \ddots & \ddots & \gamma_k \\ \boldsymbol{0} & & \gamma_k & \alpha_k \end{bmatrix}$$

Due to this form of $\boldsymbol{T}$ the interaction problem can be solved very efficiently. The convergence criterion can again be taken either on the norm of residual of the eigenvalue problem or on the differences between subsequent eigenfrequency estimates.

As pointed out before, the Lanczos method can be seen as the counterpart of the CG method for eigenproblems; both are Krylov subspace methods. Comparing the Lanczos method to the inverse iteration and subspace iteration eigensolvers, the main difference is that no information is discarded since all iterates are used to find the best approximation of the eigensolutions. As such, the Lanczos method is the most efficient way of calculating the lowest few eigenmodes of a sparse, symmetric system. Further details of the method are treated in [74]. The discussion on the Lanczos method is concluded with the following remarks:

- Similar to the CG algorithm, theoretically the current vector $\boldsymbol{z}_{k+1}$ only needs to be orthogonalized with respect to the previous vector $\boldsymbol{z}_k$. However, due to accumulation of numerical errors, full orthogonalization is needed to preserve the tri-diagonality of $\boldsymbol{T}$.
- Instead of iterating on a single vector, the Lanczos method can also be written in a block form to iterate on a block of vectors. As pointed out for the CG method, this might have advantages for modern CPUs that can run more efficiently on a block implementation.
- The Lanczos method makes use of the symmetry of the mass and stiffness matrices. The Arnoldi method is a generalization of the Lanczos method for non-symmetric systems and forms the basis for the *Implicitly Restarted Arnoldi* solver often used in structural dynamics [195]. To limit memory use when dealing with very large systems, this method consists in restarting the iterations after some number of steps using only the most relevant results obtained so far.

6.3.2 Eigensolvers & the Updating Problem

Having introduced the two most common eigensolvers in structural dynamics, namely the subspace iteration and Lanczos methods, the discussion is now returned to the updating problem for the component vibration modes. Recall that in this work it is tried to avoid the factorization of the modified stiffness matrix, which cannot be done using the traditional solvers. Hence, an alternative algorithm is desired. The most straightforward idea is to replace the inverse iteration (requiring solving a static problem) of the traditional algorithms with an iterative algorithm such as the CG method outlined in the previous section. This results in an algorithm consisting of two nested iteration loops called *inner-outer iteration*: the original iteration of the eigensolver is called the outer iteration while the static problem in eq. (6.67) is solved by the inner iteration loop.

For the Lanczos method, this approach turns out to be very inefficient. This can be understood as follows. For the sake of illustration, suppose the first eigensolution is sought of a system with the mass matrix equal to identity:

$$\boldsymbol{K}\boldsymbol{\phi}_1 = \omega_1^2 \boldsymbol{I}\boldsymbol{\phi}_1 \tag{6.72}$$

After k iterations of the Lanczos method an approximation $\boldsymbol{x}_k$ of the first is searched for in the Krylov subspace:

$$\boldsymbol{x}_k \in \operatorname{span}\{\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_k\} = \operatorname{span}\left\{\boldsymbol{z}_0, \boldsymbol{K}^{-1}\boldsymbol{z}_0, \ldots, \boldsymbol{K}^{-k+1}\boldsymbol{z}_0\right\} \tag{6.73}$$

Note that due to orthogonalization of the $\boldsymbol{z}_k$ the vectors are not exactly identical to those in the Krylov sequence, but since the orthogonalization process involves only linear operations the space spanned remains the same. To improve the approximation $\boldsymbol{x}_k$, the Krylov basis is then extended by adding the direction resulting from the inverse iteration step:

$$\boldsymbol{K}\boldsymbol{z}_{k+1} = \boldsymbol{z}_k \tag{6.74}$$

However, since it is wished not to compute $\boldsymbol{K}^{-1}$, a CG method is applied to solve the inverse iteration step: the inner iteration. This means that the new vector $\boldsymbol{z}_{k+1}$ is searched for in the following Krylov subspace (assuming the CG iterations are started with $\boldsymbol{z}_{k+1,0} = \boldsymbol{0}$):

$$\boldsymbol{z}_{k+1} \in \operatorname{span}\{\boldsymbol{z}_k, \boldsymbol{K}\boldsymbol{z}_k, \boldsymbol{K}^2\boldsymbol{z}_k, \ldots\} = \operatorname{span}\{\boldsymbol{z}_k, \boldsymbol{z}_{k-1}, \boldsymbol{z}_{k-2}, \ldots, \boldsymbol{z}_0, \boldsymbol{K}\boldsymbol{z}_0, \boldsymbol{K}^2\boldsymbol{z}_0, \ldots\} \tag{6.75}$$

This indicates that the CG algorithm starts by finding the part of $\boldsymbol{z}_{k+1}$ residing in the space of all previous $\boldsymbol{z}_{k-i}$ $(i = 0 \ldots k)$. This part of $\boldsymbol{z}_{k+1}$ is useless since it is already present in the Lanczos subspace and it will anyways be filtered out of $\boldsymbol{z}_{k+1}$ in the subsequent Lanczos orthogonalization step. The new contribution to the Lanczos space will be generated by the CG iterations from the Krylov space $\operatorname{span}\{\boldsymbol{K}\boldsymbol{z}_0, \boldsymbol{K}^2\boldsymbol{z}_0, \ldots\}$ which will first approximate the higher spectrum of the problem and is thus in contradiction with the fact that the solution $\boldsymbol{z}_{k+1}$ is expected to be relevant for the low spectrum in the Lanczos space. In other words, the useful part of the solution will be obtained only after many CG iterations.

Preconditioning the CG algorithm might help to reduce the number of iterations needed, but this fundamental problem remains. Furthermore, speeding up the convergence by reusing the iteration results in subsequent CG solves through the projection/reconjugation technique (see section 6.2.3) will also only have limited effect. The reason is that at each step one needs to solve the inverse iteration for a right hand side that is orthogonal to the previous ones.

Alternatively one could consider solving the inverse iteration step (i.e. the inner iteration) only approximately, called *inexact inverse iteration* in the literature [78]. However, the Lanczos method is not well suited to such an approach as the Krylov subspace in eq. (6.70) is no longer formed exactly. Indeed, it was already noted in the literature that the Lanczos method is sensitive to perturbations [80]; especially the first few inverse iteration steps need to be solved to high accuracy in order to preserve its fast convergence.

The above discussion motivates the need for a different approach. It is therefore proposed to apply the *inverse-free preconditioned Krylov subspace* (IFPKS) method for handling the updating problem for the vibration modes. This method has only recently been developed in the field of computational mathematics [79, 153, 154] and will be discussed in detail next.

6.3.3 Inverse-Free Preconditioned Krylov Subspace Method

6.3.3.1 Basic Concept

To understand the concept of the IFPKS method, it should be realized that the eigenproblem is in fact a non-linear minimization problem. Considering again the eigenproblem for a single vector, one namely tries to find a local minimum of the Rayleigh quotient:

$$\rho(\boldsymbol{x}) = \frac{\boldsymbol{x}^T\boldsymbol{K}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{M}\boldsymbol{x}} \tag{6.76}$$

Suppose one has some initial guess $\boldsymbol{x}_0$ for the eigenvector. In case one assumes $\boldsymbol{x}_0$ to be mass normalized, the gradient around $\boldsymbol{x}_0$ can be computed as follows:

$$\frac{1}{2}\frac{\partial \rho}{\partial \boldsymbol{x}}\bigg|_{\boldsymbol{x}_0} = \boldsymbol{K}\boldsymbol{x}_0 - \rho\left(\boldsymbol{x}_0\right)\boldsymbol{M}\boldsymbol{x}_0 = \boldsymbol{r}_0 \tag{6.77}$$

Note that the residual is also dependent on $\rho(\boldsymbol{x}_0)$ which clearly indicates that it is a residual of a non-linear problem. As in the CG method, $\boldsymbol{r}_0$ is the residual corresponding to the initial guess $\boldsymbol{x}_0$ and is the direction of steepest descent. In order to improve the eigenmode estimate, one can thus search for a new approximation $\boldsymbol{x}_1$ in the space span $\{\boldsymbol{x}_0, \boldsymbol{r}_0\}$ by minimizing $\rho\left(\boldsymbol{x}_1\right)$. However, since this is a non-linear problem, other directions orthogonal to the residual might also be useful to construct a better approximation $\boldsymbol{x}_1$. Hence one generates a set of vectors that span the following Krylov subspace:

$$\mathcal{K}_r\left(\boldsymbol{K} - \rho_0\boldsymbol{M}, \boldsymbol{r}_0\right) = \operatorname{span}\left\{\boldsymbol{r}_0, \left(\boldsymbol{K} - \rho_0\boldsymbol{M}\right)\boldsymbol{r}_0, \ldots, \left(\boldsymbol{K} - \rho_0\boldsymbol{M}\right)^{r-1}\boldsymbol{r}_0\right\} \tag{6.78}$$

Here $r \geq 1$ is some chosen number of inner iterations. The new approximate eigensolution $\boldsymbol{x}_1$ is then found in the space

$$\boldsymbol{x}_1 \in \operatorname{span}\left\{\boldsymbol{x}_0\right\} + \mathcal{K}_r\left(\boldsymbol{K} - \rho_0\boldsymbol{M}, \boldsymbol{r}_0\right) = \mathcal{K}_{r+1}\left(\boldsymbol{K} - \rho_0\boldsymbol{M}, \boldsymbol{x}_0\right) \tag{6.79}$$

by solving the interaction problem. The associated eigenvalue estimate ρ_1 can then be used to calculate the residual, subsequently generate the Krylov subspace and repeat the process until convergence. This idea forms the basis of the inverse-free Krylov subspace method as introduced in [79], the associated basic algorithm is shown in alg. 6.4. Note that in case $r = 1$ the method reduces to the steepest descent method, which can be shown to converge to an eigensolution of $(\boldsymbol{K}, \boldsymbol{M})$; for the proof and detailed convergence analysis the reader is referred to [79, 153]. Finally, it is noted that a similar idea was already proposed in 1971, although in that work the aim was to minimize mainly the computer memory usage [72].

Algorithm 6.4 Inverse-free Krylov subspace method

```
Input: $\boldsymbol{K}, \boldsymbol{M}, \boldsymbol{x}_0 : \boldsymbol{x}_0^T\boldsymbol{M}\boldsymbol{x}_0 = 1, r$
$\rho_0 = \boldsymbol{x}_0^T\boldsymbol{K}\boldsymbol{x}_0, k = 0$
while $||\rho_k - \rho_{k-1}|| > \varepsilon||\rho_k||$ do
    – Generate Krylov subspace –
    $\boldsymbol{Z} = [\boldsymbol{z}_0, \ldots, \boldsymbol{z}_r]$ such that span $\{\boldsymbol{Z}\} = \mathcal{K}_{r+1}(\boldsymbol{K} - \rho_k\boldsymbol{M}, \boldsymbol{x}_k)$
    – Interaction analysis & orthogonalization –
    $\tilde{\boldsymbol{K}} = \boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}, \tilde{\boldsymbol{M}} = \boldsymbol{Z}^T\boldsymbol{M}\boldsymbol{Z}$
    Solve: $\tilde{\boldsymbol{K}}\boldsymbol{v} = \rho_{k+1}\tilde{\boldsymbol{M}}\boldsymbol{v}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{Z}\boldsymbol{v}$
    $k = k + 1$
end while
$\boldsymbol{\phi} = \boldsymbol{x}_k$
$\omega^2 = \rho_k$
```

The actual generation of the Krylov subspace $\boldsymbol{Z}$ can be achieved in different ways. They all involve recursive application of the shifted matrix $(\boldsymbol{K} - \rho_k\boldsymbol{M})$ and subsequent orthogonalization. As such, this process is very similar to the CG and Lanczos processes described in sections 6.2.1 and 6.3.1.3, respectively. Details are addressed in [79].

6.3.3.2 Preconditioning and Block Generalization

In its form described above, the inverse-free algorithm is not yet very useful for handling the vibration mode updating problem. This has two main causes:

- The convergence rate is dependent on the spectral distribution of the shifted matrix. Suppose the aim is to converge to ω_1^2, the first eigenfrequency of $(\boldsymbol{K}, \boldsymbol{M})$, and take this as the shifting frequency. Let $\sigma_n \geq \cdots \geq \sigma_2 > \sigma_1 = 0$ then be the eigenvalues of the shifted matrix $\boldsymbol{K} - \omega_1^2 \boldsymbol{M}$ (with n the size of the problem) and assume that for the current estimate it holds that $\omega_1^2 \leq \rho_k \leq \omega_2^2$. For this case, the following bound for the convergence rate was derived in [79]:

  $$\frac{\rho_{k+1} - \omega_1^2}{\rho_k - \omega_1^2} \leq 4 \left(\frac{1 - \sqrt{\gamma}}{1 + \sqrt{\gamma}} \right)^{2r} + \mathcal{O}\left((\rho_k - \omega_1^2)^{1/2} \right) \quad \text{with} \quad \gamma = \frac{\sigma_2 - \sigma_1}{\sigma_n - \sigma_1} = \frac{\sigma_2}{\sigma_n} \tag{6.80}$$

  Hence, the convergence rate depends on the distribution of the eigenvalues of the shifted matrix instead of those of $(\boldsymbol{K}, \boldsymbol{M})$ as in the Lanczos method. Consequently, the shifted matrix may be preconditioned to improve its spectral distribution and thereby accelerating convergence, independently from the spectrum of the original eigenproblem. The preconditioner is then applied in a way similar to the linear case. In fact, it is proposed to use the same preconditioner, namely the factorization of the nominal stiffness matrix; its effect will be discussed more extensively in section 6.3.4.1. Furthermore, eq. (6.80) shows that convergence accelerates with increasing order r of the Krylov subspace this will be illustrated in chapter 8.

- Only one eigenvalue is calculated at a time. Similar to the inverse iteration method, subsequent eigensolutions can be calculated by deflation, but this has the disadvantage that information from previous solutions is thrown away and convergence is slow for closely spaced modes. Therefore a more efficient approach is to generalize the inverse-free method to a block form, much like the subspace algorithm is a block generalization of the inverse iteration.

Next the preconditioned block form of the inverse-free method as presented in [154] will be discussed, which will be referred to as the inverse-free preconditioned Krylov subspace (IFPKS) method.

Instead of iterating on a single vector, the block version iterates on a number of vectors simultaneously. To this end, the iterations are started with a set of initial approximations for the eigenvectors $\boldsymbol{X}_0 = [\boldsymbol{x}_{1,0} \dots \boldsymbol{x}_{m,0}]$ that are mass normalized such that $\boldsymbol{X}_0^T \boldsymbol{M} \boldsymbol{X}_0 = \boldsymbol{I}$. The associated Rayleigh quotients can then be calculated as:

$$\boldsymbol{\rho}_0 = \operatorname{diag}\left(\boldsymbol{X}_0^T \boldsymbol{K} \boldsymbol{X}_0 \right) \tag{6.81}$$

In order to improve the eigensolution approximations, the next steps are again:

1. Generate the Krylov basis from the shifted matrix
2. Solve the interaction problem in the Krylov space
3. Update the eigensolution approximations

Since one is dealing with m eigenvalue approximations, there is no longer a single shifted matrix. Therefore, [154] outlines a process for generating the Krylov subspace for the block of vectors simultaneously. This process, which constitutes the inner loop of the algorithm, is very similar to that of the block CG as treated in section 6.2.3.2. The inner iterations are started by simply taking the previous eigenvector estimates as the first block of search directions:

$$\boldsymbol{Z}_0 = \boldsymbol{X}_0 \tag{6.82}$$

The new set of search directions is found by application of the shifted vectors to the previous directions, so:

$$\tilde{\boldsymbol{Z}}_1 = \boldsymbol{S}^{-1} \left(\boldsymbol{K}\boldsymbol{Z}_0 - \boldsymbol{M}\boldsymbol{Z}_0\boldsymbol{\rho}_0 \right) \tag{6.83}$$

Here $\boldsymbol{S}$ is the preconditioner. In order to avoid searching in the existing subspace, these vectors are block orthogonalized with respect to $\boldsymbol{Z}_0$ like in the block CG method:

$$\boldsymbol{Z}_1 = \tilde{\boldsymbol{Z}}_1 - \boldsymbol{Z}_0\boldsymbol{\beta}_1 \quad \text{with} \quad \boldsymbol{\beta}_1 = \boldsymbol{Z}_0^T\boldsymbol{M}\tilde{\boldsymbol{Z}}_1 \tag{6.84}$$

Finally, the block of vectors need to be mass orthonormalized such that $\boldsymbol{Z}_1^T\boldsymbol{M}\boldsymbol{Z}_1 = \boldsymbol{I}$, similar as in the block CG method. This loop is repeated r times to obtain the space of the desired size, namely $\boldsymbol{Z} = \left[\boldsymbol{Z}_0 \dots \boldsymbol{Z}_r\right]$. Note that this algorithm requires two matrix-vector multiplications for every iteration: one by $\boldsymbol{K}$ and one by $\boldsymbol{M}$. In a practical implementation these products can be stored and reused to save computations when forming the reduced matrices for the interaction problem.

After the space $\boldsymbol{Z}$ has been generated, the interaction problem can be solved. This leads to new eigensolution estimates $\boldsymbol{\rho}_1$ and $\boldsymbol{X}_1$, for which the above described process can be repeated to obtain a new approximation space. This whole process is repeated until convergence, again monitored either using a frequency or residual vector criterion. The IFPKS method is summarized in algorithm 6.5.

6.3.3.3 Alternative Interpretation

In order to put the IFPKS method into perspective with respect to the existing eigensolvers, this subsection addresses an alternative interpretation of this method, namely as being a variant of the subspace iteration method. The only difference with respect to the normal subspace algorithm is that the inverse iteration step is replaced by an inner iteration loop, which is motivated as follows. Starting again from $\boldsymbol{X}_0$, suppose that the aim is to improve the eigenvector approximations by an inverse iteration (i.e. eq. (6.61)):

$$\boldsymbol{X}_1 = \boldsymbol{K}^{-1}\boldsymbol{M}\boldsymbol{X}_0 \tag{6.85}$$

Since the factorization of $\boldsymbol{K}$ is not to be computed this linear system needs to be solved iteratively using a CG method. As pointed out before, the convergence of this iterative process will be very slow. To overcome this, spectral shifting can be applied such that the linear problem has a more favorable spectral distribution and a useful approximation space is obtained after fewer iterations. For every vector $\boldsymbol{x}_i$ ($i = 1 \dots m$) in $\boldsymbol{X}$ one thus iteratively solves:[5]

$$\left(\boldsymbol{K} - \rho_{i,0}\boldsymbol{M}\right)\boldsymbol{x}_{i,1} = \boldsymbol{M}\boldsymbol{x}_{i,0} \tag{6.86}$$

---

[5] In practice one would probably use a block algorithm, this notation is for illustrative purposes only.

Algorithm 6.5 Inverse-free preconditioned Krylov subspace method

```
Input: $\boldsymbol{K}, \boldsymbol{M}, \boldsymbol{S}^{-1}, \boldsymbol{X}_0 : \boldsymbol{X}_0^T \boldsymbol{M} \boldsymbol{X}_0 = \boldsymbol{I}$
$\boldsymbol{\rho}_0 = \text{diag}\left(\boldsymbol{X}_0^T \boldsymbol{K} \boldsymbol{X}_0\right), k = 0$
while $||\boldsymbol{\rho}_k - \boldsymbol{\rho}_{k-1}|| > \varepsilon ||\boldsymbol{\rho}_k||$ do
    – Generate Krylov subspace –
    $\boldsymbol{Z}_0 = \boldsymbol{X}_k$
    for $j = 0 \ldots r$ do
        $\boldsymbol{Z}_{j+1} = \boldsymbol{S}^{-1}\left(\boldsymbol{K}\boldsymbol{Z}_j - \boldsymbol{M}\boldsymbol{Z}_j \boldsymbol{\rho}_k\right)$
        for $i = 0 \ldots j$ do
            $\boldsymbol{\beta}_i = \boldsymbol{Z}_i^T \boldsymbol{M} \boldsymbol{Z}_{j+1}$
            $\boldsymbol{Z}_{j+1} = \boldsymbol{Z}_{j+1} - \boldsymbol{Z}_i \boldsymbol{\beta}_i$
        end for
        Orthonormalize $\boldsymbol{Z}_{j+1} : \boldsymbol{Z}_{j+1}^T \boldsymbol{M} \boldsymbol{Z}_{j+1} = \boldsymbol{I}$
    end for
    $\boldsymbol{Z} = \left[\boldsymbol{Z}_0 \ldots \boldsymbol{Z}_r\right]$
    – Interaction analysis & orthogonalization –
    $\tilde{\boldsymbol{K}} = \boldsymbol{Z}^T \boldsymbol{K} \boldsymbol{Z}$
    Solve: $\tilde{\boldsymbol{K}} \boldsymbol{V} = \boldsymbol{V} \boldsymbol{\rho}_{k+1}$ with $\boldsymbol{V}^T \boldsymbol{V} = \boldsymbol{I}$
    $\boldsymbol{X}_{k+1} = \boldsymbol{Z}\boldsymbol{V}$
    $k = k + 1$
end while
$\boldsymbol{\Phi} = \boldsymbol{X}_k$
$\boldsymbol{\Omega}^2 = \boldsymbol{\rho}_k$
```

Note that, as one is now in fact applying Rayleigh quotient iteration, the eigenproblem also converges faster. When solving the linear system of eq. (6.86) using the CG method, the solution is sought in the Krylov subspace:

$$\boldsymbol{x}_{i,1} \in \mathcal{K}\left(\boldsymbol{K} - \boldsymbol{\rho}_{i,0}\boldsymbol{M}, \boldsymbol{M}\boldsymbol{x}_{i,0}\right) = \text{span}\left\{\boldsymbol{M}\boldsymbol{x}_{i,0}, \left(\boldsymbol{K} - \boldsymbol{\rho}_{i,0}\boldsymbol{M}\right)\boldsymbol{M}\boldsymbol{x}_{i,0}, \left(\boldsymbol{K} - \boldsymbol{\rho}_{i,0}\boldsymbol{M}\right)^2 \boldsymbol{M}\boldsymbol{x}_{i,0}, \ldots\right\} \tag{6.87}$$

The $\boldsymbol{x}_{i,1}$ obtained in this way are the best approximate solutions to the linear system. However, these are not necessarily the best choice for approximating the desired eigenvectors. The particularity of the IFPKS method is therefore to generate a slightly different space instead, such that a best approximation for the eigenproblem (i.e. the minimization of the Rayleigh quotients) is found. As explained before, this space is spanned by the Krylov sequence:

$$\boldsymbol{x}_{i,1} \in \mathcal{K}\left(\boldsymbol{K} - \boldsymbol{\rho}_{i,0}\boldsymbol{M}, \boldsymbol{x}_{i,0}\right) = \text{span}\left\{\boldsymbol{x}_{i,0}, \left(\boldsymbol{K} - \boldsymbol{\rho}_{i,0}\boldsymbol{M}\right)\boldsymbol{x}_{i,0}, \left(\boldsymbol{K} - \boldsymbol{\rho}_{i,0}\boldsymbol{M}\right)^2 \boldsymbol{x}_{i,0}, \ldots\right\} \tag{6.88}$$

Although the difference is subtle, the latter space gives a better approximation especially when the Rayleigh quotients in $\boldsymbol{\rho}$ are not yet very close to the desired eigenvalues [79].

In summary, the IFPKS method can be understood as a subspace iteration method where the inverse iteration step is replaced by the generation of a Krylov subspace. This is allowed since the subspace iteration method, in contrast to the Lanczos method, does not dictate the inverse iteration step to be solved with high accuracy. Indeed, at each iteration, the eigensolutions are sought only in the current subspace and there is no dependency on the previous iterates.

6.3.3.4 Practical Issues

When implementing the IFPKS method as outlined in algorithm 6.5, there are a few practical issues one needs to be aware of. The most important ones are:

- In its form of alg. 6.5, the IFPKS method iterates on blocks of vectors. These blocks contain up to m vectors, where m is the desired number of eigenmodes. As pointed out for the CG method, the efficiency of the iterative process in the inner iteration loop can break down for large block sizes. Hence, one might consider iterating on only a few vectors simultaneously in order to maximize CPU efficiency. In this case one has to apply orthogonal deflation techniques in order to prevent subsequent (blocks of) vectors from also converging to the lowest eigensolutions.
- Similar to the subspace iteration method, the use of buffer vectors may be considered for the IFPKS method by adding b vectors to the block $\boldsymbol{X}$. The sole purpose of the buffer vectors is to speed up the convergence of the first m vectors, so the iterations are stopped as soon as these modes have converged with sufficient accuracy. Although the effect is hard to quantify exactly, the convergence rate bound in eq. (6.80) seems to indicate that for a block implementation the parameter γ becomes $\gamma = \sigma_m/\sigma_n$, where m is the highest desired mode and n the size of the problem. Therefore, when using buffer vectors one has $\gamma = \sigma_{m+b}/\sigma_n \geq \sigma_m/\sigma_n$. Numerical tests seem to confirm the increased convergence rate when using buffer vectors. For choosing b, the rule of thumb used for the subspace iteration method may be adopted, namely $b = \min\{2m, m+8\}$.
- As pointed out before, the number of inner iterations r (i.e. the order of the Krylov subspace) can be freely chosen. Indeed, the bound for the convergence rate in eq. (6.80) shows that the convergence speeds up rapidly with increasing r. However, increasing r means increasing the computational effort in the inner iteration loop, so it is expected that some compromise needs to be found for optimal overall computational efficiency. This was also observed in [79] and will be illustrated in chapter 8.
- When choosing $r = 1$ for the number of inner iterations, it was previously mentioned that the IFPKS method reduces to the steepest descent method. In the literature it has been observed that for a steepest descent method, the convergence rate can be notably increased when adding the difference vector of the previous and current eigenmode estimates ($\boldsymbol{D}_k = \boldsymbol{X}_k - \boldsymbol{X}_{k-1}$) to the search directions. This forms the basis of the so-called LOBPCG method [116]. Given the low additional computational cost, it was therefore advocated in [154] to include these vectors in the basis $\boldsymbol{Z}$. In that case the basis becomes $\boldsymbol{Z} = [\boldsymbol{D}_k \;\; \boldsymbol{X}_k \;\; \boldsymbol{Z}_1 \ldots \boldsymbol{Z}_r]$.

6.3.4 Application to Updating of Vibration Modes

Now that an eigensolver has been outlined that seems suitable for handling the vibration mode updating problem, this section treats a few specific issues that affect the actual calculation of the updated vibration modes. First, it is shown how preconditioning with the factorization of the nominal stiffness matrix improves the convergence of the IFPKS method. Thereafter, an idea is outlined to enhance the preconditioning based on the information generated during the updating of the static modes. Finally, it is discussed how modal sensitivity vectors might be used in the updating computations.

6.3.4.1 Preconditioning

An in depth, mathematical analysis of preconditioning strategies for the IFPKS algorithm is out of the scope of this work, such results can be found in [79, 153, 154]. Instead, the effect of preconditioning will be illustrated from an engineering perspective in a qualitative way. The starting point is the application of the shifted operator when building the Krylov subspace in the IFPKS inner iteration loop:

$$\boldsymbol{z}_{j+1} = (\boldsymbol{K} - \rho_k \boldsymbol{M})\, \boldsymbol{z}_j \tag{6.89}$$

For simplicity the single vector version of the algorithm is considered and the preconditioner has not yet been applied. Suppose one is searching for $\boldsymbol{\phi}_1$, the first eigensolution of $(\boldsymbol{K}, \boldsymbol{M})$. To this end, the current search direction $\boldsymbol{z}_j$ is written as the following superposition:

$$\boldsymbol{z}_j = \boldsymbol{\phi}_1 + \sum_{i=2}^{n} \boldsymbol{\phi}_i \eta_i \tag{6.90}$$

In order to converge to the first mode, compute subsequent search directions should be computed that account for the higher modes. By doing so, vectors are added to the basis $\boldsymbol{Z}$ such that the desired $\boldsymbol{\phi}_1$ can be found as a combination of the vectors in the basis when solving the interaction problem.

Using the expansion in (6.90) and the spectral expansions of the stiffness and mass matrices, the application of the shifted operator in eq. (6.89) can be written as:

$$\boldsymbol{z}_{j+1} = \left(\sum_{i=1}^{n} (\omega_i^2 - \rho_k)\, \boldsymbol{M} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^T \boldsymbol{M} \right) \left(\boldsymbol{\phi}_1 + \sum_{i=2}^{n} \boldsymbol{\phi}_i \eta_i \right), \tag{6.91}$$

Applying mode orthogonality properties this can be rewritten to:

$$\boldsymbol{z}_{j+1} = (\omega_1^2 - \rho_k)\, \boldsymbol{M} \boldsymbol{\phi}_1 + \sum_{n=2}^{n} (\omega_i^2 - \rho_k)\, \boldsymbol{M} \boldsymbol{\phi}_i \eta_i \tag{6.92}$$

If it is now assumed that the first mode has nearly converged, so $\rho_k \approx \omega_1^2$, the first term becomes approximately zero. The new search direction therefore becomes:

$$\boldsymbol{z}_{j+1} \approx \sum_{n=2}^{n} (\omega_i^2 - \rho_k)\, \boldsymbol{M} \boldsymbol{\phi}_i \eta_i \tag{6.93}$$

This $\boldsymbol{z}_{j+1}$ is however not directly the desired search direction; its usefulness in the approximation basis $\boldsymbol{Z}$ depends on the conditioning (i.e. spectral distribution) of the shifted mass matrix. This is in line with the remark in section 6.3.3.2. Therefore, similar to the CG algorithm, preconditioning can be applied to generate more useful search directions. One very good preconditioner would be the factorized stiffness matrix, as will be shown next:

$$\boldsymbol{z}_{j+1} \approx \boldsymbol{K}^{-1} \sum_{n=2}^{n} (\omega_i^2 - \rho_k)\, \boldsymbol{M} \boldsymbol{\phi}_i \eta_i = \sum_{i=2}^{n} \frac{(\omega_i^2 - \rho_k)\, \boldsymbol{\phi}_i \eta_i}{\omega_i^2} \tag{6.94}$$

Furthermore, since $\rho_k / \omega_i^2 \leq 1$ $(i = 2 \ldots n)$ one can neglect ρ_k, which is reasonable especially for the higher spectrum where it holds that $\rho_k \ll \omega_i^2$. As a result, the search direction becomes approximately:

$$\boldsymbol{z}_{j+1} \approx \sum_{n=2}^{n} \boldsymbol{\phi}_i \eta_i \tag{6.95}$$

Comparing to eq. (6.90), this direction is a very useful enrichment of the basis $\boldsymbol{Z}$. It can therefore be concluded that for the lower modes $\boldsymbol{K}^{-1}$ is an effective preconditioner. However, an important feature of the basis updating strategy proposed here is that it avoids the factorization of the modified stiffness matrix $\boldsymbol{K}_{(1)}$. Therefore, the factorization of the nominal stiffness matrix $\boldsymbol{K}_{(0)}$ is again used as an approximation. As is the case for the updating of the static modes, the quality of this preconditioner is dependent on the magnitude of the design modification.

6.3.4.2 Reusing the Conjugate Gradient Iterates

When the updating problem for vibration modes is solved after the updating problem for the static modes, a possibility is to reuse the iterates generated by the CG algorithm in the IFPKS method. Indeed, the CG iterates implicitly contain information on the inverse of the stiffness matrix of the modified system. This is valuable information that can be employed to enhance the preconditioning step in the IFPKS method, as explained below.

Suppose one has solved the updating problem for the static modes using the CG method and in the process generated the iterates $\boldsymbol{P}$ and $\boldsymbol{W} = \boldsymbol{K}_{(1)}\boldsymbol{P}$, such that $\boldsymbol{P}^T\boldsymbol{K}_{(1)}\boldsymbol{P} = \boldsymbol{P}^T\boldsymbol{W} = \boldsymbol{I}$. From the previous subsection it was seen that ideally, the preconditioning step in the IFPKS algorithm has the form:

$$\boldsymbol{K}_{(1)}\boldsymbol{Z}_{j+1} = \left(\boldsymbol{K}_{(1)}\boldsymbol{Z}_j - \boldsymbol{M}_{(1)}\boldsymbol{Z}_j\boldsymbol{\rho}_k\right) = \boldsymbol{R}_j \qquad (6.96)$$

The solution can now be split into a part that lives in the space of the CG iterates $\boldsymbol{P}$ and a part in the $\boldsymbol{K}_{(1)}$-orthogonal space:

$$\boldsymbol{Z}_{j+1} = \boldsymbol{P}\boldsymbol{\alpha} + \boldsymbol{T}\boldsymbol{\beta}, \qquad (6.97)$$

where the orthogonal projector $\boldsymbol{T}$ is defined as:

$$\boldsymbol{T} = \boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^T\boldsymbol{K}_{(1)} = \boldsymbol{I} - \boldsymbol{P}\boldsymbol{W}^T \qquad (6.98)$$

Next, insert this solution in the preconditioning problem to find:

$$\boldsymbol{K}_{(1)}\left(\boldsymbol{P}\boldsymbol{\alpha} + \boldsymbol{T}\boldsymbol{\beta}\right) = \boldsymbol{R}_j \qquad (6.99)$$

The unknown amplitudes $\boldsymbol{\alpha}$ can be found by projection of this equation onto the space $\boldsymbol{P}$:

$$\boldsymbol{P}^T\boldsymbol{K}_{(1)}\left(\boldsymbol{P}\boldsymbol{\alpha} + \boldsymbol{T}\boldsymbol{\beta}\right) = \boldsymbol{P}^T\boldsymbol{R}_j \quad \rightarrow \quad \boldsymbol{\alpha} = \boldsymbol{P}^T\boldsymbol{R}_j \qquad (6.100)$$

Similarly, the amplitudes $\boldsymbol{\beta}$ are found through projection onto the space $\boldsymbol{T}$:

$$\boldsymbol{T}^T\boldsymbol{K}_{(1)}\left(\boldsymbol{P}\boldsymbol{\alpha} + \boldsymbol{T}\boldsymbol{\beta}\right) = \boldsymbol{T}^T\boldsymbol{R}_j \quad \rightarrow \quad \boldsymbol{\beta} = \boldsymbol{K}_{(1)}^{-1}\boldsymbol{R}_j \qquad (6.101)$$

However, since the factorization of the current stiffness matrix is not available an approximation is used, namely the factorization of the nominal stiffness matrix $\boldsymbol{K}_{(0)}^{-1}$. Hence the preconditioning step in the IFPKS method, when reusing the CG iterates, becomes:

$$\boldsymbol{Z}_{j+1} = \boldsymbol{P}\boldsymbol{P}^T\boldsymbol{R}_j + \left(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{W}^T\right)\boldsymbol{K}^{(0)^{-1}}\boldsymbol{R}_j \qquad (6.102)$$

The additional cost of this extended preconditioning scheme is low, since it only involves (block) vector multiplications. In chapter 8 the effectiveness of this preconditioning approach will be tested.

Note that in case the structure is subjected to a substantial design change, it could be that the basis $\boldsymbol{P}$ of the CG iterates is relatively large. Furthermore, since the basis is obtained from solving the static response to a local interface force, it might contain local information that is not very useful for predicting the global vibration modes. One way to select only the most relevant content of the basis $\boldsymbol{P}$ is by solving an interaction problem in that space and retaining only a limited number of the lowest modes for preconditioning:

$$\boldsymbol{P}^T \left(\boldsymbol{K}_{(1)} - \omega^2 \boldsymbol{M}_{(1)}\right) \boldsymbol{P}\boldsymbol{y} = \boldsymbol{0} \tag{6.103}$$

This can be written in the following form:

$$\left(\boldsymbol{I} - \omega^2 \boldsymbol{P}^T \boldsymbol{M}_{(1)} \boldsymbol{P}\right) \boldsymbol{y} = \boldsymbol{0}, \tag{6.104}$$

hence the reduced eigenproblem can be solved efficiently. Choosing the lowest m modes $\boldsymbol{Y} = \left[\boldsymbol{y}_1 \dots \boldsymbol{y}_m\right]$, the most important content of the CG iterates can be recovered by:

$$\bar{\boldsymbol{P}} = \boldsymbol{P}\boldsymbol{Y} \tag{6.105}$$

If these modes are scaled such that $\bar{\boldsymbol{P}}^T \boldsymbol{K}_{(1)} \boldsymbol{P} = \boldsymbol{I}$, this compacted basis can directly be used in eq. (6.102).

6.3.4.3 Modal Sensitivity Vectors

Based on the eigensolutions of the nominal component and the design modification, one can easily and fairly cheaply calculate so called modal sensitivity vectors. These modal sensitivities contain information on the extent and direction in which the eigensolutions have changed due to the design modification. Hence, this information can be used to enhance the initial guess for the IFPKS method, thereby hopefully accelerating its convergence.

When starting the updating calculations using the IFPKS method, one wants to solve the eigenproblem for the modified system for some mode j:

$$\left(\boldsymbol{K}_{(1)} - \omega^2_{(1),j} \boldsymbol{M}_{(1)}\right) \boldsymbol{\phi}_{(1),j} = \left(\left(\boldsymbol{K}_{(0)} + \Delta\boldsymbol{K}\right) - \omega^2_{(1),j} \left(\boldsymbol{M}_{(0)} + \Delta\boldsymbol{M}\right)\right) \boldsymbol{\phi}_{(1),j} = \boldsymbol{0} \tag{6.106}$$

This can be written as:

$$\left(\boldsymbol{K}_{(0)} - \omega^2_{(1),j} \boldsymbol{M}_{(0)}\right) \boldsymbol{\phi}_{(1),j} = -\left(\Delta\boldsymbol{K} - \omega^2_{(1),j} \Delta\boldsymbol{M}\right) \boldsymbol{\phi}_{(1),j} \tag{6.107}$$

Here the right hand side can be understood as the residual forces introduced in the nominal eigenproblem due to the design modification. However, eq. (6.107) cannot be solved directly for the new mode since the associated frequency is still unknown and factorization of the shifted matrix should be avoided. Instead, so-called modal correction or sensitivity vectors can be computed. Since the lowest eigenmodes are sought, it is therefore first assumed that the elastic forces are dominating over the inertia forces. This gives:

$$\boldsymbol{K}_{(0)} \bar{\boldsymbol{\phi}}_{(1),j} = -\left(\Delta\boldsymbol{K} + \omega^2_{(1),j} \Delta\boldsymbol{M}\right) \boldsymbol{\phi}_{(1),j} \tag{6.108}$$

However, the eigenmodes and frequencies of the modified system are unknown. Still these can be expressed as the known nominal solution plus some unknown perturbation:

$$\boldsymbol{\phi}_{(1),j} = \boldsymbol{\phi}_{(0),j} + \Delta\boldsymbol{\phi}_j \quad \text{and} \quad \omega^2_{(1),j} = \omega^2_{(0),j} + \Delta\omega^2_j. \tag{6.109}$$

Inserting in (6.108) and neglecting higher order terms gives a first order modal sensitivity vector as:

$$\bar{\boldsymbol{\phi}}_{(1),j} = -\boldsymbol{K}^{-1}_{(0)}\left(\Delta\boldsymbol{K} - \omega^2_{(0),j}\Delta\boldsymbol{M}\right)\boldsymbol{\phi}_{(0),j} \tag{6.110}$$

Such a sensitivity vector is computed for all $j = 1 \dots m$ vibration modes. Since the factorization of $\boldsymbol{K}_{(0)}$ is already computed and the design changes may be only local, this is not very expensive to compute. These "modal sensitivities" or corrections are the same as used directly in the reduction basis in the Enriched Craig-Bampton method [133]. Physically, these modal sensitivities can be interpreted as the static deformation shapes due to the force residual resulting from applying the nominal eigensolutions to the modified structure.

These modal sensitivity vectors can be employed to improve the initial guess used as input to the IFPKS method. This is done by taking:

$$\boldsymbol{X} = \left[\boldsymbol{\Phi}_{(0)} \quad \bar{\boldsymbol{\Phi}}_{(1)}\right], \tag{6.111}$$

and subsequently solving the interaction problem on $\left(\boldsymbol{K}_{(1)}, \boldsymbol{M}_{(1)}\right)$ in this space. The resulting modes are expected to provide an improved initial guess $\boldsymbol{X}_0$ over simply taking $\boldsymbol{X}_0 = \boldsymbol{\Phi}_{(0)}$, which will be examined in chapter 8.

6.4 Sequential Updating

Up to now only a single design modification and associated basis update has been considered. In practice however, a component often undergoes a series of design modifications from the nominal to the final design. In that case, not only the factorization of the nominal component's stiffness matrix might be useful for preconditioning, but also the CG iterates obtained from updating computations of previous modifications. This is very similar to what has been treated previously in section 6.3.4.2 for reusing the CG iterates in the IFPKS method.

Suppose that one has updated the reduction basis of the modified component 1 and that the iterates $\boldsymbol{P}_{(1)}$ and $\boldsymbol{W}_{(1)} = \boldsymbol{K}_{(1)}\boldsymbol{P}_{(1)}$ have been generated, such that $\boldsymbol{P}^T_{(1)}\boldsymbol{K}_{(1)}\boldsymbol{P}_{(1)} = \boldsymbol{P}^T_{(1)}\boldsymbol{W}_{(1)} = \boldsymbol{I}$. Hence, a good approximation to an arbitrary static problem

$$\boldsymbol{K}_{(1)}\boldsymbol{x} = \boldsymbol{f}, \tag{6.112}$$

may be found as:

$$\boldsymbol{x} \approx \boldsymbol{K}^{-1}_{(0)}\boldsymbol{f} + \boldsymbol{P}_{(1)}\boldsymbol{\alpha} \tag{6.113}$$

Here the amplitudes $\boldsymbol{\alpha}$ are chosen such that the residual resulting from the approximation is orthogonal to the space spanned by the CG iterates:

$$\boldsymbol{P}^T_{(1)}\left(\boldsymbol{f} - \boldsymbol{K}_{(1)}\boldsymbol{x}\right) = \boldsymbol{0} \tag{6.114}$$

Substitution of the approximation (6.113) and solving for $\boldsymbol{\alpha}$ then gives:

$$\boldsymbol{\alpha} = \boldsymbol{P}_{(1)}^{T}\left(\boldsymbol{I} - \boldsymbol{K}_{(1)}\boldsymbol{K}_{(0)}^{-1}\right)\boldsymbol{f} \tag{6.115}$$

Replacing this in (6.113) gives

$$\boldsymbol{x} \approx \left(\boldsymbol{K}_{(0)} + \boldsymbol{P}_{(1)}\boldsymbol{P}_{(1)}^{T}\left(\boldsymbol{I} - \boldsymbol{K}_{(1)}\boldsymbol{K}_{(0)}^{-1}\right)\right)\boldsymbol{f}, \tag{6.116}$$

which shows that an approximation to the inverse modified stiffness matrix may be found as:

$$\boldsymbol{K}_{(1)}^{-1} \approx \tilde{\boldsymbol{K}}_{(1)}^{-1} = \boldsymbol{P}_{(1)}\boldsymbol{P}_{(1)}^{T} + \left(\boldsymbol{I} - \boldsymbol{P}_{(1)}\boldsymbol{W}_{(1)}^{T}\right)\boldsymbol{K}_{(0)}^{-1} \tag{6.117}$$

This approximation forms a good preconditioner that can be used in subsequent calculations, as was proposed previously to extend the preconditioning of the IFPKS method. Indeed, this result is equal to that of eq. (6.102), namely a preconditioner that searches an approximation in the space of the CG iterates $\boldsymbol{P}_{(1)}$ and in the deflated space of the inverse nominal stiffness matrix, i.e. the space of $\boldsymbol{K}_{(0)}^{-1}$ that is $\boldsymbol{K}_{(1)}$-orthogonal to these iterates.

Imagine now that after the first design change and basis updating, a second modification is made to the structure and the reduction basis needs to be updated again for a stiffness matrix $\boldsymbol{K}_{(2)}$. One could use the nominal factorization as preconditioner as was done to find the static modes of $\boldsymbol{K}_{(1)}$. However, assuming that the design process is converging, usually $\boldsymbol{K}_{(2)}$ is closer to $\boldsymbol{K}_{(1)}$ than to $\boldsymbol{K}_{(0)}$. Hence it is probably more efficient to use the best knowledge one has of $\boldsymbol{K}_{(1)}$ to precondition $\boldsymbol{K}_{(2)}$. As a result, the following sequential updating procedure can be envisioned:

- In the updating to find the static modes of $\boldsymbol{K}_{(2)}$ using the conjugate gradient method, use the preconditioner $\tilde{\boldsymbol{K}}_{(1)}^{-1}$ in approximation (6.117). This updating process generates the iteration vectors $\boldsymbol{P}_{(2)}$ and $\boldsymbol{W}_{(2)}$.
- In the updating of the vibration modes using the IFPKS method on $\boldsymbol{K}_{(2)}, \boldsymbol{M}_{(2)}$, use the following preconditioner:

$$\begin{aligned}\boldsymbol{K}_{(2)}^{-1} \approx \tilde{\boldsymbol{K}}_{(2)}^{-1} &= \boldsymbol{P}_{(2)}\boldsymbol{P}_{(2)}^{T} + \left(\boldsymbol{I} - \boldsymbol{P}_{(2)}\boldsymbol{W}_{(2)}^{T}\right)\boldsymbol{K}_{(1)}^{-1}\\ &\approx \boldsymbol{P}_{(2)}\boldsymbol{P}_{(2)}^{T} + \left(\boldsymbol{I} - \boldsymbol{P}_{(2)}\boldsymbol{W}_{(2)}^{T}\right)\tilde{\boldsymbol{K}}_{(1)}^{-1}\\ &= \boldsymbol{P}_{(2)}\boldsymbol{P}_{(2)}^{T} + \left(\boldsymbol{I} - \boldsymbol{P}_{(2)}\boldsymbol{W}_{(2)}^{T}\right)\left(\boldsymbol{P}_{(1)}\boldsymbol{P}_{(1)}^{T} + \left(\boldsymbol{I} - \boldsymbol{P}_{(1)}\boldsymbol{W}_{(1)}^{T}\right)\boldsymbol{K}_{(0)}^{-1}\right)\end{aligned} \tag{6.118}$$

This process can be continued for every sequential update. Obviously as the number of updates increases the total number of vectors in $\boldsymbol{P}_{(i)}$ increases, so does the computational cost of these "deflations". An important issue then becomes selecting from all the available information the most relevant content for approximating the inverse of the current stiffness matrix. This issue is often encountered in the field of iterative solvers and has been addressed by several authors in the literature, see for instance [174, 160, 161, 83, 147]. One straightforward approach proposed here is to simply monitor the linear dependency between the search directions obtained from the different updates. For instance, one could compute first

$$\delta_{ik} = \boldsymbol{P}_{(1),i}^{T}\boldsymbol{W}_{(2),k} \tag{6.119}$$

and remove $\boldsymbol{P}_{(1),i}$ if $\delta_{ik} < \varepsilon$, where ε is some selection tolerance. Obviously more elaborate and efficient selection strategies exist, but these are beyond the scope of this work.

6.5 Summary

Application of component model reduction techniques can be troublesome in practical design settings due to the need to recompute the reduction basis when the model is subject to design modifications. In this chapter a method was therefore presented for updating the reduction bases of such modified components. The starting point for this method was the reduction basis of a nominal component model, from which a modified component model is obtained after making a small design change. In contrast to existing enrichment or approximation methods, this chapter proposed an updating strategy that obtains the genuine reduction basis for the modified component.

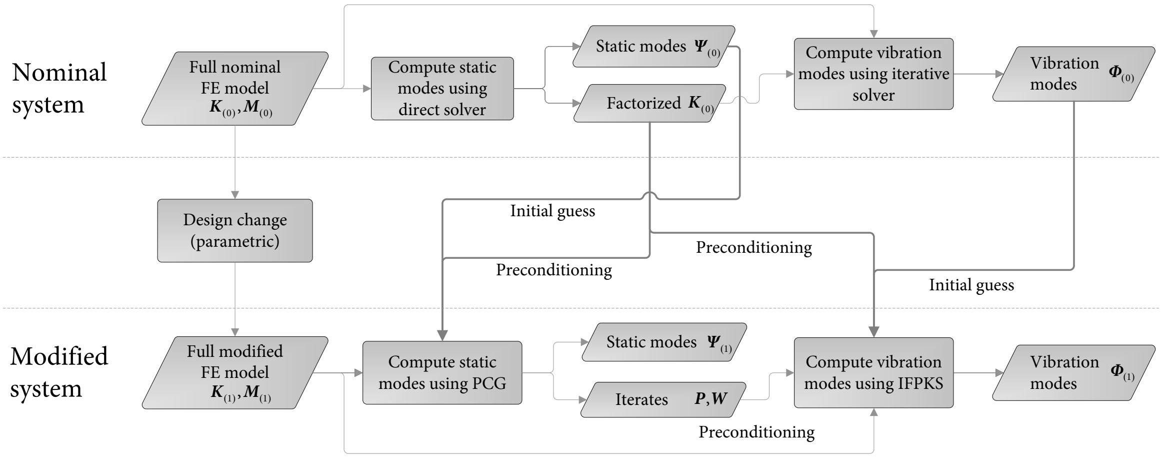

Nominal system
Full nominal FE model $K_{(0)}, M_{(0)}$
Compute static modes using direct solver
Static modes $\Psi_{(0)}$
Factorized $K_{(0)}$
Compute vibration modes using iterative solver
Vibration modes $\Phi_{(0)}$
Design change (parametric)
Initial guess
Preconditioning
Preconditioning
Initial guess
Modified system
Full modified FE model $K_{(1)}, M_{(1)}$
Compute static modes using PCG
Static modes $\Psi_{(1)}$
Iterates P, W
Compute vibration modes using IFPKS
Vibration modes $\Phi_{(1)}$
Preconditioning

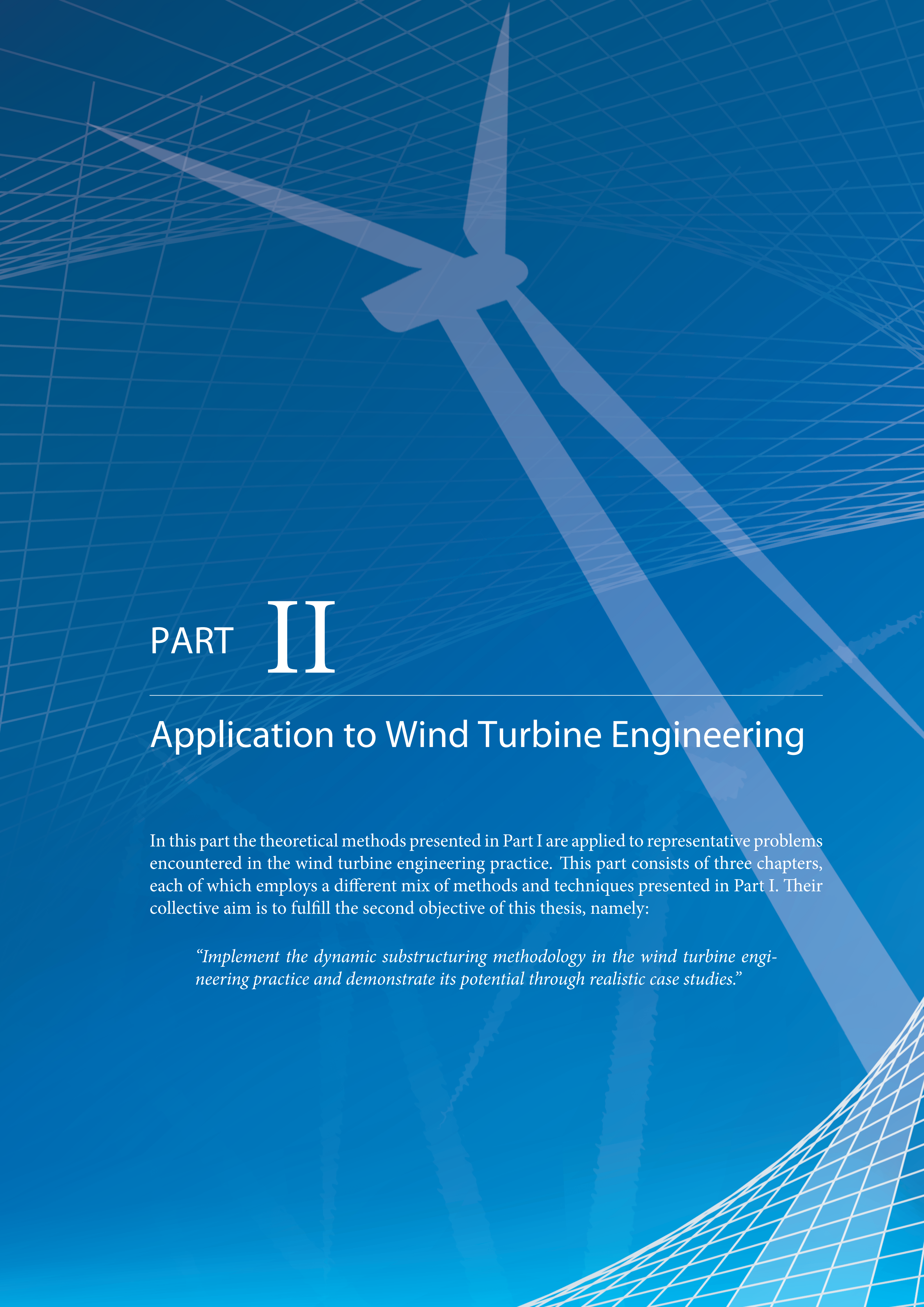
Figure 6.2: Flowchart for updating of component reduction bases.

It was established that factorization of the stiffness matrix accounts for the largest contribution to the computational cost of obtaining a reduction basis. Hence, the aim was to avoid the factorization of the modified stiffness matrix and reuse as much information as possible from the nominal component. Following the structure of a typical component reduction basis, a strategy was developed that sequentially updates the static and vibration modes. Based on the observation that the dynamics of the modified component are often close to those of the nominal structure, iterative Krylov subspace methods were proposed to achieve this, namely:

- For updating the static modes the well known conjugate gradient (CG) solver was proposed in section 6.2. The algorithm was started from the nominal static modes and preconditioned using the factorized nominal stiffness matrix. In order to handle the possibly many static modes associated to the interface DoF of a component model, the CG algorithm was cast into a variable size block form combined with a projection and reconjugation approach. Furthermore, the possibility was raised of updating the static modes with a low tolerance. A correction was introduced to make sure that in that case possible rigid body modes are still properly described.

- In section 6.3 it was proposed to update the vibration modes using the recent inverse-free preconditioned Krylov subspace (IFPKS) method. Again, the iterations were started from the nominal vibration modes and the inverse nominal stiffness matrix was applied for preconditioning. Furthermore, it was shown how the preconditioning can be enhanced using the CG iterates and the initial guess can be improved using computationally cheap modal sensitivity vectors.

Section 6.4 finally briefly addressed the issue of multiple sequential design modifications. In that case the preconditioning can be extended using the CG iterates from previous updates. Assuming that the modified model is closer to the previous model than to the nominal one, this should improve the convergence rates of both the CG and IFPKS methods. To limit the additional computational costs of this extended preconditioning, selection of the most relevant iterates can be performed. The updating scheme is summarized in the flowchart in figure 6.2.



PART II

Application to Wind Turbine Engineering

In this part the theoretical methods presented in Part I are applied to representative problems encountered in the wind turbine engineering practice. This part consists of three chapters, each of which employs a different mix of methods and techniques presented in Part I. Their collective aim is to fulfill the second objective of this thesis, namely:

> *"Implement the dynamic substructuring methodology in the wind turbine engineering practice and demonstrate its potential through realistic case studies."*

7

Dynamic Substructuring Analysis of a Yaw System

> *I can't change the direction of the wind,*
> *but I can adjust my sails to always reach my destination.*
>
> (Jimmy Dean)

7.1 Introduction

Ever since wind turbines were first applied commercially in the 1970's, a variety of different types of machines has been developed. One common classification is that of horizontal-axis versus vertical-axis wind turbines. As the name suggests, vertical-axis wind turbines (VAWTs) have the main rotor shaft oriented vertically. In general, VAWTs have the gearbox and generator near the ground, which eliminates the need for a tower. The key advantage of the vertical axis arrangement is that the turbine's orientation is independent of the wind direction. However, due to reliability issues caused by high cyclic loading inherent to the design, VAWTs never turned into a commercial success.

Horizontal-axis wind turbines (HAWTs) have the rotor main shaft arranged horizontally, with a tower-supported rotor pointed into the wind. Within the HAWT class, different configurations can be identified such as one, two or three bladed rotors, turbines using a gearbox to drive a high speed generator and those that use a low speed generator to which the rotor is directly mounted (so-called direct drive turbines). A further distinction is whether the rotor is place upwind, i.e. in front of the tower, or downwind, where the rotor positions itself behind the tower. Nowadays, all commercial multi-megawatt wind turbines are HAWTs according to the "Danish design", which is a three bladed, upwind turbine supported by a tubular tower.

An important aspect in the operation of such wind turbines is the alignment of the rotor plane with respect to the wind direction. This is important in order to maximize both the energy capture as well as the turbine's fatigue lifetime. Indeed, when the rotor is not perpendicular to the wind a so-called *yaw misalignment* (yaw denotes the rotation about the vertical axis) gives rise to cyclic varying asymmetric loads, which increase the fatigue loads on many components. Hence a *yaw system* is needed to ensure optimal yaw orientation during operation, either in a passive or active manner.

For upwind turbines passive yawing can be achieved by using a tail vane and a cone-shaped rotor. However, passive yawing can generate high yaw rates, leading to excessive gyroscopic moments on the wind turbine tower. Twisting of the cable that runs from the generator in the nacelle to the transformer in the tower base is also an issue. Hence, all modern wind turbines are equipped with an active yaw system which, based on data from a wind sensor, continuously keeps the rotor orthogonal to the wind direction. The yaw system therefore is an important part of every modern wind turbine.

In this chapter a structural dynamic model of the yaw system of a Siemens 2.3 megawatt wind turbine will be created using dynamic substructuring, more specifically through application of the theory and methods presented in chapters 2, 3 and 4. The motivation to do so is twofold. Firstly, the yaw system comprises many components and complex interfaces, making it an interesting case study for illustrating the potential of the dynamic substructuring methodology. Secondly, the yaw system is generally not taken into account in a detailed way in aero-elastic codes, but is in some cases thought to influence the overall turbine dynamics. The model created in this chapter could therefore be used to gain more insight in the yaw system dynamics.

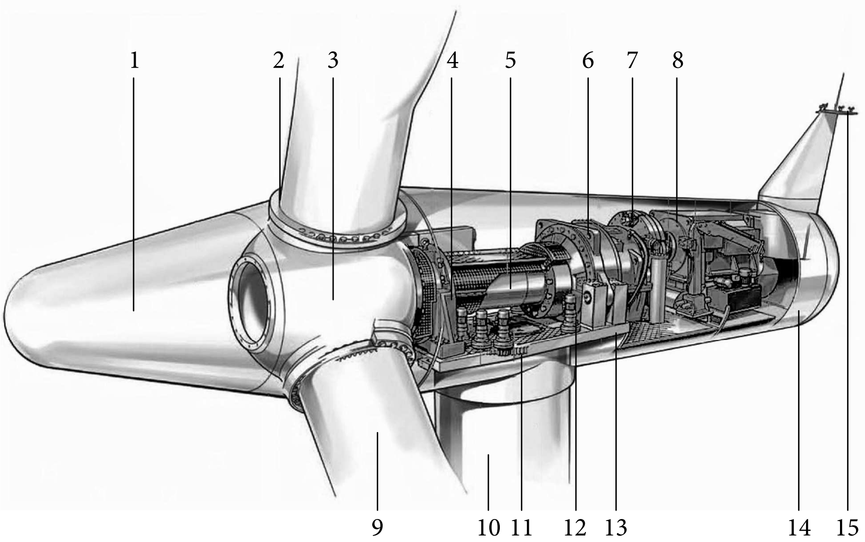
The remainder of this chapter is based on the work reported in [206, 218] and is organized as follows. Next, section 7.2 gives an overview of the Siemens 2.3 megawatt wind turbine in general and its yaw system in specific, treating the different yaw system components and their interfaces. Thereafter, sections 7.3 to 7.6 describe the modeling of these components and interfaces. Analysis results of the assembled system are presented in section 7.7. As usual the chapter is ended with a summary in section 7.8.

7.2 System Description

7.2.1 The 2.3 Megawatt Siemens Wind Turbine

The Siemens 2.3 megawatt (MW) wind turbine is according to the Danish design and equipped with a variable speed generator. Whereas in the early days of turbine technology the rotor speed was kept constant to generate an alternating current at a certain frequency (usually 50 or 60 Hz), nowadays the rotor speed can vary with the wind speed. For a wide range of wind speeds, such variable speed wind turbines can be operated at the optimal energy capture while minimizing the load on the machine. As a result of the variable rotor speed, the frequency of the generated electricity is not constant and has to be converted to the right frequency. This is done with a converter in the tower base.

The turbine is available in four variants with different rotor diameters (i.e. 82, 93, 101 and 108 meters). In order to generate the same energy output, a site with relatively low average wind speeds requires a larger rotor compared to a site with higher average wind speeds. Hence, by offering three rotor diameters a site optimized choice of turbine type can be made. The most common variant of this product family is the SWT-2.3-93; its 45 meter blades and 3 meter diameter hub create a 93 meter rotor diameter suited for moderate average wind speeds.



1	Spinner	6	Gearbox	11	Yaw ring
2	Pitch bearing	7	Brake disc	12	Yaw gearbox
3	Rotor hub	8	Generator	13	Bedplate
4	Main bearing	9	Blade	14	Canopy
5	Main shaft	10	Tower	15	Wind sensors

Figure 7.1: Nacelle arrangement of the 2.3 MW Siemens wind turbine
(Source: http://www.siemens.com/windpower)

An onshore wind turbine consists of three main parts, namely the tower, rotor and nacelle. The rotor is the assembly of the blades, pitch bearings and hub, while the tower supports the rotor-nacelle-assembly (RNA). The nacelle is the heart of the wind turbine and houses the components that convert the mechanical energy captured by the rotor into electrical energy; see figure 7.1. Within the nacelle, two main (mechanical) subsystems can be identified:

- The drivetrain is the assembly of all mechanical components directly involved in transferring the energy captured by the blades to the generator, which transforms this energy into electrical energy. Important components within the drivetrain are the main bearing, the main shaft, the gearbox and the generator.
- The yaw system consists of all components of the wind turbine which enable the rotation of the nacelle (and rotor) about the tower axis. Since this system is analyzed in this chapter, it is discussed in more detail in the next subsection.

7.2.2 Yaw System & System Boundaries

As already explained, yawing denotes the rotation of the nacelle and the rotor about the vertical tower axis. The yaw system of the SWT-2.3-93 is schematically depicted in figure 7.2. In the yaw system of this wind turbine one can identify a number of components:

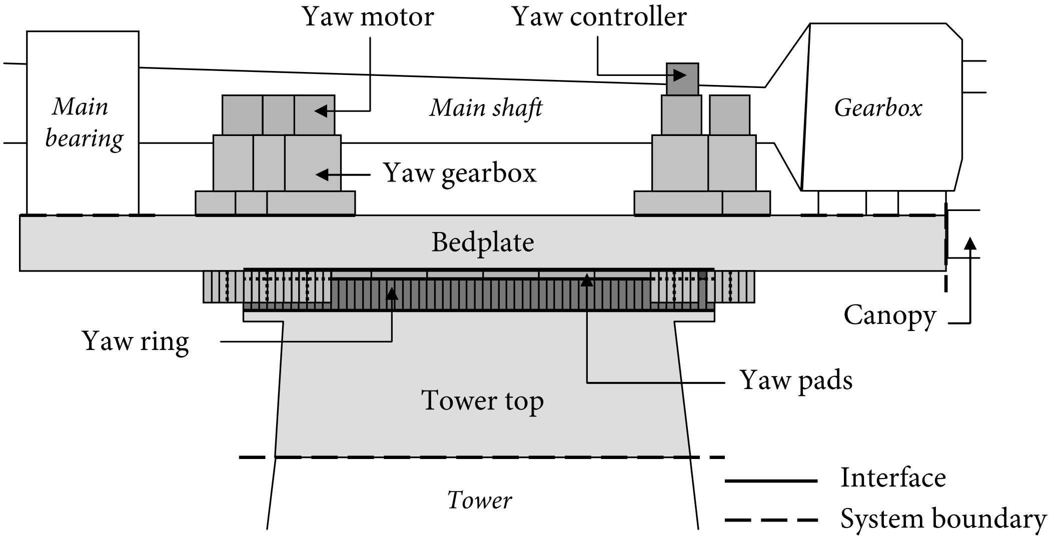


Figure 7.2: Yaw system of a 2.3 MW Siemens wind turbine

- The *bedplate* can be seen as the nacelle's "chassis". It serves as a platform for mounting the main turbine components, such as the gearbox, main bearing, canopy and several smaller parts. Furthermore, the bedplate houses the interface between the tower and RNA.
- The *tower top* is the upper section of the tapered tower. The tower top flange is welded on top of the tower and bolted to the yaw ring.
- The *yaw ring* is a big sprocket wheel driven by the yaw drives, enabling rotation of the entire RNA. The yaw ring is bolted to the tower top and journaled by the yaw pads.
- The *yaw pads* are attached to the bedplate and serve as a friction-type bearing for the yaw ring. These pads are made of polyamide material and are lubricated in operation.
- The *yaw motors* are electric motors controlled by the yaw controller. Through the *yaw gearboxes* their rotational speed is decreased and their torque increased. This is needed to overcome the RNA's inertia and static friction at the yaw pads so that the RNA can be rotated, while simultaneously keeping yaw velocities low. Combined the yaw gearbox and motors are referred to as the yaw drive.
- The *yaw controller* is a central controller for the yaw system and is instructed by the global turbine controller. This controller regulates the torque of the yaw motors.

In figure 7.2 all yaw system components are displayed in grey, while the wind turbine components shown in white are outside the system of interest. In between both the following system boundaries can be identified:

- The tower and tower top
- The bedplate and main bearing
- The bedplate and gearbox
- The bedplate and canopy

7.2.3 Component Modeling & Validation

Due to the fact that extensive stress analyses are performed on the structural components of a wind turbine, CAD and finite element models of most components are often readily available. Furthermore, most components are made from steel and are hence very well suited for FE modeling. Therefore, existing FE models can be used in a DS analysis with only some minor changes, which benefits the practical usability of the DS approach. In the coming sections the modeling of the different yaw system components, as well as their respective interfaces, will be treated in more detail.

As was outlined in chapter 1, in order to gain confidence in the component models it is important to validate them using measurements. Generally one can identify two types of modeling errors in a DS analysis, namely errors in the component models and errors in the interface models. To eliminate both errors an effective validation strategy is to first validate the component models, subsequently assemble the validated component models and finally perform a validation measurement on the assembled structure. One can then use the validation measurement to identify the errors resulting from the interface modeling.

In this work however the focus is on the modeling and numerical aspects of dynamic substructuring, so detailed model validation, both on component and assembly level, is out of the scope of this thesis. Nonetheless some component validation tests have been performed in the context of the work presented in this chapter, see appendix B for details. Validation of the assembled model will not be addressed.

7.3 Component Modeling – Bedplate

The bedplate is a central part of the nacelle and serves as a chassis for all main components of both the yaw system and the drivetrain. As such, it endures and transmits all the trust forces from the wind and is exposed to continuous variation in loading. Due to its geometry and material properties, FE modeling of the bedplate is relatively straightforward. A CAD model of the bedplate is used to create the FE model. This CAD model contains all geometrical features of the bedplate, including many small details (e.g. bolt holes). These geometric details cause (locally) very small elements when meshing the model, while not significantly influencing the global dynamic behavior. Due to limitations in the available computing resources, a number of details have been removed from the CAD model:

- All bolt holes have been removed from the structure.
- Shallow cavities exist in the bedplate for mounting the yaw pads. The depth of these cavities is small compared to the thickness of the bedplate, so they are removed from the model.
- Similar cavities exist for the yaw gearboxes, these are also removed from the CAD model.

Although the system boundaries were initially set as in figure 7.2, it was chosen to also include the main bearing housing and gearbox supports in the bedplate model. This leads to more convenient interfaces for the assembly of a drivetrain model in a later stadium.[1] Furthermore, the yaw clamps are integrated in the bedplate model; the yaw pads are mounted on the yaw clamps and thereby serve as a bearing between the bedplate and yaw ring. The bedplate is made from a high strength steel with the following homogeneous isotropic material properties:

	Density [kg/m^3]	Young's modulus [GPa]	Poisson ratio [−]
Bedplate	7850	210	0.30

The finite element model of the bedplate substructure is created in ANSYS and is depicted in figure 7.3. It is meshed with 25k quadratic (i.e. ten-node) tetrahedral elements with only translational nodal DoF resulting in a finite element model of approximately 125k DoF. Using the FEMLink toolbox (see [17]), the FE model is imported in Matlab where it is cast in the data format for the DS Toolbox and further processed using the PrepTool (see appendix A.1).

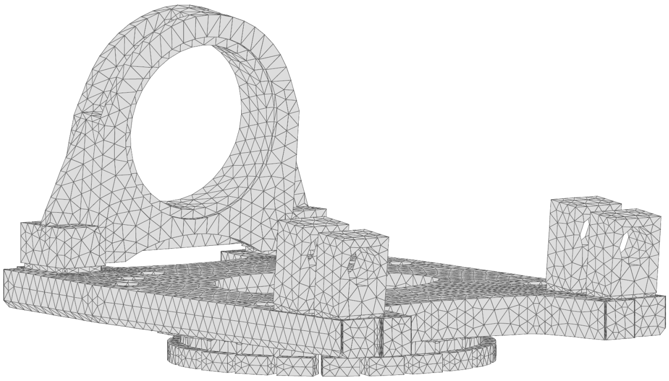


Figure 7.3: FE model of the bedplate with yaw clamps, main bearing housing and gearbox supports.

The following interfaces can be identified on the bedplate structure (see also figure 7.2):

Bedplate – Yaw gearboxes Eight yaw gearboxes are attached to the bedplate, each bolted to the bedplate by twelve high-strength bolts. It is assumed that these bolted connections ensure exact compatibility between the bedplate and yaw gearboxes, i.e. possible bolt flexibility is neglected. Furthermore, this interface is assumed to behave as a rigid region as explained in section 4.3; coupling is done through a single node with six DoF.

Yaw Pads – Bedplate Figure 7.4 shows the configuration of the yaw pads. To mount the set of upper yaw pads (22 pieces in total), special cavities are milled into the bottom side of the bedplate. Although these cavities are removed from the bedplate model, it is assumed that this design ensures perfect compatibility between the bedplate and upper yaw pads. The radial and lower yaw pads (both 22 pieces in total) are mounted in yaw clamps which are attached to the bedplate. It is assumed that this construction also leads to exact compatibility between the yaw clamps and pads.

---

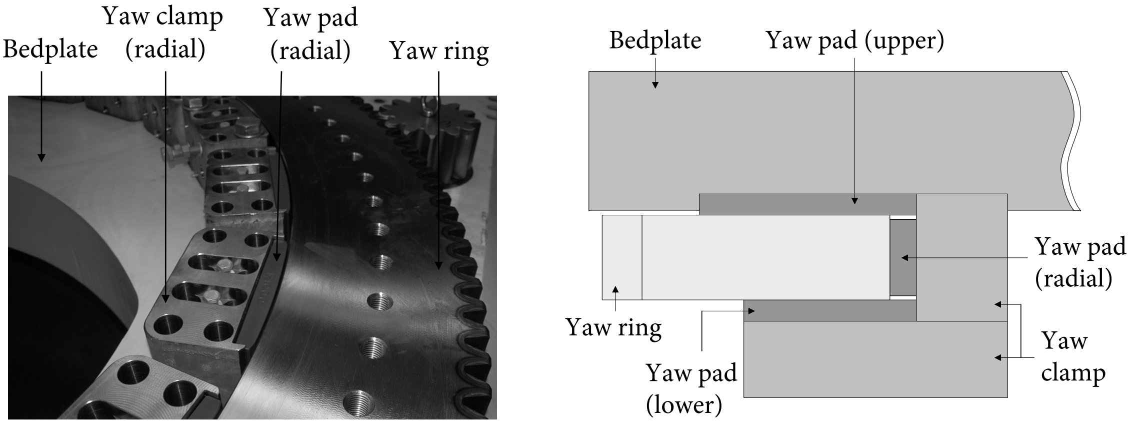
[1] Including these additions the bedplate is often referred to as the "bedframe"; for consistency this extended structure will still be referred to as the "bedplate" in the remainder of this chapter.

> Since the yaw pads cover a large part of the bedplate surface, rigidification of this interface would significantly stiffen the bedplate model. It is therefore chosen to retain the original set of interface DoF. Moreover, with the bedplate and yaw pad models created independently their meshes are incompatible, requiring the use of the node collocation techniques described in 3.6.

Note that in order to gain confidence in the bedplate model as well as the modeling of its interfaces, it has been successfully validated with measurements. The details of these experiments are out of the scope of this section but are treated in appendix B.1.

7.4 Component Modeling – Yaw Pads

In figure 7.4 the configuration of the yaw pads is shown. In total, the yaw ring is enclosed by three circular arrays of 22 yaw pads: one array at the top, one at the bottom (not present in figure 7.4 (a)) and one in the inner radius. Thereby the global motion of the bedplate is constrained to five degrees of freedom, so the yaw pads form a (lubricated) friction bearing that allows rotation of the RNA about the vertical tower axis. This chapter however considers the case where the interface is in the "stick" regime, that is, the (external) forces are not high enough to overcome the static friction.



(a) Bottomside of (part of) the yaw system (b) Schematic configuration of yaw pads

Figure 7.4: Configuration of the yaw pads.

A further simplification in the current analysis is made by including only the upper array of yaw pads. It is assumed that the other sets of yaw pads, which are significantly smaller in size, have a negligible effect on the global dynamic behavior. To facilitate assembly with both the bedplate and the yaw ring, the yaw pads are also meshed using ten-node tetrahedral elements. The (very simple) FE model of the yaw pad is again created in ANSYS and subsequently imported in Matlab using FEMLink. In Matlab the yaw pad model is copied 22 times to obtain an FE model of the upper yaw pad array. The yaw pads have interfaces at both the top and bottom surfaces:

Yaw pads – Bedplate This interface was described in section 7.3.

Yaw pads – Yaw ring As described above, it is assumed that the forces on the interface are not high enough to overcome the static friction between the yaw pads and yaw ring. Therefore exact compatibility between the interface DoF is imposed. Again the meshes on the interfaces between the yaw pads and the yaw ring are non-conforming, so the node collocation methods from section 3.6 are applied. For the same reason as the interface between the yaw pads and bedplate, no interface rigidification is applied but all interface DoF are retained.

The yaw pads are made from a polyamide material with a high wear resistance and a low (dynamic) friction coefficient. An important property of synthetic materials is their frequency dependent behavior, which, in the time domain, requires a non-linear model.[2] Since no detailed data of the polyamide was available, it was chosen to approximate its properties by the modulus of elasticity and Poisson ratio at room temperature:

	Density [kg/m^3]	Young's modulus [GPa]	Poisson ratio [−]
Yaw pads	1135	4.5	0.30

Clearly this approximation is very crude and only gives a limited idea of the substructure's behavior. However, the aim of the work reported in this chapter is to illustrate the potential of the substructuring approach for modeling the yaw system, so creating a non-linear yaw pad model was deemed out of the scope.

7.5 Component Modeling – Tower Top & Yaw Ring

The yaw ring is an important component in the yaw system as can be seen in figure 7.2. As already mentioned, the yaw ring is a large externally geared ring that is driven by the output pinions of the yaw gearboxes. It is bolted to the tower top flange and journaled to the bedplate by the yaw pads, thereby allowing the yaw gearboxes to generate a torque around the tower axis that results in the yawing motion of the RNA. To include the stiffening effect of the tower on the yaw ring, the final section of the tower, the tower top, is included in the model. It is assumed that the bolted connection between the tower top flange and yaw ring ensures exact compatibility, thereby allowing them to be combined to a single substructure model.

The yaw ring model is simplified by removing the gear teeth geometry and replacing it by an equivalent ring radius. Similar to the bedframe, this is done to avoid very small element sizes, and associated large number of DoF, that would result from meshing the detailed gear teeth geometry. The material properties of the substructure are given below.

	Density [kg/m^3]	Young's modulus [GPa]	Poisson ratio [−]
Yaw ring	7830	210	0.30
Tower top	7850	210	0.30

Using the geometries and mechanical properties of the tower top and yaw ring, they are meshed using ten-node tetrahedral solid elements. The tower top however is a cylinder with

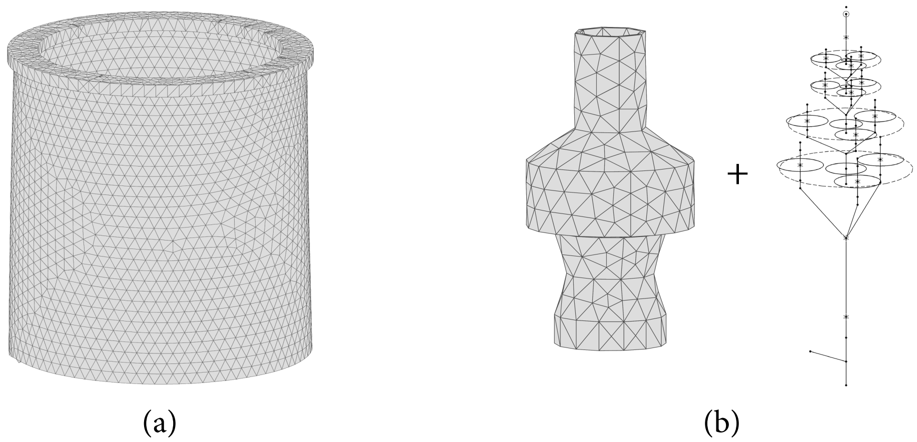
---

[2] Note that in the frequency domain this frequency dependent behavior can be directly captured by the FRFs that are used to describe the component dynamics. Hence, in frequency based substructuring (FBS) such behavior can be accounted for without additional complexity.

a small wall thickness, well suited for meshing with shell elements. Nonetheless it is chosen to use solid elements, for the reason that the number of DoF did not decrease sufficiently to justify the additional effort associated with coupling the solid and shell elements. The resulting FE model as created in ANSYS can be seen in figure 7.5 (a). From figure 7.2 it can be seen that this substructure interacts with neighboring components through two interfaces:

Yaw ring – Yaw pads This interface was described in section 7.4.

Yaw ring – Yaw gearboxes In order for the nacelle to yaw, torque is exerted on the yaw ring by eight yaw gearbox and motor assemblies. The yaw ring and gearboxes are connected through the gear teeth interaction between the yaw ring and the output pinions of the yaw gearboxes. The interface is modeled by an equivalent gear teeth stiffness based on the ISO6336 guideline [100] and modeling techniques found in the literature [119, 149], and will be described in more detail in section 7.6.2. Assembly of these two structures with the additional interface stiffness is performed using the method outlined in section 3.7.



(a) (b)

Figure 7.5: (a) Tower top and yaw ring model and (b) model of the yaw gearbox.

7.6 Component Modeling – Yaw Gearbox

One can imagine the enormous torsional moments associated with yawing a wind turbine RNA. To generate the required torque a speed reduction gearbox is essential, making the yaw gearbox a central part of the yaw system of a modern wind turbine. This yaw gearbox converts the high speed/low torque output of the electric yaw motors to low speed/high torque motion at the yaw ring. This avoids the need for large and expensive yaw motors and guarantees a low rotation speed of the RNA. The latter is important to minimize the gyroscopic loads of the yawing rotor with respect to the fixed tower. In total, eight yaw motors and yaw gears (when combined often called *yaw drive*) are employed in the 2.3 MW wind turbine. Although the yaw motors are relatively small electric motors, due to the transmission ratio of the yaw gearboxes they are able to generate more than 10^6 Nm of torque about the vertical tower axis.

The gearbox can be divided into two parts; the running gears (internal) and the gearbox housing (external). Both subcomponents are modeled separately and assembled to form the total gearbox model. Furthermore, the yaw motor is included in the yaw gearbox model. As such, the yaw gearboxes are involved in two interfaces:

Yaw gearbox – Bedplate This interface was described in section 7.3.

Yaw gearbox – Yaw ring This interface was described in section 7.5.

Yaw controller – Yaw motors The yaw controller regulates the yaw motors output torque. It is expected that during yawing action, the controller affects the global dynamic behavior. However, in this work the yaw system is analyzed during standstill conditions, so the controller is not included in the current analysis.

The yaw gearbox model is shown in figure 7.5 (b). How this model was created will be elaborated in the next three subsections.

7.6.1 Yaw Gears

The running gears in the yaw gearbox consist of four planetary gear stages, that result in a final transmission ratio of over 1000:1. Each stage has four planet gears to distribute the torque from the sun gear to the planet carrier. The ring wheel is attached to the housing and hence is stationary. In order to set up a discrete structural dynamic model of the yaw gears, the first step is to identify all relevant flexibilities and inertias in the gearbox, namely:

Internal components The individual torque-transferring components in the gearbox will deform under the applied loads. The mass, inertia and stiffness of the shafts, gears and planet carriers are important for the structural dynamics. The structural properties of the shafts and gears are modeled using Euler-Bernoulli beam elements with 6 DoF per node. The planet carriers will be modeled as rigid bodies. Given their size compared to the other shafts this seems a reasonable assumption, especially for the high speed stages where the torque is not too large. However, the inertia properties of the planet carriers must be taken into account.

Bearings The bearings (and ring wheels) are the interfaces where the running gears and housing are assembled. Although the mass and inertia associated to the bearings can be assumed to be negligible, the stiffness of the bearings probably significantly influences the dynamic behavior and thus needs to be included.

Yaw motor The yaw motor drives the input shaft of the yaw gearbox. Only the rotational inertia of the stator influences the dynamic behavior, since it is greatly amplified by the transmission ratio of the yaw gearbox. In comparison to the gearbox mass, the motor mass is small and therefore neglected. Hence, the motor is simply added as a rotational inertia at the input shaft.

Gear teeth interaction The gear teeth interaction accounts for an important part of the running gear flexibility and is also one of the hardest features to model. Section 7.6.2 will treat this model in more detail.

Due to a lack of information, a number of parameters in the current gearbox model remain unknown. These are:

- The stiffness of the bearing connecting the internal gears to the housing is estimated.
- The distribution of damping and friction in the yaw gearbox is unknown and neglected. Furthermore, damping resulting from the lubrication oil is also neglected.
- Currently the inertias of the gears and planet carriers are estimated from their geometry. Ideally these are to be determined from a detailed CAD-model or measurements.

7.6.2 Gear Interaction

The main challenge in building the yaw gear model was modeling the gear teeth interaction. In order to keep the model relatively simple and avoid the need for modeling the gear teeth contact in detail, a number of assumptions were made:

- The gear teeth stiffness is linear and time invariant. Varying stiffness effects due to changing numbers of gear teeth in contact are assumed to be small and hence neglected.
- Sliding of gear teeth is neglected, so no friction forces are taken into account.
- The gear teeth are assumed to be in contact at all times (no play). Impact forces (backlash) are thus not included.
- The yaw gear is constructed mostly of steel, so damping is assumed to be small and thus neglected. However, the lubrication oil of the gears probably adds damping.
- Since the yaw system is analyzed during standstill conditions, gyroscopic effects are not relevant. Should the system be analyzed around a certain (constant) operational yawing velocity, these effects could be added in a linearized way. Probably these gyroscopic forces are only relevant for the high speed side of the gearbox.
- In the derivation of the gear teeth stiffness matrix, the displacements and rotations are assumed to be small. This simplifies the analysis and results in a linear stiffness matrix.

Based on the above assumptions a model for the gear interaction can be constructed by modeling the gear teeth as a three dimensional linear spring. One can derive a "gear stiffness element", based on the schematic drawing in figure 7.6 showing two interacting gears. The derivation is based on similar work presented for example in [149, 119, 226].

Figure 7.6 shows an interacting gear pair. The gear mesh stiffness is shown as a spring with stiffness k_g. This gear mesh stiffness can be determined using the ISO 6336-1:2006 guidelines [100]. The gears are interacting in the *plane of action*, which is defined by an angle φ with respect to the positive x-axis. This angle is a function of both the geometric angle γ between the gear centers and the *pressure angle* α of the gears. The pressure angle is a design parameter of the gears. Note that the angle of the plane of action is dependent on the driving direction of the gear pair. If the driving direction switches, the pressure angle of the gears changes sign. Hence, one can write the angle of the plane of action as:

$$\varphi = \gamma + \left(\frac{\pi}{2} - \alpha\right) \operatorname{sign}(\tau), \tag{7.1}$$

here τ is the driving direction of the gears (positive for clockwise rotation of the central gear, negative for counterclockwise rotation). The top view of the interacting gear pair in figure 7.6 shows the helix angle β. When this angle is zero, the gears are called *spur gears*. Note that when the driving direction changes, the helix angle β changes sign.

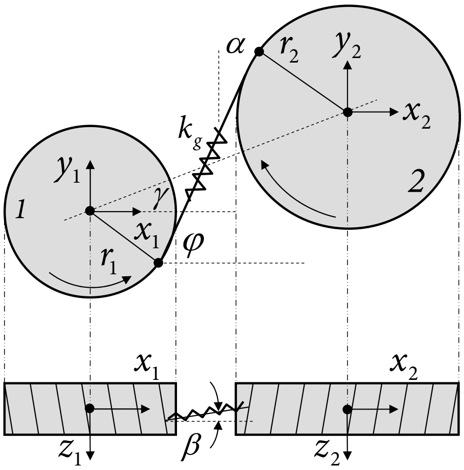


Figure 7.6: Schematic representation of gear interaction.

An energy approach can be used to derive the gear stiffness element. This requires an expression of the deflection of the gear mesh as a function of the degrees of freedom of the centers of the interacting gears. These DoF can be assembled in the vector $\boldsymbol{q}$ as:

$$\boldsymbol{q} = \begin{bmatrix} x_1 & y_1 & z_1 & \theta_{x1} & \theta_{y1} & \theta_{z1} & x_2 & y_2 & z_2 & \theta_{x2} & \theta_{y2} & \theta_{z2} \end{bmatrix}^T. \tag{7.2}$$

Since the gear mesh can be loaded by compressive forces only, a compressive deflection is taken positive. Based on the figure above, one can then derive the deflection of the gear mesh as

$$\begin{aligned} \delta = {} & (x_1 \cos\varphi - x_2 \cos\varphi + y_1 \sin\varphi - y_2 \sin\varphi + r_1\theta_{z1} + r_2\theta_{z2}) \cos(\beta \operatorname{sgn}(\tau)) \\ & + \left(z_2 - z_1 + \theta_{x1} r_1 \cos\varphi + \theta_{x2} r_2 \cos\varphi + \theta_{y1} r_1 \sin\varphi + \theta_{y2} r_2 \sin\varphi\right) \sin(\beta \operatorname{sgn}(\tau)) \end{aligned} \tag{7.3}$$

Under the assumption of small displacements, the spring deflection clearly is a linear function of the degrees of freedom of the gears. One can write the potential energy in the spring as:

$$V = \frac{1}{2} k_g \delta(\boldsymbol{q})^2 \tag{7.4}$$

The stiffness matrix can then simply be obtained by:

$$\boldsymbol{K} = \frac{\partial^2 V}{\partial \boldsymbol{q} \partial \boldsymbol{q}} \tag{7.5}$$

This gear stiffness element can now be defined between and assembled with any other structural element such as beam elements. Note that analogous to the derivation above one can derive the stiffness matrix representing the gear mesh stiffness of internal gears, i.e. the interaction between a planet and ring gear. It turns out that this stiffness matrix can be found simply from the derivation above by taking a negative radius for the internal gear [149].

Using these gear stiffness elements, a model for the dynamics of the interacting gears can be created. Together with the beam models for the shafts and carriers and simple spring models for the bearings, as described in section 7.6.1, a dynamic model for the yaw gearbox internals was created in Matlab. A plot of this model is shown in figure 7.5 (b).

7.6.3 Gearbox Housing

The internals of the gearbox are mounted in the gearbox housing, which thereby functions as an interface between the running gears and the bedplate. Since a CAD model of the gearbox housing was not available, a geometrically simplified gearbox housing is created based on the drawings from the supplier. This geometry is used in ANSYS to build the structural model; a plot of the model is shown in figure 7.5 (b). The gearbox housing is casted, therefore the mechanical properties of cast steel are used for the structural model.

	Density [kg/m^3]	Young's modulus [GPa]	Poisson ratio [−]
Yaw gearbox housing	7800	200	0.30

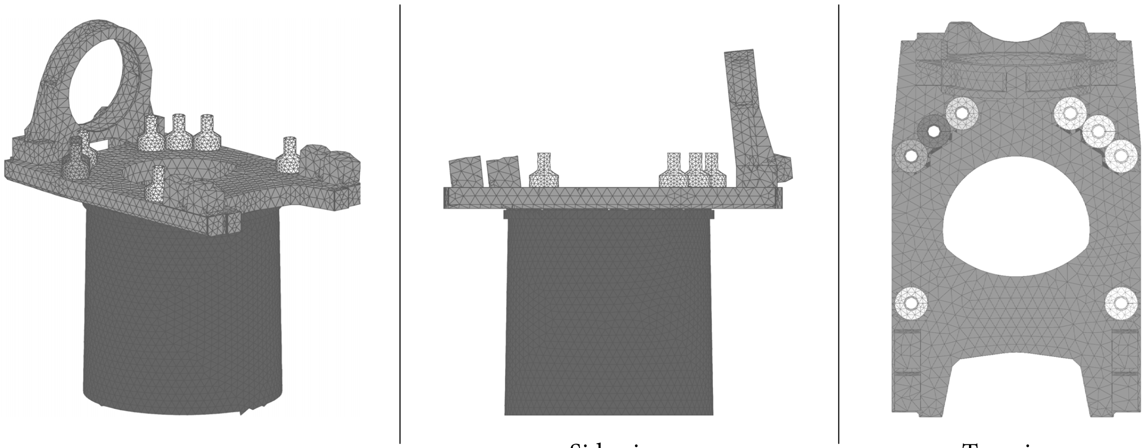
Four ring wheels and three bearings connect the housing to the internal gears. The ring gears are bolted into the housing while the bearings are pressed into position. As the housing itself is already very stiff, one can thus assume a rigid connection. Using the rigidification technique of section 4.3, seven master nodes are created in the gearbox housing to facilitate the assembly of the internal gear model. When the gearbox housing and internals are assembled they can be used in the substructuring analysis of the yaw system. A simple check on this model is to verify the number of rigid body modes. These should be seven in total: six rigid motions of the complete gearbox and one rigid rotation mode of the internals. This is indeed the case.

Finally, note that attempts have been made to validate the yaw gearbox model through dynamic measurements. However, successful measurements on the yaw gearbox proved far from trivial such that the model could not be validated. For details see appendix B.2.

7.7 Assembled Models & Analysis Results

With all components modeled, the next step is to create assembled models of the wind turbine yaw system. The following abbreviations are introduced to refer to the different component models: YR-TT for the yaw ring and towertop substructure model, YP for the yaw pad model array, BP for the bedplate component model and YGB for the yaw gearbox model array.

Firstly, all unreduced component models are assembled in a primal way, see section 3.3, leading to a yaw system model of almost 300k DoF. This assembled model is created using the AssemblyTool (see appendix A.2) and shown in figure 7.7. In the remainder of this section this unreduced assembly will serve as the reference model. An overview of the component and assembled models' properties is given in table 7.1. As can be seen the total model consists of over 7k interface DoF, which are caused by the extensive interfaces of the yaw pads. Note that due to the non-conforming interface meshes the (interface) DoF counts cannot



Side view

Top view

Figure 7.7: Assembled model of the yaw system. Note that the yaw gearbox internals are not shown in the plot due to their special element type which is not supported by the AssemblyTool's plotting function. Also, due to their location in the system the yaw pads are not visible in these plots.

simply be added and subtracted to find the total number of (interface) DoF of the assembled system. The non-conforming meshes are assembled using the discrete least squares compatibility method described in section 3.6.2.

Model	# Elt.	# Nodes	# DoF (n)	# int. DoF (n_b)	Pcs.	Σn	Σn_b
YR-TT	16411	32290	96891	4068	1	96891	4068
YP	104	243	729	378	22	16038	8316
BP	24517	41864	125631	2826	1	125631	2826
YGB	1412	2501	7899	12	8	63192	96
Totals					32	301752	15306
Primal assembly					32	293712	7299

Table 7.1: Properties of component and primal assembled models; n_b denotes the number of interface displacement DoF.

In the remainder of this section, the component models are reduced using different methods from chapters 2 and 4, and subsequently assembled to create more compact dynamic models of the yaw system. In order to assess their accuracy, these models are subjected to a modal analysis and their eigenfrequencies and mode shapes are compared to those of the full reference model. Two criteria are defined for this comparison:

- Relative frequency error: the relative error of the eigenfrequencies with respect to the reference solution is compared. In order to qualify as an "accurate" result, the relative frequency error should be no more than 1%.
- Mode shape error: the MAC matrix is computed to quantify the correlation between the expanded mode shapes from the reduced models with respect to the reference solution (see section 4.7.2). The diagonal MAC values, so those of matching modes, are

subtracted from 1 and plotted for each mode. The resulting mode shape error should be less than 0.1 (i.e. MAC value of 0.9) in order to be considered accurate. Note that the MAC is computed using (4.44), so no weighting with the full assembled mass matrix is applied.

In subsequent plots these accuracy thresholds will be indicated by a dashed black line. The next subsection will discuss the results for the component reduced models, while subsection 7.7.2 treats the results obtained when additional interface reduction is applied.

7.7.1 Results Component Reduction

In this section the component models are reduced using the Craig-Bampton (CB), Dual Craig-Bampton (DCB) and Rubin methods described in sections 2.5, as well as using the Mixed Craig-Bampton method proposed in section 2.6. Hence, all original boundary displacement DoF are retained, or replaced by interface force DoF in the case of the DCB and (possibly) MCB methods.

For the MCB method the automated selection of free or fixed interface DoF was applied as described in section 2.6.1 (with $c = 1$), and it was chosen to assemble equal-stiffness interface DoF in a primal manner. This led to the fixed and free interface DoF selection as shown in table 7.2, where n_b again denotes the number of interface displacement DoF (to be fixed during reduction) while n_g denotes the number of interface force DoF (to be left free during reduction).

Model	YR-TT		YP		BP		YGB	
	n_b	n_g	n_b	n_g	n_b	n_g	n_b	n_g
No. DoF	37	4031	8316	0	48	2778	86	10

Table 7.2: Selection of fixed and free interface DoF in the Mixed CB method; n_g denotes the number of interface force DoF.

As can be seen, all interface DoF of the yaw pads are to be reduced with fixed interface modes, its reduction basis thus corresponds to a normal CB basis. This was to be expected since the polyamide pads are very soft in comparison with the steel structures connected to it. For the tower top and yaw ring, most interface DoF are to be reduced with free interface modes, except for a few DoF with the yaw gearboxes. Their stiffness is approximately equal to that of the yaw gearboxes and hence to be assembled primally. The bedplate interface DoF are also mostly to be left free, again except for the eight (rigidified) interfaces with the yaw gearboxes.

For simplicity, all components are reduced using 30 vibration modes, except the yaw pads which are reduced using 15 modes each. Obviously, using the adaptive model reduction methods from chapter 5, a more efficient distribution of the modal DoF across the different components can be made; this is however out of the scope of this initial case study. The four assemblies of reduced component models analyzed in this section are listed in table 7.3. The abbreviations in the first column denote the names of the assemblies and correspond to the names used in subsequent figures.

From table 7.3, note that the assembly of DCB reduced components (DCB30) has more DoF than the CB30 and R30 models. This is due to the fact that now interface force DoF are

Model	YR-TT	YP	BP	YGB	Assembly			
	n_η	n_η	n_η	n_η	Σn_b	Σn_λ	Σn_η	Total no. DoF
REF	–	–	–	–	7299	–	–	293712
CB30	30	22 × 15	30	8 × 30	7299	–	630	7929
DCB30	30	22 × 15	30	8 × 30	–	8007	630	8637
R30	30	22 × 15	30	8 × 30	7299	–	630	7929
MCB30	30	22 × 15	30	8 × 30	7289	6819	630	14738

Table 7.3: Overview of the reduced assembled models; n_η denotes the number of modal DoF and n_λ denotes the number of Lagrange multipliers.

assembled instead of interface displacement DoF. As can be understood from the relations between the Boolean assembly matrices $\boldsymbol{B}_b$ and $\boldsymbol{L}_b$ in chapter 3, the number of unique Lagrange multipliers, and hence interface DoF, is equal to the total number of interface DoF minus the number of unique displacement DoF.[3] The components of the MCB reduction are assembled in a mixed way according to the procedure in section 3.5, such that both interface displacement DoF and Lagrange multipliers are needed. This leads to a rather large system of assembled equations.

The results of the modal comparison of the reduced yaw system models are presented in figures 7.8 to 7.10. Figures 7.8 and 7.9 show the relative frequency error and mode shape errors with respect to the full model, whereas the plots in figure 7.10 (a)–(d) show the MAC matrix for the modes of the four reduced models with respect to the modes of the full model. Note that in figures 7.8 and 7.9 the results from the DCB30 assembly are corrected for so-called “spurious modes” (as discussed in section 2.5.4).

From the obtained results, a number of observations can be made:

- All the reduction methods show excellent results and are accurate up to at least the eightieth eigenmode and eigenfrequency of the full model. Still, the total number of DoF are reduced by a factor of almost 40 for the CB30, DCB30 and R30 models and by a factor of 20 for the MCB30 model.
- The classic CB, DCB and Rubin methods have similar accuracy, but the MCB seems to perform slightly better across the complete frequency range. This is especially true for the error on the mode shapes. These good results could maybe be explained by the fact that the assembled MCB model has many more DoF than the other assemblies. On the other hand, this larger model size is only the result of the mixed assembly procedure; the actual component models do not contain any more relevant content (i.e. deformation shapes) than in the other models. Nonetheless, this shows that a more accurate reduced model is obtained when the reduction bases are constructed with some knowledge of the neighboring components and the assembly procedure is tailored to the content of these bases.
- In figure 7.10 (b) the MAC plot between the modes of the reference system and those of the Dual Craig-Bampton system is shown. From this plot a spurious mode can

---

[3] So here the number of unique Lagrange multipliers is 15.306 – 7.299 = 8.007. Note that if all interface meshes are conforming and each interface node is coupled to only one other node, then the number of unique Lagrange multipliers and unique interface displacement DoF are equal. In case a node belongs to m substructures, for interface corners with a “multiplicity” > 2, there would $m - 1$ multipliers and only 1 unique interface DoF.

clearly be seen at mode 58 of the reduced model. Such spurious modes originate from the fact that the DCB method only enforces weak compatibility, it therefore allows motion of the interface DoF which is physically not possible (e.g. relative sliding of the interfaces). These non-physical modes are an artifact of the reduction procedure. Depending on the selection of fixed and free interface DoF, such modes can also occur in MCB reduced models.

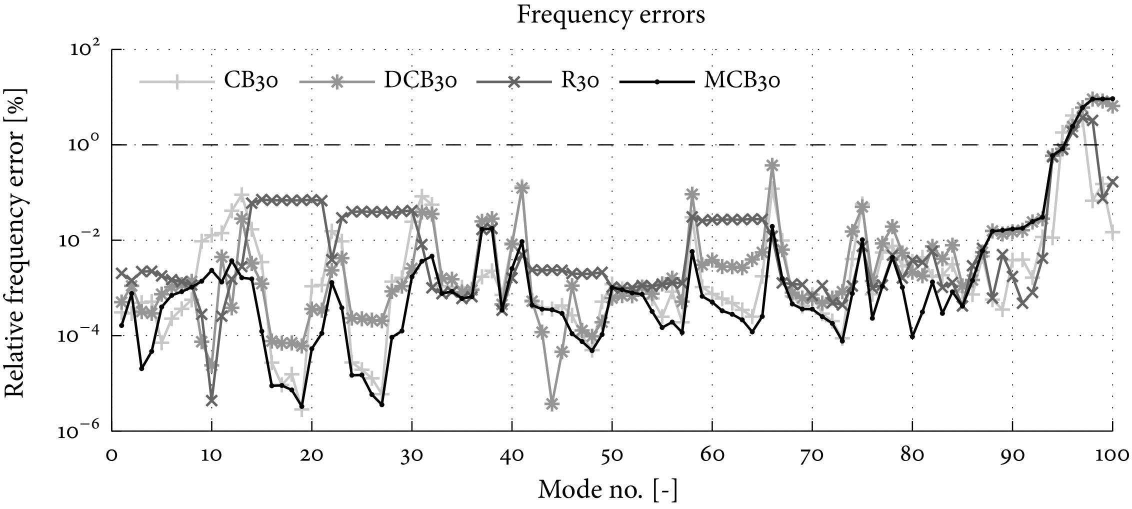


Figure 7.8: Frequency errors of the reduced models with respect to the full model.

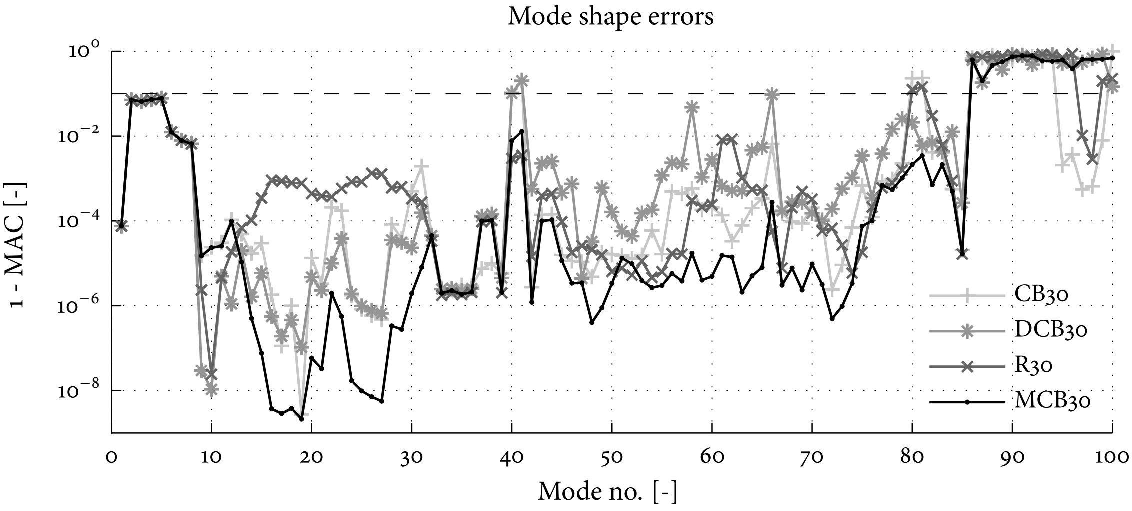


Figure 7.9: Mode shape errors of the reduced models with respect to the full model.

7.7.2 Results Component & Interface Reduction

Although the assemblies of reduced component models show a reasonable reduction of DoF with respect to the full model, with approximately 8k DoF (and even 15k DoF for the MCB30 model) they are still orders of magnitude larger than the structural models typically used in aero-elastic simulations in the wind industry. In order to reduce the number of DoF even further, the interface reduction techniques from chapter 4 are therefore applied. Since the

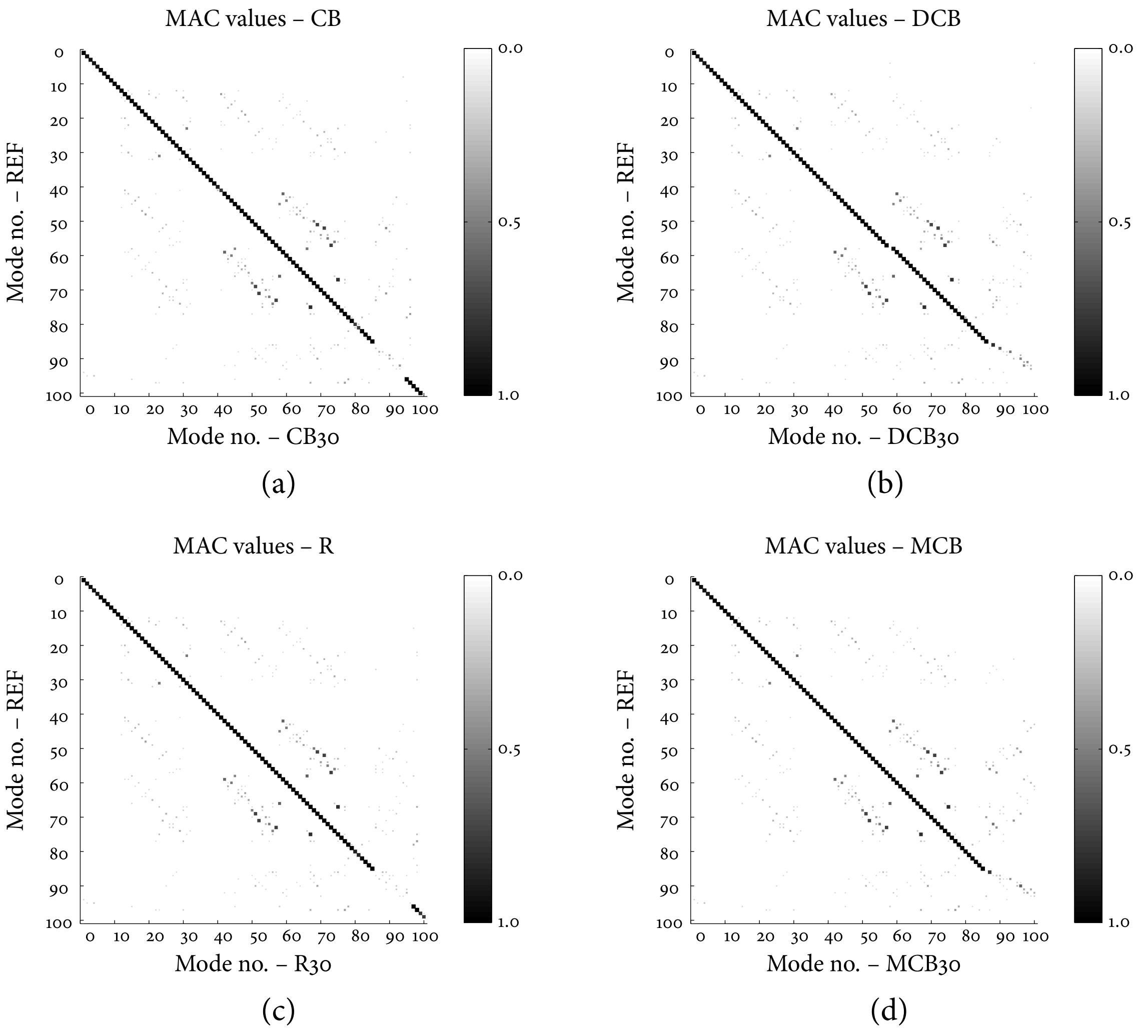


Figure 7.10: MAC between the expanded modes of the reduced models and the reference model: Craig-Bampton (a), Dual Craig-Bampton (b), Rubin (c) and Mixed Craig-Bampton (d).

number of interface modes used in the interface reduction is a crucial parameter in obtaining accurate but compact models, two interface reduced versions are created for each of the three original reduced models, employing respectively 100 and 200 interface modes. The different variants analyzed in this subsection are listed in table 7.4.

Since the MCB model contains both a unique field of interface displacement DoF as well as Lagrange multipliers, it needs to be reduced with both interface displacement modes and interface force modes (see sections 4.4 and 4.5). As can be seen from table 7.4, this again leads to reduced models that are (slightly) larger than those of the traditional reduction methods.

Next, the same modal comparison is applied to these interface reduced models as before; the results are shown in figures 7.11 to 7.14. Figures 7.11 and 7.12 show the error for the three different reduction methods with respect to the full model when the interface is reduced with 100 modes; figures 7.13 and 7.14 show the same comparison when 200 modes are used. Again, a number of observations can be made from these results:

- As expected, the interface reduced models are less accurate then the original CB30, R30 and DCB30 systems. The big advantage however, is that by applying the interface

Model	YR-TT	YP	BP	YGB	Assembly			
	n_η	n_η	n_η	n_η	$n_\gamma - \boldsymbol{\Phi}_\gamma$	$n_\gamma - \boldsymbol{\Phi}_\lambda$	Σn_η	Total
REF	–	–	–	–	7299	–	293712	
CB30ir100	30	22 × 15	30	8 × 30	100	–	630	730
CB30ir200	30	22 × 15	30	8 × 30	200	–	630	830
DCB30ir100	30	22 × 15	30	8 × 30	–	100	630	730
DCB30ir200	30	22 × 15	30	8 × 30	–	200	630	830
R30ir100	30	22 × 15	30	8 × 30	100	–	630	730
R30ir200	30	22 × 15	30	8 × 30	200	–	630	830
MCB30ir100	30	22 × 15	30	8 × 30	100	100	630	830
MCB30ir200	30	22 × 15	30	8 × 30	200	200	630	1030

Table 7.4: Overview of the interface reduced assembled models; n_γ denotes the number of interface modes, $\boldsymbol{\Phi}_\gamma$ denotes interface displacement modes and $\boldsymbol{\Phi}_\lambda$ denotes interface force modes.

reduction, the total number of DoF is reduced by a factor of approximately 300-400 with respect to the full model. This results in much shorter computation times as well as lower storage requirements for the component reduction bases.

- All the interface reduced models show good results for the relative frequency error, where they show little difference with respect to the non-interface reduced models. For the mode shape errors this is true to a lesser extent, where the accuracy deteriorates already at some of the lower modes. This is probably due to the fact that the interface deformations play an important role in this modes of the total system. It is believed that this can avoided by better distributing the modes across the different components and interfaces, without significantly increasing the model size, by application of the adaptive reduction algorithms of section 5.7. This is however out of the scope here.

- Again, the MCB model outperforms the other reduced models in terms of accuracy. However, as before, the MCB reduced model has more degrees of freedom to deform in, making direct comparison with the other models not completely fair. Nonetheless the difference in model size between the MCB and other models is only relatively small here, making the good accuracy of the MCB models quite remarkable.

- The three other models (CB30ir, DCB30ir and R30ir) on average perform more or less similarly, although differences can be observed. For instance, it seems that the DCB30ir models provide the best accuracy on the eigenfrequencies, especially for the lowest modes, but perform not as good when it comes to the accuracy of the mode shapes.

- From figures 7.13 and 7.14 it shows that the MCB model reduced with 100 interface modes performs better than the one reduced with 200 interface modes. This contradictory outcome is believed to be the result of numerical issues, probably caused by bad matrix conditioning due to mixing of force interface modes (i.e. flexibility based) and displacement interface modes (i.e. stiffness based). Due to this bad conditioning round-off errors from the solver can be amplified. Since higher modes are generally less accurately captured, this can result in errors in the MCB30ir200 reduced system that are not observed in the MCB30ir100 version. The consequence is that after mode

40 the accuracy deteriorates quite badly. This is undesirable in practice, hence some sort of indicator should be developed to signal this deterioration. To avoid this in the future, scaling and/or preconditioning could maybe be applied in the solver. Note that the R30ir200 model seems to suffer from the same problem, albeit to a lesser extent.

All in all these results show that with component and interface reduction, compact yaw system models can be created that still provide acceptable accuracy with respect to the full model. Since the reduced substructure models originate from CAD models, all geometrical information and details are accounted for. Hence, the substructuring and reduction methods provide a powerful structural dynamic analysis tool for use in wind turbine engineering.

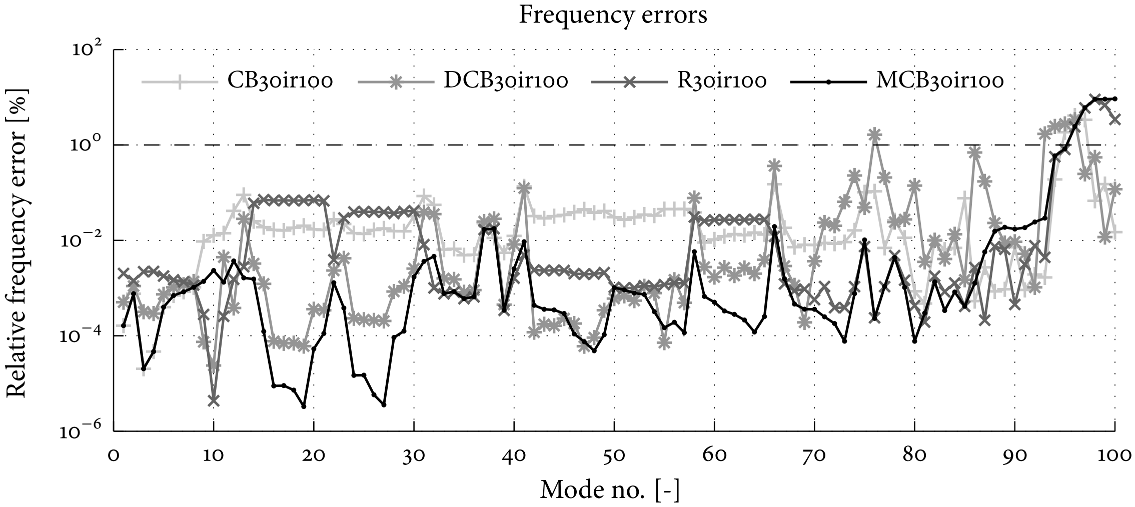


Figure 7.11: Frequency errors of the component and interface reduced models with respect to the full model, interface reduction with 100 modes.

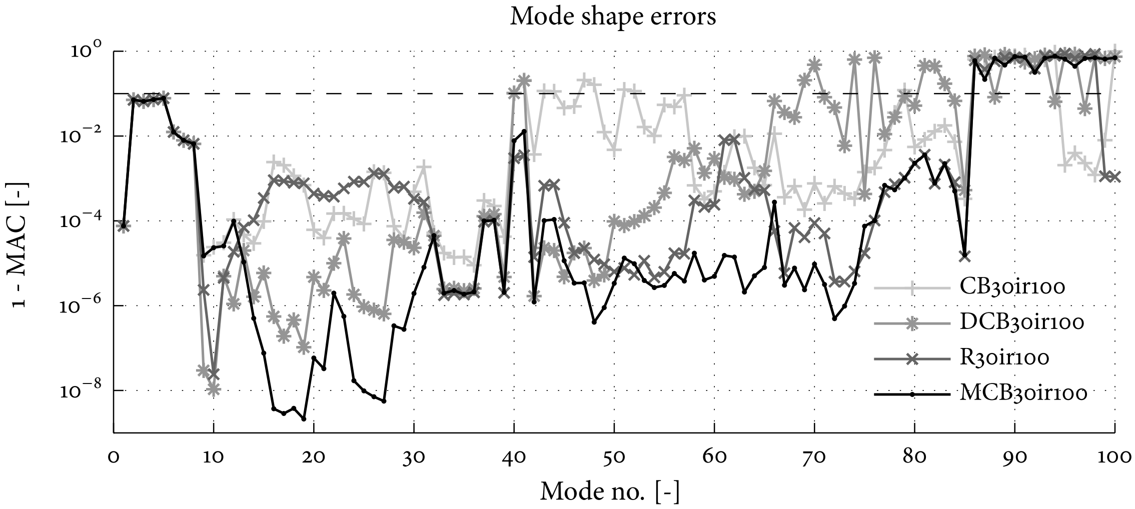


Figure 7.12: Mode shape errors of the component and interface reduced models with respect to the full model, interface reduction with 100 modes.

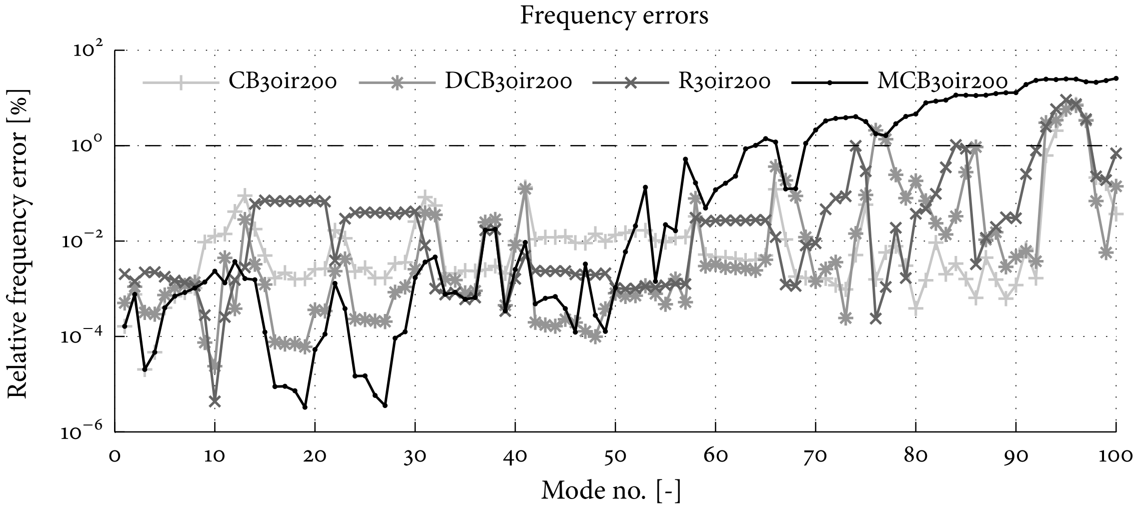


Figure 7.13: Frequency errors of the component and interface reduced models with respect to the full model, interface reduction with 200 modes.

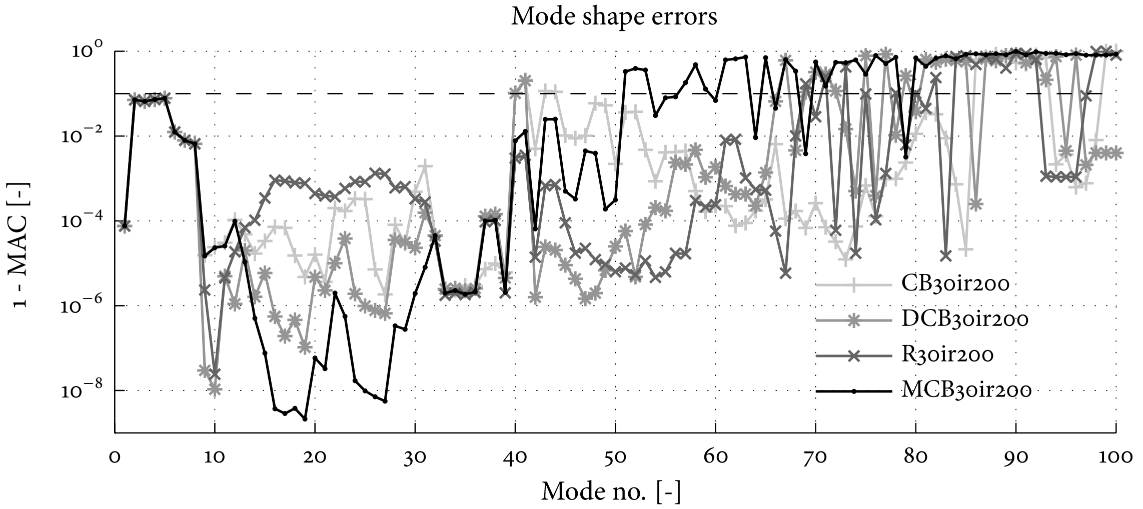


Figure 7.14: Mode shape errors of the component and interface reduced models with respect to the full model, interface reduction with 200 modes.

7.7.3 SUMAC Analysis

To conclude the analysis of the yaw system, in this subsection a SUMAC analysis is performed. Although unrelated to the previous comparisons between full and (interface) reduced models, it serves to show the use of the SUMAC indicator defined in section 4.7.3. As was explained in that section, the SUMAC correlates the trace of the assembled modes on a certain substructure (or subassembly) to the modes of that unassembled substructure. Here the SUMAC is calculated between the modes of the total yaw system model $\boldsymbol{\Phi}_{\mathrm{REF}}$ and those of the subassembly of yaw ring and towertop, yaw pads and bedplate, denoted by $\boldsymbol{\Phi}_{\mathrm{YRTT\ \&\ YP\ \&\ BP}}$. The resulting plot is shown in figure 7.15.

This SUMAC shows the correlation between the global mode shapes and the mode shapes of the yaw ring and towertop, yaw pads and bedplate subassembly. From figure 7.15, one can thus visualize the effect on the mode shapes of the addition of the yaw gearboxes onto the YRTT–

YP–BP subassembly to obtain the total assembly. The figure can be read as follows. Along the horizontal axis the modes numbers of the subassembly are shown, while the vertical axis shows those of the total assembly. Where there is a high correlation between the two mode sets, one can conclude that the addition of the gearboxes has little effect on the subassembly. Low correlation values can be attributed to modes that have significantly changed due to the addition of the yaw gearboxes or to localized modes in the gearboxes.

The SUMAC plot in figure 7.15 can thus be interpreted as follows. Firstly, due to the shifts of the red blocks, which denote a high correlation, one can identify the isolated gearbox modes in the set of global modes. But one can also see from for instance modes 14, 22 and 30 that there is an interaction between the gearboxes and the other assembled components. Finally, it is seen that the added mass and stiffness affect some modes more than others, thereby leading to higher frequencies for some modes and lower frequencies for others. This can be observed at modes 37–41 of $\boldsymbol{\Phi}_{\text{REF}}$, which have a significant correlation with modes 13–17 of $\boldsymbol{\Phi}_{\text{YRTT \& YP \& BP}}$, but where the order of the modes is altered. In conclusion, a SUMAC plot thus provides additional insight in the assembled system's behavior and the role of the substructure models therein.

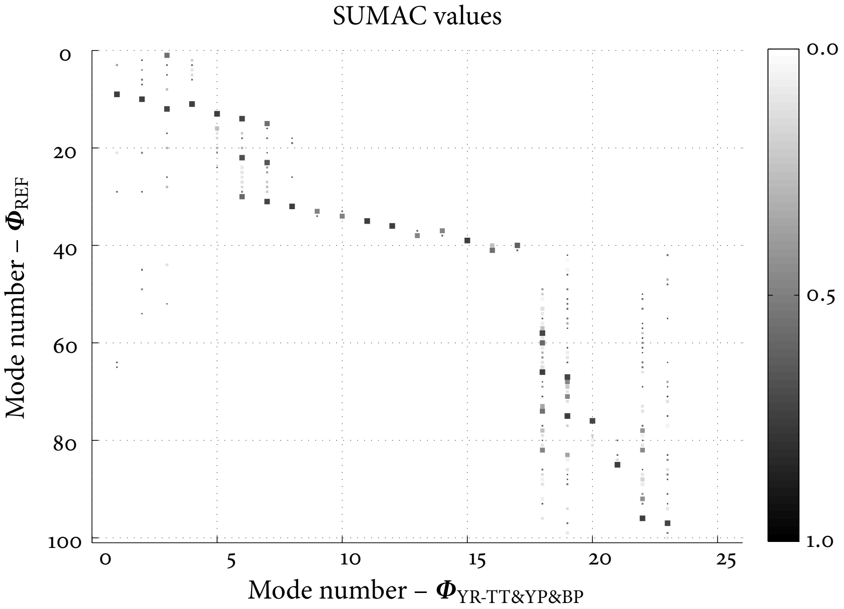


Figure 7.15: SUMAC between the modes of the total model and of the YR-TT, YP and BP subassembly.

7.8 Summary

Every modern wind turbine in the multi-megawatt class is equipped with an active yaw system, in order to keep the rotor plane orthogonal to the wind direction by rotating the rotor-nacelle assembly around the vertical tower axis. Since the yaw system is modeled in an extremely simplistic way in the aero-elastic models used in the wind industry, this chapter was concerned with creating a detailed dynamic model of the yaw system of a Siemens 2.3 MW wind turbine using the methods outlined in chapters 2, 3 and 4.

A description of the yaw system was given in section 7.2, which also established the system boundaries and identified the relevant structural components, being the bedplate, yaw pads, tower top and yaw ring and the yaw gearboxes. The modeling of these components was detailed in sections 7.3 to 7.6 respectively. Modeling of the first three components was achieved in a quite straightforward manner by reusing existing CAD and FE models, but creating a dynamic model for the yaw gearboxes required a more elaborate approach to account for the dynamics of the internal gears. Although not treated in this chapter, measurements have also been performed to validate the bedplate and yaw gearbox models. See appendix B.

The assembled yaw system models were analyzed in section 7.7. The primal assembly of the full models consisted of almost 300k DoF, more than 7k of which were on the substructure interfaces. Reduced assemblies were created using the methods of chapter 2, specifically the Craig-Bampton (CB), Dual Craig-Bampton (DCB), Rubin and Mixed Craig-Bampton (MCB) methods. For simplicity all components were reduced using 30 vibration modes, except the yaw pads which were reduced with 15 modes each. Since all interface DoF were initially retained this resulted in reduced models of around 8k DoF for the first three methods and 15k DoF for the MCB method, a reduction of a factor 40 and 20, respectively, compared to the full model. From a comparison of their modal properties, with respect to the full model, it was shown that all four reduced models provide an accurate prediction of the dynamic behavior up to the eightieth mode. The Mixed CB method showed the best accuracy of the four methods, which however comes at a larger reduced model size.

These reduced models were still deemed too large, such that the interface reduction techniques from chapter 4 were applied for further DoF reduction. This resulted in models of approximately 700-1000 DoF, achieving a reduction factor of around 300-400. Performing the same modal comparison revealed that the accuracy of these very compact models was still good, especially when compared to the non-interface reduced models. Hence, it was shown how applying component and interface reduction allows to create compact yet accurate dynamic models of wind turbine components and assemblies. Furthermore, compared to existing methods, the proposed Mixed Craig-Bampton method has proven to be a more systematic and accurate way of doing so.

8

Design Modification of a Bedframe Structure

8.1 Introduction

It is common practice among wind turbine manufacturers to base new wind turbine models on existing designs by changing the rotor diameter and/or rated generator power. This is an effective way of bringing down the cost of wind generated electricity, since these upgraded wind turbines usually provide substantially increased electricity output at often only moderate additional cost. For wind turbine manufacturers this approach also has several benefits. For instance, there is no need to design a new turbine from scratch, lessons learned from the previous design can be implemented, and last but not least the turbine designs are pushed more and more to the limit which benefits profitability and market position.

Due to the new configuration of the wind turbine, it has to be checked whether all components have sufficient capacity to withstand the increased loading. In this process small design modifications to the nominal components are often needed to ensure their reliability in the new turbine.

This chapter zooms in on this incremental design approach for one specific structure, in order to assess the efficiency of the dynamic substructuring approach for such design problems. Indeed, in chapter 6 it was explained that iterative (parametric) modifications to one or more components could partly undermine the computational efficiency. A solution was proposed in the form of preconditioned iterative updating methods that reuse as much information as possible from the nominal reduced model. This chapter therefore forms a test case for these updating methods.

The structure of interest is the bedframe, that has already been analyzed in the previous chapter. This structure was chosen due to its central function in the nacelle and, more importantly, due to the fact that it absorbs and transfers all the aerodynamic loads going into the turbine. In practice it is therefore likely that this structure needs to be modified when the loads on the turbine increase.

This chapter is organized as follows. The structural model of the bedframe and the design modifications at hand will be described in the next subsection. Subsections 8.3 and 8.4 will thereafter treat the efficiency of the updating of the static and vibration modes, respectively. The overall efficiency of the updating method will be addressed in section 8.5. The situation where design modifications are applied sequentially, and the effect on the computational efficiency of the updating method, is covered in section 8.6. Finally, the updating strategy will be compared to existing updating/reanalysis strategies in section 8.7. The chapter is ended with a summary in section 8.3. Note that the work in this chapter is based on the publications in [214, 215].

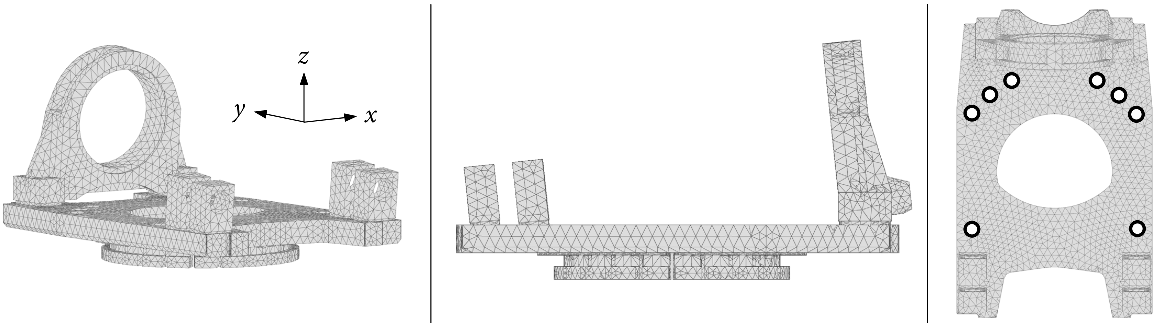
8.2 Structural Model & Design Modifications

In this chapter the bedframe structure is considered that was described for the modeling of the yaw system in section 7.3, that is, the bedplate including the main bearing housing and gearbox supports. This total structure will here be referred to as the the "bedframe". As described in section 7.3, the structural finite element model of the bedframe is based on a CAD model. For the purpose of this chapter, three finite element models have been created from this CAD model in ANSYS, which only differ in their mesh size. These models will be referred to as the "coarse", "normal" and "fine" models, respectively; their properties are listed in table 8.1. Considering three FE models allows to study the influence of the model size on the effectiveness of the iterative updating approach proposed in chapter 6.

	Coarse	Normal	Fine
# DoF	123459	246762	511953
# Elements	23651	49882	107681

Table 8.1: Bedplate FE model properties.

All three finite element models are meshed using quadratic (i.e. ten-node) tetrahedral elements with only translational nodal DoF. Note that this somewhat simplifies the updating computations, as difficulties with scaling of rotational DoF are avoided. The normal finite element model of the nominal bedframe structure is shown in figure 8.1, which also indicates the coordinate system used to define the different design modifications. The bedframe FE models will be reduced using the Craig-Bampton method, see section 2.5.2. This means that the reduction basis ingredients are the static constraint modes and fixed interface vibration modes, derived in sections 2.3.1 and 2.4.3 respectively.



Side view

Top view

Figure 8.1: Finite element model of bedframe structure.

The interfaces of the bedframe considered in this chapter are the black areas shown in the topview plot. At these locations the yaw gearboxes are connected, as was discussed extensively in the previous chapter. Each interface consists of multiple nodes which have been replaced by a single 6 DoF master node by assuming the interface region behaves rigidly. In total, this leads to 48 interface DoF, regardless of the mesh size.

The reason for not considering the other interfaces of the bedframe (i.e. interfaces with yaw pads, main bearing, gearbox) is twofold. Firstly, as was explained in section 7.3 for the interface with the yaw pads, these interfaces cannot in general be assumed to behave as rigid regions. Hence, the number of interface DoF, and consequently the number of static modes, varies with the mesh size, making one-to-one comparison between the different FE models troublesome. Secondly, larger interfaces such as the yaw pads cause many static modes which negatively affects the efficiency of the CG based updating approach, such that recomputation of the static modes might become more efficient. In practice one would however often apply interface reduction to these extensive interfaces. These interface modes can then in turn be updated using the vibration mode updating method; maybe one could devise some scheme where the static modes and interface modes are updated simultaneously, such that only a limited number of static modes need to be computed. This is however out of the scope here.

To resemble realistic situations, the bedframe structure is subjected to the following design modifications:

Case 1 – Global geometric change in x, z directions The complete bedframe geometry is modified to grow in the global x (width) and z (thickness) directions; both directions are scaled simultaneously by the same factor. The length of the bedframe as well as its structural properties, such as the Young's modulus, Poisson ratio and density, remain constant. In order to study the effect of the magnitude of the design change, the following series of 15 scaling factors is used: [1.005, 1.01, 1.02, 1.03, 1.04, 1.05, 1.075, 1.10, 1.125, 1.15, 1.20, 1.25, 1.30, 1.40, 1.50]. This means that for each FE model 15 modified variants will be created.

Case 2 – Global geometric change in y direction The complete bedframe geometry is modified to grow by a certain percentage in the global y (length) directions. The same 15 variants are considered as in case 1. The other dimensions and the structural properties remain constant.

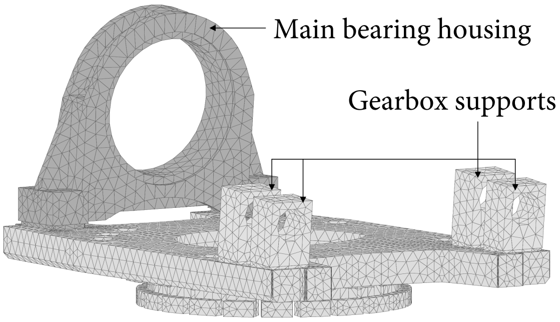


Figure 8.2: Division of the bedframe finite element model into three parts.

Case 3 – Local material property change For this case the structure is divided into three parts as shown in figure 8.2: the bedplate, main bearing housing (designated by "mb") and gearbox supports (designated by “gs”). Two variants are considered for each FE model, namely where the material properties of the bearing housing and gearbox supports are individually changed from steel to aluminium. The material properties used for both materials are listed in table 8.2.

	Density [kg/m^3]	Young’s modulus [GPa]	Poisson ration [−]
Steel	7850	210	0.30
Aluminium	2700	70	0.35

Table 8.2: Material properties.

In order to quantify the “intensity” of the series of design changes described above, the following metrics are defined based on the Frobenius norm of the structural matrices:

$$\delta_K = \frac{\|\Delta \boldsymbol{K}\|_F}{\|\boldsymbol{K}_{(0)}\|_F} \cdot 100\% \quad \text{and} \quad \delta_M = \frac{\|\Delta \boldsymbol{M}\|_F}{\|\boldsymbol{M}_{(0)}\|_F} \cdot 100\%, \tag{8.1}$$

with the Frobenius or “entrywise” norm defined as:

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{n} \sum_{i=1}^{n} |a_{ij}|^2} \tag{8.2}$$

Since the models at hand only consist of translational DoF, these numbers can be roughly interpreted as the percentage by which the global stiffness and mass properties are affected by the design modification. In the discussion on the efficiency of the updating procedures for the static and vibration modes in the next two sections, it will be addressed how the following aspects affect the results:

- Algorithm settings
- Preconditioner and initial guess
- Design modification and model size

Note that to truly gain insight in the performance of the updating methods, the next two sections consider each design modification with respect to the nominal model. Sequential updating, where the structure is subjected to a series of design modifications and updating is only performed with respect to the previous modification, is treated separately in section 8.6. The main comparison criterion in these investigations will be CPU time.

As explained in section 6.1, existing updating/reanalysis methods enrich and/or approximate the reduction basis of the modified model. This complicates direct comparison of the computational efficiency of those methods and the updating methods of chapter 6 troublesome. Instead, since the updated basis provides the same reduced model size and accuracy (see section 8.7), the CPU efficiency will be compared to that of full basis recomputation with efficient standard solvers.

All calculations in this chapter are performed in Matlab R2009b on a quad-core Intel Xeon machine running Windows XP64. The matrix factorizations and forward/backward substitutions in Matlab are performed using the SD Toolbox, and specifically using the sparse multi-frontal spfmex solver based on LDLt decomposition [17]. This compiled solver uses the routines in the SPOOLES library, written in the C-language using object oriented design (see http://www.netlib.org/linalg/spooles/spooles.2.2.html). Furthermore, all vector and matrix operations, both in the SD Toolbox and in the implementations of the updating algorithms, are performed using the BLAS routines.

8.3 Efficiency of Updating of Static Modes

The test structure has 48 interface DoF, hence an equal number of static constraint modes need to be computed. In order to obtain the static modes for the nominal component a compiled direct solver is applied based on sparse LDLT-type decomposition. This solver is part of the SD Toolbox [17]. For the three FE models, table 8.3 lists the solver's computation times for the factorization of the stiffness matrix and backsubstitution to obtain the 48 static modes.

	Coarse	Normal	Fine
Factorization [s]	11	50	260
Backsubstitution [s]	4	11	33
Total [s]	15	61	293

Table 8.3: CPU times for direct calculation of static modes.

After performing the design modifications as outlined before and rebuilding the finite element matrices, the aim is to compute the static modes of the modified structures starting from the nominal structure using an implementation of algorithm 6.2. In all subsequent calculations the iteration tolerance is set to $\varepsilon = 10^{-6}$.

8.3.1 Effect of Algorithm Settings

In the block CG algorithm shown in alg. 6.2 the block size used in the iterations can be freely chosen. In order to assess the influence of the block size on the efficiency of the CG solver, the normal FE model is taken and analyzed for all design modifications in case 1 and different block sizes ranging from 1 to 48. Figure 8.3 shows the CPU time for a specific block size normalized by the lowest CPU time of all block sizes for that design modification. The size of the markers is inversely proportional to the normalized CPU time, while the dark markers indicate lowest CPU time.

Furthermore, taking case 1, design modification 8 (i.e. 10% scaling in global x and z directions) the effect of varying the block size is listed in table 8.4.

Block size	1	2	4	6	8	12	24	48
CPU time [s]	158	104	94	81	76	88	102	123
# iterations	212	116	64	42	33	23	16	13
# vectors	212	216	224	222	225	263	293	344

Table 8.4: Effect of block size for normal FE model, case 1 – (x, z) scaling by 10%.

From these results the following is observed:

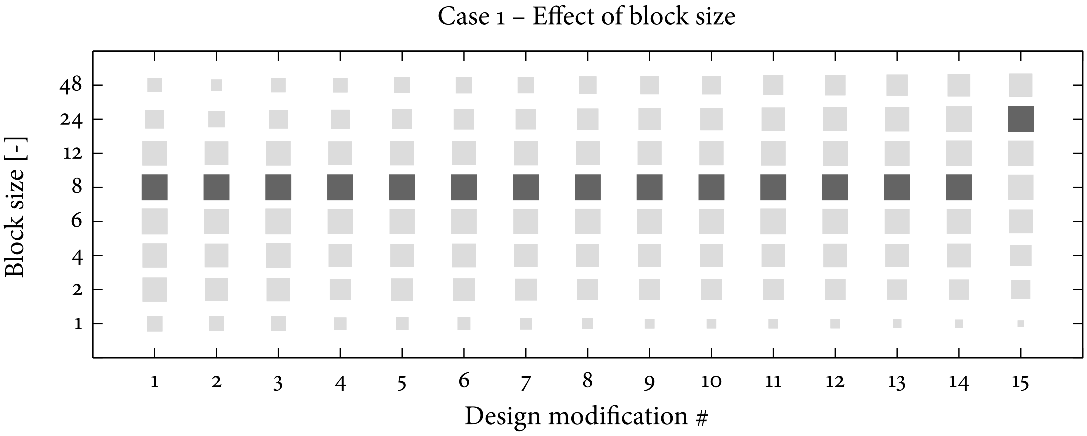


Figure 8.3: Normalized CPU time versus block size.

- For the current implementation and calculation setup, in the majority of cases a block size of 8 is most efficient. This seems to be independent of design modification and model size, as similar results are found for the coarse and fine FE model (not shown here).
- As speculated in section 6.2.3.2, the block algorithm is always more CPU efficient than its single vector equivalent, even for a block size of 2.
- From a memory usage perspective the single vector algorithm is least demanding, as it finds the solution using the lowest number of search directions. This is due to the fact that the vectors result from a higher Krylov order. However, these differences are rather small compared to the differences in CPU time.
- The block size times the number of iterations is in general not equal to the total number of vectors used in the approximation space, due to the fact that converged vectors have been removed from the iteration blocks (see discussion in section 6.2.4).

Given these results, all subsequent calculations using the block CG method are performed with a block size of 8.

8.3.2 Effect of Preconditioner & Initial Guess

In section 6.2.4 it was shown theoretically that the factorization of the nominal stiffness matrix can be a very good preconditioner for the CG iterations on the modified stiffness matrix. In this subsection this will be illustrated by comparing the results obtained using other preconditioners. Furthermore the influence of a different initial guess will be shown.

Two alternative preconditioners are considered here, namely the Jacobi preconditioner and the Symmetric Successive Over Relaxation (SSOR) [63, 12]. Many other preconditioners exist, but these have been chosen because they can be obtained at virtually no computational cost. Both preconditioners can be defined by decomposing the stiffness matrix into a diagonal part $\boldsymbol{D}$ and a strictly lower diagonal part $\boldsymbol{E}$. Due to symmetry of the stiffness matrix one can then write:

$$\boldsymbol{K} = \boldsymbol{D} + \boldsymbol{E} + \boldsymbol{E}^T \tag{8.3}$$

The Jacobi preconditioner is defined simply as $\boldsymbol{S}_J = \boldsymbol{D}$. The SSOR preconditioner is somewhat more complex and defined by:

$$\boldsymbol{S}_S = \left(\boldsymbol{D} + \boldsymbol{E}\right) \boldsymbol{D}^{-1} \left(\boldsymbol{D} + \boldsymbol{E}^T\right) \tag{8.4}$$

In order to efficiently apply this preconditioner in the CG iterations, one can observe that it can be written in the form of an LU-decomposition [178]:

$$\boldsymbol{S}_S = \boldsymbol{L}\boldsymbol{U} \quad \text{with} \quad \begin{cases} \boldsymbol{L} \equiv \left(\boldsymbol{D} + \boldsymbol{E}\right) \boldsymbol{D}^{-1} = \boldsymbol{I} + \boldsymbol{E}\boldsymbol{D}^{-1} \\ \boldsymbol{U} \equiv \boldsymbol{D} + \boldsymbol{E}^T \end{cases} \tag{8.5}$$

Indeed, this is an approximation of the LU decomposition of the stiffness matrix:

$$\boldsymbol{S}_S = \boldsymbol{D} + \boldsymbol{E} + \boldsymbol{E}^T + \boldsymbol{E}\boldsymbol{D}^{-1}\boldsymbol{E}^T = \boldsymbol{K} + \boldsymbol{E}\boldsymbol{D}^{-1}\boldsymbol{E}^T \tag{8.6}$$

This preconditioner can be applied like any other factorized preconditioner.

To quantify the effect of using different preconditioners, the 125k DoF model is subjected to design modification case 1, 1% scaling in (x, z) directions. Larger models and design modifications could not be tested due to the memory requirements for storing the CG iterates. The results are listed in table 8.5.

Preconditioner	Initial	Tol.	CPU time [s]	# iterations	# vectors	Rel. CPU [–]
$\boldsymbol{K}_{(0)}$	$\boldsymbol{\Psi}_{(0)}$	10^{-6}	10	12	82	0.7
$\boldsymbol{K}_{(0)}$	$\boldsymbol{0}$	10^{-6}	16	20	131	1.2
SSOR	$\boldsymbol{\Psi}_{(0)}$	10^{-4}	374	396	1178	28
SSOR	$\boldsymbol{0}$	10^{-4}	2718	875	2526	209
Jacobi	$\boldsymbol{\Psi}_{(0)}$	10^{-4}	3017	1095	3309	233
None	$\boldsymbol{\Psi}_{(0)}$	10^{-4}	4982	1238	4970	384

Table 8.5: Results for static modes using different preconditioners and initial guess, for case 1 – scaling by 1%.

Again, a few remarks can be made based on these results:

- Using the factorization of the nominal stiffness matrix for preconditioning is far more efficient than using the Jacobi or SSOR preconditioners, or no preconditioning at all. SSOR preconditioning is relatively more efficient than Jacobi.
- A good initial guess can really decrease the computation time since the iterations are started with a much smaller initial residual. However, the preconditioner thereafter determines the actual rate of convergence.
- In order to be able to complete the CG iterations with the available memory space, the tolerance ε for the CG iterations had to be changed from 10^{-6} to 10^{-4} when using the Jacobi and SSOR preconditioners.
- Although all computations have been performed with block size 8, the average blocksize with the Jacobi and SSOR preconditioning is much smaller. This indicates that some static modes converge much faster than others.

Finally it is noted that, at least theoretically, the Jacobi and SSOR preconditioners should become relatively more competitive when the design change increases since they are based on $\boldsymbol{K}_{(1)}$ and hence independent of the design modification. However, the initial guess is worse in those cases which again requires many additional iterations. The net effect is therefore hard to quantify.

8.3.3 Effect of Design Modification & Model Size

The most important criterion for the practical applicability of the updating strategy is its computational cost compared to that of direct methods for realistic design modifications. In order to assess this, the three FE models are subjected to the series of design changes described in section 8.2. The results are presented in figure 8.4 where the relative CPU time is plotted as a function of the intensity of the design change, expressed by δ_K as defined in eq. (8.1). The relative CPU time is defined as the actual CPU time divided by the CPU time needed for the direct solver (see table 8.3).

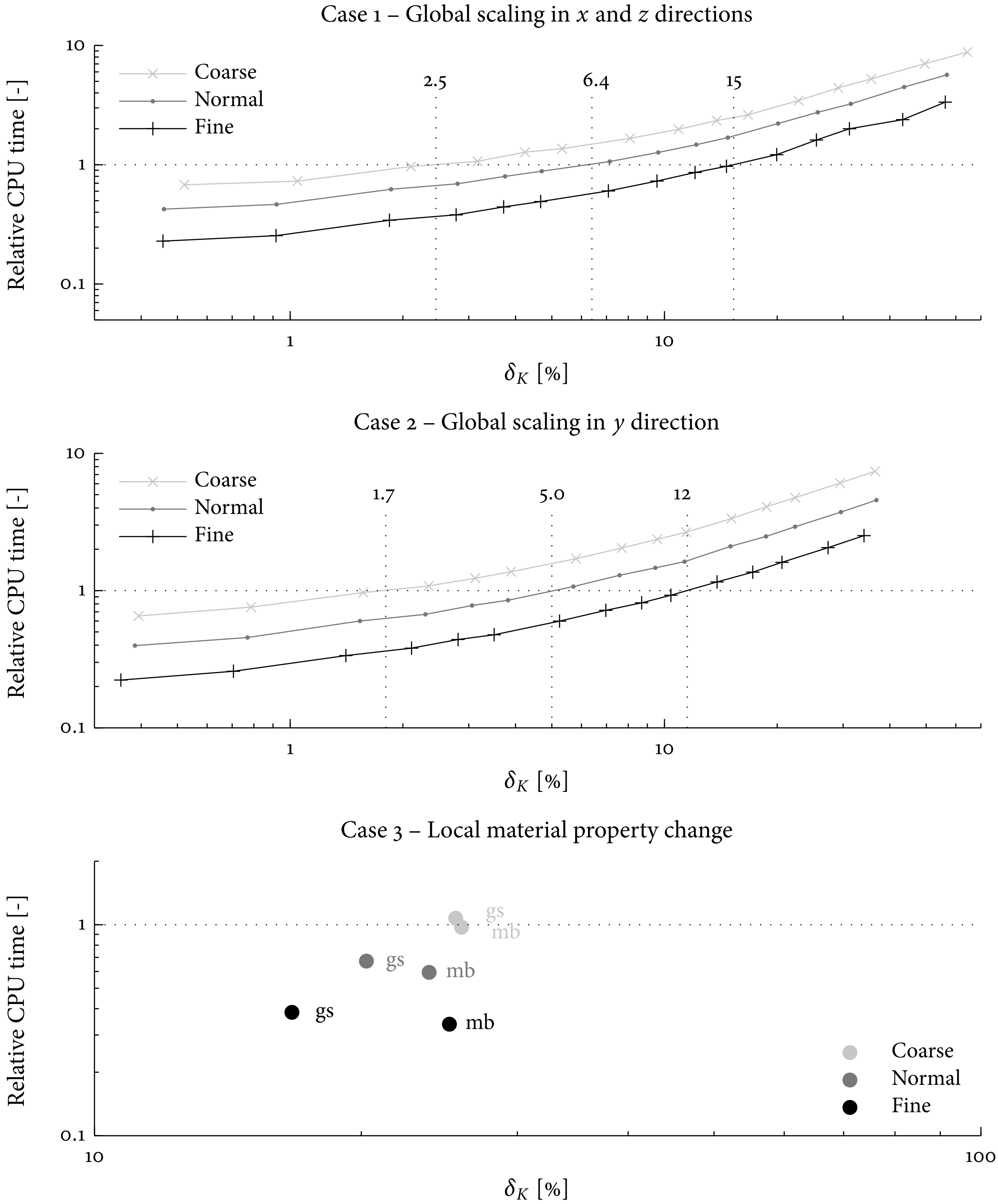
Furthermore, for each of the three design change cases one representative variant was chosen for which detailed results are provided in table 8.6. For cases 1 and 2, a global scaling of 5% was deemed realistic, while for case 3 the material change of the bearing housing is considered.

Design change		δ_K [%]	CPU [s]	# iterations	# vectors	Rel. CPU [–]
Case 1 – 5%	Coarse	5.3	21	24	156	1.40
	Normal	4.7	54	24	158	0.89
	Fine	4.7	144	24	157	0.49
Case 2 – 5%	Coarse	3.9	21	24	156	1.40
	Normal	3.8	52	24	159	0.85
	Fine	3.5	140	24	157	0.48
Case 3 – mb	Coarse	26	15	20	20	1.00
	Normal	23.9	36	18	114	0.59
	Fine	25.2	99	18	114	0.34

Table 8.6: Static modes updating results for three realistic cases.

Based on figure 8.4 and table 8.6, a number of observations and remarks can be made:

- From the results of cases 1 and 2, it appears that a somewhat linear relation seems to exist between δ_K and the CPU time, regardless of the model size.
- With increasing model size the updating approach becomes relatively more efficient. This is due to the fact that factorization of the sparse stiffness matrix takes in the order of $n \cdot b^2$ floating point operations, where n is the model size and b the matrix' diagonal bandwidth [77]. Backsubstitution requires an additional $n \cdot b$ operations for each of the p static modes, leading to a total of the order of $n \cdot b^2 + n \cdot b \cdot p$ operations for the direct solver. From table 8.6 it is seen that the number of CG iterations needed for convergence is independent of the model size. Hence the number of matrix-vector multiplications, each requiring of the order of $n \cdot b$ operations, is constant and the total number of operations is of the order of $n \cdot b \cdot p \cdot m$, where m is the number of iterations. In


Case 1 – Global scaling in x and z directions

Case 2 – Global scaling in y direction

Case 3 – Local material property change

Figure 8.4: Relative CPU time versus intensity of design modification for static modes updating.

both cases the number of operations, and hence the computation time, scales linearly with the model size n. However, the matrix bandwidth b also increases with a finer mesh. Since the computational cost of the direct solver depends quadratically on b while the CG solver's cost depends only linearly, it is clear that for larger models the iterative method becomes more and more attractive.

- For the global modifications, the plots in figure 8.4 can be used to determine the "break-even" points, i.e. the δ_K values and/or scaling factors for which the CPU time of the iterative CG solver is equal to that of the direct solver (relative CPU time equal to 1). This is indicated by the dotted lines in the plots in figure 8.4.

- Local modifications (see bottom plot in fig. 8.4) can be handled much more efficiently then global changes even though the corresponding δ_K values are larger. In fact, for the two local modifications tested here, the updating approach is always more efficient then the direct method. This is believed to be due to the fact that the design change is performed "far away" from the boundary DoF, thereby barely affecting the more localized static deformation shapes.
- In practice a design modification where the structure's dimensions are scaled by 50% is not realistic. In that case one either creates a new model or sequentially applies smaller design changes of for instance 5 to 10% scaling. As was described in section 6.4, in the latter case one can use the updating approach combined with previous CG iterates to enhance the preconditioning. This approach is studied in section 8.6.

8.4 Efficiency of Updating of Vibration Modes

In this section the attention is turned to the updating of the vibration modes of the bedframe model. Since in this chapter the Craig-Bampton reduction basis is updated, the so-called fixed interface vibration modes are considered, see section 2.4.3. It was chosen to compute 30 vibration modes for the reduction basis, which is a reasonable number in practice. The vibration modes for the nominal reduction basis are computed using the (compiled) implementation of the Lanczos method that is included in the SD Toolbox for Matlab [17]. The analysis times needed for the computation of 30 modes are shown in table 8.7, this excludes the time needed for the factorization of the stiffness matrix.

	Coarse	Normal	Fine
Lanczos eigensolution [s]	40	100	282

Table 8.7: CPU times for calculation of 30 vibration modes of the nominal bedframe models, excluding factorization of the stiffness matrix.

For the modified models described in section 8.2 the aim is to compute the vibration modes using the IFPKS method with an implementation of algorithm 6.5. Similar to the previous section, this section subsequently addresses the effect of the algorithm settings, the preconditioner and initial guess and finally the design modification and model size. All calculations are performed with an iteration tolerance of 10^{-3} on the residual, corresponding to a tolerance of approximately 10^{-6} on the eigenfrequencies.

8.4.1 Effect of Algorithm Settings

In the IFPKS method a number of algorithm settings need to be chosen, namely:

- The iteration block size b, i.e. the number of vectors in the blocks $\boldsymbol{Z}_j$ in algorithm 6.5
- The Krylov order, i.e. the number of inner iterations r in algorithm 6.5
- Whether or not to enrich the search space with difference vectors (see section 6.3.3.4)

In contrast to the CG algorithm, where the block size only affects computational efficiency, the settings of the IFPKS algorithm determine how the problem is solved and can hence strongly influence the convergence of the algorithm. Furthermore, it is expected that for different design change magnitudes these parameters might have different effects. Quantification of these effects therefore requires a parameter study, which is carried out on the coarse FE model subject to the design modifications of case 1. For this study, the Krylov order r was varied from 1 to 5, while block sizes b of [1, 2, 4, 6, 8, 10, 15, 30] were considered. In all cases preconditioning was done using the factorization of $\boldsymbol{K}_{(0)}$, the nominal modes $\boldsymbol{\Phi}_{(0)}$ were taken as initial guess and deflation was used to obtain the 30 modes.

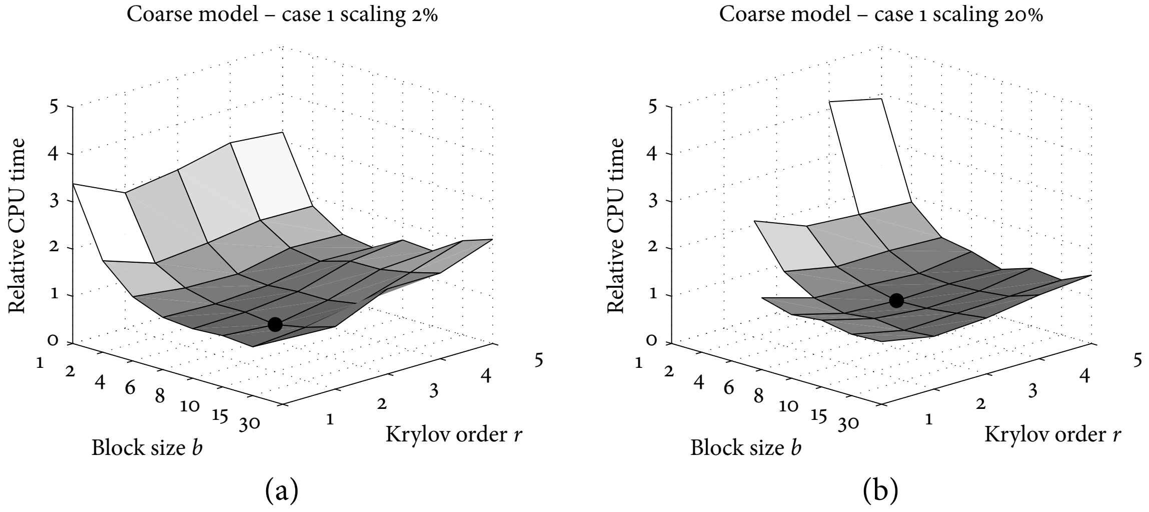
To limit the amount of data presented here, table 8.8 lists, for each design change of case 1, the block size and Krylov order settings of the IFPKS method that give the lowest CPU time. This is done with the IFPKS algorithm set to run both with and without difference vectors added to the basis; the CPU time in column 6 is relative to the lowest CPU time without difference vectors.

Due to the fact that different parameter values are shown, the number of outer iterations needed for convergence cannot be compared. Instead, table 8.8 lists the product of the number of outer iterations k, the number of inner iterations (i.e. Krylov order) r and the block size b. This number is equal to the total number of forward/backward substitutions due to application of the preconditioner, the computationally most expensive step in the iterations, and is hence a good indication of the computational cost.

Variant/ Scaling		w/o difference vectors			w/ difference vectors			
		$b \times r \times k$	b	r	Relative CPU [–]	$b \times r \times k$	b	r
1	0.5%	108	4	3	1.02	108	4	3
2	1%	160	10	2	**0.95**	144	8	3
3	2%	180	10	2	1.06	180	10	2
4	3%	208	8	2	**0.96**	150	15	1
5	4%	240	8	2	1.02	240	15	2
6	5%	256	8	2	1.02	240	10	2
7	7.5%	330	15	2	**0.91**	272	8	2
8	10%	408	8	3	1.01	360	15	2
9	12.5%	480	8	3	**0.92**	390	15	2
10	15%	585	15	3	**0.94**	510	10	3
11	20%	760	10	4	**0.93**	648	8	3
12	25%	920	10	4	**0.90**	800	8	4
13	30%	1125	15	5	**0.89**	928	8	4
14	40%	1600	8	5	**0.86**	1320	8	5
15	50%	2325	15	5	**0.75**	1620	15	4

Table 8.8: IFPKS parameter settings for design changes in case 1 for lowest CPU times. With k the number of outer iterations, b the block size and r the Krylov order.

Furthermore, for the practical usability of the IFPKS method the computational efficiency should not be too sensitive to the algorithm settings. Therefore, figure 8.5 shows the variation of the CPU time with the algorithm settings for two design modifications. The CPU time is normalized with respect to the lowest CPU time for that case, the corresponding settings are indicated by the black dots. In plot (b) some CPU time data is missing since for those settings the algorithm could not find all 30 requested modes in the set maximum number of outer iterations, which was set to $\sqrt{n} = 351$.



(a) (b)

Figure 8.5: Relative CPU time versus block size and Krylov order.

Based on the results in table 8.8 and figure 8.5, the following is observed:

- No single combination of algorithm settings gives the best results across the full range of scaling factors. However, small block sizes ($b < 8$) and low Krylov orders ($r < 2$) seem to be inefficient in all cases. In most cases, a block size of 8, 10 or 15 combined with a Krylov order of 2 to 4 gives the best results. Figure 8.5 shows that the efficiency of the IFPKS method does not change dramatically with slight variations of the settings.

- The use of difference vectors incurs some additional computational cost at each outer iteration but generally leads to (much) faster convergence, as can be seen from the product $b \times r \times k$. Especially for larger design changes this outweighs the additional cost, giving a net decrease in the CPU times of over 10%. For small design changes the benefit is less pronounced.

- Larger design modifications require larger approximation bases, either by increasing the number of inner iterations (Krylov order) or the block size. Trivially, increasing the Krylov order leads to less outer iterations. With the outer iterations becoming more expensive, the challenge is therefore to find the right balance. Furthermore, it was shown in section 6.3.4.1 that the preconditioning is most effective when the frequency shift ρ_k is close to the desired eigenfrequency ω_i^2; when $\rho_k \ll \omega_i^2$ it is more efficient to update the shift by solving the interaction problem. Therefore, to optimize its convergence properties one might consider adaptively choosing the Krylov order during the IFPKS iterations using some heuristic scheme based on the current residuals. This is however out of the scope of this work.

Given these results and observations, future calculations are performed using a blocksize of 10, Krylov order 3 and use of difference vectors enabled.

8.4.2 Effect of Preconditioner & Initial Guess

In this subsection the effect is investigated of extending the preconditioning using the CG iterates (see section 6.3.4.2) and enhancing the initial guess using sensitivity modes (see section 6.3.4.3). To this end, the coarse FE model is again subjected to the design changes of case 1. Having updated the static modes and obtained the sensitivity modes, four cases are considered for each updating problem and the results listed in table 8.9:

1. Standard preconditioning, standard initial guess: columns 3
2. Standard preconditioning, enhanced initial guess: columns 4-5
3. Extended preconditioning, standard initial guess: columns 6-7
4. Extended preconditioning, enhanced initial guess: columns 8-9

For each design modification table 8.9 shows the relative CPU time for the calculations using extended preconditioning and/or enhanced initial guess with the standard preconditioning and initial guess taken as the reference case. Since all calculations are performed with the same block size and Krylov order, the total number of outer iterations is also listed.

Preconditioning:		Std. – $\boldsymbol{K}_{(0)}$	Std. – $\boldsymbol{K}_{(0)}$		Ext. – $\boldsymbol{K}_{(0)}, \boldsymbol{P}$		Ext. – $\boldsymbol{K}_{(0)}, \boldsymbol{P}$	
Initial guess:		Std. – $\boldsymbol{\Phi}_{(0)}$	Enh. – $\boldsymbol{\Phi}_{(0)}, \bar{\boldsymbol{\Phi}}_{(1)}$		Std. – $\boldsymbol{\Phi}_{(0)}$		Enh. – $\boldsymbol{\Phi}_{(0)}, \bar{\boldsymbol{\Phi}}_{(1)}$	
Variant	Scaling	# its.	Rel. CPU	# its.	Rel. CPU	# its.	Rel. CPU	# its.
1	0.5%	5	**0.91**	4	1.03	5	0.93	4
2	1%	6	**0.76**	4	1.19	7	0.79	4
3	2%	9	**0.53**	4	0.91	8	0.55	4
4	3%	9	**0.75**	6	1.03	9	0.78	6
5	4%	10	0.78	7	1.05	10	**0.71**	6
6	5%	11	**0.70**	7	1.06	11	0.73	7
7	7.5%	14	**0.69**	9	0.98	13	0.74	9
8	10%	15	0.86	12	1.01	14	**0.78**	10
9	12.5%	19	0.78	14	0.87	15	**0.73**	12
10	15%	22	0.81	17	0.86	17	**0.64**	12
11	20%	29	0.86	24	0.79	20	**0.65**	16
12	25%	37	0.86	31	0.66	21	**0.52**	16
13	30%	47	0.87	40	0.58	23	**0.49**	19
14	40%	60	0.91	54	0.62	30	**0.51**	24
15	50%	78	0.94	72	0.60	36	**0.49**	34

Table 8.9: Results for vibration modes using different preconditioners and initial guess. Abbreviations: "Std." means standard, "Ext." extended and "Enh." enhanced.

Again some conclusions can be drawn from these results:

- Enhancing the initial guess using sensitivity vectors reduces the CPU time in all cases. However, its effect is most pronounced for small to modest design changes (scaling $<$ 10%), where it can help reducing the number of outer iterations (and hence the computational cost) by up to 40%. Since the sensitivity vectors are based on a first order perturbation analysis, they lose their effectiveness for larger design modifications.

However, even when scaling the structure by 50% the sensitivity vectors help reducing the computation time by over 5%.

- The extended preconditioning using CG iterates seems to be efficient only for larger design modifications (scaling > 10%). In those cases the increased convergence rate overcomes the additional computational effort associated to the preconditioning step (see section 6.3.4.2). This shows that for smaller design changes the information contained in the CG iterates is not very relevant for preconditioning the eigenproblem. This can be expected as the modified stiffness matrix is still close to the nominal one. As a result one already has a good preconditioner and the CG iterates mainly contain "local" information. With increasing $\Delta \boldsymbol{K}$ this is less and less the case and the extended preconditioning becomes more effective, requiring only half the number of outer iterations or 40% less CPU time when the structure is scaled by 50%.
- Combining the enhanced initial guess and extended preconditioning gives good results across the full range of design modifications. At small design modifications this is slightly less efficient than using only the enhanced initial guess, but for larger design changes there is a substantial benefit. Therefore, unless otherwise noted, the results presented in the remainder of this chapter are obtained using both techniques.

8.4.3 Effect of Design Modification & Model Size

Now the settings and ingredients for the IFPKS method have been established, the remaining question is how its computational efficiency compares to that of traditional eigensolvers when considering the updating problem. To answer this question the IFPKS method is used to update the vibration modes for all modified models described in section 8.2, for each of the three FE models. The results are plotted in figure 8.6; the CPU times are normalized for each model by the corresponding CPU time needed for full recomputation, as listed in table 8.7. Similar as for the updating of the static modes, more detailed results are listed in table 8.10 for one realistic variant of each of the three design change cases.[1]

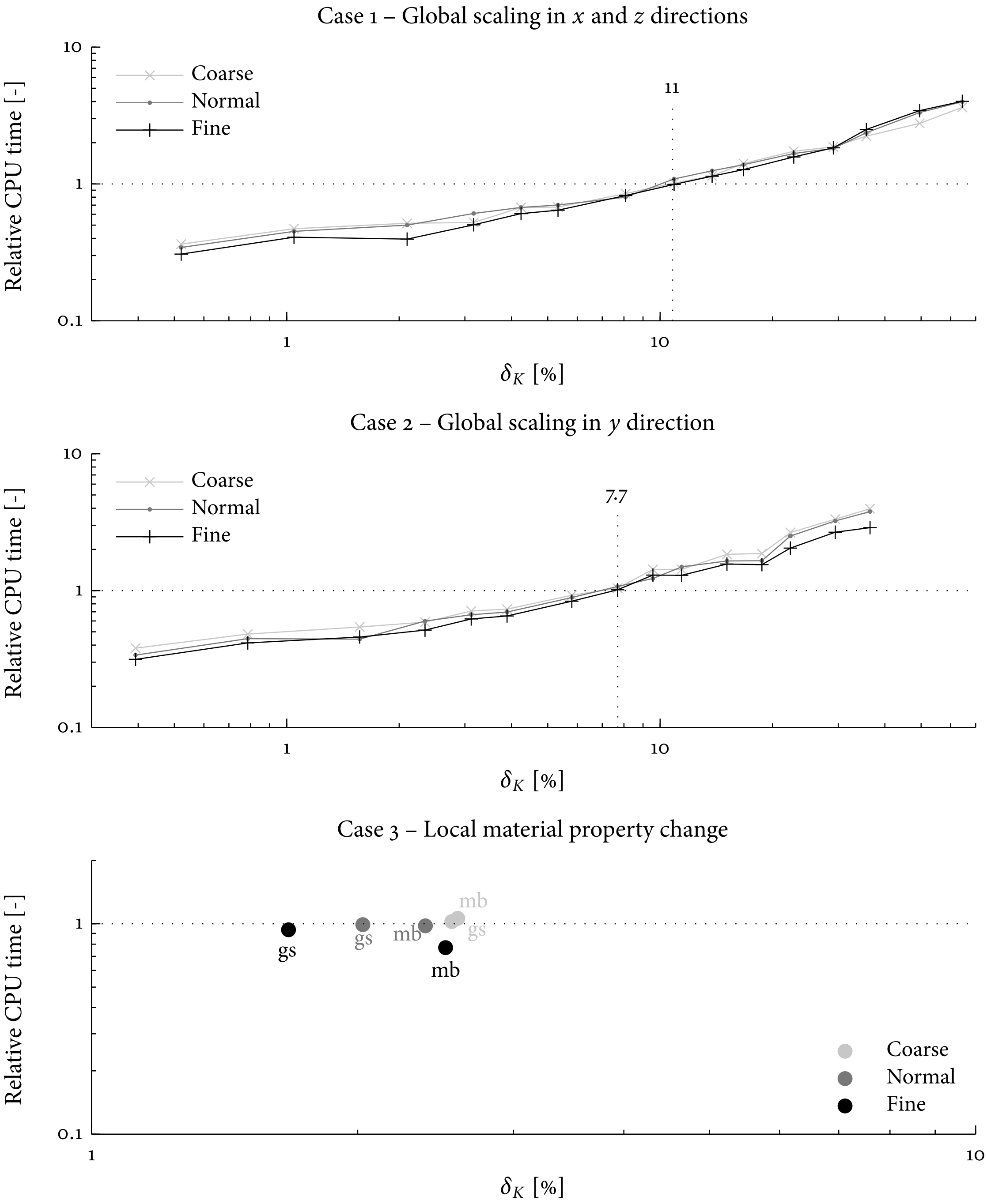
Design change		δ_K [%]	δ_M [%]	CPU [s]	# outer its	Rel. CPU [–]
Case 1 – 5%	Coarse	5.3	10.3	27	7	0.67
	Normal	4.7	10.3	70	7	0.70
	Fine	4.7	10.3	181	7	0.64
Case 2 – 5%	Coarse	3.9	5.0	29	8	0.73
	Normal	3.8	5.0	69	7	0.69
	Fine	3.5	5.0	184	7	0.65
Case 3 – mb	Coarse	26.0	24.3	41	11	1.03
	Normal	23.9	28.9	99	10	0.99
	Fine	25.2	27.5	264	12	0.94

Table 8.10: Vibration modes updating results for three realistic cases.

The following can be noted with respect to these results:

---

[1] Note that since the FE models are meshed with elements having only translational DoF, the δ_M values in table 8.10 directly indicate the global mass change of the structure. In case 1 the volume of the structure has scaled by $1.05 \times 1.05 \times 1$ and hence $\delta_M = 10.25\%$.


Case 1 – Global scaling in x and z directions

Case 2 – Global scaling in y direction

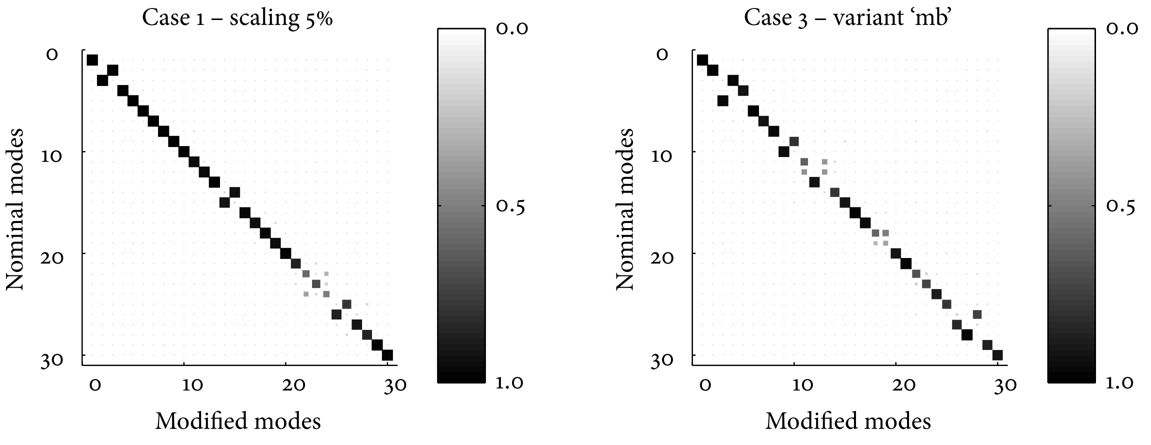
Case 3 – Local material property change

Figure 8.6: Relative CPU time versus intensity of design modification for vibration modes updating.

- In contrast to the static modes, the efficiency of the vibration modes updating is independent of the model size. This is due to the efficiency of the reference calculations with the Lanczos method being independent of the system size (see section 6.3.1), but only dependent on the spectral distribution of the eigenvalue problem (i.e. the underlying physics of the problem). These properties are more or less independent of the mesh size. This is reflected in the number of outer iterations which is equal for the different mesh sizes, see table 8.10.

- For the global modifications, the plots in figure 8.6 can be used to determine the "break-even" points, i.e. the δ_K values and scaling factors for which the relative CPU time of the IFPKS solver is equal to unity. In these plots the break-even points are indicated by the dotted lines.

- Again in contrast to the static modes, local modifications are not necessarily handled more efficiently then global design changes. Where the static modes are not really influenced by the local modification, the (fixed interface) vibration modes are much more sensitive to such changes. In figure 8.7 this is illustrated by comparing the modal assurance criterion (see section 4.7.2) values calculated for the nominal fixed interface modes and the modified modes, for both a local change and a global modification. As can be seen, for the local design change the modified modes are relatively more uncorrelated with the nominal modes.



(a) (b)

Figure 8.7: MAC of modified vs. nominal vibration modes for global (a) and local (b) design change.

8.5 Overall Efficiency of Basis Updating

By combining the results of the updating of the static and vibration modes, the overall efficiency of the basis updating strategy can be determined. Figure 8.8 shows the CPU time for the updating of both the static and vibration modes normalized by the total CPU time needed for recomputation of the reduction bases, i.e. the sum of the CPU times in tables 8.3 and 8.7. Furthermore, table 8.11 lists the results for updating the three FE models subject to the selected variants of the three design change cases.

As before, the plots indicate the break-even points for cases 1 and 2. It can be seen that in general the updating strategy becomes more efficient with increasing model size and hence is mainly attractive for large FE models. In this example for the fine FE model, the updating strategy outperforms recomputation for δ_k values up to 14%. For the same fine FE model subject to the three representative cases listed in table 8.11, updating of the reduction bases is approximately 40% faster than recomputation. Obviously, both when larger FE models or smaller design changes are considered the updating method becomes more and more efficient.

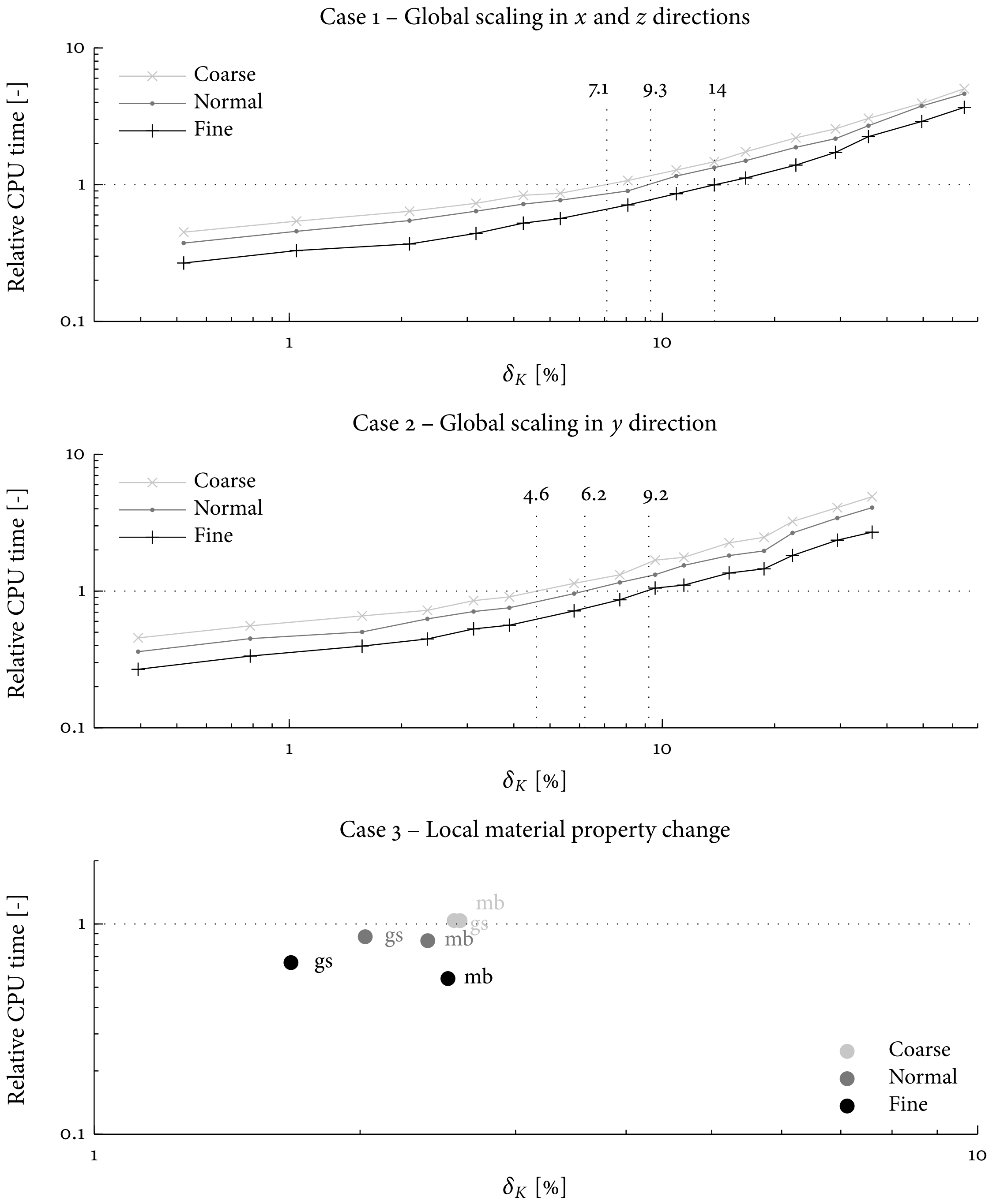


Figure 8.8: Relative CPU time versus intensity of design modification for total basis updating.

8.6 Sequential Updating

In the previous two sections the situation was considered where the updating of the reduction basis was performed with respect to the nominal component. In practice however design changes are often applied sequentially, where a component undergoes a series of design modifications from the nominal to the final design. As was explained in section 6.4, in this case not only the information from the nominal component maybe useful for preconditioning the updating algorithms but also the CG iterates from the previous update(s). This is especially true if the new component design is closer to the previous modification than to

Mesh	Case	CPU recomputation [s]		CPU updating [s]		Relative CPU [–]
		Static	Vibration	Static	Vibration	
Coarse	Case 1 – 5%	15	40	21	27	0.87
	Case 2 – 5%			21	29	0.91
	Case 3 – mb			15	41	1.02
Normal	Case 1 – 5%	61	100	54	70	0.77
	Case 2 – 5%			52	70	0.76
	Case 3 – mb			36	99	0.84
Fine	Case 1 – 5%	293	282	144	181	0.57
	Case 2 – 5%			140	184	0.56
	Case 3 – mb			99	264	0.63

Table 8.11: Efficiency of basis updating vs. recomputation for three realistic cases.

the nominal design.

In order to test the effectiveness of the extended preconditioning scheme described in section 6.4, this section treats the sequential updating of the bedframe FE models. To this end each modified model is not updated with respect to the nominal model, but with respect to the previous updated model. As before, the design changes are performed in increasing order of magnitude. Hence, when for example updating the modified model of case 1 with 10% scaling, this is done with respect to the updated model of case 1 with 7.5% scaling. This latter model was updated as before, i.e. directly with respect to the nominal model. For the preconditioning of the 10% modified model then both the factorized nominal stiffness matrix and the CG iterates from the 7.5% update are employed; the corresponding updated modes (static or dynamic, depending on which are computed) are used as initial guess. This procedure is repeated for the next model, so the 12.5% scaled model is updated with respect to the updated 10%, which in turn was updated with respect to the nominal model. This approach is illustrated in figure 8.9.

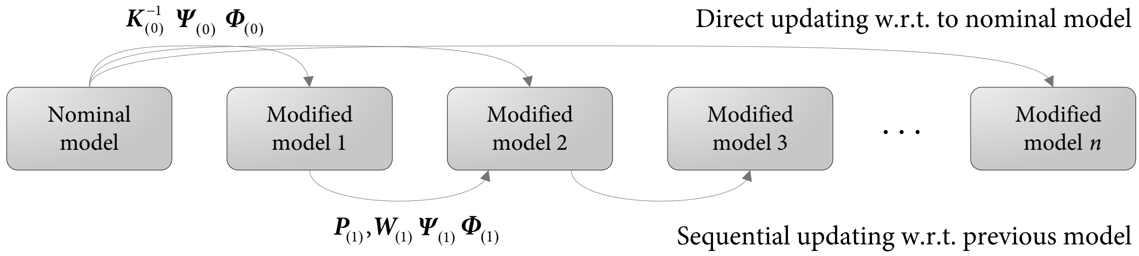

$\boldsymbol{K}_{(0)}^{-1}\ \boldsymbol{\Psi}_{(0)}\ \boldsymbol{\Phi}_{(0)}$
Direct updating w.r.t. to nominal model
Nominal model
Modified model 1
Modified model 2
Modified model 3
. . .
Modified model n
$\boldsymbol{P}_{(1)}, \boldsymbol{W}_{(1)}\ \boldsymbol{\Psi}_{(1)}\ \boldsymbol{\Phi}_{(1)}$
Sequential updating w.r.t. previous model

Figure 8.9: Schematic representation of direct and sequential updating procedures.

Another approach would be to apply *all* design modifications sequentially, meaning that the modified model of case 1 with 10% scaling would be updated using the CG iterates from all preceding updated models (i.e. those with 7.5%, 5%, 4%, ..., scaling). This would however lead to so many CG iterates that the extended preconditioning would lose its effectiveness. Selection of the most relevant content then becomes essential, as was indicated in section 6.4. This is however out of the scope of this work.

Next, the sequential updating as described above is applied to the static modes using the CG method; thereafter, subsection 8.6.2 treats sequential updating of the vibration modes. Note that for the sake of brevity here only the design changes of case 1 are considered.

8.6.1 Sequential Updating Applied to Static Modes

The sequential updating of the static modes has been performed as follows. First, each variant of the design changes in case 1 is updated with respect to the nominal model (this will be called the "direct update"). As was done before, the nominal static modes $\boldsymbol{\Psi}_{(0)}$ are used as initial guess and the factorization $\boldsymbol{K}_{(0)}^{-1}$ is used for preconditioning. As a result the updated static modes $\boldsymbol{\Psi}_{(1)}$ as well as the CG iterates $\boldsymbol{P}_{(1)}, \boldsymbol{W}_{(1)}$ are obtained for each variant. Next, variants 2-15 are updated with respective to the associated previous variant (i.e. variants 1-14, this will be called the "sequential update"). Thereby the updated static modes $\boldsymbol{\Psi}_{(1)}$ are taken as initial guess; preconditioning is achieved according to eq. (6.117) in section (6.4) using both $\boldsymbol{K}_{(0)}^{-1}$ and the CG iterates $\boldsymbol{P}_{(1)}, \boldsymbol{W}_{(1)}$. This sequential updating is applied to all three FE models while all calculations are performed with a blocksize of 8 and an iteration tolerance of $\varepsilon = 10^{-6}$. This is summarized in table 8.12.

	Direct update	Sequential update
Variants considered:	$1 \dots 14$	$k = 2 \dots 15$
Initial guess:	$\boldsymbol{\Psi}_{(0)}$	$\boldsymbol{\Psi}_{(k-1)}$
Preconditioning:	$\boldsymbol{K}_{(0)}^{-1}$	$\boldsymbol{K}_{(0)}^{-1}, \left(\boldsymbol{P}_{(k-1)}, \boldsymbol{W}_{(k-1)}\right)$
Output:	$\boldsymbol{\Psi}_{(1)}$	$\boldsymbol{\Psi}_{(k)}$

Table 8.12: Summary of sequential updating procedure for static modes.

As was established in section 8.3, the convergence rate of the CG algorithm is independent of the problem size. Hence the efficiency of the extended preconditioning is independent of the size of the FE model used. The results listed in table 8.13 are therefore true for all three FE models. In columns three and four, this table shows the number of iterations and vectors needed for the direct updating. These results are equal to those used to generate the figures in section 8.3.3. Columns seven and eight list the number of iterations and vectors needed for the sequential update. Relative CPU times are listed in column six, which denote the ratio between the CPU time of the sequential update over that of the direct update of the same modified model. Finally, column five lists the relative δ_K value between two sequential design modifications. This value signifies the "distance" between the two components that has to be "bridged" by the CG algorithm.

From table 8.13 it is clear that in all cases the number of iterations, number of vectors and the CPU time needed for convergence are lower when performing the update relative to the previous component than directly from the nominal one. Since the design modifications are ordered with increasing magnitude, the relative δ_K values are always lower than the δ_K value with respect to the nominal component. This result should therefore not come as a surprise.

Still, the gain in CPU time can be approximately a factor two for large design changes. These results are reminiscent of those for the extended preconditioning scheme for IFPKS method in section 8.4.2: despite the additional computational effort associated with the extended preconditioning the increased convergence rate proves very effective. In conclusion, when

Variant/ Scaling		Direct update		Sequential update			
		# its.	# vecs.	Rel. δ_K [%]	Rel. CPU [−]	# its.	# vecs.
1	0.5%	11	75	–	–	–	–
2	1%	12	83	0.5	0.93	11	75
3	2%	17	111	1.0	0.89	15	95
4	3%	18	123	1.0	0.80	14	93
5	4%	21	146	1.1	0.71	14	96
6	5%	24	156	1.1	0.65	14	98
7	7.5%	29	191	2.7	0.73	20	134
8	10%	33	225	2.7	0.66	23	140
9	12.5%	39	258	2.7	0.61	23	155
10	15%	42	291	2.7	0.58	26	162
11	20%	53	354	5.4	0.61	33	215
12	25%	63	420	5.4	0.57	36	228
13	30%	69	483	5.4	0.52	38	249
14	40%	87	609	10.9	0.54	50	324
15	50%	107	741	10.7	0.50	54	362

Table 8.13: Results of sequential updating of the static modes.

CG iterates from previous updates are available, and the relative "distance" to the previous modified model is closer than to the nominal one, it is always beneficial to reuse the available CG iterates.

8.6.2 Sequential Updating Applied to Vibration Modes

In this subsection sequential updating is applied to the vibration modes. To this end, first the static modes are sequentially updated according to the above described procedure. A very similar procedure is thereafter followed for the vibration modes, which is best summarized by the overview in table 8.14.

	Direct update	Sequential update
Variants considered:	$1\dots14$	$k = 2\dots15$
Initial guess:	$\boldsymbol{\Phi}_{(0)}, \bar{\boldsymbol{\Phi}}_{(1)}$	$\boldsymbol{\Phi}_{(k-1)}, \bar{\boldsymbol{\Phi}}_{(k-1)}$
Preconditioning:	$\boldsymbol{K}_{(0)}^{-1}, \left(\boldsymbol{P}_{(1)}, \boldsymbol{W}_{(1)}\right)$	$\boldsymbol{K}_{(0)}^{-1}, \left(\boldsymbol{P}_{(k-1)}, \boldsymbol{W}_{(k-1)}\right), \left(\boldsymbol{P}_{(k)}, \boldsymbol{W}_{(k)}\right)$
Output:	$\boldsymbol{\Phi}_{(1)}$	$\boldsymbol{\Phi}_{(k)}$

Table 8.14: Summary of sequential updating procedure for vibration modes.

In this table $\bar{\boldsymbol{\Phi}}_{(1)}$ denotes the sensitivity modes also used in section 8.4.2. Note that all calculations with the IFPKS algorithm are performed with blocksize 10, Krylov order 2, and the use of difference vectors enabled. As before, the iteration tolerance is set to $\varepsilon = 10^{-3}$ on the residual.

Table 8.15 shows the results obtained from the sequential updating. Similar to the static modes, the convergence of the IFPKS method is independent of the model size, such that one set of results can be shown that is applicable to all three FE models. In column three of table 8.15 the number of iterations is listed for the direct updating with respect to the nominal model. Since the block size and Krylov order are equal in both cases, it is irrelevant to also specify the $b \times r \times k$ number as in table 8.8. The number of iterations of the sequential update is listed in column six. Like before, column four lists the relative δ_K value while column five shows the relative CPU times between the direct and sequential update.

Variant/ Scaling		With respect to nominal	With respect to previous		
		# its.	Rel. δ_K [%]	Rel. CPU [–]	# its.
1	0.5%	4	–	–	–
2	1%	4	0.5	0.78	3
3	2%	4	1.0	0.82	4
4	3%	6	1.0	0.70	4
5	4%	6	1.1	0.69	4
6	5%	7	1.1	0.77	5
7	7.5%	9	2.7	0.84	8
8	10%	10	2.7	0.79	7
9	12.5%	12	2.7	0.71	7
10	15%	12	2.7	0.72	8
11	20%	16	5.4	0.75	11
12	25%	16	5.4	0.75	11
13	30%	19	5.4	0.73	13
14	40%	23	10.9	0.73	16
15	50%	23	10.7	0.81	17

Table 8.15: Results of sequential updating of the vibration modes.

Again, from table 8.15 it can be seen that for all sequential modifications considered here it is more efficient to perform the update with respect to the previous model than from the nominal one. Given the findings from section 8.4.2 where more or less the same preconditioning scheme was tested, this is again no surprise.

In contrast to the static modes however, it seems that for the vibration modes the relative gain is independent of the magnitude of the design change. This could be due to the fact that for large design changes the benefit of the increased convergence rate is partly offset by the additional cost of the extended preconditioning. Since in the IFPKS scheme now both the CG iterates $\boldsymbol{P}_{(1)}$ and $\boldsymbol{P}_{(2)}$ are used for preconditioning according to eq. (6.118), this can become quite costly indeed.

Nonetheless, computation times can be reduced by about 30% for the cases tested here without any additional computational effort. Hence, it is again concluded that when relevant CG iterates are available from previous updates, it is always advisable to apply these in the extended preconditioning scheme to increase the convergence rate.

8.7 Effectiveness of Different Reduction Bases

Up to now the focus has been on the computational efficiency of the updating method. In order to illustrate the use of the method, this section compares reduced models of a modified bedframe obtained in several ways, namely:

1. Application of the nominal Craig-Bampton reduction basis to the modified component model.

2. Recomputation of the complete Craig-Bampton basis for the modified component.

3. Enrichment of the nominal Craig-Bampton basis by the modal sensitivity vectors introduced in section 6.3.4.3, leading to the Enriched Craig-Bampton (ECB) method [133]. The static modes are not updated.

4. Updating of the Craig-Bampton reduction basis using the strategy outlined in this work, with the same tolerances as used in the two previous sections namely $\varepsilon = 10^{-6}$ for the static modes and $\varepsilon = 10^{-3}$ for the vibration modes.

5. Low tolerance updating of the Craig-Bampton reduction basis, with $\varepsilon = 10^{-4}$ for the static modes and $\varepsilon = 5 \cdot 10^{-2}$ for the vibration modes, and using the correction for the rigid body modes as outlined in 6.2.4.3.

All reduced bedframe models consist of 78 DoF (48 static modes and 30 vibration modes) except for the ECB reduced model, which due to the addition of the sensitivity modes has 108 DoF. The different reduced models are compared by computing their free interface eigenmodes and frequencies and comparing these to the exact solutions obtained from the unreduced model of the modified structure. Since modes and frequencies are compared, the results are independent of the FE mesh size. The results are shown in figure 8.10 where the relative frequency error and 1-MAC value are compared.

Since the model is free-floating, the first six eigensolutions should be rigid body modes at (approximately) 0 Hz. However, as explained in section 6.2.4.3, inaccuracies in the static modes can cause the rigid body modes to shift to higher frequencies. Therefore, the frequencies of the first six modes in fig. 8.10 are normalized by the first true flexible frequency. For modes 7 and higher the difference between the full and reduced model frequency is plotted, normalized by the corresponding full model frequency. Furthermore, the MAC values for the rigid body modes are not compared, as they can be any linear combination. From figure 8.10 a number of observations can be made:

- Both the nominal CB and ECB bases are unable to properly describe the rotational rigid body modes (modes 4 to 6) of the modified reduced model, even for a small design change of 1%. This severely handicaps the use of these models in an assembled model, as rigid motions already cause reaction forces to neighboring components. This is due to the fact that the static modes are not updated.

- Sufficient accuracy of the static modes is also important to properly represent the first few flexible modes. To accurately predict the higher modes, it is already sufficient to have an approximation of the fixed interface modes in the basis. This can be achieved by adding sensitivity modes to the basis, but even taking only the nominal modes as an approximation gives reasonable results as long as the design change remains small. Obviously, for larger changes these approximations deteriorate, leading to relatively large errors in the frequencies and mode shapes. This can be seen from the results for the scaling of 15%.

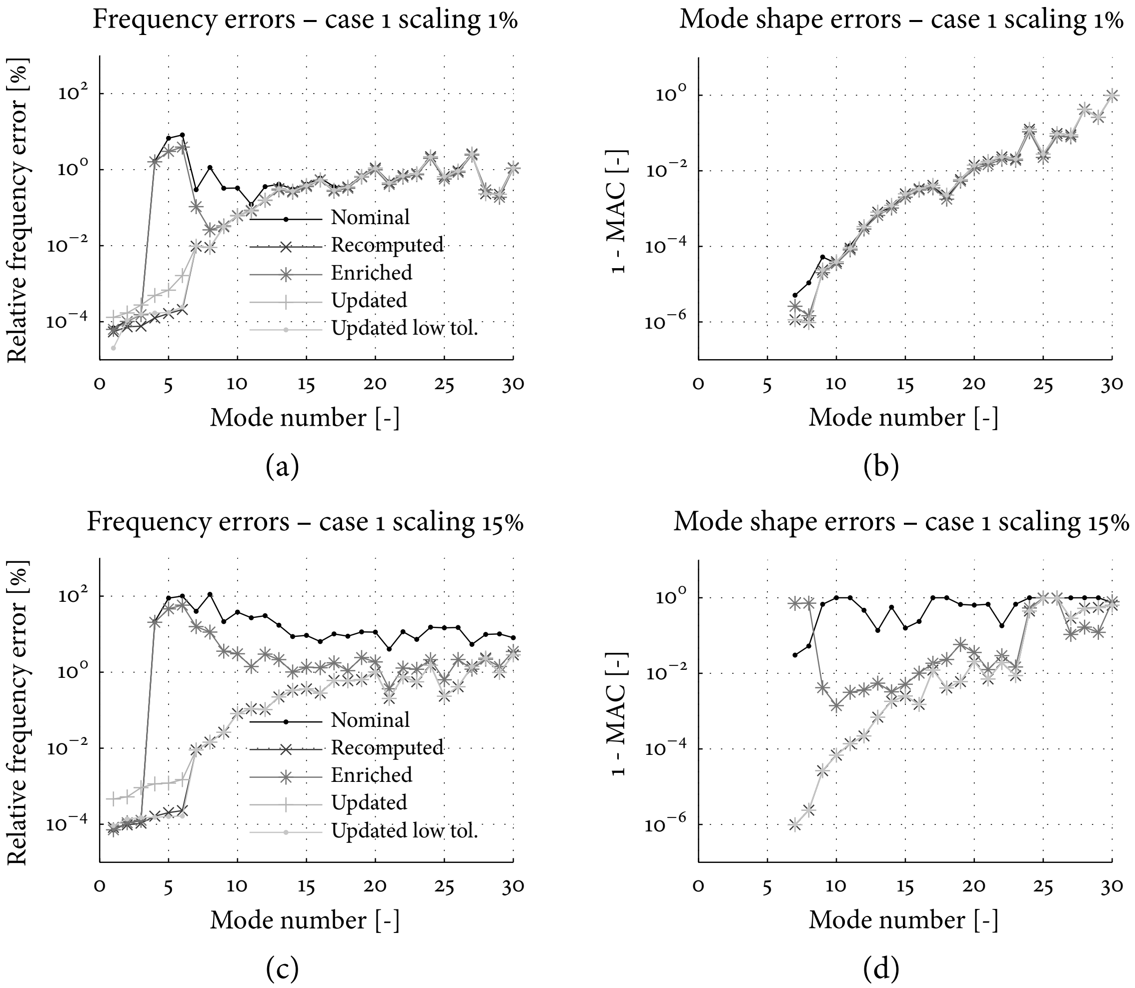


Figure 8.10: Frequency (a,c) and mode shape (b,d) errors for different reduction bases applied to the modified models.

- The updated basis shows only slightly less accurate results for the rigid body modes than the true recomputed CB basis, which is due to the CG algorithm tolerance of 10^{-6} whereas the recomputed static modes are solved to numerical precision. This effect is however negligible and as expected, all higher modes are predicted with the same accuracy as the recomputed CB basis.

- The same holds when the reduction basis is updated to a low tolerance. Due to the rigid body mode correction, the rigid body modes are even represented with the same accuracy as with the recomputed basis. Furthermore, the low tolerance updated vibration modes are sufficiently accurate to account for the dynamics of the modified component.

This comparison shows that even though the reduction basis is updated to a lower tolerance, the accuracy of the reduced model is preserved. Hence, updating the reduction basis ingredients with high accuracy is irrelevant when the approximation made by the reduction itself is not very accurate.

Finally, note that as a result of the lower tolerance settings, the efficiency of the total updating procedure is further improved. This is illustrated for case 1 in figure 8.11. Table 8.16 finally

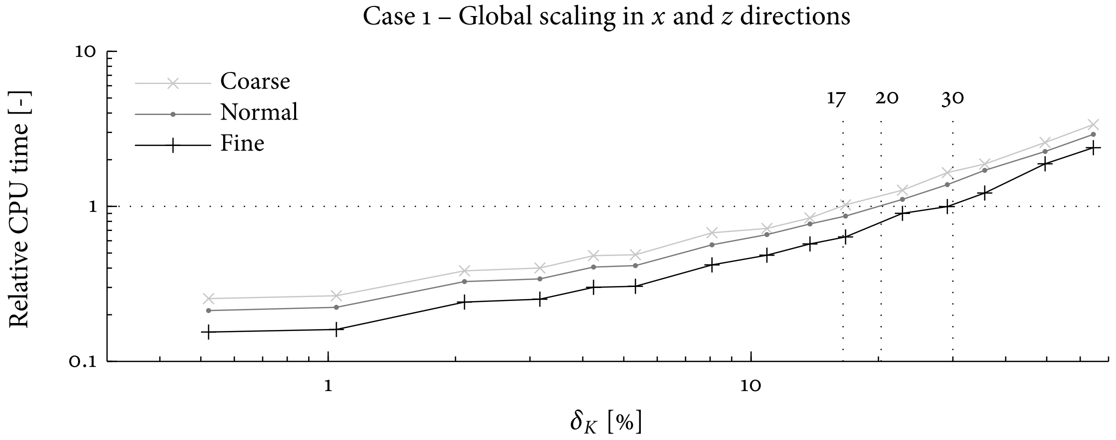

Case 1 – Global scaling in x and z directions

Figure 8.11: Relative CPU time versus intensity of design modification for low tolerance basis updating.

lists the CPU times associated to the low tolerance updating for the three realistic cases discussed before. This shows that when the updating is performed using the low tolerances mentioned before, design changes up to over 30% can be efficiently handled for the fine FE model. For the three typical cases, the updating method performs up to three times faster than recomputation.

Mesh	Case	CPU recomputation [s]		CPU updating [s]		Relative CPU [–]
		Static	Vibration	Static	Vibration	
Coarse	Case 1 – 5%	15	40	15	25	0.73
	Case 2 – 5%			14	24	0.69
	Case 3 – mb			6	25	0.56
Normal	Case 1 – 5%	61	100	39	58	0.60
	Case 2 – 5%			39	63	0.63
	Case 3 – mb			15	57	0.45
Fine	Case 1 – 5%	293	282	104	148	0.44
	Case 2 – 5%			104	144	0.43
	Case 3 – mb			39	148	0.33

Table 8.16: Efficiency of low tolerance basis updating vs. recomputation for three realistic cases.

8.8 Summary

An often seen approach in wind turbine engineering is to develop new wind turbine models by incrementally upgrading existing designs, for instance by increasing the rotor diameter or upgrading the rated generator output. This new turbine configuration requires all relevant components to be checked for their load bearing capacities, and their designs to be modified where needed. Often small design changes are made to these components in order to arrive at a reliable upgraded turbine design.

As explained in chapter 6, this approach is undermining the efficiency of using reduced component models in a practical design setting. For each design modification, a new reduction basis needs to be computed for the component which, especially for large FE models, is computationally expensive. Methods were therefore developed in chapter 6 that allow to iterative update the component reduction basis in order to avoid expensive recomputation.

In this chapter a case study was carried out on a representative structure to test the performance of the proposed updating methods. To this end, the situation is sketched where a wind turbine bedframe is subjected to a range of parametric modifications. In section 8.2, three finite element models of the bedframe were created with respectively 125k, 250k and 500k DoF. Two cases of global design modifications were defined by scaling the bedframe in the length and width/height dimensions, each with 15 different scaling factors ranging from 0.5% to 50% stretching. A third case was defined where the material properties of the bedframe were locally changed from steel to aluminium.

Section 8.3 thereafter considered the updating of the static modes for each of the modified bedframe models using the conjugate gradient based updating algorithms of section 6.2. It was shown that the CG method was capable of efficiently updating the static modes; especially for the large FE model and small to moderate design changes the updating of the static modes was a factor of 2-3 faster than recomputation. Similarly, in section 8.4 the vibration modes were updated using the IFPKS method from section 6.3. It was shown that the IFPKS algorithm is capable of efficiently updating the vibration modes, although in contrast to the static modes, its efficiency with respect to recomputation is independent of the model size. Furthermore, the extended preconditioning and enhanced initial guess methods proposed in section 6.3.4 proved to significantly speed up the analysis.

The case were multiple design changes are applied sequentially was treated in section 8.6. There it was shown that as long as the current modification is closer to the previous one than to the nominal model, it is very beneficial to extend the preconditioning with the available CG iterates.

Finally, in section 8.7 the accuracy of the updated reduction bases was investigated. Regardless of the design modification, the accuracy of the models reduced with the updated static modes proved to be on par with the accuracy obtained when the modes are recomputed from scratch. In fact, the iteration tolerance could be lowered without noticeably affecting the reduced model's accuracy, thereby further improving the computational efficiency. It can therefore be concluded that the proposed basis updating methods form an attractive alternative to both recomputation and existing approximation methods.

9

Optimal Reduced Models of an Offshore Wind Turbine

9.1 Introduction

9.1.1 Offshore Wind Energy

Ever since wind turbines are installed for commercial purposes, developers are searching for sites that maximize the energy harvested from their wind farms. With attractive onshore locations in Europe running out, the wind industry turned its attention to offshore sites. In addition to more favorable wind conditions (higher average speeds, lower turbulence levels), this trend of going offshore mitigates some of the often heard objections against wind energy, namely the acoustic and visual nuisance. Furthermore, size constraints are often less stringent when installing turbines offshore, allowing even larger turbines to be developed. Over the last decade, the offshore installed wind energy capacity has increased exponentially and many more offshore wind farms are currently planned or under construction. This is reflected by the data in table 1.1 and figure 9.1.

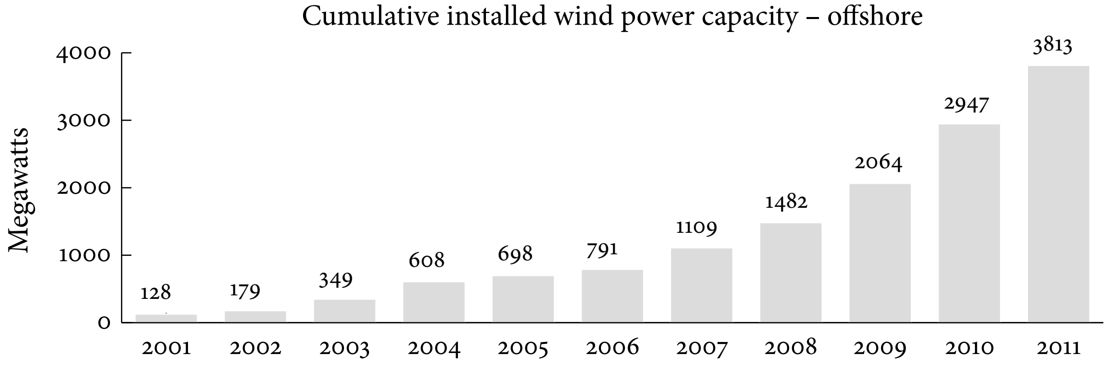


Figure 9.1: Development of installed offshore wind power capacity in Europe, source EWEA [62].

However, as was discussed in chapter 1, electricity generated from offshore wind is still too expensive to be competitive with conventional sources in unsubsidized markets. The wind energy industry is therefore working hard to achieve further cost reduction. One of the ways the industry has successfully achieved this for onshore wind energy is by scaling up the wind turbine size, as a large portion of the cost is more or less independent of the turbine size (e.g. costs of foundation, grid connection, installation). The same trend is now seen for offshore wind farms, where turbines with power ratings of six megawatts are expected to soon dominate the market. Because of size constraints on land, most of these next-generation machines are exclusively developed for offshore applications. In order to maximize the electricity production of these huge machines, developers are gradually moving towards sites farther offshore. Here average wind speeds are generally higher, but so are the water depths. Indeed, the average water depth for offshore installed wind turbines has increased from 11.8 meters in 2009 to 22.8 meters in 2011 (see [62]).

These trends have some profound implications for the foundations on which offshore wind turbines are installed. Before going into details, first the following terminology is defined, as (partly) adopted from [205]:

- *Wind turbine generator* (WTG) refers to the rotor-nacelle-assembly (RNA) and tower
- The *foundation* is the "underwater" structure including transition piece (if applicable)
- The *support structure* is the combined foundation and tower

Hence the complete offshore wind turbine (OWT) is either the assembly of WTG and foundation, or the combination of RNA and support structure. Note that the latter perspective allows a clear split between a standard, large scale produced RNA and a custom engineered support structure that is optimized to suit the local conditions. This split offers an important way to cut costs of offshore support structures by designing them in an integrated manner, as was demonstrated in [197, 86].

Up to now, by far the most popular type of offshore foundation has been the monopile due to its simplicity and robustness; at present 75% of all European offshore turbines are installed on this type of structure. An additional 21% employs a gravity based foundation while the remaining turbines are installed on more complex foundation structures [62]. These different types of foundations are illustrated in figure 9.2. However, the tendency of installing ever larger wind turbines in increasingly deeper water renders monopiles less and less attractive, since a disproportional amount of material is needed to withstand the more severe loading and fulfill the eigenfrequency requirements. More complex types of foundations then become economically attractive, with jacket structures currently as the most promising option. Next the ramifications of these trends for the engineering of offshore wind turbines, and more specifically for the required structural dynamic analysis, will be outlined.

9.1.2 Dynamic Analysis of Offshore Wind Turbines

Since the environmental conditions (water depth, soil properties) and ambient excitations (aero- and hydrodynamic loading) vary greatly across different offshore sites, each offshore wind farm is custom engineered. This means that a standard RNA is combined with a specifically designed support structure. An important part of the engineering process is to perform dynamic simulations to assess whether the support structure can withstand the loads during its specified lifetime. For certification purposes, thousands of load cases need to be

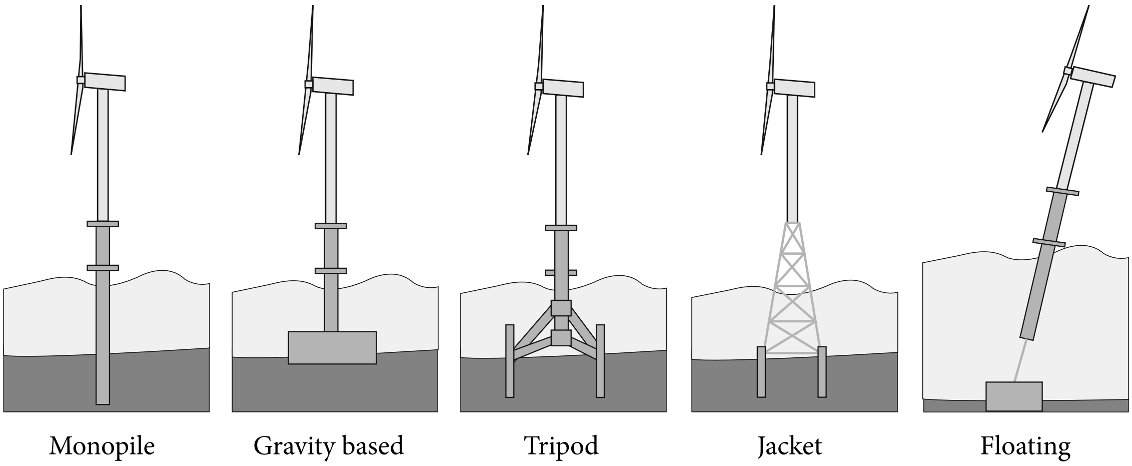


Figure 9.2: Different foundation types for (from left to right) increasing water depths.

evaluated. As was explained in section 1.3, aero-elastic codes have been developed to efficiently perform these dynamic simulations. These codes are designed to analyze the global dynamics of a wind turbine, taking into account aerodynamic loads and coupling, the wind turbine controller and hydrodynamic loads. Given the number of load cases that need to be run, these models must be as coarse as possible to keep the computation times at an acceptable level. Such aero-elastic models typically consist of 300-400 degrees of freedom (DoF), including the tower.[1]

As explained above, the offshore wind industry is gradually moving from monopile foundations to deeper water solutions such as jackets. Also, integrated design methodologies are being explored to obtain more optimized support structure designs. However, a jacket type of foundation is considerably more complex then a monopile and so is its dynamic behavior. Consequently, a more detailed model is needed to predict these dynamics. Where a monopile can typically be represented with a few beam elements leading to a model of at maximum 50 DoF, a jacket structural model easily consists of more than 1000 DoF. Needless to say, this negatively impacts the computation times of the dynamic simulations. Simplifying or even neglecting the jacket dynamics cannot be done in general since dynamic coupling can occur between the jacket, tower and RNA [183, 180]. Fully coupled dynamic simulations of the integrated system are therefore essential.

Therefore, the goal of this chapter is to create "optimal" reduced models for the combined RNA and complex support structure. These models are "optimal" in terms of their size (number of DoF) in relation to their accuracy. To this end, this chapter employs the error estimation and adaptive reduction methods of chapter 5, as well as the modal truncation augmentation method treated in section 2.7 to create compact yet accurate models of the offshore wind turbine.

---

[1] This holds true for finite element (FE) based codes (such as HAWC2 and BHawC); modal based codes (e.g. Flex5 and FAST) usually have much less DoF, typically in the order of 30-50.

9.1.3 Chapter Outline

This chapter is organized as follows. First, section 9.2 describes the reference offshore wind turbine model used throughout this chapter: the NREL 5 megawatt RNA on the OC4 support structure consisting of a jacket foundation and tubular tower. Furthermore, section 9.2 describes the hydrodynamic loading that will be applied to the model. Thereafter, the OWT model will be adaptively reduced in section 9.3 according to several criteria using the error estimation techniques of chapter 5. In section 9.4 the reduced model of the jacket foundation is subsequently augmented using the MTA method of section 2.7, to improve its response to hydrodynamic loading. Several load cases are analyzed and the results obtained from the optimal reduced model are compared to results from other reduced models. A summary concludes the chapter in section 9.5. Note that this chapter is based on the work reported in [142, 143, 216] and, in addition to the aforementioned chapters and sections, employs different methods treated in chapters 2, 3 and 4.

9.2 System Description

In order to test the effectiveness of the modeling methods, a model of an offshore wind turbine is needed that is representative for the models used in the aero-elastic codes. Therefore, use is made of a reference model described in the literature that consists of a 5 MW rotor-nacelle assembly on a tubular tower and jacket offshore foundation. Note that for practical reasons here the model is separated in a WTG (i.e. RNA and tower, hereafter simply referred to as "wind turbine") and foundation. However, the discussion is equally valid in case the split would be made between the RNA and support structure (i.e. foundation and tower).

The wind turbine model is based on the reference design created by the National Renewable Energy Laboratory (NREL) in the United States. This design, known as the "NREL 5 MW baseline wind turbine", is a 5 MW wind turbine of the conventional horizontal axis, three-bladed, upwind type on a tubular tower. Its detailed specifications can be found in [108]. The wind turbine is supported by the jacket structure initially designed for the UpWind project and subsequently used in the Offshore Code Comparison Collaboration Continuation (OC4). Details of this jacket structure can be found in [221]. Therein also some modifications are given for the NREL turbine model which are also taken into account in the subsequent model description. Next the modeling of the wind turbine and jacket foundation will be discussed in more detail, as well as the hydrodynamic loading it is subjected to.

9.2.1 Implementation of NREL 5 MW Baseline Wind Turbine

The NREL wind turbine consists of a standard tubular tower, a nacelle, hub and three blades, where the latter three form the RNA. Figure 9.3 (a) indicates the main dimensions. Since the RNA and tower will be reduced separately in the next section, they are considered as two individual components. In this work these components are modeled as follows:

Tower A conical tower is created according to the specifications given in [221]. The tower is 68 meters in height, with a base diameter of 5.6 meter and top diameter of 4.0 meter, and is modeled using 8 three-dimensional linear Euler-Bernouilli beam elements. The tower has two 6 DoF interfaces, one at the bottom (tower bottom), which will be connected to the jacket structure and one at the top (tower top) which will be connected to the nacelle.

RNA – Nacelle The nacelle is modeled as a single stiff and massless element with on one end a point mass representing the nacelle's mass (240000 kg) and inertia, and on the other a 6 DoF interface which connects to the tower top. The large stiffness and low mass of the element are created using a high Young's modulus and a low density value. The parallel axis theorem was used to calculate the moments of inertia around its local axes. By modeling the nacelle in this way, the drivetrain dynamics are neglected and the rotor is implicitly assumed to be parked. Such a simplified model is justified by the fact that here the main interest lies in the influence of the support structure on the overall dynamics.

RNA – Hub Similar to the nacelle, the hub is modeled as a single stiff and massless element. The hub has a virtual distance of 1.5 meters between its center and the connection to the blades. The hub's mass (56780 kg) is added as a point mass at one end of the element together with its moments of inertia. The element has two 6 DoF interfaces, one end connects to the tower top, the other to the three blades.

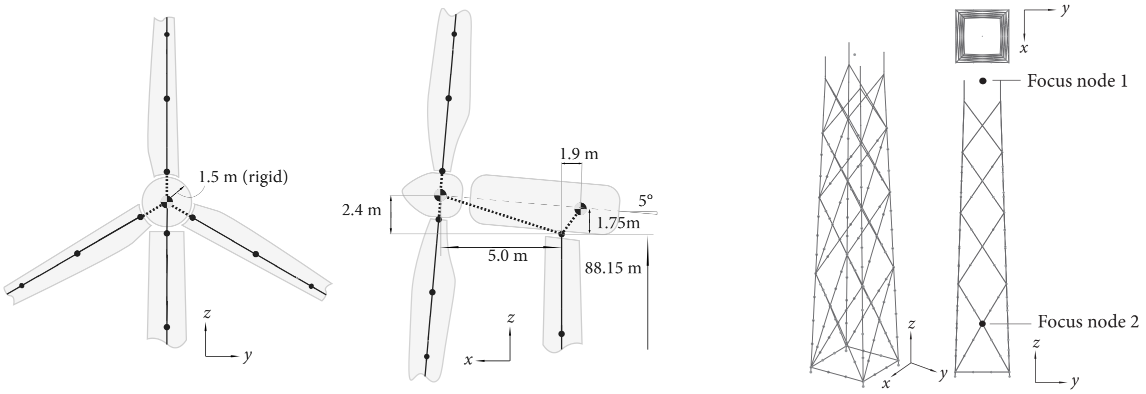
RNA – Blades The blades are 61.5 meters in length and each has a mass of 17740 kg. They are modeled using 13 three-dimensional linear Euler-Bernouilli beam elements. Again, since here the influence of the support structure on the overall dynamics is investigated and no aerodynamic loads will be applied to the rotor, a linear model for the blades is deemed sufficiently accurate. A root-element, located at the base of each blade, is modeled as a 1.5 meter long stiff and massless element to account for the virtual distance to the hub center. The root-element has a 6 DoF interface which connects to the hub element.

The tower and RNA component models are created in Matlab using the DS Toolbox (see appendix A); their numbers of DoF and elements as well as those of the assembled wind turbine model are listed in table 9.1.

Part	No. of DoF	No. of elements
Rotor nacelle assembly	252	41
Tower	54	8
Assembled wind turbine	300	49

Table 9.1: Properties of RNA and tower component models and assembled wind turbine model.

Finally damping is added to the assembled wind turbine model such that, when fixed at tower bottom, the lowest 100 modes have a modal damping ratio of 2% (or approximately 12.5% logarithmic decrement). This is done to account for the structural and aero-dynamic damping (in a vary rough way), as well as to ensure numerical stability of the time simulations performed in section 9.4. The damping is implemented by computing the corresponding undamped eigensolutions and synthesizing the damping matrix using the eigenfrequencies, mode shapes and desired damping ratios (see e.g. [44]).



(a) (b)

Figure 9.3: RNA of the NREL 5 MW baseline turbine (a) and model of the OC4 jacket (b).

9.2.2 Implementation of OC4 Jacket Foundation

The jacket foundation used in the OC4 project, hereafter called the OC4 jacket, consists of four main legs which are interconnected via four levels of cross-braces, see figure 9.3 (b). The jacket is designed for water depths of 50 meters and has a height of 68 meters. Based on the information in [221], the various parts of the jacket are modeled as follows:

Structure The main jacket structure, i.e. the four legs and cross-braces, is modeled in ANSYS using linear beam elements. Their properties are defined according to the specifications given in [221].

Seabed connection As was described in [221], the stiffness of the connection to the seabed is assumed to be high and is therefore modeled as rigidly clamped; this excludes any soil effects from the model. Furthermore, the grouted connection between the main legs and soil piles is also modeled rigidly, leaving only the bottom 0.5 meters flexible.

Transition piece The connection between the turbine and jacket is realized using a concrete transition piece with a mass of 666 ton; its inertia was calculated using this mass and the specified dimensions. This transition piece rigidly connects the top elements of the jacket structure. The interface to the tower bottom is realized by a 6 DoF coupling.

Added mass A number of factors are described in [221] that add to the mass of the jacket structure, thereby influencing its dynamic behavior. These are the marine growth on the structure, the water contained in the free-flooded legs and the hydrodynamic mass felt from the surrounding water when the jacket structure vibrates. These added masses are calculated as nodal point masses (with directional properties) using the ASAS software package, which is an extension of ANSYS specialized for offshore and maritime applications. These masses are thereafter added to the structural model in ANSYS.

After completing the jacket model in ANSYS, it is exported to Matlab using the FEMLink toolbox (see [17]) for reduction and further calculations using the DS Toolbox (see appendix A). The total jacket model consists of 229 linear beam elements and 1014 DoF. A plot is shown in figure 9.3 (b), where the point masses representing the added mass are shown as dots.

Finally, as defined in [221], damping is added to the model such that the first 100 modes have a damping ratio of 1% (or approximately 6% logarithmic decrement). This accounts for hydrodynamic and soil damping and is implemented in the same manner as described for the wind turbine model.

9.2.3 Verification of the Offshore Wind Turbine Model

By assembly of the RNA, tower and jacket models the total model of the offshore wind turbine is obtained, which consists of 1308 DoF. In order to verify the correctness of the implementation of the numerical model some checks are performed, namely:

Masses A straightforward but important check is to compare the mass contained in the models. An overview of the masses of the models created here is given in table 9.2, along with (averaged) results reported by the OC4 participants [220]. It can be seen that the mass of the turbine model is in good agreement up to the level of accuracy available from the reported OC4 results. The same holds for the mass of the transition piece and marine growth on the jacket, while minor differences are observed for the structural, flooded water and hydrodynamic added mass. These are however well below one percent and therefore considered to be within modeling tolerances.

Model	Part	Calculated mass [tons]	OC4 mass [tons]
Turbine	Rotor nacelle assembly	350	350
	Tower	217	217
Jacket	Structure	676	675
	Transition piece	666	666
	Marine growth	180	180
	Water mass flooded legs	205	204
	Hydrodynamic added mass	672	675

Table 9.2: Mass comparison of turbine and jacket models.

Eigenfrequencies Preferably an eigenfrequency comparison should be performed on the individual wind turbine and jacket models. Unfortunately, no such data was found in the literature. Therefore a modal analysis was performed on the total OWT model; the first ten damped eigensolutions are listed in table 9.3, the mode shape descriptions are adopted from [220]. The results generally compare well to those reported by the OC4 participants (see [220]), but two remarks are in place. Firstly, since the drive-train is not modeled explicitly here, the drive-train mode is absent. Secondly, the rotor is fixed in the current simplified model of the nacelle. As a result the first side-side mode is shifted to a lower frequency than the first fore-aft mode.

9.2.4 Hydrodynamic Loading

Since this chapter focusses on the modeling of the foundation structure, other influences are excluded from this case study. Therefore, the numerical experiments in the coming sections are performed with only hydrodynamic forces applied to the jacket, while the rotor is

Mode no. [-]	Frequency [Hz]	Damping ratio [%]	Mode shape description
1	0.30	3.3	First global side-side
2	0.30	3.3	First global fore-aft
3	0.74	2.1	First asymmetric flapwise yaw
4	0.77	2.1	First asymmetric flapwise pitch
5	0.80	2.1	First flapwise collective
6	0.91	3.6	First edgewise collective
7	0.97	4.1	First asymmetric edgewise pitch
8	1.23	2.1	First asymmetric edgewise yaw
9	1.26	2.1	Second global fore-aft
10	1.29	2.6	Second global side-side

Table 9.3: First 10 eigensolutions of total model.

parked and no aerodynamic loads are applied. The hydrodynamic loads on the jacket support structure have been created in the ASAS software package. Based on a JONSWAP wave spectrum[2], four hydrodynamic load cases have been considered for different sea conditions, namely:

Sea state description	Wind speed [m/s]	Significant wave height H_s [m]	Mean wave period T_s [s]	Mean current speed [m/s]
Ripple waves	4	0.2	1.9	0
Low waves	12	0.8	3.4	0
Medium waves	25	2.7	6.2	0
High waves	25	3.8	7.2	0.5

Table 9.4: Properties of hydrodynamic load cases.

In the first three cases the current velocity is zero, so the hydrodynamic forces only originate from waves colliding with the jacket structure. In the high waves case, there is a mean current speed of approximately 0.5 m/s which varies slightly with depth. The wave model used in ASAS assumes irregular linear waves; the forces on the jacket structure are calculated using the Morison equation [138]. This equation contains quadratic coupling terms between the structural and water velocities giving rise to hydroelasticity. In line with common practice in the wind industry, here the hydroelastic effect is however assumed to be negligible and the hydrodynamic loading is simply applied in an uncoupled fashion. Note that this assumption is valid as long as the relative velocities remain low. Furthermore, the effect of marine growth on the surface roughness is taken into account and the main direction of hydrodynamic loading is in the negative x-direction.

For each sea condition a (quasi-)stochastic time series of 100 seconds is obtained with time steps of 0.04 seconds, so 2500 samples in total. The time step is such that the highest frequency that can be represented in the data is higher than the highest frequency present in the signal. Figure 9.4 shows the normalized amplitude spectra of the different load cases

---

[2] JONSWAP is the abbreviation for the "Joint North Sea Wave Observation Project" carried out in the 1970's with the aim of developing wave spectra that can be applied in the engineering of offshore structures for the North Sea [87].

summed over all nodes in the x-direction, where the current (i.e. zero frequency) component in the high wave case can clearly be seen. The importance of dynamic analysis is underlined by the fact that the frequency content of the loading is in the range of the eigenfrequencies of the total model, so modal excitation is likely to occur. This can lead to an energy build-up in the system causing large displacements and high stresses, thereby reducing the fatigue lifetime of the structure.

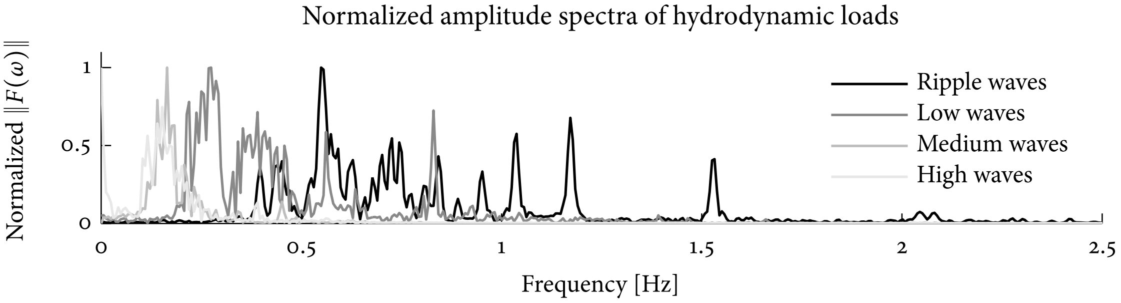


Figure 9.4: Frequency spectra of different wave time series.

Next, the hydrodynamic load time series are analyzed using the proper orthogonal decomposition (POD) method. With this method, that is explained in more detail in appendix D, the most dominant information can be extracted from the time signals in terms of proper orthogonal modes (POMs) and proper orthogonal values (POVs). The POMs describe the dominant force shapes while the POVs indicate their relative energy content. Hence, the POMs can be used to load the jacket structure, as will be done in the next section, and/or be used to compute modal truncation augmentation vectors (see section 2.7), as will demonstrated in section 9.4.

A POD analysis is performed on each of the four load cases in table 9.4. For each load case the first five POMs capture more than 95% of the total energy in the signal, showing the effectiveness of the POD method. However, a limitation of the POMs obtained in this way is that they are based on a single load case. To obtain more general force vectors an extended time series is therefore created by simply combining all four load series, from which generalized force vectors can be obtained using again the POD method.[3] The first five generalized POMs proved to capture 92% of the total energy in this combined time signal. From a vector correlation between the load case specific and generalized POMs using the MAC (see section 4.7.2) it can be seen that the generalized POMs show reasonable correlation with the POMs of the low, medium and high wave cases, but a bad correlation to those of the ripple wave case. See figure 9.5.

9.3 Adaptive Reduction Applied to Offshore Wind Turbine Model

Now the reference model is described, the next step is to apply model reduction. Indeed, with over 1300 DoF the full model is far larger than the models typically used in the wind industry and, given the number of load cases that need to be analyzed, ill-suited to perform

---

[3] To prevent the introduction of high frequencies in the combined data, a signal window was used to scale the magnitude of the loads to zero at the transitions between time series. Furthermore, because the magnitude of the force data varies for different wave types, a normalization step is taken by scaling the maximum occurring value of the individual time series to unity.

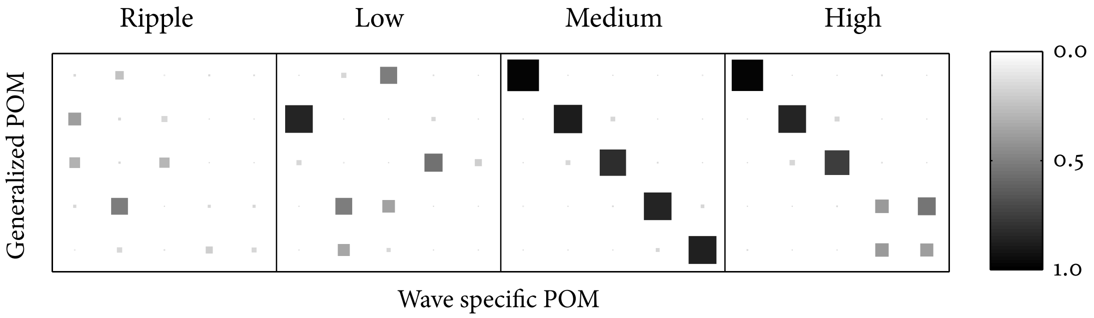


Figure 9.5: Correlation of force modes obtained from different wave time series.

actual load simulations. In this section the OWT model will therefore be adaptively reduced using the methods outlined in chapter 5. Hereby the aim is to create models that have optimal accuracy, in terms of a specified quantity, with respect to their size.

In chapter 5 the theory of error estimation and adaptive model reduction was presented. Estimates were derived that conservatively approximate the error created by a reduced model, both for local quantities of interest using goal oriented errors or in global terms using error norms. These estimates can be used to answer the question "is the reduced model accurate enough?". A logical consequence when the answer to this question is "no" is to ask "which component model(s) should be enriched to most efficiently improve the accuracy of the assembled model?". Indeed chapter 5 also showed that the error estimates not only provide global accuracy information but also give the contributions of the local components. This allows to pinpoint the component models that contribute most to the total error and select them for refinement, thereby efficiently improving the global accuracy. In wind turbine engineering this is important knowledge since one usually strives to obtain the most accurate reduced model with the least amount of DoF.

For the purpose of this case study, the total OWT model is divided into three components: the jacket foundation, the tower and the rotor-nacelle assembly (RNA). One could argue that the RNA could be further separated into individual blade, hub and nacelle models to allow more localized refinement of reduced component models. However, due to the simplicity of both the nacelle and hub models and the fact that in practice only a single reduced model for all three blades will be used, this separation is not made. The three component models are connected (using primal assembly) by two 6 DoF interfaces, see figure 9.6. The jacket structure and the RNA both have a single interface of 6 DoF which connects to the tower bottom and tower top, respectively.

In the remainder of this section adaptive model reduction is applied to the OWT model for four different cases, which are described in the next subsection. Subsections 9.3.2 to 9.3.5 thereafter present the results for the four respective cases.

9.3.1 Numerical Experiments

As was shown in section 5.8, many choices can be made when applying error estimation and adaptive reduction techniques to practical problems, such as the type of problem to be solved (harmonic, eigenproblem), the type of error estimate (global or goal oriented) and the refinement scheme. A series numerical experiments were therefore conceived to test the different aspects of these methods:

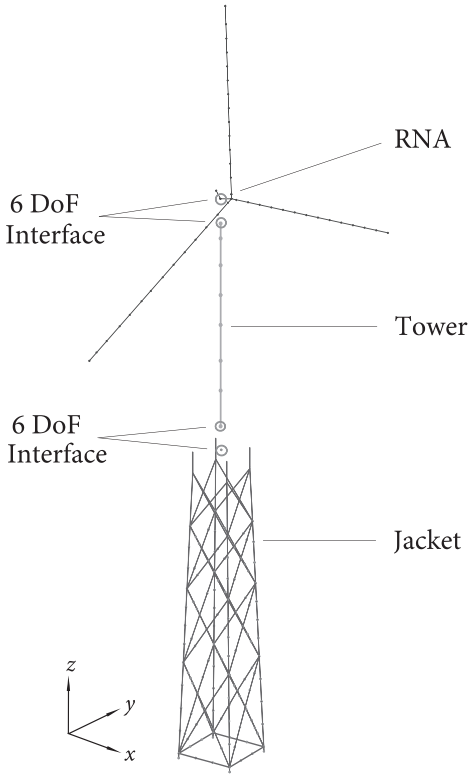


Figure 9.6: The three component models of the OWT model, connected through two interfaces.

- Test 1 – Global reduced model for harmonic excitation. In this test a reduced assembly is created using a limited total number of DoF that are to be distributed in an optimal way over the domains based on their contributions to the total error.

- Test 2 – Goal oriented reduced model for harmonic excitation. For this test a quantity of interest is defined, for which a reduced assembled model is created such that the goal oriented error falls below a chosen threshold.

- Test 3 – Global reduced model for eigensolution convergence. In this test two target eigensolutions are chosen, for which a reduced assembled model is created such that the estimated error on the target eigenfrequency satisfies a preset tolerance.

- Test 4 – Reduced jacket model for multiple criteria. Finally, this test considers the refinement of only the jacket model, such that the assembled model has sufficient accuracy. By simultaneous application of the various error estimates, a reduced model is created that satisfies multiple criteria.

In line with the transformed model description introduced in section 5.3, the assembled model consists of four domains: the internal domains of the jacket structure, tower and RNA and a global interface domain. Furthermore, in all experiments the error estimates are evaluated using the exact acceleration residual $\hat{\boldsymbol{r}}$ (see section 5.4).

In the first two experiments the assembled structure is excited by a harmonic force with a single fixed excitation frequency acting on the jacket model. These force vectors are based on the POMs from the hydrodynamic load series as described in section 9.2.4, since they represent a force shape which captures a large portion of the total energy present in the external excitation. A suitable excitation frequency for the POMs can be obtained by calculating the POD from a singular value decomposition (SVD), as explained in appendix D. By doing so the time modulation of the POMs is obtained, from which the most dominant frequency can be selected for each POM by performing a Fourier analysis. This frequency can be used as the POM's excitation frequency, and is listed in table 9.5 for the first POM of each wave type.

Sea state/wave type	Ripple	Low	Medium	High	Combined
Frequency of first POM [Hz]	1.53	0.29	0.16	0.15	0.14

Table 9.5: Frequency of first POM for different hydrodynamic loads.

9.3.2 Test 1 – Global Reduced Model for Harmonic Excitation

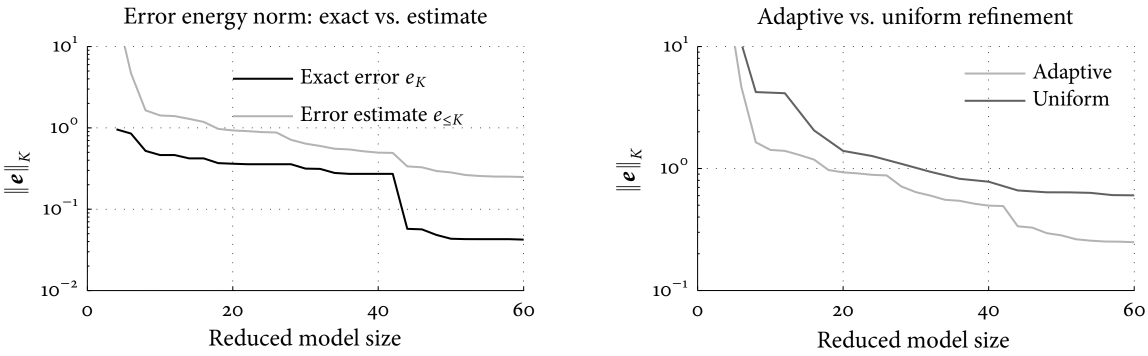
In this first test the aim is to create an optimal reduced model of the assembled OWT for a given maximum number of DoF. An optimal model is searched for in terms of the global energy norm of the error, while the model is subjected to harmonic excitation. To do so, use is made of the error estimation method derived in section 5.4.2. In short, the following settings were used for this test:

Type of problem:	Harmonic
Excitation vector:	Jacket domain loaded by first POM of combined wave load case
Excitation frequency:	0.14 Hz
Type of error estimate:	Global energy norm
Refinement scheme:	Threshold refinement (alg. 5.3) with $c = 1$ and $k_{\text{add}} = 2$
Stopping criterion:	Total model size of 60 DoF

Next different results obtained from this test will be analyzed. Firstly, it is investigated how accurate the actual error estimated is through comparison with the exact error. The latter is obtained by solving the full problem, which in this case is still very cheap for a single load case, and computing the difference with the approximate solution. The result is shown in figure 9.7 (a), where both the exact and estimated error are plotted during the adaptive refinement of the OWT model. As a reminder, the exact error e_K is computed as the left hand side of eq. (5.51) while the estimated error $e_{\leq K}$ is the corresponding right hand side.

As can be seen, the error estimate forms a proper upper bound of the exact error. When the accuracy of the reduced model is still very poor it overestimates its error quite strongly, but for more refined models the estimate is rather close to the actual error (within one order of magnitude).

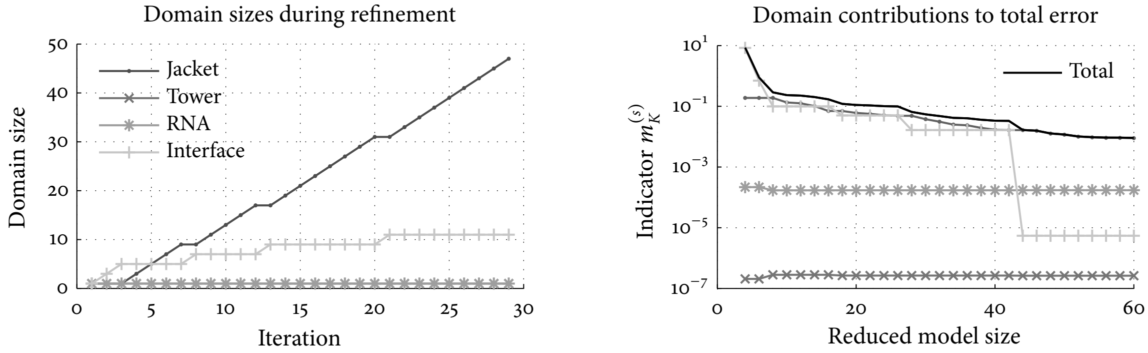
Secondly, the effect of adaptively refining the reduction basis is quantified. In this test refinement of the component models is performed using algorithm 5.3 with $c = 1$ and $k_{\text{add}} = 2$, meaning that at each iteration only the most inaccurate domain is enriched by 2 modes. This is continued until the total model size has reached 60 DoF. For the sake of comparison, the



(a) (b)

Figure 9.7: Test 1: Exact vs. estimated error (a) and adaptive vs. uniform refinement (b).

error estimate is also evaluated for the case where uniform refinement has been applied. That is, each domain is enriched by one mode until the total model again has 60 DoF. The results are shown in figure 9.7 (b). From this figure the benefit of adaptive refinement over uniform refinement can be clearly seen: the energy norm of the error is approximately a factor four times smaller for the adaptively reduced model. This is a significant difference that illustrates the power of adaptive model reduction.



(a) (b)

Figure 9.8: Test 1: Domain sizes (a) and error contributions (a) during refinement.

Furthermore, figure 9.8 (a) shows the size of the different domains during adaptive refinement; the corresponding the domain errors, expressed in terms of the domain indicators $m_K^{(s)}$ defined in eq. (5.52), can be seen in figure 9.8 (b). These plots show that in this case the model of the jacket is most refined, followed by the interface. This makes sense since only the jacket is subjected to external loading and hence shows localized deformation that requires more modes to be properly described. The response of the tower and RNA is already properly captured by only including their first mode. Of course, when the adaptive reduction would be continued these domains will at some point also be further refined, but the error in the jacket is dominant at first.

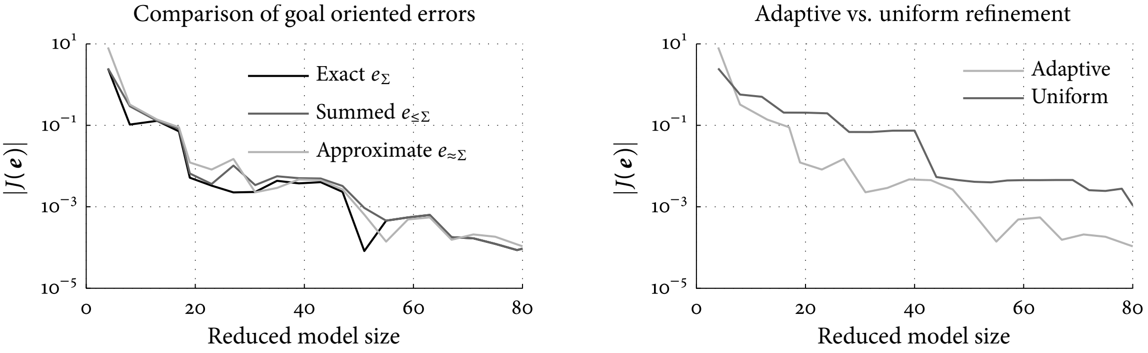
9.3.3 Test 2 – Goal Oriented Reduced Model for Harmonic Excitation

The goal of the second test is to create a reduced model that is optimal with respect to a certain quantity of interest, when subjected to harmonic loading. To this end the goal oriented error estimation technique from section 5.4.1 is applied. The problem statement can be summarized as follows:

Type of problem:	Harmonic
Excitation vector:	Jacket domain loaded by first POM of high wave load case
Excitation frequency:	0.15 Hz
Type of error estimate:	Goal oriented error
Quantity of interest:	Displacements at two focus nodes on jacket (see fig. 9.3) and at tip nodes of each blade, all weighted equally
Size of adjoint problem:	Additional 10 modes per domain
Refinement scheme:	Linear refinement (alg. 5.4) with $k_{\text{add}} = 4$
Stopping criterion:	Estimated goal oriented error $e_{\approx\Sigma} < 10^{-4}$

With the above settings the adaptive model reduction algorithm coverged after 20 iterations, with a total model size of 79 DoF (due to round-off errors only 3 DoF were added in one iteration). As for the previous test, it is first investigated how well the goal oriented error estimate approximates the exact error. In the plot in figure 9.9 (a) the exact goal oriented error e_Σ is shown (as computed from eq. (5.25)), as well as an upper bound $e_{\leq\Sigma}$ found by summation of the absolute domain contributions (i.e. eq. (5.26)) and the approximate goal oriented error $e_{\approx\Sigma}$ (the right hand side of eq. (5.27)) found from approximately solving the adjoint problem.

As expected, the summed error $e_{\leq\Sigma}$ forms a true upper bound for the exact error. This more or less also holds for the approximate goal oriented error $e_{\approx\Sigma}$, which only occasionally slightly underestimates the exact error. The quality of this approximate error estimator obviously depends on the accuracy with which the adjoint problem is solved; the more refined the adjoint model the better the error estimate and vice versa. In this case an additional 10 modes are used per domain, leading to a rather accurate adjoint model.



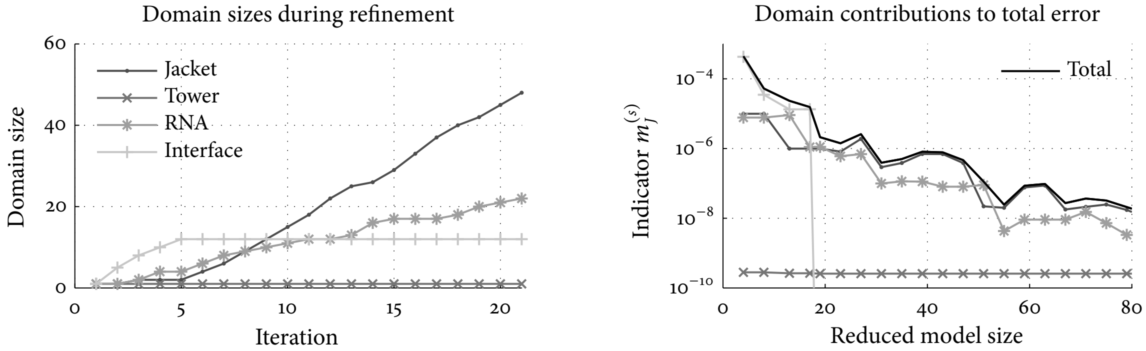
(a) (b)

Figure 9.9: Test 2: Comparison of exact and estimated goal oriented error (a) and adaptive vs. uniform model refinement (b).

Figure 9.9 (b) again shows the difference between adaptive and uniform refinement of the component reduction bases. In this test refinement is performed using the scheme in alg. 5.4 with $k_{\text{add}} = 4$, such that at each iteration 4 additional modes are distributed linearly over the domains according to their relative error contributions. Uniform refinement is again achieved by sequently adding one mode to each domain until a total model size of 80 DoF is reached. For the goal oriented error and refinement strategy considered here, the benefit of adaptive refinement is clear: at a model size of only 80 DoF, the adaptively reduced model is a factor 10 more accurate in the quantities of interest then the uniformly refined model.

The sizes of the four domains during the refinement iterations is shown in figure 9.10 (a), the respective error contributions are indicated in (b). The latter are expressed in terms of the domain indicators $m_J^{(s)}$, see eq. (5.27). It can be seen that at first mainly the interface domain is enriched, until it consists of the full 12 DoF and its error contribution drops to (approximately) zero. This is caused by the fact that part of the quantities of interest are the displacements at the interface between the jacket and tower. Furthermore, the interface is crucial in transferring the response from the tower to the RNA, where the blade tip displacements are also quantities of interest.

After the interface is fully refined, both the RNA and jacket domains are further enriched. Especially the latter is heavily refined which, as in the previous test, is due to the localized loading applied to the jacket as well as the fact that the displacements at one of the jacket brace nodes are chosen as a quantity of interest.



(a) (b)

Figure 9.10: Test 2: Domain sizes (a) and error contributions (b) during refinement.

9.3.4 Test 3 – Global Reduced Model for Eigensolution Convergence

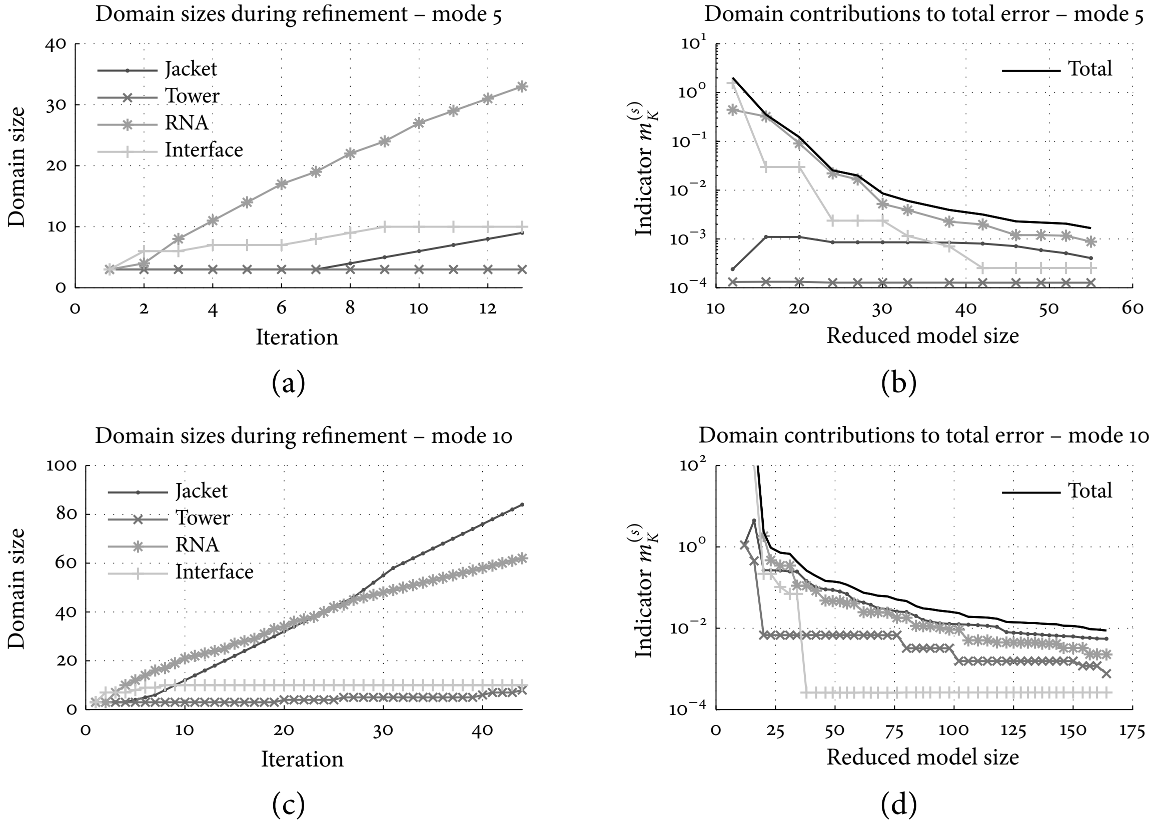
In this test adaptive model reduction is applied with the aim of converging to an eigensolution of the assembled system. Hence the error estimates for global eigensolutions, as derived in section 5.6, are applied to the OWT model. The following settings are used in the adaptive model reduction algorithm:

Type of problem:	Eigenproblem
Type of error estimate:	Eigenfrequency and eigenmode
Target eigensolution	Global mode 5 and mode 10
Refinement scheme:	Linear refinement (alg. 5.4) with $k_{\text{add}} = 4$
Stopping criterion:	Estimated eigenfrequency error $e_{\leq\omega} < 10^{-3}$

Two reduced models are created for the global eigensolution convergence, using respectively the fifth and tenth global mode as target eigensolution. As listed in table 9.3, the fifth global eigenmode has a frequency of 0.80 Hz and is the first flapwise collective motion of the blades,

Target eigensolution	Jacket	Tower	RNA	Interface	Total no. DoF
ϕ_5, ω_5 @ 0.80 Hz	9	3	33	10	55
ϕ_{10}, ω_{10} @ 1.29 Hz	84	8	62	10	164

Table 9.6: Distribution of DoF of adaptively reduced models for two target eigensolutions.



(a) (b)

(c) (d)

Figure 9.11: Test 3: Domain sizes (a,c) and error contributions (b,d) during refinement iterations for target eigensolutions 5 and 10.

also known as the coning mode; its motion is thus RNA-dominated. The tenth eigenmode, with a frequency of 1.29 Hz, is the second global side-side mode and hence involves all domains (jacket, tower, RNA and interfaces). Note that the stopping criterion is defined on the (relative) estimated error on the eigenfrequency $e_{\leq\omega}$; alternatively one can define a termination criterion based on the error estimate of the eigenmode, or a combination. Initially all domains are reduced with 3 vibration modes, i.e. the total model initially has 12 generalized DoF.

With the above settings and the fifth eigensolution as target, the adaptive algorithm converged in 13 iterations leading to a model with 55 DoF. Selecting the tenth mode as target eigensolution the algorithm needed 44 iterations to reach the required accuracy resulting in a total model of 164 DoF. The distribution of the DoF across the different domains is listed in table 9.6. As speculated above, with the fifth mode as target eigensolution the RNA is much more refined than the other domains. Using the tenth mode as target indeed leads to further refinement of all domains; for instance, 8 modes are now required for the tower domain instead of the 3 modes that were sufficient to represent the fifth global mode. The evolution of the domain sizes and error contributions during the refinement iterations are shown in

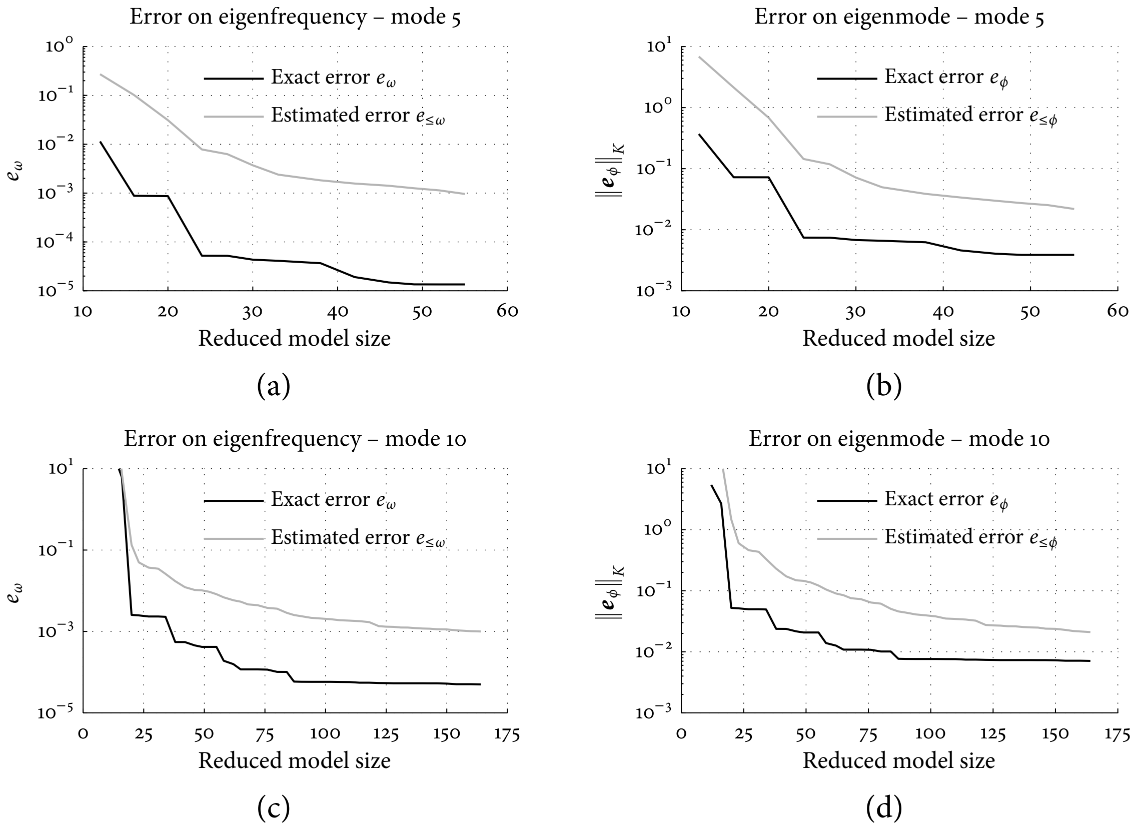


Figure 9.12: Test 3: Comparison of exact and estimated error on eigenfrequency (a,c) and eigenmode (b,d) for target eigensolutions 5 and 10.

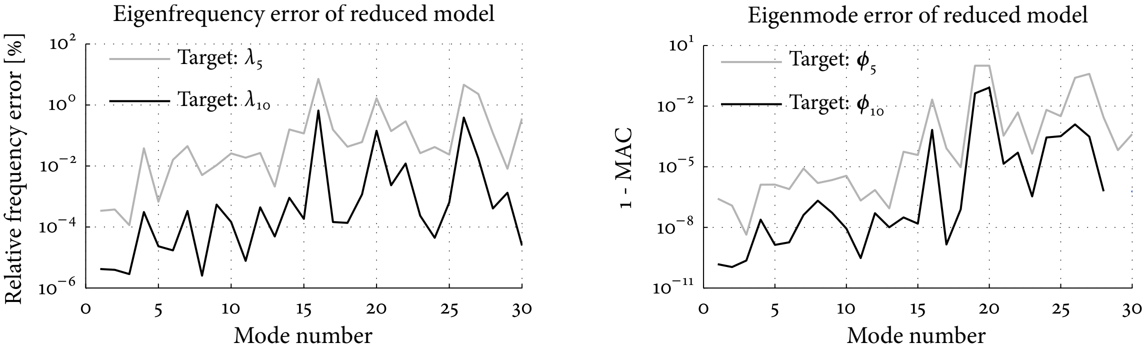
the plots in figure 9.11 for both target eigensolution 5 and 10. As before, the domain error contributions are expressed by the indicators $m_K^{(s)}$, see eq. (5.77).

Next, the error estimates are again compared to their exact counterparts, as shown in the plots in figure 9.12 for the two different target eigensolutions considered here. For the eigenfrequencies, the exact error e_ω as well as the estimated error $e_{\leq\omega}$, as computed from eq. (5.77), are shown in plots (a) and (c). Similarly, the exact and estimated errors for the eigenmodes (respectively e_ϕ and $e_{\leq\phi}$) are shown in plots (b) and (d), where the estimate is obtained from eq. (5.85). From these plots it is shown that the estimates form an upper bound for the exact errors.

Both error estimates however suffer from overestimation of the actual error, to a (much) larger extent then the global and goal oriented error estimates for harmonic problems applied in the two previous tests. This is especially true for the eigenfrequency error estimate, which overestimates the error by almost two orders of magnitude for target eigensolution 5. With increasing model size this overestimation seems to decrease as can be seen from the error estimate for target eigensolution 10. As explained in section 5.6.2, this overestimation is due to the fact that the estimate depends on the actual target eigenfrequency, see eq. (5.77), which is approximated by the reduced model and hence is quite badly overestimated when the model is coarse.

The error estimate on the eigenmode suffers from overestimation to a lesser extent, although still up to one order of magnitude. This due to the definition of the stability factor in eq. (5.84) used in the error estimate of eq. (5.85), which depends on the (exact) difference between the target eigenfrequency and its most closely spaced neighboring eigenfrequency. Again, this stability factor is evaluated with the reduced model, leading to overestimation of the true error.

Finally, in figure 9.13 the modal results for both reduced models are given. To this end the eigensolutions of the reduced models are compared to those of the unreduced model in terms of the relative frequency error and 1-MAC value. It is observed that, as expected, the more refined model obtained with target eigensolution 10 gives better results over a wider frequency range. However, at some modes of the global system the difference is less pronounced (e.g. modes 16, 19 and 20). This is believed to be due to the localized deformations associated to these modes, that are only captured reasonably well by the (fixed interface) vibration modes used in the reduction bases of both models.



(a) (b)

Figure 9.13: Test 3: Errors on eigenfrequency (a) and eigenmodes (b) with respect to the full model.

9.3.5 Test 4 – Reduced Jacket Model for Multiple Criteria

In this final test of the error estimation methods, a situation is sketched that is quite common in wind turbine engineering. Namely, the situation when a new component model is introduced to an existing (reduced) assembly. For this new component its level of enrichment has to be determined such that a certain accuracy of the assembled model is achieved. In practice this situation can be encountered for instance when adapting models of onshore wind turbines for the simulation of offshore turbines, or when exchanging a simple monopile foundation model for a complex foundation model such as a jacket structure.

To investigate the use of error estimation methods in this situation, this test is set up as follows. The tower, RNA and interface domains remain unreduced, while the jacket domain is initially reduced with a single fixed interface mode and subsequently refined by one mode at a time. Since only one substructure is refined no refinement strategy is used. Multiple criteria are defined to determine the level of refinement of the jacket model, using all the error estimates applied in the previous tests:

Eigenproblem	
Target eigensolution:	Global mode 1 @ 0.30 Hz (global side-side mode)
Criterion 1:	Eigenfrequency error $e_{\leq\omega} < 10^{-4}$
Criterion 2:	Eigenmode error $e_{\leq\phi} < 10^{-2}$
Harmonic problem	
Excitation:	First POM of combined wave load case @ 0.14 Hz
Goal quantity:	Displacements at jacket-tower interface node
Adjoint problem:	Additional 10 modes in jacket domain
Criterion 3:	Energy norm of global error $e_{\leq K} < 10^{-1}$
Criterion 4:	Goal oriented error $e_{\approx\Sigma} < 10^{-5}$

The motivation for criteria 1 and 2 is that the global side-side mode of an offshore wind turbine, due to the lack of aerodynamic damping, is in practice very lightly damped in comparison to the fore-aft mode. Vibrations in this mode hence contribute strongly to the fatigue loads on the support structure, and are for instance excited by misaligned wind and wave loading. Hence the model should be capable of accurately predicting this mode.

Criteria 3 and 4 are motivated by the fact that the jacket model should be able to properly describe the deformations due to wave loading. Especially important in this respect is the accuracy of the deformations (and loads) at the jacket-tower interface, since in practice these are post-processed by the foundation designer for detailed design purposes. Hence these deformations are chosen as the quantity of interest for the goal oriented error, for which also quite a strict criterion is chosen.

The results for these four criteria are presented in figure 9.14. In total 24 iterations were performed, such that the jacket domain contains 25 fixed interface vibration modes. It is observed that the criteria on the eigenfrequency and goal oriented errors are most stringent, both requiring approximately 20 DoF to reach the desired accuracy. Furthermore it is noted that, as explained in section 6.3.1, the error on the eigenmode is approximately the square root of the error on the corresponding frequency. Also note that whereas the three global error estimates ($e_{\leq\omega}$, $e_{\leq\phi}$ and $e_{\leq K}$) monotonically decrease with increasing model size, the goal oriented error $e_{\leq\Sigma}$ does not improve when certain modes are added to the jacket reduction basis (e.g. for modes 12-17).

In conclusion, a reduced model size of around 20 DoF for the jacket structure should provide sufficient accuracy of the total offshore wind turbine model. In the next section, it will be investigated how this model performs when it is used in the time simulation of the total OWT model.

9.4 Modal Truncation Augmentation & Time Simulation

As described in the introduction to this chapter, an important part of the engineering effort for an offshore wind turbine is constituted by the aero-elastic load simulations. In this section the reduced model developed before will therefore be used to perform time simulations. To resemble a realistic situation, this section continues on the situation sketched in test 4 of the previous section, namely when an existing aero-elastic code is extended with the model of a complex offshore support structure. To this end, the full model of the wind turbine will be combined with a reduced foundation model.

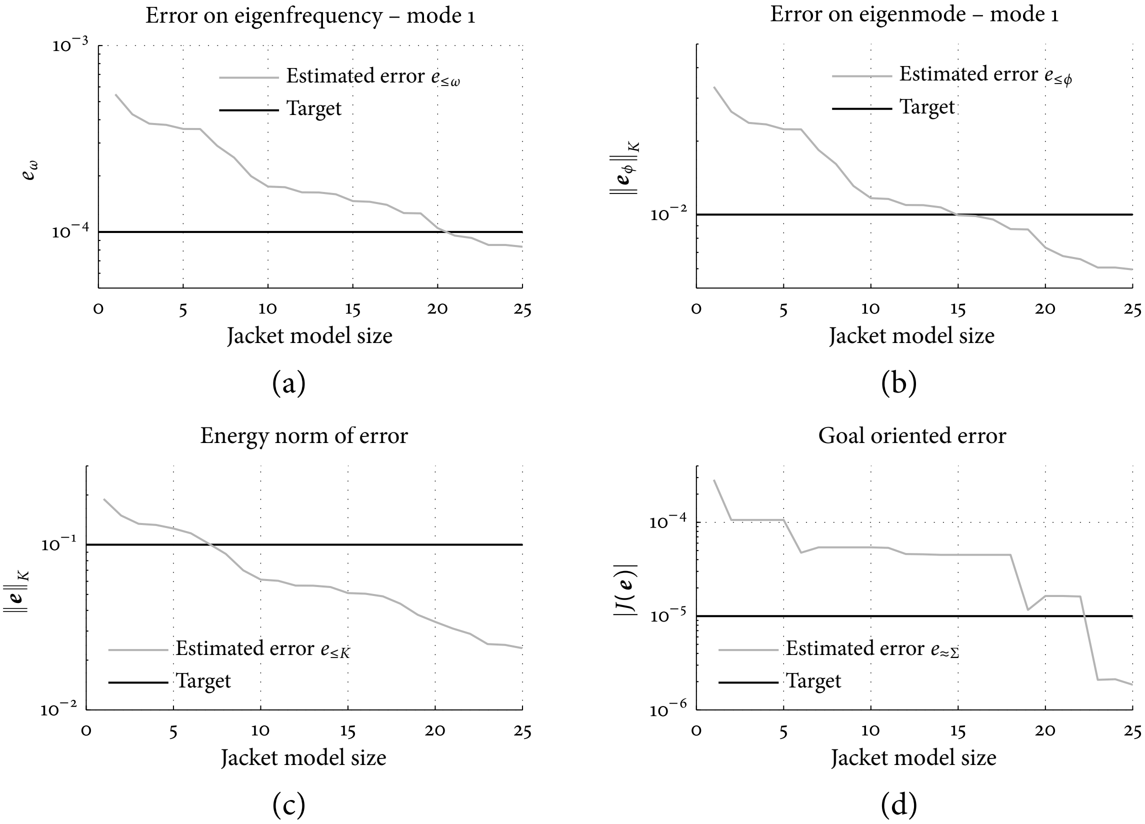


Figure 9.14: Test 4: Comparison of different error estimates during refinement of jacket structure.

Although in the previous section it was shown that the total model could be accurately described when the jacket model is reduced using the Craig-Bampton method with 20 fixed interface modes, this section addresses an often overlooked problem associated with such a reduced model. Indeed, since the jacket foundation is subjected to hydrodynamic loading at a large portion of its DoF, traditional reduction methods fail to properly describe the component's transient response to this excitation. Hence in this section the reduced model is extended using the concept of modal truncation augmentation. This methods was described in section 2.7 and consists in adding load dependent "residual vectors" to the reduction basis. In this it is aimed to combine the best of two worlds: the computational efficiency of reduced component modeling on the one hand, with the accuracy and hydrodynamic loading capabilities of unreduced coupling approaches (such as described in [110, 184]) on the other hand.

This section is organized as follows. The next subsection treats the different reduced models of the jacket foundation that will be assembled to the full wind turbine model. Thereafter, section 9.4.2 will show the results of a modal comparison of these models, while subsection 9.4.3 will show a comparison of time simulation results. In all comparisons the unreduced model will be the reference.

9.4.1 Reduced Models

In subsection 9.3.5 it was shown that when the jacket internal domain is reduced with 20 fixed interface modes, good accuracy of the assembled system is found. As discussed above, it will be shown in this section that for transient analyses with internal loading, the accuracy might actually not be so good. Therefore, modal truncation augmentation vectors will be added to the reduction basis to improve the transient response.

In order to make a fair comparison, the total jacket reduced model size will be kept constant at 26 DoF, six of which are reserved for the static modes. Hence 20 DoF can be used for any combination of vibration modes and force dependent MTAs. In section 2.7 it was shown that MTA vectors can be based on force vectors arising from interface or external loading. Here the focus lies on the latter, by creating MTAs based on the spatial force vectors obtained from the time varying hydrodynamic load data using the POD method described in section 9.2.4. Based on these POMs first and second order MTAs are created and added to the reduction basis, yielding 10 MTAs in total. Since in total 20 DoF can be allocated, the reduced jacket models including MTAs carry an additional 10 vibration modes.

Sometimes it is advocated in the wind industry to simply represent the jacket model as a 6 DoF equivalent mass and stiffness (see e.g. [185]), since a jacket structure is often regarded as being a very stiff structure that is only participating quasi-statically in the global dynamics. For the sake of comparison, the jacket model is therefore also reduced using constraint modes only, resulting in a 6 DoF model. This is indeed a Guyan type of reduced component, see section 2.5.1, where the internal dynamics are discarded.

All reduced jacket foundation models are assembled to the unreduced wind turbine model; an overview of the created models is given in table 9.7. The last letter in the name of the reduced models with MTAs indicates the load case on which the MTA vectors are based (r = ripple, l = low, m = medium, h = high and c = combined).

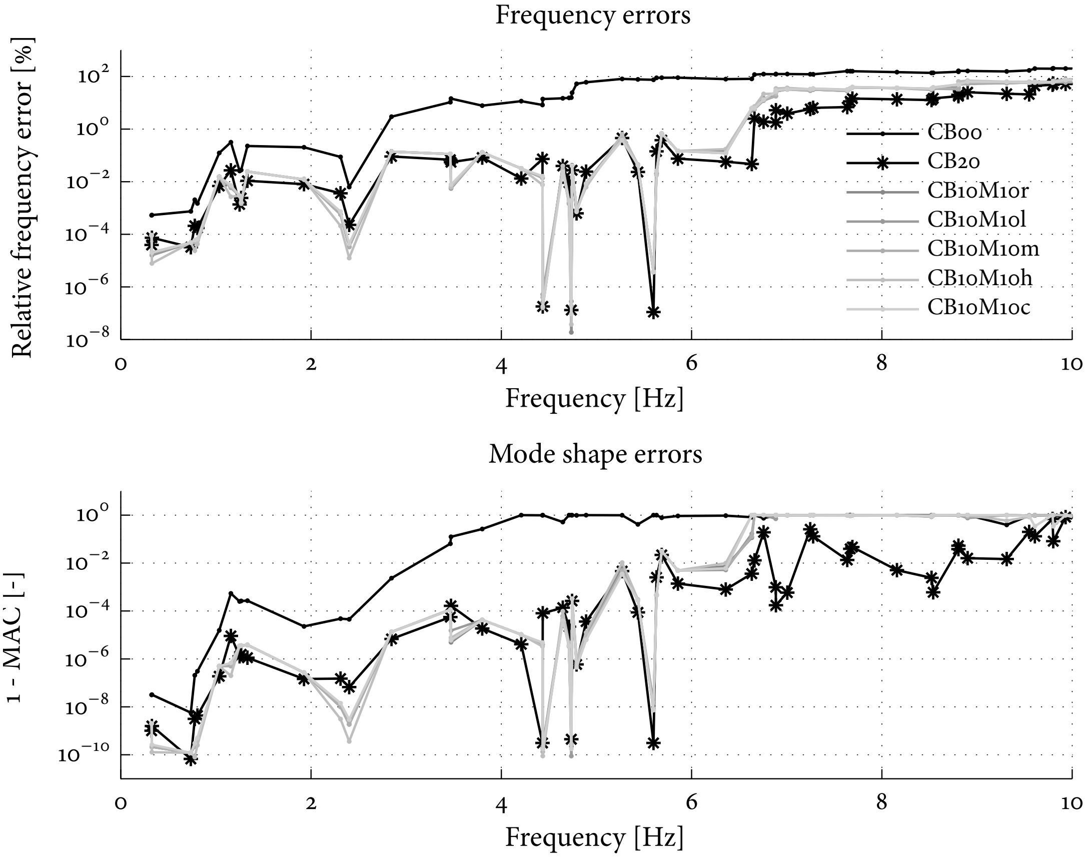
Jacket					Turbine	Total model	
Load case for MTA	# Ψ_c	# Φ_i	# Φ_{Mi}	# DoF	# DoF	# DoF	Name
-	-	-	-	1014	300	1308	REF
-	6	-	-	6	300	300	CB00
-	6	20	0	26	300	320	CB20
Ripple	6	10	10	26	300	320	CB10M10r
Low	6	10	10	26	300	320	CB10M10l
Medium	6	10	10	26	300	320	CB10M10m
High	6	10	10	26	300	320	CB10M10h
Combined	6	10	10	26	300	320	CB10M10c

Table 9.7: Overview of numerical models of baseline offshore turbine.

With the reduced models created, their accuracy with respect to the full model can be assessed. This is done on the basis of the outcomes of both a modal and time response analysis, the respective results are presented in the next two subsections.

9.4.2 Modal Comparison Results

The models listed in table 9.7 are subjected to a modal analysis and the solutions of the reduced models are compared to those of the full model. In figure 9.15 the results for all modes up to 10 Hz are plotted, with the frequency values on the horizontal axis taken from the reference model. As before, the accuracy of the eigenfrequencies is expressed by relative the relative frequency error with respect to the corresponding eigenfrequency of the full model, while the accuracy of the mode shapes is expressed in terms of the 1-MAC value.


Frequency errors

Mode shape errors

Figure 9.15: Modal results of the reduced wind turbine models.

From figure 9.15 a number of observations can be made:

- Obviously, the addition of fixed interface vibration modes and/or MTA vectors greatly improves the modal results compared to the Guyan reduced model (CB00). Indeed, the CB00 model is only accurate up to approximately 2.5 Hz (i.e. the lowest 13 modes); higher eigenfrequencies are at least 10% off and the associated modes become increasingly inaccurate. In practice this is unacceptable, since excitation of the internal jacket dynamics (i.e. modes of the system with predominant displacement of the jacket members) in this frequency range cannot be predicted with the CB00 model, or in fact any equivalent 6 DoF model.
- The modal results of all five models reduced with MTA vectors are very similar. For the modes up to around 6 Hz (mode 32) the results on both the eigenmodes and eigenfrequencies are of similar quality as the CB20 model, hereafter however the CB20 model produces better results as expected. Nevertheless, in practice a 5 MW offshore wind turbine probably experiences little excitation at frequencies higher than 6 Hz. Furthermore, the associated mode shapes are rather complex, so that modal excitation is not very likely.

- Very high accuracy is observed for three modes between 4 and 6 Hz for all models, with the exception of the CB00 model. Upon inspection of these mode shapes it was found that they show very localized deformation of the jacket structure which is exactly represented by the fixed interface vibration modes. These modes are present in all reduction bases (except the CB00 model) and thus explain the high correlation.

9.4.3 Time Integration Results

Next, the models are simulated in time in order to obtain the dynamic response of the offshore wind turbine to the hydrodynamic loads described in section 9.2.4. Due to the fact that no aerodynamic forces are applied and the rotor is parked, the wind turbine model is linear. Hence the linear Newmark time integration scheme is employed, which was described in section 4.6.3. For the full model, this scheme took 11 seconds to perform the time integration with a length of 100 seconds and time step of 0.04 seconds. The reduced models all took only around 4 seconds to integrate, so about a factor 3 faster. Given the fact that thousands of 10 minute simulations need to be evaluated in practice of every wind farm, and usually a time step of 0.02 seconds is required, the use of a reduced jacket model adds up to a serious reduction of computational cost.

As was explained in section 4.7.4, comparing results obtained from time simulations is always a difficult task and many techniques exist to do so. Here it was chosen to directly compare the time series at the two focus nodes shown in black in figure 9.3. Focus node 1 is the interface node at tower bottom, focus node 2 is one of the nodes in the lowest level of cross braces. To limit the amount of information presented, only the displacements in x-direction (the main loading direction) are shown for only the low and high wave load cases. Additionally, to obtain a comparison on a global level, it was also chosen to show the total deformation energy in the jacket structure over time. The time series are shown in figures 9.16 and 9.17 for the low and high load cases, respectively. Some statistics are also listed, where the differences are defined with respect to the reference solution. A few remarks can be made based on these results:

- All displacements are in general comparatively small. This is due to a number of factors, namely: the very stiff and heavy jacket structure, the relatively low intensity of the hydrodynamic loads and the absence of aerodynamic loading.

- For both load cases the displacements at the interface DoF (focus node 1) are accurately predicted by all models, even the CB00 model. This is believed to be partly due to the fact that the internal dynamics are not observed on the interface, and party because of the “uncoupling” effect of the heavy transition piece and the absence of aerodynamic loads on the turbine. Such good agreement for the CB00 model might not be found when more dynamic interaction is taking place, for instance under other loading conditions and for more optimized jacket designs.

- When considering the internal DoF (focus node 2) the picture looks totally different and clear discrepancies can be observed between the models. For both load cases the CB00 model underestimates the response by approximately 30 to 40 percent. Without further postprocessing in the form of a "retrieval run"[4] this could lead to severe underestimation of the loads/stresses in the structure. However, as was shown by recent investigations [208], even when performing a retrieval run in a dynamic fashion, it cannot be ensured that the correct internal jacket displacements are obtained.

The CB20 model on the other hand overestimates the response by some 10 percent. The models containing MTA vectors finally, are able to very accurately follow the responses of the full model.

- An indication of the global accuracy of the responses is given by the plots of the instantaneous deformation energy in the system. It can be seen that the CB00 model contains on average 30 to 40 percent less deformation energy, indicating that it is much too stiff and deformations (and hence stresses) are underestimated. The CB20 system already captures more of the deformation energy but still serious underestimation can occur, up to 24% for the low wave case. This can greatly affect the calculated fatigue loads in the structure and hence emphasizes the need for more accurate reduced models. Indeed, the energy plots show that the use of MTAs in the reduced jacket model allows to predict the global deformations with very high accuracy, especially given the compact size of the models.
- Another interesting observation is that the system with MTA vectors based on the combined POMs (CB10M10c), yields results that are comparable to those when load case specific MTAs are used (CB10M10m and CB10M10f). This justifies the approach of obtaining POMs from combined load data to produce generalized MTAs and thus a generalized reduced model, which greatly benefits the practical usability of the method.

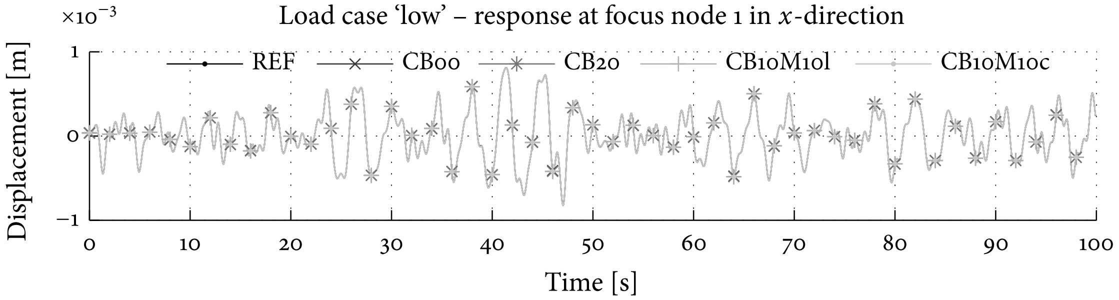
From all results it can be concluded that the CB00 model is the least accurate, which was to be expected since the jacket model is statically reduced to only 6 DoF. Given the large discrepancies found when using this model, it is advisable to refrain from using such oversimplified models in practice. The addition of both fixed interface vibration modes and MTA vectors improve the results, but the best results for the forced transient response are by far obtained by adding MTAs. Thereby it makes little difference whether they were obtained from specific load cases or from more general load data.

9.5 Summary

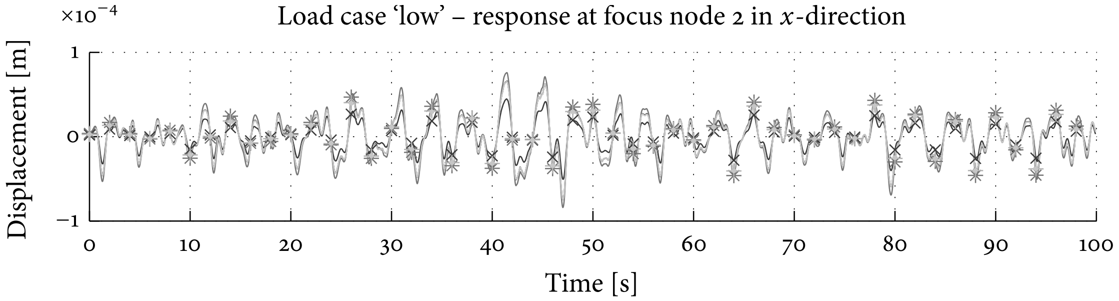
Offshore wind energy has the potential to generate "green" electricity on a large scale and has seen tremendous growth in installed capacity during the last decade. However, since environmental conditions vary greatly across offshore sites, the support structures for offshore wind farms are custom engineered. This requires thousands of load cases to be evaluated with aero-elastic codes, which use computationally efficient but geometrically coarse models. While monopiles long were the default choice of foundation type, the wind industry is gradually moving towards more complex foundations such as jackets to cope with ever larger

---

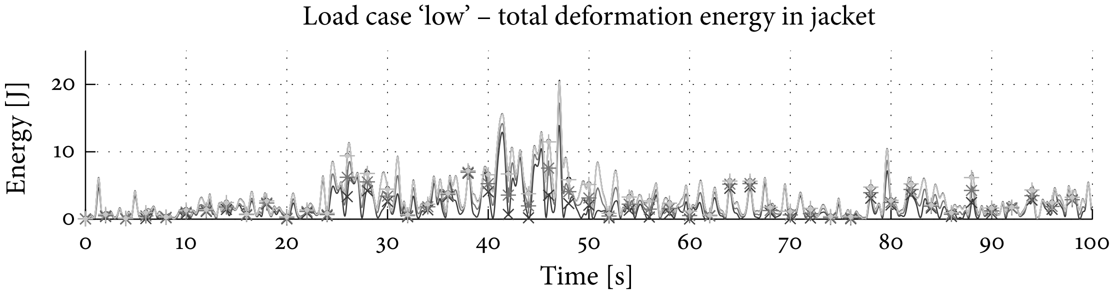
[4] A retrieval run, as defined in [184, 185], amounts to the following. After the initial coupled dynamic simulation, internal forces and moments at the interface are extracted. These are subsequently applied to the stand-alone jacket structure in a quasi-static or dynamic fashion, possibly combined with a prescribed interface motion resulting from the initial coupled simulation. The internal jacket deformations are obtained from this additional analysis. See references for details.



Model	Maximum absolute displacement		Standard deviation	
	Value [m]	Difference w.r.t. REF [%]	Value [m]	Difference w.r.t. REF [%]
REF	$8.24 \cdot 10^{-4}$	-	$2.70 \cdot 10^{-4}$	-
CB00	$8.13 \cdot 10^{-4}$	−1.2	$2.69 \cdot 10^{-4}$	−0.6
CB20	$8.22 \cdot 10^{-4}$	−0.2	$2.70 \cdot 10^{-4}$	−0.1
CB10M10m	$8.24 \cdot 10^{-4}$	0.07	$2.70 \cdot 10^{-4}$	−0.005
CB10M10c	$8.24 \cdot 10^{-4}$	0.08	$2.70 \cdot 10^{-4}$	−0.004

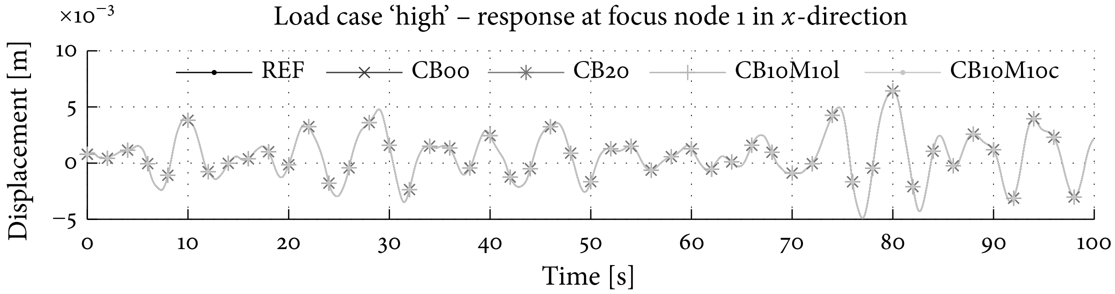


Model	Maximum absolute displacement		Standard deviation	
	Value [m]	Difference w.r.t. REF [%]	Value [m]	Difference w.r.t. REF [%]
REF	$7.48 \cdot 10^{-5}$	-	$2.23 \cdot 10^{-5}$	-
CB00	$5.06 \cdot 10^{-5}$	−32	$1.49 \cdot 10^{-5}$	−33
CB20	$8.39 \cdot 10^{-5}$	12	$2.59 \cdot 10^{-5}$	16
CB10M10m	$7.48 \cdot 10^{-5}$	−0.05	$2.24 \cdot 10^{-5}$	0.34
CB10M10c	$7.43 \cdot 10^{-5}$	−0.71	$2.22 \cdot 10^{-5}$	−0.78

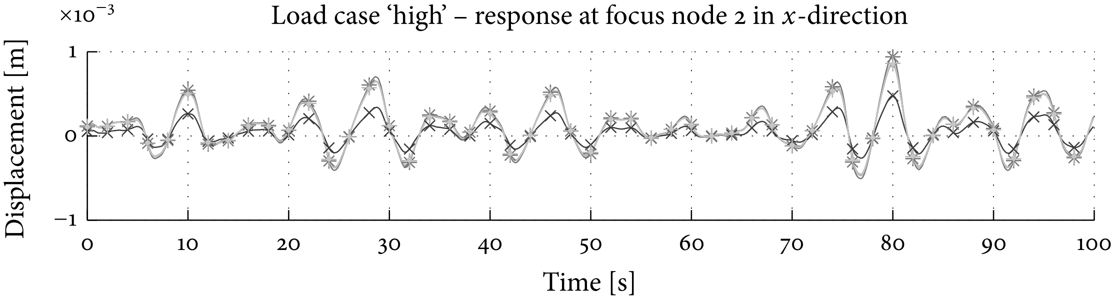


Model	Mean		Maximum		Standard deviation	
	Value [J]	Diff. REF [%]	Max [J]	Diff. REF [%]	Value [J]	Diff. REF [%]
REF	3.19	-	20.6	-	2.75	-
CB00	1.78	−44	13.9	−32	1.98	−28
CB20	2.42	−24	17.2	−16	2.30	−16
CB10M10m	3.12	−2.2	20.5	−0.7	2.75	−0.04
CB10M10c	3.11	−2.4	20.4	−0.8	2.75	−0.06

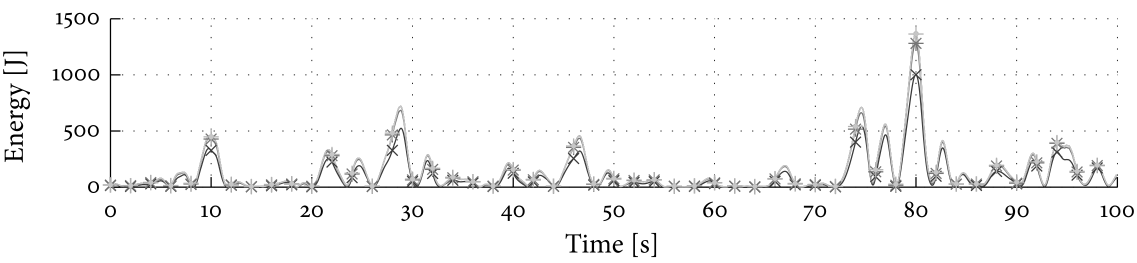
Figure 9.16: Results for low waves load case.


Load case 'high' – response at focus node 1 in x-direction

Model	Maximum absolute displacement		Standard deviation	
	Value [m]	Difference w.r.t. REF [%]	Value [m]	Difference w.r.t. REF [%]
REF	$6.42 \cdot 10^{-3}$	-	$2.04 \cdot 10^{-3}$	-
CB00	$6.42 \cdot 10^{-3}$	−0.058	$2.04 \cdot 10^{-3}$	−0.12
CB20	$6.42 \cdot 10^{-3}$	−0.009	$2.04 \cdot 10^{-3}$	−0.01
CB10M10f	$6.42 \cdot 10^{-3}$	0.01	$2.04 \cdot 10^{-3}$	0.003
CB10M10c	$6.42 \cdot 10^{-3}$	0.007	$2.04 \cdot 10^{-3}$	0.005


Load case 'high' – response at focus node 2 in x-direction

Model	Maximum absolute displacement		Standard deviation	
	Value [m]	Difference w.r.t. REF [%]	Value [m]	Difference w.r.t. REF [%]
REF	$8.48 \cdot 10^{-4}$	-	$2.39 \cdot 10^{-4}$	-
CB00	$4.76 \cdot 10^{-4}$	−44	$1.30 \cdot 10^{-4}$	−46
CB20	$9.41 \cdot 10^{-4}$	11	$2.61 \cdot 10^{-4}$	10
CB10M10f	$8.47 \cdot 10^{-4}$	−0.09	$2.39 \cdot 10^{-4}$	−0.006
CB10M10c	$8.51 \cdot 10^{-4}$	0.38	$2.39 \cdot 10^{-4}$	−0.05


Load case 'high' – total deformation energy in jacket

Model	Mean		Maximum		Standard deviation	
	Value [J]	Diff. REF [%]	Max [J]	Diff. REF [%]	Value [J]	Diff. REF [%]
REF	$1.37 \cdot 10^{2}$	-	$1.37 \cdot 10^{3}$	-	$1.94 \cdot 10^{2}$	-
CB00	$1.00 \cdot 10^{2}$	−27	$1.00 \cdot 10^{3}$	−27	$1.45 \cdot 10^{2}$	−25
CB20	$1.29 \cdot 10^{2}$	−6	$1.28 \cdot 10^{3}$	−6	$1.84 \cdot 10^{2}$	−5
CB10M10f	$1.37 \cdot 10^{2}$	−0.3	$1.37 \cdot 10^{3}$	−0.1	$1.94 \cdot 10^{2}$	−0.1
CB10M10c	$1.37 \cdot 10^{2}$	−0.1	$1.37 \cdot 10^{3}$	−0.2	$1.94 \cdot 10^{2}$	−0.2

Figure 9.17: Results for high waves load case.

turbines and greater water depths. Even the simplest models of such structures have many more degrees of freedom (DoF) than the complete wind turbine models in use today, leading to excessive computation times. To overcome this, the current chapter outlined modeling strategies to create "optimal" models of offshore wind turbines on complex foundations.

Firstly, a reference offshore wind turbine (OWT) model from the literature was described and modeled in section 9.2. The model consisted of the NREL 5MW turbine placed on the OC4 jacket foundation structure. Using simple beam models, the wind turbine was modeled with 300 DoF while the jacket model consisted of over 1000 DoF, leading to an assembled model of the total OWT with 1308 DoF. Furthermore, transient hydrodynamic loads were defined for four different sea states. Using the POD method, the most dominant "load shapes" and frequencies were extracted from this loading.

Thereafter, this model was subjected to adaptive model reduction in section 9.3. Using the error estimation and adaptive refinement methods from chapter 5, optimal models were created for four different combinations of problem types (harmonic, modal) and quantities of interest (local, global). The error estimates proved to be true upper bounds of the exact errors, although at some occasions the error was quite strongly overestimated. Nonetheless, in all cases the benefit of adaptive reduction could be clearly observed, leading to reduced models with up to ten times higher accuracy with respect to a uniformly reduced model of the same size. In the fourth test it was established that reducing the jacket model with 20 vibration modes, while leaving the wind turbine model unreduced, should give good accuracy of the combined system.

However, in section 9.4 it was shown that not so good accuracy was found when using this model for transient simulations with the jacket subjected to hydrodynamic loading. Since in this case the jacket is subjected to loading at a large portion of its DoF, traditional reduction methods fail to properly describe the transient response. Therefore the MTA method of section 2.7 was used to add load dependent MTA vectors to the reduction basis, which were computed using the dominant load shapes obtained from the POD analysis. For a fair comparison, 10 MTAs were used to replace an equal amount of vibration modes. As expected, the reduced models with MTAs in their bases perform not as good as the original 20 DoF reduced model in a modal comparison. However, their accuracy in transient analyses was orders of magnitude better, which is most important in the wind engineering practice.

In conclusion, this chapter illustrated the use of adaptive model reduction and modal truncation augmentation. It was shown that when combined, these methods provide a powerful way of obtaining very compact as well as accurate dynamic models, enabling efficient integrated simulation of offshore wind turbines.

PART III

Conclusions & Recommendations

10

Conclusions & Recommendations

If you follow reason far enough
it always leads to conclusions that are contrary to reason.

(Samuel Butler)

10.1 Conclusions

To ensure a leading role in a future sustainable energy supply, the cost of wind energy must be further reduced. A prerequisite for further driving down these costs is thorough insight in the structural dynamic behavior of wind turbines. Current dynamic analysis methods used in the wind industry, based on aero-elastic simulations and static finite element analysis, have some serious shortcomings when it comes to integrated and detailed dynamic analysis. In this thesis it was therefore proposed to fill this need for a new dynamic analysis tool using the paradigm of dynamic substructuring. As a result, the objective of this thesis was defined in chapter 1 as:

> *"Develop a practical modeling framework based on the concept of dynamic substructuring that enables detailed, integrated structural dynamic analysis of wind turbines without compromising on computational efficiency."*

In this thesis, the methodology of dynamic substructuring has proven to enable detailed dynamic analysis while providing versatility through the use of generalized model assembly techniques. With the concepts of model reduction, interface reduction, error estimation and basis updating the computational efficiency of the approach has been maximized. Through realistic case studies it was demonstrated that by combining all these methods, a valuable modeling framework has been developed for use in the wind turbine engineering practice. It can therefore be concluded that the goal of this PhD project has been achieved. To motivate this overall conclusion, next more detailed conclusions will be drawn for the three main topics addressed in this work, namely component model reduction and assembly, error estimation and reduction basis updating.

10.1.1 Conclusions on Component Model Reduction & Assembly

In part I of this thesis it was shown that existing component model reduction techniques can be written in a common notation framework. This allowed to derive a new component reduction technique, called the Mixed Craig-Bampton (MCB) method. It was shown that this method is a natural generalization of the existing CB and DCB methods and overcomes the issue of choosing fixed or free interface vibration modes in the reduction basis. The augmentation of the reduction basis with excitation-dependent MTA vectors was also addressed. Additionally, a general framework was developed for the assembly of component models. Again, this framework is a generalization of existing assembly techniques and allows to combine reduced component models with different interface representations. Furthermore, this framework was extended with methods to incorporate interface physics (stiffness and inertia) and handle non-conforming interface meshes. Hence two conclusions can be drawn from the theoretical contributions on the topics of model reduction and assembly: with the MCB method component models can be created in a systematic and possibly automated way, while the assembly framework allows all types of reduced models to be treated as “superelements”, such that they can be created truly independently thereby enhancing the modularity of the DS methodology.

The application of these methods in wind turbine engineering was treated in part II, chapter 7, by modeling the yaw system of a Siemens 2.3 megawatt wind turbine. Starting from their CAD models, reduced component models were created and assembled to obtain detailed dynamic models of the yaw system. The models were partly validated with measurements. Furthermore, interface reduction was applied to obtain truly compact models (a factor 300-400 smaller than the original model), which proved to accurately describe the dynamics of the full system (less than 1% error on the frequencies and mode shapes up to mode 80). Especially the MCB reduced model showed very good accuracy. As such, these models might be used directly in aero-elastic simulations for specific load cases or could serve as a reference for tuning a more simplified yaw system model. Either way, the models are helpful in gaining more insight in the yaw system’s dynamic behavior and its role in the global wind turbine dynamics. From this result it can be concluded that (interface) reduction methods provide a powerful way of obtaining compact dynamic models, that are useful in the wind turbine engineering practice.

10.1.2 Conclusions on Error Estimation

Error estimation methods were translated to and derived in an algebraic setting in chapter 5. The benefits of these methods are twofold: they allow to estimate the accuracy of a reduced model without the (expensive) evaluation of the full problem, as well as provide insight in the contribution of the different components to the total error. The latter information allows to adaptively obtain an optimal reduced model. In conclusion, the theory of error estimation is a valuable tool in the creation of reduced models, especially when the associated full models are very large.

To illustrate the use of these methods in the wind turbine engineering practice, the modeling of an offshore wind turbine on a complex support structure was considered in chapter 9. This application is especially important in view of the industry trend of installing ever larger turbines in ever deeper waters. From the application of error estimation techniques the potential of adaptive model reduction was shown. Compared to uniformly reduced models of the same size, adaptive reduction led to models with up to ten times higher accuracy for the selected properties. Additionally, augmentation of reduction bases with load dependent vectors was shown to be crucial to improve the model's response to external (hydrodynamic) loading. Results from modal and time domain simulations showed that it is paramount to properly model the coupled dynamic behavior for OWTs on complex foundations, with orders of magnitude higher accuracy compared to equally-sized non-augmented models. Therefore it can be concluded that augmented reduced models are essential for load simulations of wind turbines on for instance jacket foundations, while error estimation and adaptive reduction help to keep the model size to a minimum.

10.1.3 Conclusions on Reduction Basis Updating

To conclude part I, techniques were developed in chapter 6 for the efficient updating of reduction bases of components subjected to parametric design modifications. It was shown how iterative preconditioned methods may be employed to reuse the available information from the nominal model to efficiently obtain the reduction basis for the modified component. Especially in the case of small, incremental design changes the proposed methods offer an efficient alternative to recomputation of the reduced model. Hence it can be concluded that these methods enable the use of reduced models in practical design settings or optimization loops where components are subjected to slight modifications.

In order to demonstrate the application of these methods in wind turbine engineering, a common approach in turbine design was followed in chapter 8. Here the incremental improvement of component designs was considered; more specifically, the bedframe structure of the Siemens wind turbine was subjected to parametric design modifications. From comparisons with recomputation and other reanalysis techniques it was established that the reduced bedframe model could be efficiently updated, even up to reasonably large modifications. For typical design changes a gain in computation time of up to a factor three was found. This proved that reduced component models may be efficiently used in wind turbine design processes.

10.2 Recommendations

Although the main aspects of dynamic substructuring and its application in wind turbine engineering have been covered in this thesis, there is always room for further research. In this section some recommendations will be given as to the focus of future efforts in these areas. This will be done separately for the theoretical and application parts. Regarding the theoretical topics, recommendations for future work are:

- Further develop the efficient use of reduced models in a floating frame of reference (FFR) formulation, to enable simulation of reduced models undergoing large motions. For wind turbines, this is for instance relevant when simulating operational load cases using reduced models of the rotor or drivetrain. For details see the discussion and references in section 2.5.5. In the context of this project a preliminary study was performed, investigating possible simplifications of the FFR theory to speed up the calculations while maintaining accuracy. This showed promising results, but remains to be further worked out and implemented.

- Damping in reduced models. In this work damping of component models has not be considered, although most methods are also valid for lightly damped structures with modal or Rayleigh damping. However, these types of damping are known to be very simplistic representations of real life sources of structural dissipation. Further research is thus required to incorporate more realistic damping models in reduced components (see also discussion and references in section 2.5.5).

- A priori interface reduction. To enable a truly modular "building block" approach, it would be desirable to perform interface reduction in an a priori fashion on substructure level. The issue with such ideas is that one needs to determine the interface reduction basis without knowing anything from neighboring components. One therefore encounters the problem of incompatible interface representations, such that the requiring only discrete interface compatibility can result in significant errors. To overcome this problem one will have to resort to more sophisticated Mortar-type of methods.

- Development of error estimates for the Dual Craig-Bampton and Mixed Craig-Bampton methods. As indicated in chapter 5, this should be possible since the DCB method can be cast in a similar stiffness-uncoupled format as the CB method. However, due to the "saddle point" nature of the DCB equations of motion, other definitions of the error norms are required. This requires additional research. Since the MCB method is a generalization of the CB and DCB methods, it is speculated that once error estimates are derived for the DCB method, they can be generalized to suit the MCB method.

- Combining modal truncation augmentation and error estimation. When applying error estimates for goal oriented quantities, one idea is to enrich the domain reduction bases with MTA vectors based on the domain's force residual. It is speculated that by doing so the reduced model will show very fast spatial convergence towards the exact response, posing an effective way to reducing the goal oriented error. The computation of the additional MTA vectors per iteration requires only minor changes in the refinement algorithms. However, a prerequisite for doing so is the ability to estimate the error from a reduction basis that contains MTA vectors. This might not be so trivial since MTAs are related to specific "inputs" and are not true eigensolutions (even though they can be considered as pseudo-modes). Hence additional research is required.

- A number of other topics can be identified that do not follow directly from this work, but would be very valuable additions to the developed modeling framework. These include:

  - Research related to other classes of (non-modal) reduction techniques, such as balanced truncation from the control engineering community or moment matching techniques (that are similar to the MTA method). The debate on what method to use when, depending on the purpose of the reduced model, is still ongoing.

  - Reduction of non-linear component models. For wind turbine modeling this is for instance relevant for the blades, which can show geometric non-linearity, to model non-linear soil-structure interaction, etc.
  - Reduction of components with multiphysical coupling, so coupling of the structural dynamics with other dynamic physical effects (e.g. aerodynamics, hydrodynamics, acoustics, controller dynamics).

In addition to the recommendations on the theoretical part, the following future work can be envisioned on the applications in wind turbine engineering:

- Reduced model of the complete wind turbine. An interesting application of the DS modeling framework would be to create a reduced model of a complete turbine, including all reduced models from relevant components obtained from detailed FE models. This model could serve two purposes:

  1. Tune/design the coarse structural model used in aero-elastic simulations, for instance by identifying where relevant structural flexibility and inertia is located.
  2. Use directly to simulate a set of reference load cases to verify the aero-elastic model and/or perform detailed investigations on component level.

  The first steps towards such a model were already made in the course of this project by extending the yaw system model with a number of components.

- Include controller models. Controller dynamics strongly influence the overall dynamics of a wind turbine. It would therefore be interesting to include (simplified) controller models in the substructured models of (parts of) the wind turbine.

- Generalization of MTA vectors. To further enhance the practical value of load dependent MTAs, it could be tested whether more generalized MTAs can be calculated such that they can be used for example for multiple loading directions and wind/wave misalignment load cases.

- Error estimation for eigensolutions in a frequency range. In the application in this work, the error estimation method for global eigensolutions was tested only when a single target eigensolution was selected. In practice one usually strives to create a reduced model with a certain accuracy for all eigensolutions in a given frequency range. Therefore it could be tried to apply the adaptive model reduction for instance using several target eigensolutions (and hence error estimates) simultaneously.

- Incorporate experimental models. For wind turbine components that are very difficult to model properly, such as the main gearbox, it would be interesting to try to obtain an experimental model and include it in a dynamic substructuring analysis.

- Validation using measurements. Although in this work a first attempt was made to validate some component models using measurement data, thorough validation of component as well as assembled models is still an open issue. In this respect the updating of component and interface models is a relevant issue.

- Use of the validated reduced model. Once the reduced model has been validated it can be used in the wind turbine design practice for instance for model-based health-monitoring.

Bibliography

[1] Akcay Perdahcioglu, D., Ellenbroek, M., Geijselaers, H., and de Boer, A. Updating the Craig-Bampton Reduction Basis for Efficient Structural Reanalysis. *International Journal for Numerical Methods in Engineering 85*, 5 (2010), 607–624.

[2] Allemang, R. The Modal Assurance Criterion - Twenty Years of Use and Abuse. *Sound and Vibration* (August 2003), 14–21.

[3] Allemang, R., and Brown, D. A Correlation Coefficient for Modal Vector Analysis. In *Proceedings of the International Modal Analysis Conference* (1982), Society for Experimental Mechanics, pp. 110–116.

[4] Allen, M., and Mayes, R. Comparison of FRF and Modal Methods for Combining Experimental and Analytical Substructures. In *Proceedings of the Twentyfifth International Modal Analysis Conference, Orlando, FL* (Bethel, CT, 2007), Society for Experimental Mechanics.

[5] Allen, M. S., and Kuether, R. J. Substructuring with nonlinear subcomponents: A nonlinear normal mode perspective. In *Topics in Experimental Dynamics Substructuring and Wind Turbine Dynamics, Volume 2* (2012), vol. 27 of *Conference Proceedings of the Society for Experimental Mechanics Series*, Springer New York, pp. 109–121.

[6] Allen, M. S., Mayes, R. L., and Bergman, E. J. Experimental modal substructuring to couple and uncouple substructures with flexible fixtures and multi-point connections. *Journal of Sound and Vibration 329* (2010), 4891–4906.

[7] Antoulas, A. C. *Approximation of Large-Scale Dynamical Systems*. Advances in Design and Control. Society for Industrial and Applied Mathematics, 2005.

[8] Aoyama, Y., and Yagawa, G. Component Mode Synthesis for Large-Scale Structural Eigenanalysis. *Computers & Structures 79* (2001), 605–615.

[9] Araujo, X. V., Fransen, S., Germès, S., and Thiry, N. Validation of Equivalent Viscous Damping Methodologies. *CEAS Space Journal (submitted)* (2012).

[10] Avitabile, P. 20 Years of Structural Dynamic Modification - A Review. *Sound and Vibration* (January 2003), 14–25.

[11] Avitabile, P., and Chipman, C. Expansion of transient opertating data. In *Proceedings of the Twenty-seventh International Modal Analysis Conference, Orlando, Florida* (Bethel, CT, February 2009), Society for Experimental Mechanics.

[12] Axelsson, O. A generalized SSOR method. *BIT Numerical Mathematics 12*, 4 (1972), 443–467.

[13] Balmès, E. Optimal Ritz Vectors for Component Mode Synthesis Using the Singular Value Decomposition. *AIAA Journal 34*, 6 (June 1996), 1256–1260.

[14] Balmès, E. Parametric Families of Reduced Finite Element Models. Theory and Applications. *Mechanical Systems and Signal Processing 10* (1996), 381–394.

[15] Balmès, E. Use of Generalized Interface Degrees of Freedom in Component Mode Synthesis. In *Proceedings of the Fourteenth International Modal Analysis Conference, Dearborn, MI* (Bethel, CT, February 1996), Society for Experimental Mechanics.

[16] Balmès, E. Model Reduction for Systems with Frequency Dependent Damping Properties. In *Proceedings of the Fifteenth International Modal Analysis Conference* (Bethel, CT, February 1997), Society for Experimental Mechanics.

[17] Balmès, E., Bianchi, J., and Leclère, J. *Structural Dynamics Toolbox & FEMLink – User's Guide*. Paris, France, October 2011.

[18] Bangerth, W., and Rannacher, R. *Adaptive Finite Element Methods for Differential Equations*. Birkhäuser Verlag, 2003.

[19] Bathe, K.-J. *Finite Element Procedures*. Prentice Hall, Englewood Cliffs, New Jersey, 1996.

[20] Bennighof, J., Kaplan, M., and Muller, M. Extending the Frequency Response Capabilities of Automated Multi-Level Substructuring. In *Structures, Structural Dynamics and Material Conference* (Atlanta, April 3-6 2000), 41st AIAA/ASME/ASCE/AHS/ASC. AIAA-2000-1574.

[21] Bennighof, J., and Lehoucq, R. An Automated Multilevel Substructuring Method for Eigenspace Computation in Linear Elastodynamics. *SIAM Journal on Scientific Computing 25*, 6 (2002), 2084–2106. http.//www.cs.sandia.gov/ lehoucq/papers.html.

[22] Bernardi, C., Maday, Y., and Patera, T. A New Non Conforming Approach to Domain Decomposition: the Mortar Element Method. In *Nonlinear Partial Differential Equations and their Applications*, H. Brezis and J. Lions, Eds. Pitman, London, 1994, pp. 13–51.

[23] Bourquin, F. Component Mode Synthesis and Eigenvalues of Second Order Operators. Discretization and Algorithm. *Mathematical Modelling and Numerical Analysis 26*, 3 (1992), 385–423.

[24] Brahmi, K., and Bouhaddi, N. Improved Component Mode Synthesis Based on Double Condensation Method. In *Proceedings of the Fifteenth International Modal Analysis Conference, Orlando, FL* (Bethel, CT, February 1997), Society for Experimental Mechanics, pp. 1469–1475.

[25] Brüls, O. *Integrated Simulation and Reduced-Order Modeling of Controlled Flexible Multibody Systems*. PhD thesis, Université de Liège, Liège, Belgium, February 2005.

[26] Brüls, O., Duysinx, P., and Golinval, J.-C. The global modal parameterization for nonlinear model-order reduction in flexible multibody dynamics. *International Journal for Numerical Methods in Engineering 69*, 5 (2007), 948–977.

[27] Buhl, M., and Manjock, A. A Comparison of Wind Turbine Aeroelastic Codes Used for Certification. In *44th AIAA Aerospace Sciences Meeting and Exhibit* (2006). Conference Paper NREL/CP-500-39113.

[28] Bulgak, H., and Zenger, C., Eds. *Error Control and Adaptivity in Scientific Computing.* Kluwer, August 1999.

[29] Calvetti, D., Reichel, L., and Sorensen, D. C. An Implicitly Restarted Lanczos Method for Large Symmetric Eigenvalue Problems. *Electronic Transactions on Numerical Analysis 2* (1994), 1–21.

[30] Cerulli, C., van Keulen, F., and Rixen, D. Dynamic Reanalysis and Component Mode Synthesis to Improve Aircraft Modeling for Loads Calculation. In *Structures, Structural Dynamics and Material Conference and Exhibit* (Waikiki, Hawaii, U.S.A., Apr. 23-26 2007), 15th AIAA/ASME/AHS Adaptive Structures Conference.

[31] Chandler, K., and Tinker, M. A General Mass-Additive Method for Component Mode Synthesis. In *Structures, Structural Dynamics and Material Conference* (Kissimmee, FL, Apr. 7-10 1997), 38st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit. AIAA-97-1381.

[32] Chatterjee, A. An introduction to the proper orthogonal decomposition. *Current Science 78*, 7 (April 2000), 808–817.

[33] Chaturantabut, S., and Sorensen, D. Discrete Empirical Interpolation for nonlinear model reduction. In *Proceedings of the 48th IEEE Conference on Decision and Control* (2009), pp. 4316–4321.

[34] Clough, R. The Finite Element Method in Plane Stress Analysis. In *Proceedings of 2nd ASCE Conference on Electronic Computation* (1960).

[35] Courant, R. Variational Methods for the Solutions of Problems of Equilibrium and Vibrations. *Bulletin of the American Mathematical Society 49* (1943).

[36] Craig, R. Coupling of Substructures for Dynamic Analyses – An Overview. In *Proceedings of AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit* (April 2000), pp. 1573–1584.

[37] Craig, R., and Bampton, M. Coupling of Substructures for Dynamic Analysis. *AIAA Journal 6*, 7 (1968), 1313–1319.

[38] Craig, R., and Chang, C. Free-Interface Methods of Substructures Coupling for Dynamic Analysis. *AIAA Journal 14*, 11 (1976), 1633–1635. Technical Notes.

[39] Craig, R., and Chang, C. On the Use of Attachement Modes in Substructure Coupling for Dynamics Analysis. In *Eighteenth Structures, Structural Dynamics and Material Conference* (San Diego, March 21-23 1977), AIAA/ASME, pp. 89–99. AIAA 77-405.

[40] Craig, R., and Chang, C. Substructure Coupling for Dynamic Analysis and Testing. Tech. Rep. CR-2781, NASA, 1977.

[41] Craig, R., and Chung, Y. Generalized Substructure Coupling Procedure for Damping Systems. *AIAA Journal 20*, 3 (1982), 442–444.

[42] Craig, R., and Hale, A. A Review of Time-Domain and Frequency Domain Component Mode Synthesis Methods. *International Journal of Analytical & Experimental Modal Analysis 2*, 2 (1987), 59–72.

[43] Craig, R., and Hale, A. Block-Krylov Component Synthesis Method for Structural Model Reduction. *AIAA Journal of Guidance, Control, and Dynamics 11*, 6 (November-December 1988), 562–570.

[44] Craig, R., and Kurdila, A. *Fundamentals of Structural Dynamics*, second edition ed. John Wiley and Sons, Ltd., New York, London, Sydney, 2006.

[45] Craig, R., and Ni, Z. Component Mode Synthesis for Model Order Reduction of Nonclassically Damped Systems. *AIAA Journal of Guidance, Control, and Dynamics 12*, 4 (July-August 1989), 577–584.

[46] Crowley, J., Klosterman, A., Rocklin, G., and Vold, H. Direct Structural Modification using Frequency Response Functions. In *Proceedings of the Second International Modal Analysis Conference, Orlando, FL* (Bethel, CT, February 1984), Society for Experimental Mechanics, pp. 58–65.

[47] D'Ambrogio, W., and Fregolent, A. Promises and Pitfalls of Decoupling Techniques. In *Proceedings of the Twenty Sixth International Modal Analysis Conference* (Bethel, CT, February 2008), Society for Experimental Mechanics.

[48] D'Ambrogio, W., and Fregolent, A. Decoupling procedures in the general framework of Frequency Based Substructuring. In *Proceedings of the Twenty Seventh International Modal Analysis Conference* (Bethel, CT, February 2009), Society for Experimental Mechanics.

[49] D'Ambrogio, W., and Fregolent, A. The role of interface DoFs in decoupling of substructures based on the dual domain decomposition. *Mechanical Systems and Signal Processing 24* (2010), 2035–2048.

[50] D'Ambrogio, W., and Sestieri, A. A unified Approach to Substructuring and Structural Modification Problems. *Shock and Vibration 11*, 3 (August 2004), 295–309.

[51] de Klerk, D. *Dynamic Response Characterization of Complex Systems through Operational Identification and Dynamic Substructuring.* PhD thesis, Delft University of Technology, Delft, the Netherlands, March 2009.

[52] de Klerk, D., Rixen, D., and de Jong, J. The Frequency Based Substructuring (FBS) Method Reformulated According to the Dual Domain Decomposition Method. In *Proceedings of the Twenty Fourth International Modal Analysis Conference, St. Louis, MO* (Bethel, CT, February 2006), Society for Experimental Mechanics.

[53] de Klerk, D., Rixen, D., and Voormeeren, S. General Framework for Dynamic Substructuring: History, Review and Classification of Techniques. *AIAA Journal 46*, 5 (May 2008), 1169–1181.

[54] de Klerk, D., Rixen, D., Voormeeren, S., and Pasteuning, F. Solving the RDoF Problem in Experimental Dynamic Substructuring. In *Proceedings of the Twenty Sixth International Modal Analysis Conference, Orlando, FL* (Bethel, CT, February 2008), Society for Experimental Mechanics. Paper no. 129.

[55] den Dekker, D. Efficient Modeling of Rotational Effects for Wind Turbine Structural Dynamic Analysis September. Master's thesis, Delft University of Technology, Delft, The Netherlands, September 2010.

[56] Deraemaeker, A., Ladevèze, P., and Leconte, P. Reduced Bases for Model Updating in Structural Dynamics based on Constitutive Relation Error. *Computer Methods in Applied Mechanics and Engineering 191* (2002), 2427–2444.

[57] Dickens, J. A Critique of Mode Acceleration and Modal Truncation Augmentation Methods for Modal Response Analysis. *Computers & Structures 62*, 6 (1997), 985–998.

[58] Dostál, Z. Conjugate gradient method with preconditioning by projector. *International Journal of Computer Mathematics 23*, 3-4 (1988), 315–323.

[59] Duarte, M., and Ewins, D. Improved Experimental Component Mode Synthesis (IECMS) with Residual Compensation Based Purely on Experimental Results. In *Proceedings of the Fourteenth International Modal Analysis Conference* (Bethel, CT, February 1996), Society for Experimental Mechanics, pp. 641–647.

[60] Duncan, W. The admittance method for obtaining the natural frequencies of systems. *Philosophical Magazine 32* (1941), 401–409.

[61] European Wind Energy Association. The European offshore wind industry key trends and statistics 2010.

[62] European Wind Energy Association. The European offshore wind industry key trends and statistics 2011.

[63] Evans, D. The Use of Pre-conditioning in Iterative Methods for Solving Linear Equations with Symmetric Positive Definite Matrices. *IMA Journal of Applied Mathematics 4*, 3 (1968), 295–314.

[64] Ewins, D. J. *Modal Testing. Theory and Practice.* John Wiley & Sons Inc., Taunton, England, 1989.

[65] Farhat, C., and Roux, F.-X. A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm. *International Journal of Numerical Methods in Engineering 32* (1991), 1205–1227.

[66] Feng, Y., Owen, D., and Peric, D. A block conjugate gradient method applied to linear systems with multiple right-hand sides. *Computer Methods in Applied Mechanics and Engineering 127* (1995), 203–215.

[67] Fransen, S. Data recovery methodologies for reduced dynamic substructure models with internal load. *AIAA Journal 42*, 10 (2004), 2130–2142.

[68] Fransen, S. *Methodologies for Launcher-Payload Coupled Dynamic Analysis*. PhD thesis, Delft University of Technology, Delft, The Netherlands, February 2006.

[69] Fransen, S., Fischer, H., Kiryenko, S., Levesque, D., and Henriksen, T. Damping methodology for condensed solid rocket motor structural models. In *Structural Dynamics, Volume 3* (2010), vol. 12 of *Conference Proceedings of the Society for Experimental Mechanics Series*, Springer New York, pp. 273–281.

[70] Freund, R. Krylov-subspace methods for reduced-order modeling in circuit simulation. *Journal of Computational and Applied Mathematics 123*, 1-2 (2000), 395–421.

[71] Gade, S., Schlombs, R., Hundeck, C., and Fenselau, C. Operational modal analysis on a wind turbine gearbox. In *Proceedings of the Twenty Seventh International Modal Analysis Conference* (Bethel, CT, 2009), Society for Experimental Mechanics.

[72] Géradin, M. The computational efficiency of a new minimization algorithm for eigenvalue analysis. *Journal of Sound and Vibration 19*, 3 (1971), 319–331.

[73] Géradin, M., and Cordona, A. *Flexible Multibody Dynamics – A Finite Element Approach*. John Wiley & Sons, Chichester, England, 2001.

[74] Géradin, M., and Rixen, D. *Mechanical Vibrations – Theory and Application to Structural Dynamics*. John Wiley & Sons, Chichester, England, December 1997.

[75] Gladwell, G. Branch Mode Analysis of Vibrating Systems. *Journal of Sound and Vibration 1* (1964), 41–59.

[76] Global Wind Energy Council (GWEC). Global Wind Report – Annual market update 2011. April 2012.

[77] Golub, G., and van Loan, C. *Matrix Computations*, third edition ed. The John Hopkins Press Ltd., London, UK, 1996.

[78] Golub, G., and Ye, Q. Inexact Inverse Iterations for the Generalized Eigenvalue Problems. *BIT Numerical Mathematics 40*, 4 (2000), 671–684.

[79] Golub, G., and Ye, Q. An inverse free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems. *SIAM J. Sci. Comput. 24*, 1 (2002), 312–334.

[80] Golub, G., Zhang, Z., and Zha, H. Large sparse symmetric eigenvalue problems with homogeneous linear constraints: the Lanczos process with inner-outer iterations. *Linear Algebra and its Applications 309* (2000), 289–306.

[81] Gordis, J. Structural Synthesis in the Frequency Domain: A General Formulation. In *Proceedings of the Twelfth International Modal Analysis Conference, Honolulu, HI* (Bethel, CT, February 1994), Society for Experimental Mechanics, pp. 575–581.

[82] Gordis, J., Bielawa, R., and Flannelly, W. A General Theory for Frequency Domain Structural Synthesis. *Journal of Sound and Vibration 150* (October 1991), 139–158.

[83] Gosselet, P., and Rey, C. On a selective reuse of Krylov subspaces in Newton-Krylov approaches for nonlinear elasticity. In *Fourteenth Conference on Domain Decomposition Methods* (2002), pp. 419–426.

[84] Guo, X. Energy-weighted modes selection in reduced-order nonlinear simulations. In *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* (April 2011), AIAA, Ed., no. AIAA 2011-2063, AIAA.

[85] Guyan, R. Reduction of Stiffness and Mass Matrices. *AIAA Journal 3* (February 1965), 380.

[86] Haghi, R. Integrated Design and Optimization of an Offshore Wind Turbine Monopile Support Structure. Master's thesis, Delft University of Technology, Delft, The Netherlands, 2011.

[87] Hasselmann, K., Barnett, T., Bouws, E., Carlson, H., Cartwright, D., Enke, K., Ewing, J., Gienapp, H., Hasselmann, D., Kruseman, P., Meerburg, A., Mller, P., Olbers, D., Richter, K., Sell, W., and Walden, H. Measurements of wind-wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP). Tech. Rep. 12, Deutsches Hydrographisches Institut, Hamburg, 1973.

[88] Heirman, G. *Model Reduction Techniques to Improve the Efficiency of Flexible Multibody Simulations.* PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 2011.

[89] Heirman, G. H. K., Naets, F., and Desmet, W. A system-level model reduction technique for the efficient simulation of flexible multibody systems. *International Journal for Numerical Methods in Engineering 85*, 3 (2011), 330–354.

[90] Hestnes, M. R., and Stiefel, E. Method of Conjugate Gradients for Solving Linear Systems. *Journal Research of the National Bureau of Standards 49* (1952), 409–438.

[91] Hilber, H., Hughes, T., and Taylor, R. Improved numerical dissipation forthe time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics 5* (1977), 283292.

[92] Hochman, A., Bond, B., and White, J. A stabilized discrete empirical interpolation method for model reduction of electrical, thermal, and microelectromechanical systems. In *Proceedings of the 48th Design Automation Conference* (New York, 2011), pp. 540–545.

[93] Holm-Jørgensen, K., and Nielsen, S. A component mode synthesis algorithm for multibody dynamics of wind turbines. *Journal of Sound and Vibration 326*, 3-5 (2009), 753767.

[94] Hrennikoff, A. Solution of problems of elasticity by the frame-work method. *ASME Journal of Applied Mechanics 8*, 3 (1941), 169–175.

[95] Hughes, T. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Dover Publications, 2000.

[96] Hurty, W. Dynamic Analysis of Structural Systems using Component Modes. *AIAA Journal 3*, 4 (1965), 678–685.

[97] Hurty, W. C. Vibrations of Structural Systems by Component Mode Synthesis. *Journal of Engineering Mechanics, Division American Society of Civil Engineers 86*, 4 (1960), 51–69.

[98] Idelsohn, S., and Cardona, A. A reduction method for nonlinear structural dynamic analysis. *Computer Methods in Applied Mechanics and Engineering 49*, 3 (1985), 253–279.

[99] Ind, P., and Ewins, D. Impedance based decoupling and its application to indirect modal testing and component measurement: A numerical investigation. In *Proceedings of the Twenty First International Modal Analysis Conference, Kissimmee, FL* (Bethel, CT, February 2003), Society for Experimental Mechanics.

[100] International Organization for Standardization (ISO). ISO 6336-1: Calculation of load capacity of spur and helical gears – Part 1: Basic principles, introduction and general influence factors. Tech. rep., Geneva, Switzerland, 2006.

[101] Irons, B. Free-Interface Methods of Substructures Coupling for Dynamic Analysis. *International Journal of Numerical Methods in Engineering 2* (1970), 5–32.

[102] Jakobsson, H. *Adaptive Reduction of Finite Element Models in Computational Solid Mechanics*. PhD thesis, Umeå University, Department of Mathematics and Mathematical Statistics, Sweden, 2011.

[103] Jakobsson, H., Bengzon, F., and Larson, M. G. Adaptive component mode synthesis in linear elasticity. *International Journal for Numerical Methods in Engineering 86*, 7 (2011), 829–844.

[104] Jakobsson, H., and Larson, M. A posteriori error analysis of component mode synthesis for the elliptic eigenvalue problem. *Computer Methods in Applied Mechanics and Engineering 200* (2011), 2840–2847.

[105] Japhet, C., Maday, Y., and Nataf, F. A new Cement to Glue non-conforming Grids with Robin interface conditions: the finite element case. *ArXiv e-prints 705* (May 2007).

[106] Jetmundsen, B., Bielawa, R., and Flannelly, W. Generalized Frequency Domain Substructure Synthesis. *Journal of the American Helicopter Society 33* (January 1988), 55–65.

[107] Jezequel, L. A Method of Damping Synthesis from Substructure Tests. In *Design Engineering Techical Conference* (St. Louis, MO, Sept. 1979), ASME Paper 79-DET-11.

[108] Jonkman, J., Butterfield, S., Musial, W., and Scott, G. Definition of a 5 MW Reference Wind Turbine for Offshore System Development. Tech. Rep. NREL/TP-500-38060, National Renewable Energy Laboratory, 2009.

[109] Junge, M., Brunner, D., Becker, J., and Gaul, L. Interface-reduction for the Craig-Bampton and Rubin method applied to FE-BE coupling with a large fluid-structure interface. *International Journal for Numerical Methods in Engineering 77* (2009), 1731–1752.

[110] Kaufer, D., Cosack, N., Böker, C., Seidel, M., and Kühn, M. Integrated Analysis of the Dynamics of Offshore Wind Turbines with Arbitrary Support Structures. In *Proceedings of the European Wind Energy Conference 2009, Marseille, France* (2009).

[111] Kerschen, G., and Golinval, J. Physical Interpretation of the Proper Orthogonal Modes using the Singular Value Decomposition. *Journal of Sound and Vibration 249*, 5 (2002), 849–865.

[112] Kerschen, G., Golinval, J.-C., Vakakis, A., and Bergman, L. The Method of Proper Orthogonal Decomposition for Dynamical Characterization and Order Reduction of Mechanical Systems: An Overview. *Nonlinear Dynamics 41* (2005), 147–169.

[113] Kerschen, G., Peeters, M., Golinval, J., and Vakakis, A. Nonlinear normal modes, Part I: A useful framework for the structural dynamicist. *Mechanical Systems and Signal Processing 23* (2009), 170–194.

[114] Kirsch, U. *Reanalysis of Structures. A Unified Approach for Linear, Nonlinear, Static and Dynamic Systems*, vol. 151 of *Solid Mechanics and Its Applications*. Springer, The Netherlands, 2008.

[115] Klosterman, A. *On the Experimental Determination and Use of Modal Representations of Dynamic Characteristics*. PhD thesis, University of Cincinnati, Department of Mechanical Engineering, 1971.

[116] Knyazev, A. Toward the optimal preconditioned eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method. *SIAM J. Sci. Comput. 23*, 2 (2001), 517–541.

[117] Krenk, S. Energy conservation in Newmark based time integration algorithms. *Computer Methods in Applied Mechanics and Engineering 195* (2006), 6110–6124.

[118] Krenk, S. *Energy conservation and high-frequency damping in numerical time integration*, vol. Part 1. Taylor & Francis, 2008.

[119] Kubur, M., Kahraman, A., Zini, D. M., and Kienzle, K. Dynamic Analysis of a Multi-Shaft Helical Gear Transmission by Finite Elements: Model and Experiment. *Journal of Vibrations and Acoustics 126* (July 2004), 398–406.

[120] Kuhl, D., and Crisfield, M. Energy-Conserving and Decaying Algorithms in Non-Linear Structural Dynamics. *International Journal for Numerical Methods in Engineering 45* (1999), 569–599.

[121] Lanczos, C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards 45* (1950), 255–282.

[122] Larson, M. A posteriori and a priori error analysis for finite element approximations of self-adjoint elliptic eigenvalue problems. *SIAM Journal on Numerical Analysis 38*, 2 (2001), 608–625.

[123] Lehoucq, R., and Sorensen, D. C. Deflation Techniques for an Implicitly Restarted Arnoldi Iteration. *SIAM J. Matrix Anal. Appl. 17* (1996), 789–821.

[124] Leung, Y.-T. An Accurate Method of Dynamic Condensation in Structural Analysis. *International Journal of Numerical Methods in Engineering 12* (1978), 1705–1715.

[125] Liang, Y., Lee, H., Lim, S., Lin, W., Lee, K., and Wu, C. Proper Orthogonal Decomposition and its Applications – Part I: Theory. *Journal of Sound and Vibration 252* (2002), 527–544.

[126] Lieu, T., Farhat, C., and Lesoinne, M. Reduced-order fluid/structure modeling of a complete aircraft configuration. *Computer Methods in Applied Mechanics and Engineering 195* (2006), 5730–5742.

[127] Lohmann, B., and Salimbahram, B. *Methods and Applications in Automation*. Shaker Verlag, Aachen, 2003, ch. Introduction to Krylov Subspace Methods in Model Order Reduction, pp. 1–13.

[128] Lu, X., McElroy, M. B., and Kiviluomac, J. Global potential for wind-generated electricity. *Proceedings of the National Academy of Sciences of the United States 106*, 27 (2009), 1093310938.

[129] MacNeal, R. A Hybrid Method of Component Mode Synthesis. *Computers & Structures 1*, 4 (1971), 581–601.

[130] Maia, N. *Theoretical and Experimental Modal Analysis*. Mechanical Engineering Research Studies: Engineering Dynamics Series. Wiley-Blackwell, 1997.

[131] Majed, A., Henkel, E., and Wilson, C. Improved Method of Mixed-Boundary Component-Mode Representation for Structural Dynamic Analysis. *AIAA Journal of Spacecraft and Rockets 42* (2005), 825–831.

[132] Martinez., D., Carne, T., Gregory, D., and Miller, A. Combined Experimental/Analytical Modeling Using Component Mode Synthesis. In *Structures, Structural Dynamics and Materials Conference* (Palm Springs, CA, May 1984), 25th AIAA/ASME/ASCE/AHS, pp. 140–152. AIAA Paper 84-0941.

[133] Masson, G., Brik, B. A., Cogan, S., and Bouhaddi, N. Component Mode Synthesis (CMS) Based on an Enriched Ritz Approach for Efficient Structural Optimization. *Journal of Sound and Vibration 296*, 4-5 (October 2006), 845–860.

[134] Mayes, R., and Stasiunas, E. Combining Lightly Damped Experimental Substructures with Analytical Substructures. In *Proceedings of the Twentyfifth International Modal Analysis Conference, Orlando, FL* (Bethel, CT, 2007), Society for Experimental Mechanics.

[135] Meyer, M., and Matthies, H. G. Efficient model reduction in non-linear dynamics using the Karhunen-Loève expansion and dual-weighted-residual methods. *Computational Mechanics* (2003), 179–191.

[136] Mignolet, M., Przekop, A., Rizzi, S., and Spottswood, S. A review of indirect/non-intrusive reduced order modeling of non-linear geometric structures. *Jnl. Sound and Vibration* (submitted 2011).

[137] Molenaar, D-P. *Cost-effective Design & Operation of Variable Speed Wind Turbines: Closing the Gap between the Control Engineering & Wind Engineering Community*. PhD thesis, Delft University of Technology, Delft, the Netherlands, February 2003.

[138] Morison, J., O'Brien, M., Johnson, J., and Schaaf, S. The force exerted by surface waves on piles. *Petroleum Transactions (American Institute of Mining Engineers) 189* (1950), 149–154.

[139] NASA. NASA-STD-5002, Loads Analyses of Spacecraft and Payloads. Tech. Rep. NASA-STD-5002, 1996.

[140] Newmark, N. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division of ASCE 85 (EM3)* (1959), 67–94.

[141] Nicgorski, D., and Avitabile, P. Conditioning of FRF measurements for use with frequency based substructuring . *Mechanical Systems and Signal Processing 24*, 2 (2010), 340–351.

[142] Nortier, B. Residual Vectors & Error Estimation in Substructure based Model Reduction. Master's thesis, Delft University of Technology, Delft, The Netherlands, September 2011.

[143] Nortier, B. P., Voormeeren, S. N., and Rixen, D. J. Application of residual vectors to superelement modeling of an offshore wind turbine foundation. In *Topics in Experimental Dynamics Substructuring and Wind Turbine Dynamics, Volume 2* (2012), vol. 27 of *Conference Proceedings of the Society for Experimental Mechanics Series*, Springer New York, pp. 149–163.

[144] O'Callahan, J., Avitabile, P., and Riemer, R. System Equivalent Reduction Expansion Process (SEREP). In *Proceedings of the Seven International Modal Analysis Conference, Las Vegas, NV* (Bethel, CT, February 1989), Society for Experimental Mechanics, pp. 29–37.

[145] O'Leary, D. P. The block conjugate gradient algorithm and related methods. *Linear Algebra and its Applications 29* (February 1980), 293–322. doi:10.1016/0024-3795(80)90247-5.

[146] Park, K. C., Justino, M., and Felippa, C. An algebraically partitioned FETI method for parallel structural analysis: algorithm description. *International Journal for Numerical Methods in Engineering 40* (1997), 2717–2737.

[147] Parks, M. L., Sturler, E. D., Mackey, G., Johnson, D. D., and Maiti, S. Recycling Krylov Subspaces for Sequences of Linear Systems. Tech. rep., SIAM J. Sci. Comput, 2004.

[148] Passon, P., Kühn, M., Butterfield, S., Jonkman, J., Camp, T., and Larsen, T. OC3-Benchmark Exercise of Aero-elastic Offshore Wind Turbine Codes. *Journal of Physics: Conference Series 75* (2007), 1–12.

[149] Peeters, J., Vandepitte, D., and Sas, P. Analysis of Internal Drive Train Dynamics in a Wind Turbine. *Wind Energy 9* (2006), 141–161.

[150] Peeters, M., Kerschen, G., and Golinval, J. Dynamic testing of nonlinear vibrating structures using nonlinear normal modes. *Journal of Sound and Vibration 330* (2011), 486–509.

[151] Peeters, M., Viguié, R., Sérandour, G., Kerschen, G., and Golinval, J.-C. Nonlinear normal modes, Part II: Toward a practical computation using numerical continuation techniques. *Mechanical Systems and Signal Processing 23* (2009), 195–216.

[152] Qu, Z.-Q. *Model Order Reduction Techniques – With Applications in Finite Element Analysis*. Springer-Verlag, London, 2004.

[153] Quillen, P. *Generalizations of an inverse-free Krylov subspace algorithm for the symmetric generalized eigenvalue problem*. PhD thesis, University of Kentucky, 2005.

[154] Quillen, P., and Ye, Q. A block inverse-free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems. *Journal of Computational and Applied Mathematics 233* (2010), 1298–1313.

[155] Ramani, A., and Knight, C. E. Two-step component-mode synthesis for the eigensolution of large systems. *AI 34*, 7 (July 1996), 1529–1525.

[156] Rao, V., Chaudhuri, S., and Gupta, V. Mode-Acceleration Approach to Seismic Response of Multiply-Supported Secondary Systems. *Earthquake Engineering & Structural Dynamics 31* (2002), 1603–1621.

[157] Rayleigh, J. *The Theory of Sound*. Dover Publications, 1896.

[158] Ren, Y., and Beards, C. A Generalized Recepetance Coupling Technique. In *Proceedings of the Eleventh International Modal Analysis Conference, Orlando, FL* (Bethel, CT, February 1993), Society for Experimental Mechanics, pp. 868–871.

[159] Reuss, P., Zeumer, B., Herrmann, J., and Gaul, L. Consideration of Interface Damping in Dynamic Substructuring. In *Proceedings of IMAC XXX, Jacksonville, FL* (2012), Society for Experimental Mechanics.

[160] Rey, C., Devries, F., and Lene, F. Parallelism in non linear computation of heterogeneous structures. *Calculateurs Parallèles 7*, 3 (1995).

[161] Risler, F., and Rey, C. On the reuse of Ritz vectors for the solution to nonlinear elasticity problems by domain decomposition methods. In *Tenth International Conference on Domain Decomposition Methods* (1998), Contemporary Mathematics, American Mathematical Soc.

[162] Rixen, D. Dual Schur Complement Method for Semi-Definite Problems. *Contemporary Mathematics 218* (1998), 341–348.

[163] Rixen, D. Generalized Mode Acceleration Methods and Modal Truncation Augmentation. In *Structures, Structural Dynamics and Material Conference and Exhibit* (Seattle, WA, USA, April 2001), 42st AIAA/ASME/ASCE/AHS/ASC. AIAA, 2001-1300.

[164] Rixen, D. A Lanczos Procedure for Efficient Mode Superposition in Dynamic Analysis. In *43st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Material Conference and Exhibit* (Denver, USA, April 2002). AIAA 2002-1393.

[165] Rixen, D. High Order Static Correction Modes for Component Mode Synthesis. In *Fifth World Congress on Computational Mechanics* (Vienna, Austria, July 2002). http.//wccm.tuwien.ac.at, isbn 3-9501554-0-6.

[166] Rixen, D. Dual Craig-Bampton with enrichment to avoid spurious modes. In *Proceedings of the Twenty Seventh International Modal Analysis Conference, Orlando, FL* (Bethel, CT, February 2009), Society for Experimental Mechanics.

[167] Rixen, D. Modal Truncation Augmentation and Moment Matching: two faces of the same coin. In *Autumn School on Future Developments in Model Order Reduction* (Terschelling, The Netherlands, September 2009).

[168] Rixen, D. A Substructuring Technique Based on Measured and Computed Impulse Response Functions of Components. In *Proceedings of the International Conference on Noise and Vibration Engineering (ISMA 2010), Leuven, Belgium, 20-22 September 2010* (2010).

[169] Rixen, D. Substructuring using Impulse Response Functions for Impact Analysis. In *Proceedings of the IMAC - XXVIII, Jacksonville, Florida USA* (2010).

[170] Rixen, D. Interface Reduction in the Dual Craig-Bampton method based on dual interface modes. In *Proceedings of the IMAC - XXIV, Jacksonville, Florida USA* (Bethel, CT, February 2011), Society for Experimental Mechanics.

[171] Rixen, D. J. *Substructuring and Dual Methods in Structural Analysis*. PhD thesis, Université de Liège, 1997.

[172] Rixen, D. J. A dual Craig-Bampton method for dynamic substructuring. *Journal of Computational and Applied Mathematics 168* (2004), 383–391.

[173] Rixen, D. J., and Haghighat, N. Truncating the Impulse Responses of Substructures to Speed Up the Impulse-Based Substructuring. In *Proceedings of the SEM IMAC XXX Conference Jan. 30 - Feb. 2, 2012, Jacksonville, FL USA* (2012).

[174] Roux, F.-X. Parallel implementation of a domain decomposition method for nonlinear elasticity problems. In *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering* (1995), D. Keyes, Y. Saad, and D. Truhlar, Eds., SIAM, pp. 161–175.

[175] Roy, N., Germès, S., Lefevre, B., and Balmès, E. Damping Allocation in Automotive Structures using Reduced Models. In *Proceedings of ISMA 2006, Leuven, Belgium* (2006).

[176] Rubin, S. Improved Component-Mode Representation for Structural Dynamic Analysis. *AIAA Journal 13* (1975), 995–1006.

[177] Saad, Y. On the Lanczos method for solving symmetric linear systems with several right-hand sides. *Math. Comput. 48* (1987), 651–662.

[178] Saad, Y. *Iterative Methods for Sparse Linear Systems*, second edition ed. Society for Industrial and Applied Mathematics, 2003.

[179] Saad, Y., and Schultz, M. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comput. 7*, 3 (July 1986), 856–869.

[180] Schaumann, P., Dubois, J., Achmus, M., Abdel-Rahman, K., and Seidel, M. Local Dynamics of Jacket Support Structures for Offshore Wind Turbines. In *Proceedings of EWEA Offshore 2011, Amsterdam, Netherlands* (2011).

[181] Schilders, W., van der Vorst, H., and Rommes, J., Eds. *Model Order Reduction: Theory, Research Aspects and Applications*. Mathematics in Industry, Vol. 13. Springer, 2008.

[182] Segalman, D. J. Model reduction of systems with localized nonlinearities. *Journal of Computational and Nonlinear Dynamics 2*, 6 (2007), 249–266. Transactions of the ASME.

[183] Seidel, M., and Foss, G. Impact of different substructures on turbine loading and dynamic behaviour for the DOWNVInD Project in 45m water depth. In *Proceedings of the European Wind Energy Conference 2006, Athens, Greece* (2006).

[184] Seidel, M., Ostermann, F., Curvers, A., Kühn, M., Kaufer, D., and Böker, C. Validation of Offshore load simulations using measurement data from the DOWNVInD project. In *Proceedings of European Offshore Wind Conference 2009, Stockholm, Sweden* (2009).

[185] Seidel, M., von Mutius, M., Rix, P., and Steudel, D. Integrated analysis of wind and wave loading for complex support structures of Offshore Wind Turbines. In *Proceedings of European Offshore Wind Conference 2005, Copenhagen, Denmark* (2005).

[186] Shabana, A. A. *Dynamics of Multibody Systems*, third ed. Cambridge University Press, Cambridge, United Kingdom, 2005.

[187] Shyu, W.-H., Gu, J., Hulbert, G. M., and Ma, Z.-D. On the use of Multiple Quasi-Static Mode Compensation Sets for Component Mode Synthesis of Complex Structures. *Finite Elements in Analysis and Design 35* (2000), 119–140.

[188] Shyu, W.-H., Ma, Z.-D., and Hulbert, G. M. A new component mode synthesis method: Quasi-static mode compensation. *Finite Elements Anal. Des. 24*, 4 (1997), 197–298.

[189] Simo, J. C., Wriggers, P., and Taylor, R. L. A perturbed Lagrangian formulation for the finite element solution of contact problems. *Computer Methods in Applied Mechanics and Engineering 50* (1985), 163–180.

[190] Sjövall, P. *Identification and Synthesis of Components for Vibration Transfer Path Analysis*. PhD thesis, Chalmers University of Technology, Göteborg, Sweden, October 2007.

[191] Sjövall, P., and Abrahamsson, T. Component System Identification and State-Space Model Synthesis. *Mechanical Systems and Signal Processing 21* (2007), 2697–2714.

[192] Sjövall, P., and Abrahamsson, T. Substructure System Identification from Coupled System Test Data. *Mechanical Systems and Signal Processing 22* (June 2007), 15–33.

[193] Sleijpen, G. L., and van der Vorst, H. A. An Overview of Approaches for the Stable Computation of Hybrid BiCG Methods. *Applied Numerical Mathematics 19*, 3 (December 1995), 235–254. Special Issue on Iterative Methods for Linear Equations.

[194] Sofrin, T. The combination of dynamical systems. *Journal of the Aeronautical Sciences 13*, 6 (1946), 281–288.

[195] Sorensen, D. Implicit applications of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. Appl. 13* (1992), 357–385.

[196] Su, T.-J., and Craig, R. Model Reduction and Control of Flexible Structures Using Krylov Vectors. *AIAA Journal of Guidance, Control, and Dynamics 14*, 2 (March-April 1991), 260–267.

[197] Subroto, H., and Godfroy, P. Optimization of the integrated tower and monopile support structure – A first step in the elaborate path to integrated design optimization of an OWT support structure. In *Proceedings of TORQUE 2010, the science of making torque from wind* (2010).

[198] Tiso, P., and Rixen, D. Reduction methods for MEMS nonlinear dynamic analysis. In *Nonlinear Modeling and Applications, Volume 2* (2011), vol. 11 of *Conference Proceedings of the Society for Experimental Mechanics Series*, Springer New York, pp. 53–65.

[199] Tosseli, A., and Widlund, O. *Domain Decomposition Methods – Algorithms and Theory*, vol. 34. Springer, 2004.

[200] Tran, D.-M. Méthodes de Synsthèse Modale Mixtes. *Revue Européenne des éléments finis, 1*, 2 (1992), 137–179.

[201] Tran, D.-M. Component Mode Synthesis Methods Using Interface Modes: Application to Structures with Cyclic Symmetry. *Computers & Structures 79* (2001), 209–222.

[202] Tran, D.-M. Component mode synthesis methods using partial interface modes: Application to tuned and mistuned structures with cyclic symmetry. *Computers & Structures 87* (2009), 1141–1153.

[203] Troeng, T. Frequency Response Analysis using Component Mode Synthesis. Master's thesis, Umeå University, Department of Mathematics and Mathematical Statistics, Sweden, 2010.

[204] U.S. Energy Information Administration. Levelized Cost of New Generation Resources in the Annual Energy Outlook 2012. Tech. Rep. DOE/EIA-0383(2012), June 2012.

[205] van der Tempel, J. *Design of Support Structures for Offshore Wind Turbines*. PhD thesis, Delft University of Technology, 2006.

[206] van der Valk, P. Model Reduction & Interface Modeling in Dynamic Substructuring. Master's thesis, Delft University of Technology, Delft, the Netherlands, January 2010.

[207] van der Valk, P., and Rixen, D. An Effective Method for Assembling Impulse Response Functions to Linear and Non-Linear Finite Element Models. In *Topics in Experimental Dynamics Substructuring and Wind Turbine Dynamics, Volume 2* (2012), vol. 27 of *Conference Proceedings of the Society for Experimental Mechanics Series*, Springer New York, pp. 123–135.

[208] van der Valk, P., and Voormeeren, S. An Overview of Modeling Approaches for Complex Offshore Wind Turbine Support Structures. In *Proceedings of ISMA 2012, Leuven, Belgium* (Leuven, Belgium, September 2012).

[209] Voormeeren, S., de Klerk, D., and Rixen, D. Uncertainty Quantification in Experimental Frequency Based Substructuring. *Mechanical Systems and Signal Processing 24*, 1 (2010), 106–118. doi:10.1016/j.ymssp.2009.01.016.

[210] Voormeeren, S., den Dekker, D., and Nortier, B. Dynamic Substructuring Toolbox – User Manual. Tech. rep., Delft University of Technology & Siemens Wind Power, The Netherlands, 2011.

[211] Voormeeren, S., and Rixen, D. Substructure Decoupling Techniques – a Review and Uncertainty Propagation Analysis. In *Proceedings of the Twenty Seventh International Modal Analysis Conference, Orlando, FL* (Bethel, CT, February 2009), Society for Experimental Mechanics.

[212] Voormeeren, S., and Rixen, D. A Dual Approach to Substructure Decoupling Techniques. In *Structural Dynamics, Volume 3* (2010), vol. 12 of *Conference Proceedings of the Society for Experimental Mechanics Series*, Springer New York, pp. 601–616.

[213] Voormeeren, S., and Rixen, D. A Family of Substructure Decoupling Techniques Based on a Dual Assembly Approach. *Mechanical Systems and Signal Processing 27* (2012), 379–396.

[214] Voormeeren, S., and Rixen, D. Efficient Updating of Static Modes in the Craig-Bampton Reduction Basis. In *Topics in Experimental Dynamics Substructuring and Wind Turbine Dynamics, Volume 2* (2012), vol. 27 of *Conference Proceedings of the Society for Experimental Mechanics Series*, Springer New York, pp. 299–317.

[215] Voormeeren, S., and Rixen, D. Updating Component Reduction Bases of Static and Vibration Modes Using Preconditioned Iterative Techniques. *Computer Methods in Applied Mechanics and Engineering* (2012). Accepted for publication, 10 October 2012.

[216] Voormeeren, S., van der Valk, P., Nortier, B., Molenaar, D-P., and Rixen, D. Accurate and Efficient Modeling of Complex Offshore Wind Turbine Support Structures using Augmented Superelements. *Wind Energy* (2012). Revision submitted, October 2012.

[217] Voormeeren, S., van der Valk, P., and Rixen, D. A General Mixed Boundary Model Reduction Method for Component Mode Synthesis. *IOP Conf. Series: Materials Science and Engineering 10*, 1 (2010).

[218] Voormeeren, S., van der Valk, P., and Rixen, D. Practical Aspects of Dynamic Substructuring in Wind Turbine Engineering. In *Structural Dynamics and Renewable Energy, Volume 1* (2010), vol. 10 of *Conference Proceedings of the Society for Experimental Mechanics Series*, Springer New York, pp. 163–185.

[219] Voormeeren, S., van der Valk, P., and Rixen, D. Generalized Methodology for Assembly and Reduction of Component Models for Dynamic Substructuring. *AIAA Journal 49*, 5 (May 2011), 1010–1020.

[220] Vorpahl, F., and Popko, W. IEA Wind Annex 30 – OC4 Project. The Offshore Code Comparison Collaboration Continuation. In *IEA Wind Side Event at EWEA 2011 Annual Event* (Brussels, Belgium, March 2011).

[221] Vorpahl, F., Popko, W., and Kaufer, D. Description of a basic model of the ”Upwind reference jacket” for code comparison in the OC4 project under IEA Wind Annex XXX. Tech. rep., 2011.

[222] Wilson, E., Yuan, M., and Dickens, J. Dynamic Analysis by Direct Superposition of Ritz Vectors. *Earthquake Engineering & Structural Dynamics 10* (1982), 813–821.

[223] Witteveen, W., and Irschik, H. Efficient modal formulation for vibration analysis of solid structures with bolted joints. In *Proceedings of the Twenty Fifth International Modal Analysis Conference, Orlando, FL* (Bethel, CT, February 2007), Society for Experimental Mechanics.

[224] Witteveen, W., and Irschik, H. Efficient Mode-Based Computational Approach for Jointed Structures: Joint Interface Modes. *AIAA Journal 47*, 1 (January 2009), 252–263.

[225] World Wind Energy Association (WWEA). World Wind Energy Report 2011. April 2012.

[226] Ziekman, R.-W. Dynamic modelling of a planetary gearbox of a wind turbine. Master’s thesis, Delft University of Technology, faculty of 3mE, June 2006. - Confidential -.

PART IV

Appendices

A

The Dynamic Substructuring Toolbox

One task of the work reported in this thesis was to implement the dynamic substructuring methods and procedures, such that they can be used in a practical setting. This is a non-trivial task that involves a lot of "bookkeeping", especially when generality of the tools is required. Over the course of this PhD project a toolbox was therefore developed, which contains most of the methods treated in chapters 2, 3 and 4. This Matlab based toolbox, called the *Dynamic Substructuring Toolbox* (DS Toolbox), employs a data format that is specifically designed to simplify the handling of component models that can be modeled by finite elements, by analytical methods or from experimental data. The actual toolbox consists of three separate *graphical user interfaces* (GUIs), namely:

- The *Preparation Tool* (PrepTool) is built to import finite element models from popular FE codes, using the FEMLink toolbox (which in turn is part of the commercial SD Toolbox, see [17]), and cast them in the correct data format. Along with the FE data, a Matlab file should be created by the user describing interface properties of the model. Hereafter, model reduction techniques can be applied to create a superelement.

- The *Assembly Tool* (AssemblyTool) is used to assemble the (reduced) component models. The component models are loaded and hereafter the user can select which interfaces of the different component models are to be assembled. Interface reduction can be applied to the assembled model if the number of interface DoF is considered too high.

- The *Postprocessing Tool* (PostTool) is developed to analyze components and assemblies, respectively created in the PrepTool and AssemblyTool. Two structures can be analyzed simultaneously and the results can be compared. Various types of analysis can be performed, such as a modal analysis, frequency response comparison and time integration.

Next, section A.1 treats the functionality of the PrepTool in further detail. Section A.2 explains the AssemblyTool and finally section A.3 addresses the PostTool. These descriptions are a summary of the more extensive user manual [210], which also contains a step by step *walk through* introducing the new user to the functionality of the tools.

A.1 Preparation Tool

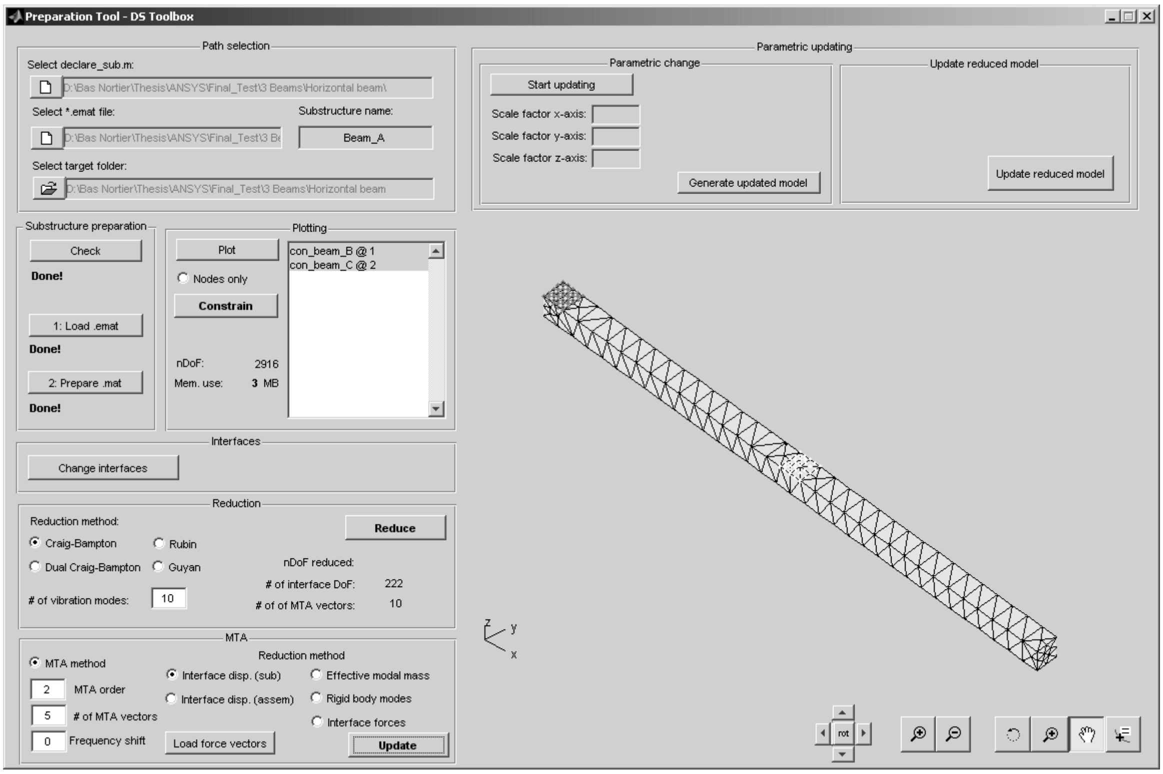
The PrepTool is used to import FE models and to transform them into component models with the DS Toolbox data structure, its GUI is shown in figure A.1. The functionality of this tool will be briefly described in this section.

Preparation & Import The first step in the use of the PrepTool is to select the correct file and folder in the `Path selection`. Firstly, the input file containing the interface information, such as the number and geometry of the interfaces, has to be selected. This file is named `Declare_sub.m` and is specifically created for each substructure. If the component model is created using FE software one must additionally select a data file type `.emat`, accompanied with a `.rst` and `.mode` file which are used by FEMLink (for a description of the FEMLink toolbox functionality see [17]). The selected target folder is used as an output folder during the following steps. Next in the `Substructure preparation` section the `Check` button can be used to check if all the necessary data files are present in the selected folders, if so the user is allowed to continue to the next step. Based on the data contained in `Declare_sub.m` the `Load .emat` button is enabled which starts the data conversion using FEMLink. Hereafter the `Prepare .mat` button can be pressed which creates a compatible data structure for the component model.

Checking & Visualization The `Plotting` section allows the user to visualize and check the component model and its interfaces. Various element types can be plotted, such as beams, shells, tetrahedron etc. A variety of plotting tools are incorporated in the docked plotting window such as rotate, zoom, move, data cursor, etc. The nodes belonging to the selected interfaces are highlighted using colors. A useful option, the `Constrain` button, allows the user to constrain the structure at the selected interface. This allows natural boundary conditions to be applied to the substructure models. If during the visual check it is found that the specified interface information is incorrect one can efficiently change this using the `Change interfaces` button. This opens the `Declare_sub.m` file to allow direct altering. Hereafter the user is guided to restart at the substructure preparation since the component model information has been changed.

Model Reduction In the `Reduction` section the user can choose any of the reduction methods discussed in section 2.5 to generate a reduced model. Next, the number of vibration modes to be included in the basis has to be specified; hereafter the reduction process can be started using the `Reduce` button. In case it wished to augment the reduction basis using MTAs (see section 2.7), this can be specified in the `MTA` section. The user input for this method is the desired order of MTAs and optionally the method to reduce the number of MTAs, the reduced number of MTA vectors per order and the frequency shift. If some external excitation on the component model is known and associated spatial force vectors exist, these can be loaded by pressing the `Load force vectors` button. These vectors are added to static response vectors used to compute the MTAs.

Design Modification & Updating The `Parametric updating` section is a functionality that is still in development. It allows the user to change certain parameters of the component model (such as the width, length or height) and hereafter iteratively update its reduced mass and stiffness matrices using the methods of chapter 6. This allows the user to make (small) design changes and efficiently produce an updated component model.


Preparation Tool - DS Toolbox

Select declare_sub.m: D:\Bas Nortier\Thesis\ANSYS\Final_Test\3 Beams\Horizontal beam\

Select *.emat file: D:\Bas Nortier\Thesis\ANSYS\Final_Test\3 Be

Substructure name: Beam_A

Select target folder: D:\Bas Nortier\Thesis\ANSYS\Final_Test\3 Beams\Horizontal beam

Check — Done!

1: Load .emat — Done!

2: Prepare .mat — Done!

○ Nodes only

con_beam_B @ 1
con_beam_C @ 2

nDoF: 2916
Mem. use: 3 MB

Reduction method:
● Craig-Bampton ○ Rubin
○ Dual Craig-Bampton ○ Guyan
# of vibration modes: 10

nDoF reduced:
# of interface DoF: 222
# of of MTA vectors: 10

● MTA method
2 MTA order
5 # of MTA vectors
0 Frequency shift

Reduction method
● Interface disp. (sub) ○ Effective modal mass
○ Interface disp. (assem) ○ Rigid body modes
○ Interface forces

Scale factor x-axis:
Scale factor y-axis:
Scale factor z-axis:

Figure A.1: Graphical user interface of the PrepTool

A.2 Assembly Tool

If the substructures are properly created and the interfaces are correctly identified, the AssemblyTool can be used to build assembled models. Its GUI is shown in figure A.2.

Loading Component Models The `Load substructures` section allows the user to load or delete components models using the `Load` and `Delete` buttons, respectively. The tool also allows to import and export complete assemblies via the `Import -/Export assembly` buttons. The memory use is also indicated. The `Model summary` section provides information of the loaded component models such as the unreduced number of DoF (`NDof`), the sparsity of the matrices (`K-sparsity`, `M-sparsity`, `C-sparsity`, respectively), the reduced number of DoF (`Ndof reduced`), the size of the FRF (`FRF size`).

Assembly Options In the `Assembly options` section the required options for the assembly of the substructures can be set. The first option is in which domain the component models have to be assembled, either the time or frequency domain. The `clear other domain` option deletes previous couplings to save memory and required storage space. The option `Copy coupling(s) to other domain` copies the couplings from the selected domain to the other. The component models can be assembled per node or per interface (`Assembly per node` or `Assembly per interface`, respectively), and in specified directions only (`Assembly DoF`). If assembly via interfaces is selected the user can choose the algorithm used for node collocation, selecting either `Normal` or `Least squares`, see section 3.6.

Defining Couplings After choosing the correct settings, the couplings between the component models can be defined and visualized in the `Define couplings section`. Two substructures can be selected which are also plotted. Two tables display the available connections per model. Depending on the settings this list either shows individual nodes or complete interfaces. From the tables the connection for each model can be selected, which is highlighted in the plotting window, and submitted using `Submit coupling(s)`. The coupling then appears in the `Assembly` section table. Note that the coupling section allows to couple either the full or reduced component model. Erroneous couplings can be corrected by selecting the coupling(s) and pressing `Delete coupling(s)`. Further visualization can be obtained using the `Preview assembly` which plots the assembly in an external window.

Performing Assembly The second `Assembly options` section has a number of options. The `Reload substructures` reloads the substructures after the assembly is performed, for instance to create another assembly using different settings. The `Delete substructures` discards the loaded component model information to minimize memory use. The assembly can be further reduced by applying interface reduction using the `Apply interface reduction` option (see sections 4.4 and 4.5); the user must specify the number of interface modes to be used in the input field.

The `Assemble` button is pressed to truly assemble the component models. In case only substructures with either stiffness or flexibility interface representations are to be connected, primal assembly is applied by default to obtain an assembled a model with a minimum number of DoF (see sections 3.3 and 3.4). Mixed assembly is automatically applied when both stiffness and flexibility type of components are connected, see section 3.5. Finally, when applying the Mixed Craig-Bampton method (see section 2.6), the loaded substructures are both reduced (with a number of modes to be specified in a pop-up window) and assembled using the `Mixed CB reduction & assembly` button. As the name suggests, the reduced components are connected using mixed assembly.

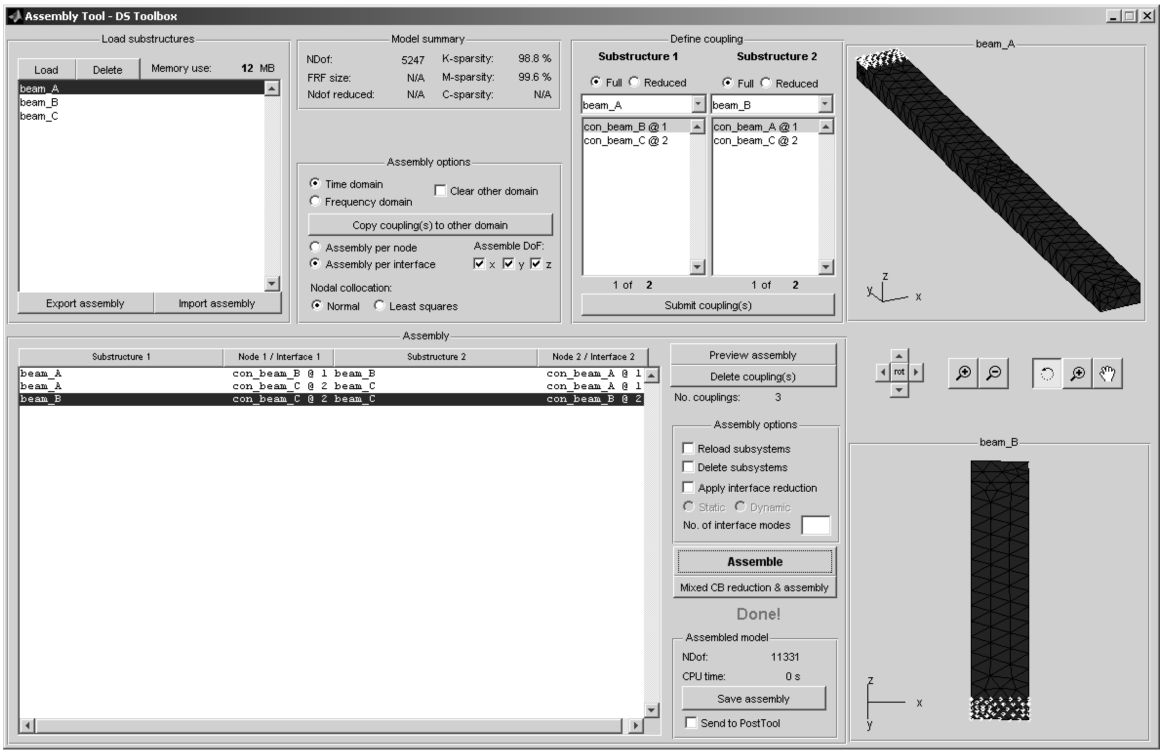
Saving Data In the `Assembled model` section information is given on the assembled model, such as the number of DoF and the CPU time to needed for assembly. Using the `Save assembly` the newly assembled model can be saved to a desired location. By checking the option `Send to PostTool` the PostTool is opened with the assembly already loaded after it is saved.

A.3 Postprocessing Tool

The third and last part of the DS Toolbox is the PostTool which is used to analyse the (reduced) component models and (reduced) assemblies, the GUI is displayed in figure A.3.

Loading & Selecting Models In the `Load systems` two systems from either the PrepTool or the AssemblyTool can be loaded. Information on the loaded systems is displayed is this section, such as its name, the number of DoF or the (reduced) system (`NDoF`), the number of DoF when expanded (`NDoF expanded`) and, if available, the size of the FRF (`FRF size`). When assemblies are loaded the two tables display its component models. The `Active` and `All` checkboxes are respectively used to select which system(s) and component model(s) are to be analyzed and/or plotted.

Modal Analysis The `Modal Analysis` section enables modal analysis of the models. When a reduced model is analyzed the `Expand` option can be checked to expand the reduced eigenmodes such that they can be visualized and/or correlated to unreduced results.



Assembly Tool - DS Toolbox

Load substructures

Memory use: 12 MB

beam_A
beam_B
beam_C

Model summary

NDof: 5247
FRF size: N/A
Ndof reduced: N/A
K-sparsity: 98.8 %
M-sparsity: 99.6 %
C-sparsity: N/A

Assembly options

● Time domain
○ Frequency domain
☐ Clear other domain
○ Assembly per node
● Assembly per interface
Assemble DoF: ☑ x ☑ y ☑ z
Nodal collocation: ● Normal ○ Least squares

Define coupling

Substructure 1: ● Full ○ Reduced
beam_A
con_beam_B @ 1
con_beam_C @ 2
1 of 2

Substructure 2: ● Full ○ Reduced
beam_B
con_beam_A @ 1
con_beam_C @ 2
1 of 2

Assembly

Substructure 1	Node 1 / Interface 1	Substructure 2	Node 2 / Interface 2
beam_A	con_beam_B @ 1	beam_B	con_beam_A @ 1
beam_A	con_beam_C @ 2	beam_C	con_beam_A @ 1
beam_B	con_beam_C @ 2	beam_C	con_beam_B @ 2

No. couplings: 3

Assembly options

☐ Reload subsystems
☐ Delete subsystems
☐ Apply interface reduction
○ Static ○ Dynamic
No. of interface modes

Done!

Assembled model

NDof: 11331
CPU time: 0 s
☐ Send to PostTool

Figure A.2: Graphical user interface of the AssemblyTool

The number of modes to be computed can be specified in the `No. modes [-]` box. If the loaded system(s) possess a damping matrix, the checkbox `Complex modes` can be checked to allow the algorithm to compute complex valued eigenmodes. By pressing `Modal analysis` the eigensolutions of the system(s) are computed. When these results are obtained the eigenfrequencies are displayed in the listboxes.

Visualization & Plotting Tools One of the eigenfrequencies of the system(s) can be selected, when the button `Plot` is pressed the selected component model(s) of the system(s) are plotted in the plot area. By toggling the `Active` checkboxes in the `Load systems` section and selecting component models in the listboxes, the user can choose which part(s) of the system(s) are plotted. By pressing the `Animate` button the system is animated to deform in the selected eigenmode.

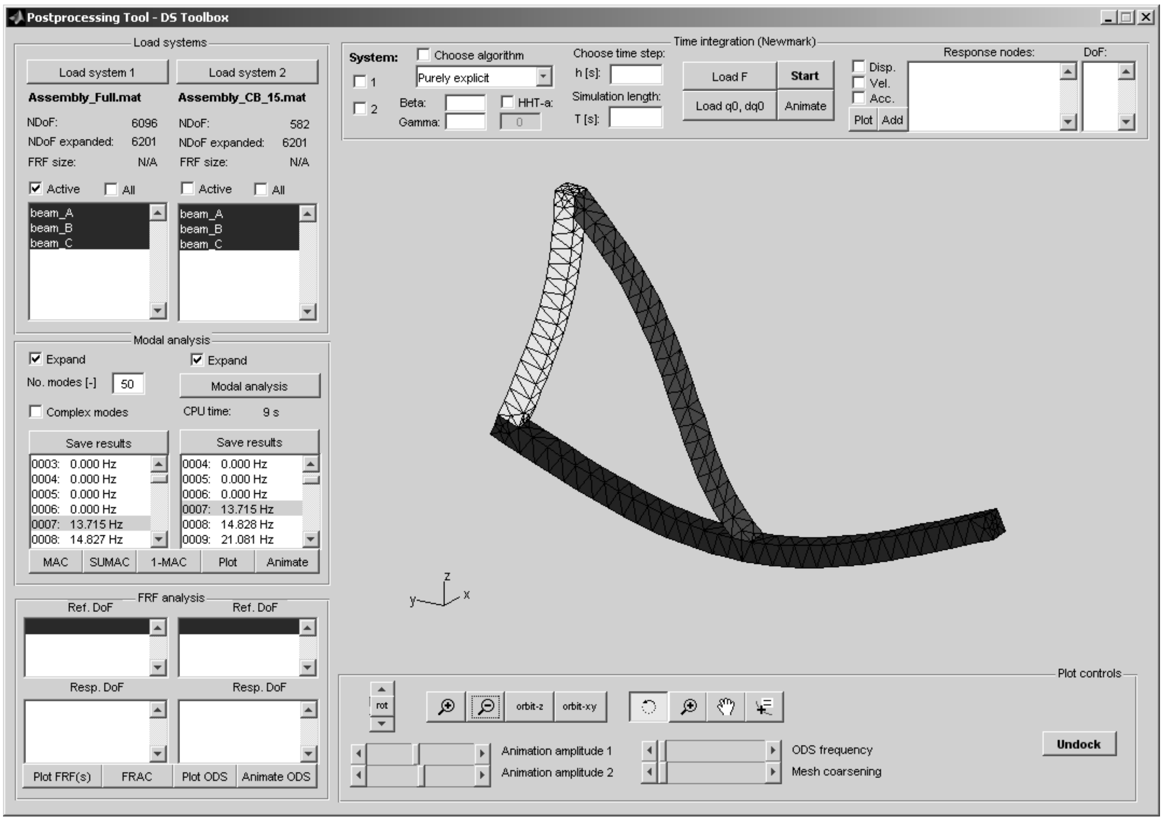
In the `Plot controls` section the amplitude of deformation can be scaled using the slider bars denoted by `Animation amplitude`. Furthermore, in the `Plot controls` section the user can, amongst others, rotate, move, orbit and zoom the plotted model. The mesh at which the model is shown can be coarsened to reduce plotting time (which can be necessary for large FE models) using the `Mesh coarsening` slider. Finally the `Undock` button can be used to undock the current plot into an external window, allowing it to be saved and/or other Matlab plotting tools to be used.

Correlation Direct comparisons between all eigenmodes can be performed via the MAC or SUMAC (see section 4.7), using `MAC` or `SUMAC`, where the results are plotted in the plot area. The `1-MAC` option plots the error of the diagonal of the MAC matrix on a semi logarithmic scale. Finally the `Save results` allows the user to save the eigenfrequencies and eigenmodes, for instance to export or perform further analysis elsewhere.

Time Integration The `Time integration (Newmark)` section allows the user to perform time integration of the model. First the user can specify which system(s) to be analyzed using the checkboxes. Hereafter the user can specify values for the `Beta` and `Gamma` parameters of the Newmark time integration scheme. Alternatively, the user can selected from several standard forms of the algorithm by checking `Choose algorithm` and selecting the desired version. By checking the `HHT-a:` the Newmark scheme is modified using to the Hilber-Hughes-Taylor α-method which introduces numerical damping in the method without degrading the order of the accuracy. The value for α has to be specified.

The time step and simulation length has to be specified in `h [s]` and `T [s]`, respectively. Note that the allowable time step is dependent on the values for β and γ for the Newmark scheme. Initial conditions for the system's displacements and velocities can be loading using the `Load q0, dq0` button. The external (time-varying) force can be loaded using the `Load F` button. Pressing the `Start` button initiates the time integration algorithm.

The `Animate` button initiates and stops the animations of the computed response of the system. Furthermore, using the `Disp.`, `Vel.` and `Acc.` checkboxes the user can choose between plotting displacement, velocity or acceleration responses, respectively. From the listbox beneath `Response nodes:` the user can select a certain node of interest and hereafter select the desired `DoF` for the listbox beneath `DoF:`. Using the `Add` and `Plot` the selected response can be added and plotted.



Postprocessing Tool - DS Toolbox

Assembly_Full.mat
NDoF: 6096
NDoF expanded: 6201
FRF size: N/A
☑ Active ☐ All
beam_A
beam_B
beam_C

Assembly_CB_15.mat
NDoF: 582
NDoF expanded: 6201
FRF size: N/A
☐ Active ☐ All
beam_A
beam_B
beam_C

☑ Expand
No. modes [-] 50
☐ Complex modes
0003: 0.000 Hz
0004: 0.000 Hz
0005: 0.000 Hz
0006: 0.000 Hz
0007: 13.715 Hz
0008: 14.827 Hz

☑ Expand
CPU time: 9 s
0004: 0.000 Hz
0005: 0.000 Hz
0006: 0.000 Hz
0007: 13.715 Hz
0008: 14.828 Hz
0009: 21.081 Hz

System: ☐ 1 ☐ 2
☐ Choose algorithm
Purely explicit
Beta:
Gamma:
☐ HHT-a: 0
h [s]:
T [s]:
☐ Disp.
☐ Vel.
☐ Acc.

Figure A.3: Graphical user interface of the PostTool

B

Component Validation Measurements

In chapter 7 the dynamics of the yaw system of the Siemens 2.3 MW wind turbine were analyzed. Parallel to the numerical modeling of the components, efforts were spent validating some component models and subassemblies. These efforts are described in this appendix. Specifically, the validation of the important bedplate component will be treated next, while the attempts to validate the yaw gearbox model will be described in section B.2. Finally, section B.3 addresses the validation of the subassembly of bedplate and yaw gearboxes.

B.1 Experimental Modal Analysis of the Bedplate

In order to validate the finite element model of the bedplate, an experimental modal analysis has been performed. Here the bare bedplate is considered, that is, without the main bearing housing, yaw clamps and gearbox mounts. A schematic overview of the measurement setup is given in figure B.1, photos of the actual measurement are shown in figure B.2.

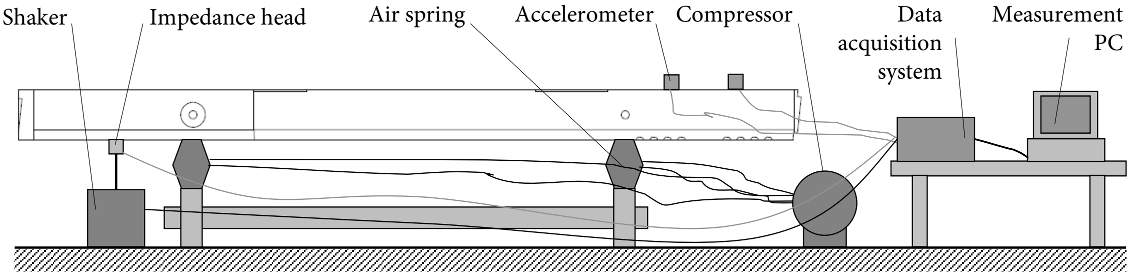


Figure B.1: Schematic overview of the bedplate test setup.

The bedplate was suspended using four air springs . These air springs were pressurized at 5.8 bar and created a low stiffness suspension; the rigid body eigenfrequencies were all around 2 Hz and well below the first flexible eigenfrequency.

Using a total of nine tri-axial ICP accelerometers, 33 locations were measured in four steps. Excitation of the bedplate was achieved using a shaker with a random noise signal. The measurements were performed using Pulse LabShop software from Brüel & Kjær. The measured

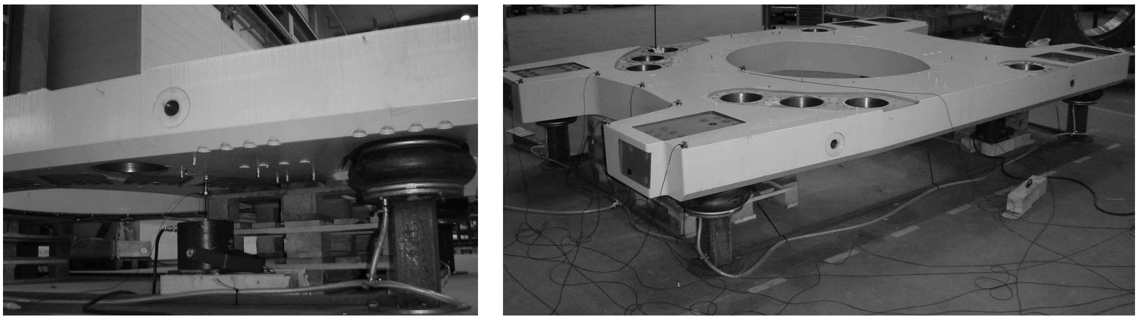
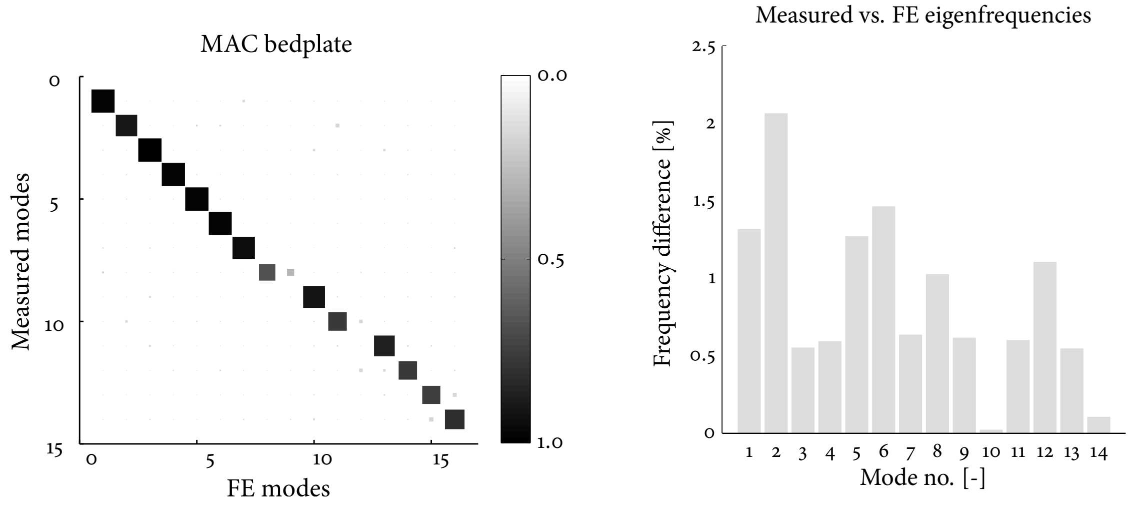


Figure B.2: Photos of the bedplate measurement.

frequency response functions (FRFs) were exported in universal file format and imported to Matlab using the SD Toolbox. This same toolbox was subsequently used perform the modal identification, i.e. the identification of the bedplate's eigenmodes and eigenfrequencies from the measured FRFs.

For the actual validation, the measured modes were expanded using the SEREP technique [144] and a MAC analysis was performed to visualize the correlation between the measured modes and the modes from the finite element model. See figure B.3 (a). The eigenfrequencies are compared by simply checking the relative frequency difference, see figure B.3 (b). As can be seen, the lowest 7 modes compare very with MAC values above 0.9. The low cross-correlation at FE mode 8 and mode 9 is due to the fact that both seem to be in-plane modes, whereas the excitation during the measurement was out-of-plane. Hence, these modes are missing in the set of measured modes. FE mode 10 shows a good correlation to measured mode 9, at the higher modes the correlation deteriorates due to both measurement and modeling inaccuracies. This is however well beyond the frequency range of interest. Furthermore, the difference between the measured eigenfrequencies and the FE eigenfrequencies can be seen to be less than 2% for all identified modes.



(a) (b)

Figure B.3: Results of the bedplate measurement.

As described in section 7.3, in modeling the bedplate it was assumed that the interfaces to the yaw gearboxes behave as locally rigid regions. In order to validate this assumption, two yaw gearbox interfaces have each been equipped with 4 tri-axial accelerometers during the bedplate measurements, as can be seen in figure B.4. By projecting the measured FRFs onto the (local) rigid motions and dividing their norm by the norm of the FRFs, a measure for the rigidity is obtained that is similar to eq. (4.7):

$$\text{rigidity} = \frac{\left\| \boldsymbol{R}\left(\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}\right)^{-1}\boldsymbol{R}^{\mathrm{T}}\boldsymbol{Y} \right\|}{\|\boldsymbol{Y}\|} 100\% \tag{B.1}$$

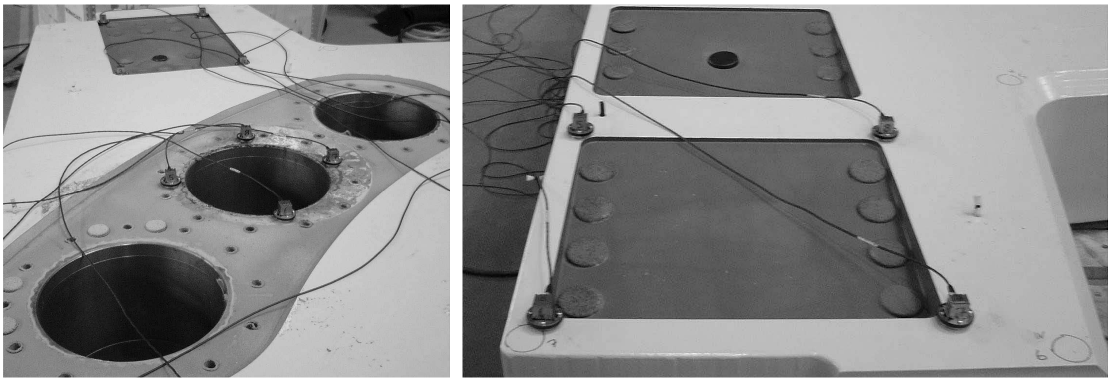


Figure B.4: Measurements performed for checking the rigidity of the interfaces.

It can be seen from figure B.5 that the interfaces of the yaw gearboxes indeed behave more or less rigidly up to a normalized frequency of approximately 0.85, while the frequency range of interest is up to a normalized frequency of 0.5. The rigidities of the interfaces to the main bearing housing and to the main gearbox mounts are also determined and are also shown in figure B.5. It is clear that the interface to the main gearbox mounts can be assumed to behave rigidly up to a normalized frequency of approximately 0.85, whereas the interface to the main bearing housing clearly shows flexibility within the lower frequency range and therefore cannot be considered rigid.

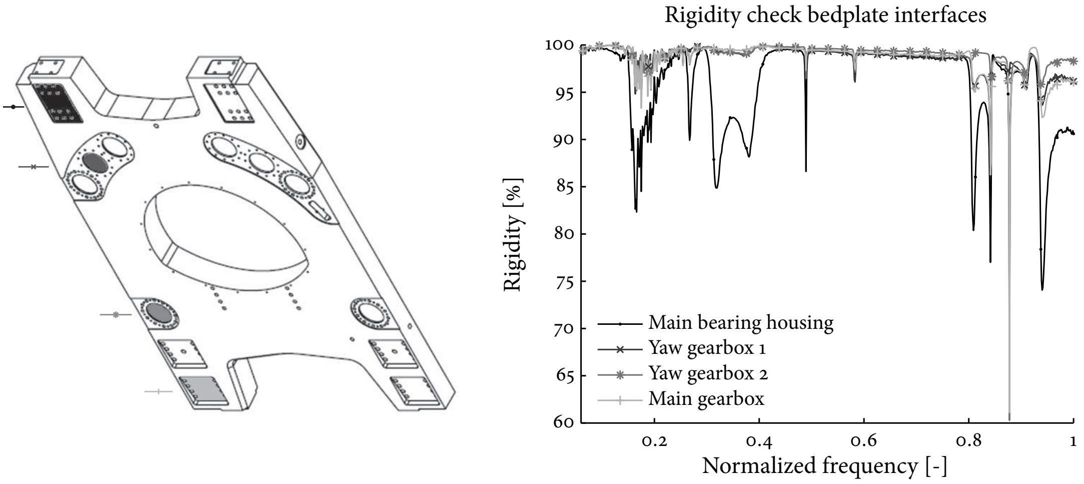


Figure B.5: Rigidity of the interfaces on the top surface of the bedplate.

Finally, two of the yaw pad interfaces on the bottom side of the bedplate were measured to determine their rigidity, as can be seen in figure B.6. In section 7.3 it was mentioned that

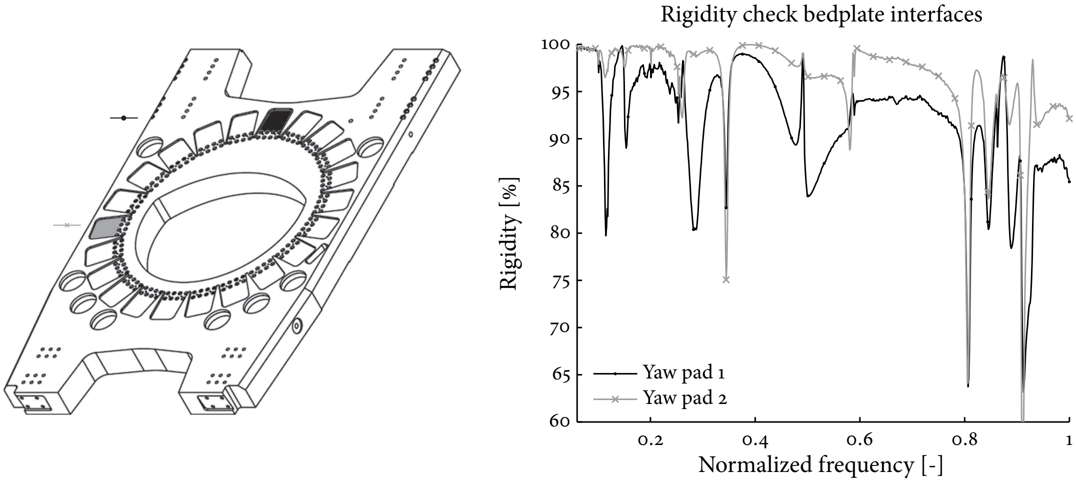


Figure B.6: Rigidity of the interfaces on the bottom surface of the bedplate.

these interfaces could not be modeled rigidly. From the experimental results in figure B.6 it is clear that neither of the two yaw pad interfaces considered here can be assumed to behave rigidly and hence indeed need to be modeled as flexible. It can thus be concluded that with these measurements on the bedplate both the bedplate FE model itself and the approach to modeling its interfaces are valid.

B.2 Validation Measurements on the Yaw Gearbox

As was mentioned in section 7.6, several measurements have been performed in order to try to measure the dynamic behavior of the gearbox internals and to validate the model created in Matlab. These measurements did however not give the desired results. Nonetheless this section briefly describes the different attempts made and lessons learned.

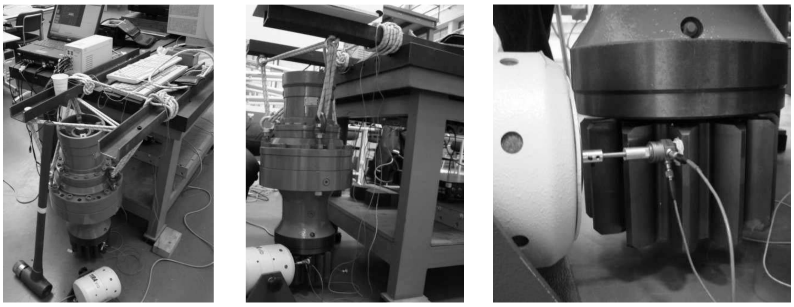


Figure B.7: The yaw gearbox measurement setup.

The setup for the yaw gearbox measurements is shown in figure B.7. The gearbox is suspended using an elastic rope; due to its flexibility and the mass of the gearbox the rigid body eigenfrequencies are very low. From figure B.7 it can be seen that there are two locations where one can excite the internal dynamics: the input pinion and the output pinion. Both have their disadvantages, which will briefly be described here. Excitation through the input shaft has a number of disadvantages:

- Due to the large transmission ratio, small input signals are reduced to negligible amplitudes at the lower stages making them difficult (if not impossible) to measure.
- Shaker excitation is also challenging, since relatively small forces result in large rotations of the input shaft. Therefore the shaker cannot be mounted on the input shaft.
- Due to the play between the gear teeth, gears will shake loose during the measurement, resulting in a (non-linear) time varying system instead of a linear time invariant one.
- Fixing or applying pretension through the output pinion is also infeasible, since due to the transmission ratio of 1000:1 (from output to input) the required torque at the output pinion will be very large. Applying such a large torque is practically impossible in the current setup.

Excitation through the output pinion also has some disadvantages:

- The input displacement amplitude will be amplified. This results in a amplification of the inertia of the upper stages and will also amplify the friction of the upper stages.
- Again, due to the play between the gear teeth, gears will shake loose during the measurement.

However, there are two major benefits that excitation through the output pinion has over excitation through the input pinion:

- It can practically be done. Since exciting the output pinion with (large) forces only leads to small rotations; a shaker stinger can be mounted on the output pinion.
- Since the input force is reduced instead of amplified, the input shaft can be pretensioned (or fixed) with limited pretensioning forces in order to minimize the gear play.

In order to get some idea of the eigenfrequencies of the system, an impact hammer was used to excite the output pinion. No useful results where obtained from these measurements, which is likely caused by the high damping due to friction and gear play. The energy put into the system is dissipated in a very short time, leading to useless FRFs.

The approach taken next was to replace the hammer by a shaker (as can be seen in figure B.7), in order to put energy in the structure during the entire measurement. A number of input signals were used to obtain the measurement data: random noise, chirp and full sine sweep. However, none of the input signals resulted in reproducible results, one of the FRFs found is shown in figure B.8. From these shaker measurements a number of crucial variables were discovered:

- The gear play has a major influence on the measurements. In the extreme cases the input pinion had to be rotated up to five full rotations in order for all the gears to be in contact. From the results one could see when gear teeth contact was lost and the system changed during the measurement.
- Pretension is essential in order to keep the gear teeth in contact with each other, to obtain a system which is as constant as possible.
- The force amplitude should be sufficient to overcome the internal gearbox friction.

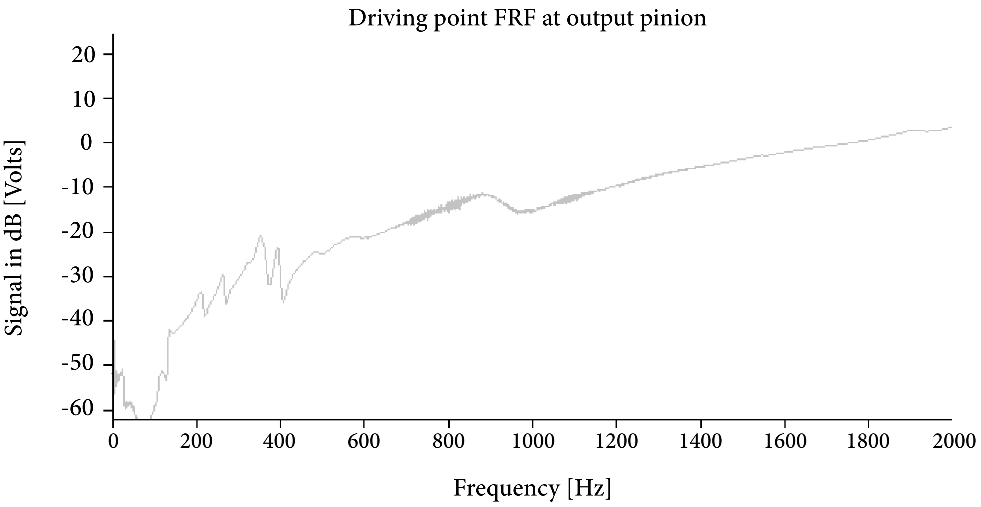


Figure B.8: Example FRF from the first measurements on the yaw gearbox.

A second measurement was attempted, taking into account these lessons learned. The internal friction was determined by measuring the moment needed to turn the input pinion of the yaw gearbox. Knowing this friction, a pretensioning torque was applied to the input pinion to ensure gear teeth contact during the measurements. The pretension was applied by a mass suspended on a cable, which generated a torque through the use of a lever (the radius of the input pinion), as can be seen in figure B.9.

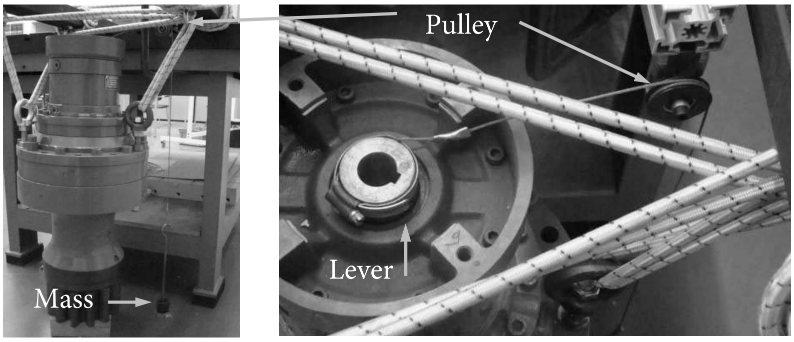


Figure B.9: Pre-tensioning of the yaw gearbox.

With this setup the vibration measurements were performed for a second time. Again no useful results were obtained, which could in fact be due to the pretension applied to the system. Since the pretension was achieved through a mass that generated a constant torque, inertia was also added. Due to the gearbox ratio, this inertia is amplified by a factor of over one million (since inertia scales quadratically with transmission ratio), which could obviously significantly influence the measurement results.

From these attempts it can be concluded that excitation through a shaker and/or hammer did not give the desired results. One idea to come to usable measurement data is to mount an electric motor on the input shaft and apply a constant sine (constant rotation), which will ensure gear teeth contact, and superimpose a random excitation. Operational modal analysis techniques would then have to be used to obtain the eigenmodes and frequencies (see [71]).

B.3 Validation of the Assembly of Bedplate & Yaw Gearboxes

The setup for the final measurement, which was performed after the bedplate validation measurements described in section B.1, was created by mounting four yaw gearboxes to the bedplate. This setup is shown in figure B.10. Again the bedplate was suspended using four air springs. These air springs were pressurized at approximately 6 bar, which was slightly higher in comparison to the first measurement to account for the additional mass of the gearboxes. Like before, the rigid body eigenfrequencies were all found to be well below the first flexible eigenfrequency.

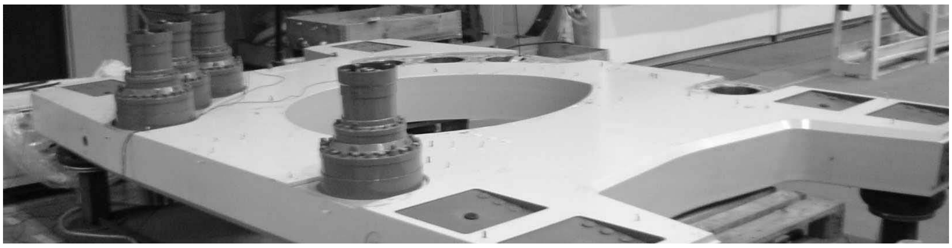


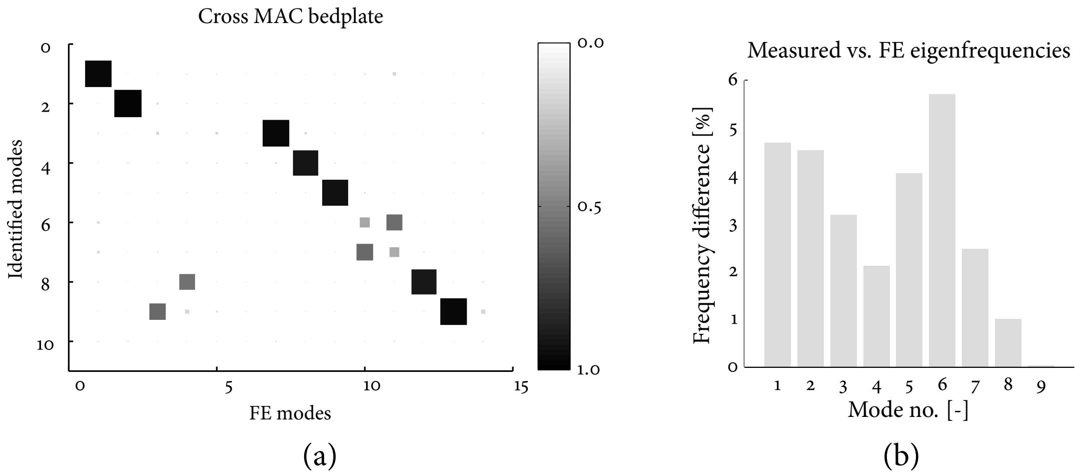
Figure B.10: Photo of the measurement setup

With this setup, an experimental modal analysis was performed on this assembly in a similar fashion as for the isolated bedplate structure. This means that nine tri-axial ICP accelerometers were used to measure 33 locations were measured on the bedplate and 8 locations on the yaw gearboxes. Excitation of the structure was applied by a shaker with a random noise signal. Again, the SD Toolbox in Matlab was used to identify the eigenfrequencies and eigenmodes.

An assembled FE model was created that resembles the measurement setup using the bedplate and yaw gearbox component models. The FE eigenfrequencies and -modes were computed, where the latter were subsequently used to again expand the measured modes using the SEREP technique. Thereafter, a MAC analysis was performed to correlate the measured modes and the finite element modes, the results are shown in figure B.11 (a). Figure B.11 (b) shows the relative frequency difference between the measured and FE eigenfrequencies. Note that in these plots the rigid body modes of the FE model are not shown.

In figure B.11 (b) it can be clearly seen that the first and second (flexible) eigenmodes computed from the assembled FE model have a high correlation with the first two measured modes. It can also be seen that there are four modes missing in the measurement, or alternatively four additional modes in the FE results. These modes correspond to motion of the gearbox internals; since no sensors were placed at the gearbox input and output pinions these could not be measured. Furthermore, since the yaw gearbox model was not validated no conclusions can be drawn from these modes. The next few modes show a good correlation to the measurement, except for FE modes 10 and 11. The modeled and measured eigenfrequencies match reasonably well and are below 5 % for most modes. Given the fact that the yaw gearbox models still need to be validated and updated with measured parameters, this is considered to be not too bad.

From these results it can be concluded that the added mass and stiffness effects of assembling the yaw gearboxes to the bedplate are also present in the assembled FE model, but still have

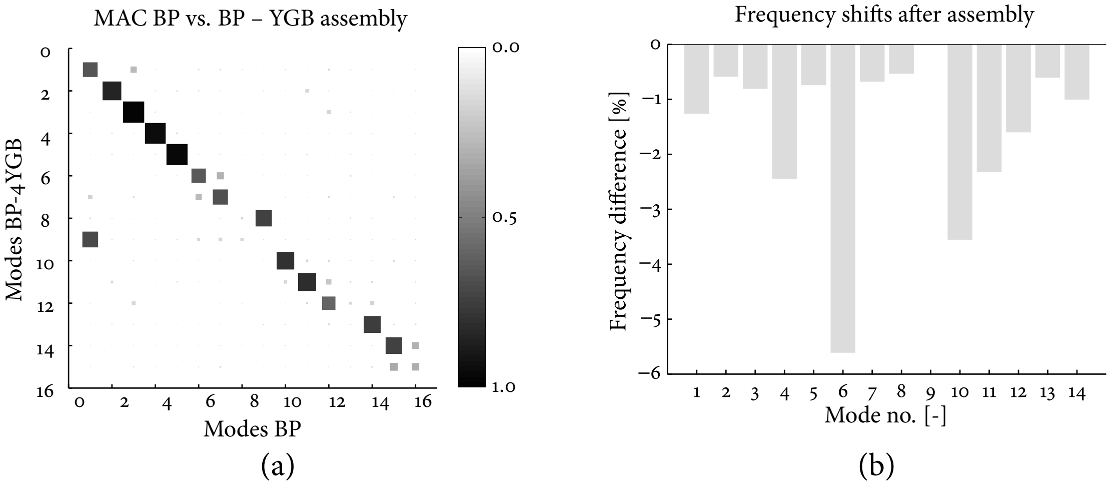


(a) (b)

Figure B.11: Measurement results of the assembly of a bare bedplate and four yaw gearboxes.

a slight discrepancy with respect to the measured data. It is expected that with validated yaw gearbox models these small differences will disappear.

Finally, the effects of assembling the yaw gearboxes on the eigenfrequencies and modeshapes of bedplate are investigated. Using the expanded measured mode shapes, a SUMAC analysis is performed on the (trace of) modes of the bedplate. The resulting plot is shown in figure B.12, as well as the eigenfrequency differences between both systems.



(a) (b)

Figure B.12: Effect on results due to the assembly of the yaw gearboxes

These results show that both the modeshapes and eigenfrequencies change due to the added mass and stiffness of the yaw gearboxes. As expected, some modes are more influenced by the assembly of the gearboxes than others. For instance, the correlation of modes 3, 4 and 5 is still very high and it can thus be concluded that these modes are insensitive for the added gearbox stiffness and mass, whereas the other modes clearly change due to the added mass and stiffness.

C

Detailed Derivation of Error Estimates for Global Eigensolutions

In this appendix some of the steps in the derivation of error estimates for global eigensolutions, as treated in section 5.6, are worked out in detail. The first derivation step elaborated here is the sequence of manipulations needed to go from equation (5.64) to (5.65). The former is repeated here:

$$J\left(\tilde{\boldsymbol{\phi}}\right) = \left(\left(\boldsymbol{I} - \boldsymbol{P}\right)\boldsymbol{d}\right)^T \bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} + \left(\boldsymbol{P}\boldsymbol{d}\right)^T \bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} - \omega_j^2 \boldsymbol{d}^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \tag{5.64}$$

Now the second to last term on the right hand side, $\left(\boldsymbol{P}\boldsymbol{d}\right)^T \bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}}$, is rewritten. To do so, it is first realized that the block diagonal projector $\boldsymbol{P}$, containing along its diagonal the domain projectors $\boldsymbol{P}^{(s)}$ defined in eq. (5.29), in fact corresponds to:

$$\begin{aligned} \boldsymbol{P}^T &= \operatorname{diag}\left(\boldsymbol{M}^{(s)}\boldsymbol{\Phi}_j^{(s)}\boldsymbol{\Phi}_j^{(s)^T}\right) \\ &= \operatorname{diag}\left(\boldsymbol{M}^{(s)}\right)\operatorname{diag}\left(\boldsymbol{\Phi}_j^{(s)}\boldsymbol{\Phi}_j^{(s)^T}\right) \\ &= \operatorname{diag}\left(\boldsymbol{M}^{(s)}\right)\boldsymbol{R}\boldsymbol{R}^T \end{aligned} \tag{C.1}$$

Here $\boldsymbol{R}$ is the global reduction matrix as defined in section 5.3. Hence, one can write:

$$\begin{aligned} \boldsymbol{d}^T\boldsymbol{P}^T\bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} &= \boldsymbol{d}^T\operatorname{diag}\left(\boldsymbol{M}^{(s)}\right)\boldsymbol{R}\boldsymbol{R}^T\bar{\boldsymbol{K}}\boldsymbol{R}\tilde{\boldsymbol{\phi}}_q \\ &= \boldsymbol{d}^T\operatorname{diag}\left(\boldsymbol{M}^{(s)}\right)\boldsymbol{R}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{\phi}}_q \\ &= \tilde{\omega}^2\boldsymbol{d}^T\operatorname{diag}\left(\boldsymbol{M}^{(s)}\right)\boldsymbol{R}\tilde{\boldsymbol{M}}\tilde{\boldsymbol{\phi}}_q \end{aligned} \tag{C.2}$$

Here $\tilde{\boldsymbol{\phi}}_q$ is the approximate eigensolution in terms of the generalized DoF of the reduced model. In the last step use was made of the solution to the eigenproblem of the reduced system for mode j, i.e.:

$$\tilde{\boldsymbol{K}}\tilde{\boldsymbol{\phi}}_q = \tilde{\omega}^2\tilde{\boldsymbol{M}}\tilde{\boldsymbol{\phi}}_q \tag{C.3}$$

As a result, the second to last term in (5.64) can be expressed as:

$$\begin{aligned}(\boldsymbol{Pd})^T \bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} &= \tilde{\omega}^2 (\boldsymbol{Pd})^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \\ &= -\tilde{\omega}^2 ((\boldsymbol{I}-\boldsymbol{P})\boldsymbol{d})^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} + \tilde{\omega}^2 \boldsymbol{d}^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}}\end{aligned} \tag{C.4}$$

In the second step the same split was applied as before to obtain (5.64). Substitution of this result in (5.64) allows the goal oriented error to be written as:

$$J\left(\tilde{\boldsymbol{\phi}}\right) = ((\boldsymbol{I}-\boldsymbol{P})\boldsymbol{d})^T \bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} - \tilde{\omega}^2 ((\boldsymbol{I}-\boldsymbol{P})\boldsymbol{d})^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} + \tilde{\omega}^2 \boldsymbol{d}^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} - \omega_j^2 \boldsymbol{d}^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \tag{C.5}$$

Rearranging terms finally leads to equation (5.65) in section 5.6:

$$J\left(\tilde{\boldsymbol{\phi}}\right) = ((\boldsymbol{I}-\boldsymbol{P})\boldsymbol{d})^T \left(\bar{\boldsymbol{K}} - \tilde{\omega}^2\bar{\boldsymbol{M}}\right)\tilde{\boldsymbol{\phi}} + \left(\tilde{\omega}^2 - \omega_j^2\right)\boldsymbol{d}^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \tag{5.65}$$

C.1 Error Estimates for Global Eigenfrequency

In the derivation of the error estimate for the approximate global eigenfrequency in section 5.6.2, the starting point is the goal oriented error of eq. (5.66):

$$\boldsymbol{0} = ((\boldsymbol{I}-\boldsymbol{P})\boldsymbol{d})^T \left(\bar{\boldsymbol{K}} - \tilde{\omega}^2\bar{\boldsymbol{M}}\right)\tilde{\boldsymbol{\phi}} + \left(\tilde{\omega}^2 - \omega_j^2\right)\boldsymbol{d}^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}}, \tag{5.66}$$

The first step in the derivation is to analyze the last term in this expression, i.e. $\boldsymbol{d}^T\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}}$. Using the adjoint solution $\boldsymbol{d}$ as shown in eq. (5.68), this term can be written as:

$$\begin{aligned}\boldsymbol{d}^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} &= \tilde{\boldsymbol{\phi}}^T \boldsymbol{P}_j^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \\ &= \tilde{\boldsymbol{\phi}}^T \bar{\boldsymbol{M}}\boldsymbol{\phi}_j\boldsymbol{\phi}_j^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \\ &= \left\|\boldsymbol{P}_j\tilde{\boldsymbol{\phi}}\right\|^2 \\ &= 1 - \left\|\left(\boldsymbol{I}-\boldsymbol{P}_j\right)\tilde{\boldsymbol{\phi}}\right\|^2\end{aligned} \tag{C.6}$$

Next, a so-called safety factor $0 \leq \delta < 1$ is defined for which it is assumed that:

$$\left\|\left(\boldsymbol{I}-\boldsymbol{P}_j\right)\tilde{\boldsymbol{\phi}}\right\|^2 \leq \delta \tag{C.7}$$

Such a safety factor is often used in the literature on error estimation to ensure that an error bound remains conservative [104, 122]. The validity of this safety factor can, in an ad hoc fashion, be shown as follows. Expanding the term $\left\|\left(\boldsymbol{I}-\boldsymbol{P}_j\right)\tilde{\boldsymbol{\phi}}\right\|^2$ gives:

$$\begin{aligned}\left\|\left(\boldsymbol{I}-\boldsymbol{P}_j\right)\tilde{\boldsymbol{\phi}}\right\|^2 &= \tilde{\boldsymbol{\phi}}^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} - \tilde{\boldsymbol{\phi}}^T \bar{\boldsymbol{M}}\boldsymbol{\phi}_j\boldsymbol{\phi}_j^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \\ &= 1 - \tilde{\boldsymbol{\phi}}^T \bar{\boldsymbol{M}}\boldsymbol{\phi}_j\boldsymbol{\phi}_j^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}}\end{aligned} \tag{C.8}$$

From a spectral expansion of the mass matrix it indeed follows that:

$$\tilde{\boldsymbol{\phi}}^T \bar{\boldsymbol{M}}\boldsymbol{\phi}_j\boldsymbol{\phi}_j^T \bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \leq \tilde{\boldsymbol{\phi}}^T \left(\sum_{i=1}^{n_a} \bar{\boldsymbol{M}}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^T \bar{\boldsymbol{M}}\right)\tilde{\boldsymbol{\phi}} = 1 \tag{C.9}$$

Hence, it is found that $\boldsymbol{d}^T\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \leq 1 - \delta$.

C.2 Error Estimates for Global Eigenmode

As explained in section 5.6.3, the first step in the derivation of the error estimate for an approximate global eigenmode is to transform the goal oriented error in eq. (5.65) to a global error norm. This is done by choosing $\boldsymbol{a} = \bar{\boldsymbol{K}}\boldsymbol{e}_{\phi}$. Here it is shown how this choice leads to an error estimate in the energy norm:

$$
\begin{aligned}
J\left(\tilde{\boldsymbol{\phi}}\right) &= \boldsymbol{e}_{\phi}^{T}\bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} = \tilde{\boldsymbol{\phi}}^{T}\left(\boldsymbol{I} - \boldsymbol{P}_{j}^{T}\right)\bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} \\
&= \tilde{\boldsymbol{\phi}}^{T}\left(\boldsymbol{I} - \bar{\boldsymbol{M}}\boldsymbol{\phi}_{j}\boldsymbol{\phi}_{j}^{T}\right)\bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} \\
&= \tilde{\boldsymbol{\phi}}^{T}\bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} - \tilde{\boldsymbol{\phi}}^{T}\bar{\boldsymbol{M}}\boldsymbol{\phi}_{j}\boldsymbol{\phi}_{j}^{T}\bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} \\
&= \tilde{\boldsymbol{\phi}}^{T}\bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} - \tilde{\boldsymbol{\phi}}^{T}\bar{\boldsymbol{M}}\boldsymbol{\phi}_{j}\omega_{j}^{2}\boldsymbol{\phi}_{j}^{T}\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \\
&= \tilde{\boldsymbol{\phi}}^{T}\bar{\boldsymbol{K}}\tilde{\boldsymbol{\phi}} - \tilde{\boldsymbol{\phi}}^{T}\bar{\boldsymbol{M}}\boldsymbol{\phi}_{j}\boldsymbol{\phi}_{j}^{T}\bar{\boldsymbol{K}}\boldsymbol{\phi}_{j}\boldsymbol{\phi}_{j}^{T}\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} \\
&= \tilde{\boldsymbol{\phi}}^{T}\left(\boldsymbol{I} - \boldsymbol{P}_{j}^{T}\right)\bar{\boldsymbol{K}}\left(\boldsymbol{I} - \boldsymbol{P}_{j}\right)\tilde{\boldsymbol{\phi}} = \left\|\boldsymbol{e}_{\phi}\right\|_{K}^{2}
\end{aligned}
\tag{C.10}
$$

In the fourth and fifth steps it was used that $\left(\omega_{j}^{2}, \boldsymbol{\phi}_{j}\right)$ is a solution to the unreduced eigenproblem and the mode $\boldsymbol{\phi}_{j}$ is mass normalized, i.e.:

$$
\bar{\boldsymbol{K}}\boldsymbol{\phi}_{j} = \omega_{j}^{2}\bar{\boldsymbol{M}}\boldsymbol{\phi}_{j} \quad \text{and} \quad \boldsymbol{\phi}_{j}^{T}\bar{\boldsymbol{K}}\boldsymbol{\phi}_{j} = \omega_{j}^{2}
\tag{C.11}
$$

The second step in the derivation of the global error norm is to find a bound for the last term of eq. (5.79), $\left(\tilde{\omega}^{2} - \omega_{j}^{2}\right)\boldsymbol{d}^{T}\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}}$. To do so, the adjoint solution $\boldsymbol{d}$ is required. Recalling the definition of this problem for an approximate eigensolution (eq. (5.63)) and inserting the choice for $\boldsymbol{a}$, the following adjoint solution is obtained:

$$
\boldsymbol{d} = \left(\bar{\boldsymbol{K}} - \omega_{j}^{2}\bar{\boldsymbol{M}}\right)^{-1}\bar{\boldsymbol{K}}\boldsymbol{e}_{\phi} = \left(\sum_{i \neq j}^{n_a}\left(\omega_{i}^{2} - \omega_{j}^{2}\right)^{-1}\boldsymbol{\phi}_{i}\boldsymbol{\phi}_{i}^{T}\right)\bar{\boldsymbol{K}}\boldsymbol{e}_{\phi}
\tag{C.12}
$$

Here spectral expansion has been used to express the adjoint solution as a summation of modes. The term $\boldsymbol{d}^{T}\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}}$ can thus be written as:

$$
\left(\tilde{\omega}^{2} - \omega_{j}^{2}\right)\boldsymbol{d}^{T}\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} = \left(\tilde{\omega}^{2} - \omega_{j}^{2}\right)\boldsymbol{e}_{\phi}^{T}\bar{\boldsymbol{K}}\left(\sum_{i \neq j}^{n_a}\left(\omega_{i}^{2} - \omega_{j}^{2}\right)^{-1}\boldsymbol{\phi}_{i}\boldsymbol{\phi}_{i}^{T}\right)\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}}
\tag{C.13}
$$

Next, the approximate eigenmode $\tilde{\boldsymbol{\phi}}$ can be split into a part that lives in the space $\boldsymbol{P}_{j}$ and a part that lives in the orthogonal space, i.e. $\left(\boldsymbol{I} - \boldsymbol{P}_{j}\right)\tilde{\boldsymbol{\phi}} = \boldsymbol{e}_{\phi}$. By doing so and using mode orthogonality properties, one finds:

$$
\begin{aligned}
\left(\tilde{\omega}^{2} - \omega_{j}^{2}\right)\boldsymbol{d}^{T}\bar{\boldsymbol{M}}\tilde{\boldsymbol{\phi}} &= \left(\tilde{\omega}^{2} - \omega_{j}^{2}\right)\boldsymbol{e}_{\phi}^{T}\bar{\boldsymbol{K}}\left(\sum_{i \neq j}^{n_a}\left(\omega_{i}^{2} - \omega_{j}^{2}\right)^{-1}\boldsymbol{\phi}_{i}\boldsymbol{\phi}_{i}^{T}\right)\left(\bar{\boldsymbol{M}}\left(\boldsymbol{I} - \boldsymbol{P}_{j}\right)\tilde{\boldsymbol{\phi}} + \bar{\boldsymbol{M}}\boldsymbol{P}_{j}\tilde{\boldsymbol{\phi}}\right) \\
&= \left(\tilde{\omega}^{2} - \omega_{j}^{2}\right)\boldsymbol{e}_{\phi}^{T}\bar{\boldsymbol{K}}\left(\sum_{i \neq j}^{n_a}\left(\omega_{i}^{2} - \omega_{j}^{2}\right)^{-1}\boldsymbol{\phi}_{i}\boldsymbol{\phi}_{i}^{T}\right)\bar{\boldsymbol{M}}\boldsymbol{e}_{\phi}
\end{aligned}
\tag{C.14}
$$

Now a bound for $\left(\tilde{\omega}^2 - \omega_j^2\right) \boldsymbol{d}^T \boldsymbol{M} \tilde{\boldsymbol{\phi}}$ can be found by maximization as follows:

$$
\begin{aligned}
\left(\tilde{\omega}^2 - \omega_j^2\right) \boldsymbol{d}^T \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}} &= \left(\tilde{\omega}^2 - \omega_j^2\right) \boldsymbol{e}_\phi^T \bar{\boldsymbol{K}} \left(\sum_{i \neq j}^{n_a} \left(\omega_i^2 - \omega_j^2\right)^{-1} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^T \bar{\boldsymbol{M}} \right) \boldsymbol{e}_\phi \\
&\leq \left(\tilde{\omega}^2 - \omega_j^2\right) \max_{i \neq j} \left(\omega_i^2 - \omega_j^2\right)^{-1} \boldsymbol{e}_\phi^T \bar{\boldsymbol{K}} \boldsymbol{e}_\phi \\
&\leq \left(\tilde{\omega}^2 - \omega_j^2\right) \max_{i \neq j} \left(\omega_i^2 - \omega_j^2\right)^{-1} \left\| \boldsymbol{e}_\phi \right\|_K^2
\end{aligned}
\tag{C.15}
$$

Using again the safety factor δ, with $0 \leq \delta < 1$, this bound can be written as:

$$
\left(\tilde{\omega}^2 - \omega_j^2\right) \boldsymbol{d}^T \bar{\boldsymbol{M}} \tilde{\boldsymbol{\phi}} \leq \delta \left\| \boldsymbol{e}_\phi \right\|_K^2 \tag{C.16}
$$

Hence, for the safety factor it should now hold that:

$$
\max_{i \neq j} \left(\frac{\tilde{\omega}^2 - \omega_j^2}{\omega_i^2 - \omega_j^2} \right) \leq \delta \tag{C.17}
$$

Due to the fact that the ω_j are bounded since the model has a finite number of DoF, the condition on δ always holds (see [103, 122]).

D

The Proper Orthogonal Decomposition Method

The proper orthogonal decomposition (POD), also known as principal component analysis (PCA) or Karhunen-Loève transformation (KLT), is a mathematical data analysis method for efficient analysis of complex data. The POD is a so-called orthogonal linear transformation, transforming data dependent on n possibly correlated variables into a reduced or equal set of uncorrelated variables, called principal components. A nice description of the Proper Orthogonal Decomposition (POD) is given in [125] and will be quoted here:

> *The proper orthogonal decomposition (POD) is a powerful and elegant method for data analysis aimed at obtaining low-dimensional approximate descriptions of a high-dimensional process ... The most striking feature of the POD is its optimality: it provides the most efficient way of capturing the dominant components of an infinite-dimensional process with only a finite number of "modes", and often surprisingly few "modes".*

In this appendix the general theory of the POD method will be outlined. An excellent explanation of the mathematical formulation of the POD is given in [32] and [112] and will not be repeated here. Instead, the practical implementation of the method will be addressed by a discussion of the discrete version of theory.

Suppose that from some time-varying signal a number of snapshots m can be obtained; each snapshot is a vector $\boldsymbol{z}_i$ containing the instantaneous values of n output variables. These vectors can for instance represent a structural response or external loading at a specific time instant. Collecting these vectors in a matrix $\boldsymbol{Z}$ gives the $(n \times m)$ response matrix:

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1 & \cdots & \boldsymbol{z}_m \end{bmatrix} = \begin{bmatrix} z_{11} & \cdots & z_{1m} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nm} \end{bmatrix} \tag{D.1}$$

One can now construct the so-called sample covariance matrix:

$$\boldsymbol{C} = \sum_{i=1}^{m} \mathrm{E}\left[(\boldsymbol{z}_i - \boldsymbol{\mu}) (\boldsymbol{z}_i - \boldsymbol{\mu})^T \right], \tag{D.2}$$

where $\boldsymbol{\mu}$ is the average of the snapshots. Now choose the snapshots such that they have a zero mean, i.e.

$$\boldsymbol{x}_i = \boldsymbol{z}_i - \boldsymbol{\mu}, \tag{D.3}$$

such that the covariance matrix can be simply expressed by:

$$\boldsymbol{C} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i \boldsymbol{x}^T = \frac{1}{m}\boldsymbol{X}\boldsymbol{X}^T \tag{D.4}$$

The eigensolutions of the $(n \times n)$ covariance matrix $\boldsymbol{C}$ characterize the POD. These eigensolutions are such that:

$$\boldsymbol{C}\boldsymbol{\phi}_j = \lambda_j \boldsymbol{\phi}_j \qquad j = 1 \ldots n \tag{D.5}$$

Since $\boldsymbol{C}$ is real and symmetric due to its definition, the eigenvectors form an orthogonal basis. Hence, the eigenvectors $\boldsymbol{\phi}_j$ are the proper orthogonal modes (POMs) and the eigevalues λ_j are the proper orthogonal values (POVs). Note that the POV is a measure of the relative energy of the system dynamics contained in the associated POM [111].

Computing simply the eigensolutions of $\boldsymbol{C}$ is one approach to obtain the POMs and POVs. However, one could also use the singular value decomposition (SVD), this will provide some additional information on the decomposition. To this end, recall that the SVD of the real-valued $(n \times m)$ response matrix $\boldsymbol{X}$ can be written as:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{D.6}$$

Here $\boldsymbol{U}$ is the $(n \times n)$ matrix of left singular vectors, $\boldsymbol{\Sigma}$ is the $(n \times m)$ pseudo-diagonal matrix containing the singular values and $\boldsymbol{V}$ is the $(m \times m)$ matrix of right singular vectors. Now note that:

$$\begin{aligned} \boldsymbol{X}\boldsymbol{X}^T &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{U}^T = \boldsymbol{U}\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T\right)\boldsymbol{U}^T \\ \boldsymbol{X}^T\boldsymbol{X} &= \boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{V}\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T\right)\boldsymbol{V}^T \end{aligned} \tag{D.7}$$

Here the right hand sides describe the eigenvalue decompositions of $\boldsymbol{X}\boldsymbol{X}^T$ and $\boldsymbol{X}^T\boldsymbol{X}$, respectively. Hence, the singular values of $\boldsymbol{X}$ are the square roots of the eigenvalues of $\boldsymbol{X}\boldsymbol{X}^T$ and thus correspond to the proper orthogonal values multiplied by the number of samples m. Also, the left singular vectors $\boldsymbol{U}$ of $\boldsymbol{X}$ correspond to the eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$ and thus to the POMs. The advantage of using an SVD to compute the POD is that additional information is obtained in the form of the matrix $\boldsymbol{V}$, the columns of which contain the time modulation of the corresponding POM, normalized by the singular value [112]. This can be valuable information, as it provides insight into the frequency of oscillation of the POM. Furthermore, from a numerical/computational point of view it is more efficient to apply the SVD on the snapshot matrix instead of an eigenvalue analysis on the covariance matrix, as it avoids the (possibly expensive) computation of the covariance matrix.

It can be shown that, due to its formulation, the POD is optimal in a least squares sense when considering the energy in the snapshots. This means that the POMs form a vector basis that captures more energy per mode than any other set of basis vectors [112, 125].

Curriculum Vitae

Sven Voormeeren was born on the 8th of July 1984 in Hellevoetsluis. After completing his pre-university eduction at Penta College CSG Jacob van Liesveldt, Hellevoetsluis in 2002, he started his study of mechanical engineering at Delft University of Technology. Following the completion of the B.Sc. program in 2005, he enrolled in the M.Sc. specialization Engineering Mechanics and Dynamics and an additional "Honors Track" program. In October 2007 he graduated cum laude after carrying out the graduation project at BMW's R&D center in Munich, under the guidance of dr. Dennis de Klerk. For his M.Sc. thesis entitled "Coupling Procedure Improvement & Uncertainty Quantification in Experimental Dynamic Substructuring" he received the "Corus Award" for best M.Sc. graduate in the year 2007–2008 in the Netherlands in the field of mechanical and materials engineering.

Since May 2008 he has been working as a PhD researcher in a collaborative project between Siemens Wind Power (SWP) and Delft University of Technology (DUT), of which this thesis is the result, under the supervision of prof. Daniel Rixen and dr. David Molenaar. During this period, he also implemented some of the tools developed in this project in the engineering practice at Siemens, helped to extend the SWP-DUT collaboration through a second PhD project, and supervised a number of B.Sc. and M.Sc. students.

Index

Accelerometers, 273
Adaptive model reduction, 122, 219
Adaptivity schemes, 123
Adjoint problem, 110, 111, 113, 118, 123
Aero-elastic codes, 6, 212
Air springs, 273
Assembly, 56
    Flexibility, 62
    Mixed, 64
    Non-conforming meshes, 68
    Stiffness, 56
    Three field formulation, 57
Attachment modes, 22

Bedplate, 164
    Model validation, 273
    Modeling, 165
    Reduction basis updating, 186

Cauchy-Schwarz inequality, 112, 113
Compatibility condition, 57, 80
    Weakened, 36, 63, 64
Component mode synthesis, 17
Component model reduction, 15, 18
    Craig-Bampton method, 30, 175
    Dual Craig-Bampton method, 35, 175
    Guyan method, 29
    MacNeal method, 31, 33
    Mixed Craig-Bampton method, 40, 175
    Modal truncation augmentation, 44
    Non-linearity, 38
    Overview, 54
    Rubin method, 31, 34, 175
Conjugate gradient, 130
    Block algorithm, 135, 189
    Orthogonalization, 138
    Preconditioning, 133, 136, 190
    Projection and reconjugation, 134
Constraint modes, 21
Craig-Bampton method, 30, 175

Damping, 39
Danish design, 2, 161
Dirichlet-Neumann assembly, 60, 65
Discrete least squares compatibility, 70
Discretization error, 104
Dual assembly, 59
Dual Craig-Bampton method, 35, 175
Dual decoupling, 79
DWR method, 109

Equilibrium condition, 80
Error estimates, 106, 228
    Eigenfrequency, 118, 225
    Eigenmode, 120, 225
    Global eigensolutions, 117, 225
    Global norms, 111
    Goal oriented, 109, 223
    Harmonic solutions, 109, 222, 223
    Time-stepping solutions, 115
Error types
    Discretization error, 104
    Reduction error, 104

Fixed interface modes, 28
Floating frame of reference, 39
Free interface modes, 25
Frequency based substructuring, 15

Galerkin orthogonality, 109
Gear interaction, 171
gearbox housing, 173
Guyan method, 29

Harmonic analysis, 94
HAWT, 161
Hydrodynamic loading, 217

IFPKS method, 146, 194
    Implementation, 151
    Interpretation, 149
    Preconditioning, 148, 197
Impulse based substructuring, 16
Interface, 72
    Inertia, 74, 77
    Stiffness, 72, 76
Interface modes
    Displacement modes, 90

    Force modes, 93
    Joint modes, 87
    Rigid modes, 89
Interface reduction, 86, 177
    Modal, 89, 92
    Rigid, 88
Interface reduction basis, 89, 90, 93
Inverse iteration method, 140

Joint interface modes, 87

Krylov sequence, 132, 144
Krylov subspace, 133, 144

Lanczos method, 143
Levelized cost of energy, 4

MAC, 98, 175, 232
MacNeal method, 31, 33
Mixed Craig-Bampton method, 40, 175
Modal analysis, 95
Modal truncation augmentation, 44, 229
    Fixed interface, 48
    Free interface, 50

Newmark method, 96, 233
Node collocation, 68

Offshore wind energy, 2, 211
Offshore wind turbine
    Foundations, 212
    Model, 214, 217

POD, 38, 46, 100, 219, 285
Preconditioned conjugate gradient, 133
Primal assembly, 60, 63

Reduction basis, 19
Reduction basis updating, 128, 186, 200, 205
Reduction error, 104
Refinement schemes, 124
    Linear, 124
    Threshold, 124
Residual attachment modes, 24
Rigid body modes, 26
Rigidification, 89
Rigidity, 89
Rubin method, 31, 34, 175

Sequential basis updating, 155, 201
SEREP, 274
Solvers, 129
    Conjugate gradient method, 130
    Direct, 129
    Eigensolvers, 140, 145
    Inverse iteration method, 140
    Iterative, 129
    Lanczos method, 143
    Subspace iteration method, 142
Static condensation, 22
Static modes, 21
    Attachment, 22
    Constraint, 21
    Residual attachment, 24
Static modes updating, 130, 189
    Low tolerance, 138
    Preconditioning, 136
    Sequential, 155, 203
Subspace iteration method, 142
Substructure decoupling, 77
SUMAC, 99, 181
Superelement, 30

Three field formulation, 57
Time integration, 96, 233
Tower top, 164, 168
TRAC, 100
Triangle inequality, 111, 113, 114

VAWT, 161
Vibration modes, 25
    Fixed interface, 28
    Free interface, 25
    Rigid body, 26
Vibration modes updating, 140, 145, 151, 194
    Extended preconditioning, 153
    Initial guess, 154
    Preconditioning, 152
    Sequential, 155, 204

Yaw controller, 164
Yaw gearbox, 164
    Measurements, 276
    Modeling, 169
Yaw gears, 170
Yaw motor, 164
Yaw pads, 164, 167
Yaw ring, 164, 168