Department of Precision and Microsystems Engineering

FAST TOPOLOGY OPTIMIZATION FOR TRANSIENT MECHANICAL PROBLEMS

T.M. van der Linde

Report no	:	EM 2016.001
Coach	:	dr. ir. M. Langelaar
Professor	:	prof. dr. ir. F. van Keulen
Specialisation	:	Engineering Mechanics
Type of report	:	MSc. Thesis
Date	:	7 January 2016

Delft University of Technology

Challenge the future

Fast Topology Optimization for Transient Mechanical Problems

Combining Model Order Reduction and Optimization

by

Thomas van der Linde

in partial fulfillment of the requirements for the degree of

Master of Science
in Mechanical Engineering

at the Delft University of Technology,
to be defended publicly on Thursday January 7, 2015 at 12:45 AM.

Supervisor:	dr. ir. M. Langelaar	TU Delft
Thesis committee:	dr. ir. J.F.L Goosen	TU Delft
	dr. ir. R.A.J. van Ostayen	TU Delft
	dr. ir. H. Vallery	TU Delft

Abstract

Topology optimization is the process of finding the optimal shape of a structure while making no assumptions on its final shape. It is characterized by the large number of associated design variables, and therefore relatively high computational costs. Within the field of structural mechanics, most applications of topology optimization focus on linear statics.

An extension to time-domain dynamics will have many useful applications, since dynamics often play an important role in structural mechanics. In the past 15 years considerable effort has been spent on research of this topic, but industrial applications are hindered by the high costs of analysis of time-domain problems. These costs are generally several orders of magnitude larger than those associated with statics or frequency-domain dynamics. The main goal of this research is to investigate the possibility of reducing these costs by applying model order reduction.

Model order reduction consists of projecting the response of a system onto a subspace with a lower dimension, spanned by a set of basis vectors. The classical choice for these basis vectors is the eigenvectors of the system, leading to the so-called modal superposition method. This method can result in large errors in the model, making it unsuitable for topology optimization. A possible improvement on this method, the modal acceleration method, augments the response with the quasi-static displacement due to the external load. While the resulting accuracy is often sufficient for optimization, further improvements can be found by using a basis of Ritz vectors. These are cheaper to compute than eigenvectors, and can approximate the results of the full model with a very high accuracy.

This thesis describes how to implement model order reduction as part of the topology optimization process for time-domain dynamics. The most expensive part of the optimization is the solution of two systems of differential equations: the equations of motion and the adjoint equation. Model order reduction is applied to solve these cheaply. In this document several important implementation issues are discussed, after which the accuracy of the different reduction methods is tested on a set of representative numerical examples. The results indicate that the response of the full model can be approximated with a very high accuracy by using a small set of Ritz vectors. The number of basis vectors is an important parameter, a method is proposed to automatically determine the right value.

Although the response can be approximated with very high accuracy, topology optimization also requires design sensitivities, the derivative of the response with respect to the design variables. The error in these sensitivities is generally several orders of magnitude larger. It is demonstrated that this is partly determined by the choice of the objective function, the function to be minimized in the optimization. The design sensitivities found using Ritz vectors are shown to be more accurate than those found using the modal acceleration method.

Finally, some optimization examples are investigated to study the effect of the errors in the sensitivities. Optimization using the full model is compared to optimization using the various model reduction methods. Although comparing optimization results is difficult, the results show that a model using Ritz vectors converges similar to a full model, even for a very small number of basis vectors. This leads to significant improvements in performance: the simulation is shown to be 20 times faster, the total optimization up to 8 times. The memory requirements, which can be critical for this type of optimization, are also shown to be reduced by up to two orders of magnitude.

Concluding, model order reduction can significantly increase the performance of topology optimization for time-domain dynamics. Ritz vectors are in most cases the most efficient reduction basis. Some recommendations for future research are given, including possible ways of making the described method more efficient.

Acknowledgements

This thesis is the conclusion of my research of the past 15 months, as well as of my studies here in Delft. I would like to thank a few people for their support. First of all, my supervisor, Matthijs Langelaar. Thank you for your eternal enthusiasm, in-depth reviewing, and general support during the past year. Secondly I want to thank Paolo Tiso, for helping me find this subject, and some very useful discussions in the early stages of this research.

In general, I have been very lucky to end up at the department of PME. I experienced this as a great environment, with excellent staff and committed students, which provided me with the best possible motivation. Special thanks goes out to Sarah, Anton, Stefan and Rolf, my fellow board members of study association Taylor. Working together with you guys for the past year was a lot of fun, especially the trip to Japan, which was definitely a great ending to all the hours we invested. Furthermore, I want to single out Anoop, Caspar, Daan, Marius and Panagiotis, for all the endless coffee breaks and discussions about our research. Anoop deserves a special mention, for our frequent conversations about the details of model order reduction. I also want to thank everyone who reviewed parts of this document, especially Wouter, Simon and Jurgen.

During my time as a student I have met many new friends. To all my climbing and skiing buddies: I hope we can keep on spending as much time as possible together in the mountains. To my room-mates Marjolijn, Jurgen, Simon and Elena, thank you guys for always being around and generally trying to make life as much fun as possible. Final thanks goes out to my family. During the past few years I have been very busy, so I have not been around often. Thank you for your understanding and support during these years and before.

Thomas van der Linde
Delft, December 2015

NOMENCLATURE

SYMBOLS

Symbol	Units	Description
β	–	Newmark integration scheme parameter
$\mathbf{C}$	N/m/s	Damping matrix
E	N/m^2	Young modulus
e	–	Error estimate for Model Order Reduction
$\mathbf{f}$	N	Load vector
g		Constraint function
γ	–	Newmark integration scheme parameter
h	h	Step size for time integration
$\mathbf{I}$		Identity matrix
$\mathbf{K}$	N/m	Stiffness matrix
$\mathbf{L}$		Spatial load vector
$\boldsymbol{\lambda}$		Adjoint variable used in sensitivity analysis
$\mathbf{M}$	kg	Mass matrix
N	–	Size of full system (number of degrees of freedom)
n	–	Size of reduced system (number of basis vectors)
n_{el}	–	Number of elements
n_{t}	–	Number of time steps
Ω	rad/s	Load frequency
ω_i	rad/s	i-th eigenfrequency
p	–	Penalization parameter
Φ		General function used in sensitivity analysis
ϕ		General function used in sensitivity analysis
ψ		Objective function
$\mathbf{q}$	m	Nodal coordinates, 2 per node. Sorted as $[x_1, y_1, x_2, y_2, ...]$
$\mathbf{R}$		Term resulting from adjoint sensitivity formulation, see Section 2.4.3
$\mathbf{r}$	m	Residual displacement
$\hat{\mathbf{r}}$	N	Residual force
ρ	kg/m^3	Density
T	s	Final time of simution
t	s	Time
τ	s	Alternative time variable, equal to $T - t$
$\boldsymbol{\theta}$		Reduced set of of coordinates for adjoint variable
$\mathbf{V}$	–	Set of reduced basis vectors
$\mathbf{x}$	–	Densities of elements (design variables)
x_{max}	–	Maximal volume fraction
$\mathbf{z}$	m	Reduced set of coordinates

ACRONYMS

Acronym	Explanation
AVM	Adjoint Variable Method
DoF	Degree of Freedom
FEM	Finite Element Method
MAM	Mode Acceleration Method
MMA	Method of Moving Asymptotes
MOR	Model Order Reduction
MSM	Modal Superposition Method
MTA	Modal Truncation Augmentation
PCG	Preconditioned Conjugate Gradient method
PI	Polynomial Interpolation penalization
POD	Proper Orthogonal Decomposition
QSRV	Quasi-Static Ritz Vectors
RAMP	Rational Approximation of Material Properties
SIMP	Solid Isotropic Material with Penalization
SQP	Sequential Quadratic Programming
TO	Topology Optimization

Contents

LIST OF FIGURES

1

INTRODUCTION

During the past fifty years, there have been dramatic changes in engineering. In the 1960s engineers at NASA still used slide rules to design and build the rockets that brought men to the moon. Today, computers have become an integral part of precision engineering, enabling designs with nanometre accuracy such as wafer steppers or particle accelerators. With increasing computer power new design tools are gaining popularity, such as topology optimization.

But computer power is not unlimited. Topology optimization is increasingly applied to many problems, but for time-domain dynamics it is still relatively expensive. The associated computational costs can be several orders of magnitude larger than for similar statics problems. Therefore this sees very little application.

An optimization method for time-domain dynamics could be a valuable tool in design, since these dynamics often play an important part in structural mechanics. A well-known method of reducing the cost of time-domain simulation is model order reduction. In this research, this will be applied to topology optimization. The separate fields of topology optimization and model order reduction have seen a lot of research, but little is known on combining these fields. Therefore, in this thesis an answer is sought to the following research questions:

1. Can model order reduction be applied to increase the efficiency of topology optimization for transient, linear structural problems?
   (a) What are the differences between optimizing a transient problem and a static problem?
   (b) How can model order reduction and topology optimization be combined?
   (c) What method of model order reduction is most suitable?
   (d) What is the effect of errors in the model order reduction on the optimization process?
2. Can this be used in engineering practice?
   (a) What are the limitations?
   (b) Which parameters are important, and how can they be chosen?
   (c) How can the quality of the reduced model be evaluated during the optimization?

1.1. OUTLINE OF THESIS

This document is divided into seven chapters, Fig. 1.1 shows an overview of these chapters and their connection. After this introduction, the second chapter describes all relevant theory and recent research. Its aim is to provide the reader with the necessary knowledge to understand the rest of the work. This includes the relation between this work and all relevant literature. Chapters three and four discuss how the described theory and methods can be combined and implemented. Chapter three treats the implementation of optimization for time-domain dynamics, chapter four discusses model order reduction and its effect on the optimization.

In chapter five the various model order reduction methods are tested on some numerical examples. The focus is on accuracy and the effect of various parameters. In chapter six the total proposed method is tested on various topology optimization problems, to study the overall efficiency. Finally chapter seven compiles the conclusions and recommendations found in this thesis.

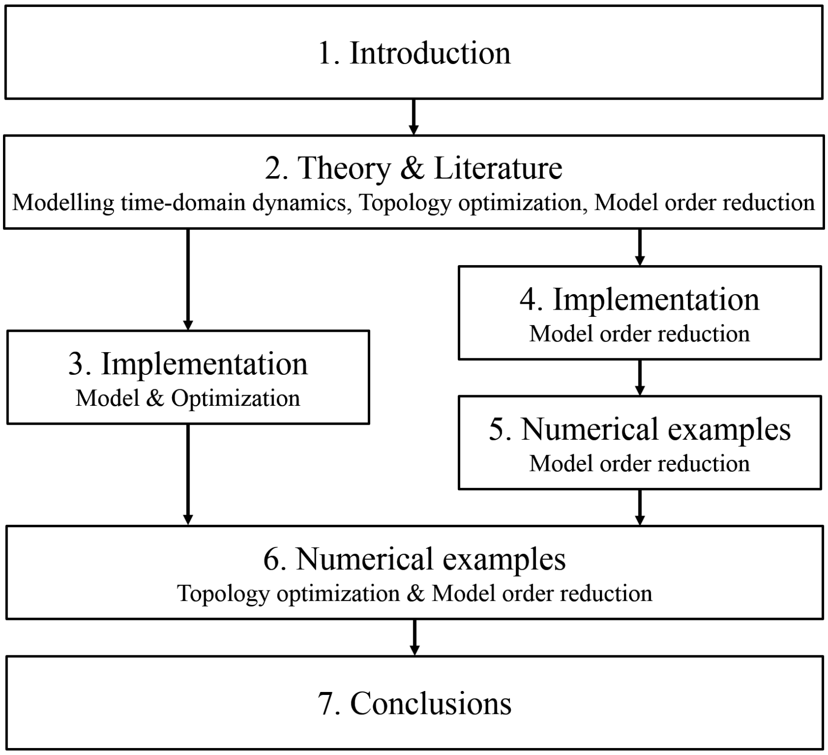


Figure 1.1: Overview of chapters in this thesis.

1.2. NOTATIONS

This thesis treats topics from three different disciplines: structural dynamics, topology optimization and model order reduction. The most commonly used nomenclature and notations of each of these fields are used. A full list of symbols and abbreviations can be found at the beginning of this document, page vii.

The mathematics in this document will contain many derivatives. Time derivatives will be abbreviated using a dot, for example $\dot{q}$,$\ddot{q}$. Derivatives with respect to the design parameters will be shown as $\frac{\partial q}{\partial \mathbf{x}}$ or for compactness as q'. Other notations follow common engineering conventions: bold lower-case indicates vectors ($\mathbf{q}$), bold capitals indicate matrices ($\mathbf{V}$), where lower-case bold symbols are used to refer to a column of this matrix ($\mathbf{v}_i$).

2

Theoretical Fundamentals & Literature Review

2.1. Introduction

In this research three fields are investigated: topology optimization, time-domain structural dynamics, and model order reduction. The aim of this chapter is to give an overview of relevant theory and research in all these fields. We will start by reviewing each separate field, after which we investigate combining them, and end with a short summary.

2.2. Topology optimization

Topology optimization is the optimization of a structure within a design domain, without making any assumptions beforehand on its shape. It has been a very active research field for the last decades, and has been applied to many different types of problems. To keep this section compact only the chosen method of topology optimization will be explained. For alternative approaches, see Appendix C.

In this research a density based topology optimization is studied, which is by far the most common method. Density based topology optimization consists of the following steps:

1. Define design domain, external forces and boundary conditions (Fig. 2.1a).

2. Divide design domain into separate elements (Fig. 2.1b). Assign density variable x_i to each element, normalized such that $0 \leq x_i \leq 1$[1], where 0 corresponds to a void element and 1 to a filled element. These densities will be used as design variables in the optimization.

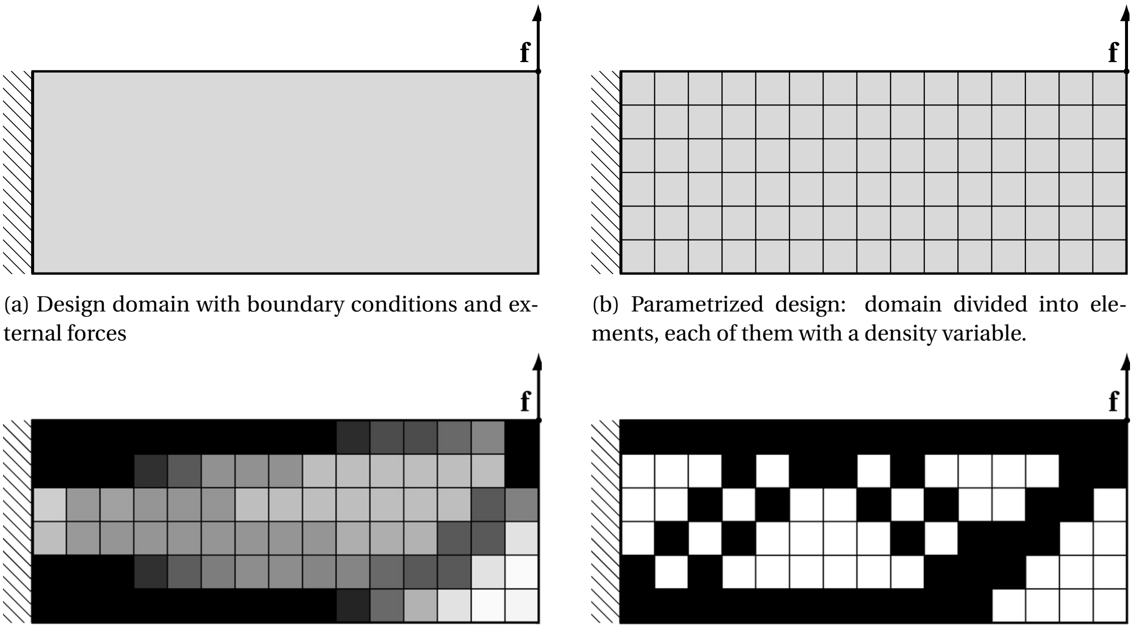
3. Define optimization problem. This consists of an objective function to be minimized/maximized, with possibly a number of constraints.

4. Find the values of x_i that result in an optimal value of the objective function within the constraints. Optimization will iteratively find better designs (Fig. 2.1c) and converge towards an optimal design (Fig. 2.1d).

The most important characteristic of this method is that the number of design variables is very high. This has a few consequences. First of all, this means gradient-based optimization algorithms are much more effective than gradient-free algorithms. But the calculation of gradient information (sensitivities) is complicated because of the large number of design variables. This means that so-called adjoint sensitivities are the only efficient method. This will be discussed in Section 2.4.3.

Furthermore, the high number of variables complicates analytical solutions of the optimization problem. Therefore the vast majority of research relies on several numerical examples to prove their proposed methods.

[1] Care should be taken to avoid singular matrices caused by a completely void element. The easiest way to solve this is by using a non-zero minimum density, in this research 1×10^{-3} is used.



(a) Design domain with boundary conditions and external forces

(b) Parametrized design: domain divided into elements, each of them with a density variable.

(c) Intermediate design. Color of elements corresponds to element density: white is 0, black 1, grey intermediate.

(d) Optimized design. There are no grey elements, but there is some *checkerboarding*: some elements are only connected at their corners.

Figure 2.1: Density based topology optimization can be summarized in these steps. A design domain is defined (a) and parametrized(b). Each element has a variable determining its density, the optimization process finds the optimal values of these variables, illustrated by (c) and (d)

2.2.1. PENALIZATION

In the density based methods it is necessary to avoid intermediate densities. For easy manufacturability structures should only have elements with either zero (white) or maximum (black) densities. Elements with intermediate (grey) densities could physically be interpreted as having a different sheet thickness for two-dimensional problems, or as having a micro-structure with voids for three-dimensional structures. Although not physically impossible, manufacturing a design with grey elements is clearly more difficult than black-and-white designs.

Black & white designs are most commonly achieved through penalization. This involves modifying the model in such a way that intermediate densities result in a higher value of the objective function, which causes the optimizer to avoid them. For structural models, this can be described by:

$$E_p(x_i) = f_E(x_i, \mathbf{p})E \tag{2.1}$$

$$\rho_p(x_i) = f_\rho(x_i, \mathbf{p})\rho\,, \tag{2.2}$$

where f is a function dependent on the design variable and penalization parameters $\mathbf{p}$. The density is usually not penalized ($f_\rho = 1$). A few notable options for the penalization of the stiffness are[1]:

$$\text{SIMP: } f_E(x_i, p) = x_i^p \tag{2.3}$$

$$\text{RAMP: } f_\rho(x_i, p) = \frac{x_i}{1 + p(1 - x_i)} \tag{2.4}$$

$$p \geq 1, \qquad 0 \leq x_i \leq 1\,.$$

These are used in almost all compliance minimization problems. Both methods results in a lower Young modulus for intermediate densities. In contrast to SIMP, RAMP results in a non-zero sensitivity at zero density, which is for example beneficial when gravity loading is studied.

In the work of Zhu *et al.* [2] a slightly modified version is used, which prevents so called *local modes*. These are vibration modes that appear in low-density parts of the structure, and do not contain any information about the response of the parts of the structure that do contain material. This will be treated briefly in Section 5.2, for more details the reader is referred to the work of Pedersen [3]. SIMP penalization results in the appearance of local modes, RAMP does not. Zhu *et al.* [2] propose a third method, named *Polynomial Interpolation* (PI):

$$\begin{aligned} f_E(x_i, p) &= p_2 x_i^{p_1} + (1 - p_2) x_i \\ f_\rho(x_i, p) &= 1 \\ p_1 = 3, \quad p_2 &= 15/16, \quad 0 \le x_i \le 1\,. \end{aligned} \tag{2.5}$$

This method is very similar to SIMP and RAMP, see Fig. 2.2. It is also based on the assumption that compliant material is adverse to the objective function. While this might be true for many transient problems, in general it is not. A more sophisticated approach was proposed by van de Ven *et al.* [4], which resulted in black & white design for a transient thermo-mechanical problem. For a transient mechanical problem, this so called *material penalization method* consists of optimizing the following problem:

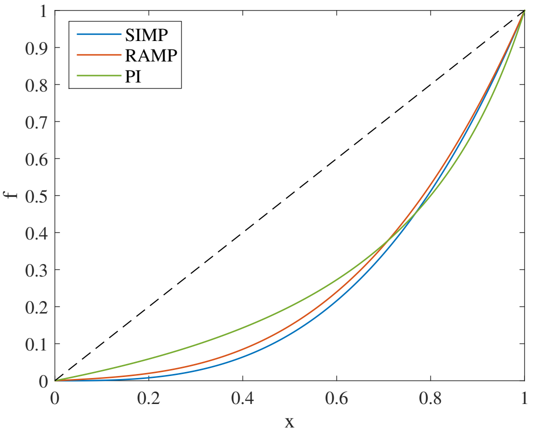
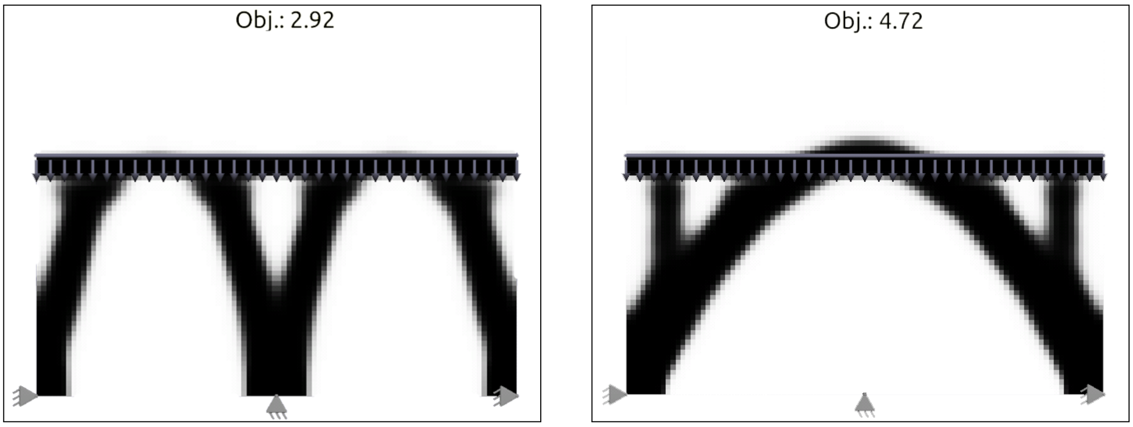


Figure 2.2: Plot of penalization functions for $p = 3$, $p_2 = 15/16$. Some differences are visible between the three penalization methods, but they all result in a lower stiffness for intermediate densities. This causes the optimizer to avoid these, leading to the desired black-white designs.

$$\begin{aligned} &\min_{x_i} \max_{p_{i,j}} \Psi(\mathbf{q}, t) \\ &\text{subject to: } \mathbf{M}(x_i, p_{i,j})\ddot{\mathbf{q}} + \mathbf{C}(x_i, p_{i,j})\dot{\mathbf{q}} + \mathbf{K}(x_i, p_{i,j})\mathbf{q} = \mathbf{f} \\ &i \in [1, N_{\text{el}}], \quad j \in [1, 2]\,. \end{aligned} \tag{2.6}$$

Every element has its own set of penalization parameters (one for stiffness, one for density), which are chosen to maximize the problem. This in turn results in a higher objective function for intermediate densities, which causes the optimizer to avoid these, producing black and white designs. This method can be used with many different penalization functions. The material penalization approach is expected to work for every transient topology optimization problem, although it does triple the number of design variables.

Penalization has an important disadvantage. It makes the design space less convex, increasing the possibility of finding local minima. In topology optimization, local minima often correspond to a missing structural element, illustrated by Fig. 2.3. Creating the missing component would involve removing material somewhere, and initially creating elements with an intermediate density at the locating of the wanted component. The latter step is unattractive due to the penalization, therefore the optimization problem contains a local constrained optimum.



(a) Global optimum. The value of the objective function is 2.92.

(b) Local optimum, note the significantly higher objective function (4.72).

Figure 2.3: Illustration of local optima in topology optimization. Three supports and a line load are defined, for which the static compliance is minimized. To ensure a black-white design, SIMP penalization is applied. This clearly leads to local optima. The second design is not the global optimum, but formation of a column in the middle of the structure would initially increase the compliance, this means the optimization algorithm is stuck in this local optimum. The images were generated using the highly recommended mobile application *Topopt*[5].

2.2.2. FILTERING

A problem often encountered in topology optimization is *checkerboarding*. This can clearly be seen in Fig. 2.1d: we see that in a few places the elements are only connected in the corners, forming a checkerboard pattern. If this design would be constructed, there would be very high stress concentrations. But the finite element model does not capture this.

Another issue is mesh dependence. This means that the solution changes when the mesh is refined, which is undesirable. Both problems can be solved in various ways [6], of which the most commonly used is filtering. In this method a blurring filter is applied to either the density or sensitivities, effectively restricting the minimum feature size to the filter radius. This removes both checkerboarding and mesh dependence, with minimal extra computational effort.

2.2.3. OPTIMIZATION ALGORITHMS

In gradient based topology optimization, roughly two types of algorithms are used. The first of these are the optimality criteria methods. These combine sensitivity information with some heuristics, and can be very efficient. However, the algorithm must be tailored to the specific problem [7].

A more general approach is using a local approximation. The optimization problem is approximated using an another function based on the response and sensitivity information. These approximation functions are chosen to be easily optimizable, popular choices are linear functions (*linear programming*), or the Method of Moving Asymptotes (MMA)[8]. This last method is the most commonly used one in topology optimization, and approximates the problem using a convex function. Finally another popular algorithm is Sequential Quadratic Programming (SQP), which consists of using a quasi-Newton method on the (Karush-Kuhn-Tucker) optimality conditions.

The Optimization Toolbox of MATLAB has an implementation of linear programming, SQP and a few other methods. There is also a MATLAB implementation of MMA, which is used by a great number of researchers in Topology Optimization. This was provided by prof. Svanberg, the author of MMA.

2.3. Transient analysis

The finite element method results in a system of second order differential equations. In this section, it will be discussed how the dynamic response can be calculated from these equations. Again, this work is restricted to linear systems. These can be described by the following *equations of motion*:

$$\begin{aligned}\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{C}\dot{\mathbf{q}}(t) + \mathbf{K}\mathbf{q}(t) &= \mathbf{f}(t) \\ \mathbf{q}(t=0) &= \mathbf{q}_0 \\ \dot{\mathbf{q}}(t=0) &= \dot{\mathbf{q}}_0\,, \end{aligned} \tag{2.7}$$

where $\mathbf{q}_0$ and $\dot{\mathbf{q}}_0$ are the initial conditions. Later on model order reduction will be applied to these equations, so to avoid confusion Eq. (2.7) will be referred to as the *full equations of motion*, and its solution as the *full solution*. Furthermore, this research is restricted to loads of the following type:

$$\mathbf{f}(t) = \sum_{i=1}^{n_\mathrm{L}} \mathbf{L}_i f_i(t)\,. \tag{2.8}$$

This load are described by number of spatial terms $\mathbf{L}_i$ and a temporal component $f_i(t)$. In many cases n_L is equal to 1 or a small number. This will turn out to be necessary for efficient model order reduction.

2.3.1. Analytical solutions

For some loads (for example harmonic), the equations of motion can be solved analytically, using classical vibrations theory. This does require a full analysis of the eigenmodes of the system, and results in an analytical expression of the motion of the system. For arbitrary loads, the undamped equations of motion can be decoupled, and solved individually using Duhamel's integral [9]. This is computationally cheaper than full numerical integration of the equations, but also requires a full eigenanalysis.

2.3.2. Numerical integration (Newmark method)

While analytical solutions are only possible in certain cases, direct numerical integration of the equations of motion will always work. Various methods exist for this, for example Euler, Runge-Kutta, etc, which can be classified in many different ways: implicit/explicit, single/multistep, etc. For structural dynamics, the most commonly used method is the Newmark scheme. Compared to other methods, the Newmark method will generally allow larger time steps for similar accuracy. A detailed analysis of this can be found in the textbook by Hughes [10]. The core of the Newmark process is given by the following algorithm:

$$\dot{\mathbf{q}}(t_{n+1}) = \dot{\mathbf{q}}(t_n) + (1-\gamma)h\ddot{\mathbf{q}}(t_n) + \gamma h\ddot{\mathbf{q}}(t_{n+1}) \tag{2.9}$$

$$\mathbf{q}(t_{n+1}) = \mathbf{q}(t_n) + h\dot{\mathbf{q}}(t_n) + h^2(\frac{1}{2}-\beta)\ddot{\mathbf{q}}(t_n) + h^2\beta\ddot{\mathbf{q}}(t_{n+1})\,, \tag{2.10}$$

where h is the time step, and β and γ are the Newmark parameters. Using these, the properties of the scheme can be chosen. This algorithm is in general implicit, which means that it contains a term dependent on the new time step, $\ddot{\mathbf{q}}(t_{n+1})$. To find this term a linear equation has to be solved. This is easier to describe when the algorithm is written in a slightly different form:

$$\text{Prediction:}\quad \dot{\mathbf{q}}^*(t_{n+1}) = \dot{\mathbf{q}}(t_n) + (1-\gamma)h\ddot{\mathbf{q}}(t_n) \tag{2.11}$$

$$\mathbf{q}^*(t_{n+1}) = \mathbf{q}(t_n) + h\dot{\mathbf{q}}(t_n) + h^2(\frac{1}{2}-\beta)\ddot{\mathbf{q}}(t_n) \tag{2.12}$$

$$\text{Calculate accelerations:}\quad \left(\mathbf{M} + h\gamma\mathbf{C} + h^2\beta\mathbf{K}\right)\ddot{\mathbf{q}}(t_{n+1}) = \mathbf{f}(t_{n+1}) - \mathbf{C}\dot{\mathbf{q}}^*(t_{n+1}) - \mathbf{K}\mathbf{q}^*(t_{n+1}) \tag{2.13}$$

$$\text{Correction:}\quad \dot{\mathbf{q}}(t_{n+1}) = \dot{\mathbf{q}}^*(t_{n+1}) + h\gamma\ddot{\mathbf{q}}(t_{n+1}) \tag{2.14}$$

$$\mathbf{q}(t_{n+1}) = \mathbf{q}^*(t_{n+1}) + h^2\beta\ddot{\mathbf{q}}(t_{n+1})\,. \tag{2.15}$$

The step shown in Eq. (2.13) is the most expensive of the Newmark algorithm. When using a fixed step size the linear equation does have a constant left hand side, which allows reuse of the factorized matrix for each time step.

There are a few possibilities for the Newmark parameters. Some choices will result in an explicit algorithm, which avoids solving the linear system, but at the cost of a very small step size. Others add some numerical damping. The scope of this thesis does not require a thorough investigation of this, so we will use a very common selection for the parameters: $\beta = 1/4$, $\gamma = 1/2$. This yields an unconditionally stable algorithm known as *trapezoidal* or *average constant acceleration*. For an overview of the selection of Newmark parameters, and the resulting accuracy and stability, the reader is referred to the textbook by Hughes [10].

The Newmark scheme has another crucial parameter, the step size h. This controls the trade-off between accuracy and numerical speed. The resulting error will be discussed in Chapter 3.

2.4. Transient analysis & sensitivities

For efficient optimization, we are not just interested in the response of the system, but also in the derivatives (sensitivities) with respect to the design variables. In this section, we discuss sensitivity analysis for transient problems. The literature used for this section includes the textbook by Choi and Kim [11] and review papers by Kang *et al.* [12] and by van Keulen *et al.* [13]. Furthermore, the MSc. thesis of Emiel van de Ven [14] was very useful, which describes the sensitivity analysis for a thermal transient problem in great detail.

In optimization problems, one is interested in the sensitivity of the objective function and the constraints, both with respect to the design variables $\mathbf{x}$. These will be investigated using a very general function:

$$\Psi = \phi(\mathbf{q}(T), \mathbf{x}) + \int_0^T \Phi(\mathbf{q}(\mathbf{t}), \mathbf{x})\, \mathrm{d}t, \tag{2.16}$$

where T is the final time of the simulation. This formulation allows optimization of quantities at the end of the simulation (ϕ, for example final deformation) as well as during the simulation (Φ, for example deviation from a pre-described displacement curve). For completeness, a slightly different formulation should be used, incorporating:

- ϕ and Φ as a function of not only $\mathbf{q}$ and $\mathbf{x}$, but also of $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$.
- A non constant final time T, that is determined by a condition on $\mathbf{q}$ and its derivatives.

These changes do not modify the overall process significantly, both for the direct and adjoint method. They do however add many terms to the equations, making them less insightful. For sensitivity analysis incorporating these two extra possibilities, the reader is referred to respectively the paper by Kang *et al.* [12] and the textbook by Choi and Kim [11].

2.4.1. Global finite differences

The derivative of Eq. (2.16) with respect to $\mathbf{x}$ can be found with minimal implementation effort by a finite difference approximation. This does require one additional simulation per design variable. For the large number of design variables associated with topology optimization, this is not a feasible approach.

2.4.2. Direct differentiation

A more sophisticated approach starts by taking the derivative of the objective function with respect to design variables $\mathbf{x}$:

$$\frac{\mathrm{d}\Psi}{\mathrm{d}\mathbf{x}} = \frac{\partial\phi}{\partial\mathbf{x}} + \frac{\partial\phi}{\partial\mathbf{q}(T)}\frac{\partial\mathbf{q}(T)}{\partial\mathbf{x}} + \int_0^T \frac{\partial\Phi}{\partial\mathbf{x}} + \frac{\partial\Phi}{\partial\mathbf{q}}\frac{\partial\mathbf{q}}{\partial\mathbf{x}}\,\mathrm{d}t. \tag{2.17}$$

These terms can be evaluated separately:

- $\frac{\partial\phi}{\partial\mathbf{x}}, \frac{\partial\Phi}{\partial\mathbf{x}}$: direct dependency of objective function on design variables, usually zero.
- $\frac{\partial\phi}{\partial\mathbf{q}(T)}, \frac{\partial\Phi}{\partial\mathbf{q}}$: easy to evaluate by analytical differentiation of the function considered.
- $\frac{\partial\mathbf{q}}{\partial\mathbf{x}}, \frac{\partial\mathbf{q}(\mathbf{T})}{\partial\mathbf{x}}$: difficult to evaluate. These terms could be replaced by using an adjoint formulation, which will be described in on the next page. First it will be demonstrated how to find them using the direct method.

To find $\frac{\partial \mathbf{q}}{\partial \mathbf{x}}$, we start from the equations of motion, and differentiate these once with respect to the design variables. Regrouping yields:

$$\mathbf{M}\ddot{\mathbf{q}}' + \mathbf{C}\dot{\mathbf{q}}' + \mathbf{K}\mathbf{q}' = \mathbf{f}' - \mathbf{M}'\ddot{\mathbf{q}} - \mathbf{C}'\dot{\mathbf{q}} - \mathbf{K}'\mathbf{q}, \tag{2.18}$$

usually with initial conditions $\dot{\mathbf{q}}'(t=0) = \mathbf{q}'(t=0) = \mathbf{0}$, unless the initial conditions of the equations of motion depend on the design parameters. The right hand side of this equation is called *pseudo-load* and represents the change in load (both external and internal) when the design parameters change. The matrices $\mathbf{M}'$,$\mathbf{C}'$ and $\mathbf{K}'$ can be derived analytically, and $\mathbf{q}$ and its time derivatives are known. So Eq. (2.18) can be solved to find $\mathbf{q}'$. It makes sense to do this using the Newmark algorithm with the same parameters as used in the normal time integration. This enables us to reuse the factorized matrices.

Compared to finite differences, the same number of time integrations is performed, but with the same left hand side for each. So it is more efficient, but still requires many expensive integration steps. Model order reduction can be applied to reduce this [13].

2.4.3. ADJOINT VARIABLE METHOD

A more efficient approach called the Adjoint Variable Method (AVM) will be discussed in this section. This is based on adding a term to the objective function[2], which will allow us to cancel out the expensive term $\frac{\partial \mathbf{q}}{\partial \mathbf{x}}$. For this problem, the additional term is the equations of motion, multiplied by Lagrange multiplier $\boldsymbol{\lambda}^{\mathrm{T}}(t)$, and integrated over the time span.

$$\Psi_{\mathrm{A}} = \Psi + \int_0^T \boldsymbol{\lambda}^{\mathrm{T}}(t) \underbrace{\left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} - \mathbf{f}\right)}_{=\mathbf{0}} \mathrm{d}t. \tag{2.19}$$

For now we focus on the added term. This should contain the terms that we want to cancel out in the sensitivity. Differentiating once with respect to $\mathbf{x}$ yields after some reordering:

$$\begin{aligned} &\int_0^T \boldsymbol{\lambda}^{\mathrm{T}}\left(\mathbf{M}\ddot{\mathbf{q}}' + \mathbf{C}\dot{\mathbf{q}}' + \mathbf{K}\mathbf{q}'\right) - \mathbf{R}\,\mathrm{d}t = \mathbf{0} \\ &\mathbf{R} = \boldsymbol{\lambda}^{\mathrm{T}}\left(\mathbf{f}' - \mathbf{M}'\ddot{\mathbf{q}} - \mathbf{C}'\dot{\mathbf{q}} - \mathbf{K}'\mathbf{q}\right). \end{aligned} \tag{2.20}$$

Next integration by parts is applied to reduce the order of the time derivatives of $\mathbf{q}'$:

$$\int_0^T \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{M}\ddot{\mathbf{q}}'\,\mathrm{d}t = \int_0^T \ddot{\boldsymbol{\lambda}}^{\mathrm{T}}\mathbf{M}\mathbf{q}'\,\mathrm{d}t - \left[\dot{\boldsymbol{\lambda}}^{\mathrm{T}}\mathbf{M}\mathbf{q}' + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{q}}'\right]_0^{\mathrm{T}} \tag{2.21}$$

$$\int_0^T \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{C}\dot{\mathbf{q}}'\,\mathrm{d}t = -\int_0^T \dot{\boldsymbol{\lambda}}^{\mathrm{T}}\mathbf{C}\mathbf{q}'\,\mathrm{d}t + \left[\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{C}\mathbf{q}'\right]_0^{\mathrm{T}} \tag{2.22}$$

$$\begin{aligned} \int_0^T \boldsymbol{\lambda}^{\mathrm{T}}\left(\mathbf{M}\ddot{\mathbf{q}}' + \mathbf{C}\dot{\mathbf{q}}' + \mathbf{K}\mathbf{q}'\right) - \mathbf{R}\,\mathrm{d}t = &\int_0^T \left(\left(\ddot{\boldsymbol{\lambda}}^{\mathrm{T}}\mathbf{M} - \dot{\boldsymbol{\lambda}}^{\mathrm{T}}\mathbf{C} + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{K}\right)\mathbf{q}' - \mathbf{R}\right)\mathrm{d}t \\ &- \left[\dot{\boldsymbol{\lambda}}^{\mathrm{T}}\mathbf{M}\mathbf{q}' + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{q}}' + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{C}\mathbf{q}'\right]_0^{\mathrm{T}} = 0. \end{aligned} \tag{2.23}$$

For simplicity it is assumed that the initial conditions do not depend on the design variables, so $\dot{\mathbf{q}}'(0) = \mathbf{q}'(0) = \mathbf{0}$; and that all matrices are symmetric. Adding Eq. (2.23) to Eq. (2.17) and regrouping yields:

$$\begin{aligned} \frac{\mathrm{d}\Psi_{\mathrm{A}}}{\mathrm{d}\mathbf{x}} = &\frac{\partial \phi}{\partial \mathbf{x}} + \int_0^T \frac{\partial \Phi}{\partial \mathbf{x}} - \mathbf{R} + \underbrace{\left(\mathbf{M}\ddot{\boldsymbol{\lambda}} - \mathbf{C}\dot{\boldsymbol{\lambda}} + \mathbf{K}\boldsymbol{\lambda} + \frac{\partial \Phi}{\partial \mathbf{q}}^{\mathrm{T}}\right)^{\mathrm{T}}}_{\text{Adjoint equation}} \frac{\partial \mathbf{q}}{\partial \mathbf{x}}\,\mathrm{d}t \\ &- \underbrace{\left(\mathbf{M}\dot{\boldsymbol{\lambda}}(T) + \mathbf{C}\boldsymbol{\lambda}(T) - \frac{\partial \phi}{\partial \mathbf{q}(T)}^{\mathrm{T}}\right)^{\mathrm{T}}}_{\text{Terminal condition 1}} \frac{\partial \mathbf{q}(T)}{\partial \mathbf{x}} + \underbrace{\boldsymbol{\lambda}^{\mathrm{T}}(T)\mathbf{M}}_{\text{Term. cond. 2}} \frac{\partial \dot{\mathbf{q}}(T)}{\partial \mathbf{x}}. \end{aligned} \tag{2.24}$$

[2] This can also be interpreted as switching the order of the calculations[15]. In the direct differentiation method, the pseudo-load is calculated first, and time integration is used to calculate the resulting sensitivity. In the adjoint method, the adjoint variable is calculated first. This term ($\boldsymbol{\lambda}$) can be interpreted as the compliance of the objective with respect to changes of the internal and external loads[16]. Multiplying this with the pseudo-load and integrating over time yields the design sensitivity.

Thus when the following equations hold, all term in the sensitivity containing $\mathbf{q}'$ are zero:

$$\mathbf{M}\ddot{\boldsymbol{\lambda}} - \mathbf{C}\dot{\boldsymbol{\lambda}} + \mathbf{K}\boldsymbol{\lambda} = -\frac{\partial \Phi}{\partial \mathbf{q}}^{\mathrm{T}} \qquad \text{(coefficients of } \mathbf{q}' \text{ in integral)} \tag{2.25}$$

$$\mathbf{M}\boldsymbol{\lambda}(T) = \mathbf{0} \qquad \text{(coefficients of } \dot{\mathbf{q}}'(T)\text{)} \tag{2.26}$$

$$\mathbf{M}\dot{\boldsymbol{\lambda}}(T) = -\mathbf{C}\boldsymbol{\lambda}(T) + \frac{\partial \phi}{\partial \mathbf{q}(T)}^{\mathrm{T}} \qquad \text{(coefficients of } \mathbf{q}'(T)\text{)}\,. \tag{2.27}$$

This results in the adjoint variable sensitivity analysis, which is not dependent on $\mathbf{q}'$. It does require the computation of $\boldsymbol{\lambda}$, subject to the terminal value problem of equation 2.25 to 2.27. Equation (2.25) will be called the *adjoint equation*.

If we had used a more general function ψ, the right hand side of all three equations would contain additional terms. These can all be easily evaluated analytically and do not change the overall complexity.

Finally, combining everything yields the following equation for the sensitivity:

$$\begin{aligned}\frac{\mathrm{d}\Psi_{\mathrm{A}}}{\mathrm{d}\mathbf{x}} &= \frac{\partial \phi}{\partial \mathbf{x}} + \int_0^T \frac{\partial \Phi}{\partial \mathbf{x}} - \mathbf{R}\,\mathrm{d}t \\ \mathbf{R} &= \boldsymbol{\lambda}^{\mathrm{T}}\left(\mathbf{f}' - \mathbf{M}'\ddot{\mathbf{q}} - \mathbf{C}'\dot{\mathbf{q}} - \mathbf{K}'\mathbf{q}\right).\end{aligned} \tag{2.28}$$

Because the equations of motion will have been solved before starting the sensitivity analysis, the dynamic response $\mathbf{q}$ and its derivatives are available. The adjoint variable can be found by solving the adjoint equation. The other terms can easily be evaluated analytically.

The required computational effort is low compared to the other methods, only one time integration is required. This step is very similar to the normal time integration, so we can even reuse the factorized matrices. However, the full time history of $\mathbf{q}$ and its derivatives has to be stored, which requires a large amount of memory.

2.4.4. COMPARISON BETWEEN STATICS, FREQUENCY-DOMAIN AND TIME-DOMAIN DYNAMICS

Up to this point the simulation and sensitivity analysis for time-domain dynamics have been studied. To put this into context, this can be compared to problems concerning statics and frequency-domain dynamics.

Most topology optimization problems use linear statics. A typical problem studies the static displacement $\mathbf{q}$, which can be found by solving:

$$\mathbf{K}\mathbf{q} = \mathbf{f}. \tag{2.29}$$

The sensitivity analysis (adjoint variable method) can be summarized by[7]:

$$\begin{aligned}\Psi &= \phi(\mathbf{q}) \\ \frac{\mathrm{d}\Psi}{\mathrm{d}\mathbf{x}} &= \frac{\partial \Psi}{\partial \mathbf{x}} + \boldsymbol{\lambda}^{\mathrm{T}}\left(\mathbf{f}' - \mathbf{K}'\mathbf{q}\right) \\ \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{K} - \frac{\partial \phi}{\partial \mathbf{q}} &= \mathbf{0}.\end{aligned} \tag{2.30}$$

In general this means that for the simulation and sensitivity analysis two solutions are needed of a system with matrix $\mathbf{K}$. In the common case of minimum compliance optimization ($\Psi = \mathbf{f}^{\mathrm{T}}\mathbf{q}$), $\boldsymbol{\lambda}$ will be equal to $\mathbf{q}$, which reduces the entire analysis to just one solution of a linear system and a number of matrix multiplications.

Frequency domain dynamics studies the behaviour of the structure under a harmonic load with frequency ω. It is assumed that the force and displacement are given by[9]:

$$\mathbf{f} = \bar{\mathbf{f}}e^{i\omega t}, \qquad \mathbf{q} = \bar{\mathbf{q}}e^{i\omega t}. \tag{2.31}$$

Substituting this into the equations of motion yields:

$$\left(-\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K}\right)\bar{\mathbf{q}} = \mathbf{S}\bar{\mathbf{q}} = \bar{\mathbf{f}}. \tag{2.32}$$

To find the amplitude of the system at a certain frequency, one solution of a linear system with matrix **S** is needed. For the sensitivity analysis we again use an adjoint formulation, with a general function of the displacement response as objective[17]:

$$\begin{aligned} \Psi &= \phi(\bar{\mathbf{q}}) \\ \frac{\mathrm{d}\Psi}{\mathrm{d}\mathbf{x}} &= \frac{\partial\Psi}{\partial\mathbf{x}} + \boldsymbol{\lambda}^{\mathrm{T}}\left(\bar{\mathbf{f}}' - \mathbf{S}'\bar{\mathbf{q}}\right) \\ \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{S} - \frac{\partial\phi}{\partial\mathbf{q}} &= \mathbf{0}\,. \end{aligned} \tag{2.33}$$

The similarity to the static problem is obvious, the solution will take a similar amount of computational effort, although this is increased by the fact that matrix **S** is complex. Furthermore, using the *dynamic compliance* ($\Psi = \bar{\mathbf{f}}^{\mathrm{T}}\bar{\mathbf{q}}$) as objective function will reduce the costs of the sensitivity analysis [17], analogous to the minimum compliance case for a static problem. This would require only one solution of a linear system for simulation plus sensitivity analysis.

Summarizing, the statics and frequency domain problems described here require a similar amount of computational effort to solve, in general two solutions of linear systems per objective function. In contrast, for time-domain problems with full Newmark integration two solutions of similar systems are required for every timestep. A typical simulation uses several hundred timesteps, which results in a computational effort that is two orders of magnitude larger.

For both statics and frequency domain dynamics the sensitivity analysis is simplified when using the compliance of the system as objective function. In some cases this is also applicable to time-domain dynamics, this will briefly be discussed in Section 3.3.

To conclude this discussion, we can state that simulation and sensitivity analysis for time-domain dynamics is significantly more expensive than for statics or frequency-domain dynamics.

2.5. Model order reduction

Because the finite element method typically results in a very large system of equations, it is attractive to reduce these to a smaller system. This process is called model order reduction, and is applied in many different fields, including structural mechanics, numerical mathematics and control engineering, for a comparison see [18]. In this thesis the methods and nomenclature as used in structural mechanics are chosen.

2.5.1. Methods for linear systems

For linear systems, model order reduction means we represent N coordinates $\mathbf{q}$ by a set of *basis vectors* $\mathbf{V}$ and *reduced coordinates* $\mathbf{z}$. Mathematically this can be seen as projecting the displacement onto the space spanned by $\mathbf{V}$, which has a lower dimension. This results in a *residual displacement* $\mathbf{r}$:

$$\underset{N\times 1}{\mathbf{q}(t)} = \underset{N\times n}{\mathbf{V}}\ \underset{n\times 1}{\mathbf{z}(t)} + \underset{N\times 1}{\mathbf{r}(t)} \simeq \mathbf{V}\mathbf{z}(t)\,. \tag{2.34}$$

This equation is possibly clearer when the size of matrices is illustrated:

$$\begin{bmatrix} q_1(t) \\ \\ \vdots \\ \\ q_N(t) \end{bmatrix} \approx \begin{bmatrix} \\ \mathbf{v}_1 & \cdots & \mathbf{v}_n \\ \\ \end{bmatrix} \begin{bmatrix} z_1(t) \\ \vdots \\ z_n(t) \end{bmatrix} \tag{2.35}$$

This method be applied to many types of problems, not just limited to structural mechanics. Focussing on topology optimization for structural mechanics, there are example of statics[19] and frequency-domain dynamics[17]. The scope of this research and thus this section will be limited to time-domain dynamics.

Analogous to Eq. (2.34) the velocity and acceleration are also projected onto $\mathbf{V}$. Substituting this into the equations of motion, and pre-multiplying by $\mathbf{V}^{\mathrm{T}}$ yields a set of n equations, which we call the *reduced equations of motion*:

$$\begin{aligned} \mathbf{M}_{\mathrm{r}}\ddot{\mathbf{z}} + \mathbf{C}_{\mathrm{r}}\dot{\mathbf{z}} + \mathbf{K}_{\mathrm{r}}\mathbf{z} &= \mathbf{V}^{\mathrm{T}}\mathbf{f} + \mathbf{V}^{\mathrm{T}}\hat{\mathbf{r}} \\ \mathbf{M}_{\mathrm{r}} &= \mathbf{V}^{\mathrm{T}}\mathbf{M}\mathbf{V} \\ \mathbf{C}_{\mathrm{r}} &= \mathbf{V}^{\mathrm{T}}\mathbf{C}\mathbf{V} \\ \mathbf{K}_{\mathrm{r}} &= \mathbf{V}^{\mathrm{T}}\mathbf{K}\mathbf{V} \\ \hat{\mathbf{r}} &= -(\mathbf{M}\ddot{\mathbf{r}} + \mathbf{C}\dot{\mathbf{r}} + \mathbf{K}\mathbf{r}) = \mathbf{M}\mathbf{V}\ddot{\mathbf{z}} + \mathbf{C}\mathbf{V}\dot{\mathbf{z}} + \mathbf{K}\mathbf{V}\mathbf{z} - \mathbf{f}\,. \end{aligned} \tag{2.36}$$

The magnitude of the basis vectors can be chosen freely, convention is to orthonormalize every basis vector such that $\mathbf{V}^{\mathrm{T}}\mathbf{M}\mathbf{V} = \mathbf{I}$. The solution of the reduced equations of motion shall be reffered to as *reduced solution*. The residual displacement $\mathbf{r}$ can be studied in some more detail by describing it by:

$$\underset{N\times 1}{\mathbf{r}} = \underset{N\times(N-n)}{\mathbf{V}_r} \cdot \underset{(N-n)\times 1}{\mathbf{z}_r}\,, \tag{2.37}$$

where $\mathbf{V}_r$ and $\mathbf{V}$ together span the entire solution space. Using $\mathbf{V}_r$, the error can be rewritten as:

$$\mathbf{V}^{\mathrm{T}}\hat{\mathbf{r}} = -\mathbf{V}^{\mathrm{T}}(\mathbf{M}\mathbf{V}_r\ddot{\mathbf{z}}_r + \mathbf{C}\mathbf{V}_r\dot{\mathbf{z}}_r + \mathbf{K}\mathbf{V}_r\mathbf{z}_r)\,. \tag{2.38}$$

This shows that the *force residual* $\mathbf{V}\hat{\mathbf{r}}$ will only be zero if the used basis vectors are mass-, damping- and stiffness-orthogonal. Any coupling between the basis vectors will result in inaccuracy in $\mathbf{z}$. It is always possible to choose $\mathbf{V}_r$ such that is mass- and stiffness-orthogonal with respect to $\mathbf{V}$[3]. In the case of modal damping, orthogonality with respect to damping is also ensured. This means $\mathbf{V}^{\mathrm{T}}\hat{\mathbf{r}}$ will be zero. In mathematics this is known as a *Galerkin projection.*

[3] For any type of basis vectors diagonal system matrices can be found by calculating the eigenvectors of the reduced system. This allows uncoupling of the basis vectors. The exception here is the damping matrix, which will only be diagonal in the case of modal damping.

The choice of basis vectors is very important. A good set should retain most of the characteristics of the full system using only a few vectors, which means that the goal is to minimize **r**. On the other hand, the basis vectors should be cheap to compute, otherwise part of the computational gain is lost. The most important choices are named below. These can be grouped as follows:

- Model based reduction
  - Modal based reduction[4]
    - ◇ *Modal Superposition Method (MSM)*
  - Load based reduction
    - ◇ *Mode Acceleration Method (MAM)*
    - ◇ *Ritz vectors*
    - ◇ *Quasi-Static Ritz Vectors (QSRV)*
  - Condensation methods
  - Combined methods
- Data based reduction
  - *Proper Orthogonal Decomposition (POD)*

MODAL SUPERPOSITION METHOD

The basis vectors in this method are the eigenvectors of the system[20]. This is also commonly called *Modal Displacement Method* or *Modal Truncation*, which implies that we truncate the full basis to find the reduced basis. In this research the full basis is never used, MSM is always used to refer to the reduced set of eigenvectors,. These are the solutions of:

$$(\mathbf{K}-\omega_i^2\mathbf{M})\mathbf{v}_i = \mathbf{0}\,. \tag{2.39}$$

Each eigenvector has an associated eigenfrequency ω_i. For MOR, eigenvectors associated to the lowest n eigenfrequencies are commonly used. There are various algorithms for finding eigenvectors. The most frequently used is the Lanczos algorithm[5], which requires the solution of 2-4 linear systems per vector [21].

Eigenvectors are orthogonal with respect to both the mass and stiffness matrix. If they are not coupled through the damping matrix, all system matrices will be diagonal, which reduces the cost of solving them.

The convergence of the MSM can be studied by examining the undamped response to a harmonic excitation: $\mathbf{f}(t) = \mathbf{L}\sin(\omega t)$. This response can be expressed as a function of the eigenvectors:

$$\mathbf{q}(t) = \sin(\omega t)\sum_{i=1}^{n}\frac{\mathbf{V}_i\mathbf{V}_i^{\mathrm{T}}\mathbf{L}}{(\omega_i^2-\omega^2)}\,. \tag{2.40}$$

One can distinguish between spatial and spectral convergence. Spatial convergence depends on $\mathbf{V}^{\mathrm{T}}\mathbf{L}$, which is difficult to predict without computing the eigenvector. Spectral convergence depends on $\frac{1}{\omega_i^2-\omega^2}$. So vectors with eigenfrequencies close to the excitation will have a larger amplitude, those with distant eigenfrequencies a smaller amplitude. This leads to the common method of including all eigenfrequencies up to a multiple of the load frequency.

Eigenvectors are not dependent on the load, this is both an advantage and a disadvantage. When the load on a structure is changed, the eigenvectors do not have to be recomputed. But the lack of load information in the basis can lead to large errors in the calculated response.

[4] Note the distinction between *model* and *modal*, modal refers to eigenmodes of the model.

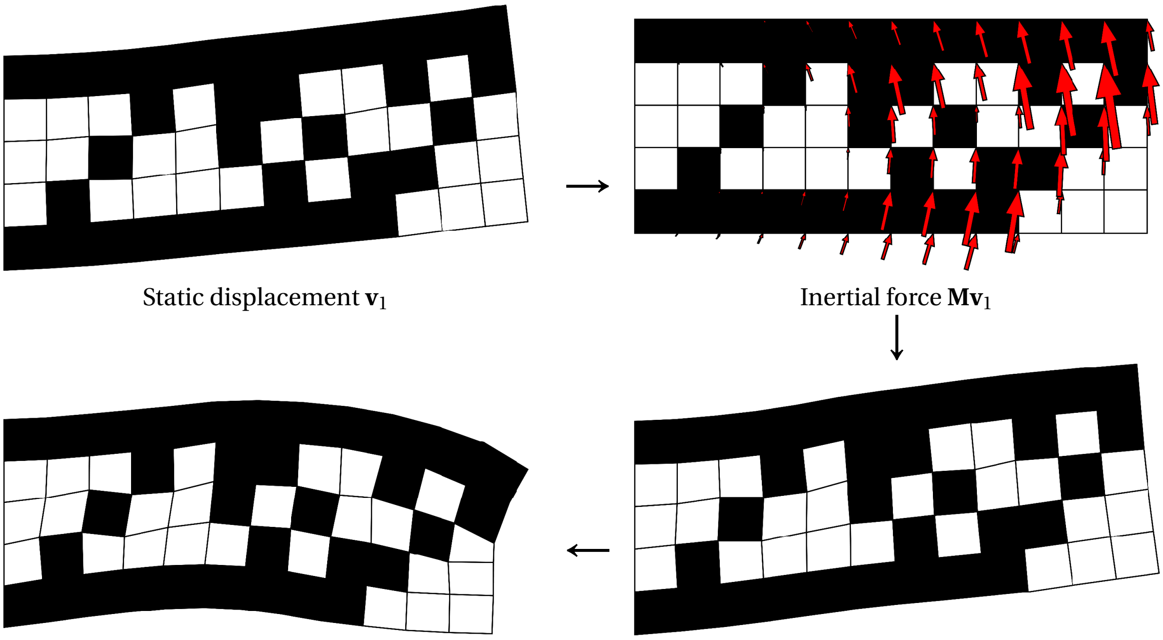
[5] The Lanczos algorithm is based on reducing the equations of motion, and finding the eigenvectors of the reduced system. The reduced basis consists of one random vector plus additional vectors generated using the Arnoldi algorithm, which will be discussed in the next subsection. The main cost is calculating the reduced basis, which is typically 2-4 times larger than the requested number of eigenvectors. For details, see [9, 21].

RITZ VECTORS

Also known Load Dependent Ritz Vector method [22, 23]. This method is based on creating a (Krylov) subspace from the static displacement and subsequent multiplications by $\mathbf{K}^{-1}\mathbf{M}$:

$$\begin{aligned} \mathbf{v}_1 &= \mathbf{K}^{-1}\mathbf{f} \\ \mathbf{V} &= \mathrm{span}\{\mathbf{v}_1, \mathbf{K}^{-1}\mathbf{M}\mathbf{v}_1, (\mathbf{K}^{-1}\mathbf{M})^2\mathbf{v}_1, \ldots, (\mathbf{K}^{-1}\mathbf{M})^{n-1}\mathbf{v}_1\}\,. \end{aligned} \tag{2.41}$$

This has a clear physical interpretation, illustrated in Fig. 2.4. The first vector is the static displacement $\mathbf{v}_1$. If we would have an acceleration in this direction, we would find inertia forces $\mathbf{M}\mathbf{v}_1$, resulting in an approximate displacement $\mathbf{K}^{-1}\mathbf{M}\mathbf{v}_1$. Each Ritz vector can be seen as a dynamic correction to the previous one. For better numerical results, these vectors should be orthonormalized. This leads to Algorithm 2.1, where steps 3-9 are known as the *Arnoldi algorithm.*


Static displacement $\mathbf{v}_1$

Inertial force $\mathbf{M}\mathbf{v}_1$

New vector $\mathbf{v}_2$ (orthogonalized)

Dynamic correction $\mathbf{K}^{-1}\mathbf{M}\mathbf{v}_1$

Figure 2.4: Illustration of physical interpretation of Ritz vectors. The first vector is the static displacement of the structure. To approximate the dynamic response of the structure, the first vector is multiplied by the mass matrix, yielding the associated inertia forces. The displacement caused by these forces is approximated by multiplying with $\mathbf{K}^{-1}$. This results in the new Ritz vector, but there is still a large overlap between the vectors. To remove this the new vector is orthogonalized. These steps illustrate the fact that each vector is a dynamic correction to the previous one.

Algorithm 2.1 Algorithm for calculating Ritz vectors

```
1: $\mathbf{v}_1 \leftarrow \mathbf{K}^{-1}\mathbf{f}$ {Create first vector}
2: $\mathbf{v}_1 \leftarrow \mathbf{v}_1 / \sqrt{\mathbf{v}_1^T\mathbf{M}\mathbf{v}_1}$ {Mass-normalize}
3: for $i = 2$ to $n$ do
4:     $\mathbf{v}_i \leftarrow \mathbf{K}^{-1}\mathbf{M}\mathbf{v}_{j-1}$ {Create new vector}
5:     for $j = 1$ to $i-1$ do
6:         $\mathbf{v}_i \leftarrow \mathbf{v}_i - \mathbf{v}_j^T\mathbf{M}\mathbf{v}_i\mathbf{v}_j$ {Mass-orthogonalize}
7:     end for
8:     $\mathbf{v}_i \leftarrow \mathbf{v}_i / \sqrt{\mathbf{v}_i^T\mathbf{M}\mathbf{v}_i}$ {Mass-normalize}
9: end for
```

Steps 5-7 form a variant on the *modified Gramm-Schmidt* process[24]. Normally this produces vectors that are orthogonal to each other, in this version the vectors will be mass-orthogonal. Although other orthogonalization algorithms are also possible and might offer better numerical stability, the used algorithm is cheap and in most cases sufficient. Orthogonalization errors will lead to poorly conditioned reduced system matrices, which can easily be detected.

Compared to eigenvectors, Ritz vectors are cheap to compute, requiring only one solution of a linear system per vector. The reduced mass matrix will be diagonal, and all its entries are 1. The reduced stiffness matrix however will generally be full, which increases solution effort, and complicates analytical solutions. Two possible methods[25] to improve this are either diagonalizing the reduced matrices by computing the eigenvectors of the reduced system or switching to so called *Lanczos coordinates*, which result in tri-diagonal matrices. If the time integration of the reduced system is a major cost in the total optimization, either of these methods can be used to lower this.

A disadvantage of Ritz vectors is the fact that a clearly defined load vector is required. If there is no load vector, there is no static displacement. This means the load should be of the form $\mathbf{f}(t) = \mathbf{L} \cdot f(t)$ (Eq. (2.8)), a constant spatial vector times a scalar term. If the load can be described using a summation of a few of these terms (for example one load vector for the gravity load, one for an external load, etc.), separate Ritz vectors can be calculated for each part of the load. This will become inefficient for complicated loads.

On the convergence of Ritz vectors we can make an important note. Because of the repeated multiplication with $\mathbf{K}^{-1}\mathbf{M}$ (*power iteration* procedure) the subspace will converge towards the first eigenvector. Once the first eigenvector is included in the subspace, the second eigenvector will start appearing, etc. Based on this, we can expect a good representation of low frequency content.

For problems featuring rigid body modes, applying MSM is straight-forward. The rigid body modes will appear as basis vectors with zero stiffness. For Ritz vectors, they do cause some issues, since matrix $\mathbf{K}$ will be singular. This can be solved by including the rigid body modes (nullspace of $\mathbf{K}$) in the basis, and using a generalized inverse for the rest of the computations.

Quasi-Static Ritz Vectors

This is a variation on Ritz vectors [26]. The difference is in the power iteration procedure of step 1 (and the generation of the first vector). We replace $\mathbf{K}^{-1}$ by $(\mathbf{K} - \omega_{\mathrm{c}}^2\mathbf{M})^{-1}$, where ω_c is the so called centre frequency. This results in a spectral shift, so the subspace will converge towards the eigenvector closest to the centre frequency. Based on this, we can expect a higher accuracy for problems where the frequencies close to ω_{c} are relevant. Possible variations include multiple values for ω_{c}, or ways to treat multiple load cases, both using the block Lanczos method, see [26, 27].

A difficulty with this method is the fact that when ω_c approaches an eigenvalue, we have to invert/factorize a (nearly) singular matrix. Because eigenvalues are likely to change during optimization, this should at least be monitored. Rigid body modes however will not be a problem, these do not cause singularities in the iteration matrix.

Condensation methods

Another type of reduction are condensation methods. These consist of 'condensing' a subset of the degrees of freedom onto the remaining ones. In the most common method, Guyan Reduction [28], a part of the degrees of freedom is quasi-statically condensed. If this part has a low inertia, this assumption will not lead to a large error.

There could be some useful applications of this in topology optimization, for example condensing the void domain onto the solid domain to reduce the size of the system without affecting the accuracy significantly. The order reduction however depends on the size of the structure that we do want to model fully, and this limits the overall reduction. So compared to the other model based methods, static condensation will require a larger basis.

Modal acceleration method

By reducing the system inevitably some of the information in the response will be lost. In the modal superposition approach, the residual can be approximated. In the so called MAM [21], a quasi-static term is added. This means we ignore the inertia and damping of the system, and just look at the stiffness. Using spectral expansion of the stiffness matrix, this quasi-static correction can be written as either:

$$\mathbf{q}_{\text{qs}} = \sum_{i=n+1}^{N} \frac{\mathbf{v}_i \mathbf{v}_i^{\text{T}} \mathbf{f}}{\mu_i \omega_i^2} \tag{2.42}$$

$$\mathbf{q}_{\text{qs}} = \mathbf{K}^{-1}\mathbf{f} - \sum_{i=1}^{n} \frac{\mathbf{v}_i \mathbf{v}_i^{\text{T}} \mathbf{f}}{\mu_i \omega_i^2}\,. \tag{2.43}$$

The first form requires a full eigenbasis, so the second form is usually more convenient. This addition represents the quasi-static response of the part of the system that is left out by the MOR.

The higher modes have a high stiffness/mass ratio, so a quasi-static approximation can be very accurate. The computational effort is dependent on the calculation of $\mathbf{K}^{-1}\mathbf{f}$. This term has to be calculated for every timestep. Note how this compares to the (full) Newmark method, the solution of a similar system is required. This of course nullifies the performance gained by MOR. Fortunately, there are two factors that speed up the calculation. First of all, the factorized **K** matrix is often already available from an earlier calculation. Secondly if the load can be described as $\mathbf{f}(t) = \mathbf{L} \cdot f(t)$, the quasi-static correction can be rewritten as:

$$\mathbf{q}_{\text{qs}} = \left(\mathbf{K}^{-1}\mathbf{L} - \sum_{i=1}^{n} \frac{\mathbf{v}_i \mathbf{v}_i^{\text{T}} \mathbf{L}}{\mu_i \omega_i^2} \right) f(t)\,, \tag{2.44}$$

which is very cheap to calculate, since the term within brackets only has to be evaluated once. This condition on the load is essential for both efficient MAM and load-dependent MOR.

The Modal Acceleration Method in this form is only applicable to MSM, because it relies on an eigendecomposition of the system matrices.

One final remark can be made on the relation between MAM and Ritz vectors. The MAM is identical to adding a single orthogonalized Ritz vector to the MSM basis, and calculating its associated quasi-static response. Based on this MAM will always have inferior accuracy compared to modal truncation augmentation, which will be discussed later.

Proper Orthogonal Decomposition

In this technique the basis vectors are calculated from a set of samples, most commonly using a singular value decomposition [29]. This method is easy to apply to non-linear systems, and it is often used in computational fluid dynamics. However generating the samples is expensive, since it requires many solutions of the full equations. Thus for this research it does not make sense to use POD.

2.5.2. Combined methods

A reduction basis can be built out of various types of vectors, including a combination of the methods discusses earlier. The one exception here is the MAM, which is not a reduction method, but a correction to one. Combining basis vectors allows us to use the strength of each method. It is recommended to always mass-orthogonalize all vectors with respect to one other for higher numerical accuracy and speed.

Modal Truncation Augmentation

The most frequently used combined method is called *modal truncation augmentation (MTA)* [30], which uses a basis of eigenvectors augmented with Ritz vectors. This is closely connected to the modal acceleration method, and usually outperforms it[31]. A further advantage of this method is that the Ritz vectors are usually calculated as an intermediate step in the Lanczos algorithm used to find the eigenvectors.

POD & load

Furthermore there is some merit in the combination of dependent vectors and POD. As stated before, we need a spatial representation of the load to calculate load dependent vectors. If this is not available, this can be approximated by applying proper orthogonal decomposition to the load data. The vectors produced by this method can be used as load vectors for the model reduction.

DIAGONALIZATION

Finally for every type of reduction we can diagonalize the reduced system matrices by calculating their eigenbasis, according to:

$$\left(\mathbf{K}_\mathrm{r} - \omega_{\mathrm{r},i}\mathbf{M}_\mathrm{r}\right)\mathbf{w}_i = \mathbf{0}\,. \tag{2.45}$$

where $\mathbf{W}$ is the eigenbasis of the reduced system. Ignoring damping, this yields the following uncoupled set of basis vectors:

$$\mathbf{V}_\mathrm{diag} = \mathbf{VW}\,. \tag{2.46}$$

This uncouples the equations of motion[6], making analytical solutions possible even for load based reduction methods. This also increases the numerical stability of the model by forcing the basis to be orthogonal. Thus built-up of errors from the orthogonalization is reduced.

2.5.3. COMPARISON & ESTIMATED COMPUTATIONAL GAIN

Now that the different methods have been described, the next step is comparing them. Eigenvectors are expensive to calculate compared to Ritz vectors. They are mostly suited to systems where the eigenmodes dominate the dynamics, such as resonating systems. If the load dominates the dynamics, we usually need a high number of eigenvectors for a good approximation. Load dependent vectors will be better in this case, Ritz vectors for low frequency loads and QSRV for high frequency loads. If an individual method is insufficient, a combination might work, such as Model Truncation Augmentation.

The modal acceleration method can increase the accuracy of MSM at a low cost, if the load is described by a term that has a constant spatial distribution. This requirement is also applicable to the other load-based reduction methods. In contrast to all the other methods the MAM does require computing the full solution ($\mathbf{q} = \mathbf{Vz}$), which is not expensive but does require a large amount of memory. Other methods (Ritz vectors, modal truncation augmentation) are expected to result in higher accuracy with lower costs.

Before we actually start using model order reduction, we can make an estimate of the improvement in computational effort. The most expensive step in both the Newmark algorithm and the most important reduction methods (MSM, Ritz, QSRV) is the repeated solving of a linear system. The computational costs associated with this can be estimated. For dense matrices factorization has a cost of $N^3/3$ (Cholesky), the cost of solution of the system by substitution is $2N^2$. This results in the following total costs:

Table 2.1: Estimated costs of different MOR methods in flops. c is the number of Lanczos iterations per computed eigenvector, N the size of the full system, n the number of basis vectors, n_t the number of time steps. This shows that the main effect of model order reduction is reducing the number of times the full system has to be solved.

Method	Factorization (full)	Solution (full)	Factorization (red.)	Solution (red.)
Full	$N^3/3$	$n_t \cdot 2N^2$	-	-
MSM	$N^3/3$	$n \cdot 2N^2 \cdot c$	-	$n_t \cdot 2n^2$
Ritz/QSRV	$N^3/3$	$n \cdot 2N^2$	$n^3/3$	$n_t \cdot 2n^2$

This is a very rough estimate, the exact costs will be highly dependent on the sparsity and bandwidth of the matrices. The most expensive computations are those performed on the full equations. Since all methods use the matrices found from the same finite element analysis, the costs of factorization are equal. The required number of solutions of the full system determines the possible cost reduction. This step is often one of the bottlenecks of the simulation. Based on this, it can be concluded that model order reduction is efficient if the required number of basis vectors is small compared to the number of time steps.

2.5.4. OTHER METHODS

We will discuss two other research fields: dynamic substructuring and parametric model order reduction.

In dynamic substructuring [32], of which the most common type is component mode synthesis, a structure is divided into parts. This by itself does not reduce the cost of the analysis. However, if we have clearly divided

[6] Note that although basis $\mathbf{V}_\mathrm{diag}$ uncouples the system matrices, the basis vectors are not necessarily identical to the exact eigenvectors of the system.

subcomponents, these can be analysed and changed individually, which does simplify analysis . Reduction methods are also frequently applied to subcomponents. In topology optimization, we often have only one component, in which case there is no need for dynamic substructuring. In some cases it might be beneficial, for example if a part of the structure is not changed, or if the structure has clearly defined subcomponents.

Parametric model order reduction is more interesting for this research. In this field, reduced models are generated as a function of certain parameters. By using various interpolation techniques, a reduced model can be cheaply obtained for different values of these parameters within a certain design space.

An attractive idea would be using this to update the basis vectors when the design changes. This does require a large number of samples, to represent the design space. In topology optimization, the design space has a very high dimension. Because analysis is expensive for transient problems this excludes the use of parametric model order reduction. Other methods of updating the basis will be discussed in Section 2.5.6.

2.5.5. ERROR ESTIMATES

The choice of the number of basis vectors is a difficult one. The perfect reduced model produces results identical to the full model. In reality MOR will introduce an error. If we can estimate this error, we can choose the number of vectors such that this error is within a certain range, a process called *apaptive model order reduction*. A number of few error estimates here will be discussed here. They are all normalized such that an estimate of 0 represents a perfect approximation, while an estimate in the order of 1 represent a very inaccurate representation.

FULL ANALYSIS

First of all, we can compare the results to those obtained from the full model. This is costly, the point of MOR is to avoid computations on the full model. In optimization however it might be wise to verify the accuracy of the reduced model once or twice. If the extra costs guarantee that the hundreds of iterations that follow are accurate, they are easily justified.

The error is of course given by $\mathbf{q}_{\text{full}} - \mathbf{Vz}$. But there are multiple ways of computing a scalar normalized error measure from this. The following one will be used in this document:

$$e = \frac{\|q_{j,\text{full}}(t_i) - q_{j,\text{red}}(t_i)\|}{\|q_{j,\text{full}}(t_i)\|}, \qquad 0 \le i \le n_t, \tag{2.47}$$

where j corresponds to a certain DoF, usually the one where a force is applied. There are many other possibilities, which for example use the error information for all DoFs. In Chapter 5 we will see this one is sufficient.

This will be used as the reference error. A good estimate will produce the same results at a lower cost. Some possible error estimates will be discussed next. The main distinction is whether the error is estimated before (*a priori*) or after the simulation (*a posteriori*). The following methods will be described:

- A priori estimates:
  - *Spatial load estimate*
  - *Spectral load estimate*
- A posteriori estimates:
  - *Dual Weighted Residual estimate*
  - *Refinement estimate.*

SPATIAL LOAD ESTIMATE

The first estimate is based on the load. It indicates how well the direction of the load is represented by the basis vectors [23]:

$$e = \frac{\mathbf{f}^{\mathrm{T}}\mathbf{e}}{\mathbf{f}^{\mathrm{T}}\mathbf{f}}, \tag{2.48}$$

where $\mathbf{e}$ is given by:

$$\mathbf{e} = \mathbf{f} - \sum_{i=1}^{n} \mathbf{v}_i^{\mathrm{T}} \mathbf{f} \mathbf{M} \mathbf{v}_i \,. \tag{2.49}$$

The value of e will vary between 1 when the load is fully mass-orthogonal to the vectors, to 0 when the load is fully represented.

Spectral load estimate

For the MSM we have seen that spectral convergence can easily be estimated. For the other methods this is also possible, by calculating the eigenspectrum of the reduced system. A necessary condition for accurate model order reduction is that the frequency spectrum of the load and the reduced model overlap. Two error estimates based on this are[33]:

$$e = \frac{\max \omega_{\mathrm{load}}}{\max \omega_{\mathrm{red}}} \,, \qquad e = \frac{\min \omega_{\mathrm{red}}}{\min \omega_{\mathrm{load}}} \,, \tag{2.50}$$

where ω_{load} and ω_{red} represent the frequencies present in respectively the load and reduced model. The first estimate is often the most relevant, since most reduction bases start at low frequency dynamics, where additional basis vectors account account for higher frequency dynamics. If the reduction method is based on spectral shifting, the second error estimate will indicate whether the low-frequency dynamics are also captured. However, in this research, only the first one will be used.

For details on the calculation of ω_{load}, see Appendix D.

Dual Weighted Residual estimate

From the reduced results we can calculate the residual dynamic force:

$$\mathbf{M}\mathbf{V}\ddot{\mathbf{z}} + \mathbf{C}\mathbf{V}\dot{\mathbf{z}} + \mathbf{K}\mathbf{V}\mathbf{z} - \mathbf{f} = \hat{\mathbf{r}} \tag{2.51}$$

$$\hat{\mathbf{r}} = -(\mathbf{M}\ddot{\mathbf{r}} + \mathbf{C}\dot{\mathbf{r}} + \mathbf{K}\mathbf{r}) \,. \tag{2.52}$$

In model reduction we normally project this on $\mathbf{V}$ to remove it. However $\hat{\mathbf{r}}$ physically represents residual internal forces, which will decrease with higher model accuracy. While this has a clear physical meaning, it is quite difficult to calculate a relevant error estimate. Residual forces could largely cancel each other out and do not directly indicate response errors. An example of dealing with this is the so called *Dual Weighted Residual* method, which was studied extensively in the MSc. thesis of Nortier [34]. For harmonically excited system this method can be summarized by:

$$e = \mathbf{a}^{\mathrm{T}} \mathbf{r} = ((\mathbf{K} - \omega^2 \mathbf{M})^{-1})\mathbf{a})^{\mathrm{T}} \hat{\mathbf{r}} \,, \tag{2.53}$$

where ω is the loading frequency and $\mathbf{a}$ is a vector used to select degrees of freedom. Physically this error represents the displacement at the DoFs in $\mathbf{a}$ due to the residual force of a harmonic load. But the fact that the full system has to be solved makes it very expensive, although this is reduced because the factorized matrix is usually available from the creation of the basis vectors. Furthermore the selection of $\mathbf{a}$ is non-trivial.

Nortier suggests some ways to make this error estimate more efficient[34]. It can be rewritten to be a function of $\mathbf{M}^{-1}$ instead of $\mathbf{K}^{-1}$, which is usually cheaper, but still expensive. Alternatively, we can approximate $\mathbf{r}$ by refining the model further. This does introduce an additional error, and does not change the fact that the described method is only valid for harmonic response. A similar method for arbitrary dynamic response has not been found.

Refinement estimate

A much cheaper estimate was formulated, extending the research by Han [35]). It is most suited to load-based reduction (Ritz vectors & QSRV). The first load based vector captures the (quasi-)static displacement, the higher ones add a dynamic correction. So we can assume that the contribution and thus the kinetic energy associated with a vector will be lower for higher order vectors. This will be demonstrated in Chapter 5. Based on this the following error estimate is defined:

$$e = \frac{\int_0^T E_{\mathrm{kin,n}} \,\mathrm{d}t}{\int_0^T E_{\mathrm{kin,total}} \,\mathrm{d}t} = \frac{\int_0^T \dot{z}_n^2(t) \,\mathrm{d}t}{\sum_{j=1}^{n} \int_0^T \dot{z}_j^2(t) \,\mathrm{d}t} \,. \tag{2.54}$$

Because the reduced mass matrix is the identity matrix, $1/2\dot{\mathbf{z}}^{\mathrm{T}}\dot{\mathbf{z}}$ is the kinetic energy of the reduced coordinate. This gives this error estimate a clear physical interpretation: the kinetic energy of the highest basis vectors normalized by the total kinetic energy. If we ignore coupling between the basis vectors (insignificant at small amplitudes), this estimate is similar to refining the model by adding an extra vector.

The results for modal based reduction however have to be judged carefully, because the assumption that higher basis vectors lose significance will not hold for all cases.

2.5.6. Updating of basis vectors

In any type of structural optimization, the structure will change. A new structure corresponds to a new set of basis vectors. There are some techniques which allow us to compute these in a cheap way. In this section, this will be discussed this for Ritz vectors. This can also be applied to QSRV, and similar methods exist for eigenvectors [36].

The most expensive step of computing Ritz vector is the solution of $\mathbf{v}_{i+1} = \mathbf{K}^{-1}\mathbf{M}\mathbf{v}_i$. This is similar to finding the updated static displacement of a structure, which has been researched extensively in the field of structural optimization, see for example [37, 38]. The most commonly used method is the *pre-conditioned conjugate gradient* method. Another frequently used term is the *combined approximations* method, these methods produce identical results when we use the approach described below [39].

We start from the original stiffness equation:

$$\mathbf{K}_0\mathbf{q}_0 = \mathbf{f}. \tag{2.55}$$

This is solved directly, by Cholesky factorization. Now we assume $\mathbf{K}_0$ is updated according to:

$$\mathbf{K}_1 = \mathbf{K}_0 + \Delta\mathbf{K} \tag{2.56}$$

$$\mathbf{K}_1\mathbf{q}_1 = \mathbf{f}. \tag{2.57}$$

We want to find $\mathbf{q}_1$ while avoiding the expensive step of factorizing and solving the system. This can be done using the conjugate gradient algorithm [39]. This is an iterative algorithm for solving linear systems, which only requires a couple of matrix multiplications per iteration. The mathematical details will be omitted here. As an initial guess, we use $\mathbf{q}_0$. To ensure proper convergence, the condition number of $\mathbf{K}_1$ should be as low as possible. This can be improved by using *pre-conditioning*. We perform a change of variables:

$$\mathbf{q}_1 = \mathbf{C}\tilde{\mathbf{q}}_1 \tag{2.58}$$

$$\mathbf{C}^{\mathrm{T}}\mathbf{K}_1\mathbf{C}\tilde{\mathbf{q}}_1 = \mathbf{C}^{\mathrm{T}}\mathbf{f}. \tag{2.59}$$

The *pre-conditioner* $\mathbf{C}$ should be chosen to minimize the condition number of $\mathbf{K}_1$. The ideal pre-conditioner would transform $\mathbf{C}^{\mathrm{T}}\mathbf{K}_1\mathbf{C}$ into the identity matrix, this would mean $\mathbf{C}$ is the inverse of the Cholesky factor of $\mathbf{K}_1$. This is expensive, we want to avoid further factorizations. A good approximation is the factorization we already have, that of $\mathbf{K}_0$. The algorithm requires two solutions per iteration of a linear system with matrix $\mathbf{C}$.

To summarize, we can avoid repeated factorizations of matrix $\mathbf{K}$, at the cost of a larger number of solution of linear systems. The exact amount depends on the convergence. MATLAB contains an implementation of the preconditioned gradient method, the *pcg* function.

2.5.7. Non-linear systems

Model order reduction is more difficult for non-linear systems. It is however still possible, and can result in better performance compared to non-reduced systems. This is an ongoing topic of research with many interesting methods. However, for the sake of simplicity, this research is limited to linear systems.

2.6. COMBINING OPTIMIZATION AND MODEL ORDER REDUCTION

Finally in this section we will investigate the combination of the theory and methods discussed earlier. There is a lot of recent research, so this section will serve as a literature review. Later in this thesis the combination will be studied in more detail, see Sections 4.1 and 5.5. The current section is split in two parts: the combination of topology optimization and structural dynamics; and the combination of model order reduction and optimization. For both of these the most relevant research will be highlighted and discussed. A short overview of this can be found in Table 2.2.

Table 2.2: Comparison of relevant research. Abbreviations: Topology Optimization (TO), Shape Optimization (SO), Sensitivity Analysis (SA), Frequency-Domain (FD), Time-Domain (TD). On the combination of TO and MOR for time-domain problems, the only known research is by Zhao and Wang [40]. On the separate subjects, there is a lot of relevant literature.

Year	Author	Optimization	Dynamics	MOR	Comments
1970	Fox and Kapoor [41]	SO	TD&FD		
1991	Greene and Haftka [42]	SA	TD		
1999	Min *et al.* [43]	TO	TD		
2004	Pedersen [44]	TO	TD		
2005	Han *et al.* [45]	SO	TD	Yes	
2006	Kang *et al.* [12]	Review	TD		Review on transient optimization
2010	Yoon [17]	TO	FD	Yes	
2013	Han [35]	SA	TD	Yes	
2015	Zhao and Wang [40]	TO	TD	Yes	Uses MSM and MAM

2.6.1. TOPOLOGY OPTIMIZATION FOR STRUCTURAL DYNAMICS

The classical problem in topology optimization is that of static compliance minimization, however, structural dynamics problems have also been studied extensively. In this section we will discuss the most relevant research. The focus here is on dynamics in the time-domain, so any research on frequency-domain problems will not be discussed.

- B.S. Kang, G.J. Park, and J.S. Arora, *A review of optimization of structures subjected to transient loads*, Structural and Multidisciplinary Optimization 31, 81 (2006)[12].
  This is the most recent known review paper on the subject of optimization and transient loads. It discusses ways of treating time-dependent constraints, approximation methods and sensitivity analysis. In this last part, the use of model order reduction is mentioned for direct sensitivity analysis. The adjoint variable method for transient problems is described in great detail.

- R. L. Fox and M. P. Kapoor, *Structural optimization in the dynamics response regime - a computational approach*, AIAA Journal 8, 1798 (1970) [41].
  In this paper, the optimization of a truss-frame is discussed, for both frequency response and transient response. This is the earliest known example of transient response shape optimization.

- S.Min, N. Kikuchi, Y. C. Park, S. Kim, and S. Chang, *Optimal topology design of structures under dynamic loads*, Structural optimization 17, 208 (1999) [43].
  First example of topology optimization (using homogenization) in transient structural dynamics. Sensitivity analysis is performed using direct differentiation, and an optimality criteria scheme is used to find the optimal design.

- C. B. W. Pedersen, *Crashworthiness design of transient frame structures using topology optimization*, Computer Methods in Applied Mechanics and Engineering 193, 653 (2004) [44].
  One of the many papers on crashworthiness, but one of the few treating topology optimization. A ground structure consisting of non-linear beam elements is optimized to achieve a desired acceleration response. This is not directly relevant for this thesis. However an extension to non-linear systems would have many practical applications. In the field of crashworthiness there is already quite some related research, the interested reader is referred to the work of Pedersen.

Concluding this part, we can say that there are many examples of structural optimization applied to transient dynamics. There is a lot of literature on this, which is very helpful in implementation. There are no examples of industrial applications known to the author.

2.6.2. Model order reduction and optimization

In this section we highlight the most relevant research combining optimization and Model Order Reduction (MOR).

- Zhao and C. Wang, *Dynamic response topology optimization in the time domain using model reduction method*, Structural and Multidisciplinary Optimization , 1 (2015).
  This recent paper has a very close relation to this thesis, it will be discussed in more detail later in this section.

- J. S. Han, E. B. Rudnyi, and J. G. Korvink, *Efficient optimization of transient dynamic problems in MEMS devices using model order reduction*, Journal of Micromechanics & Microengineering 15, 822 (2005) [45].
  This paper describes the shape optimization (4 design variables) of a MEMS. The finite element analysis is performed in ANSYS, and the MOR uses Ritz vectors. The authors claim that the optimization is at least an order of magnitude faster than one using a full model. Sensitivity analysis is performed using global finite differences on the reduced model.

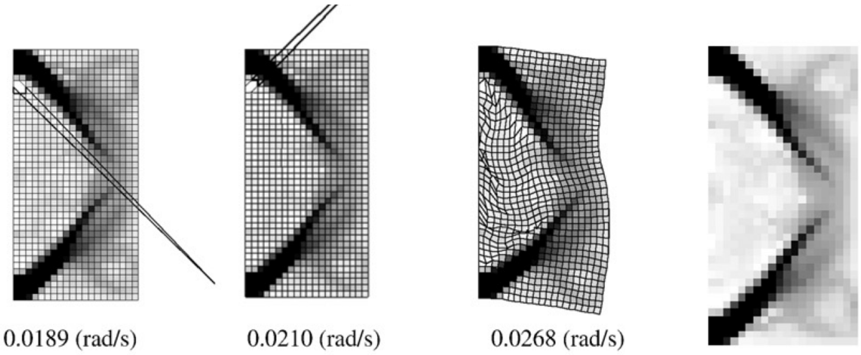
- G. H. Yoon, *Structural topology optimization for frequency response problem using model reduction schemes*, Computer Methods in Applied Mechanics and Engineering 199, 1744 (2010) [17].
  This is the first research combining Topology Optimization and MOR. The optimization concerns minimizing dynamic compliance (frequency domain). Different model order reduction schemes are applied: Modal Superposition, Ritz Vectors and Quasi-Static Ritz Vectors. The author shows that when using the MSM method local eigenmodes appear, disturbing the accuracy of the reduced model. This causes incorrect optimized results (see Fig. 2.5), different ways to solve this are discussed.

  Furthermore the accuracy of the different reduction methods is compared. Sensitivity analysis is performed using an adjoint variable method. As was discussed in Section 2.4.4, this is more expensive for time-domain problems.

  A few minor improvements on this research can be found in the recent work of Liu *et al.* [46]. They extend the work of Yoon to large-scale 3D problems, with multiple load frequencies.

- J. S. Han, *Calculation of design sensitivity for large-size transient dynamic problems using Krylov subspace-based model order reduction*, Journal of Mechanical Science and Technology 27, 2789 (2013) [35].
  In this research MOR (Ritz vectors) is applied to transient sensitivity analysis using direct differentiation. The author shows that this can significantly decrease the computational effort, while maintaining a reasonable accuracy.

- W. H. Greene and R. Haftka, *Computational aspects of sensitivity calculations in linear transient structural analysis*, Structural optimization 3, 176 (1991)[42] This paper contains a detailed description of transient simulation and sensitivity analysis, including model order reduction. Sensitivities are calculated using direct differentiation with modal superposition, where the effect of eigenvector sensitivities is studied. Neglecting eigenvector sensitivities is shown to lead to a significant error. As an alternative to this, the sensitivities are calculated using the modal acceleration method, which is also shown to result in a high accuracy.



(a) First three eigenmodes of optimized structure.

(b) Optimized structure using MSM.

Figure 2.5: Selected results from Yoon [17], showing a optimized structure and its first three eigenmodes. The structure has a suboptimal shape, caused by the poor accuracy of the MSM model. This is partially caused by local modes, the first two eigenmodes do not contain any useful information about the response of the structure.

At the start of this project, to the extent of my knowledge there was no published research combining topology optimization, model order reduction and time domain structural dynamics. However, near the end of this project (August 2015), a paper was published by Zhao and Wang [40]. This describes the use of MOR to reduce the cost of topology optimization for dynamic response in the time domain. We will highlight some of the similarities and differences between this thesis and the paper describing Zhao & Wang's research.

- Both can be seen as an extension to the work of Yoon [17]. Yoon treated optimization of frequency domain problems, which is extended to time domain problems.

- Both use the adjoint variable method for sensitivity analysis.

- Zhao & Wang apply model order reduction, using modal superposition and modal acceleration. Modal superposition is shown to produce suboptimal designs, which they relate to its low accuracy. Modal acceleration does result in optimal designs, and in some cases results in a 50% reduction in computation time.

- Other reduction methods are not used by Zhao & Wang, while Yoon showed that Ritz vectors and Quasi-Static Ritz Vectors lead to more accurate results in most cases. This thesis investigates this in more detail.

- Zhao & Wang study the effect of changing the time-scale of the problem. Other parameters, most notably the number of vectors in the reduced basis, are not investigated fully.

So while the work of Zhao & Wang is in some ways similar to this thesis, there are enough open questions that require further investigation.

Both Yoon and Zhao & Wang show that an inaccurate reduction leads to both slow convergence and suboptimal designs. This is a critical point for a successful application, so investigating the accuracy of the model reduction will be a large part of this thesis.

2.7. Summary & Discussion

This chapter has given an overview of relevant theory and research in the fields of topology optimization, transient analysis and model order reduction. Based on this the following methods will be used in the remainder of this research:

- Topology optimization using element densities as design variables.
- Transient analysis using Newmark integration.
- Sensitivity analysis using the adjoint variable method.
- Model order reduction using the following methods:
  - Modal superposition method
  - Ritz vectors
  - Quasi-static Ritz vectors
  - Modal acceleration method

In general topology optimization for time-domain dynamics is much more expensive than for similar static or frequency-domain problems. Model order reduction is expected to reduce the costs significantly. The expected reduction in costs is approximately equal to the number of time steps divided by the number of basis vectors. Furthermore, methods of estimating the accuracy of the reduced model have been proposed. These can be used to estimate the correct amount of basis vectors at a low cost.

There is some research combining model order reduction and topology optimization. The most relevant example is by Zhao and Wang [40], which features time-domain dynamics. There are still many open questions. This thesis will extend their work by studying the effect of different MOR methods and their accuracy. In the next two chapters the exact implementation will be discussed.

3

Implementation: Model & Optimization

Based on the theory discussed in the previous chapter a topology optimization code using model order reduction was created. The important aspects of the implementation will be described in the following two chapters. This first chapter focuses on everything but the model order reduction, starting with a section containing an overview of the full process, and all the errors involved. The next section discusses the implementation of the model and sensitivity. The remaining section treats the optimization details.

3.1. Overview

3.1.1. Process and error overview

The implemented optimization of a time-domain problem can be summarized in a few steps, see Fig. 3.1. Starting from an initial design, a finite element model is created. Using numerical integration of the equations of motion the dynamic response is found. From this the objective function is calculated, which is combined with the response information to numerically integrate the adjoint equation. This data is passed on to the sensitivity analysis, and finally a design update is calculated by the optimizer. This is repeated until convergence.

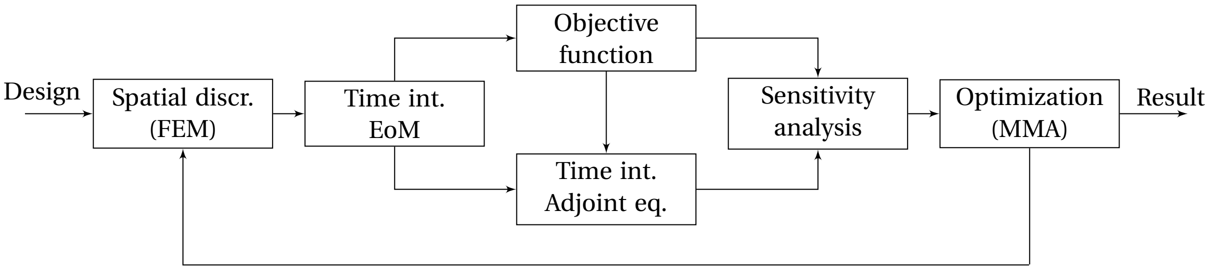


Figure 3.1: Overview of steps in optimization of a structure for transient dynamics.

Using these steps a systematic overview can be made of all the errors involved:

- *Spatial discretization*: in modelling a physical structure with a finite element model, there is always an error. This depends on the used elements and mesh. In general, using a finer mesh and higher-order elements reduces this error[22]. In this thesis, this will not be investigated, for research into optimization and spatial discretization see [47].

- *Time integration of equations of motion*: this introduces three errors.

  - Temporal discretization: similar to spatial discretization, moving from a continuous to a discrete formulation will lead to an error.
  - Newmark integration: the discretized values of the response in time are found using numerical integration, which also leads to an error. The trapezoidal Newmark scheme guarantees conserva-

tion of energy, although there will be an error in the period of simulated oscillation. This error is determined by the frequency content of the response [10, 21][1].

   - Model order reduction: discussed in Section 3.1.2.

- *Objective function*: this is usually an integral over time, which is approximated numerically, again leading to an error.
- *Time integration of adjoint equation*: this is subject to the same errors as the time integration of the equation of motion. Furthermore the adjoint equation is a function of the motion of the structure, which means any error in the previous time integration will influence this term.
- *Sensitivity analysis*: In this step all the information of the previous steps is combined. This introduces a new error in the evaluation of the integral in Eq. (2.28). Furthermore all previous errors influence the sensitivity, so the result might be significantly different from the physical sensitivity.
- *Optimization:* the algorithm (MMA) approximates the design space to find a new design. This approximation itself can contain a significant error.
- *Round-off error:* finally, whenever dealing with floating point variables, there is numerical round-off error. This is in most cases not significant. It will be when subtracting large, nearly identical numbers, most notably in the Gramm-Schmidt orthogonalization used for computing basis vectors.

For sensitivity analysis the distinction can be made between *accuracy* and *consistency*. Accuracy refers to the difference between the computed sensitivity and the exact sensitivity of the physical response. Consistency refers to the difference between the computed sensitivity and the modelled response, where errors in the model are not taken into account. This is illustrated by Fig. 3.2.

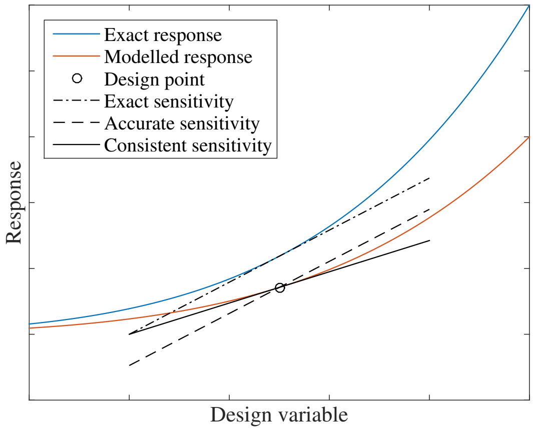


Figure 3.2: Illustration of the difference between consistent and accurate sensitivities. The exact physical and modelled response are shown, there is a modelling error. At a certain design point, three sensitivities are drawn. The first is the exact sensitivity of the exact solution model, the second is an accurate sensitivity, which is identical to the first, but not representative of the model. The third is a consistent sensitivity, which is representative of the model, but there is an error with respect to the exact solution.

The relevance of these two quantities is dependent on the application. The goal of this research is to speed up optimization using model order reduction. The accuracy of the used model is thus of low importance. The consistency of the sensitivities and the effect of MOR on this will affect the convergence of the optimization, so this is relevant. Therefore, two types of errors defined so far will be not be investigated. The first is the spatial discretization error. While this is important for successful application of optimization, it can be ignored in this research. The second are errors that are not affected by model order reduction, such as the approximation error in MMA. These determine the rate of convergence of the optimization, but there is no direct link to model order reduction.

The other listed errors may be affected by model order reduction. This will discussed next.

[1] For an undamped oscillation with exact frequency ω, time step h and period t_e, the trapezoidal scheme results in the period $t_e\left(1+\frac{\omega^2 h^2}{24}\right)$.

3.1.2. EFFECT OF MOR ON OTHER ERRORS

Model order reduction by itself introduces two errors. The first is a residual in the dynamic response, which is neglected, causing an error in both time integration steps. The second is in basis vector sensitivities, this will be treated in Section 4.2. Both errors can be significant, and will be examined in detail.

The question is whether the errors already present in temporal discretization and Newmark integration are changed by model order reduction. An accurate reduced model will have the same frequency content, and thus the same error. Therefore the assumption is made that when accurate model order reduction is used, these two errors will not change. This will be tested in Chapter 5.

The other steps all use data from the time integration, so these will be dependent on the errors resulting from the model order reduction. However, unlike the Newmark integration, none of these steps are directly affected by the model order reduction, just by the possible residual. In other words, for a reduced order model with zero residual, the objective function, optimization and sensitivity analysis (basis vector sensitivities aside) will be the same as for a full model.

Therefore, in this thesis the main goal is investigating effect of the residual errors of the reduced model, and the effect of basis vector sensitivities. The consistency of the resulting design sensitivities is the main point of interest. A secondary goal is to see whether the errors made in temporal discretization and Newmark integration are affected by model order reduction. All other errors might be important for successful optimization, but they are less relevant for this thesis.

3.2. IMPLEMENTATION OF MODEL

This section highlights some choices that have been made in creating the model used in the optimization, which calculates the dynamic response of a design. Some more details can be found in Appendix D, which also contains a list of physical parameters.

3.2.1. FINITE ELEMENT ANALYSIS

The Finite Element Method (FEM) is a method of discretizing partial differential equations. Readers are expected to be familiar with it, for reference see for example [9, 22]. In this research two types of elements have been implemented:

- *Q4 elements*: a two-dimensional quadrilateral element with 4 nodes. Allows displacements in both directions, with a bi-linear displacement field defined by two displacements per node, 8 in total. Simple and easy to implement, but poor accuracy, partially due to shear locking. Meshing is straightforward, by just filling the domain with identical non-overlapping elements, see for example Fig. 2.1b. As design variable, the density of the material is used, normalized to be between zero and one.
- *Beam elements*: a two dimensional element with 2 nodes. This is an Euler-Bernoulli beam combined with a bar, which allows two displacements and one rotation per node, six degrees of freedom in total. The displacement variable is the width of the (square) cross-section. A ground structure is defined using a grid of nodes and various beams connecting them.

While both elements are implemented in the code, all investigations will be done using Q4 elements. This choice was made mainly because most research on topology optimization uses Q4 elements, which allows better comparison to literature. Secondly, using Q4 elements allows easier finite element analysis (identical element matrices for every element) and volume constraints (linear, identical for every element). An argument for beam elements would be that using them avoids checkerboarding and allows better manufacturability. But one still has to deal with mesh-dependency and unwanted small elements, complicating this. Techniques like filtering and penalization could be applied to beam elements, but again this is simpler for Q4 elements.

3.2.2. DAMPING

Damping is often ignored in structural dynamics. However, in transient problems, it makes sense to include it. The most commonly used model is that of Rayleigh damping, where the damping matrix is given by the following equation:

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}, \tag{3.1}$$

where α and β are coefficients that determine the damping. The main advantage of Rayleigh damping is the fact that when using modal superposition, the damping matrix will be diagonal similar to the mass and stiffness matrices. Thus Rayleigh damping does not introduce any coupling between the modes.

3.2.3. DIRECT TIME INTEGRATION VERSUS ANALYTICAL METHODS

In Section 2.3 it was determined that the full equations of motion are best solved used Newmark integration. This may not be the case for the reduced equations of motion. If these are transformed to a diagonal form, the most efficient method of time integration uses Duhamel's integral [9]. Still, for this research, Newmark integration will be used for every time integration. There a few reasons for this:

- The reduced analysis using Newmark integration will be more similar to the full analysis, enabling more relevant comparison.
- Implementing Newmark integration is less complex, while the computational effort is not significant in the full process.
- The fact that the reduced matrices can be dense has some benefits. Obviously, the step of diagonalization can be omitted. But dense matrices also allow the use of inaccurate basis vectors, which might be cheaper to compute than the exact ones.

For these reasons, early in this research it was decided to restrict the time integration to the Newmark method. For many cases using Duhamel's integral might result in slightly faster computations.

3.2.4. TIME INTEGRATION

The finite element analysis yields the equations of motion, as well as $\mathbf{K}', \mathbf{M}'$ and $\mathbf{C}'$ and the adjoint equation. As discussed in Section 2.3, the Newmark algorithm was chosen solve these. This is straight-forward to implement, except for the backwards time integration. This requires a change of variables which will be explained next.

The sensitivity analysis requires the solution $\boldsymbol{\lambda}(t)$ corresponding to the terminal value problem described by Eq. (2.25) to (2.27). This can be converted to a standard initial value problem by using a change of variables:

$$\tau = T - t \tag{3.2}$$

$$\boldsymbol{\lambda}(\tau) = \boldsymbol{\lambda}(T-t) \tag{3.3}$$

$$\mathbf{M}\frac{\mathrm{d}^2\boldsymbol{\lambda}}{\mathrm{d}\tau^2}(\tau) + \mathbf{C}\frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}\tau}(\tau) + \mathbf{K}\boldsymbol{\lambda}(\tau) = -\frac{\partial\Phi}{\partial\mathbf{q}(\tau)}^{\mathrm{T}} \tag{3.4}$$

$$\boldsymbol{\lambda}(\tau=0) = \mathbf{0} \tag{3.5}$$

$$\mathbf{M}\frac{\mathrm{d}\boldsymbol{\lambda}(\tau=0)}{\mathrm{d}\tau} = \left(\frac{\partial\phi}{\partial\mathbf{q}(\tau=0)}\right)^{\mathrm{T}}. \tag{3.6}$$

This equation can be numerically integrated using the exact same algorithm and factorized matrices used for the equations of motion.

3.2.5. SENSITIVITY ANALYSIS

After the equations of motion and adjoint equation are solved, the design sensitivity can be evaluated using Eq. (2.28):

$$\begin{gathered}\frac{\mathrm{d}\Psi_{\mathrm{A}}}{\mathrm{d}\mathbf{x}} = \frac{\partial\phi}{\partial\mathbf{x}} + \int_0^T \frac{\partial\Phi}{\partial\mathbf{x}} - \mathbf{R}\,\mathrm{d}t \qquad . \\ \mathbf{R} = \boldsymbol{\lambda}^{\mathrm{T}}\left(\mathbf{f}' - \mathbf{M}'\ddot{\mathbf{q}} - \mathbf{C}'\dot{\mathbf{q}} - \mathbf{K}'\mathbf{q}\right)\end{gathered} \tag{3.7}$$

This equation is exact. The solutions for $\mathbf{q}$ and $\boldsymbol{\lambda}$ however are not. This approach was based on first differentiating the equations, after which they are discretized in time. From literature it is know that this leads to a consistency error [48]. Both $\boldsymbol{\lambda}$ and the pseudo-load will generally be periodic functions, for which Newmark integration will lead to a frequency error. This has a different influence on the two terms, because $\mathbf{q}$ is calculated starting from $t = 0$, and $\boldsymbol{\lambda}$ is calculated starting from $t = T$. Thus direct time integration will cause a phase error between the two terms, leading to a consistency error in the sensitivity.

An alternative, more consistent method is based on the reverse approach: the equations are discretized first, after which they are differentiated[2]. This leads to a different set of discretized equations, for details see [48]. The approach described by Jensen *et al.* [48] was implemented and tested. The results were similar to those found in literature, for details see Appendix D.3. Using the alternative sensitivity analysis did not result in faster convergence of the optimization, therefore in the remainder of this research the approach given by Eq. (3.7) is used.

3.3. OPTIMIZATION

When the response and sensitivities are known, a design update can be applied to the system. This section will first discuss different objective functions and constraints. The final part briefly treats filtering.

The optimization problems discussed in this document can all be described by:

$$\begin{aligned} &\min_{\mathbf{x}} && \int_0^T \Phi(\mathbf{q},\mathbf{x})\,\mathrm{d}t \\ &\text{subject to:} && \mathbf{M}\ddot{\mathbf{q}}(t)+\mathbf{C}\dot{\mathbf{q}}(t)+\mathbf{K}\mathbf{q}(t)=\mathbf{f}(t) \\ & && g(\mathbf{x})=\sum_{i=1}^{n_{\text{el}}} x_i/n_{\text{el}}-x_{\max}\leq 0 \\ & && 1\times 10^{-3}\leq x_i\leq 1, \qquad i=1,\dots,n_{\text{el}}. \end{aligned} \tag{3.8}$$

A non-zero minimum value for x_i is used to avoid singular matrices.

3.3.1. OBJECTIVE FUNCTIONS

Throughout this thesis the following general objective function was used:

$$\Psi=\phi(\mathbf{q}(T),\mathbf{x})+\int_0^T \Phi(\mathbf{q},\mathbf{x})\,\mathrm{d}t,. \tag{3.9}$$

where ϕ and Φ are general functions evaluated at respectively the end of the simulation and during the time span. We will now define some more specific objective functions, and calculate the analytical derivatives that are needed for the sensitivity analysis.

All functions discussed here use a fixed simulation time, and are only dependent on the displacement of the system, not on the velocities or accelerations. As discussed earlier, this could be changed, making the mathematics slightly more complex. Furthermore, one final requirement is imposed on every objective function. In the sensitivity analysis, the term $-\frac{\partial\Phi}{\partial\mathbf{q}}$ will appear as a force in the adjoint equation. For most reduction methods (MAM, Ritz vectors, QSRV) we need a constant spatial description of the load, which may be scaled by a time-dependent scalar. So $-\frac{\partial\Phi}{\partial\mathbf{q}}$ should be of the form $\mathbf{L}_\text{s} f(\mathbf{q},t)$.

All the described objective functions satisfy this requirement. Furthermore, if the load vector in the equations of motion is parallel to the load vector in the adjoint equation, model order reduction will result in the same basis for each. This is true for all the described objective functions, unless the force is applied to a point and the response measured in a different point. In that case, a new basis is calculated for the adjoint equation.

All of the described objective functions require the numerical evaluation of integrals. These can be approximated using a Riemann sum:

$$\int_0^T f(t)\,\mathrm{d}t\approx\sum_{i=1}^{n_t} f(t_i)\cdot h. \tag{3.10}$$

According to [48], this yields a significant error (especially in the sensitivity analysis), which can be reduced by a few orders of magnitude by using trapezoidal integration:

$$\int_0^T f(t)\,\mathrm{d}t\approx\frac{T}{2n_t-2}\sum_{i=1}^{n_t-1} f(t_i)+f(t_{i+1}). \tag{3.11}$$

[2] Similar methods exist for spatial discretization and differentiation. In this research, the spatial domain is discretized, then differentiated. The resulting sensitivities are known as *discrete derivatives*. The reverse approach, where the spatial domain is first differentiated, then discretized, results in *continuum derivatives*[13].

In MATLAB, both methods are easily implemented. There is a built-in function for trapezoidal integration (*trapz*), but this is approximately 50-100 times slower than simply summing. For this reason a new function for this was created, (*fasttrapz*), which has costs similar to summing, but less functionality.

Reference curve: absolute difference

The first objective function defined here aims to minimize the difference between the response at DoF i and a reference curve $q_{\text{ref}}(t)$.

$$\Psi = \int_0^T \left| q_i(t) - q_{\text{ref}}(t) \right| \mathrm{d}t. \tag{3.12}$$

This yields the following discontinuous derivatives:

$$\begin{aligned} \frac{\partial \phi}{\partial \mathbf{q}} &= \mathbf{0} \\ \frac{\partial \Phi}{\partial q_j} &= 0 \qquad j \neq i \\ \frac{\partial \Phi}{\partial q_i} &= \begin{cases} 1 & \text{if } q_i(t) - q_{\text{ref}}(t) > 0 \\ -1 & \text{if } q_i(t) - q_{\text{ref}}(t) < 0 \\ 0 & \text{if } q_i(t) - q_{\text{ref}}(t) = 0. \end{cases} \end{aligned} \tag{3.13}$$

The last case is mathematically not fully correct, the derivative of an absolute function does not exist at zero.

Reference curve: squared difference

The second objective defined here is similar to the first, instead of taken the absolute difference we now take the squared:

$$\Psi = \int_0^T \left(q_i(t) - q_{\text{ref}}(t) \right)^2 \mathrm{d}t. \tag{3.14}$$

The derivatives are continuous in this case, but they do depend on $\mathbf{q}_i$:

$$\begin{aligned} \frac{\partial \phi}{\partial q_i} &= 0 \\ \frac{\partial \Phi}{\partial q_i} &= 2(q_i(t) - q_{\text{ref}}(t)) \\ \frac{\partial \Phi}{\partial q_j} &= 0 \qquad j \neq i. \end{aligned} \tag{3.15}$$

Minimum dynamic compliance

This is an extension of standard minimum compliance problems, minimizing the compliance over time:

$$\Psi = \int_0^T \mathbf{f}^{\mathrm{T}}(t)\mathbf{q}(t)\,\mathrm{d}t \cdot \frac{1}{T}. \tag{3.16}$$

The associated analytical derivatives are:

$$\begin{aligned} \frac{\partial \phi}{\partial \mathbf{q}} &= \mathbf{0} \\ \frac{\partial \Phi}{\partial \mathbf{q}} &= \mathbf{f}^{\mathrm{T}}(t) \cdot \frac{1}{T}. \end{aligned} \tag{3.17}$$

This objective function has some useful properties. When studying minimum compliance in a statics problem, the adjoint variable turns out to be the static displacement, i.e. a term that is already available. Solution of the adjoint equation is not necessary in this case.

The minimum dynamic compliance problem shows a similar property. Because $\frac{\partial \Phi}{\partial \mathbf{q}}$ is just $\mathbf{f}$ with a scaling term, the solution of the adjoint equation can in some cases be found by simply multiplying $\mathbf{q}$ with $1/T$. An example is when $\mathbf{f}(t) = \mathbf{f}(T - t)$, with $\mathbf{q}(t = 0) = \dot{\mathbf{q}}(t = 0) = \mathbf{0}$. This results in

$\boldsymbol{\lambda}(t) = -\mathbf{q}(T-t) * 1/T$. This would mean time integration is not required for solving the adjoint equation.

Although this could yield significant computational benefits, this property is intentionally *not used* in this thesis, because it is only applicable to this objective function. For the sake of generality, all objective functions are treated in the same way, with forward and backward time integration.

OTHER OBJECTIVES

There are many other possibilities for objective functions, none of these are implemented. One notable option is minimizing the strain energy: $\Phi = 1/2\mathbf{q}^{\mathrm{T}}\mathbf{Kq}$. This cannot be implemented, because its derivative with respect to $\mathbf{q}$ is $\mathbf{Kq}$, which cannot be written as a constant spatial term times a temporal term, a necessary condition for all reduction methods that use load information (MAM, Ritz vectors).

3.3.2. CONSTRAINTS

Almost every topology optimization code includes a volume constraint. This makes sure the total volume remains below a fraction $x_{\max}$:

$$g(\mathbf{x}) = \frac{1}{n_{\mathrm{el}}}\sum_{i=1}^{n_{\mathrm{el}}} x_i - x_{\max} \leq 0 \tag{3.18}$$

$$\frac{\mathrm{d}g}{\mathrm{d}\mathbf{x}} = \mathbf{1}/n_{\mathrm{el}}\,. \tag{3.19}$$

Many other constraints are possible, but none are currently implemented. Of special interest are constraints that are a function of $\mathbf{q}$ and its derivatives in the time domain. An example would be a constraint on the maximal displacement of the structure for the entire time domain. These can be implemented in many different ways, for details see [12].

3.3.3. FILTERING

In the created optimization code a density filter is included. For reference a sensitivity filter is implemented as well, since this is used by Zhao and Wang [40]. However the density filter is preferred, since this does not compromise the consistency of the sensitivities.

Both filters are based on the code of Andreassen *et al.* [49]. The sensitivity filter modifies the sensitivities using the following equation:

$$\begin{aligned}\frac{\mathrm{d}\Psi}{\mathrm{d}x_e} &= \frac{1}{\sum_{i=1}^{n_{\mathrm{el}}} H_{ei}}\sum_{i=1}^{n_{\mathrm{el}}} H_{ei}x_i\frac{\mathrm{d}\Psi}{\mathrm{d}x_i}\\ H_{ei} &= \max(0, r_{\min} - \Delta(e,i))\,,\end{aligned} \tag{3.20}$$

where $r_{\min}$ is the filter radius, and $\Delta(e,i)$ is the distance (centre to centre) between elements e and i. The density filter (illustrated by Fig. 3.3) is given by:

$$x_e = \frac{\sum_{i=1}^{n_{\mathrm{el}}} H_{ei}x_i}{\sum_{i=1}^{n_{\mathrm{el}}} H_{ei}}\,, \tag{3.21}$$

which means the objective and constraint sensitivities also need to be updated according to:

$$\frac{\mathrm{d}\Psi}{\mathrm{d}x_e} = \sum_{e=1}^{n_{\mathrm{el}}} \frac{H_{ei}}{\sum_{i=1}^{n_{\mathrm{el}}} H_{ei}}\frac{\mathrm{d}\Psi}{\mathrm{d}x_i}, \qquad \frac{\mathrm{d}g}{\mathrm{d}x_e} = \sum_{e=1}^{n_{\mathrm{el}}} \frac{H_{ei}}{\sum_{i=1}^{n_{\mathrm{el}}} H_{ei}}\frac{\mathrm{d}g}{\mathrm{d}x_i}\,. \tag{3.22}$$

The computational effort in both cases is minimal. Matrix $\mathbf{H}$ is only dependent on the mesh, and thus has to be computed just once. The rest of the steps involve some matrix-vector multiplications.

3.4. NOTES ON SOFTWARE IMPLEMENTATION

For this project MATLAB was used. This was chosen mainly because of its extensive set of built-in functions and the author's familiarity with it. It is not the fastest programming language, but with good programming practice it is sufficient.

The majority of the code was written by the author, details can be found in Appendix D.

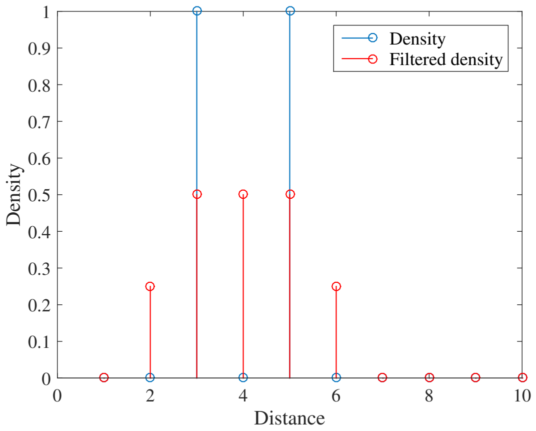


Figure 3.3: Illustration of a density filter with a radius of 2 on a one-dimensional problem. Starting point is a set of design variables, with a checkerboard: at location 3 and 5, there is material present, at location 4 there is not. After applying the filter, the checkerboard is gone, while the total volume is preserved. This does lead to intermediate densities, most these will be removed in the optimization process.

3.4.1. Measuring performance

The performance of an algorithm is difficult to measure. Simply running two algorithms on the same machine and measuring the CPU-time is not always reliable, since it not only depends on the number of floating point operations (FLOPS) involved, but also on the way the computer architecture is used. This is dependent on the chosen programming language, and even more so on the exact implementation. Using various performance optimization techniques can decrease CPU-time dramatically, this will be discussed in Section 3.4.2. Furthermore, some MATLAB algorithms are highly optimized, such as the eigenvector algorithm, this complicates the performance measurement even further.

For the best comparison between algorithms both should be implemented and optimized in a lower-level language such as Fortran. In this project this is not a feasible approach. To ensure a accurate comparison using only MATLAB, a few steps have been taken. First of all, whenever computation times are compared, the methods use as much shared code as possible. Secondly, every critical algorithm is optimized to the best ability of the author. Finally, for simulations where the CPU-time is measured, the average of 10 runs is taken, for example in Section 5.3.2.

3.4.2. Optimizing performance

Vectorization

Vectorization, also known as array programming, refers to performing operations on entire arrays, instead of on an element-by-element basis. Vectorization enables the avoiding of loops, which are notoriously slow in MATLAB[50]. This is not possible when a loop requires the solution of the previous iteration, for example in the case of the Newmark algorithm and the Arnoldi iterations. In all other critical algorithms vectorization is used.

Factorization & Reordering

Direct solving of linear systems always consists of factorizing the matrix into multiple triangular or diagonal matrices, after which the solution can be found using substitution. These steps can be made more efficient by *reordering*, which optimizes the sparsity of the factors.

The linear solver built into MATLAB internally uses factorization as well as reordering. However when a linear system is solved repeatedly for different right-hand sides, it is inefficient to factorize the matrix each time. In this case some computational effort can be saved by manually reordering and factorizing the equations once. For reordering, the implementation uses the *symmetric approximate minimum degree* (MATLAB: *symamd)*[51] algorithm. For factorization, the Cholesky algorithm is the most efficient, but this is restricted to positive definite matrices. For other symmetric matrices, the LDL algorithm is used.

Sparse matrix implementation

Sparse matrices are matrices with a large number of zeros. This can be used to improve performance in two ways. First of all, using an alternative storage scheme that does not store zeros can reduce memory requirements. Secondly, the zeros in the matrices can often be ignored in calculations.

MATLAB uses *Compressed Sparse Column (CSC)* storage[52]. This uses three vectors: one containing all entries, a second containing the corresponding row indices, and finally a third indicating the first element in each column. This is efficient for most sparse matrices.

There is one notable exception in the scope of this project. Derivatives of system matrices ($\mathbf{K}', \mathbf{C}', \mathbf{M}'$ are extremely sparse, for Q4 elements they only contain 64 elements. Using the CSC format is not efficient in this case, a 10,000x10,0000 matrix with 64 entries uses 81032 bytes, while a 64 entry full matrix requires only 512 bytes. This issue can be avoided by computing the matrix derivatives per element, and storing them as a full matrix with a list of the corresponding DoFs.

3.5. Summary & Discussion

There are many important aspects to the implementation of topology optimization for transient dynamics, even without model order reduction. In this chapter the most relevant points have been discussed.

In the optimization process, there are many steps that result in errors. Within the scope of this research, some of these will be ignored, such as spatial discretization. Others, most notably the errors involved with model order reduction, will be investigated extensively. Multiple objective functions to be used in the optimization have been described, as well as a constraint and filtering methods. Finally some notes on the software implementation have been made, including on how to optimize and measure the performance of algorithms.

Using the methods described up to this point, a topology optimization code for time-domain problems was developed in MATLAB. In the next chapter, the effect of model order reduction will be discussed.

4

Implementation: Model Order Reduction

4.1. Model order reduction

In Section 2.5 different methods of model order reduction were discussed. The following have been implemented:

- MSM: Modal Superposition Method
- Ritz: Load Dependent Ritz Vectors
- MAM: Modal Superposition Method + Modal Acceleration

Quasi-static Ritz vectors (QSRV) have also been implemented, but these were found to be effective only for high-frequency problems. At the same time, they are numerically less stable than Ritz vectors, this is caused by the poorer conditioning of the iteration matrix. Therefore only the latter have been used in this research.

4.1.1. MOR and sensitivity analysis

In this section we will start by analytically investigating the effect of combining MOR and optimization. As stated in the previous chapter, the most computationally expensive part of the model is solving the equations of motion and the adjoint equation, which are given by:

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{C}\dot{\mathbf{q}}(t) + \mathbf{K}\mathbf{q}(t) &= \mathbf{f}(t) \\ \mathbf{M}\ddot{\boldsymbol{\lambda}} - \mathbf{C}^{\mathrm{T}}\dot{\boldsymbol{\lambda}} + \mathbf{K}\boldsymbol{\lambda} &= -\frac{\partial \Phi}{\partial \mathbf{q}}^{\mathrm{T}}. \end{aligned} \tag{4.1}$$

Solving both equations allows to compute the objective function and its sensitivities. This section will discuss two methods of solving these equations using model order reduction. Note that the alternative method of sensitivity analysis (Section 3.2.5 and Appendix D.3) will not be discussed here.

4.1.2. Limited reduction

This method, used in the research of Zhao and Wang[40], restricts the MOR to just Eq. (4.1). Each of these is reduced and solved, after which the nodal coordinates are calculated using $\mathbf{q} \approx \mathbf{V}\mathbf{z}$ and $\boldsymbol{\lambda} \approx \mathbf{V}\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is the reduced coordinate for the adjoint equation[1]. One can also use different bases for each equation, which is useful if the forcing terms are dissimilar.

This approach is easy to implement and understand: only the time integration is changed, the rest of the optimization runs as if there is no model order reduction. This results in an error in the objective function, as well as an error in the design sensitivity, which is caused by four terms:

1. Error in $\mathbf{q}$, given by $\mathbf{r} = \mathbf{q}_{\mathrm{full}} - \mathbf{V}\mathbf{z}$.
2. Error in $\boldsymbol{\lambda}$, given by $\mathbf{r}_{\lambda} = \boldsymbol{\lambda}_{\mathrm{full}} - \mathbf{V}\boldsymbol{\theta}$.

[1] Note that the full nodal solutions $\mathbf{q}$ and $\boldsymbol{\lambda}$ are not necessarily calculated. This step would require a large amount of memory, it is more efficient to use $\mathbf{V}\mathbf{z}$ and $\mathbf{V}\boldsymbol{\theta}$ whenever either value is required in the implementation. Exception to this is the modal acceleration method, the quasi-static correction cannot be expressed in reduced coordinates.

3. Error in $\frac{\partial \Phi}{\partial \mathbf{q}}^{\mathrm{T}}$, caused by the error in $\mathbf{q}$. This also results in an error in $\boldsymbol{\lambda}$.
4. Error in adjoint formulation Ψ_A, the equations of motion are no longer exactly solved, thus the term added to the equation of motion might not be zero. The residual is equal to $\hat{\mathbf{r}}$, the force residual. This results in:

$$\Psi_{\mathrm{A}} = \Psi + \int_0^T \boldsymbol{\lambda}^{\mathrm{T}}(t) \left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} - \mathbf{f}\right) \mathrm{d}t = \Psi + \int_0^T \boldsymbol{\lambda}^{\mathrm{T}}(t) \hat{\mathbf{r}} \, \mathrm{d}t, \tag{4.2}$$

where we have used $\hat{\mathbf{r}} = \mathbf{MV}\ddot{\mathbf{z}} + \mathbf{CV}\dot{\mathbf{z}} + \mathbf{KVz} - \mathbf{f}$. Note that this does not affect the objective function, because $\mathbf{V}\boldsymbol{\theta}$ is substituted for $\boldsymbol{\lambda}$, and $\mathbf{V}^{\mathrm{T}}\hat{\mathbf{r}} = 0$. However, in the derivative there will be an extra term:

$$\left[\int_0^T \boldsymbol{\lambda}^{\mathrm{T}}(t) \hat{\mathbf{r}} \, \mathrm{d}t\right]' = \int_0^T (\mathbf{V}'\boldsymbol{\theta})^{\mathrm{T}}\hat{\mathbf{r}} + \underbrace{(\mathbf{V}\boldsymbol{\theta}')^{\mathrm{T}}\hat{\mathbf{r}}}_{=0} + (\mathbf{V}\boldsymbol{\theta})^{\mathrm{T}}\hat{\mathbf{r}}' \, \mathrm{d}t. \tag{4.3}$$

The second term is zero, the third is the usual adjoint term. However, the first term is not zero. Because $\hat{\mathbf{r}}$ will be small for accurate MOR, and $\mathbf{V}'$ is expensive to calculate, this term is neglected, causing a fourth error in the sensitivity.

All these errors go to zero when the reduced solution converges towards the exact one. So when the MOR is very accurate, this method will perform similar to full analysis, but possibly with lower computational costs.

4.1.3. Detailed analysis

To gain some more insight a detailed analysis will be done next. This starts by repeating the derivation of the adjoint variable method, this time with MOR.

Without MOR, the design sensitivity is given by:

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}} \Psi(\mathbf{q}, \mathbf{x}) = \frac{\partial \Psi}{\partial \mathbf{x}} + \frac{\partial \Psi}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \mathbf{x}}, \tag{4.4}$$

where the term $\frac{\partial \mathbf{q}}{\partial \mathbf{x}}$ is expensive to evaluate, and thus replaced using an adjoint formulation. But when model order reduction is used, this changes to:

$$\frac{d}{d\mathbf{x}} \Psi(\mathbf{q} = \mathbf{Vz} + \mathbf{r}, \mathbf{x}) = \frac{\partial \Psi}{\partial \mathbf{x}} + \frac{\partial \Psi}{\partial \mathbf{q}} \left(\frac{\partial \mathbf{V}}{\partial \mathbf{x}} \mathbf{z} + \mathbf{V} \frac{\partial \mathbf{z}}{\partial \mathbf{x}} + \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \right). \tag{4.5}$$

Similar to the full case, the term $\frac{\partial \mathbf{z}}{\partial \mathbf{x}}$ can be replaced using an adjoint formulation. The other new terms ($\frac{\partial \mathbf{V}}{\partial \mathbf{x}}$ and $\frac{\partial \mathbf{r}}{\partial \mathbf{x}}$) will be treated later.

Direct Differentiation Method

This is very similar to the DDM seen in Chapter 2. Although direct differentiation is expensive and thus impractical, it clearly shows the effect of model reduction, and it is a good exercise before deriving the adjoint method. Starting point are the reduced equations of motion:

$$\mathbf{M}_{\mathrm{r}}\ddot{\mathbf{z}} + \mathbf{C}_{\mathrm{r}}\dot{\mathbf{z}} + \mathbf{K}_{\mathrm{r}}\mathbf{z} = \mathbf{V}^{\mathrm{T}}\mathbf{f}. \tag{4.6}$$

We differentiate these with respect to the design variables $\mathbf{x}$:

$$\mathbf{M}_{\mathrm{r}}\ddot{\mathbf{z}}' + \mathbf{C}_{\mathrm{r}}\dot{\mathbf{z}}' + \mathbf{K}_{\mathrm{r}}\mathbf{z}' = \mathbf{f}_{\mathrm{r}}' - \mathbf{M}_{\mathrm{r}}'\ddot{\mathbf{z}} - \mathbf{C}_{\mathrm{r}}'\dot{\mathbf{z}} - \mathbf{K}_{\mathrm{r}}'\mathbf{z} \tag{4.7}$$

$$\mathbf{f}_{\mathrm{r}}' = \mathbf{V}^{\mathrm{T}}\mathbf{f}' + \mathbf{V}'^{T}\mathbf{f} \tag{4.8}$$

$$\mathbf{M}_{\mathrm{r}}' = \mathbf{V}^{\mathrm{T}}\mathbf{M}'\mathbf{V} + 2 \cdot \mathbf{V}^{\mathrm{T}}\mathbf{M}\mathbf{V}' \tag{4.9}$$

$$\mathbf{C}_{\mathrm{r}}' = \mathbf{V}^{\mathrm{T}}\mathbf{C}'\mathbf{V} + 2 \cdot \mathbf{V}^{\mathrm{T}}\mathbf{C}\mathbf{V}' \tag{4.10}$$

$$\mathbf{K}_{\mathrm{r}}' = \mathbf{V}^{\mathrm{T}}\mathbf{K}'\mathbf{V} + 2 \cdot \mathbf{V}^{\mathrm{T}}\mathbf{K}\mathbf{V}'. \tag{4.11}$$

Regrouping the right hand side yields:

$$\mathbf{M}_{\mathrm{r}}\ddot{\mathbf{z}}' + \mathbf{C}_{\mathrm{r}}\dot{\mathbf{z}}' + \mathbf{K}_{\mathrm{r}}\mathbf{z}' = \mathbf{V}^{\mathrm{T}}\mathbf{f}_{\mathrm{ps},1} + \mathbf{V}'^{T}\mathbf{f}_{\mathrm{ps},2} \tag{4.12}$$

$$\mathbf{f}_{\mathrm{ps},1} = \mathbf{f}' - \mathbf{M}'\mathbf{V}\ddot{\mathbf{z}} - \mathbf{C}'\mathbf{V}\dot{\mathbf{z}} - \mathbf{K}'\mathbf{V}\mathbf{z} \tag{4.13}$$

$$\mathbf{f}_{\mathrm{ps},2} = \mathbf{f} - 2\mathbf{MV}\ddot{\mathbf{z}} - 2\mathbf{CV}\dot{\mathbf{z}} - 2\mathbf{KVz} = -\mathbf{f} - 2\hat{\mathbf{r}}. \tag{4.14}$$

Note that the normal pseudo-load appears again, as well as a term dependent on $\mathbf{V}'$. This second term contains both the force and the residual force.

ADJOINT VARIABLE METHOD

We start by trying to find a way to replace $\frac{\partial \mathbf{z}}{\partial \mathbf{x}}$. The adjoint objective is formed by augmenting Ψ with a term, the reduced equations of motion multiplied by an adjoint variable:

$$\Psi_{\mathrm{A}} = \Psi + \int_0^{\mathrm{T}} \boldsymbol{\theta}^{\mathrm{T}} \underbrace{\left(\mathbf{M}_{\mathrm{r}}\ddot{\mathbf{z}} + \mathbf{C}_{\mathrm{r}}\dot{\mathbf{z}} + \mathbf{K}_{\mathrm{r}}\mathbf{z} - \mathbf{V}^{\mathrm{T}}\mathbf{f}\right)}_{=\mathbf{0}} \mathrm{d}t\,. \tag{4.15}$$

Again, we focus for now on the added term. When this is differentiated with respect to **x**, we find:

$$\int_0^{\mathrm{T}} \boldsymbol{\theta}^{\mathrm{T}} \left(\mathbf{M}_{\mathrm{r}}\ddot{\mathbf{z}}' + \mathbf{C}_{\mathrm{r}}\dot{\mathbf{z}}' + \mathbf{K}_{\mathrm{r}}\mathbf{z}'\right) - \mathbf{R}\,\mathrm{d}t = 0 \tag{4.16}$$

$$\mathbf{R} = \boldsymbol{\theta}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}}\left(\mathbf{f}' - \mathbf{M}'\mathbf{V}\ddot{\mathbf{z}} - \mathbf{C}'\mathbf{V}\dot{\mathbf{z}} - \mathbf{K}'\mathbf{V}\mathbf{z}\right) + \boldsymbol{\theta}^{\mathrm{T}}\mathbf{V}'^{T}\mathbf{f}_{\mathrm{ps},2}\,. \tag{4.17}$$

The next steps are identical to what we did in the earlier part: move derivatives using integration by parts, and regroup terms. This yields[2]:

$$\mathbf{M}_{\mathrm{r}}\ddot{\boldsymbol{\theta}} - \mathbf{C}_{\mathrm{r}}^{\mathrm{T}}\dot{\boldsymbol{\theta}} + \mathbf{K}_{\mathrm{r}}\boldsymbol{\theta} = -\left(\frac{\partial \Phi}{\partial \mathbf{q}}\mathbf{V}\right)^{\mathrm{T}} \tag{4.18}$$

$$\boldsymbol{\theta}(T) = \mathbf{0} \tag{4.19}$$

$$\mathbf{M}\dot{\boldsymbol{\theta}}(T) = \left(\frac{\partial \phi}{\partial \mathbf{q}(T)}\mathbf{V}\right)^{\mathrm{T}} \tag{4.20}$$

$$\frac{\mathrm{d}\Psi_{\mathrm{A}}}{\mathrm{d}\mathbf{x}} = \frac{\partial \Psi}{\partial \mathbf{x}} + \int_0^{T} -\mathbf{R}\,\mathrm{d}t + \frac{\partial \Psi}{\partial \mathbf{q}}\left(\mathbf{V}'\mathbf{z} + \mathbf{r}'\right) \tag{4.21}$$

$$\mathbf{R} = (\mathbf{V}\boldsymbol{\theta})^{\mathrm{T}}\left(\mathbf{f}' - \mathbf{M}'\mathbf{V}\ddot{\mathbf{z}} - \mathbf{C}'\mathbf{V}\dot{\mathbf{z}} - \mathbf{K}'\mathbf{V}\mathbf{z}\right) + \left(\mathbf{V}'\boldsymbol{\theta}\right)^{T}\mathbf{f}_{\mathrm{ps},2}\,. \tag{4.22}$$

$$\tag{4.23}$$

Analogous to the full problem, using the adjoint formulation a different way of calculating the design sensitivity has been found, which involves the solution of an initial and a terminal value problem. There are more terms though, which can be investigated separately:

- $\frac{\partial \Psi}{\partial \mathbf{x}}$: direct dependency, usually zero or available analytically.
- $\int_0^T -(\mathbf{V}\boldsymbol{\theta})^{\mathrm{T}}\left(\mathbf{f}' - \mathbf{M}'\mathbf{V}\ddot{\mathbf{z}} - \mathbf{C}'\mathbf{V}\dot{\mathbf{z}} - \mathbf{K}'\mathbf{V}\mathbf{z}\right)\mathrm{d}t$: the most important term, adjoint variable multiplied by pseudo-load. If **r** and $\mathbf{r}_\lambda$ are zero, $\mathbf{V}\mathbf{z} = \mathbf{q}$ and $\mathbf{V}\boldsymbol{\theta} = \boldsymbol{\lambda}$, which makes this term identical to the sensitivity equation for the full system.

These two terms were included in the limited reduction discussed earlier. All the other terms are new:

- $\int_0^T \left(\mathbf{V}'\boldsymbol{\theta}\right)^T \mathbf{f}\,\mathrm{d}t$: difficult to evaluate, discussed in the remainder of this section.
- $\int_0^T \left(\mathbf{V}'\boldsymbol{\theta}\right)^T 2\hat{\mathbf{r}}\,\mathrm{d}t$: this term is very similar to the previous one, but for accurate MOR $\mathbf{f} \gg \hat{\mathbf{r}} \approx \mathbf{0}$. So although $\hat{\mathbf{r}}$ can be calculated cheaply, neglecting this term will not lead to large errors.
- $\frac{\partial \Psi}{\partial \mathbf{q}}\mathbf{V}'\mathbf{z}$: difficult to evaluate, discussed in the remainder of this section.
- $\frac{\partial \Psi}{\partial \mathbf{q}}\mathbf{r}'$: similar to the previous one. Because **r** is unknown, evaluation is difficult. But since $\mathbf{V}\mathbf{z} \gg \mathbf{r}' \approx \mathbf{0}$ this term can be neglected as well.

Note that the analysis so far is exact, with only one exception. Equations (4.18) and (4.20) contain the term $\frac{\partial \Phi}{\partial \mathbf{q}}$ which is often dependent on **q**. But **q** is not exactly known, there is a residual **r**. As with any term containing the residual, this is neglected. To retain compact mathematics this was not included in the derivation.

In summary, the design sensitivity consists of the terms found in the sensitivity analysis without MOR, plus two terms dependent on $\mathbf{V}'$, and two terms dependent on the residual, which can be neglected for accurate

[2] Note that $\boldsymbol{\theta}$ turns out to be the solution of the reduced adjoint equation. In other words, the $\boldsymbol{\theta}$ used as Lagrange multiplier in the detailed analysis is the same as the one found in the limited analysis.

MOR. In general computing basis vector sensitivities $\mathbf{V}'$ is expensive, so these are often neglected. To investigate their significance, the terms containing $\mathbf{V}'$ will be examined. Ignoring the residual, these are:

$$\Psi'_{V'} = \frac{\partial\Psi}{\partial\mathbf{q}}\mathbf{V}'\mathbf{z} + \int_0^T \left(\mathbf{V}'\boldsymbol{\theta}\right)^{\mathrm{T}}\mathbf{f}\,\mathrm{d}t\,. \tag{4.24}$$

Since this term is the main difference between sensitivity analysis with and without MOR, we will study it in some detail. First of all, Eq. (4.24) can be rewritten as:

$$\Psi'_{V'} = \sum_{i=1}^{n}(\mathbf{V}'_i)^{\mathrm{T}}\left[\left(\frac{\partial\Psi}{\partial\mathbf{q}}\right)^{\mathrm{T}} z_i + \int_0^t \theta_i\mathbf{f}\,\mathrm{d}t\right] \tag{4.25}$$

$$\Psi'_{V'} = \sum_{i=1}^{n}(\mathbf{V}'_i)^{\mathrm{T}}\left[\left(\frac{\partial\phi}{\partial\mathbf{q}(T)}^{\mathrm{T}} + \int_0^T \frac{\partial\Phi}{\partial\mathbf{q}}^{\mathrm{T}} z_i\,\mathrm{d}t\right) + \int_0^T \theta_i\mathbf{f}\,\mathrm{d}t\right]. \tag{4.26}$$

The term within brackets can be computed without evaluating $\mathbf{V}'$, which might open possibilities for a cheap error estimate. For a special case this term can be shown to be equal to zero. If the objective function is the integral of the compliance over time, Eq. (4.24) is reduced to a much simpler form:

$$\Psi = \int_0^T \mathbf{f}^{\mathrm{T}}(t)\mathbf{q}(t)\,\mathrm{d}\frac{1}{T} \tag{4.27}$$

$$\frac{\partial\Psi}{\partial\mathbf{q}}\mathbf{V}'\mathbf{z} + \int_0^T \left(\mathbf{V}'\boldsymbol{\theta}\right)^{T}\mathbf{f}\,\mathrm{d}t = \int_0^T \mathbf{f}^{\mathrm{T}}\mathbf{V}'(\frac{\mathbf{z}}{T} + \boldsymbol{\theta})\,\mathrm{d}t\,. \tag{4.28}$$

The adjoint equation is also much simpler in this case:

$$\mathbf{M}_{\mathrm{r}}\ddot{\mathbf{z}} + \mathbf{C}_{\mathrm{r}}\dot{\mathbf{z}} + \mathbf{K}_{\mathrm{r}}\mathbf{z} = \mathbf{V}^{\mathrm{T}}\mathbf{f} \tag{4.29}$$

$$\mathbf{M}_{\mathrm{r}}\ddot{\boldsymbol{\theta}} - \mathbf{C}_{\mathrm{r}}^{\mathrm{T}}\dot{\boldsymbol{\theta}} + \mathbf{K}_{\mathrm{r}}\boldsymbol{\theta} = -\left(\frac{\partial\Phi}{\partial\mathbf{q}}\mathbf{V}\right)^{\mathrm{T}} = -\frac{\mathbf{V}^{\mathrm{T}}\mathbf{f}}{T}\,. \tag{4.30}$$

This means the primal and adjoint equation are nearly identical, except for a scaling term and the initial/terminal conditions. Finally the assumption is made that the load is 'symmetric' in time, which means:

$$\mathbf{f}(\tau) = \mathbf{f}(T-\tau), \qquad 0 \le \tau \le T\,. \tag{4.31}$$

When $\mathbf{z}(0) = \dot{\mathbf{z}}(0) = \mathbf{0}$, this leads to:

$$\boldsymbol{\theta}(\tau) = -\mathbf{z}(t)/T \tag{4.32}$$

$$\int_0^T \frac{\partial\Phi}{\partial\mathbf{q}}^{\mathrm{T}} z_i\,\mathrm{d}t + \int_0^T \theta_i\mathbf{f}\,\mathrm{d}t = \int_0^T \mathbf{f}z_i/T\,\mathrm{d}t + \int_0^T -\mathbf{f}z_i/T\,\mathrm{d}\tau = 0\,. \tag{4.33}$$

So for the described case Eq. (4.24) will be zero, which means that the contribution of the basis vector sensitivities to the total sensitivity is negligible.

In general however this will not hold, Eq. (4.24) will not be zero. To demonstrate this, analytical expressions for $\mathbf{z}$ and $\boldsymbol{\theta}$ were found using Duhamel's integral. The resulting mathematics span a page, and are not very insightful. The same can be shown for a simple system with only one degree of freedom, where $m = k = 1$, $c = 0$, and $V = 1$, so $\mathbf{q} = \mathbf{z}$ and $\boldsymbol{\lambda} = \boldsymbol{\theta}$. Eq. (4.24) is evaluated for three cases: dynamic compliance with symmetric load, different objective with symmetric load, and dynamic compliance with non-symmetric load:

		Different objective	Different force (ramp)
Objective	$\Psi = \int_0^T f q \,\mathrm{d}t$	$\Psi = \int_0^T z^2 \,\mathrm{d}t$	$\Psi = \int_0^T f q \,\mathrm{d}t$
Equations of motion:	$\ddot{z} + z = f = 1$ $z(t=0) = \dot{z}(t=0) = 0$	$\ddot{z} + z = f = 1$ $z(t=0) = \dot{z}(t=0) = 0$	$\ddot{z} + z = f = t$ $z(t=0) = \dot{z}(t=0) = 0$
Adjoint equation:	$\ddot{\theta} + \theta = -1$ $\theta(\tau = 0) = \dot{\theta}(\tau = 0) = 0$	$\ddot{\theta} + \theta = -2z$ $\theta(t = T) = \dot{\theta}(t = T) = 0$	$\ddot{\theta} + \theta = -t$ $\theta(t = T) = \dot{\theta}(t = T) = 0$
Solution for $T = \pi$:	$z(t) = 1 - \cos(t)$ $\theta(\tau) = -1 + \cos(\tau)$	$z(t) = 1 - \cos(t)$ $\theta(t) = (t - \pi + 2)\sin(t)$ $-2\cos(t) - 2$	$z(t) = t - \sin(t)$ $\theta(t) = -t + T\cos(t) + \sin(t)$
Eq. (4.24)=	0	$4\mathbf{V}'$	$-2T\mathbf{V}'$

For the single DoF system this does not lead to an error in sensitivities, since $\mathbf{V}$ is constant and therefore $\mathbf{V}'$ is zero. But in general, this does show that the basis vector sensitivities have an effect on the design sensitivity. In Chapter 5 this will be investigated on some numerical examples.

4.1.4. Summary

Model order reduction can be used for solving the equations of motion and the adjoint equation efficiently. The easiest way of doing this ignores both MOR residuals and basis vector sensitivities. This leads to multiple errors, most notably in the computed sensitivity. The errors caused by ignoring the residuals will be of the same order as the residuals themselves. This means that for accurate MOR these can be neglected. The effect of basis vector sensitivities will depend on the objective function and load. The error resulting from ignoring this term can be numerically evaluated, but this does require the (expensive) calculation of $\mathbf{V}'$.

4.2. Basis vector sensitivities

There are multiple ways to treat basis vector sensitivities:

- Calculate $\mathbf{V}'$ in one of the following ways:
  - Finite difference: not attractive due to the large amount of design variables.
  - Direct differentiation & adjoint methods: there are various possibilities here, depending on the type of basis vector. For literature on eigenvector sensitivities, see for example [53]. For a brief investigation into Ritz vector sensitivities, see Appendix B.
- Assume $\mathbf{V}' = \mathbf{0}$. This is very common, for example in [35]. In all known research on combining topology optimization and MOR, namely the work of Yoon [17] and Zhao & Wang [40], the issue of basis vector sensitivities is not mentioned. It is therefore assumed that they also used $\mathbf{V}' = \mathbf{0}$.

Eigenvector sensitivities have been used to increase the accuracy of sensitivity analysis, for example in the work of Greene and Haftka [42], or in the PhD. project of Evert Hooijkamp. But in general, calculating basis vector sensitivities is expensive. There are multiple ways to proceed from here, the simplest is assuming $\mathbf{V}'$ is zero. If the solution of the two differential equations can be found accurately using model order reduction, there will be no difference between the results of the full and reduced model. This leads to the hypothesis that basis vector sensitivities have a negligible influence on the design sensitivity when the model order reduction is accurate.

However, as was shown before, there will be an error, given by Eq. (4.24). To verify if this really is negligible, an analysis of the errors in design sensitivities will be performed in Chapter 5. The difference between the sensitivities of the full and reduced models will be compared, and the contribution of Eq. (4.24) to the error will be evaluated. To find $\mathbf{V}'$ a finite difference approximation will be used.

4.3. Adaptive Model Order Reduction

In Section 2.5.5 multiple ways of estimating the error caused by MOR were discussed. Based on these, two algorithms for Adaptive Model Order Reduction (AMOR) are defined, Algorithms 4.1 and 4.2. The first uses only estimates based on the model and basis vectors (a priori), specifically the *spatial load estimate* and the *spectral load estimate*, as defined in Section 2.5.5. The second algorithm uses response (a posteriori) information to calculate the *refinement estimate*. Both aim to select the number of basis vectors such that a certain fixed accuracy is reached.

Algorithm 4.1 Algorithm for a priori adaptive Model Order Reduction.

```
$n \leftarrow n_{\text{min}}$
Calculate $\mathbf{V}$
Calculate $e_{\text{spatial}}$ and $e_{\text{spectral}}$
while $e_{\text{spatial}} > e_{\text{spatial,ref}}$ and $e_{\text{spectral}} > e_{\text{spetral,ref}}$ and $n < n_{\text{max}}$ do
    $n \leftarrow n + 1$
    Add vector to $\mathbf{V}$
    Calculate $e_{\text{spatial}}$ and $e_{\text{spectral}}$
end while
```

Algorithm 4.2 Algorithm for a posteriori adaptive Model Order Reduction.

```
1: $e = \frac{\int_0^T \dot{z}_n^2(t)\,\mathrm{d}t}{\sum_{j=1}^{n} \int_0^T \dot{z}_j^2(t)\,\mathrm{d}t}$
2: if $e > e_{\text{ref}}$ and $n < n_{\text{max}}$ then
3:     $n \leftarrow n + (\Delta n - 1)$
4: else if $n > n_{\text{min}}$ then
5:     $n \leftarrow n - (\Delta n - 1)$
6: end if
```

In Algorithm 4.2 there are multiple possibilities for Δn. In this research the value e/e_{ref} will be used for positive Δn (line 3), and e_{ref}/e for negative (line 5). In both cases the value is rounded up, and limited to a maximum change of 10.

This algorithm does not lead to an constant number of basis vectors, in the ideal case it switches every iteration between a certain n and $n+1$. Although this could easily be avoided using a different algorithm where the single reference error is replace by an upper and lower bound, it is not seen as a problem, so the simple implementation is used. To avoid larger fluctuations, Δn is reduced by 1 in line 3 and 5, this results in a damping effect on the change in n.

These two algorithms are both used in the optimization process, as illustrated by Figure 4.1. Algorithm 4.1 is used before the optimization, to estimate the correct number of basis vectors. Algorithm 4.2 is used in the optimization loop, changing the number of basis vectors used by the next iteration.

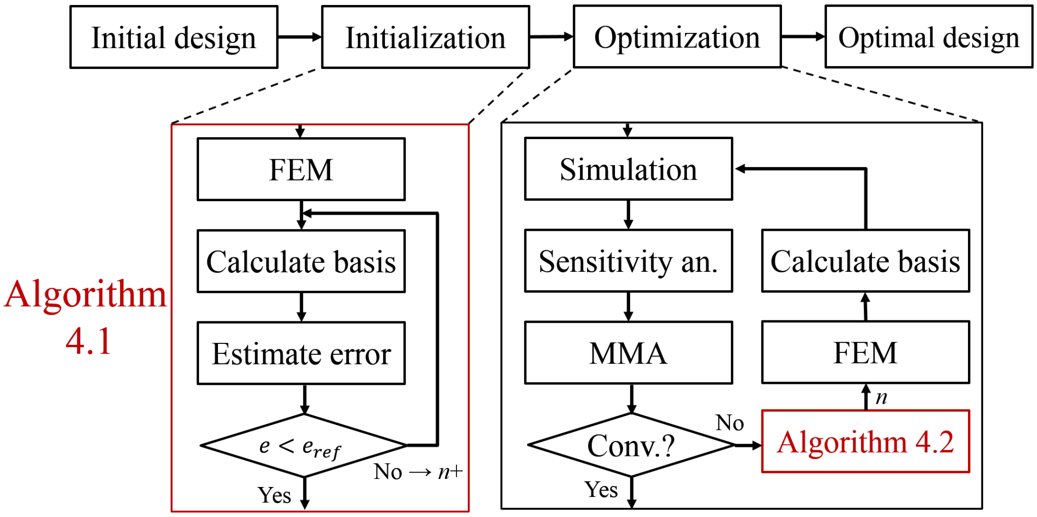


Figure 4.1: Overview of optimization loop with adaptive model order reduction. Before starting the optimization, Algorithm 4.1 is used to estimate the correct number of basis vectors. During the optimization Algorithm 4.2 is used to maintain a constant accuracy.

4.4. SUMMARY & DISCUSSION

In this chapter the implementation of model order reduction has been discussed in the context of the total optimization loop. Two approaches have been discussed. The first uses model reduction only to solve the equations of motion and adjoint equation, neglecting all residuals. This is expected to be efficient when the model order reduction is accurate. To investigate this further, a detailed analysis was performed. This provided more information on the resulting error in the sensitivity analysis, including the contribution of basis vector sensitivities. When the latter are ignored, the approaches are identical. The resulting accuracy will be studied in the next chapter.

Finally, a method has been proposed for adaptive model order reduction as part of the optimization.

5

Numerical Examples: Model Order Reduction

In this chapter we will examine the properties of Model Order Reduction. The aim is to investigate the accuracy and computational effort of the different methods. First the issue of local modes and penalization will be examined. Next the discussed methods of MOR will be tested on various test structures and two load cases, investigating the error in both the response and the sensitivities. Finally we will test the basis updating scheme discussed in Section 2.5.6.

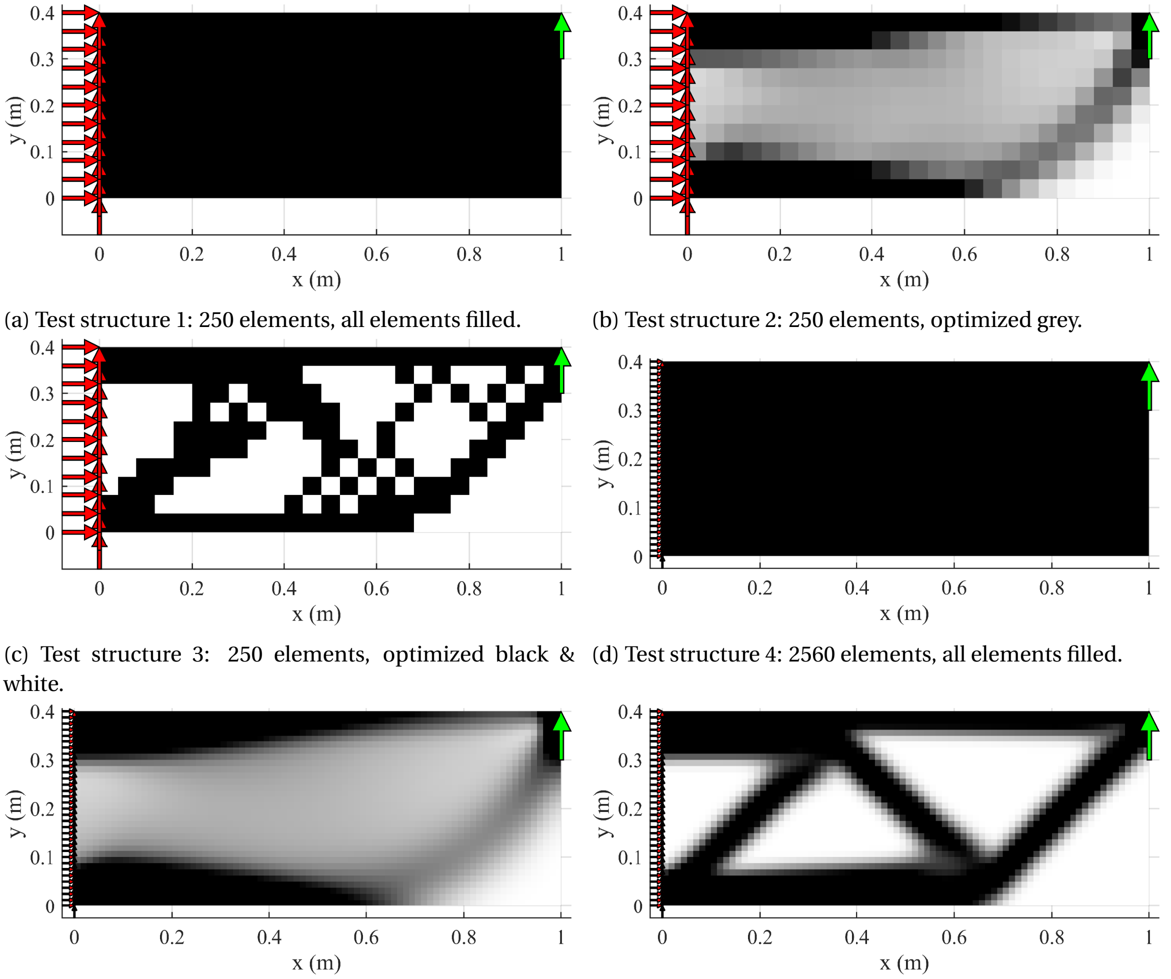
5.1. Test problems

A good set of test problems should be representative of any structure that could be found during optimization. Therefore six test structures were used, with either 250 or 2560 elements, with the following geometries:

- Completely filled
- Optimized grey: minimum static compliance, without penalization or filtering.
- Optimized black & white, also minimum static compliance. The structure was optimized using SIMP penalization, with $p = 3$. A density filter with a radius of three elements was used to find the structure with 2560 elements.

As explained in the previous section, the finite element analysis uses PI penalization, with $p_1 = 3$ and $p_2 = 15/16$. The structures are shown in Fig. 5.1, the corresponding number of DoFs is respectively 550 and 5280.

In the previous chapter multiple definitions of accuracy were discussed. For sensitivity analysis, the differentiation is often made between accuracy and consistency, which respectively refer to the error made with respect to the physical reality and the model. In this chapter a third measure will be used, named *MOR error*. This refers to the error between the full and reduced model, which could be seen as part of the consistency. To distinguish between different errors, this third definition is necessary.



(a) Test structure 1: 250 elements, all elements filled.

(b) Test structure 2: 250 elements, optimized grey.

(c) Test structure 3: 250 elements, optimized black & white.

(d) Test structure 4: 2560 elements, all elements filled.

(e) Test structure 5: 2560 elements, optimized grey

(f) Test structure 6: 2560 elements, optimized black & white. A density filter was used to calculate this structure, with a radius of 3 elements.

Figure 5.1: Test structures used for analysis of Model Order Reduction. Black areas indicate maximum density ($x = 1$), white minimum density ($x = 1 \times 10^{-3}$). Green arrows indicate the applied force, red arrows boundary conditions (clamped on left side). The structures were optimized for minimum compliance.

5.2. LOCAL MODES & PENALIZATION

It is well known from literature that some penalization techniques lead to the existence of *local modes*[3]. This happens for example when SIMP penalization is applied to the stiffness. Void areas will have a very high mass/stiffness ratio, causing a very low eigenfrequency. The eigensolver will find these local modes, but they are undesired because they contain very little information about the actual behaviour of the structure. In this section it will briefly verified that we can avoid them using an different penalization method (Section 2.2.1).

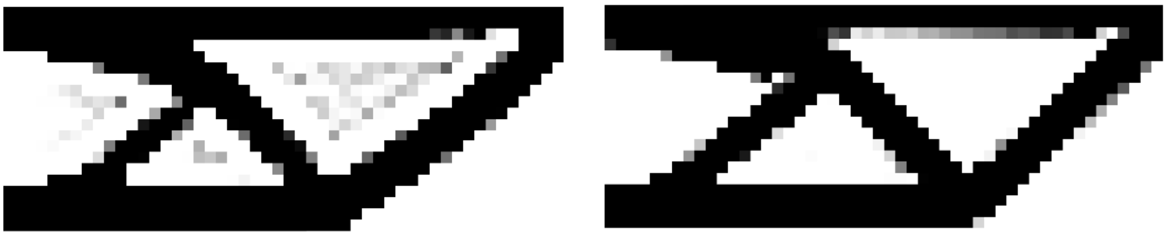
For all types of penalization discussed in this document, the first three eigenfrequencies are shown in Table 5.1. SIMP penalization clearly produces very low values for structures with low-density areas. For both PI and SIMP penalization, the first 3 eigenmodes (labelled as MSM) and Ritz vectors are plotted in Table 5.2. This was done for test structure 4-6, the results for structures 1-3 are similar.

Table 5.1: First 3 eigenfrequencies for both SIMP and PI penalization. All eigenfrequencies are in rad/s. Local modes are indicated by a red background. Penalization does not affect completely filled structure (structure 4). SIMP penalization result in local modes. the other two methods do not. They do lower the eigenfrequency, which was expected, since penalization results in lower stiffness.

	Mode nr	Structure 4	Structure 5	Structure 6
None	1	1851	2583	2229
	2	7850	8104	5642
	3	7984	9568	7025
SIMP	1	1851	24.31	3.614
	2	7850	39.17	5.745
	3	7984	50.40	6.589
RAMP	1	1851	2065	2035
	2	7850	5916	4423
	3	7984	7637	5669
PI	1	1851	1712	1982
	2	7850	4810	4005
	3	7984	6700	5134

In these results local modes are present in grey/void areas when we apply SIMP penalization. The corresponding eigenfrequencies are very low. This is also seen in Ritz vectors, which is caused by their convergence towards the first eigenmode. This means the reduced basis will contain less information on the vibration of the actual structure.

RAMP and PI penalization on the other hand do not produce any local modes. Therefore both were tested on optimization problems, where PI showed better results (fewer grey areas). This is illustrated by Fig. 5.2. Therefore, in the rest of this research PI penalization will be used.

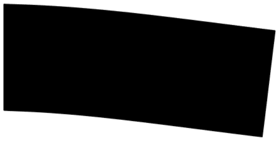
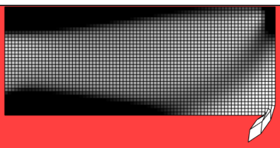
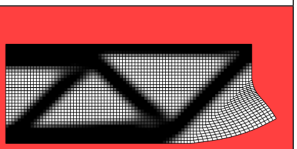
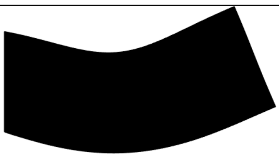
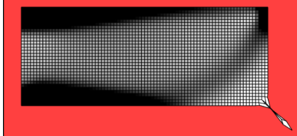
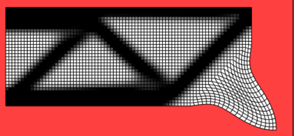
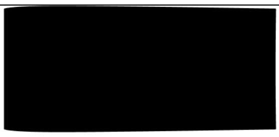
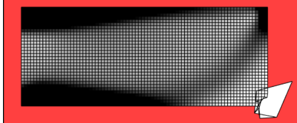
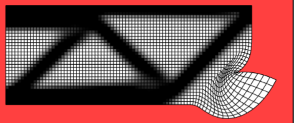
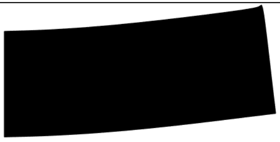
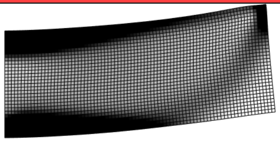
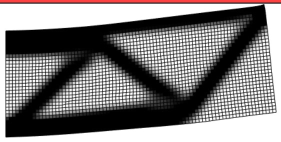
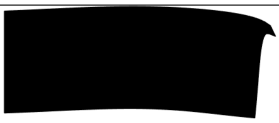
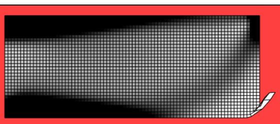
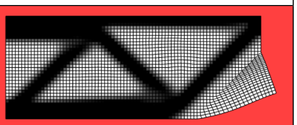
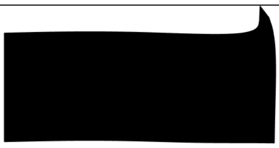
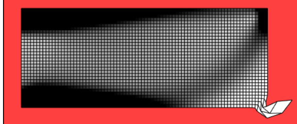
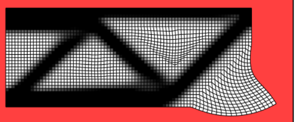
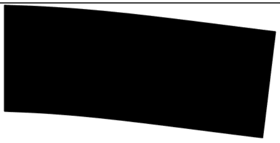
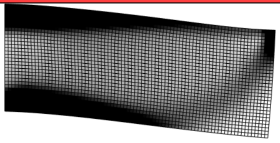
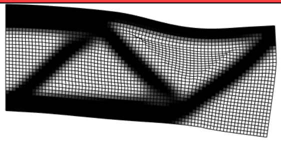
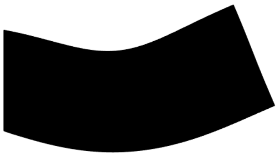
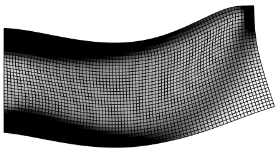
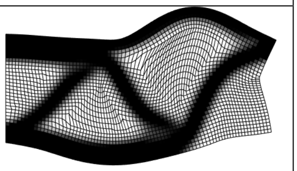
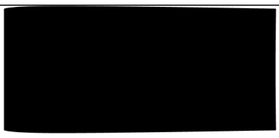
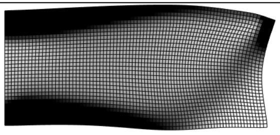
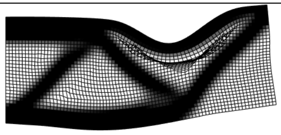
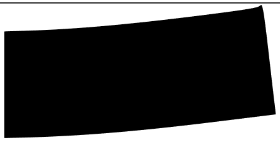
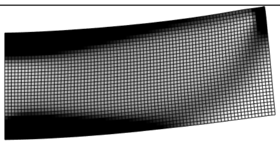
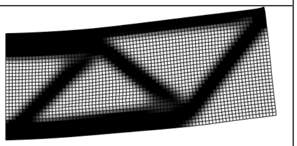
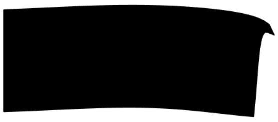
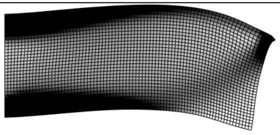
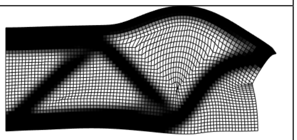
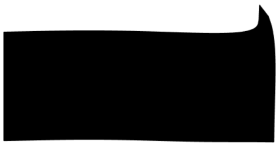
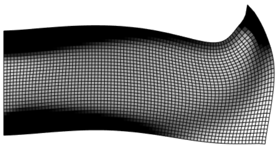
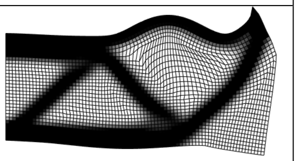


(a) RAMP penalization.

(b) PI penalization

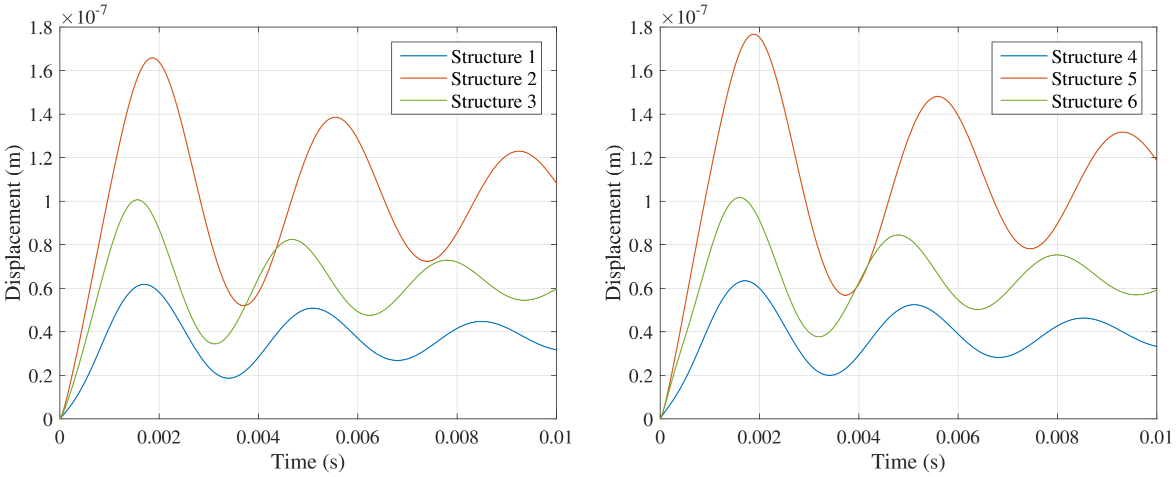
Figure 5.2: Optimized designs for both RAMP and PI penalization. PI penalization results in fewer grey areas, so this is used for the rest of this research.

Table 5.2: Basis vectors using SIMP & PI penalization. MSM is modal superposition method (eigenvectors), red background indicates local mode. SIMP penalization leads to local modes, this also affect any calculated Ritz vectors since these converge towards the first eigenvector.

	Vector nr	Structure 4	Structure 5	Structure 6
SIMP	MSM 1		🟥 	🟥 
	MSM 2		🟥 	🟥 
	MSM 3		🟥 	🟥 
	Ritz 1			
	Ritz 2		🟥 	🟥 
	Ritz 3		🟥 	🟥 
PI	MSM 1			
	MSM 2			
	MSM 3			
	Ritz 1			
	Ritz 2			
	Ritz 3			

5.3. Example 1: step response

For the first numerical example a step response will be studied. At this point, only the simulation is performed, the objective function and sensitivity analysis will be treated in Section 5.5. Starting from initial conditions $\mathbf{q} = \dot{\mathbf{q}} = \mathbf{0}$ all six test structures are subjected to a vertical force $f = 1\,\mathrm{N}$ for $t \geq 0$, applied at the top right corner. The simulated timespan is 0.01 s, which is discretized into 500 timesteps. For a list of other parameters see Appendix D. The resulting displacement is plotted in Fig. 5.3. All structures show a similar dynamic response.



(a) Response of structure 1-3.

(b) Response of structure 4-6.

Figure 5.3: Displacement versus time of all test structures when subject to a step force. Displacement measured is vertical at top-right corner, the force is applied to the same DoF. The response consists of the static displacement and a decaying oscillating part, which is dominated by the first eigenmode.

5.3.1. Model order reduction error

For this load case various methods of model order reduction will be compared. The most suitable for this problem are modal superposition and Ritz vectors. This is a low-frequency problem, so Quasi-Static Ritz Vectors are not suitable, furthermore there is no obvious center frequency. The modal acceleration method is expected to be inferior to Ritz vectors, but for reference it will be included in the test.

To calculate the accuracy of the response of a reduced system, the response of the full system is calculated first. The error caused by the model reduction will be calculated by:

$$e = \frac{\|q_{j,\text{full}}(t_i) - q_{j,\text{red}}(t_i)\|}{\|q_{j,\text{full}}(t_i)\|}, \qquad 0 \leq i \leq n_t, \tag{5.1}$$

where j corresponds to a certain DoF, in this case the same as the DoF of the load. Other error measures are possible, for example using information of all degrees of freedom. The results were found to be similar. For this research it makes sense to focus on one degree of freedom, since all used objective functions use only one degree of freedom as well.

The results are shown in Fig. 5.4. From these some preliminary observations can be made:

- For this combination of load and structure the MSM has a very slow convergence.
- The MAM reduces the error by approximately one order of magnitude.
- Ritz vectors show much faster convergence, reaching 10^{-13} using approximately 50 basis vectors. The remaining error is contributed to numerical round-off.
- The MOR error of the Ritz vectors is dependent on the structure. The completely black structures are more accurate than the others, the grey structures are slightly more accurate than the black & white

ones. This was found to be caused both by the structure and the penalization. Penalization and large differences in element densities both lead to higher condition numbers for the system matrices. These result in larger errors in the Ritz vectors, leading to larger MOR errors for the simulation. For details, see Fig. 5.6.

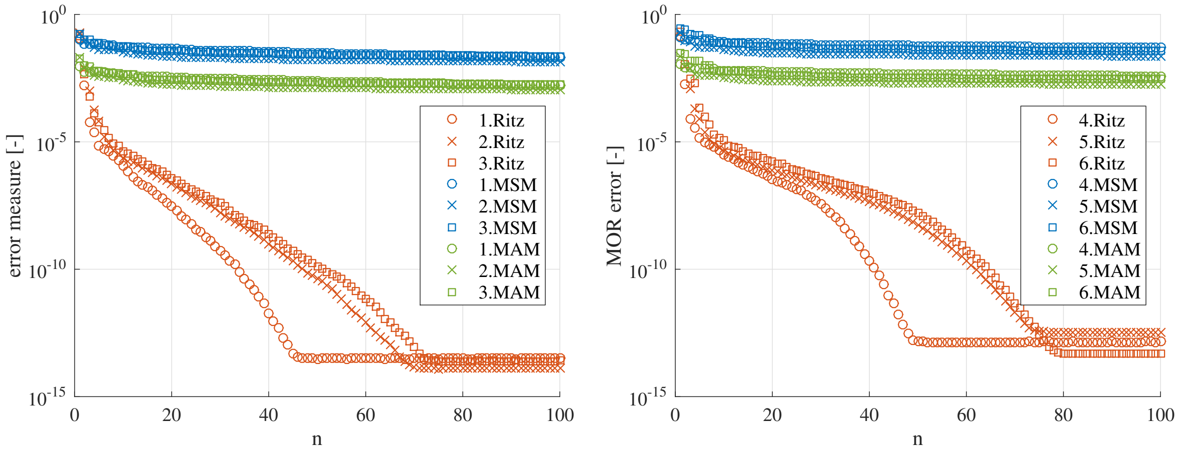
For the modal methods the grey structures result in a slightly higher accuracy, but the effect is much smaller than for the Ritz vectors.

- The convergence of all methods is dependent on the size of the problem. This is especially visible for the Ritz vectors. But while the size of the problems is one order of magnitude apart, the required basis size for a typical accuracy differs only by a small amount.

5.3.2. Computational effort

Next for this load case the computational effort will be examined. In Fig. 5.5 the CPU-time of the reduced simulation is shown (calculation of basis vectors + numerical integration), divided by the CPU-time of the full simulation. This is highly dependent on the simulation parameters, especially the number of time steps. Still it does provide some insight into the efficiency of the different methods. The results show that MOR can significantly decrease the computation time, especially for larger systems. The MSM is shown to be more expensive than the Ritz vector method, which is caused by the higher costs of basis generation. The extra CPU-time associated with the MAM is relatively small.

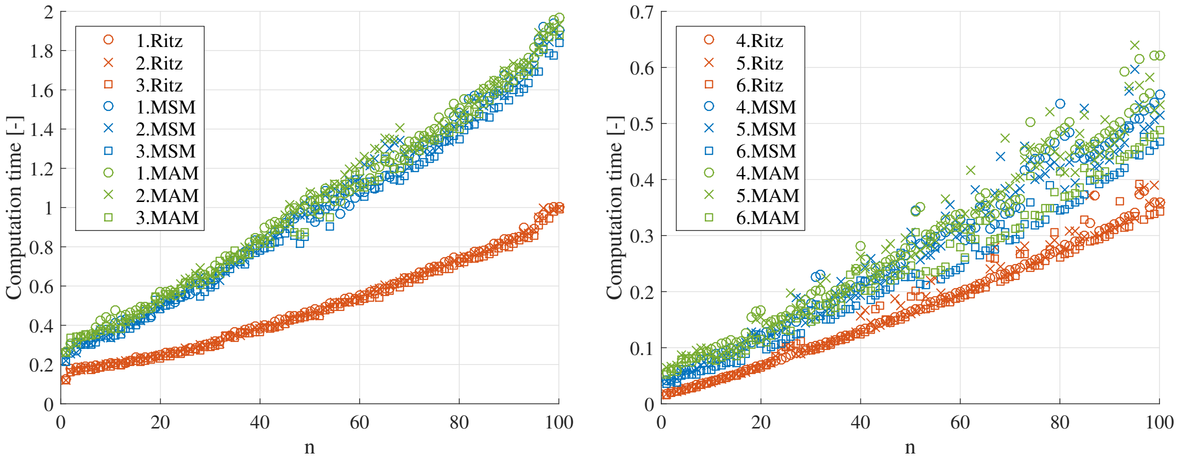
In Chapter 6 the CPU time will be discussed as part of the total optimization loop, where the time will also be separated into basis generation and time integration.



(a) Accuracy of structure 1-3. The number in the legend indicates which structure is used.

(b) Accuracy of structure 4-6. The number in the legend indicates which structure is used.

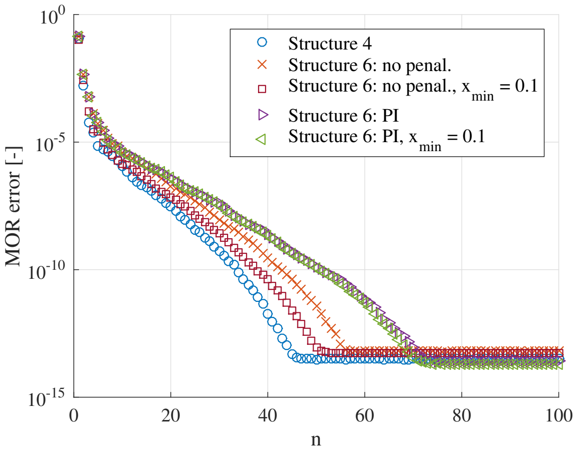
Figure 5.4: Error caused by model order reduction for all six test structures, for step response, versus number of basis vectors n. Numbers in legend correspond to test structure. Modal superposition (MSM) result in poor accuracy, modal acceleration (MAM) improves this by an order of magnitude. Ritz vectors show superior convergence.



(a) Computation time of structure 1-3.

(b) Computation time of structure 4-6.

Figure 5.5: Computation time of reduced analysis, divided over computation time of full analysis. Plotted versus number of basis vectors, results averaged over 10 simulations. The data shows that the Ritz vector method is cheaper than the modal methods, furthermore the efficiency of all methods increases with the size of the system.



(a) MOR error for structures with different penalization and lower limits for the design variables.

Struct.	Penal.	$x_{\min}$	Cond. **K**	Cond. **M**
4	None	1	4.95×10^{4}	2.24×10^{1}
6	None	1×10^{-3}	2.11×10^{6}	2.24×10^{5}
6	None	0.1	7.51×10^{4}	2.24×10^{2}
6	PI	1×10^{-3}	3.25×10^{7}	2.24×10^{5}
6	PI	0.1	1.12×10^{5}	2.24×10^{2}

(b) Condition numbers (2-norm) associated with the system matrices of the tested structures

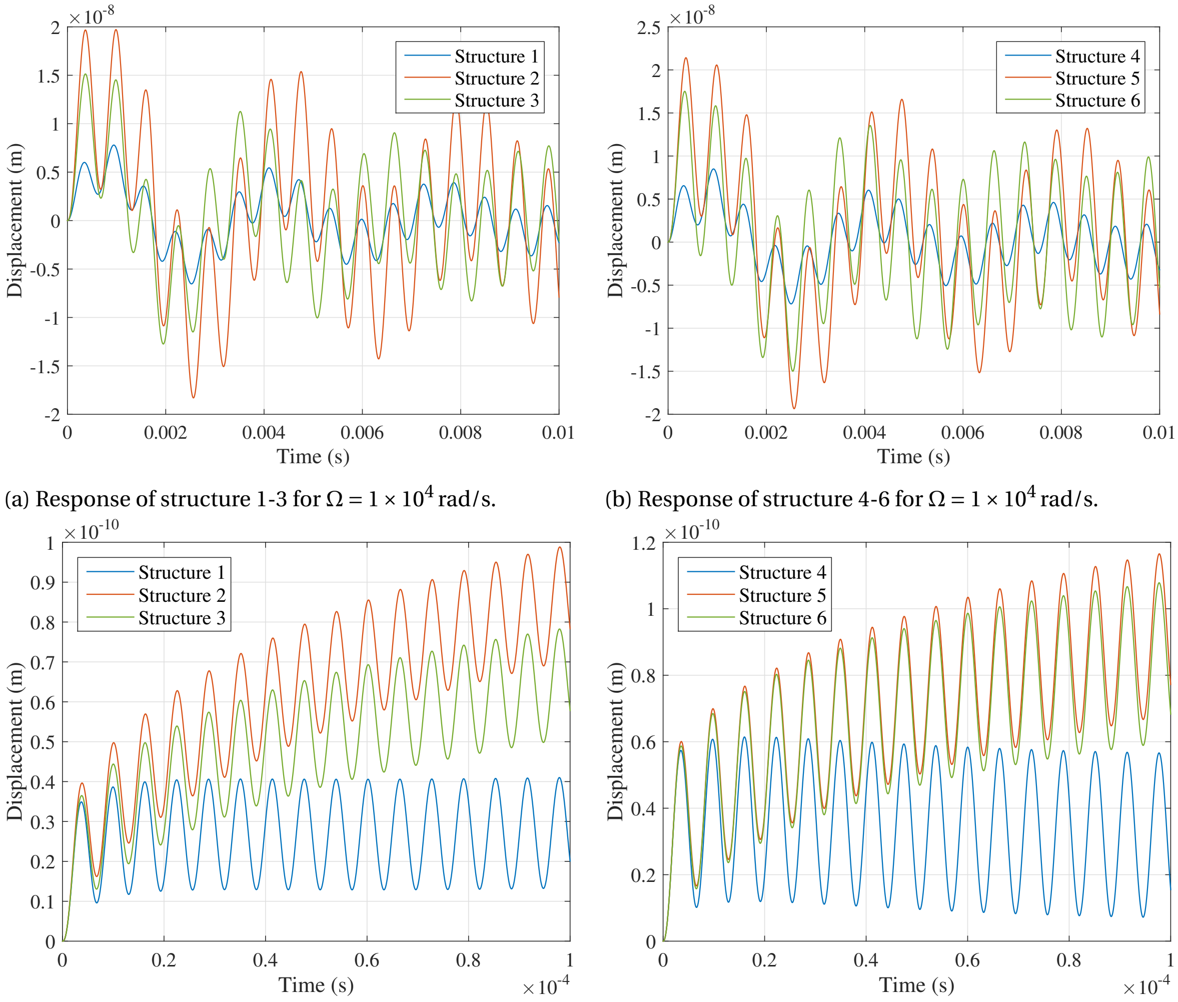
Figure 5.6: MOR error of Ritz vector method for structures with different numerical conditioning. Structure 4 is completely filled, its system matrices therefore have relatively low condition numbers. Structure 6 has many void elements, leading to poorer conditioning. This is increased by penalization. The results indicate that this increases the MOR error in the Ritz vector method, this is caused by the less accurate Ritz vectors.

5.4. Example 2: harmonic response

In the previous load case the Ritz vector method showed very promising results, which was expected: the load was low-frequency, and the static displacement was an important part of the response. So for the next load case a higher frequency load will be used, to see how this affects accuracy of the various MOR methods. The same six structures were simulated with a sinusoidal load, with an amplitude of 1 N and two values for the load frequency Ω: 1×10^4 rad/s and 1×10^6 rad/s. The simulation times are scaled accordingly to guarantee similar accuracy for each case, the number of time steps is 1000. For the full structure, these frequencies are close to respectively the fifth and 200th eigenfrequency.

The calculated response to the highest of the two frequencies will not be accurate with respect to the real structure, for such high eigenmodes the accuracy of the finite element method will be low[1]. The point of this example however is to show whether the full discretized model can be approximated using a reduced model, errors in the full model are not relevant.

The resulting displacement is shown in Fig. 5.7, the accuracy in Fig. 5.8. The CPU time is similar to the other loadcase, these graphs are not shown. Compared to the results for a step response, Ritz vectors show slower convergence, even more so for the high frequency load. The MAM actually decreases the accuracy of the simulation in this case. This is because it uses a quasi-static response, which only takes the stiffness into account, neglecting the inertia.



(a) Response of structure 1-3 for $\Omega = 1 \times 10^4$ rad/s.

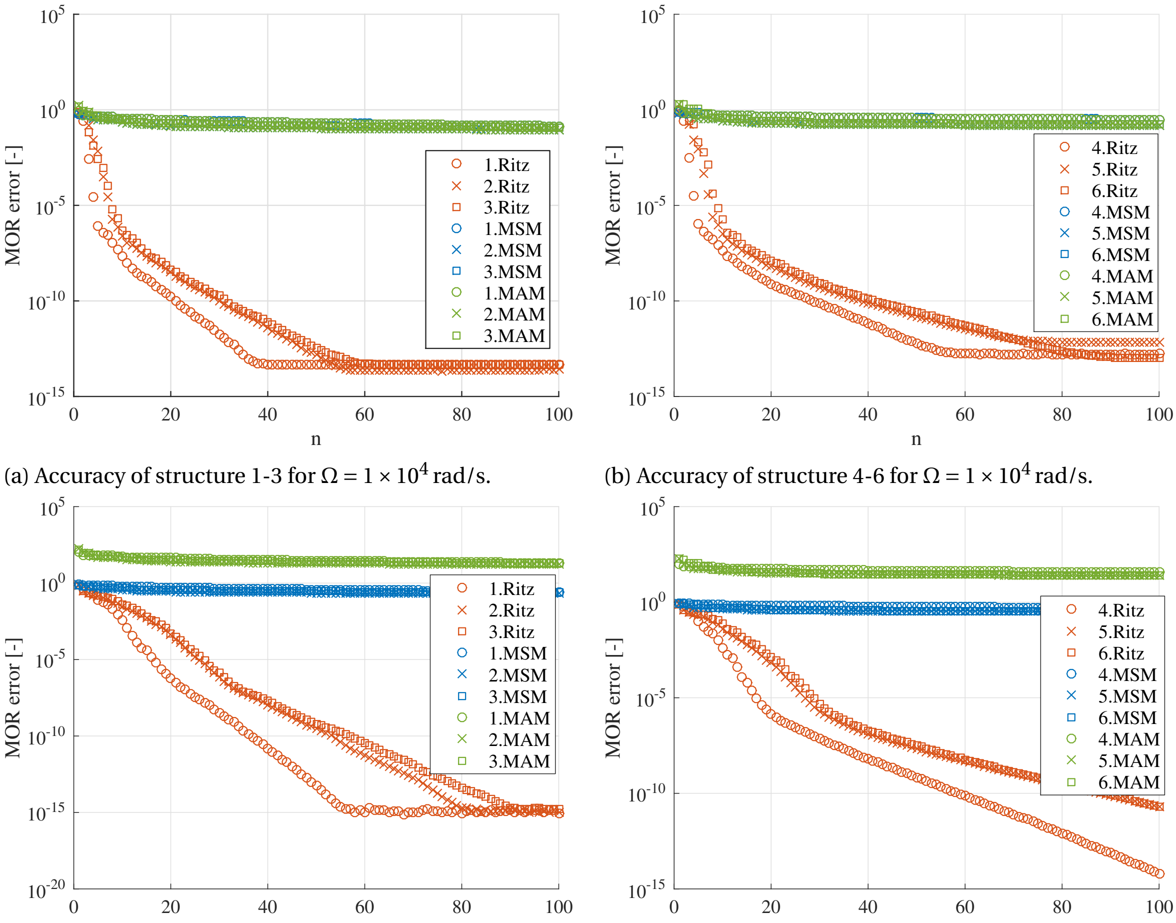
(b) Response of structure 4-6 for $\Omega = 1 \times 10^4$ rad/s.

(c) Response of structure 1-3 for $\Omega = 1 \times 10^6$ rad/s.

(d) Response of structure 4-6 for $\Omega = 1 \times 10^6$ rad/s.

Figure 5.7: Displacement under harmonic load, for all six test structures.

[1] This is clearly visible in Figs. 5.7c and 5.7d. Although structure 1 and 4 represent the same physical structure, their response is different



(a) Accuracy of structure 1-3 for $\Omega = 1 \times 10^{4}$ rad/s.

(b) Accuracy of structure 4-6 for $\Omega = 1 \times 10^{4}$ rad/s.

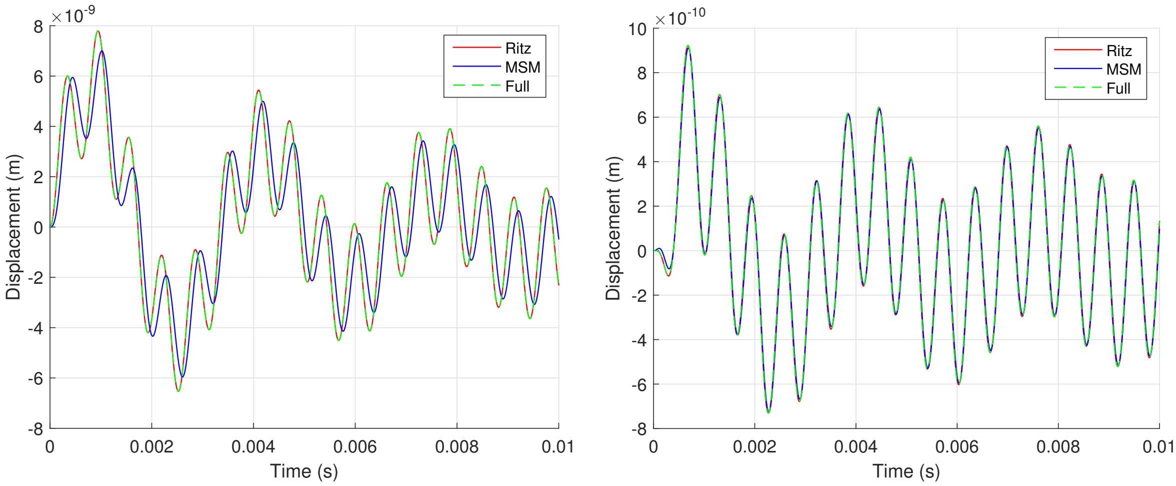
(c) Accuracy of structure 1-3 for $\Omega = 1 \times 10^{6}$ rad/s.

(d) Accuracy of structure 4-6 for $\Omega = 1 \times 10^{6}$ rad/s.

Figure 5.8: Accuracy versus basis vectors for all six test structures, for two sinusoidal loads. The MAM method is not effective for this case, and should be changed to take inertia effects into accounts. The convergence of Ritz vectors is slightly slower compared to the step response.

A remark has to be made here. In measuring the error, only the response of a single degree of freedom has been taken into account. The MSM shows better performance when a different point is examined, illustrated by Fig. 5.9. So the results of MSM are often not as inaccurate as shown in this chapter. However, since this research focuses on point loads, with displacement measured at the same point, this chapter still clearly shows MSM is inefficient, and Ritz vectors show much better performance. There is one exception, which will be discussed next.

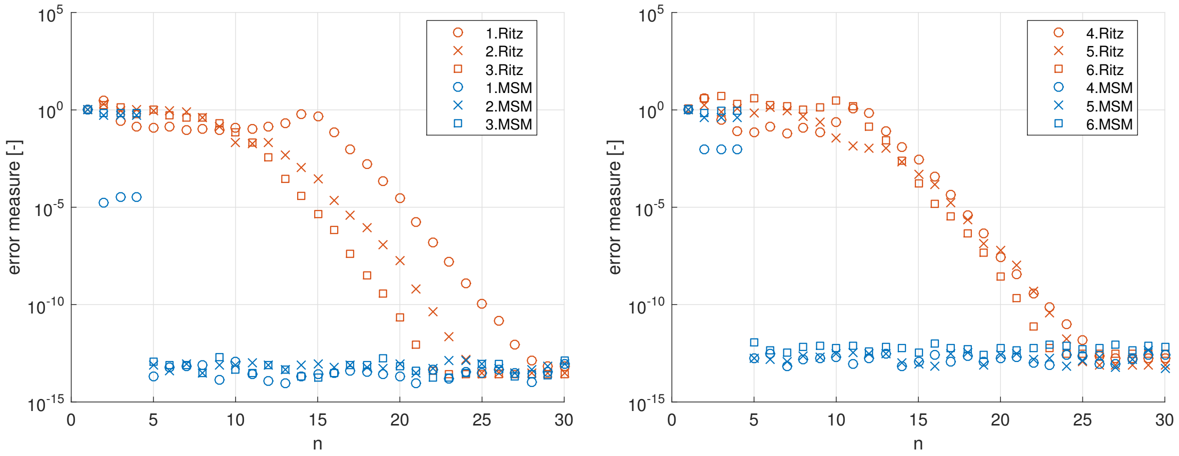
The final example is a case where the displacement is fully described by two eigenmodes, in this example the second and fifth are chosen arbitrarily. This achieved by using an initial displacement given by a combination of these modes, and not applying an external load. For optimization, this has very limited practical use. It does correspond to the absolute best case for modal superposition, and allows us to compare this to Ritz vectors. The Ritz vectors are generated using the same load vectors as before. Since there is no external load, the Ritz vector basis is just an approximation of the eigenvectors. The results are shown in Fig. 5.10 The results clearly show that MSM converges with just 5 vectors, only the second and fifth vector are excited. As expected, Ritz vectors result in inferior performance. So for this specific case it might be best to use MSM. However this load case is not very likely to occur in reality, the dynamic response will usually be distributed over a wide range of frequencies.



(a) Response of structure 1 measured at same DoF as force.

(b) Response of structure 1 measured at DoF number 100.

Figure 5.9: Displacement versus time for a single degree of freedom, for various reduction methods, using 5 basis vectors. The MSM clearly has much better accuracy for the arbitrarily chosen DoF number 100 than for the DoF where the force is applied. However the focus of this research is on objective functions that use the response of a the point where the force is applied.



(a) Response of structure 1-3.

(b) Response of structure 4-6.

Figure 5.10: Accuracy versus number of basis vectors for pure modal excitation. Only mode 2 and 5 are excited, this clearly leads to fast convergence of the MSM. The Ritz vector method is just an inefficient eigenvector algorithm in this example, it therefore converges much slower.

5.5. Objective function & sensitivity analysis

5.5.1. Model order reduction error

In the previous sections the accuracy of the simulations has been investigated. This will now be extended to the objective function and sensitivity analysis. The following error measures are used:

- Objective function:

$$e = \left| \frac{\Psi_{\text{full}} - \Psi_{\text{red}}}{\Psi_{\text{full}}} \right| . \tag{5.2}$$

- Adjoint variable:

$$e = \frac{\|\boldsymbol{\lambda}_{\text{full}}(t_i) - \boldsymbol{\lambda}_{\text{red}}(t_i)\|}{\|\boldsymbol{\lambda}_{\text{full}}(t_i)\|}, \qquad 0 \le i \le n_t . \tag{5.3}$$

  For the displacement, the chosen error measure used only one degree of freedom. For the adjoint the data from all DoFs will be used, so the discretized variable $\boldsymbol{\lambda}(t)$ is a $N \times n_t$matrix. The used norm is: $\|\boldsymbol{\lambda}(\mathbf{t})\| = \sqrt{\sum_i \sum_j \lambda_i^2(t_j)}$ (Frobenius norm).

- Sensitivity:

$$e = \frac{\left\| \frac{\mathrm{d}\Psi_{\text{full}}}{\mathrm{d}d_i} - \frac{\mathrm{d}\Psi_{\text{red}}}{\mathrm{d}d_i} \right\|}{\left\| \frac{\mathrm{d}\Psi_{\text{full}}}{\mathrm{d}d_i} \right\|}, \qquad 1 \le i \le n_{\text{el}} . \tag{5.4}$$

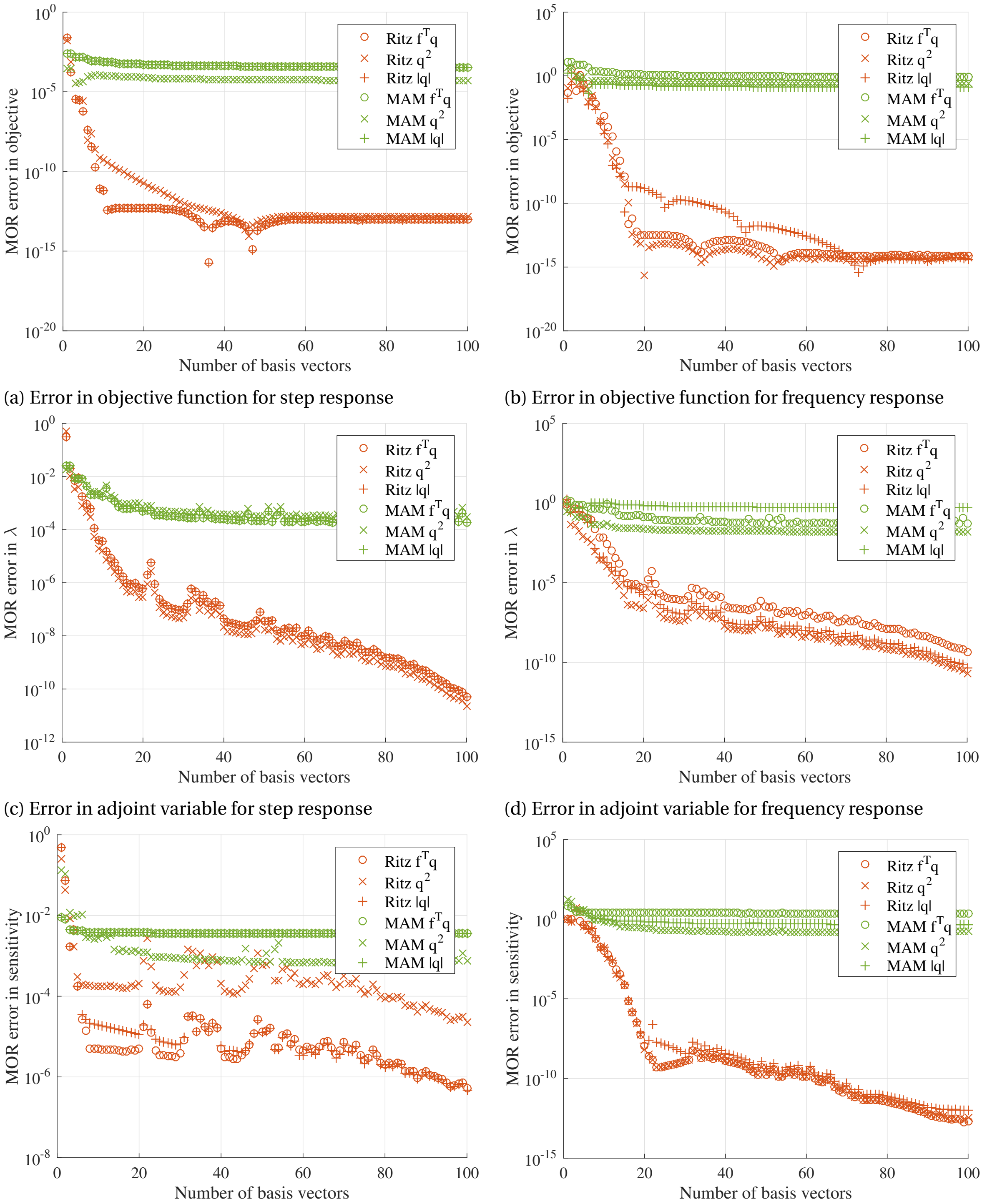
For all combinations of the six test structures, reduction methods, both load cases, and all three objective functions these errors were investigated as a function of the number of basis vectors. To keep this section compact, only the data for the sixth structure is shown, the black-white design with 2560 elements. Using any of the other structures generally results in smaller MOR errors.

Furthermore, in the previous section the modal superposition method was shown to lead to large errors in the dynamic response. The errors in the sensitivities are even larger. When using a large basis (100 eigenvectors), the sensitivity error is still 0.4, which is two orders of magnitude larger than the error found using the MAM. Therefore the modal superposition method is clearly less suitable than Ritz vectors or the MAM, so only the results of the latter two methods are shown, see Fig. 5.11.

The number of basis vectors n is clearly a critical parameter for achieving a low MOR error, especially for Ritz vectors. Additionally, the influence of the number of time steps was also studied, see Fig. 5.12.

From the results, the following conclusions can be drawn:

- The Ritz vector method in most cases results in significantly smaller errors than the MAM.
- As seen before, the MOR error of the MAM is dependent on the type of load: the results of the step response are much better than those of the frequency response.
- In general, a larger number of Ritz vectors results in a smaller MOR error. The convergence for the objective function is faster than for the dynamic response. The adjoint variable converges slower, and the sensitivity even slower.
- The convergence of the adjoint variable, and especially of the sensitivity, is not monotonic. In some cases a larger basis actually causes larger errors. This appears to be a numerical issue, for a structure with better conditioned matrices (for example structure 4) this does not occur.
- The objective function is an important factor in the convergence of the sensitivity. The *squared displacement* objective leads to the largest MOR error. This is caused by the term $\frac{\partial \Phi}{\partial \mathbf{q}}$. For this objective, this term is highly dependent on $\mathbf{q}$, thus also on the MOR error in the response. For the dynamic compliance on the other hand, $\frac{\partial \Phi}{\partial \mathbf{q}}$ is not dependent on $\mathbf{q}$, resulting in a smaller MOR error.
- The number of time steps is an important parameter. When this is increased, the MOR error in the sensitivities becomes smaller. However the number of basis vectors is a more significant parameter.



(a) Error in objective function for step response

(b) Error in objective function for frequency response

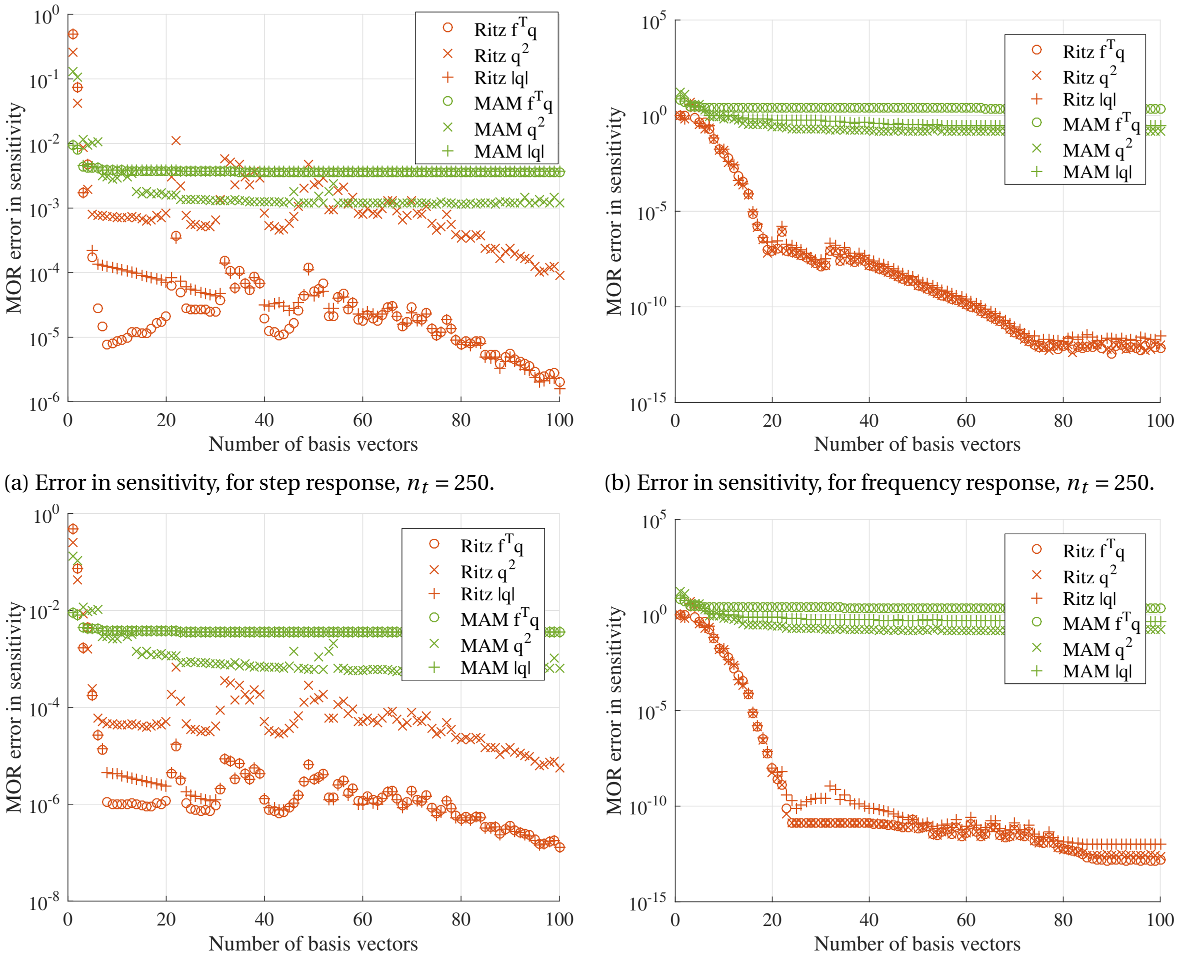
(c) Error in adjoint variable for step response

(d) Error in adjoint variable for frequency response

(e) Error in sensitivity for step response

(f) Error in sensitivity for frequency response

Figure 5.11: Error in objective function versus number of basis vectors, for various objective functions. The analysis uses 500 timesteps. The results show that the while objective function converges quickly, the error in the adjoint variable and especially the design sensitivity is significantly larger. Furthermore, the choice of objective function has a large effect on the accuracy. For Ritz vectors the quadratic displacement function q^2 leads to larger errors than the dynamic compliance $\mathbf{f}^T\mathbf{q}$ or absolute displacement ($|q|$) functions.



(a) Error in sensitivity, for step response, $n_t = 250$.

(b) Error in sensitivity, for frequency response, $n_t = 250$.

(c) Error in sensitivity, for step response, $n_t = 1000$.

(d) Error in sensitivity, for step response, $n_t = 1000$.

Figure 5.12: Numerical examples: MOR error in sensitivity analysis, for different number of timesteps. The data shows that number of time steps is a factor in the MOR error in the sensitivities. Increasing the number of time steps decreases the error, however the effect of the number of basis vectors is larger.

5.5.2. Influence of basis vector sensitivities

Up to this point, in the sensitivity analysis both basis vector sensitivities and all terms containing MOR residuals have been neglected. Still, the resulting MOR errors in the sensitivities have been found to be relatively small. According to literature [40] the sensitivities found using the MAM are sufficiently close to the full model for efficient optimization. The Ritz vector method has been found to result in even smaller errors. This suggests that the simple MOR implemented so far, where both residual terms and basis vectors sensitivities are neglected, is sufficient for optimization purposes.

To investigate this further, Equation (4.24) will be evaluated, which was found to be the main contribution of the basis vector sensitivities to the design sensitivity:

$$\Psi'_{V'} = \frac{\partial \Psi}{\partial \mathbf{q}} \mathbf{V}' \mathbf{z} + \int_0^T (\mathbf{V}' \boldsymbol{\theta})^{\mathrm{T}} \mathbf{f} \mathrm{d}t. \tag{5.5}$$

All terms are known, except for $\mathbf{V}'$. As discussed in Section 4.2, this is expensive to calculate. A finite difference approximation was used for this research, with a step size of $\Delta x = 1 \times 10^{-6}$.[2] To study the contribution of $\Psi'_{V'}$,

[2] The accuracy of finite difference sensitivities can be highly dependent on the size of the disturbance used. When the exact sensitivity is not known, the accuracy can be estimated by computing the sensitivity using various step sizes. In the range $1 \times 10^{-7} < \Delta x < 1 \times 10^{-5}$, the largest relative variation in $\mathbf{V}'$ is less than 2%.

it was calculated for the third test structure, using both load cases and all three objective functions. Analogous to the earlier error analysis, the following error measure is used:

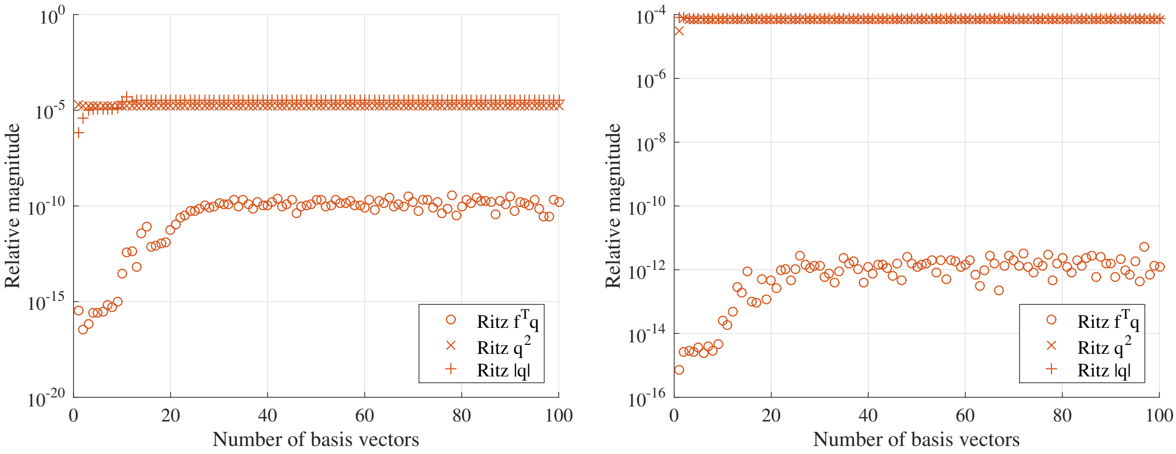
$$\frac{\|\Psi'_{V'}\|}{\|\Psi'_{\text{full}}\|}\,. \tag{5.6}$$

This is not evaluated for the modal acceleration method, since the analytical derivation of Eq. (5.5) did not include the quasi-static correction. The results for Ritz vectors are shown in Fig. 5.13. From these, the following conclusions can be drawn:

- The contribution of $\Psi'_{V'}$ to the total sensitivity is relatively small. For the step response, it should theoretically be zero for the objective function $\Psi = \mathbf{f}^{\mathrm{T}}\mathbf{q}$, since this is the special case discussed in Section 4.1. In the numerical example this results in a magnitude of $\Psi'_{V'}$ that is several orders orders of magnitude smaller than for the other objective functions.
- For the frequency response, the results are similar.
- This is still larger than the error measured between the full and reduced model. The discrepancy is likely caused by errors in the terms of Eq. (5.5), most of these terms are not exactly known.

Summarizing, the error in the sensitivities between the full and reduced model is caused by both the residual and $\mathbf{V}'$. In the shown numerical example it has not been possible to distinguish between the contribution of these terms. Furthermore, although it is expected that $\Psi'_{V'}$ converges to zero for an increasingly accurate reduced model, this cannot be shown using this data.

However, the total MOR error in the sensitivities is still relatively small when using Ritz vectors. Therefore, the assumption that both the MOR residual and the basis vector sensitivities can be neglected is valid for these numerical examples.



(a) Contribution of eigenvector sensitivities for step response.

(b) Contribution of eigenvector sensitivities for frequency response.

Figure 5.13: Magnitude of $\frac{\|\Psi'_{V'}\|}{\|\Psi'_{\text{full}}\|}$ versus number of basis vectors. As expected, this is smaller for the minimum compliance objective. For all cases, the magnitude is small compared to the design sensitivities. A detailed study is not possible using this data, there are too many errors in the term of $\Psi'_{V'}$. Still, combined with the data shown in Fig. 5.12, it can be concluded that in the sensitivity analysis for a model using Ritz vectors, basis vector sensitivities can be neglected.

5.6. UPDATING OF BASIS VECTORS

As part of the optimization process, the structure will be changed. In Section 2.5.6 a method was proposed to calculate new basis vectors using the preconditioned conjugate gradient method. To test this method an experiment was performed, consisting of the following steps:

- Create random vector $\mathbf{x}_0$, $0 < x_i < 1$ to $\mathbf{x}_0$.
- Create disturbed vector $\mathbf{x}_1$ by adding a random vector $0 < \Delta x_i < 0.01$.
- Run finite element method for both to find $\mathbf{K}_0$ and $\mathbf{K}_1$. Reorder using approximate minimum degree algorithm (MATLAB: *symamd*). Define force $\mathbf{f}$, unit amplitude, acting on top right corner, vertical.
- Factorize (Cholesky) $\mathbf{K}_0$.
- Solve $\mathbf{K}_0\mathbf{q}_0 = \mathbf{f}$ to find $\mathbf{q}_0$.
- For reference solve $\mathbf{K}_1\mathbf{q}_1 = \mathbf{f}$ using the direct method.
- Using the Cholesky factor of $\mathbf{K}_0$ as preconditioner and $\mathbf{q}_0$ as starting vector, run preconditioned conjugate gradient algorithm to find $\mathbf{q}_1$.

As mentioned before, the PCG algorithm avoids the factorization of $\mathbf{K}$ at the cost of a number of solutions of linear systems. This method was tested on different 3 problems, the results are summarized in Table 5.3.

Table 5.3: Computation time for updating of basis vectors. All times are in seconds and based on the average of a minimum of 10 runs.

Nr. of elements Nr. of DoFs	1000 2080	10,000 20,400	100,000 200,800
Nr. of PCG iter	4	5	4
CPU-time (s) for:			
PCG	0.0123	0.105	1.75
Factorization	0.0047	0.066	1.33
Direct solution	0.0013	0.024	0.681

Based on these results this method is not effective. The point of the PCG is avoiding factorizations, but in this implementation even one PCG solve is more expensive than a factorization. Compared to a direct solve, it is 4-10 times slower, which makes sense, every iteration equation Eq. (2.59) has to be solved twice, making it more expensive. By lowering the tolerance of the algorithm, the costs could be reduced. But they will still be higher than the cost of a direct solution.

For the same systems two other preconditioners were also tested: incomplete Cholesky factorization (with zero fill) and Jacobi (diagonal). Both result in faster iterations, the speed-up is approximately a factor two for incomplete Cholesky, a factor 10 for Jacobi. But the number of iterations required for convergence rises to respectively >100 and >200, making the total solution more expensive, even if a lower tolerance is used.

This does not mean that updating basis vectors using PCG is not effective in every case. Comparing algorithms is highly dependent on programming language and implementation. Amir and Sigmund [37] report a much better performance. They also suggest finding a different preconditioner to reduce the cost even further. However, for systems of fewer then 100,000 elements, we can conclude that in MATLAB the described method of updating basis vectors is ineffective. It can currently only be recommended if the step of factorization is the dominant cost.

For larger systems, this is likely. The PCG method is also more suitable for parallelization, and does not require assembly of the full system matrix. Further improvements can be found in using more advanced algorithms, such as multi-grid linear solvers [54].

5.7. Summary & Discussion

In this chapter model order reduction has been studied in the context of solving the equations of motion and sensitivity analysis. Six test structures were defined, to represent possible designs found during the optimization process. The issue of local modes was treated briefly, for the chosen penalization method these are not an issue.

Model order reduction was tested on the six test structures for various loads. It was found that the dynamic response of the structure can be found with high accuracy and low costs, especially for the Ritz vector method. The modal superposition method leads to large errors, the modal acceleration method reduces this for simulation with low-frequency excitation. The effect of model order reduction on the objective function, adjoint variable and sensitivity analysis was also verified. The results show that the MOR error in the sensitivities can be significantly larger than the error in the dynamic response. The Ritz vector method resulted in smaller errors than the other methods. These methods will be tested on optimization problems in the next chapter.

Finally, the basis vector updating scheme discussed in Chapter 2 was found to be ineffective for the examples used in this research. This could be improved by using another preconditioner.

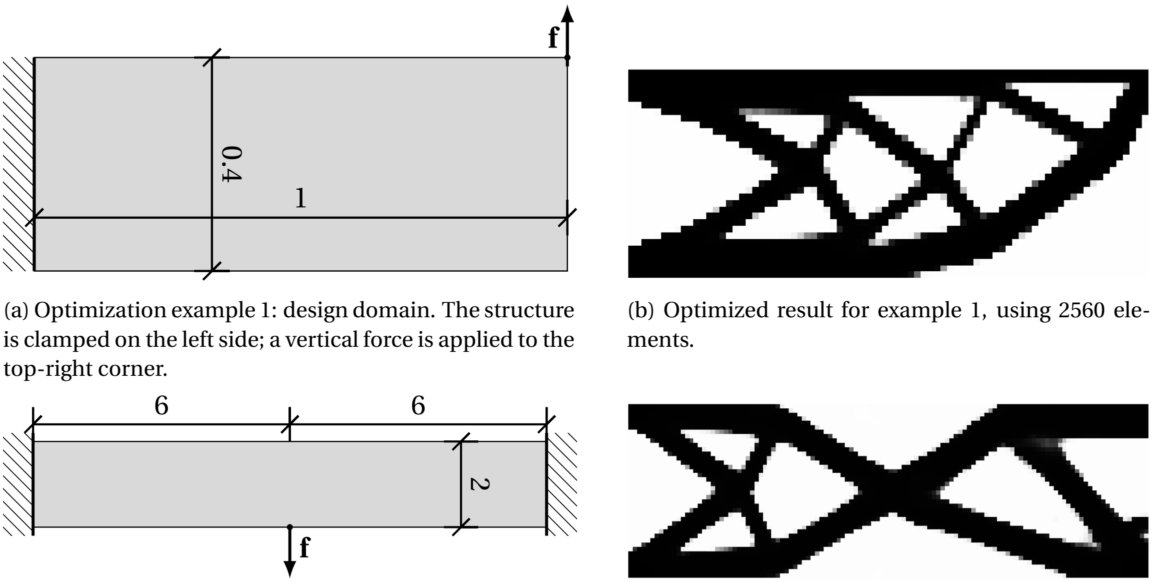
6

Numerical Examples: Optimization

In the previous chapter, the properties of model order reduction have been studied. This will be now be applied to optimization problems. The main part of this chapter studies the application of model order reduction with a fixed number of basis vectors, using two example problems. The remaining part concerns the adaptive model order reduction algorithm, alternative objective functions, and finally a case where penalization is not effective.

6.1. Test cases

Two numerical examples were defined, these are shown in Fig. 6.1. For a full list of parameters see Appendix D.The first numerical example consist of a structure very similar to structure 4-6 studied in the previous chapter. This has 2560 elements, with zero initial deformation and velocity. For $t \geq 0$ a vertical force of 1 kN is applied to the top-right corner. The simulation time is 0.01 s, which is divided into 500 time steps. The second example was taken from the paper by Zhao and Wang [40], this allows a comparison between their results and the outcome of this research. Only the left half of this structure is modelled, this consists of 4800 elements. The simulation time is 0.1 s, divided into 500 time steps. The applied force is again vertical, given by a cosine with a period of 0.2 s and an amplitude of 1 kN. Initially, both examples will be optimized using the *dynamic compliance* objective function defined in Section 3.3, $\Psi = \int_0^T \mathbf{f}^\mathrm{T}\mathbf{q}\,\mathrm{d}t$. In Section 6.3.1 the influence of the objective function will be examined.



(a) Optimization example 1: design domain. The structure is clamped on the left side; a vertical force is applied to the top-right corner.

(b) Optimized result for example 1, using 2560 elements.

(c) Optimization example 2: design domain. Because the structure and load are symmetric, only the left half is modelled.

(d) Optimized result for example 2 (left half), using 4800 elements.

Figure 6.1: Description of optimization test cases. All distances are in meters. The external force on the first example is described by a step function, on the second structure it is a low-frequency cosine.

6.1.1. Comparing optimization performance

There are multiple criteria that determine the performance of an optimization. The most important ones are the optimality of the final design, and the CPU-time required to find it. Especially the CPU-time is highly dependent on the criteria used to determine whether the optimization is converged.

To reliably compare the performance of different optimizations, two tests are performed. First, the optimization is run for a relatively small amount of iterations (25), after which the objective function and CPU-time of the different methods are compared. Secondly, the optimization runs until converged (maximum change in design variable $x_i < 0.01$). These results allow a comparison of the calculated optimized designs.

Furthermore, to compare two designs, only the value of the objective function is shown. This was chosen because most results are visually very similar (illustrated by Figs. 6.2b to 6.2d) and thus hard to compare. The value of the objective is a much more reliable

6.2. Optimization & Model Order Reduction

6.2.1. The modal acceleration method and Ritz vectors

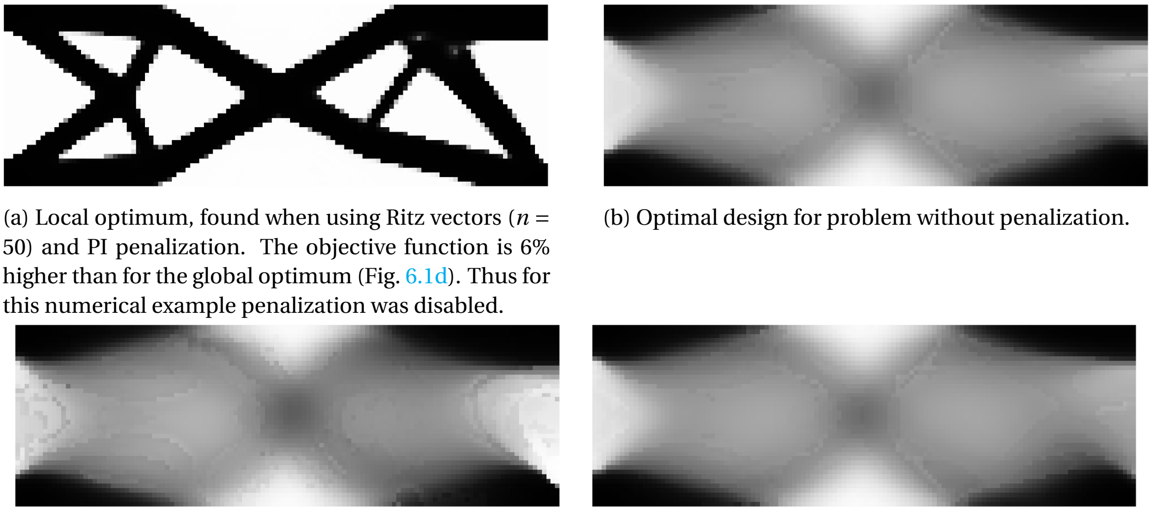
In the previous chapter it was found that the modal superposition method leads to large MOR errors, the effect of this on optimization will be tested in the next section. The other two implemented reduction methods, Ritz vectors and the modal acceleration method, resulted in much small MOR errors. The main question is however what effect these errors have on the optimization. To investigate this both test cases were optimized, using model order reduction with a fixed number of basis vectors.

For the second test case, comparing optimizations proved to be more difficult, because of local optima (illustrated by Fig. 6.2a). To solve this, penalization was disabled. This makes the design space more convex, as a result, all tested optimizations converged towards the same optimum (Fig. 6.2b). Furthermore, Figs. 6.2c and 6.2d show two optimal designs found using model order reduction. The aim of these figures is to illustrate the difference between the found optima: an error of 0.46 is clearly visible, while an error of 2.7×10^{-3} is not.

Before examining the convergence of the various optimizations, we will first look into the computational costs in more detail. The CPU-time of the different parts of the optimization is shown in Table 6.1. In the full optimization, the Newmark integration is clearly the most expensive step. The various reduction methods decrease the cost of this step dramatically. Furthermore, some of the pitfalls of comparing CPU-time can be seen here. First of all, the costs of time-integration for the simulations with model order reduction should scale approximately quadratically with the number of basis vectors, but the measured effect is much smaller. The implementation of the Newmark algorithm contains a loop, this causes extra computation time which does not scale with the size of the system.

Secondly, the CPU-time for computing eigenvectors is less then expected when compared to that of computing Ritz vectors. MATLAB uses a highly optimized Fortran library to compute eigenvectors, while the Ritz vectors are computed using a much slower MATLAB code. Even with these drawbacks, the table does illustrate the effect of model order reduction on the CPU-time.

Using MOR allows reduced memory requirements. As discussed frequently in this thesis, for transient sensitivity analysis the response of the system has to be stored. For a full system this uses $3 \cdot N \cdot n_\mathrm{t}$ variables. The reduced system needs $N \cdot n$ variables for storing the basis vectors, and $3 \cdot n \cdot n_\mathrm{t}$ bytes for storing reduced coordinates. Furthermore, the full system matrices are only required for computing the basis vectors, and can be replaced by the much smaller reduced matrices. This is slightly different for the modal acceleration method, since this requires both the full stiffness matrix and the displacement in full coordinates. The resulting memory use is shown in Table 6.2. The modal acceleration method reduces memory requirements by a factor three, but using Ritz vector results in a factor 30 reduction in memory use, even for a relatively large number of basis vectors.



(a) Local optimum, found when using Ritz vectors (n = 50) and PI penalization. The objective function is 6% higher than for the global optimum (Fig. 6.1d). Thus for this numerical example penalization was disabled.

(b) Optimal design for problem without penalization.

(c) Optimal design found using MAM with $n = 5$. The error in the objective function is 0.46, the structure is clearly different from the optimal design found using the full model.

(d) Optimal design found using MAM with $n = 25$. The error in the objective is 2.7×10^{-3}, which is large compared to for example the results found using Ritz vectors. Still the structure are visually nearly identical.

Figure 6.2: Various optimized designs for numerical example 2. A local optimum is shown, as well as various designs optimized without penalization. These are visually nearly identical, therefore no further designs will be shown. To compare optimizations, the value of the objective function will be given.

Table 6.1: Overview of CPU-time in seconds for numerical example 1, for 25 iterations. Three different optimizations are shown: using the full model, MAM with small basis, Ritz vectors with medium basis size. The aim is to illustrate the costs of the various steps, data for the other optimizations is similar and thus omitted. The time listed for the Newmark algorithm includes the numerical integration of both the equations of motion and the adjoint equation. In the full simulation, this step is clearly the dominant cost. Model order reduction reduces this significantly. The added cost of computing basis vectors is small.

	Full	MAM $n = 5$	Ritz $n = 25$
Total	143	29.4	29.3
Newmark	125	5.82	5.88
Basis vectors	-	3.20	4.06
MAM correction	-	1.56	-
Sensitivity analysis	10.7	11.6	11.5
FEM	6.69	6.65	6.76

Table 6.2: Memory use for optimization with model order reduction (n = 50). All data is in millions of bytes. For these examples, the MAM results in a factor 3 lower memory requirements, and Ritz vectors in a factor 30. This will be even larger when fewer basis vectors are used, the memory use scale approximately linearly with the size of the basis.

	Example 1			Example 2		
Variable	Full	MAM	Ritz	Full	MAM	Ritz
System matrices	1.20	0.46	.06	5.70	1.96	0.06
Response	25.2	9.00	0.60	117.6	39.8	0.60
Basis vectors	-	0.84	0.84	-	3.92	3.92
Total	26.4	10.3	1.51	123.3	45	4.58

The next step is comparing the convergence of the various methods. The results are listed in Tables 6.3 and 6.4. From these, the following conclusions can be drawn:

- The goal of the proposed method is to approximate the results of the full model at lower costs. Therefore, the difference between the objective functions of the designs found using the full and reduced model is important, this should be as small as possible. A large negative value implies that the reduced model converges to a better design than the full model, this is also undesired.
- All MOR methods results in large decreases of the CPU-time. For the same number of basis vectors, the Ritz vector method is slightly more efficient than the MAM.
- The MAM with 5 basis vectors in some cases does not converge to the optimal solution. This is contributed to its low accuracy. The MAM with 25 or 50 basis vectors converges similar to the full model.
- Even when using just 5 basis vectors, the optimization using Ritz vectors converges similar to the full solution. Furthermore, the differences in the objective function are smaller than for the MAM.
- The results are similar to those reported by Zhao and Wang [40]. A big discrepancy however was found in the reported CPU-time per iteration. Their results show a maximum factor 2 speed-up for the MAM with 40 basis vectors, while on the same problem in this research a factor 5 or higher was achieved. This is even more surprising because they use Duhamel's integral to solve the equations of motion, which is theoretically faster than Newmark integration. The difference is most likely caused by a more efficient implementation. Therefore the MAM can be significantly more efficient than suggested in literature.

Since both problems concern a low-frequency load, these results should be compared to Fig. 5.11e. This suggests the following relation between the MOR error in the sensitivities and the convergence of the optimization. When the sensitivities are very inaccurate (MSM, or MAM with 5 basis vectors), the convergence will be hindered. All other results show convergence similar to the optimization using the full Newmark method. Apparently, other errors in the optimization loop (most likely the MMA approximation error) dominate the convergence, so decreasing the MOR error does not improve it.

Based on the results shown here, the conclusion can be drawn that relatively coarse model order reduction is the most efficient for topology optimization. Furthermore, Ritz vectors are shown to be more efficient than the modal acceleration method.

6.2.2. THE MODAL SUPERPOSITION METHOD

The results in the previous chapter indicated that the modal superposition method results in large errors for both the dynamic response and the sensitivities. To see how this affects the optimization, example 1 was optimized until convergence using a (relatively large) basis of 100 eigenvectors. The result is shown in Fig. 6.3.

The optimized design does not have any material in the top-right corner, where the force is applied. Since this is obviously not optimal, it seems like the optimization is exploiting the errors in the model, possibly by making the eigenvectors orthogonal to the load. While this does lead to a low objective according to the reduced model, it results in a suboptimal design. Using Ritz vectors or MAM avoids this issue, since the load is directly used in both methods.

Table 6.3: Numerical examples: comparison of optimization and the effect of MOR. The results after 25 iterations are shown, all times are divided by the CPU-time of the full optimization (respectively 143 s and 429 s). Compared to the MAM, the Ritz vector method leads to designs that are closer to the full design, at lower computational cost.

		MAM			Ritz vectors		
Reduction	Full	$n = 5$	$n = 25.$	$n = 50$	$n = 5$	$n = 25$	$n = 50$
Example 1:							
CPU time	1	0.202	0.239	0.295	0.174	0.202	0.238
Simulation time	0.873	0.073	0.103	0.150	0.047	0.069	0.100
Error in obj.	0	2.2×10^{-4}	-1.8×10^{-5}	-1.6×10^{-5}	5.2×10^{-8}	-5.8×10^{-8}	-2.6×10^{-8}
Example 2:							
CPU time	1	0.120	0.141	0.238	0.115	0.138	0.187
Simulation time	0.921	0.058	0.076	0.133	0.048	0.070	0.113
Error in obj.	0	1.9×10^{-2}	2.4×10^{-4}	9.2×10^{-4}	1.3×10^{-6}	-7.5×10^{-7}	-1.3×10^{-6}

Table 6.4: Numerical examples: comparison of optimization and the effect of MOR. The optimization is converged ($\max_i \Delta x_i = 0.01$), all times are divided by the CPU-time per iteration of the full optimization (respectively 5.85 s and 15.8 s). Similar to the previous results, the Ritz vectors method shows nearly identical convergence to the full method. The MAM leads to significant differences, especially when using a small number of basis vectors.

		MAM			Ritz vectors		
Reduction	Full	$n = 5$	$n = 25.$	$n = 50$	$n = 5$	$n = 25$	$n = 50$
Example 1:							
Nr. of iterations	152	183	162	182	152	152	153
CPU-time per iter.	1	0.188	0.233	0.278	0.159	0.183	0.218
Error in obj.	0	-6.4×10^{-4}	-3.6×10^{-4}	-5.8×10^{-4}	-3.4×10^{-7}	-1.2×10^{-5}	-5.9×10^{-6}
Example 2:							
Nr. of iterations	27	256	27	50	27	27	27
CPU-time per iter.	1	0.112	0.162	0.186	0.125	0.148	0.181
Error in obj.	0	.46	2.7×10^{-3}	-5.0×10^{-4}	1.9×10^{-6}	-2.0×10^{-6}	2.3×10^{-7}

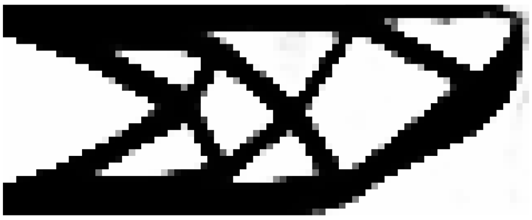


Figure 6.3: Structure optimized using modal superposition method with 100 basis vectors. The force is applied to the top-right corner, note that the optimization removes material around that area. The optimization was run for 500 iterations, but was still not converged. The resulting objective function is 3.15 times higher than the one associated with the optimal design. Total CPU-time is 1.3 times larger than the full optimization.

6.3. Other aspects of optimization and model order reduction

6.3.1. Objective functions

In the previous chapter is was found that the objective function is an important factor in the MOR error in the sensitivity. A test was performed to see how this affects the optimization, see Table 6.5. In Section 5.5 it was found that the *squared displacement* objective resulted in the largest errors, while using the *absolute displacement* and *dynamic compliance* functions yielded significantly smaller errors.

The effect of this is clearly visible in the results. When the number of basis vectors is increased, the optimized designs for the squared displacement converge towards the result of the full model. However, the other two objective functions converge much faster. This effectively means that using the squared displacement function necessitates a larger number of basis vectors to achieve similar convergence.

Table 6.5: Numerical example 1: comparison of different objective functions and the effect of MOR. Data shown is the error in the objective function after 25 iterations. The error between the different objective functions should not be compared, only the trend for a specific objective is relevant. From this data, it is apparent that the choice of objective function affects the convergence with respect to the number of basis vectors. In Section 5.5 it was found that the squared displacement objective leads to less accurate sensitivities, this clearly results in slower convergence with respect to the number of basis vectors.

		MAM		Ritz vectors			
	Full	$n = 25.$	$n = 50$	$n = 5$	$n = 25$	$n = 50$	$n = 100$
$\Phi = q_i^2$	0	4.5×10^{-2}	4.3×10^{-3}	3×10^{-4}	5.4×10^{-3}	-8.1×10^{-5}	-4.5×10^{-6}
$\Phi = \mathbf{f}^{\mathrm{T}}\mathbf{q}$	0	-1.8×10^{-5}	-1.6×10^{-5}	5.2×10^{-8}	-5.8×10^{-8}	-2.6×10^{-8}	-2.7×10^{-8}
$\Phi = \lvert q_i \rvert$	0	-2.2×10^{-3}	3.2×10^{-4}	-1.8×10^{-5}	-3.9×10^{-8}	-3.6×10^{-7}	-3.1×10^{-8}

6.3.2. Optimization & Adaptive Model Order Reduction

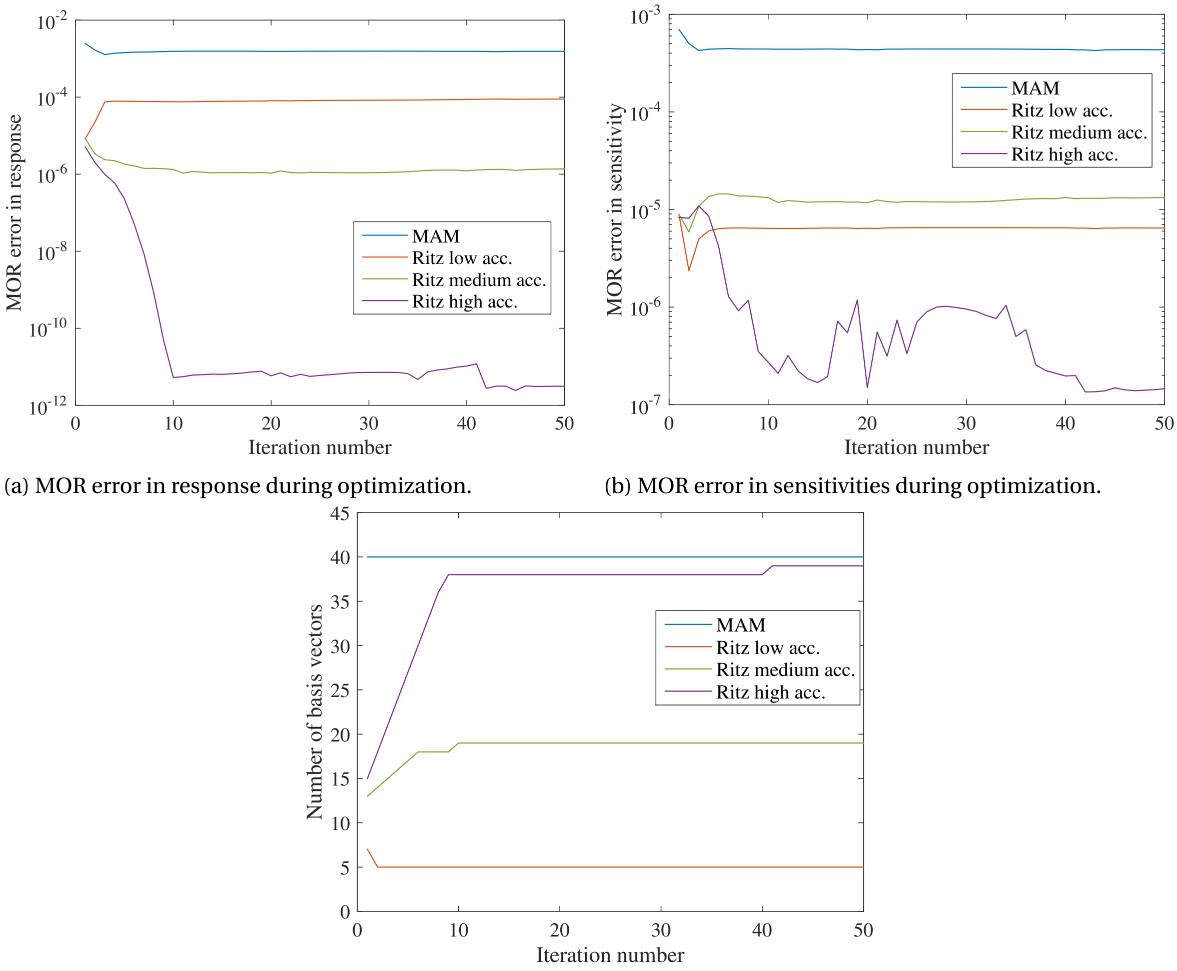
In Section 4.3 two algorithms for adaptive model order reduction were defined. The first of these uses only model data, the second uses response data as well. The aim of both algorithms is to maintain a pre-defined accuracy for the model order reduction. To test this, three accuracy levels are defined. Reference values for the different error estimates were found experimentally (see Appendix D.4 for details).

Only adaptivity in the number of Ritz vectors will be studied. The same methods could be used for the modal superposition method, but since this is not suitable for optimization this will not be investigated. Modal acceleration might be suitable for optimization, but the effect of the quasi-static correction cannot be estimated using the described methods. To test the performance of the algorithm, it was used on optimization example 1, for reference after every iteration the full model was used to calculate the accuracy. The results are shown in Fig. 6.4.

The method is clearly not perfect. The initial estimate for the number of basis vectors is significantly different from the final values, leading to initial large variations in the MOR error. This is partly caused by large changes in the design in the first iterations. But it could still be improved with proper tuning of the parameters. After 10 iterations, the algorithm maintain a nearly constant number of basis vectors, which results in a constant MOR error in the response. However, as seen before in this research, this does not always translate into accurate sensitivities.

Next it will be checked whether the method does result in an optimal design. Using the same three settings for the accuracy, the first numerical example was optimized. The results after 25 iterations and of the converged solution can be found in Tables 6.6 and 6.7. These are similar to the results for a fixed number of Ritz vectors: even for low accuracy, the convergence is similar to that of the full model.

Summarizing, the results show that the adaptive MOR method is effective. While there is no clear performance benefit over using a fixed number of basis vectors, it is still an important step towards an automatic implementation of model order reduction, for example as part of a finite element and optimization software package.



(a) MOR error in response during optimization.

(b) MOR error in sensitivities during optimization.

(c) Number of basis vectors during optimization.

Figure 6.4: Test of AMOR algorithms in optimization loop. The tested problem is optimization example 1. The initial estimate for the number of basis vector is different than the final value, leading to a non-constant accuracy in the first iterations. But Algorithm 4.2 is effective: the final accuracy of the response is constant. However, this does not guarantee accurate sensitivities.

Table 6.6: Results for topology optimization with adaptive model order reduction.

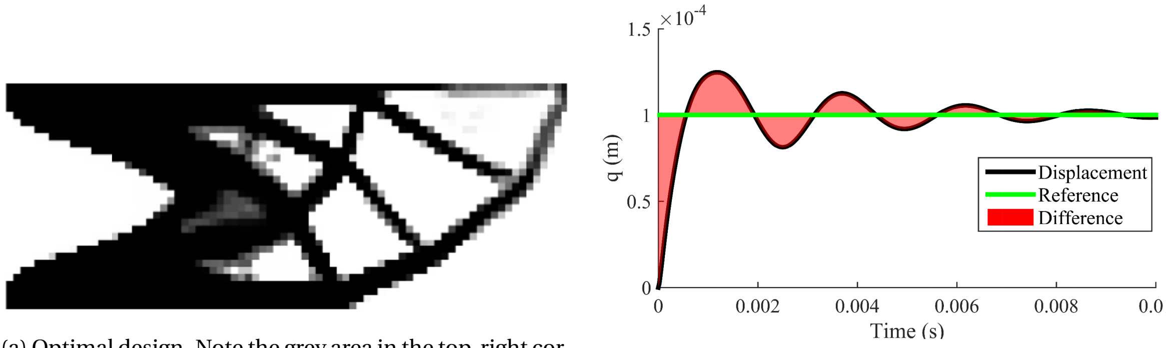
Accuracy:	Full	Low	Medium	High
Average nr. of basis vectors	-	5	7.9	46.4
CPU-time	1	0.179	0.180	0.231
Error in obj.	0	5×10^{-8}	-3.7×10^{-7}	-1.34×10^{-5}

Table 6.7: Results for topology optimization with adaptive model order reduction.

Accuracy:	Full	Low	Medium	High
Average nr. of basis vectors	-	5	8	58.8
Nr. of iterations	152	152	152	150
Error in obj.	0	-3.4×10^{-7}	3.6×10^{-9}	-3.6×10^{-5}

6.3.3. Penalization and transient problems

Up to this point using PI penalization has led to black-white designs. However, as discussed in Section 2.2.1, this might not be true for all transient problems. An example of this can be found by studying a problem similar to example 1. But instead of minimizing the displacement, the displacement with respect to a predecribed curve will be minimized. As reference curve a constant value is used. This is chosen to be approximately twice the value of the static displacement of the previously found optimized structure (Fig. 6.1b). Thus the goal of the optimization is no longer to minimize displacement. A lower stiffness does not necessarily result in a higher objective function, which is the assumption behind the SIMP/RAMP/PI methods.



(a) Optimal design. Note the grey area in the top-right corner, penalization fails to remove all grey areas in this case.

(b) Response of optimal design versus time. The reference curve is also drawn, the optimization attempts to minimize the difference (red area) between the response and the reference.

Figure 6.5: Result for optimization problem with non-zero reference curve. The used penalization method is shown to be ineffective for this example.

Therefore for some optimization problems alternative penalizations methods will have to be used, for example the *material penalization* method proposed by van de Ven *et al.* [4]. But in most cases the PI penalization method will be sufficient.

6.4. Summary & Discussion

In the previous chapter it was found that using model order reduction can significantly reduce the costs of simulation and sensitivity analysis for topology optimization. This does lead to errors in the sensitivities. The aim of this chapter was to see the effect of these errors on the complete optimization process.

The results show that very inaccurate sensitivities, such as those found by the MSM or the MAM with a small basis, will lead to suboptimal designs. When the sensitivities are more accurate the convergence of the reduced model is nearly identical to that of the full model. The most efficient method was found to be model order reduction using a small basis of Ritz vectors.

Two important conclusions can be drawn when these results are compared to literature. The first is that the MAM is more efficient than reported in previous research. The second is that the Ritz vector method can be used to lower computational costs even farther.

7

Conclusions & Recommendations

7.1. Conclusions

At the start of this project, two research questions were defined. These were: can model order reduction be applied to increase the efficiency of topology optimization for transient, linear structural problems; and can this be used in engineering practice. The answer to both questions is yes, in most cases applying model reduction is possible, and this can increase the efficiency of the optimization. In more detail, the following conclusions are drawn:

- By applying model order reduction, the costs of topology optimization for time domain structural dynamics can be significantly decreased.
    - The increase in computational speed is dependent on parameters, most notably the number of time steps and the required number of basis vectors.
    - For the numerical examples included in this research, a speed-up of a factor 20 was shown for the time integration. This results in a factor 8 speed-up for the total optimization. The computational efficiency increases with the size of the system, as well as with the number of time steps in the time integration.
    - Applying model order reduction reduces the memory requirements. In the numerical examples a factor 30 decrease in memory use was shown. The relative decrease in memory is approximately equal to the number of DoFs in the full system divided by the number of basis vectors, thus it will also be even more beneficial for larger systems.
    - Using a reduced basis consisting of Ritz vectors is in most cases more efficient than the modal superposition or modal acceleration method.
- The performance of the optimization is dependent on the error caused by the model order reduction.
    - A large MOR error results in poor convergence. This was shown for the modal superposition method, and in some cases for the modal acceleration method when the number of basis vectors is small.
    - Decreasing the error below a certain level does not increase the convergence rate of the optimization. For the shown examples this corresponds to using as little as five Ritz vectors.
    - When compared to the modal acceleration method, the Ritz vector method is cheaper, and results in smaller errors with respect to the full model.
    - For MOR using Ritz vectors, neglecting basis vector sensitivities does not lead to a significant error.
    - An algorithm is proposed for determining the optimal number of basis vectors. This does not improve the performance of the optimization, but it is helpful in automating the process.
- Model order reduction can be applied to most optimization problems involving linear transient dynamics. For this to be efficient, the requirements are:
    - The number of time steps must be significantly larger than the required number of basis vectors.
    - The applied load should consist of a constant spatial term, possibly multiplied by a non-constant temporal term. So the force should be of the form $\mathbf{f} = \mathbf{L} f_t(t)$. This is a necessary condition for both Ritz vectors and the modal acceleration method. A combination of a small number of these forces is also possible.

  - The same requirement applies to the derivative $\frac{\partial \Phi}{\partial \mathbf{q}}$, this should also be a constant spatial term possibly multiplied by a non-constant scalar.
  - The choice of objective function influences the error in the sensitivities. The error will be larger when the derivative $\frac{\partial \Phi}{\partial \mathbf{q}}$ is highly dependent on **q**.

Concluding, model order reduction is a promising tool in topology optimization for transient structural dynamics, it can be used to significantly reduce the computational costs. This will bring successful industrial applications one step closer.

7.2. RECOMMENDATIONS

The proposed method can be improved in many ways, further research on the following topics is recommended:

- Using results from previous iterations to make the model order reduction more efficient. Possibilities are:
  - Updating or reusing of basis vectors instead of recomputing them. The method described in Section 2.5.6 can be improved.
  - Calculating a set of basis vectors from the dynamic response of the previous iteration, for example using Proper Orthogonal Decomposition.
- Detailed research into the link between penalization, conditioning of the system matrices, and accuracy of the Ritz vector method.
- Investigate whether the adjoint variable can be computed cheaply when using the dynamic compliance objective function.
- Combination of model order reduction and additional constraints, such as on maximum displacement or stress.
- Investigate adaptive model order reduction with varying accuracy, for example with higher accuracy in the final iterations to ensure optimality.
- Extension to non-linear structural mechanics. This will have many useful applications, for example in designing structures for impact response.

A

Bibliography

[1] J. D. Deaton and R. V. Grandhi, *A survey of structural and multidisciplinary continuum topology optimization: post 2000,* Structural and Multidisciplinary Optimization **49**, 1 (2013).

[2] J. Zhu, W. Zhang, and P. Beckers, *Integrated layout design of multi-component system,* International Journal for Numerical Methods in Engineering **78**, 631 (2009).

[3] N. L. Pedersen, *Maximization of eigenvalues using topology optimization,* Structural and multidisciplinary optimization **20**, 2 (2000).

[4] E. van de Ven, E. Hooijkamp, M. Langelaar, and F. van Keulen, *Topology optimization of a transient thermo-mechanical problem using material penalization,* 11th World Congress on Structural and Multidisciplinary Optimization .

[5] N. Aage, M. Nobel-Jørgensen, C. Andreasen, and O. Sigmund, *Interactive topology optimization on handheld devices,* Structural and Multidisciplinary Optimization **47**, 1 (2013).

[6] O. Sigmund and J. Petersson, *Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima,* Structural Optimization **16**, 68 (1998).

[7] M. P. Bendsoe and O. Sigmund, *Topology optimization: theory, methods and applications* (Spinger, 2003).

[8] K. Svanberg, *The method of moving asymptotes- a new method for structural optimization,* International journal for numerical methods in engineering **24**, 359 (1987).

[9] K.-J. Bathe, *Finite element procedures* (Prentice Hall, 1996).

[10] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis* (Courier Corporation, 2000).

[11] K. K. Choi and N.-H. Kim, *Structural sensitivity analysis and optimization 1: linear systems* (Springer, 2006).

[12] B.-S. Kang, G.-J. Park, and J. S. Arora, *A review of optimization of structures subjected to transient loads,* Structural and Multidisciplinary Optimization **31**, 81 (2006).

[13] F. van Keulen, R. T. Haftka, and N. H. Kim, *Review of options for structural design sensitivity analysis. part 1: Linear systems,* Computer Methods in Applied Mechanics and Engineering **194**, 3213 (2005).

[14] E. v. d. Ven, *Topology optimization for transient-thermal-mechanical systems (msc. thesis),* (2014).

[15] G. N. Vanderplaats, *Comment on" methods of design sensitivity analysis in structural optimization",* AIAA Journal **18**, 1406 (1980).

[16] A. D. Belegundu and J. S. Arora, *A sensitivity interpretation of adjoint variables in optimal design,* Computer methods in applied mechanics and engineering **48**, 81 (1985).

[17] G. H. Yoon, *Structural topology optimization for frequency response problem using model reduction schemes,* Computer Methods in Applied Mechanics and Engineering **199**, 1744 (2010).

[18] B. Besselink, U. Tabak, A. Lutowska, N. van de Wouw, H. Nijmeijer, D. J. Rixen, M. E. Hochstenbach, and W. H. A. Schilders, *A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control,* Journal of Sound and Vibration **332**, 4403 (2013).

[19] C. Gogu, *Improving the efficiency of large scale topology optimization through on-the-fly reduced order model construction,* International Journal for Numerical Methods in Engineering **101**, 281 (2015).

[20] D. J. Rixen, *Engineering Dynamics - WB1418 Lecture notes* (2008).

[21] D. J. Rixen, *Numerical methods in Engineering Dynamics - WB1416 Lecture notes* (2011).

[22] R. D. Cook, D. S. Malkus, and M. E. Plesha, *Concepts and Applications of Finite Element Analysis* (John Wiley & Sons, 2002).

[23] E. L. Wilson, M.-W. Yuan, and J. M. Dickens, *Dynamic analysis by direct superposition of Ritz vectors,* Earthquake Engineering & Structural Dynamics **10**, 813 (1982).

[24] D. C. Lay, *Linear algebra and its applications,* Addison-Wesley Publishing Company **3**, 124 (2003).

[25] P. Léger and E. L. Wilson, *Generation of load dependent ritz transformation vectors in structural dynamics,* Engineering Computations **4**, 309 (1987).

[26] J. Gu, Z.-D. Ma, and G. M. Hulbert, *A new load-dependent Ritz vector method for structural dynamics analyses: quasi-static Ritz vectors,* Finite Elements in Analysis and Design **36**, 261 (2000).

[27] G. H. Yoon, *Toward a multifrequency quasi-static Ritz vector method for frequency-dependent acoustic system application,* International Journal for Numerical Methods in Engineering **89**, 1451 (2012).

[28] R. J. Guyan, *Reduction of stiffness and mass matrices,* AIAA journal **3**, 380 (1965).

[29] Y. C. Liang, H. P. Lee, S. P. Lim, W. Z. Lin, K. H. Lee, and C. G. Wu, *Proper orthogonal decomposition and its applicationspart i: Theory,* Journal of Sound and Vibration **252**, 527 (2002).

[30] J. Dickens, J. Nakagawa, and M. Wittbrodt, *A critique of mode acceleration and modal truncation augmentation methods for modal response analysis,* Computers & Structures **62**, 985 (1997).

[31] D. Rixen, *Generalized mode acceleration methods and modal truncation augmentation,* in *Structures, Structural Dynamics and Material Conference and Exhibit, 42st AIAA/ASME/ASCE/AHS/ASC. AIAA*, Vol. 1300.

[32] D. d. Klerk, D. J. Rixen, and S. Voormeeren, *General framework for dynamic substructuring: history, review and classification of techniques,* AIAA journal **46**, 1169 (2008).

[33] A. Ibrahimbegovic and E. L. Wilson, *Automated truncation of ritz vector basis in modal transformation,* Journal of engineering mechanics **116**, 2506 (1990).

[34] B. Nortier, *Residual Vectors & Error Estimation in Substructure based Model Reduction (MSc. thesis)* (TU Delft, 2011).

[35] J. S. Han, *Calculation of design sensitivity for large-size transient dynamic problems using Krylov subspace-based model order reduction,* Journal of Mechanical Science and Technology **27**, 2789 (2013).

[36] S. Voormeeren, *Dynamic substructuring methodologies for integrated dynamic analysis of wind turbines (PhD. dissertation),* (2012), http://dx.doi.org/10.4233/uuid:f45f0548-d5ec-46aa-be7e-7f1c2b57590d.

[37] O. Amir and O. Sigmund, *On reducing computational effort in topology optimization: how far can we go?* Structural and Multidisciplinary Optimization **44**, 25 (2010).

[38] U. Kirsch, M. Bogomolni, and I. Sheinman, *Efficient structural optimization using reanalysis and sensitivity reanalysis,* Engineering with Computers **23**, 229 (2007).

[39] U. Kirsch, M. Kocvara, and J. Zowe, *Accurate reanalysis of structures by a preconditioned conjugate gradient method,* International Journal for Numerical Methods in Engineering **55**, 233 (2002).

[40] J. Zhao and C. Wang, *Dynamic response topology optimization in the time domain using model reduction method,* Structural and Multidisciplinary Optimization , 1 (2015).

[41] R. L. Fox and M. P. Kapoor, *Structural optimization in the dynamics response regime - a computational approach,* AIAA Journal **8**, 1798 (1970).

[42] W. H. Greene and R. Haftka, *Computational aspects of sensitivity calculations in linear transient structural analysis,* Structural optimization **3**, 176 (1991).

[43] S. Min, N. Kikuchi, Y. C. Park, S. Kim, and S. Chang, *Optimal topology design of structures under dynamic loads,* Structural optimization **17**, 208 (1999).

[44] C. B. W. Pedersen, *Crashworthiness design of transient frame structures using topology optimization,* Computer Methods in Applied Mechanics and Engineering **193**, 653 (2004).

[45] J. S. Han, E. B. Rudnyi, and J. G. Korvink, *Efficient optimization of transient dynamic problems in MEMS devices using model order reduction,* Journal of Micromechanics and Microengineering **15**, 822 (2005).

[46] H. Liu, W. Zhang, and T. Gao, *A comparative study of dynamic analysis methods for structural topology optimization under harmonic force excitations,* Structural and Multidisciplinary Optimization **51**, 1321 (2015).

[47] A. C. Paul, A. Dutta, and C. V. Ramakrishnan, *Accurate computation of design sensitivities for dynamically loaded structures with displacement constraints,* AIAA Journal **34**, 1670 (1996).

[48] J. S. Jensen, P. B. Nakshatrala, and D. A. Tortorelli, *On the consistency of adjoint sensitivity analysis for structural optimization of linear dynamic problems,* Structural and Multidisciplinary Optimization **49**, 831 (2013).

[49] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund, *Efficient topology optimization in MATLAB using 88 lines of code,* Structural and Multidisciplinary Optimization **43**, 1 (2010).

[50] MathWorks, *MATLAB Documentation: Techniques to Improve Performance* (Access Date 04-12-15).

[51] MathWorks, *MATLAB Documentation: symamd,* (Access Date 04-12-2015).

[52] J. R. Gilbert, C. Moler, and R. Schreiber, *SPARSE MATRICES IN MATLAB: DESIGN AND IMPLEMENTATION* (Mathworks, 1991).

[53] T. Lee, *An adjoint variable method for structural design sensitivity analysis of a distinct eigenvalue problem,* KSME International Journal **13**, 470 (1999).

[54] O. Amir, N. Aage, and B. S. Lazarov, *On multigrid-CG for efficient topology optimization,* Structural and Multidisciplinary Optimization **49**, 815 (2013).

[55] O. Sigmund and K. Maute, *Topology optimization approaches,* Structural and Multidisciplinary Optimization (2013), 10.1007/s00158-013-0978-6.

[56] O. Sigmund, *A 99 line topology optimization code written in MATLAB,* Structural and Multidisciplinary Optimization **21**, 120 (2001).

[57] N. P. van Dijk, K. Maute, M. Langelaar, and F. van Keulen, *Level-set methods for structural topology optimization: a review,* Structural and Multidisciplinary Optimization **48**, 437 (2013).

[58] N. M. Patel, D. Tillotson, J. E. Renaud, A. Tovar, and K. Izui, *Comparative study of topology optimization techniques,* AIAA Journal **46**, 1963 (2008).

[59] N. M. Patel, B.-S. Kang, J. E. Renaud, and A. Tovar, *Crashworthiness design using topology optimization,* Journal of Mechanical Design **131**, 061013 (2009).

[60] M. P. Bendsøe and N. Kikuchi, *Generating optimal topologies in structural design using a homogenization method,* Computer methods in applied mechanics and engineering **71**, 197 (1988).

[61] S. Eddins, *MATLAB Documenation: plotting the DTFT using the output of fft, access date 2015-09-15,* (2010).

B

Basis vector sensitivities

In topology optimization adjoint methods are generally preferred over direct methods of sensitivity calculation. This section describes an adjoint method for eigenvector sensitivity and discusses the derivation of a similar method for Ritz vectors.

B.1. Eigenvector sensitivities

For eigenvectors, there are multiple ways of calculating sensitivities. One way is by using the following adjoint formulation [53]:

$$\Psi_{\mathrm{A}} = \Psi(\mathbf{v}_i) + \boldsymbol{\gamma}^T(\mathbf{K} - \omega_i^2\mathbf{M})\mathbf{v}_i + \mu(\mathbf{v}_i^T\mathbf{M}\mathbf{v}_i - 1)\,. \tag{B.1}$$

This leads to the following sensitivity and adjoint equation:

$$\frac{\mathrm{d}\Psi_{\mathrm{A}}}{\mathrm{d}\mathbf{x}} = \boldsymbol{\gamma}^T(\mathbf{K}' - \omega_i^2\mathbf{M})\mathbf{v}_i \tag{B.2}$$

$$\begin{bmatrix} \mathbf{K} - \omega_i^2\mathbf{M}' & -2\mathbf{M}\mathbf{v}_i \\ -2\mathbf{v}_i^T\mathbf{M} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma} \\ \mu \end{bmatrix} = \begin{bmatrix} -\frac{\partial\Psi}{\partial\mathbf{v}_i}^T \\ -\frac{\partial\Psi}{\partial\omega_i^2} \end{bmatrix}\,. \tag{B.3}$$

This approach does require the solution of Eq. (B.3) for every eigenvector sensitivity. Thus for transient sensitivity analysis one has to solve n additional equations. To put this into context, the calculation of the eigenvectors requires 2-4 solutions of a similar linear system per vector.

B.2. Ritz vector sensitivities

Calculating sensitivities of Ritz vectors is much more complex. There was no information found in literature on this subject.

This section will explore the possibilities for direct differentiation and adjoint formulations for Ritz vectors. Direct differentiation is based on differentiation of the constitutive equation, while adjoint formulation are based on adding the constitutive equation multiplied by an adjoint variable to the objective function. The constitutive equations for Ritz vectors are:

$$\mathbf{v}_1 = \frac{\tilde{\mathbf{v}}_1}{\sqrt{\tilde{\mathbf{v}}_1^T\mathbf{M}\tilde{\mathbf{v}}_1}} \tag{B.4}$$

$$\tilde{\mathbf{v}}_1 = \mathbf{K}^{-1}\mathbf{f}. \tag{B.5}$$

For $i > 1$ [1]:

$$\mathbf{v}_i = \frac{\hat{\mathbf{v}}_i}{\sqrt{\hat{\mathbf{v}}_i^T\mathbf{M}\hat{\mathbf{v}}_i}} \tag{B.6}$$

$$\hat{\mathbf{v}}_i = \tilde{\mathbf{v}}_i - \sum_{j=1}^{i-1}\tilde{\mathbf{v}}_i^T\mathbf{M}\mathbf{v}_j\mathbf{v}_j \tag{B.7}$$

$$\tilde{\mathbf{v}}_i = \mathbf{K}^{-1}\mathbf{M}\mathbf{v}_i - 1\,. \tag{B.8}$$

[1] Note that for simplicity instead of the modified Gramm-Schidt process the original Gramm-Schmidt process is used. The orginal process uses $\mathbf{v}_i^{\mathrm{T}}\mathbf{M}\mathbf{v}_j$ for $j = 1,\ldots,i-1$. The modified process is given in Algorithm 2.1, this replaces $\mathbf{v}_j$ by the iterate of the previous step. This is numerically less stable, but much simpler[24].

One might think that the procedure is similar to the sensitivity analysis for $\mathbf{K}\mathbf{q} = \mathbf{f}$, since the first Ritz vector is calculated using this. But there is also a scaling step, which complicates the procedure. Direct differentiation of the equation for $\mathbf{v}_1$ yields:

$$\mathbf{v}_1' = \left[\frac{\tilde{\mathbf{v}}_1}{\sqrt{\tilde{\mathbf{v}}_1^T\mathbf{M}\tilde{\mathbf{v}}_1}}\right]' = \frac{\tilde{\mathbf{v}}_1'\sqrt{\tilde{\mathbf{v}}_1^T\mathbf{M}\tilde{\mathbf{v}}_1} - \tilde{\mathbf{v}}_1\frac{(2\tilde{\mathbf{v}}_1^T\mathbf{M}\tilde{\mathbf{v}}_1' + \tilde{\mathbf{v}}_1^T\mathbf{M}'\tilde{\mathbf{v}}_1)}{2\sqrt{\tilde{\mathbf{v}}_1^T\mathbf{M}\tilde{\mathbf{v}}_1}}}{\tilde{\mathbf{v}}_1^T\mathbf{M}\tilde{\mathbf{v}}_1} \tag{B.9}$$

$$\mathbf{K}\tilde{\mathbf{v}}_1' = \mathbf{f}' - \mathbf{K}'\tilde{\mathbf{v}}_1\,. \tag{B.10}$$

It is possible to calculate the sensitivity of the first vector in this way, but this is clearly more expensive than a finite difference approximation. A cheap adjoint formulation also seems unlikely. This requires first of all that all terms containing $\mathbf{v}_i'$ can be combined in an adjoint equations, preferably explicit. The remaining terms should also be cheap to evaluate. Both seem unlikely, the orthonormalization steps severely complicate the mathematics.

To find a possible alternative, it is assumed that the scaling and orthogonalization terms are independent of $\mathbf{x}$:

$$\left[\tilde{\mathbf{v}}_i^T\mathbf{M}\tilde{\mathbf{v}}_i\right]' = \left[\tilde{\mathbf{v}}_i^T\mathbf{M}\mathbf{v}_j\right]' \approx 0\,. \tag{B.11}$$

This is an unproven, possibly very inaccurate approximation, which will be tested later. But it does simplify the equations dramatically:

$$[\mathbf{K}\mathbf{v}_1 - \mathbf{f}*1/\mu_1]' \approx \mathbf{K}\mathbf{v}_1' + \mathbf{K}'\mathbf{v}_1 - \mathbf{f}'*1/\mu_1\,, \tag{B.12}$$

which is very similar to the sensitivity seen for the static problem in Section 2.4.4. An inexpensive adjoint formulation is possible in this case. This can be extended to the second Ritz vector:

$$\mathbf{0} = \left[\mathbf{K}\mathbf{v}_2 - \mathbf{K}\frac{\tilde{\mathbf{v}}_2 - \tilde{\mathbf{v}}_2^T\mathbf{M}\mathbf{v}_1\mathbf{v}_1}{\mu_2}\right]', \qquad \tilde{\mathbf{v}}_2 = \mathbf{K}^{-1}\mathbf{M}\mathbf{v}_1\,, \qquad \mu_i = \sqrt{\tilde{\mathbf{v}}_i^{\mathrm{T}}\mathbf{M}\tilde{\mathbf{v}}_i} \tag{B.13}$$

$$\approx \mathbf{K}\left(\mathbf{v}_2' - \frac{\tilde{\mathbf{v}}_2' - \tilde{\mathbf{v}}_2^T\mathbf{M}\mathbf{v}_1\mathbf{v}_1'}{\mu_2}\right) + \mathbf{K}'\left(\underbrace{\mathbf{v}_2 - \frac{\tilde{\mathbf{v}}_2 - \tilde{\mathbf{v}}_2^T\mathbf{M}\mathbf{v}_1\mathbf{v}_1}{\mu_2}}_{=0}\right), \qquad \mathbf{K}\tilde{\mathbf{v}}_2' = \mathbf{M}'\mathbf{v}_1 + \mathbf{M}\mathbf{v}_1' - \mathbf{K}'\tilde{\mathbf{v}}_2\,. \tag{B.14}$$

This leads to the following adjoint formulation:

$$\Psi_{\mathrm{A}} = \Psi(\mathbf{v}_1, \mathbf{v}_2) + \boldsymbol{\gamma}_1^T\left(\mathbf{K}\mathbf{v}_1 - \mathbf{f}*1/\mu_1\right) + \boldsymbol{\gamma}_2^T\mathbf{K}\left(\mathbf{v}_2 - \frac{\tilde{\mathbf{v}}_2 - \tilde{\mathbf{v}}_2^T\mathbf{M}\mathbf{v}_1\mathbf{v}_1}{\mu_2}\right) \tag{B.15}$$

$$\frac{\mathrm{d}\Psi_{\mathrm{A}}}{\mathrm{d}\mathbf{x}} \approx \frac{\partial\Psi}{\partial\mathbf{v}_1}\frac{\partial\mathbf{v}_1}{\partial\mathbf{x}} + \frac{\partial\Psi}{\partial\mathbf{v}_2}\frac{\partial\mathbf{v}_2}{\partial\mathbf{x}} + \boldsymbol{\gamma}_1^T\left(\mathbf{K}\mathbf{v}_1' + \mathbf{K}'\mathbf{v}_1 - \mathbf{f}'*1/\mu_1\right) + \boldsymbol{\gamma}_2^T\left(\mathbf{K}\mathbf{v}_2' - \mathbf{K}\frac{\mathbf{M}'\mathbf{v}_1 - \mathbf{K}'\tilde{\mathbf{v}}_2 - (\tilde{\mathbf{v}}_2^T\mathbf{M}\mathbf{v}_1\mathbf{I} + \mathbf{M})\mathbf{v}_1'}{\mu_2}\right). \tag{B.16}$$

When the terms are regrouped, the adjoint equations can be extracted:

$$\frac{\mathrm{d}\Psi_{\mathrm{A}}}{\mathrm{d}\mathbf{x}} \approx \underbrace{\left(\frac{\partial\Psi}{\partial\mathbf{v}_1} + \boldsymbol{\gamma}_1^T\mathbf{K} + \boldsymbol{\gamma}_2^T\mathbf{K}(\tilde{\mathbf{v}}_2^T\mathbf{M}\mathbf{v}_1\mathbf{I} + \mathbf{M})\right)}_{\text{Adjoint equation 1}}\mathbf{v}_1' + \underbrace{\left(\frac{\partial\Psi}{\partial\mathbf{v}_2} + \boldsymbol{\gamma}_2^T\mathbf{K}\right)}_{\text{Adjoint equation 2}}\mathbf{v}_2' + \boldsymbol{\gamma}_1^T\left(\mathbf{K}'\mathbf{v}_1 - \mathbf{f}'*1/\mu_1\right) - \boldsymbol{\gamma}_2^T\left(\mathbf{K}\frac{\mathbf{M}'\mathbf{v}_1 - \mathbf{K}'\tilde{\mathbf{v}}_2}{\mu_2}\right). \tag{B.17}$$

This can easily be extended to an arbitrary numbers of basis vectors, but that is not necessary at this point. The general procedure for the approximated adjoint method is clear:

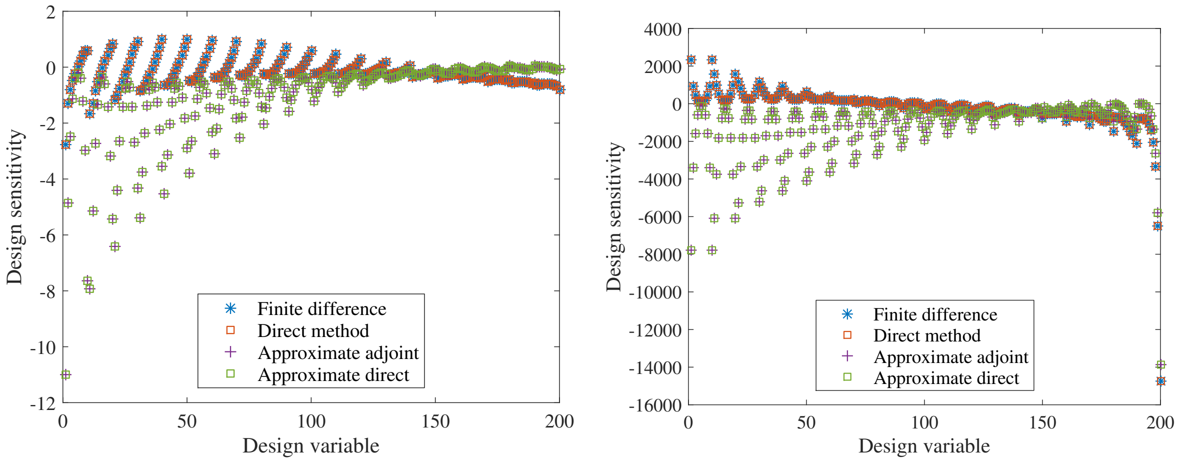
1. Calculate basis vectors, storing intermediate vectors $\tilde{\mathbf{v}}_i$.
2. Solve highest adjoint equations to find $\boldsymbol{\gamma}_n$.
3. Solve next adjoint equation to find $\boldsymbol{\gamma}_{n-1}$, etc., up to $\boldsymbol{\gamma}_1$.

4. Calculate adjoint sensitivity. This just consists of matrix multiplications, so it is relatively cheap.

This procedure is similar to many other adjoint methods: the simulation is performed forwards, and a similar simulation is performed backwards. The main cost is the solution of the adjoint equations: this requires one solution of a linear system per basis vector. This is similar to the cost of generation the basis vectors, so it is much cheaper than finite difference or direct methods.

However the entire derivation is reliant on the unproven assumption shown in Eq. (B.11). The best way of checking the validity of this assumption is by computing the sensitivity using the various methods. This was done by defining the objective function $\Psi = \mathbf{L}^T\mathbf{v}_1$, and calculating the design sensitivity for the various methods. The results are shown in Figure B.1. Only the first basis vector is examined. The direct method is consistent with the finite difference method, so these are assumed to be relatively accurate. The approximate adjoint method leads to significant errors. To verify the role of assumption B.11, the direct method was changed to use the same assumption. The results are identical to the approximate adjoint method.

Concluding, an efficient, accurate method has not been found. The approximate adjoint method leads to significant errors. Since it was found in Chapter 5 that Ritz vector sensitivities can be neglected, this subject was not investigated further.



(a) Ritz vector sensitivity for $\mathbf{L} = \mathbf{1}$.

(b) Ritz vector sensitivity for $\mathbf{L} = \mathbf{0}$, except for the last entry in $\mathbf{L}$, which is 1.

Figure B.1: Comparison of different methods of computing the sensitivity of the first basis vector. The approximated methods lead to significant errors.

C
Approaches to Topology Optimization

Structural optimization is the process of finding the optimal shape of a structure. This is done by defining the geometry of the structure as a function of a number of variables, and finding the optimal value for these variables. Topology optimization is a special type of structural optimization which allows a lot of freedom for the design. In shape optimization, the design variables are usually associated with the size and position of an element. On the other hand, in topology optimization, the design variables determine the amount of material at a certain location. This generally leads to more design freedom, at the cost of an increase in the number of design variables.

There are multiples approaches to topology optimization, these will be discussed in this chapter.

C.1. Literature

Within the vast amount of literature on topology optimization, recommended references are the standard book by Bendsoe and Sigmund [7] and review papers by Sigmund and Maute [55] and Deaton and Grandhi [1]. Also, the educational Matlab programs *top99.m* [56] and *top88.m* [49] are a great start for any research on this subject.

C.2. Approaches

In topology optimization the first important choice is what type of design parameters to use. There are a few possibilities, the most common choices are:

- *Density based methods*: also known as material distribution methods. In this approach the design space is divided into different elements, which are given a continuous density variable. These densities are the design variables.
- *Boundary evolution*: the most common method here is the level-set method. In this approach instead of the elements the boundaries of the structure are controlled by the design parameters. There are many different possibilities here, see for example the recent review paper by van Dijk[57].
- *Heuristic*: in these methods, a model of the structure is made of the structure. From this model, we determine where to remove or add material, for example at a location with low respectively high stress. This approach is quite intuitive, and very different from the other approaches.
  Because of the heuristic algorithm and the discrete design variables, the optimality and convergence of the solution are difficult to prove. For highly non-linear problems they can be be efficient, for example using a biologically inspired optimization algorithm. See [58] for a comparative study, and [59] for an application.

The boundary evolution methods have some advantages, most notably in that they can naturally produce black-and-white designs. In the majority of literature density based methods are used, because the procedures involved are simpler.

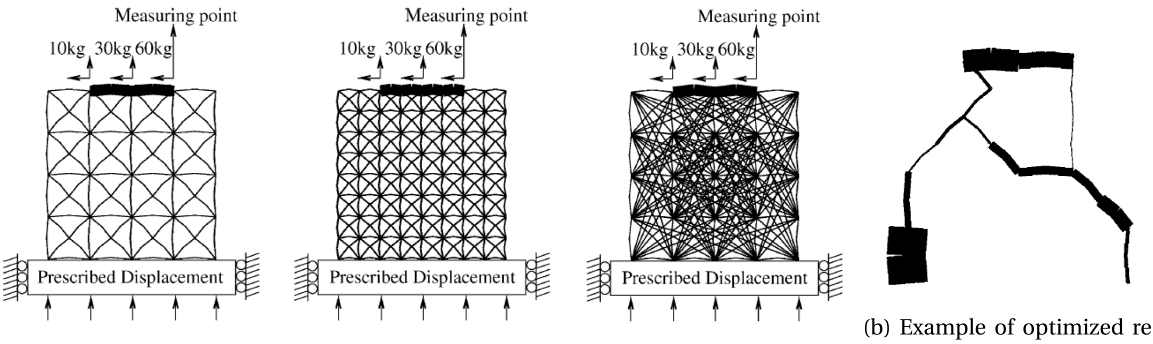
C.2.1. Density based methods

Within density based methods there are multiple ways to parametrize the system. The most notable choices all use the Finite Element Method, they are:

- *Homogenization* [60]: in this approach, the structure is divided into cells, each of which contains a small void. The size of this void is the design variable. Using homogenization theory, the full system can be analysed and optimized. This method has been replaced by the other methods named in this chapter.

- *Ground structure*: in this approach, a structure is defined using a number of elements, usually Euler beams. See Fig. C.1 for an example. A size parameter of these elements is used as a design variable. This method allows relatively easy modelling of non-linear problems, by using a non-linear beam formulation.

- *'Sheet thickness'*: finally, in this approach, the structure is divided into elements. These are usually linear rectangular/cubical elements, but other shapes or higher order basis functions are possible. The material density of these elements is used as a design variable. In 2D, this could be interpreted as sheet thickness.



(a) Different ground structures.

(b) Example of optimized result.

Figure C.1: Ground structures used in topology optimization, consisting of beams (lattice structure). Image source: [44]. The size of these beams is used as design variable.

The final two methods are most commonly used. One could argue that they are the same, just with a different element formulation and ground structure layout. In this research, the 'sheet thickness' density based approach is used, for an illustration see Fig. 2.1.

D

Details of implementation

This chapter contains some additional details about the implementation of the methods discussed in this research.

D.1. Overview of parameters

For the finite elements analysis a few parameters have to be selected, including material properties. These do not have a major influence on the outcome of this results, so a systematic investigation was deemed unnecessary. The material properties represent steel. The Rayleigh damping properties were chosen to reproduce the results from literature, or chosen such that the lowest frequencies are lightly damped.

Table D.1: Values of physical parameters used in this research.

Parameter	Symbol	Value	Unit
In-plane thickness of elements		0.01	m
Young Modulus	E	2×10^{11}	$\mathrm{N/m^2}$
Density	ρ	7800	$\mathrm{kg/m^3}$
Poisson ratio	ν	0.30	-
Rayleigh damping coefficient 1 (optimization example 2)	α	10	$\mathrm{s^{-1}}$
Rayleigh damping coefficient 2 (optimization example 2)	β	1×10^{-5}	s
Rayleigh damping coefficient 1 (all other cases)	α	1×10^{-4}	$\mathrm{s^{-1}}$
Rayleigh damping coefficient 2 (all other cases)	β	1×10^{-4}	s

D.2. Details of MATLAB implementation

The majority of the code was written by the author. There are a few notable exceptions:

- When a built-in function exists, this is generally used, unless a significantly faster alternative can be found/created.
- Some code from *top88.m*[49] is used, for example in matrix assembly.
- The implementation of MMA was provided by its author, prof. Svanberg.
- A few minor functions used in exporting figures were found on the MATLAB file exchange.

The main code consists of a set of functions, for example for finite element analysis, Newmark integration, sensitivity analysis, etc, adding up to a combined total of approximately 2000 lines of code. Each function starts with documentation on its implementation and syntax. A user-friendly GUI is also included, which allows easy experimentation with topology optimization and MOR, as well as visualization of basis vectors and a few other features. To generate the various figures in this document a large number of scripts was created, which use the various functions to perform the described experiments.

The full code is available upon request.

D.3. ALTERNATIVE SENSITIVITY ANALYSIS

In this research, the sensitivity analysis was based on first differentiating the governing equations, and performing temporal discretization second. According to literature [48], this leads to inconsistent sensitivities, as was discussed in Section 3.2.5. An alternative, more consistent method is based on the reverse approach: the equations are temporally discretized first, after which they are differentiated. This leads to a different set of discretized equations, for details see [48]. The resulting sensitivity is given by:

$$\Psi = \int_0^T \Phi(\mathbf{q})\,\mathrm{d}t \approx \sum_{i=1}^{n_t} \Phi(t_i) \tag{D.1}$$

$$\frac{\mathrm{d}\Psi_\mathrm{B}}{\mathrm{d}\mathbf{x}} = \sum_{i=1}^{n_\mathrm{t}} \boldsymbol{\lambda}^\mathrm{T}(t_i)\left(\mathbf{f}' - \mathbf{M}'\ddot{\mathbf{q}}(t_i) + \mathbf{C}'\dot{\mathbf{q}}(t_i) + \mathbf{K}'\mathbf{q}(t_i)\right)\mathrm{d}t\,. \tag{D.2}$$

where Ψ_B is the augmented objective function. This consists of the objective Ψ with an additional adjoint term: the discretized equations of motion as found by the Newmark algorithm. The resulting sensitivity is much more accurate when using trapezoidal integration instead of summation [48], for brevity this is not shown in the equation. The summation symbols in the equations should be interpreted as trapezoidal integration.

In this approach, the adjoint variable is calculated differently. This is briefly described here, using $\boldsymbol{\Lambda}_{n_t-i} = \boldsymbol{\lambda}(t_i)$ to simplify notations for backwards time integration. The following steps are performed sequentially:

For $i = 1$:

$$\boldsymbol{\mu}_1 = \frac{\partial\Phi}{\partial\mathbf{q}}^\mathrm{T}(t = T) \tag{D.3}$$

$$\boldsymbol{\nu}_1 = \mathbf{0} \tag{D.4}$$

$$\left(\mathbf{M} + \gamma h\mathbf{C} + \beta h^2\mathbf{M}\right)\boldsymbol{\Lambda}_i = -\beta h^2\boldsymbol{\mu}_1 - \gamma h\boldsymbol{\nu}_1\,. \tag{D.5}$$

For $2 \leq i < n_\mathrm{t}$:

$$\boldsymbol{\mu}_i = \frac{\partial\Phi}{\partial\mathbf{q}}^\mathrm{T}(t = T - t_i) + \mathbf{K}\boldsymbol{\Lambda}_{i-1} + \boldsymbol{\mu}_{i-1} \tag{D.6}$$

$$\boldsymbol{\nu}_i = h\left(\mathbf{K}\boldsymbol{\Lambda}_{i-1} + \boldsymbol{\mu}_{i-1}\right) + \mathbf{C}\boldsymbol{\Lambda}_{i-1} + \boldsymbol{\nu}_{i-1} \tag{D.7}$$

$$\left(\mathbf{M} + \gamma h\mathbf{C} + \beta h^2\mathbf{M}\right)\boldsymbol{\Lambda}_i = -\beta h^2\boldsymbol{\mu}_i - \gamma h\boldsymbol{\nu}_i - (0.5-\beta)h^2(\mathbf{K}\boldsymbol{\Lambda}_{i-1} + \boldsymbol{\mu}_i - 1) - (1-\gamma)h(\mathbf{C}\boldsymbol{\Lambda}_{i-1} + \boldsymbol{\nu}_{i-1})\,. \tag{D.8}$$

Finally for $i = n_\mathrm{t}$:

$$\mathbf{M}\boldsymbol{\Lambda}_i = -\left((0.5-\beta)h^2((\mathbf{K}\boldsymbol{\Lambda}_{i-1} + \boldsymbol{\mu}_i - 1)) + (1-\gamma)h(\mathbf{C}\boldsymbol{\Lambda}_{i-1} + \boldsymbol{\nu}_{i-1})\right)\,. \tag{D.9}$$

This looks much more complex than the other approach. But for the computation of $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ only a few matrix multiplications are needed. This make the total cost similar to the normal Newmark integration.

Summarizing, two methods of computing λ were described. The first was directly based on the theory described in Section 2.4. This approach can be described as *differentiate-then-discretize*. According to the work of Jensen *et al.* [48], this can lead to inconsistent sensitivities. An second method named *discretize-then-differentiate* is described, which is based on differentiating the discretized equations. This is slightly more expensive. Both methods have been implemented, only the computation of $\boldsymbol{\lambda}$ is changed. The accuracy and consistency of both methods will be investigated in Chapter 5.

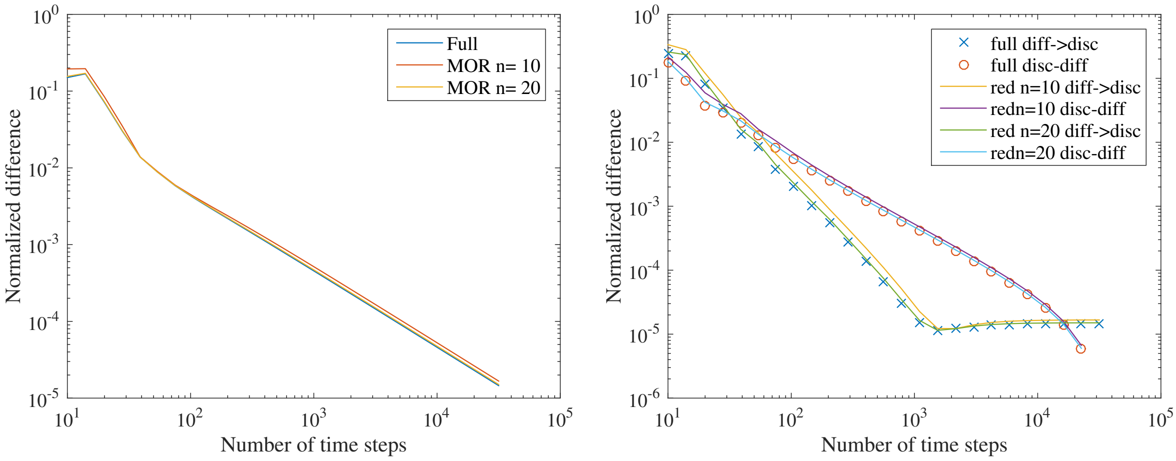
More important in the scope of this thesis is whether this consistency error is affected by model reduction. To test this, both methods of sensitivity analysis were used on the same problem, both with and without model reduction (using Ritz vectors). The used error measure is:

$$e = \frac{\left\|\frac{\mathrm{d}\Psi_\mathrm{ref}}{\mathrm{d}d_i} - \frac{\mathrm{d}\Psi}{\mathrm{d}d_i}\right\|}{\left\|\frac{\mathrm{d}\Psi_\mathrm{ref}}{\mathrm{d}d_i}\right\|}, \qquad 1 \leq i \leq n_\mathrm{el}\,. \tag{D.10}$$

The difference between the two methods converges to zero for an increasing number of timesteps, as illustrated by Fig. D.1a. Model order reduction does not affect this error, the results are similar to those found by

Jensen *et al.* [48]. Verifying whether the new method indeed leads to more consistent sensitivities is difficult, since the exact reference sensitivity is unknown. The best available approximate is the sensitivity of the full model, calculated using the new approach with the largest number of timesteps. When this is used as reference, the results shown in Fig. D.1b are found.

Both methods can be implemented in the same code with minimal effort. Both were tested on a few optimization problems, where the new approach did not results in faster convergence. It is expected that other errors dominate the convergence, thus this was not investigated further. The original approach has the further benefit of allowing relatively easy analytical investigation of the effect of model order reduction, as seen in Section 4.1.



(a) Difference between both methods of sensitivity analysis versus number of time steps. The difference decreases when the number of timesteps increases, but more important: model order reduction (Ritz vectors) does not significantly change this.

(b) Error with respect to accurate sensitivity (discretize-differentiate on full model, for 32000 timesteps) for various methods, versus number of time steps.

Figure D.1: Results of alternative sensitivity analysis. The data shows that model order reduction does not influence the difference between the two approaches.

D.4. Details of error estimation

The adaptive model order reduction algorithm using in this thesis relies on several error estimates, as explained in Section 2.5.5. This chapter discusses the associated details and results.

D.4.1. Fourier analysis of load

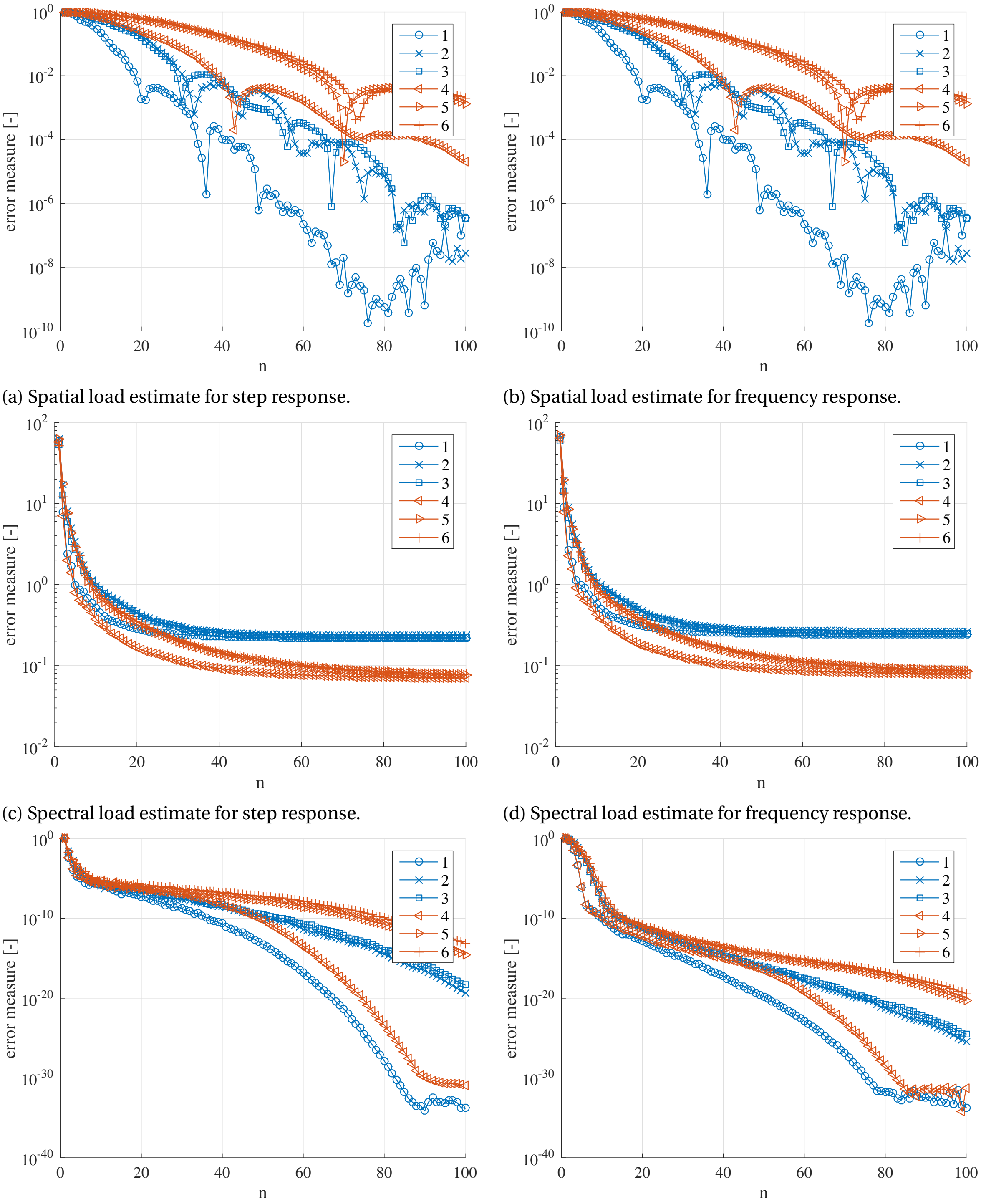
The *spectral load estimate* is calculated by Eq. (2.50):

$$e = \frac{\max \omega_{\text{load}}}{\max \omega_{\text{red}}}, \qquad e = \frac{\min \omega_{\text{red}}}{\min \omega_{\text{load}}}, \tag{D.11}$$

The frequency content of the model can be found by calculating the eigenvectors of the reduced model. The frequency of the load is more complicated. For harmonic loads this is directly available. However, for other cases such a step response it is not. To calculate the frequency content, the discrete time Fourier transform is employed (MATLAB: *fft*), following the procedure described by [61]. In the calculated frequency spectrum, a cut-off frequency is found such that 95% of the area below the spectrum is included. This frequency is used as ω_{load} in the error estimate.

D.4.2. Detailed results of error estimates

In Chapter 5 the MOR error was investigated of the response calculated using the reduced model. This section details the results of the various error estimates. For both the step response and frequency response ($\Omega = 1 \times 10^4$ rad/s problems used in Chapter 5 this was studied, the results are shown in Fig. D.2.



(a) Spatial load estimate for step response.

(b) Spatial load estimate for frequency response.

(c) Spectral load estimate for step response.

(d) Spectral load estimate for frequency response.

(e) Refinement estimate for step response.

(f) Refinement estimate for frequency response.

Figure D.2: Results of error estimatation. All estimates converge to zero, except for the spectral load estimate. Furthermore, the spatial load estimate is non-monotonic, making it less useful for adaptive model order reduction. The refinement estimates shows the desired monotonic convergence, which can be linked to the MOR errors found in Chapter 5. However, its magnitude is dependent on the frequency of the load, which implies that this should be taken into account when selecting the correct reference values.