Technische Universität München
Fakultät für Maschinenwesen
Institut für Luft- und Raumfahrt
Fachgebiet Leichtbau

Fast Reanalysis for Large Scale Multicriteria Structural Optimization

Daniel Franz Xaver Heiserer

Vollständiger Abdruck der von der Fakultät für Maschinenwesen
der Technischen Universität München zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. habil. Heinz Ulbrich
Prüfer der Disseration:

1. Univ.-Prof. Dr.-Ing. Horst Baier
2. Univ.-Prof. Dr.-Ing. Wolfgang A. Wall

Die Dissertation wurde am 15.06.2004 bei der Technischen Universität eingereicht
und durch die Fakultät für Maschinenwesen am 17.12.2004 angenommen.

Preface

This thesis was created during and beside my engagement as numerical engineer at the BMW research and development center FIZ.

I thank Prof. Dr.-Ing. H. Baier for his scientific coaching and supervising the thesis during the last five years. His experience in optimization and confidence in the work as well as his willingness for discussion were vital for the success of this work.

I also thank Prof. Dr.-Ing. W.A. Wall for his interest in the work.

Special credit goes to my management at BMW and my employer. Without the necessary capacity provided and their steady interest in pushing the envelope in research and development this work could not have been created. The good environment and the support of colleagues was always helpful. I thank Dr.-Ing. habil. F. Duddeck, Dr.-Ing. habil. F. Ihlenburg, and Dr. Louis Komzsik for critical proofreading. Their valuable advices improved the quality of the work. I owe peculiar thanks to M. Chargin and Dr. H. Miura for their countless discussions. M. Chargin's experience and his countless jokes helped me to overcome even the gravest hurdles in the Nastran software environment.

Special thanks go to my girl friend Tanja for her support and patience during the last four years and to my family and friends whose time I could share only very limited while working on this thesis.

Munich, June 2004

Daniel Heiserer

Contents

Preface i

Nomenclature viii

Abstract xi

1 Introduction and goals 1

2 General demands for multi criteria optimization 5

2.1 Mathematical formulation . . . 5

2.1.1 Multicriteria optimization . . . 6

2.2 Practical formulation . . . 6

2.3 Numerical approaches and requirements . . . 6

2.4 The classical ten-bar truss problem . . . 7

3 Geometry and kinematics of bodies 9

3.1 Abstraction of Euclidean space . . . 9

3.1.1 Concept of manifolds . . . 9

3.1.2 Vectors and tensors . . . 10

3.1.3 Riemannian manifold . . . 11

3.1.4 Maps of manifolds . . . 11

3.2 Motion and deformation of a body on $\mathbb{R}^3$ . . . 11

3.2.1 Configurations . . . 11

3.2.2 Deformation gradient . . . 12

3.2.3 Strain tensors . . . 12

3.2.4 Conservation of Mass . . . 12

3.2.5 Velocity and acceleration . . . 13

4 From the variational formulation to the discretized equations of motion 15

4.1 Energy functions of deformable bodies 15
  4.1.1 Stress tensors and constitutive relation 15
  4.1.2 Strain energy 16
  4.1.3 Kinetic energy 16
4.2 Hamiltonian principle 17
4.3 Spatial discretization of the Lagrangian 17
  4.3.1 Kinematic assumptions 18
  4.3.2 Linearization 18
    4.3.2.1 Material linearization 18
    4.3.2.2 Geometric linearization 19
    4.3.2.3 Summary 20
  4.3.3 Discretization using the Rayleigh-Ritz procedure 21
    4.3.3.1 The finite elements 21
    4.3.3.2 Discrete form of kinetic energy 22
    4.3.3.3 Discrete form of strain energy 22
    4.3.3.4 Discrete form of nonconservative work 23
4.4 Equations of motion in matrix notation 24
  4.4.1 Eigenvalues for low damped dynamic systems and the characteristic equation 25
  4.4.2 System matrices for the ten-bar truss model 25

5 Parameterized equations of motion and their solution manifold 29

5.1 Preliminaries and assumptions 29
5.2 The solution space properties for the discretized problem 31
5.3 Local approximation of the 1-dimensional manifold 33
5.4 Determination of coefficients using the Rayleigh-Ritz procedure 38
  5.4.1 The equations of motion in transformed coordinates 39
    5.4.1.1 The mass matrix in transformed coordinates 39
    5.4.1.2 The stiffness matrix in transformed coordinates 39
    5.4.1.3 Nonconservative terms in transformed coordinates 40
5.5 Local approximation of the *d*-dimensional manifold 40
5.6 Global and mixed local global approximation using interpolation 42

6 Practical implementation and implications for structural problems 43

6.1 Typical design variables in structural optimization 43

6.1.1 Derivatives of system matrices for shell elements 44

6.1.2 Derivatives of system matrices for beam elements 45

6.1.3 Derivatives of system matrices for mass elements 47

6.1.4 Derivatives of system matrices for spring/damper elements 47

6.2 Practical calculation of derivative modes 48

6.3 Krylov sequence and Krylov subspaces 49

6.3.1 Dynamic steady state: frequency response 49

6.3.2 Exact results and the rank of the Krylov subspace 50

6.4 Derivative modes of the ten-bar truss problem 50

6.5 Data flow recipe 52

6.6 Related approaches and similar procedures in literature 53

6.6.1 Global approximation in design space 53

6.6.2 Combined approximation in the context 54

6.6.3 The pure natural modes approach 54

6.7 Mechanical interpretation 54

6.7.1 Role of the strain energy 55

6.7.2 Load path interpretation 55

7 Computational complexities and performance considerations 57

7.1 Floating point operations and performance 57

7.2 Floating point operation counts for basic matrix operations 58

7.2.1 Matrix addition and subtraction 58

7.2.2 Matrix multiplication 59

7.2.3 Solving symmetric systems 59

7.2.4 Gram-Schmidt ortho-normalization 60

7.3 Matrix properties for structural problems 61

7.3.1 Approximation of front-width and non-zeroes in the Cholesky factor . . . 64

7.3.1.1 Behavior of the front-width 66

7.3.1.2 Behavior of the number of non-zeroes in the Cholesky factor . . 68

7.3.1.3 Behavior of real world models 69

7.4 Computational complexity for a structural optimization problem 75

7.4.1 Standard approach . . . 75
7.4.1.1 Non-sensitivity based solution . . . 75
7.4.1.2 Gradient based solution . . . 75
7.4.2 *Approximate Response Manifold* (ARM) based solution . . . 77
7.4.2.1 Non-sensitivity based solution . . . 78
7.4.2.2 Gradient based solution . . . 79
7.4.3 Discussion and comparison of numerical efficiency . . . 79
7.4.3.1 Megaflop examples and computing time . . . 81

8 Applications to structural design problems 83
8.1 Uni parametric study: Pareto optimal stiffness of engine hood . . . 83
8.1.1 Model and loading conditions . . . 83
8.1.2 Parametric task . . . 84
8.1.3 Procedure . . . 84
8.1.4 The solution manifold . . . 85
8.1.5 Comparison of *approximate response manifold* with *response surface method* 86
8.1.5.1 The *approximate response manifold* . . . 86
8.1.5.2 The *response surface method* . . . 87
8.1.5.3 *Approximate response manifold* versus *response surface method* . 90
8.1.6 Corresponding weight potential of the interior sheet using Pareto optimization . . . 91
8.1.7 Corresponding weight potential of the interior sheet using topology optimization . . . 93
8.2 Multi dimensional parametric study: static stiffnesses of a complete car body . . 97
8.2.1 Model and loading conditions . . . 97
8.2.2 ARM specific formulation . . . 100
8.2.3 Comparison of the exact solution to the ARM method . . . 101
8.2.4 Numerical effort . . . 102

9 Summary 105

A Matrix Utility Operations 107
A.1 Physical sets introduced by discretization . . . 107
A.2 Matrix Notations . . . 108
A.3 Multi-point Constraint Elimination . . . 109

A.3.1 Multi-point Constraint Elimination for full sized matrices 109

A.3.2 Multi-point Constraint Elimination for sub-models (components) 109

A.4 Single-point Constraint Elimination 111

B Synthesis of system matrices **113**

B.1 Analytical description 113

B.1.1 Example for shell elements with transverse shear 114

C Properties of Krylov subspaces **115**

C.1 Krylov subspaces 115

List of Figures **117**

Bibliography **119**

Curriculum Vitae **129**

Nomenclature

The notation in this thesis is not fixed in a way that a specific symbol necessarily means the same thing throughout the whole thesis. The symbols are defined according to common usage and the formulas are aiming at readability instead of fulfilling a fixed symbol scheme.

The convention is the following:

- the symbols defined in the nomenclature represent the default usage for the whole thesis
- any symbol can be redefined in the thesis. This redefinition is then valid from the point of redefinition till the end of the current hierarchy such as chapter/section/subsection.

Symbols:

$\boldsymbol{X}$: **bold face**, upper case Latin letters denote tensors of second order (matrices) or higher in *symbolic notation*

$\boldsymbol{x}$: **bold face**, lower case Latin letters denote tensors of first order (vectors) *in symbolic notation*

x_i : denotes the i-th element of $\boldsymbol{x}$

$\boldsymbol{x}_i$: denotes the vector $\boldsymbol{x}$ with index i

d : amount of free parameters

Λ : d-dimensional parameter space

$\boldsymbol{\lambda}$: vector of parameters $\in \Lambda$

$\mathcal{B}$: reference configuration of a body

X : a point in $\mathcal{B}$

$\mathcal{S}$: space in which the body moves

x : a point in $\mathcal{S}$

$\boldsymbol{g}$: Riemannian metric on $\mathcal{B}$

$\boldsymbol{G}$: Riemannian metric on $\mathcal{S}$

M^d : d-dimensional manifold

T_xM^d : Tangent space of a manifold at at point x

$\boldsymbol{F}$: deformation gradient

$\boldsymbol{C}$: right Cauchy-Green-Tensor

$\boldsymbol{E}$: material (Lagrange) or Green strain tensor

$\boldsymbol{S}$: second Piola-Kirchhoff stress tensor
$\mathcal{C}$: elasticity tensor
$\boldsymbol{D}$: flexibility tensor
$\boldsymbol{M}$: mass matrix
$\boldsymbol{K}$: stiffness matrix
$\boldsymbol{B}$: damping matrix
$\boldsymbol{N}^i$: local shape function introduced by the finite element approach
N : number of nodes
DOF : degrees of freedom, in our case displacement degrees of freedom
n : number of degrees of freedom (DOF), in our case displacement degrees of freedom including rotations
PDF : parametric degrees of freedom
$\mathcal{N}$: solution space for displacements, $\mathbb{R}^n$
$\mathcal{D}$: subspace of $\mathbb{R}^n$, designed to embed the solution manifold M^d approximately
$\boldsymbol{D}^i$: *mode*, see description below, a global shape function, *response manifold approach*
Φ : operator transforming $\mathcal{D}$ to $\mathcal{N}$

In this thesis the word *mode* is used multiple times. Basically it describes a vector of displacement degrees of freedom. Therefore this vector represents some deflection shape. It is distinguished between

- *normal modes*, solutions of the generalized eigenvalue problem $[\boldsymbol{K} - \lambda \boldsymbol{M}]\boldsymbol{x}$.
- *derivative modes* defined in the thesis.
- *residual modes*, additional displacement vectors. Defined in the thesis.

Abstract

This thesis is about a fast and accurate reanalysis procedure. It focuses on elastic structures used in parametric design, optimization and reliability analysis. The mathematical description bases on the manifold structure of parametric problems described by a non iterative system of equations. Therefore the method will be called *approximate response manifold* (ARM). The finite element method is used as the underlying method of simulating these structures.

An efficient set of solution vectors is derived to embed the solution manifold of the parametric design space within an Euclidean subspace of the original problem. The solution is calculated using the Hamilton principle and the Rayleigh-Ritz procedure.

Different strategies to obtain the solution vectors are discussed, depending on the parameter range of the underlying reanalysis problem. After introducing the procedure, the numerical effort and complexity is outlined in comparison to a traditional analysis.

The procedure is designed for large and complex optimization problems, characteristic for real world problems in industrial structural design, such as the automotive design process. Focused on a real world problem the efficiency of the formulation is discussed and compared, in terms of accuracy and numerical effort, to a traditional surrogate model such as the *response surface method* (RSM).

It is shown that the method can solve large scaled linear static structural problems with up to hundreds of design variables very fast on low end computing systems. It is therefore suitable for multi-criteria optimization, trade off studies and reliability analyses.

1 Introduction and goals

In previous times the *finite element* (FE) based simulation had primarily a more advising role concerning design decisions in the car body development. Main reasons for that were missing accuracy of the simulation models and a too time consuming process to create a simulation model out of *computer assisted design* (CAD) based geometry. Due to a steady accelerating modeling process and a constant increase of computing power, as predicted by *Moore's law*, enough (affordable) computing resources were provided to solve models with enough details to come close to the desired accuracy (linear statics, modal dynamics). The crossover point was reached by dramatically improving the accuracy of the finite element models and limiting the error, of displacements and stresses, to an acceptable engineering range of a few percent for static and dynamic simulations. The most dramatic change however was not made by evolutionary increase of model size and therefore affordable geometric details solvable on higher *megaflop* machines. Instead the breakthrough occurred with a quantum leap considering connectivity data such as weld spots and seem welds (see Vopel and Hillmann [120], Zhang and Richter [124], and Heiserer, Sielaff and Chargin [39]).

After the introduction of this process-oriented spot weld approach many commercially available finite element solvers came up with similar solutions (see [93]). This changed the role of the simulation significantly. From a more consulting role it changed to a confirming and more designing role. The challenge nowadays is that, instead of only confirming or objecting design changes in the car body development, design changes have to be suggested and introduced by the FE based simulation.

Currently there are mainly two major bottlenecks visible in this design process. One is the correlation of different simulation objectives (such as statics, dynamics, acoustics, crashworthiness, durability, fatigue, etc.) among a certain CAD status. Design suggestions emerging from one discipline must be reevaluated by other disciplines to obtain a coherent status. The other bottleneck is the automated search for design improvements using numerical optimizers. A common design evaluation for different criteria and disciplines is extremely time consuming.

Facing the fact that one single simulation of these large scaled models is still a crucial task (hours and days of computing time) an automated search for improved designs, especially the usage of mathematical optimization, is immense and turns out to be either not affordable or at least not practicable in the given time frame if many criteria or disciplines have to be considered. Due to the fact that most models are still growing with the computing resources, because of the need of more precise design forecasts, the hopes that the steady *megaflop growth* will solve the multi-criteria optimization problem by itself have no real basis. This seems to be valid at least for the next 5-8 years if we extrapolate the recent growth concerning computing power.

Beside the well known modal method for frequency response calculation there are few examples in the finite element simulation which generate and reuse a priori knowledge of the system in order to accelerate subsequent parametric analysis.

Fox and Miura [29] use a design of experiment to form a reduced basis which is used afterwards as the solution space. Noor and Lowder [101], [102], and Noor [100], [97], [98] and [99], while focusing on the solution of nonlinear systems, derive a reduced basis using continuous differentiation of the initial solution. By iterating using a change in the stiffness operator Kirsch and Liu [65], [61] use a polynomial expansion to form a reduced basis. Based on these *Combined Approximations* a change in a sizing variable, a geometrical change (Kirsch and Pabalambros [69]) a topological change (Kirsch [57], Kirsch and Pabalambros [70]) is efficiently approximated. However, multiple parameters were not addressed with this approach.

A basis consisting of normal modes is used by Mass [87], Witta [121], and Freymann [30] to predict the frequency response of moderate perturbed dynamic systems.

Common to all above listed approaches is the basic idea to reuse the results of prior calculations and predict the subsequent analyses faster with a basis of a reduced Euclidean space. In this thesis it is shown how such a subspace, which encapsulates the manifold of the parametric solution most efficiently concerning accuracy and computing effort, can be generated.

Despite the above mentioned successes in this area of reanalysis a common theory for these approaches is still missing. Due to that fact that a global understanding of the reduced basis technique was difficult, an estimation of it's mightiness and limitations was hardly possible.

The goal of this work is to contribute a common theory to this area and extend it to multi parameters. By serving as a fast evaluation engine the approach can then, in multiple instances representing different load cases, efficiently be used for multi criteria optimization.

Therefore the thesis focuses on the following contents and sub-ordinate targets:

- In chapter 2 the mathematical fundamentals of optimization are summarized to give an overview of the design formulation. A typical optimization problem for structural problems is presented and mathematically formulated. The deficiencies in todays procedures facing large scaled optimization problems, as for example used in the automotive industry, are outlined.

- Chapter 3 introduces the basic concept of the manifold, which represents the generalized *response surface* of parametric systems. Chapter 3 also introduces the kinematic variables to describe deformation and movement of elastic structures, which describe in conjunction with the fundamental physical principles and equations of chapter 4 the structural system.

- Starting from a differential geometric point of view, the solution space of a multi parametric model is examined in chapter 5. Based on the analytic formulation of the equations of motion, a convergent procedure for an approximation of the corresponding solution manifold is given. This procedure will be called *approximate response manifold* (ARM).

Motivated by the before mentioned works of Fox/Miura and Noor, the *Rayleigh-Ritz* method, the in 4 shown principle of the finite element method, is used to calculate efficiently the approximated solution of the response manifold in the reduced basis.

- In chapter 6 it is shown that the solutions of fox [29], Kirsch [65], Noor [101], Mass [87] and Witta [121] and even partly the Lanczos eigenvalue calculation, see Parlett [107] and Komzsik [77], can be seen as special cases of this general scheme.
- The numerical complexity of the required operations is outlined in great detail in chapter 7 in order to estimate the numerical costs of the ARM method *a priori*.
- The last chapter 8 demonstrates the usage of the ARM method. Large scaled real world problems are addressed, demonstrating accuracy and efficiency. The approach is demonstrated with design variables typical for multi-criteria and multidisciplinary optimizations. Based on an example, a comparison between the classical *response surface method* (RSM) and the *approximate response manifold* (ARM) approach is given. It is shown that the ARM approach can be interpreted as a *physical based* response surface instead of the most widely used (at maximum) second order Taylor series based mathematical approaches for response surfaces.

Special emphasis is put on the flexible use of existing FE programs. The formulation requires mainly, in most programs existing, mathematical operators such as forward-backward substitutions and decompositions. The data types are pure matrix and tensor based and the scheme can be implemented in general FE programs which have a clean separation between mathematical routines and finite element based modules. Therefore the approach is coherent with the requirement of the finite element method to be a flexible and modular approach for solving structural mechanic problems.

2 General demands for multi criteria and multidisciplinary optimization

The mathematical formulation of a multi-criteria parametric problem is introduced. Typical questions in engineering problems occurring in the automotive industry are presented and mathematically formulated. A small example is introduced, which will be used throughout this thesis, on which the formulation is demonstrated and the numerical approaches are applied later on.

2.1 Mathematical formulation

The general theory of mathematical optimization, as defined for example in Baier [8] or Eschenauer [22], bases on the following definitions:

$$\boldsymbol{\lambda} \text{ is a d-dimensional vector}, \quad \boldsymbol{\lambda} \in D \text{ and } D \subseteq \mathbb{R}^d \text{ open.} \tag{2.1}$$

$$f, h_i : D \to \mathbb{R}, \quad i = 1, \dots, m+p \quad \text{are real functions.} \tag{2.2}$$

The problem formulation looks like this:

$$\text{Minimize } f(\boldsymbol{\lambda}) \tag{2.3}$$

$$\text{with the constraints} \tag{2.4}$$

$$h_i(\boldsymbol{\lambda}) \le 0, \qquad \text{for } i = 1, \dots, m \tag{2.5}$$

$$h_i(\boldsymbol{\lambda}) = 0, \qquad \text{for } i = m+1, \dots, m+p \tag{2.6}$$

We call $f(\boldsymbol{\lambda})$ the objective function. $h_i(\boldsymbol{\lambda})$ the constraints, and

$$R = \{\boldsymbol{\lambda} \quad | \quad h_i(\boldsymbol{\lambda}) \le 0, \text{ for } i = 1, \dots, m;\ h_i(\boldsymbol{\lambda}) = 0, \text{ for } i = m+1, \dots, m+p\} \tag{2.7}$$

the feasible region. $\boldsymbol{\lambda} \in R$ is a feasible solution and

$$\bar{\boldsymbol{\lambda}} \in R, f(\bar{\boldsymbol{\lambda}}) \le f(\boldsymbol{\lambda}), \quad \forall \quad \boldsymbol{\lambda} \in R \tag{2.8}$$

is called the minimum, or global minimum. A local minimum $\bar{\boldsymbol{\lambda}}$ is defined as

$$\exists \quad \epsilon \in \mathbb{R} \quad \ni f(\bar{\boldsymbol{\lambda}}) \le f(\boldsymbol{\lambda}), \quad \forall \quad \boldsymbol{\lambda} \in S \tag{2.9}$$

$$\text{and} \quad S = \{\boldsymbol{\lambda} \quad | \quad \boldsymbol{\lambda} \in R, \|\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}}\| \le \epsilon\} \tag{2.10}$$

To maximize a function $\bar{f}(\boldsymbol{\lambda})$ we can define $f(\boldsymbol{\lambda}) = -\bar{f}(\boldsymbol{\lambda})$.

2.1.1 Multicriteria optimization

The multi-criteria problem can be formulated, see also Das [18], as:

$$F(\boldsymbol{\lambda}) = \begin{bmatrix} k_1(\boldsymbol{\lambda}) \\ k_2(\boldsymbol{\lambda}) \\ \cdots \\ k_n(\boldsymbol{\lambda}) \end{bmatrix} \tag{2.11}$$

$F(\hat{\boldsymbol{\lambda}})$ is said to dominate another $F(\bar{\boldsymbol{\lambda}})$, denoted with $F(\hat{\boldsymbol{\lambda}}) \prec F(\bar{\boldsymbol{\lambda}})$ if and only if $k_i(\hat{\boldsymbol{\lambda}}) \leq k_i(\bar{\boldsymbol{\lambda}})$ for all $i \in \{1, 2, \ldots, n\}$ and $k_j(\hat{\boldsymbol{\lambda}}) < k_j(\bar{\boldsymbol{\lambda}})$ for some $j \in \{1, 2, \ldots, n\}$. A point $\boldsymbol{\lambda}^* \in \boldsymbol{\Lambda}$ is said to be *Pareto optimal* if and only if there does not exist $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$ satisfying $F(\hat{\boldsymbol{\lambda}}) \prec F(\bar{\boldsymbol{\lambda}}^*)$.

2.2 Practical formulation

In order to solve a physical problem numerically we have to describe it mathematically. Therefore we will call the λ_i input parameters or design variables. In the case of a structural mechanical problem these might be cross section values, sheet thicknesses etc.. The upper and lower boundaries on these parameters form $\mathbb{D}$, which is called the *definition space*. Physical properties which are design goals of our problem such as local and global stiffness, durability, etc., will be called output parameters and denoted with $k_i(\boldsymbol{\lambda})$. Also other properties such as manufacturing costs, weight etc. will be denoted as $k_i(\boldsymbol{\lambda})$. Generally we have certain restrictions or targets for the $k_i(\boldsymbol{\lambda})$. Therefore the $k_i(\boldsymbol{\lambda})$ are mapped to the $h_i(\boldsymbol{\lambda}) = h_i(k_j(\boldsymbol{\lambda}))$ and $f(\boldsymbol{\lambda}) = f(h_i(\boldsymbol{\lambda}))$. $\boldsymbol{\lambda}$'s which do not violate any constraints form the *design space*. For certain problems trade-off studies between different k_i are of interest instead of formulating a single target function f. Others just require multidimensional parametric studies to understand relations between input and output parameters or to show up the continuous evolvement from different points in the design space.

2.3 Numerical approaches and requirements

A lot of different numerical procedures exist to solve the optimization problem defined in 2.1 (see for example Goldberg [33], Ingber [50] and [51], Fletcher and Reeves [26], Schitkowski [116]). They all are suitable for different kinds of problems, such as problems for convex design spaces, global or local search, linear and/or nonlinear constraints, continuous or discrete design variables and so on. They all have in common that they need a lot of function evaluations to derive the $k_j(\boldsymbol{\lambda})$. In many times the required function evaluations depend extremely on the number of design variables. The in many cases least demanding optimization strategy concerning the required function evaluations uses sensitivity information. Calculating the sensitivity, if done by forward-/backward- or central-difference, would require an amount of simulations proportional

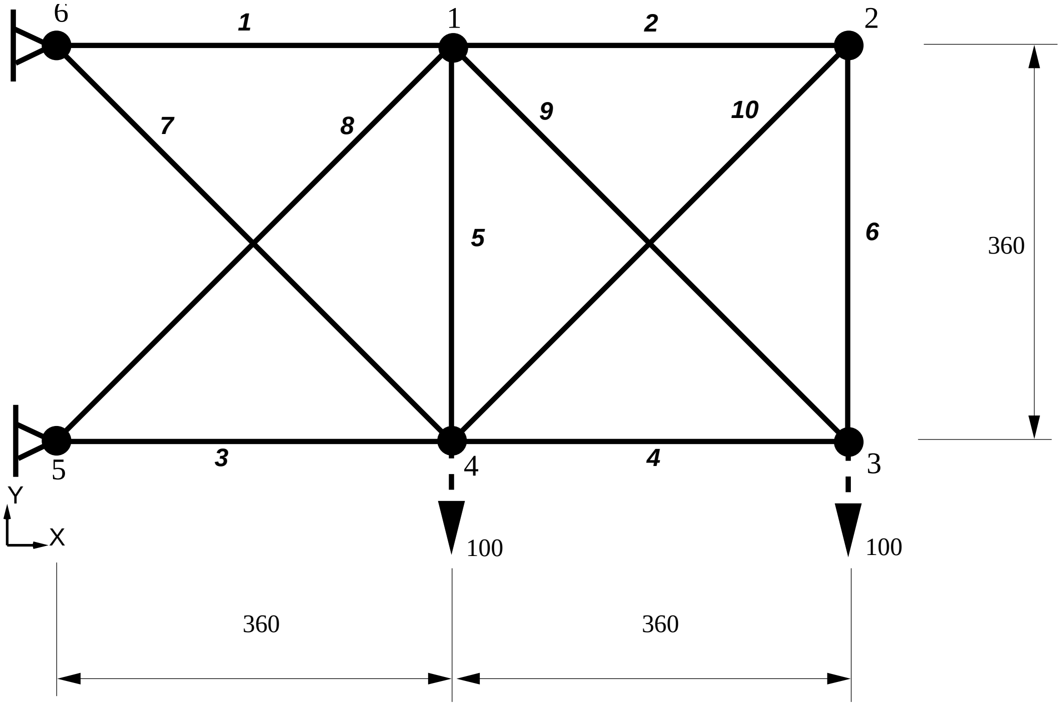


Figure 2.1: Ten-bar truss problem. *Italic* numbers indicate element identifiers, the others are node numbers.

to the number of design variables. Other search strategies can require significantly more function evaluations. So the time for the function evaluation of the k_i often becomes a bottleneck.

2.4 The classical ten-bar truss problem

Let's look at the structure in Fig. 2.1. The structure has cross section areas of 1.0. The Young's modulus is 30,000. The structure is loaded with two nodal forces at the grid points number 3 and 4 in vertical direction. The goal is to reduce the weight of the structure by changing the cross sections of the trusses. As constraints the displacements in vertical direction (coordinate y) of points 3 and 4 should not increase and the maximum elastic strain in all members (1 to 10) should be below 1%.

A mathematical formulation would then look like:

$$\boldsymbol{\lambda} \in \mathbb{R}^{10} : \quad \text{design-variables, 10 cross sections} \tag{2.12}$$

$$f(\boldsymbol{\lambda}) : \quad \text{weight of the structure} \tag{2.13}$$

Minimize $f(\boldsymbol{\lambda})$ with the constraints

$$h_i(\boldsymbol{\lambda}) = -\lambda_i \leq 0, \quad \text{for } i = 1, \ldots, 10 \quad \text{design variable constraints} \tag{2.14}$$

$$h_i(\boldsymbol{\lambda}) \leq 0, \quad \text{for } i = 11, \ldots, 21 \quad \text{strain constraints} \tag{2.15}$$

$$h_i(\boldsymbol{\lambda}) \leq 0, \quad \text{for } i = 22 \quad \text{displacement constraint for degree of freedom 3-y} \tag{2.16}$$

$$h_i(\boldsymbol{\lambda}) \leq 0, \quad \text{for } i = 23 \quad \text{displacement constraint for degree of freedom 4-y} \tag{2.17}$$

This model will be used throughout the rest of the thesis to demonstrate some of the introduced topics.

3 Geometry and kinematics of bodies

The basic concepts of motion and elasticity are introduced using the abstract mathematical formulation of tensors and manifolds. The outline is based on the formulations of Marsden and Hughes [86], and Scheck [115]. The general concept of the manifold, introduced in 3.1 is essential for understanding the mathematical structure of the parametric solution which is embedded in an abstract, higher dimensional mathematical space. The space of all displacement degrees of freedom.

In the second part 3.2 the different configurations as well as motion and deformation specific variables are introduced. This part is not essential for understanding the core of the thesis so it can be skipped by the impatient reader. The interested reader is encouraged to immerse himself in a complete and more exact derivation in the specific literature such as Marsden and Hughes [86].

The introductory concept is oriented on the formulation of Haenle [35].

In continuum mechanics the word *kinematics* describes the deformation and motion of a body. Such a body or continuum fills the open set $B \in \mathbb{R}^3$ of the Euclidean space with material particles.

3.1 Abstraction of Euclidean space

3.1.1 Concept of manifolds

A *manifold* is the abstract mathematical generalization of a curve or a surface in the three dimensional space. A d-dimensional manifold M^d is essentially a space which is locally similar to Euclidean space in a way that it can be covered by coordinate patches in $\mathbb{R}^d$ [37]. The only concepts defined by the manifold structure are those which are independent of the choice of a coordinate system. The abstraction gets rid of the embedding space and describes lower dimensional objects, such as curves or surfaces in $\mathbb{R}^3$, independent from the embedding space. In the context with manifolds the mapping onto the parameter space is called a chart or coordinate system. A set of C^r-differentiable charts covering the whole manifold is called a C^r-*atlas*.

3.1.2 Vectors and tensors

The tangent space T_xM^d of a d-dimensional manifold at a point $x \in M^d$ is spanned by the tangent vectors on M^d at x. A *tangentvector* v on M^d in $x \in M^d$ is a real function

$$v : \mathcal{F}(M^d) \to \mathbb{R} \tag{3.1}$$

with the properties

$$v_x(\alpha f + \beta g) = \alpha v_x(f) + \beta v_x(g) \qquad \text{R-linearity} \tag{3.2}$$

$$v_x(fg) = f(x)v_x(g) + g(x)v_x(f) \qquad \text{Leibniz rule} \tag{3.3}$$

Where $f, g \in \mathcal{F}(M)$ and $\alpha, \beta \in \mathbb{R}$. v_x is the tangent vector on M^d at point x. A linear function fulfilling Eq. (3.2) and Eq. (3.3) is called a *derivation*. Partial derivatives of a function $G \in \mathcal{F}(M)$ on M^d cannot be build in general, but for the image of g in local charts. Lets assume (ϕ, U) is a chart, $p \in U$ a point of M^d and g a function on M^d. Then the derivative of $g \circ \phi^{-1}$ is well defined:

$$\partial_i\Big|_p(g) \equiv \frac{\partial g}{\partial x^i}\Big|_p := \frac{\partial(g \circ \phi^{-1})}{\partial f^i}(\phi(p)) \tag{3.4}$$

The functions

$$\partial_i\Big|_p \equiv \frac{\partial}{\partial x^i}\Big|_p : \mathcal{F}(M) \to \mathbb{R} : g \to \frac{\partial g}{\partial x^i}\Big|_p, i = 1, 2, \ldots, n \tag{3.5}$$

have the properties Eq. (3.2) and Eq. (3.3) and are therefore tangent vectors at M in point $p \in U \subset M$. With Eq. (3.5) we gain two things. First we can define derivatives of smooth functions g on M by projecting G via charts on an Euclidean space. Second we have a basis of the tangent space T_pM and every vector v in M has the *natural basis*

$$v = \sum_{i=1}^{n} v(x^i)\partial_i \tag{3.6}$$

The dual vector space to the tangent space T_pM in p is the *cotangent space* T_p^*M. It is spanned by its natural co-basis, which offers the following relationship

$$dx^i\Big|_p(\partial_j\Big|_p) = \delta^i_j, \quad \delta^i_j : \text{Kronecker-Delta} \tag{3.7}$$

The vectors of the cotangent space T_p^*M are called covariant vectors, differential forms of degree 1, 1-*forms* or Pfaff forms. A contravariant vector is also called a tensor of type $\binom{1}{0}$, while a covariant vector is called a tensor of type $\binom{0}{1}$. Multilinearforms

$$\boldsymbol{A} = A^{i_1,\ldots,i_p}_{j_1,\ldots,j_q}(\partial_{i_1} \otimes \ldots \otimes \partial_{i_p} \otimes dx^{j_1} \otimes \ldots \otimes dx^{j_q}) \tag{3.8}$$

having the type $\binom{p}{q}$ are called *contravariant* of rank p and *covariant* of rank q. Tensors not only defined in one point on the manifold but on the whole manifold are called *tensor fields.* $\otimes$ is the *tensor product* also called *outer product.* Further on we will also use Einstein summation convention. If a sub/super script appears twice in a product, that double-appearance implies a summation over that index:

$$a^jb_j = \sum_j a^jb_j \tag{3.9}$$

3.1.3 Riemannian manifold

A differential manifold M is called Riemannian if it has a continuous $\binom{0}{2}$ tensor field $\boldsymbol{g}$ with the following properties:

- $\boldsymbol{g}$ is symmetric
- $\boldsymbol{g}(v, v) > 0$, $v \in T_x M^n$ if $v \neq 0$

The metric tensor $\boldsymbol{g}$ defines a scalar product $< \cdot, \cdot >$ on $T_x M^n$:

$$< v, w >= g_x(v, w) \quad \forall v, w \in T_x M^n \tag{3.10}$$

3.1.4 Maps of manifolds

Let X and Y be manifolds and

$$\phi : X \to Y \tag{3.11}$$

a continuous map. ϕ is called a C^r*-diffeomorphism* if ϕ and ϕ^{-1} are C^r continuous (see also Hawking [37]). For each scalar field g on Y the mapping ϕ defines the function $\phi^* g$ on X as the function whose value at the point x of X is the value of g at $\phi(x)$:

$$\phi^* g = g(\phi(x)) \tag{3.12}$$

$\phi^* g$ is called the *pull-back* of g to X under ϕ. Thus when ϕ maps points from X to Y, ϕ^* maps functions linearly from Y to X. With ϕ_* we denote a linear operator (*push-forward*)

$$\phi_*(x) : T_x X \to T_{\phi(x)} Y \tag{3.13}$$

3.2 Motion and deformation of a body on $\mathbb{R}^3$

3.2.1 Configurations

A *body* is an open set $\mathcal{B} \in \mathbb{R}^3$. A *configuration* of $\mathcal{B}$ is a mapping $\phi : \mathbb{R}^3 \supset \mathcal{B} \to \mathcal{D} \subset \mathbb{R}^3$. The configuration represents a deformed state of the body [86]. Therefore we differ between the *reference configuration* $\phi =$ **identity** and the *physical configuration*. We will denote the coordinates of points in $\mathcal{B}$ with capital letters $X^i (i = 1, 2, 3)$ and those in the current configuration $\phi(X)$ with lower case letters $x^i (i = 1, 2, 3)$.

The displacements $\boldsymbol{u}$ are then denoted with:

$$\boldsymbol{u}(\boldsymbol{X}, t) = \boldsymbol{x}(\boldsymbol{X}, t) - \boldsymbol{X} \tag{3.14}$$

3.2.2 Deformation gradient

The components of the *deformation gradient* F is defined via

$$F^a{}_b = \frac{\partial \phi^a}{\partial X^b} = \frac{\partial x^a}{\partial X^b} \tag{3.15}$$

Using Eq. (3.14) we get:

$$\frac{\partial x^a}{\partial X^b} = \frac{\partial X^a}{\partial X^b} + \frac{\partial u^a}{\partial X^b} \tag{3.16}$$

$$\leftrightarrow \frac{\partial x^a}{\partial X^b} = \delta^a{}_b + \frac{\partial u^a}{\partial X^b} \tag{3.17}$$

$$\leftrightarrow \frac{\partial u^a}{\partial X^b} = \frac{\partial x^a}{\partial X^b} - \delta^a{}_b \tag{3.18}$$

3.2.3 Strain tensors

Although the deformation gradient implies the whole information of the motion of all particles in our body, he is not useful for measuring strain. Rigid body rotations are contained within the deformation gradient but do not contribute to the strain measurement. We assume that ϕ is a one-to-one mapping, in mathematical terms bijective, and therefore the inverse ϕ^{-1} exists (our deformation can be undone). We further assume that $det(\phi) > 0$ (which implies together with the bijectivity of ϕ that our mapping can be differentially small, otherwise we have to cross $det(\phi) = 0$ if we start from the *identity* mapping $det(\phi) = 1$). The *right Cauchy-Green-Tensor* is defined via

$$\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F} \quad \text{positive definite and symmetric} \tag{3.19}$$

$$\tag{3.20}$$

in components:

$$C_{ab} = F_{ja} F^j{}_b \tag{3.21}$$

$$\leftrightarrow C_{ab} = g_{ij} F^i{}_a F^j{}_b \tag{3.22}$$

$$\tag{3.23}$$

The *material* (Lagrange) or *Green strain tensor* E is defined then by

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - 1) \tag{3.24}$$

$$E^{ab} = \frac{1}{2}(C_{ab} - \delta_{ab}) \tag{3.25}$$

3.2.4 Conservation of Mass

In classical mechanics the volume of a body B may change, but not its mass. Therefore we have for the mass:

$$m = \int_B \rho_{ref} dV = \int_{\phi(B)} \rho dv \tag{3.26}$$

Where ρ_{ref} is the *mass density* of the reference configuration and ρ the mass density of the current configuration. Focusing on the infinitesimal volume element dV of the reference configuration and dv of the current one, we obtain using the transformation rule

$$dv = JdV \tag{3.27}$$

with the (always positive) *Jacobian*

$$J = \det(F)\frac{\sqrt{\det(g(x))}}{\sqrt{\det(G(X)}} \tag{3.28}$$

Combining Eq. (3.26) and Eq. (3.27) we get the correlation between the mass densities of the two configurations

$$\rho_{ref} = J\rho \tag{3.29}$$

3.2.5 Velocity and acceleration

A motion of a body $\mathcal{B}$ is a mapping $t \rightarrow \phi_t(X) = \phi(X,t), X \in \mathcal{B}, \quad t \in \mathbb{R}$. The map starting from the reference configuration $V_t : \mathcal{B} \rightarrow \mathbb{R}^3$

$$V_t(X) = V(X,t) = \frac{\partial \phi(X,t)}{\partial t}, \quad V_t(X) \in T_X M \tag{3.30}$$

is called the *material velocity* of the motion. Because X is defined on whole $\mathcal{B}, \quad V_t(X)$ defines a vector field. According to Eq. (3.30) the *material acceleration* is defined by

$$A_t(X) = A(X,t) = \frac{\partial}{\partial t}V(X,t) = \frac{\partial^2 \phi(X,t)}{\partial t^2}, \quad A_t(X) \in T_X M \tag{3.31}$$

A Map starting from the current configuration $\phi_t(\mathcal{B}) \rightarrow \mathbb{R}^3$ follows the movement of the body in space. The *spatial velocity* v_t is defined by

$$v_t : \phi_t(\mathcal{B}) \rightarrow \mathbb{R}^3, \quad v_t = V_t \circ \phi_t^{-1}, \quad v_t \in T_x M \tag{3.32}$$

$$V_t(X) = v_t(x) \tag{3.33}$$

For the *spatial acceleration* a_t we have

$$a_t : \phi_t(\mathcal{B}) \rightarrow \mathbb{R}^3, \quad a_t = A_t \circ \phi_t^{-1}, \quad a_t \in T_x M \tag{3.34}$$

$$\tag{3.35}$$

It is derived from the spatial velocity v_t using the *covariant derivative* ∇_v

$$a_t = \frac{D}{Dt}v = \frac{d}{dt}v\Big|_X = \frac{\partial v(x,t)}{\partial t} + \nabla_v v \tag{3.36}$$

$$\nabla_v = (\frac{\partial v^a}{\partial x^b}v^b + {\Gamma^a}_{bc}v^b v^a)\frac{\partial}{\partial x^a} \tag{3.37}$$

$${\Gamma^a}_{bc} \quad \text{is the Christoffel symbol} \tag{3.38}$$

4 From the variational formulation to the discretized equations of motion

In this chapter the equations of motion for structural dynamic systems using a variational formulation are derived, based on the Hamiltonian principle. The resulting Lagrange equations, basing on the conservative energies and nonconservative virtual work, introduced at the beginning of this chapter, lead to a weak form of equilibrium representing the equations of motion.

At the end of this chapter general coordinates holding the discretization degrees of freedom resulting from simplification of the underlying geometry are introduced and the equations of motion are formulated for numerical analysis.

The core of the thesis focuses on structural optimization concerning load cases which assume a linear structural behavior. However this introductory chapter starts with the general nonlinear structural continuum. For the interested reader this introduction might be too rough and partly incomplete, while the impatient reader who wants to get to the core of the thesis quickly is bothered with the nonlinear overhead. The interested reader is encouraged to get a more detailed and exact description in literature such as MARSDEN and HUGHES [86]. The impatient one can skip forward to sections 4.2 and 4.4 to get the basics necessary for the next chapters.

4.1 Energy functions of deformable bodies

In this section the kinetic and the strain energy of a continuum are formulated. The term kinetic energy describes the macroscopic observable energy of the body. To calculate the strain energy the Green-Lagrange strain tensor, earlier defined in Eq. (3.25), and a conjugated stress tensor are needed.

4.1.1 Stress tensors and constitutive relation

According to the *stress principle of Cauchy* (see MARSDEN AND HUGHES [86]) there is a time dependent vector field $t(x, t, n)$ defined on any closed surface within a body or on the bounding surface which counteracts in terms of force and moment an external force. $t(x, t, n)$ represents a force per unit area on a surface element with normal n.

Based on *Cauchy's theorem* a unique, symmetric tensor field $\sigma(x,t)$ called *Cauchy stress tensor* is defined, such that

$$\boldsymbol{t}(x,t,n) = < \boldsymbol{\sigma}(x,t), \boldsymbol{n} > \tag{4.1}$$

$$t^a = \sigma^{ai} g_{ij} n^j = \sigma^a_i n^i \tag{4.2}$$

$$\tag{4.3}$$

The *first Piola-Kirchhoff stress tensor* P is obtained by performing a Piola transformation on the second index of σ:

$$P^{ab} = J (F^{-1})^a{}_i \sigma^{bi} \tag{4.4}$$

P is not symmetric and J is the Jacobi determinant defined in Eq. (3.28). The *second Piola-Kirchhoff stress tensor* S is obtained by *pulling* the first leg of P *back* by ϕ_t:

$$S^{ab} = (F^{-1})^a{}_i P^{ib} = J (F^{-1})^a{}_i \sigma^{ij} (F^{-1})^b{}_j \tag{4.5}$$

The *elasticity tensor* $\mathcal{C}$ is derived from the second Piola-Kirchhoff strain tensor S and the right Cauchy-Green tensor C Eq. (3.25):

$$\mathcal{C}^{abcc} = \frac{\partial S^{ab}}{\partial C^{cd}} \tag{4.6}$$

It holds the following symmetry, due to the symmetry of $\boldsymbol{E}$ (Eq. (3.23)) and $\boldsymbol{S}$ Eq. (4.5).

$$\mathcal{C}^{abcd} = \mathcal{C}^{bacd} = \mathcal{C}^{abdc} \tag{4.7}$$

For *hyper elastic materials* (see also MARSDEN and HUGHES[86]) there is an additional symmetry:

$$\mathcal{C}^{abcd} = \mathcal{C}^{cdab} \tag{4.8}$$

4.1.2 Strain energy

Integrating over the total volume of a body the total strain energy V is obtained by

$$V = \frac{1}{2} \int_V S^{ij} E_{ij} \, dV \tag{4.9}$$

$$V = \frac{1}{2} \int_V E_{ij} \mathcal{C}^{ijkl} E_{kl} \, dV \tag{4.10}$$

where E_{ab} is the *Green-Lagrange strain tensor* and $\mathcal{C}^{abcd}$ the *elasticity tensor.*

4.1.3 Kinetic energy

The kinetic energy T of a body is given by

$$T = \frac{1}{2} \int_V \rho v^a v_a \, dV \tag{4.11}$$

where ρ is the density of the body and $\boldsymbol{v}(X,t) = \dot{\boldsymbol{x}}(X,t)$ the velocity of the body.

4.2 Hamiltonian principle

The *Hamiltonian principle of least action* states that for a conservative system the *action integral* $S : C^1([t_1, t_2]) \Rightarrow \mathbb{R}$ is stationary (see Abraham and Marsden [2], Scheck [115] and Landau [79]):

$$S(\boldsymbol{q}) = \int_{t_1}^{t_2} L\left(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), t\right) dt \quad , \boldsymbol{q} \in C^1([t_1, t_2]) \tag{4.12}$$

$$L : \quad \text{Lagrange function} \tag{4.13}$$

$$\boldsymbol{q} := [q_1, \ldots, q_N] \quad \text{vector of generalized coordinates} \tag{4.14}$$

$$[t_1, t_2] : \quad \text{time interval} \tag{4.15}$$

If nonconservative terms, such as external forces or damping forces occur they have to be added as an external work to the Lagrangian (Clough and Penzien [12]):

$$S(\boldsymbol{q}) = \int_{t_1}^{t_2} \left(L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), t) + W_{noncon}\right) dt \quad , q \in C^1([t_1, t_2]) \tag{4.16}$$

$$W_{noncon} : \quad \text{nonconservative work such as done by external forces or damping forces} \tag{4.17}$$

$$\boldsymbol{q} := [q_1, \ldots, q_N] \quad \text{vector of generalized coordinates} \tag{4.18}$$

The condition of stationarity states that the variation of the action integral is zero and leads to the Euler-Lagrange equations:

$$\delta S(\boldsymbol{q}) = 0 \tag{4.19}$$

$$\Longrightarrow \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial L}{\partial \boldsymbol{q}} - \boldsymbol{Q}_{noncon} = 0 \tag{4.20}$$

$$\text{with} \quad \delta W = \boldsymbol{Q}_{noncon} \delta \boldsymbol{q} \tag{4.21}$$

$$\boldsymbol{Q}_{noncon} = [Q_{1\,noncon}, \ldots, Q_{N\,noncon}] \quad \text{vector of generalized, non conservative forces} \tag{4.22}$$

This system of differential equations is a weak form of equilibrium. The vector $\boldsymbol{q}$ represents the generalized degrees of freedom, which have, depending on the formulation, different meaning. For a structural problem, the natural form of the Lagrangian is given by (see Scheck [115]):

$$L = T - V \tag{4.23}$$

$$T : \text{kinetic energy} \tag{4.24}$$

$$V : \text{strain energy} \tag{4.25}$$

4.3 Spatial discretization of the Lagrangian

Arbitrary shaped domains are spatially discretized for solving the corresponding problems practically. The resulting time and space depended nodal values, for mixed formulations also stress values, have in general, despite to the more global approximation functions as in Rayleigh and Ritz (see Ritz [113]), only local supports. See also Bathe [10] or Zienkiewicz and Taylor [125],[126].

4.3.1 Kinematic assumptions

The discretization will base on the following assumptions:

- The motion of a body can result in large displacements, but the resulting strain and the displacement derivatives and rotations will stay small.
- Further it is assumed that the material is elastic, based on a generalized Hooke's law. This means that there is a linear constitutive equation relating strain and stress.
- Furthermore hypotheses for a reduction of the continuum will be allowed. This can include the assumption of grave rigid bodies, one dimensional spring systems, as well as the Euler-Bernoulli- or Timoshenko-beam theory, and shell theory. Using the beam theory the mapping of the continuum leads to one-dimensional reference curves assuming plane cross sections. In the shell theory using for example the Kirchhoff-Love-hypothesis the continuum is mapped to a reference plane (shell mid surface).

4.3.2 Linearization

Generally nonlinear problems are not solvable in a one-step approach. The established procedure is an iterative one. The standard approach for such systems is for example the Newton-Raphson procedure (see also also Bathe [10] or Zienkiewicz and Taylor [125],[126]) which can guarantee quadratic convergence if the corresponding analytical derivatives exist. In these procedures each iterative step bases on a linearization of the current configuration. This involves the introduction of the *tangent stiffness matrix*, which will be defined in the next subsections (see Eq. (4.67)).

The focus in this thesis is on such a linearized step only, or more concrete, an underlying purely *linear model* is assumed.

Subject of the linearization is the strain energy given in Eq. (4.9).

4.3.2.1 Material linearization

The constitutive relation (see Mang [85]):

$$E_{ab} = \mathcal{G}(S_{cd}) \tag{4.26}$$

basically couples the strain and the stress. Assuming an ideal elastic material 4.3.1 the relation

$$E_{ab} = D^{abij} S_{ij} \tag{4.27}$$

holds, where $\boldsymbol{D}$ is the flexibility tensor. Using the *elasticity tensor* $\boldsymbol{\mathcal{C}}$ as the inverse of the flexibility tensor $\boldsymbol{D}$ *second Piola-Kirchhoff stress tensor* can be written as

$$S_{ab} = \mathcal{C}^{abij} E_{ij} \tag{4.28}$$

where the *elasticity tensor* $\mathcal{C}$ does not depend on the *Green strain tensor* E and contains only constants such as the Young's modulus and Poisson's ratio. This converts Eq. (4.9) to

$$V = \frac{1}{2} \int_V E_{ij} \mathcal{C}^{ijkl} E_{kl} \, dV \tag{4.29}$$

As seen in Eq. (4.7) and following the 4-th order tensor $\boldsymbol{\mathcal{C}}$ holds multiple symmetries. This reduces the amount of independent components as well.

The first symmetry, due to the symmetry of $\boldsymbol{E}$, reduces the 81 components to 54.

The second symmetry, due to the symmetry of $\boldsymbol{S}$, reduces them further to 36.

The third symmetry, due to the choice for a hyper elastic material, finally reduces them further to 21 independent components.

4.3.2.2 Geometric linearization

Inserting the definition of the *right Cauchy-Green-Tensor* Eq. (3.23), the *Green strain tensor* Eq. (3.25) becomes:

$$E_{ab} = \frac{1}{2} (g_{ij} F^i{}_a F^j{}_b - \delta_{ab}) \tag{4.30}$$

and with the *deformation gradient* Eq. (3.15):

$$E_{ab} = \frac{1}{2} \left(g_{ij} \frac{\partial x^i}{\partial X^a} \frac{\partial x^j}{\partial X^b} - \delta_{ab} \right) \tag{4.31}$$

Using Eq. (3.17) this leads to:

$$\begin{aligned}
E_{ab} &= \frac{1}{2} \left[g_{ij} \left(\delta^i{}_a + \frac{\partial u^i}{\partial X^a} \right) \left(\delta^j{}_b + \frac{\partial u^j}{\partial X^b} \right) - \delta_{ab} \right] && (4.32)\\
&= \frac{1}{2} \left[g_{ij} \left(\delta^i{}_a \delta^j{}_b + \delta^j{}_b \frac{\partial u^i}{\partial X^a} + \delta^i{}_a \frac{\partial u^j}{\partial X^b} + \frac{\partial u^i}{\partial X^a} \frac{\partial u^j}{\partial X^b} \right) - \delta_{ab} \right] && (4.33)\\
&= \frac{1}{2} \left[\delta_{ja} \delta^j{}_b + \delta_{ib} \frac{\partial u^i}{\partial X^a} + \delta_{ja} \frac{\partial u^j}{\partial X^b} + \frac{\partial u_j}{\partial X^a} \frac{\partial u^j}{\partial X^b} - \delta_{ab} \right] && (4.34)\\
&= \frac{1}{2} \left[\delta_{ab} + \frac{\partial u^b}{\partial X^a} + \frac{\partial u^a}{\partial X^b} + \frac{\partial u_j}{\partial X^a} \frac{\partial u^j}{\partial X^b} - \delta_{ab} \right] && (4.35)\\
&= \frac{1}{2} \left[\frac{\partial u^b}{\partial X^a} + \frac{\partial u^a}{\partial X^b} + \frac{\partial u_j}{\partial X^a} \frac{\partial u^j}{\partial X^b} \right] && (4.36)
\end{aligned}$$

The linearization of the strain tensor $\boldsymbol{E}$ uses the assumption of small displacement derivatives in 4.3.1 (see Mang [85]):

$$\frac{\partial u_i}{\partial X_j} = \frac{\partial u_i}{\partial x_k}\frac{\partial x_k}{\partial X_j} \tag{4.37}$$

$$\leftrightarrow \frac{\partial u_i}{\partial X_j} = \frac{\partial u_i}{\partial x_k}(\delta_{kj} + \frac{\partial u_k}{\partial X_j})\,, \quad \text{using Eq. (3.17)} \tag{4.38}$$

$$\leftrightarrow \frac{\partial u_i}{\partial X_j} \approx \frac{\partial u_i}{\partial x_j}\,, \quad \text{for} \quad \frac{\partial u_k}{\partial X_j} \ll 1 \tag{4.39}$$

This linearization scheme transforms the *Green strain tensor* $\boldsymbol{E}$ into the *linearized strain tensor* $\boldsymbol{\varepsilon}$:

$$E_{ij} \longrightarrow \varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{4.40}$$

Exchanging the indices it can be seen that the *linearized strain tensor* is symmetric. Therefore only 6 of the 9 components are independent. The 6x1 *strain vector* $\boldsymbol{\epsilon}$:

$$\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \epsilon_3, \gamma_{12}, \gamma_{23}, \gamma_{31})^T \tag{4.41}$$

$$\boldsymbol{\epsilon} = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, 2\varepsilon_{12}, 2\varepsilon_{23}, 2\varepsilon_{31})^T \tag{4.42}$$

contains only the independent components of $\boldsymbol{\varepsilon}$.

Because the type of the strain object was changed and the indices of the *linearized strain tensor* were rearranged in a way that they fit into a vector this has to be done simultaneously for the *elasticity tensor* in order to keep the strain energy expression invariant. By this transformation the resulting symmetric 6x6 *elasticity matrix* $\bar{\boldsymbol{\mathcal{C}}}$ has only 21 independent components.

Invariance of the strain energy:

$$\frac{1}{2}\int_V \varepsilon_{ij}\mathcal{C}^{ijkl}\varepsilon_{cd}\,dV = \frac{1}{2}\int_V \epsilon_\alpha \bar{\mathcal{C}}^{\alpha\beta}\epsilon_\beta\,dV \tag{4.43}$$

It has to be denoted that Latin lower case indices such as i, j, k represent here the three spatial coordinates and range from $1, 2, 3$ while in this equation Greek lower case indices such as the α, β represent the abstract 6 displacement degrees of freedom (translation and rotation) and range from $1, 2, \ldots, 6$.

4.3.2.3 Summary

The total linearization scheme can be summed up for the strain energy as follows:

$$V = \frac{1}{2}\int_V E_{ab}S^{ab}\,dV \tag{4.44}$$

$$\xrightarrow{\text{material linearization}} \frac{1}{2}\int_V E_{ab}\mathcal{C}^{abcd}E_{cd}\,dV \tag{4.45}$$

$$\xrightarrow{\text{geometric linearization}} \frac{1}{2}\int_V \varepsilon_{ab}\mathcal{C}^{abcd}\varepsilon_{cd}\,dV \tag{4.46}$$

$$\xrightarrow{\text{notation simplification}} \frac{1}{2}\int_V \epsilon_\alpha\bar{\mathcal{C}}^{\alpha\beta}\epsilon_\beta\,dV \tag{4.47}$$

For an isotropic, elastic material for example the *elasticity matrix* $\bar{\boldsymbol{\mathcal{C}}}$ becomes:

$$\bar{\boldsymbol{\mathcal{C}}} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(1-2\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}(1-2\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(1-2\nu) \end{pmatrix} \tag{4.48}$$

Where E is Young's modulus and ν Poisson's ratio.

4.3.3 Discretization using the Rayleigh-Ritz procedure

The different forms of energy described in the previous sections and chapters consist, due to the assumptions, of an infinite number of points defining the space of these bodies. (A molecular or atomic structure is neglected.) This makes it impossible to solve the above equations for complex geometries exactly. Nevertheless approximate solutions are possible. Their geometric approximation error can practically be made as small as desired, limited only by the computing resources. Besides the very general method of the *weighted residuals*, primarily the *Galerkin procedure* (see ZIENKIEWICZ and TAYLOR [125]), the *Rayleigh-Ritz procedure* (see RITZ [113]) is the more suitable for mechanical systems which can be described by the Hamiltonian principle. In the Rayleigh-Ritz procedure the unknowns, in our case the displacements $\boldsymbol{u}(\boldsymbol{X}, t)$ are approximated using a finite sum of *local shape functions* ${N^I}_a$:

$$u_a(\boldsymbol{X}, t) = \sum_{I=1}^{N} q_I(t) {N^I}_a(\boldsymbol{X}) \tag{4.49}$$

$$u_a = q_I {N^I}_a \tag{4.50}$$

Here uppercase indices such as I indicate that:

$$\text{their range is:} \quad I = 1, \ldots, N \quad \text{and not } 1, 2, 3 \text{ as for lowercase ones} \tag{4.51}$$

$$\text{the metric is Euclidian:} \quad g_{IJ} = \delta_{IJ} \tag{4.52}$$

The $\boldsymbol{q}$ are the unknowns and serve as the general coordinates $\boldsymbol{q}$ introduced in Eq. (4.14). This leaves N unknowns which have to be determined. This approach has the further benefit to separate the time from the space dependency. It has to be reminded that the choice (and number) of the *local shape functions* ${N^I}_a$ entirely determines the error of our approach due to geometrical approximation.

4.3.3.1 The finite elements

The Rayleigh-Ritz procedure however does not make any restrictions to the shape functions beside the fact that they should be linear independent. Practically the geometrical body is split into sub-domains also called *finite elements*, for which corresponding solutions exist. These sub-domains mostly are BEAM, TRIANGULAR and QUADRILATERAL, HEXAHEDRAL, PENTAHEDRAL and TETRAHEDRAL bodies out of which any 1-dimensional, 2-dimensional and 3-dimensional body can be approximately built. The according shape functions differ from zero only locally, which simplifies the integration over geometrically complex bodies and leads later on to sparse matrices.

4.3.3.2 Discrete form of kinetic energy

With the relation:

$$\begin{aligned} v_a &= \dot{x}_a && (4.53) \\ &= \dot{u}_a \quad \text{, using Eq. (3.14)} && (4.54) \\ & && (4.55) \end{aligned}$$

Eq. (4.11) gives after inserting Eq. (3.14) and Eq. (4.49) (see also Bathe [10] or Zienkiewicz and Taylor [125],[126]):

$$\begin{aligned} T &= \frac{1}{2}\int_V \rho \dot{q}^I N_{Ia} \dot{q}_J N^{Ja}\, dV && (4.56) \\ &= \frac{1}{2}\dot{q}^I \dot{q}_J \int_V \rho N_{Ia} N^{Ja}\, dV && (4.57) \\ &= \frac{1}{2}\dot{q}^I \dot{q}^J M_{IJ} && (4.58) \end{aligned}$$

using the definition of the *mass matrix* $\boldsymbol{M}$:

$$M_{IJ} = \int_V \rho N_{Ia} N^{Ja}\, dV \tag{4.59}$$

4.3.3.3 Discrete form of strain energy

The 6x3 differential operator $\hat{\boldsymbol{B}}$ is used to express the strain energy in a similar way: $\hat{\boldsymbol{B}}$:

$$\hat{\boldsymbol{B}} = \begin{pmatrix} \partial_{X_1} & 0 & 0 \\ 0 & \partial_{X_2} & 0 \\ 0 & 0 & \partial_{X_3} \\ \partial_{X_2} & \partial_{X_1} & 0 \\ 0 & \partial_{X_3} & \partial_{X_2} \\ \partial_{X_3} & 0 & \partial_{X_1} \end{pmatrix} \quad \text{, with} \quad \partial_{X_a} = \frac{\partial}{\partial X^a} \tag{4.60}$$

$$\rightarrow \epsilon_\alpha = \hat{B}_\alpha{}^i u_i \tag{4.61}$$

$$\rightarrow \epsilon_\alpha = \hat{B}_\alpha{}^i q_J N^J{}_i \quad \text{, using approximation Eq. (4.49)} \tag{4.62}$$

Inserting Eq. (4.62) into Eq. (4.47) gives: (see also Bathe [10] or Zienkiewicz and Taylor [125],[126]):

$$V = \frac{1}{2} \int_V \hat{B}_K{}^i q_I N^I{}_i \bar{\mathcal{C}}^{KL} \hat{B}_L{}^j q_J N^J{}_j \, dV \tag{4.63}$$

$$= \frac{1}{2} q_I q_J \int_V \hat{B}_K{}^i N^I{}_i \bar{\mathcal{C}}^{KL} \hat{B}_L{}^j N^J{}_j \, dV \tag{4.64}$$

$$= \frac{1}{2} q_I q_J K^{IJ} \tag{4.65}$$

$$= \frac{1}{2} q^I q^J K_{IJ} \quad \text{due to the Euclidean metric} \tag{4.66}$$

where the *stiffness matrix* $\boldsymbol{K}$ is defined by:

$$K_{IJ} = \int_V \hat{B}_K{}^i N^I{}_i \bar{\mathcal{C}}^{KL} \hat{B}_L{}^j N^J{}_j \, dV \tag{4.67}$$

This *stiffness matrix* is identical with the *tangent stiffness matrix* introduced on page 18.

4.3.3.4 Discrete form of nonconservative work

Two types of nonconservative work are considered:

- work created by external forces, $W_{noncon,ext}$
- internal work performed by the system, $W_{noncon,int}$. In general this includes damping forces which dissipate energy.

The variational nonconservative external work shall be defined by:

$$\delta W_{noncon,ext} = \int_V h^i(\boldsymbol{X}, t) \delta x_i \, dV + \int_\Omega p^i(\boldsymbol{X}, t) \delta x_i \, dA \tag{4.68}$$

Where the $h_i(\boldsymbol{X}, t)$ denote volume forces such as gravitational ones and $p_i(\boldsymbol{X}, t)$ denote surface forces (pressure) on the bodies surface. A similar approach for h and p as it was done for $\boldsymbol{x}$ in Eq. (4.49) gives:

$$h_i(\boldsymbol{X}, t) = \sum_{J=1}^{N} H_J(t) N_i^J(\boldsymbol{X}) \tag{4.69}$$

$$p_i(\boldsymbol{X}, t) = \sum_{J=1}^{N} P_J(t) N_i^J(\boldsymbol{X}) \tag{4.70}$$

Inserting Eq. (4.69), Eq. (4.70) and Eq. (4.49) into Eq. (4.68) gives:

$$\delta W_{noncon,ext} = \int_V H_J(t) N^{Ji} \delta(q_I N^I{}_i) \, dV + \int_{\partial V} P_J(t) N^{Ji} \delta(q_I N^I{}_i) \, dS \tag{4.71}$$

$$= H_J \int_V N^{Ji} N^I{}_i \, dV (\delta q_I) + P_J \int_\Omega N^{Ji} N^I{}_i \, dA (\delta q_I) \tag{4.72}$$

assuming ortho-normality between the shape functions

$$= H_J \delta^{IJ} \delta q_I + P_J \delta^{IJ} (\delta q_I) \tag{4.73}$$

$$= (H_I + P_I) \delta q_I \tag{4.74}$$

$$= F_I \delta q_I \tag{4.75}$$

$$\rightarrow Q_{noncon,ext_I} = F_I \tag{4.76}$$

However, dissipated, internal nonconservative work and the responsible damping forces however are difficult to obtain. In most cases a velocity proportional force, basically resulting from a strain rate depended stress tensor, is sufficient (see also Clough and Penzien [12]). Therefore the used approach for the virtual work performed by these forces is:

$$\delta W_{noncon,int} = -\sum_{J=1}^{N} B_{IJ} \dot{q}^J \delta q^I \tag{4.77}$$

$$= -B_{IJ} \dot{q}^J \delta q^I \tag{4.78}$$

$$\rightarrow Q_{noncon,int_I} = -B_{IJ} \dot{q}^J \tag{4.79}$$

with a symmetric damping matrix $\boldsymbol{B}$.

4.4 Equations of motion in matrix notation

Inserting Eq. (4.23), the discretized forms of kinetic energy Eq. (4.58), strain energy Eq. (4.66) and the nonconservative generalized forces Eq. (4.76) and Eq. (4.79) in the variational formulation Eq. (4.20) lead to:

$$\frac{d}{dt}\left(\frac{\partial(\frac{1}{2}\dot{q}^I\dot{q}^J M_{IJ} - \frac{1}{2}q^I q^J K_{IJ})}{\partial \dot{q}_I}\right) - \frac{\partial(\frac{1}{2}\dot{q}^I\dot{q}^J M_{IJ} - \frac{1}{2}q^I q^J K_{IJ})}{\partial q_I} - F_I + B_{IJ}\dot{q}^J = 0 \tag{4.80}$$

These are N equations for all I. Carrying out the differentiation in Eq. (4.80) gives:

$$M_{IJ} \ddot{q}^J + B_{IJ} \dot{q}^J + K_{IJ} q^J = F_I \tag{4.81}$$

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{B}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{F} \quad , \quad \text{symbolic notation} \tag{4.82}$$

If there is a harmonic loading and a steady state can be assumed the equation simplifies further:

$$\boldsymbol{F}(t) = \bar{\boldsymbol{F}} e^{i\omega t} \tag{4.83}$$

$$\xrightarrow{assumption} \boldsymbol{q}(t) = \bar{\boldsymbol{q}} e^{i\omega t} \tag{4.84}$$

$$\dot{\boldsymbol{q}} = i\omega \boldsymbol{q} \tag{4.85}$$

$$\ddot{\boldsymbol{q}} = -\omega^2 \boldsymbol{q} \tag{4.86}$$

$$\tag{4.87}$$

Inserting Eq. (4.83), Eq. (4.85), and Eq. (4.86) into Eq. (4.82) and dividing by $e^{i\omega t} \neq 0$ leads to:

$$-\omega^2 M_{IJ} q^J + i\omega B_{IJ} q^J + K_{IJ} q^J = F_I \tag{4.88}$$

$$-\omega^2 \boldsymbol{Mq} + i\omega \boldsymbol{Bq} + \boldsymbol{Kq} = \boldsymbol{F} \quad , \quad \text{symbolic notation} \tag{4.89}$$

4.4.1 Eigenvalues for low damped dynamic systems and the characteristic equation

Neglecting damping and loading forces Eq. (4.89) results in:

$$-\omega^2 \boldsymbol{Mq} + \boldsymbol{Kq} = 0 \quad , \quad \text{symbolic notation} \tag{4.90}$$

$$[-\lambda \boldsymbol{M} + \boldsymbol{K}]\, \boldsymbol{q} = 0 \quad \boldsymbol{K}, \boldsymbol{M} \in \mathbb{R}^{NxN} \tag{4.91}$$

This is a general eigenvalue problem and Eq. (4.90) is also called the *secular equation*. It has N solutions with corresponding eigenvalues λ_i and eigenvectors $\boldsymbol{q}_i$ also called *natural* or *normal modes*. The eigenvalues as well as the eigenvectors aren't necessarily unique.

4.4.2 System matrices for the ten-bar truss model

Using the integration scheme Eq. (4.59), allowing only translations in the x-y-plane and after eliminating the constraints using the procedure described in the appendix A.4 the following (lumped) mass matrix is obtained for the ten-bar truss model in Fig. 2.1:

$$\boldsymbol{M} = 10^{-6} \begin{pmatrix} 8.20 & & & & & & & \\ & 8.20 & & & & & & \\ & & 4.81 & & & & & \\ & & & 4.81 & & & & \\ & & & & 4.81 & & & \\ & & & & & 4.81 & & \\ & & & & & & 8.20 & \\ & & & & & & & 8.20 \end{pmatrix} \tag{4.92}$$

The same procedure holds for the stiffness matrix when using Eq. (4.67):

$$\boldsymbol{K} = \begin{pmatrix} 225.59 & 0.00 & -83.33 & 0.00 & -29.46 & 29.46 & 0.00 & 0.00 \\ 0.00 & 142.26 & 0.00 & 0.00 & 29.46 & -29.46 & 0.00 & -83.33 \\ -83.33 & 0.00 & 112.80 & 29.46 & 0.00 & 0.00 & -29.46 & -29.46 \\ 0.00 & 0.00 & 29.46 & 112.80 & 0.00 & -83.33 & -29.46 & -29.46 \\ -29.46 & 29.46 & 0.00 & 0.00 & 112.80 & -29.46 & -83.33 & 0.00 \\ 29.46 & -29.46 & 0.00 & -83.33 & -29.46 & 112.80 & 0.00 & 0.00 \\ 0.00 & 0.00 & -29.46 & -29.46 & -83.33 & 0.00 & 225.59 & 0.00 \\ 0.00 & -83.33 & -29.46 & -29.46 & 0.00 & 0.00 & 0.00 & 142.26 \end{pmatrix} \tag{4.93}$$

The load vector, see also Eq. (4.76):

$$\boldsymbol{F} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -100 \\ 0 \\ -100 \end{pmatrix} \tag{4.94}$$

The degrees of freedom in the above mentioned matrices refer to (node id,degree-of-freedom):

$$\text{DOF-label} = \begin{pmatrix} 1, x\text{-displacement} \\ 1, y\text{-displacement} \\ 2, x\text{-displacement} \\ 2, y\text{-displacement} \\ 3, x\text{-displacement} \\ 3, y\text{-displacement} \\ 4, x\text{-displacement} \\ 4, y\text{-displacement} \end{pmatrix} \tag{4.95}$$

Solving the system of equations in the static ($\omega = 0$) case, see Eq. (4.89), the displacement vector becomes:

$$\boldsymbol{x} = \begin{pmatrix} 2.3441 \\ -5.5807 \\ 2.8254 \\ -12.6489 \\ -3.1737 \\ -13.1304 \\ -2.4553 \\ -6.0066 \end{pmatrix} \tag{4.96}$$

The eigenfrequencies

$$f_i = \frac{1}{2\pi}\sqrt{\lambda_i} \tag{4.97}$$

of this model are

mode No. i	eigenvalue λ_i	radian ω_i	eigenfrequency f_i
1	7.935579E+05	8.908187E+02	141.78
2	6.241317E+06	2.498263E+03	397.61
3	6.791647E+06	2.606079E+03	414.77
4	2.260450E+07	4.754419E+03	756.69
5	2.642408E+07	5.140436E+03	818.13
6	3.464264E+07	5.885799E+03	936.75
7	3.485494E+07	5.903808E+03	939.62
8	5.120596E+07	7.155833E+03	1138.89

5 Parameterized equations of motion and their solution manifold

In this chapter the solution manifolds of nonlinear parameter dependent equations will be presented. The term *solution manifold* describes the general concept of a response surface, as it is known in literature. The parameters spanning up this space will be design variables for numerical optimization, reliability analyses, or similar approaches. After the general description the results will be applied on the equations of motion for structural problems, introduced in the previous chapter. It will be shown that for the discretized problems the solution manifold can be approximately embedded in a subspace with lower dimension. Therefore this approach will be called *approximate response manifold*, short ARM.

For analytical operators this subspace is defined by the tangent space $T_{\boldsymbol{x}_0}M^d$ on the manifold, where $\boldsymbol{x}_0$ is the baseline result, and optional higher derivatives. This leads directly to the development of a convergent series of global shape functions, which can be used for an efficient reanalysis.

5.1 Preliminaries and assumptions

A nonlinear parameter dependent system of equations F shall be defined by:

$$F(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{y}_0 \tag{5.1}$$

where F is a mapping from a Banach space $Z = X \times \Lambda$ to a Banach space Y. X represents a state space and Λ an d-dimensional parameter space, with $1 \leq d < \infty$. Depending on certain conditions the solutions of Eq. (5.1) form an d-dimensional manifold.

Since Z and hence X are in general infinite dimensional, any computational examination requires the introduction of finite dimensional approximations. Questions arising about the resulting errors between these solution manifolds and their discretized versions are handled in Fink [25] and Lee [81].

In this thesis it will be assumed that the discretization error introduced is sufficiently small for $\boldsymbol{\lambda} = 0$ (the baseline analysis). Furthermore it will be assumed that $\forall \boldsymbol{\lambda} \in \hat{\Lambda} \subset \Lambda$ the same baseline discretization is *valid* in the engineering sense, or more precisely, sufficient concerning

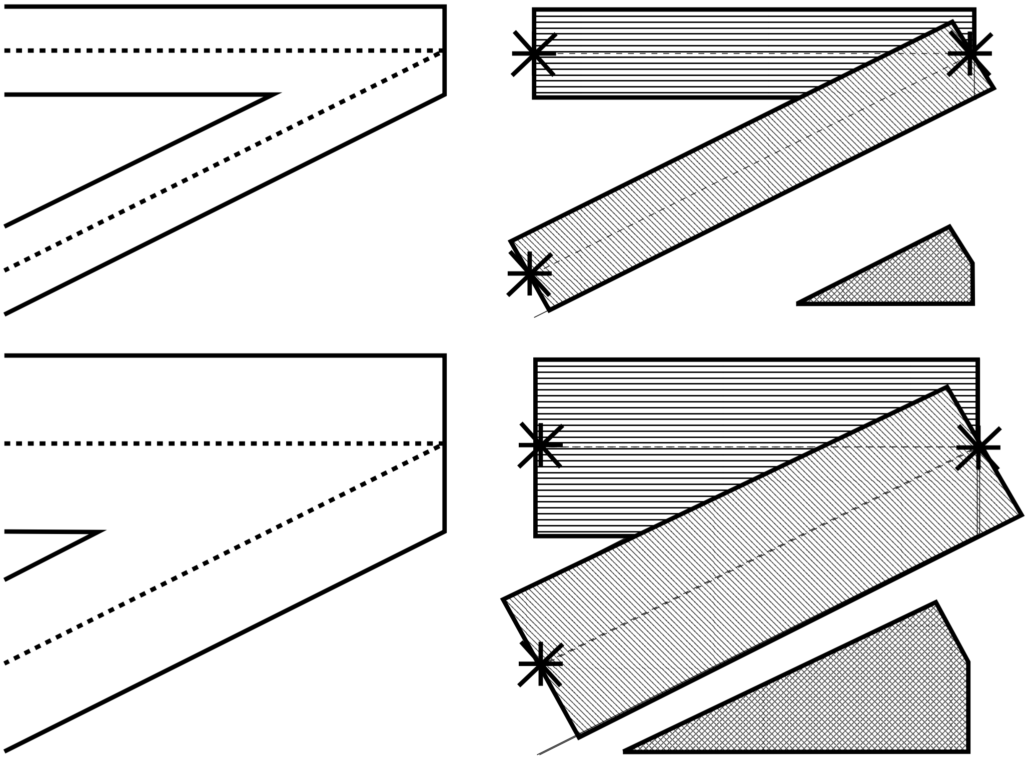


Figure 5.1: Schematic representation of the increase of the discretization/modeling error by increasing the cross section diameter. The left side shows the geometry, while the right side the respective, with BEAM elements discretized FE model. The patches on the right represent the error.

its discretization or modeling error. Obviously this puts restrictions to the *move limit* of the design variables λ_i and therefore defines the circumference of $\hat{\Lambda} \subset \Lambda$.

These assumptions are absolutely not trivial especially when reductions of the continuum are involved as mentioned in 4.3.1 or certain parameters λ_i correspond to geometric properties as they are used in shape optimization. An example can be seen in Fig. 5.1. There, a parameter λ_i represents a cross section diameter. Due to the small angle formed by the two beams a large amount of material is modeled *twice*. This error introduced depends on the size of of the cross section diameter. A parametric change increases or decreases then also the modeling error.

5.2 The solution space properties for the discretized problem

As mentioned above this thesis focuses on the discretized problem. Information about the general solution and its discretization error can be found in Fink and Rheinboldt [23]. The unknowns are the state space variables x_i, $i = 1, \ldots, n$ in the configuration space $\mathcal{N} = \mathbb{R}^n$. Introducing parameters into the equations of motion generally parametries the solution vector:

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\lambda}) \tag{5.2}$$

In general the $\boldsymbol{x}(\boldsymbol{\lambda})$ form a solution manifold. It will be shown now that this solution manifold is a sub manifold of the $\mathbb{R}^n$, where n is the amount of degrees of freedom introduced during the discretization process Eq. (4.49).

Definition 5.1 *(see also [27]) Let $\Lambda \subset \mathbb{R}^d$ be open. A continuous differentiable map*

$$\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n) : \Lambda \to \mathbb{R}^n \tag{5.3}$$

is called an immersion if

$$\textbf{rank}\left(\frac{\partial(x_1, \ldots, x_n)}{\partial(\phi_1, \ldots, \phi_d)}\right)(\boldsymbol{\lambda}) = d \quad \textit{for all } \lambda \in \Lambda \tag{5.4}$$

Annotation: due to the fact that $\textbf{rank}\left(\frac{\partial(x_1,\ldots,x_n)}{\partial(\phi_1,\ldots,\phi_d)}\right)(\boldsymbol{\lambda}) \leq min(d, n)$ *d has to be smaller or equal to n (or $\Lambda = \emptyset$).*

Theorem 5.1 *(see also* Forster *[28]) Let $\Lambda \subset \mathbb{R}^d$ be open and*

$$\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n) : \Lambda \to \mathbb{R}^n \tag{5.5}$$

an immersion of class C^a ($a \geq 1$). Then there exists $\forall c \in \Lambda$ an open set $\hat{\Lambda} \subset \Lambda$ such that $\boldsymbol{\phi}(\hat{\Lambda})$ is a d-dimensional sub manifold of class C^a and

$$\boldsymbol{\phi} : \hat{\Lambda} \to \boldsymbol{\phi}(\hat{\Lambda}) \tag{5.6}$$

is a homomorphism.

The proof is presented by Forster [28].

Theorem 5.2 *The derivative of the inverse of a matrix $\boldsymbol{X}$ is given by:*

$$\partial_\lambda \boldsymbol{X}^{-1} = -\boldsymbol{X}^{-1}(\partial_\lambda \boldsymbol{X})\boldsymbol{X}^{-1} \tag{5.7}$$

$$\textit{where} \quad \partial_\lambda = \frac{\partial}{\partial \lambda} \tag{5.8}$$

The proof is simple:

Proof 5.1

$$\boldsymbol{X}\boldsymbol{X}^{-1} = 1 \tag{5.9}$$

$$\rightarrow \partial_\lambda(\boldsymbol{X}\boldsymbol{X}^{-1}) = 0 \tag{5.10}$$

$$\rightarrow (\partial_\lambda \boldsymbol{X})\boldsymbol{X}^{-1} + \boldsymbol{X}(\partial_\lambda \boldsymbol{X}^{-1}) = 0 \tag{5.11}$$

$$\rightarrow \boldsymbol{X}(\partial_\lambda \boldsymbol{X}^{-1}) = -(\partial_\lambda \boldsymbol{X})\boldsymbol{X}^{-1} \tag{5.12}$$

$$\rightarrow \partial_\lambda \boldsymbol{X}^{-1} = -\boldsymbol{X}^{-1}(\partial_\lambda \boldsymbol{X})\boldsymbol{X}^{-1} \tag{5.13}$$

q.e.d.

The following examples illustrate the abstract theory:

Example 5.1 *Look at the ten-bar truss model Fig. 2.1. It consists only of truss elements under a static load. From Eq. (4.89) the following system of equations describing the deflection of the nodes is obtained:*

$$K(\boldsymbol{\lambda})_{ij}x_j = F_i \tag{5.14}$$

$$x_j = K(\boldsymbol{\lambda})^{-1}_{ij}F_i \tag{5.15}$$

where $i = 1, \ldots, n$, *and* $k = 1, 2$. *Now let* λ_1 *represent the cross section area* (A_1) *and* λ_2 *represent the Young's modulus* (E_1) *of truss number 1. Truss elements have only membrane stiffness* k^m *which is proportional to the product of their cross section area and their Young's modulus:*

$$k^m_1 \approx E_1 A_1 \quad \rightarrow \tag{5.16}$$

$$\frac{\partial}{\partial E_1} \approx \frac{\partial}{\partial k^m_1} \tag{5.17}$$

$$\frac{\partial}{\partial A_1} \approx \frac{\partial}{\partial k^m_1} \tag{5.18}$$

$$\tag{5.19}$$

The gradients

$$\frac{\partial(x_1, \ldots, x_n)}{\partial \lambda_1}, \frac{\partial(x_1, \ldots, x_n)}{\partial \lambda_2} \tag{5.20}$$

are then linear dependent :

$$\frac{\partial(x_1, \ldots, x_n)}{\partial \lambda_1} \approx \frac{\partial(x_1, \ldots, x_n)}{\partial k^m_1} \tag{5.21}$$

$$\frac{\partial(x_1, \ldots, x_n)}{\partial \lambda_2} \approx \frac{\partial(x_1, \ldots, x_n)}{\partial k^m_1} \tag{5.22}$$

$$\rightarrow \mathbf{rank}\left(\frac{\partial(x_1, \ldots, x_n)}{\partial \lambda_1}, \frac{\partial(x_1, \ldots, x_n)}{\partial \lambda_2}\right) = 1 \tag{5.23}$$

Therefore the solution manifold is only 1-dimensional. This means that any change of the solution vector introduced by changing E_1 *can also be achieved by changing* A_1.

Example 5.2 *Now look at the same model as above but under a dynamic harmonic load.*

$$\left[-\omega^2 M(\boldsymbol{\lambda})_{ij} + i\omega B(\boldsymbol{\lambda})_{ij} + K(\boldsymbol{\lambda})_{ij}\right] x_j = F_i \tag{5.24}$$

$$x_j = X(\boldsymbol{\lambda})^{-1}{}_{ij} F_i \tag{5.25}$$

$$\boldsymbol{X}(\boldsymbol{\lambda}) = -\omega^2 \boldsymbol{M}(\boldsymbol{\lambda}) + i\omega \boldsymbol{B}(\boldsymbol{\lambda}) + \boldsymbol{K}(\boldsymbol{\lambda}) \tag{5.26}$$

Using Eq. (5.7) the dimension of the manifold is derived by:

$$\mathbf{rank}\left(\frac{\partial(x_1,\ldots,x_n)}{\partial\lambda_1}, \frac{\partial(x_1,\ldots,x_n)}{\partial\lambda_2}\right) = \tag{5.27}$$

$$= \mathbf{rank}(\boldsymbol{X}^{-1}(\partial_{\lambda_1}X)\boldsymbol{X}^{-1}\boldsymbol{F}, \boldsymbol{X}^{-1}(\partial_{\lambda_2}X)\boldsymbol{X}^{-1}\boldsymbol{F}) \tag{5.28}$$

$$= \mathbf{rank}(\boldsymbol{X}^{-1}(-\omega^2\frac{\partial}{\partial A_1}\boldsymbol{M} + \frac{\partial}{\partial A_1}\boldsymbol{K}))\boldsymbol{X}^{-1}\boldsymbol{F}, \boldsymbol{X}^{-1}(\frac{\partial}{\partial E_1}\boldsymbol{K}))\boldsymbol{X}^{-1}\boldsymbol{F}) \tag{5.29}$$

In general the two derivatives will be linear independent and therefore the solution manifold is 2-dimensional. This will result in a two dimensional response surface embedded in the eight dimensional space of the independent displacement degrees of freedom.

5.3 Local approximation of the 1-dimensional manifold

In this section it will be discussed how $\boldsymbol{x}(\lambda)$ can be computed, when $\boldsymbol{\Lambda} \subset \mathbb{R}^1$. The general d-dimensional case will be shown in the next section basing on the 1-dimensional case.

Let $\boldsymbol{x}(\lambda)$ be *analytic*, which means $\boldsymbol{x}(\lambda)$ can be expressed as an (infinite) power series

$$\boldsymbol{x}(\lambda) = \sum_{i=0}^{\infty} \boldsymbol{a}_i \lambda^i \tag{5.30}$$

or, more specifically as a Taylor series:

$$\boldsymbol{x}(\lambda) = \sum_{i=0}^{m} \frac{\lambda^i}{i!}\frac{\partial^i \boldsymbol{x}}{\partial\lambda^i}\bigg|_{\lambda=0} + R_m(\lambda) \tag{5.31}$$

$$\text{with} \quad R_m(\lambda) = \sum_{i=m+1}^{\infty} \frac{\lambda^i}{i!}\frac{\partial^i \boldsymbol{x}}{\partial\lambda^i}\bigg|_{\lambda=0} \tag{5.32}$$

with a converging R_m:

$$\lim_{m\to\infty} R_m(\lambda) = 0 \tag{5.33}$$

Introducing

$$\alpha_i = \frac{\lambda^i}{i!} \tag{5.34}$$

$$\partial^i \boldsymbol{x} = \frac{\partial^i \boldsymbol{x}}{\partial\lambda^i}\bigg|_{\lambda=0} \tag{5.35}$$

$$\boldsymbol{\partial}_i = \partial^i \boldsymbol{x} \tag{5.36}$$

Eq. (5.31) becomes

$$\boldsymbol{x}(\lambda) = \sum_{i=0}^{\infty} \alpha_i(\partial^i \boldsymbol{x}) \quad . \tag{5.37}$$

As $\boldsymbol{x} \in \mathbb{R}^n$ it follows that

$$\mathbf{rank}\left(\left[\partial^0 \boldsymbol{x}, \partial^1 \boldsymbol{x}, \ldots, \partial^\infty \boldsymbol{x}\right]\right) = m \leq n \quad . \tag{5.38}$$

Theorem 5.3 *Without restricting generality let d be such that $\{\partial^0 \boldsymbol{x}, \partial^1 \boldsymbol{x}, \ldots, \partial^m \boldsymbol{x}\}$ are linear independent and*

$$\partial^{m+1} \boldsymbol{x} = \sum_{i=0}^{m} \eta_i(\partial^i \boldsymbol{x}) \tag{5.39}$$

Then $\forall s > m$ the following holds:

$$\partial^s \boldsymbol{x} = \sum_{i=0}^{m} \theta_i(\partial^i \boldsymbol{x}) \tag{5.40}$$

Proof 5.2 *Proof is done by induction.*

State $s = m + 1$ was part of the assumption.
Lets look at $s \to s + 1$:

$$\begin{aligned}
\partial^{s+1} \boldsymbol{x} &= \partial(\partial^s \boldsymbol{x}) && (5.41)\\
&= \partial\left[\sum_{i=0}^{m} \eta_i(\partial^i \boldsymbol{x})\right] && (5.42)\\
&= \sum_{i=0}^{m} \eta_i(\partial^{i+1} \boldsymbol{x}) && (5.43)\\
&= \sum_{i=1}^{m} \eta_{i-1}(\partial^i \boldsymbol{x}) + \eta_m \partial^{m+1} \boldsymbol{x} && (5.44)\\
&= \sum_{i=1}^{m} \eta_{i-1}(\partial^i \boldsymbol{x}) + \eta_m \sum_{i=0}^{m} \eta_i(\partial^i \boldsymbol{x}) && (5.45)\\
&= \sum_{i=0}^{m} \theta_i(\partial^i \boldsymbol{x}) && (5.46)
\end{aligned}$$

$$\textit{using} \quad \theta_i = \eta_m \eta_i + \eta_{i-1},\ \eta_{-1} = 0$$

q.e.d.

As seen the first m derivatives, including the 0-th, are linear independent and therefore form, a basis of a $(m + 1)$-dimensional subspace $\mathcal{D}$ of $\mathbb{R}^n$:

$$\mathcal{D} = \mathbf{span}\left(\partial^0 \boldsymbol{x}, \partial^1 \boldsymbol{x}, \ldots, \partial^m \boldsymbol{x}\right) \tag{5.47}$$

Because all higher derivatives are linear dependent they also are embedded within this subspace $\mathcal{D}$. This subspace is therefore, due to assumption Eq. (5.40) and the analyticity of $\boldsymbol{x}(\lambda)$, as given in Eq. (5.30), the embedding space of the response manifold:

$$M^1 \subseteq \mathcal{D} \subseteq \mathcal{N} \tag{5.48}$$

An orthonormal basis for $\mathcal{D}$ in $\mathbb{R}^n$ can now be created using for example Gram-Schmidt *ortho-normalization*. Let

$$\overline{\boldsymbol{D}}^0 = \partial^0 \boldsymbol{x} \tag{5.49}$$

$$\boldsymbol{D}^0 = \frac{1}{\|\overline{\boldsymbol{D}}^0\|_2} \quad . \tag{5.50}$$

and for $i = 1, \ldots, m$:

$$\overline{\boldsymbol{D}}^i = \partial^i \boldsymbol{x} - \sum_{j=1}^{i-1} < \partial^i \boldsymbol{x}; \boldsymbol{D}^j > \boldsymbol{D}^j \tag{5.51}$$

$$\boldsymbol{D}^i = \frac{1}{\|\overline{\boldsymbol{D}}^i\|_2} \tag{5.52}$$

Where $< \boldsymbol{a}; \boldsymbol{b} >$ is the scalar product or inner tensor product of the two vectors.

Using the ortho-normalized $\boldsymbol{D}^j$ the $\partial^i \boldsymbol{x}$ can be expressed through them:

$$\partial^i \boldsymbol{x} = \sum_{j=0}^{\min(i,m+1)} \beta_{ij} \boldsymbol{D}^j \tag{5.53}$$

using

$$\beta_{ij} =< (\partial^i \boldsymbol{x}); (\boldsymbol{D}^j) > \tag{5.54}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_{00} & & & & & \\ \beta_{10} & \beta_{11} & & & & \\ \beta_{20} & \beta_{21} & \beta_{22} & & & \\ \beta_{30} & \beta_{31} & \beta_{32} & \beta_{33} & & \\ \cdots & & & & & \\ \beta_{(m+1)0} & \beta_{(m+1)1} & \beta_{(m+1)2} & \beta_{(m+1)3} & \cdots \; \beta_{(m+1)m} & \beta_{(m+1)(m+1)} \\ \beta_{(m+2)0} & \beta_{(m+2)1} & \beta_{(m+2)2} & \beta_{(m+2)3} & \cdots \; \beta_{(m+2)m} & \beta_{(m+2)(m+1)} \\ \cdots & & & & & \end{bmatrix} \in \mathbb{C}^{\infty \times (m+1)} \tag{5.55}$$

Instead of expressing $\boldsymbol{x}(\lambda)$ through an infinite series of $\partial^i \boldsymbol{x}$ as in Eq. (5.37) it can be expressed now with a finite series of $\boldsymbol{D}^i$:

$$\begin{aligned}
\boldsymbol{x}(\lambda) &= \sum_{i=0}^{\infty} \alpha_i (\partial^i \boldsymbol{x}) \\
&= \sum_{i=0}^{\infty} \alpha_i \left(\sum_{j=0}^{\min(i,m)} \beta_{ij} \boldsymbol{D}^j \right) \\
&= \alpha_0 \left(\beta_{00} \boldsymbol{D}^0 \right) + \\
&\quad + \alpha_1 \left(\beta_{10} \boldsymbol{D}^0 + \beta_{11} \boldsymbol{D}^1 \right) + \\
&\quad + \alpha_2 \left(\beta_{20} \boldsymbol{D}^0 + \beta_{21} \boldsymbol{D}^1 + \beta_{22} \boldsymbol{D}^2 \right) + \\
&\quad + \ldots \\
&= \boldsymbol{D}^0 \left(\alpha_0 \beta_{00} + \alpha_1 \beta_{10} + \alpha_2 \beta_{20} + \ldots \right) + \\
&\quad + \boldsymbol{D}^1 \left(\alpha_1 \beta_{11} + \alpha_2 \beta_{21} + \alpha_3 \beta_{31} + \ldots \right) + \\
&\quad + \boldsymbol{D}^2 \left(\alpha_2 \beta_{22} + \alpha_3 \beta_{32} + \alpha_4 \beta_{42} + \ldots \right) + \\
&\quad + \ldots \\
&\quad + \boldsymbol{D}^m \left(\alpha_m \beta_{mm} + \alpha_{m+1} \beta_{m+1,m} + \alpha_{m+2} \beta_{m+2,m} + \ldots \right) \\
&= \sum_{i=0}^{m} \gamma_i \boldsymbol{D}^i
\end{aligned} \tag{5.56}$$

with

$$\gamma_i = \sum_{j=i}^{\infty} \alpha_j \beta_{ji} \tag{5.57}$$

Basing on this series,

$$\boldsymbol{x}^{(j)}(\lambda) = \sum_{i=0}^{j} \gamma_i \boldsymbol{D}^i \tag{5.58}$$

can be defined.

Theorem 5.4 *The sequence* $\boldsymbol{x}^{(j)}(\lambda)$*,* $j = 0, 1, \ldots,$ *converges to* $\boldsymbol{x}(\lambda)$

Proof 5.3 *Show that for each* $\epsilon > 0 \quad \exists p \in \mathbb{N} \ni \quad \|\boldsymbol{x}(\lambda) - \boldsymbol{x}^{(j)}(\lambda)\| < \epsilon \quad \forall \quad j \geq p$ *(*$\| \, \|$ *denotes* *the 2-norm):*

$$\|\boldsymbol{x}(\lambda) - \boldsymbol{x}^{(j)}(\lambda)\| = \sqrt{< (\boldsymbol{x}(\lambda) - \boldsymbol{x}^{(j)}(\lambda)); (\boldsymbol{x}(\lambda) - \boldsymbol{x}^{(j)}(\lambda)) >} \tag{5.59}$$

$$= \sqrt{< (\sum_{i=0}^{m} \gamma_i \boldsymbol{D}^i - \sum_{i=0}^{j} \gamma_i \boldsymbol{D}^i); (\sum_{i=0}^{m} \gamma_i \boldsymbol{D}^i - \sum_{i=0}^{j} \gamma_i \boldsymbol{D}^i) >} \tag{5.60}$$

$$= \sqrt{< (\sum_{i=j+1}^{m} \gamma_i \boldsymbol{D}^i); (\sum_{i=j+1}^{m} \gamma_i \boldsymbol{D}^i) >} \tag{5.61}$$

$$= \sqrt{\sum_{i=j+1}^{m} \sum_{k=j+1}^{m} {\gamma_i}^* \gamma_k < \boldsymbol{D}^i; \boldsymbol{D}^k >} \quad {}^*\textit{denotes complex conjugated value} \tag{5.62}$$

$$= \sqrt{\sum_{i=j+1}^{m} \sum_{k=j+1}^{m} {\gamma_i}^* \gamma_k \delta^{ik}} \quad \textit{using ortho-normality of } \boldsymbol{D}^i \tag{5.63}$$

$$= \sqrt{\sum_{i=j+1}^{m} |\gamma_i|^2} \tag{5.64}$$

$$\|\boldsymbol{x}(\lambda) - \boldsymbol{x}^{(j)}(\lambda)\| < \epsilon \tag{5.65}$$

$$\Longleftrightarrow \sqrt{\sum_{i=j+1}^{m} |\gamma_i|^2} < \epsilon \tag{5.66}$$

$$\Longleftrightarrow \sum_{i=j+1}^{m} |\gamma_i|^2 < \epsilon^2 \tag{5.67}$$

Trivially this holds for $j = m$. For $j < m$, p is defined by ϵ and $\gamma_m, \gamma_{m-1}, \ldots$ respectively. q.e.d.

Now it will be examined what sort of condition is necessary for such a rapid convergence that any additional term improves the accuracy of the series less then the last one did:

$$\|\boldsymbol{x}^{(j)}(\lambda) - \boldsymbol{x}^{(j-1)}(\lambda)\| > \|\boldsymbol{x}^{(j+1)}(\lambda) - \boldsymbol{x}^{(j)}(\lambda)\| \tag{5.68}$$

$$\Longleftrightarrow \|\gamma_j \boldsymbol{D}^j\| > \|\gamma_{j+1} \boldsymbol{D}^{j+1}\| \tag{5.69}$$

$$\Longleftrightarrow |\gamma_j|^2 > |\gamma_{j+1}|^2 \tag{5.70}$$

$$\Longleftrightarrow \frac{|\gamma_{j+1}|}{|\gamma_j|} \leq \epsilon < 1 \quad \forall j \geq p \tag{5.71}$$

$$\tag{5.72}$$

Once this rapid convergence is fulfilled also *d'Alembert's* criteria for an absolute convergence would be fulfilled.

5.4 Determination of coefficients using the Rayleigh-Ritz procedure

As seen in Eq. (5.56) the solution for any point on the response manifold can be expressed by a finite series:

$$\boldsymbol{x}(\lambda) = \sum_{i=0}^{m} \gamma_i \boldsymbol{D}^i \tag{5.73}$$

The problem by calculating the finite coefficients γ_i of this series is that they are defined via an infinite sum Eq. (5.57):

$$\gamma_a = \sum_{j=i}^{\infty} \alpha_j \beta_{ja} \tag{5.74}$$

To determine these γ_i the baseline principles of the finite element method are used, described in detail in the earlier chapter 4.2. These two principles are the *Hamiltonian principle* and the *Rayleigh-Ritz* procedure for the determination of an approximate solution. The only difference is, that instead of choosing the *local shape functions* $\boldsymbol{N}^i$, derived from the earlier introduced *finite elements*, now *global shape functions* $\boldsymbol{D}^i$ are used. The $\boldsymbol{D}^i$ will be denoted as *derivative modes*, borrowing the term *mode* from the *normal modes* resulting from an eigenvalue calculation. Therefore instead of having displacements as degrees of freedom $\boldsymbol{q}$ now *manifold coordinates* or *modal coordinates* γ_i are the unknowns:

$$\overline{q}_i = \gamma_i \tag{5.75}$$

In order to distinguish them from the displacement coordinates all ARM related items will be $\overline{\text{overlined}}$. The approach Eq. (4.49) becomes:

$$\boldsymbol{u}(\boldsymbol{X}) = \sum_{i=1}^{m} \overline{q}_i \boldsymbol{D}^i(\boldsymbol{X}) \tag{5.76}$$

$$\boldsymbol{u} = \overline{q}_i \boldsymbol{D}^i \tag{5.77}$$

Once determined, the *global shape functions* can be expressed by the *local shape functions*:

$$\boldsymbol{D}^a(\boldsymbol{X}) = \sum_{i=1}^{n} \Phi_i{}^a \boldsymbol{N}^i(\boldsymbol{X}) \tag{5.78}$$

$$\boldsymbol{D}^a = \Phi_i{}^a \boldsymbol{N}^i \tag{5.79}$$

$$\tag{5.80}$$

Inserting Eq. (5.78) into Eq. (5.76) leads to:

$$\boldsymbol{u} = \sum_{j=1}^{m} \overline{q}_j \sum_{i=1}^{n} \phi_i{}^j \boldsymbol{N}^i \tag{5.81}$$

$$= \sum_{i=1}^{n} (\sum_{j=1}^{m} \overline{q}_j \phi_i{}^j) \boldsymbol{N}^i \tag{5.82}$$

Comparing Eq. (5.82) with Eq. (4.49) gives:

$$q_i = \sum_{j=1}^{m} \phi_i{}^j \overline{q}_j \tag{5.83}$$

$$\boldsymbol{q} = \boldsymbol{\phi}\overline{\boldsymbol{q}} \tag{5.84}$$

5.4.1 The equations of motion in transformed coordinates

5.4.1.1 The mass matrix in transformed coordinates

Similar to Eq. (4.58) the Eq. (3.14) and Eq. (4.49) lead Eq. (4.11) to:

$$T = \frac{1}{2}\int_V \rho[\dot{\overline{q}}_i D^i]_\alpha [\dot{\overline{q}}_j D^j]^\alpha \, dV \tag{5.85}$$

$$= \frac{1}{2}\dot{\overline{q}}_i \dot{\overline{q}}_j \int_V \rho[D^i]_\alpha [D^j]^\alpha \, dV \tag{5.86}$$

$$= \frac{1}{2}\dot{\overline{q}}_i \dot{\overline{q}}_j \int_V \rho[\phi_k{}^i N^k]_\alpha [\phi_l{}^j N^l]^\alpha \, dV \tag{5.87}$$

$$= \frac{1}{2}\dot{\overline{q}}_i \dot{\overline{q}}_j \phi_k{}^i \phi_l{}^j \int_V \rho[N^k]_\alpha [N^l]^\alpha \, dV \tag{5.88}$$

$$= \frac{1}{2}\dot{\overline{q}}_i \dot{\overline{q}}_j \phi_k{}^i \phi_l{}^j M^{kl} \tag{5.89}$$

$$= \frac{1}{2}\dot{\overline{q}}_i \dot{\overline{q}}_j \overline{M}^{ij} \tag{5.90}$$

$$\tag{5.91}$$

with the *modal mass matrix* $\overline{\boldsymbol{M}}$:

$$\overline{M}^{ij} = \phi_k{}^i \phi_l{}^j M^{kl} \tag{5.92}$$

$$\overline{\boldsymbol{M}} = \boldsymbol{\phi}^t \boldsymbol{M} \boldsymbol{\phi} \tag{5.93}$$

5.4.1.2 The stiffness matrix in transformed coordinates

Inserting Eq. (4.62) into Eq. (4.47) gives:

$$V = \frac{1}{2}\int_V B_{\overline{\alpha}}{}^{\beta}[\overline{q}_i D^i]_\beta \bar{C}^{\overline{\alpha\gamma}} B_{\overline{\gamma}}{}^{\delta}[\overline{q}_j D^j]_\delta \, dV \tag{5.94}$$

$$= \frac{1}{2}\overline{q}_i \overline{q}_j \int_V B_{\overline{\alpha}}{}^{\beta}[\phi_k{}^i N^k]_\beta \bar{C}^{\overline{\alpha\gamma}} B_{\overline{\gamma}}{}^{\delta}[\phi_l{}^j N^l]_\delta \, dV \tag{5.95}$$

$$= \frac{1}{2}\overline{q}_i \overline{q}_j \phi_k{}^i \phi_l{}^j \int_V B_{\overline{\alpha}}{}^{\beta}[N^k]_\beta \bar{C}^{\overline{\alpha\gamma}} B_{\overline{\gamma}}{}^{\delta}[N^l]_\delta \, dV \tag{5.96}$$

$$= \frac{1}{2}\overline{q}_i \overline{q}_j \phi_k{}^i \phi_l{}^j K^{kl} \tag{5.97}$$

$$= \frac{1}{2}\overline{q}_i \overline{q}_j \overline{K}^{ij} \tag{5.98}$$

with the *modal stiffness matrix* $\overline{\boldsymbol{K}}$:

$$\overline{K}^{ij} = {\phi_k}^i {\phi_l}^j K^{kl} \tag{5.99}$$

$$\overline{\boldsymbol{K}} = \boldsymbol{\phi}^t \boldsymbol{K} \boldsymbol{\phi} \tag{5.100}$$

5.4.1.3 Nonconservative terms in transformed coordinates

The damping matrix $\boldsymbol{B}$ in Eq. (4.79) is transformed the same way as the stiffness matrix:

$$\overline{C}^{ij} = {\phi_k}^i {\phi_l}^j B^{ij} \tag{5.101}$$

$$\overline{\boldsymbol{B}} = \boldsymbol{\phi}^t \boldsymbol{B} \boldsymbol{\phi} \tag{5.102}$$

The nonconservative external work is invariant and is given in *modal* coordinates by:

$$\delta W_{noncon,ext} = \overline{F}_i \delta \overline{q}^i \tag{5.103}$$

The governing equation in *displacement coordinates* was:

$$\boldsymbol{F} = [-\omega^2 \boldsymbol{M} + \mathrm{i}\omega \boldsymbol{B} + \boldsymbol{K}]\boldsymbol{q} \tag{5.104}$$

As the equation of motion must stay the same, it is in *modal coordinates*:

$$\overline{\boldsymbol{F}} = [-\omega^2 \overline{\boldsymbol{M}} + \mathrm{i}\omega \overline{\boldsymbol{B}} + \overline{\boldsymbol{K}}]\overline{\boldsymbol{q}} \tag{5.105}$$

Inserting Eq. (5.93), Eq. (5.102), Eq. (5.100) into Eq. (5.105) gives:

$$\overline{\boldsymbol{F}} = [-\omega^2 \boldsymbol{\phi}^t \boldsymbol{M} \boldsymbol{\phi} + \mathrm{i}\omega \boldsymbol{\phi}^t \boldsymbol{B} \boldsymbol{\phi} + \boldsymbol{\phi}^t \boldsymbol{K} \boldsymbol{\phi}]\overline{\boldsymbol{q}} \tag{5.106}$$

$$= \boldsymbol{\phi}^t [-\omega^2 \boldsymbol{M} + \mathrm{i}\omega \boldsymbol{B} + \boldsymbol{K}]\boldsymbol{\phi}\overline{\boldsymbol{q}} \tag{5.107}$$

with Eq. (5.84)

$$= \boldsymbol{\phi}^t [-\omega^2 \boldsymbol{M} + \mathrm{i}\omega \boldsymbol{B} + \boldsymbol{K}]\boldsymbol{q} \tag{5.108}$$

with Eq. (5.104)

$$= \boldsymbol{\phi}^t \boldsymbol{F} \tag{5.109}$$

5.5 Local approximation of the d-dimensional manifold

In general design problems, the amount of design variables is not restricted to one. The general d-dimensional design problem has a d-dimensional solution manifold. Therefore the power series

Eq. (5.30) must be represented with a multidimensional Taylor series (see also [27]):

$$\boldsymbol{x}(\boldsymbol{\lambda}) = \sum_{|\bar{\boldsymbol{\alpha}}| \le k} \frac{\boldsymbol{\lambda}^{\bar{\boldsymbol{\alpha}}}}{\bar{\boldsymbol{\alpha}}!} \partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}} \boldsymbol{x} \bigg|_{\lambda=0} + R_k(\lambda) \tag{5.110}$$

$$\text{with:} \quad \boldsymbol{\lambda} \in \mathbb{R}^d \tag{5.111}$$

$$\bar{\boldsymbol{\alpha}} = (\bar{\alpha}_1, \dots, \bar{\alpha}_d) \in \mathbb{N}^d \tag{5.112}$$

$$|\bar{\boldsymbol{\alpha}}| = \bar{\alpha}_1 + \dots + \bar{\alpha}_d \tag{5.113}$$

$$\bar{\boldsymbol{\alpha}}! = \bar{\alpha}_1! \, \bar{\alpha}_2! \cdot \ldots \cdot \bar{\alpha}_d \tag{5.114}$$

$$\boldsymbol{\lambda}^{\bar{\boldsymbol{\alpha}}} = \lambda_1^{\bar{\alpha}_1} \lambda_2^{\bar{\alpha}_2} \cdot \ldots \cdot \lambda_d^{\bar{\alpha}_d} \tag{5.115}$$

$$\partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}} \boldsymbol{x} = \frac{\partial^{|\bar{\boldsymbol{\alpha}}|} \boldsymbol{x}}{\partial^{\bar{\alpha}_1} \lambda_1 \partial^{\bar{\alpha}_2} \lambda_2 \cdot \ldots \cdot \partial^{\bar{\alpha}_d} \lambda_d} \tag{5.116}$$

$$R_k(\lambda) = \sum_{|\bar{\boldsymbol{\alpha}}| > k} \frac{\boldsymbol{\lambda}^{\bar{\boldsymbol{\alpha}}}}{\bar{\boldsymbol{\alpha}}!} \partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}} \boldsymbol{x} \bigg|_{\lambda=0} \tag{5.117}$$

In analogy to the one-dimensional case in chapter 5.3 the differential vectors are orthonormalized:

$$\partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}}|_{\boldsymbol{\alpha}=(0,0,\dots,0)} \to \boldsymbol{D}^0 \tag{5.118}$$

$$\partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}}|_{\boldsymbol{\alpha}=(1,0,\dots,0)} \to \boldsymbol{D}^1 \tag{5.119}$$

$$\partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}}|_{\boldsymbol{\alpha}=(0,1,\dots,0)} \to \boldsymbol{D}^2 \tag{5.120}$$

$$\dots \tag{5.121}$$

$$\partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}}|_{\boldsymbol{\alpha}=(0,0,\dots,d)} \to \boldsymbol{D}^{1+d} \tag{5.122}$$

$$\partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}}|_{\boldsymbol{\alpha}=(2,0,\dots,0)} \to \boldsymbol{D}^{1+d+1} \tag{5.123}$$

$$\partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}}|_{\boldsymbol{\alpha}=(1,1,\dots,0)} \to \boldsymbol{D}^{1+d+2} \tag{5.124}$$

$$\dots \tag{5.125}$$

$$\partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}}|_{\boldsymbol{\alpha}=(0,0,\dots,2)} \to \boldsymbol{D}^{d^0+d^1+d^2} \tag{5.126}$$

$$\dots \tag{5.127}$$

The amount of differential vectors, which will serve as a basis for the response manifold, grows with d^k. k denotes the k-th derivative. This means on one side a lot of computing effort to reach a certain step k on the other side it has a positive effect for enriching the constructing basis of the manifold and therefore improves accuracy. In many cases, as seen later on in the example 8.1, the derivative vectors $\boldsymbol{D}^i$ saturate the basis very fast and therefore the differentiation can be stopped after the first derivatives.

It has to be noted that the mixed terms are in general not commutative, due to the fact that the matrix product is not commutative:

$$\boldsymbol{AB} \neq \boldsymbol{BA} \,\text{with}\, \boldsymbol{A}, \boldsymbol{B} \text{ rectangular matrices} \tag{5.128}$$

$$\partial_{\boldsymbol{\lambda}}^{(\overline{\alpha}_1, \overline{\alpha}_2, \ldots)} \neq \partial_{\boldsymbol{\lambda}}^{(\overline{\alpha}_2, \overline{\alpha}_1, \ldots)} \tag{5.129}$$

Furthermore it should be kept in mind that for practical reasons, when more then one derivative per parameter is considered, the derivatives should not be calculated mulish in each direction with the same effort. Sensitive variables should get more attention then others. However, this has to be decided during the process of building up the response manifold and is very problem specific.

As proven in the section about the 1-dimensional approximation the evaluation of the γ_i coefficients for an actual $\boldsymbol{\lambda}$ is no problem. Once the $\boldsymbol{D}^i$ are determined, the direct use of the *Hamiltonian principle* and the *Rayleigh-Ritz procedure* produces the missing coefficients.

5.6 Global and mixed local global approximation using interpolation

On page 36 it was shown that for any $\epsilon > 0$ at least p basis vectors $\boldsymbol{D}^i$ were needed for approximating the manifold with an error smaller then a desired ϵ. The number of necessary basis vectors p depends significantly on $\boldsymbol{\lambda}$. The smaller $\boldsymbol{\lambda}$ or the larger the accepted error ϵ the less basis vectors are needed.

Up to this point the presented approach was basically a local one, giving better results the closer $\boldsymbol{\lambda}$ is to $\boldsymbol{\lambda}_0 = 0$. Even if it is demonstrated later in example 8.2 that reasonably accurate results for engineering type problems and parameter ranges can be obtained when applying this *local* approach there may be a desire to improve accuracy for certain problems.

For example an exact answer from the *response manifold* at $\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 (= 0)$ *and* $\boldsymbol{\lambda} = \boldsymbol{\lambda}_1$ is desired. Where $\boldsymbol{\lambda}_1$ can be arbitrarily *far away* from $\boldsymbol{\lambda}_0$. For $\boldsymbol{\lambda}_0$ additionally an exact solution for the first k derivatives is desired and therefore a good *local* approximation. This will be achieved creating the required k derivative modes at $\boldsymbol{\lambda} = \boldsymbol{\lambda}_0$ and adding to the basis $[\boldsymbol{D}^0, \ldots, \boldsymbol{D}^k]$ an additional residual vector $\boldsymbol{D}^{k+1}$ resulting from the *exact* result at point $\boldsymbol{\lambda} = \boldsymbol{\lambda}_1$. As illustrated in 4.2 the *Hamiltonian principle* and the *Rayleigh-Ritz procedure* will ensure that our resulting manifold will have the desired accuracy locally and at $\boldsymbol{\lambda} = \boldsymbol{\lambda}_1$. An excellent example using and comparing local and global behavior can be seen in Fig. 8.6 through Fig. 8.9.

Basically with this approach a *response manifold* can be built up being able to approximate any point $\boldsymbol{\lambda} \in \Lambda$ with any desired accuracy. It is a question of which and how many *modes* are invested in the construction of the basis vectors of the manifold. The practical limitation will simply be the amount of computing time necessary to create the reduced subspace and to evaluate the *approximate response manifold* (ARM).

According to the accuracy discussed in this chapter, the question of efficiency will be discussed in chapter 7.

6 Practical implementation and implications for structural problems

While the previous chapter introduced the concept of the response manifolds and proved the convergence for general nonlinear equations, the focus in this chapter is on the consequences for structural problems. By applying the assumption of analyticity imposed at Eq. (5.30) to typical structural variables the process can be simplified. At the end the procedure is summarized in a ready to code recipe.

6.1 Typical design variables in structural optimization

The focus of this thesis is on structural design and parameters common in optimization and reliability analyses. These primarily contain *material properties* and *elemental properties* (see MEYWERK [89], PAAS [104]). In this chapter examples will be given for:

- material properties: *Young's modulus, mass density*
- shell element properties: *thickness*
- beam element properties: *area, moment of inertia*
- mass element properties: *mass*
- spring/damper element properties: *stiffness, damping*

Independent from the choice of elements and discretization the grid point coordinates can be referenced by design variables as well. However this sort of design, *shape optimization* (see also [40], [112]), will not be addressed in this thesis.

In general a structural model consists of various parts. Recovering the definition of the system matrices $\boldsymbol{K}$ in Eq. (4.67), $\boldsymbol{M}$ in Eq. (4.59), and $\boldsymbol{B}$ in Eq. (4.78) and assuming that $\boldsymbol{X}$ represents either one of them or any combination, it is seen that the volume integral covers the whole body B defined in 3:

$$\boldsymbol{X} = \int_V \ldots dV \tag{6.1}$$

As the body got discretized in Eq. (4.14) the volume integral in Eq. (6.1) can be replaced through the sum over all the volumes of the E individual elements

$$\boldsymbol{X} = \sum_{e=1}^{E} \int_{V_e} \dots dV \tag{6.2}$$

or by clustering all the elements belonging to the same part and sharing the same properties

$$\begin{aligned} \boldsymbol{X} &= \sum_{p=1}^{P} \int_{V_p} \dots dV && (6.3) \\ &= \sum_{p=1}^{P} \sum_{e_p=1}^{E_p} \int_{V_{e_p}} \dots dV && (6.4) \\ &= \sum_{p=1}^{P} \boldsymbol{X}_i && (6.5) \end{aligned}$$

where P is the amount of different parts in the structural model. In most cases grid points and therefore their corresponding (displacement) degrees of freedom are connected to one part only. Nevertheless some degrees of freedom must be connected to at least two parts to prevent the structure from falling apart. This results in a part related pattern for the $\boldsymbol{X}$-matrix:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_{11}^{i} & \boldsymbol{X}_{12}^{i} & 0 \\ \boldsymbol{X}_{21}^{i} & \boldsymbol{X}_{22}^{i} + \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ 0 & \boldsymbol{R}_{21} & \boldsymbol{R}_{22} \end{pmatrix} \tag{6.6}$$

Here $\boldsymbol{X}^i$ denotes the sub matrix of $\boldsymbol{X}$ belonging to part i and $\boldsymbol{R}$ represents the rest of the structure defined by the other parts.

6.1.1 Derivatives of system matrices for shell elements

The contribution of shell elements to the stiffness matrix is based primarily on two different kind of stiffnesses according to *Kirchhoff* shell theory (see BATHE [10]):

- *membrane stiffness*, which is linear to the thickness
- *bending stiffness*, which has a cubic relationship to the shell thickness

If *transverse shear* cannot be neglected *Mindlin* theory (see MINDLIN [90] and TIMOSHENKO [119]) has to be applied. In this case the stiffness matrix of part i becomes:

$$\boldsymbol{K}_i = E_i(t_i \boldsymbol{K}^m{}_i + t_i{}^3 \boldsymbol{K}^b{}_i + \boldsymbol{K}^s{}_i(t_i)) \tag{6.7}$$

with indices m for membrane, b for bending and s for transverse shear (6.8)

t_i: thickness of part i (6.9)

E_i: Young's modulus of part i (6.10)

For the mass matrix there is a linear relationship concerning the shell thickness:

$$\boldsymbol{M}_i = \rho_i(t_i\boldsymbol{M}^a{}_i) \tag{6.11}$$

$$\text{with index } a \text{ for defining an area related matrix} \tag{6.12}$$

$$\rho_i\text{: mass density of part } i\text{'s material} \tag{6.13}$$

$$\tag{6.14}$$

The derivatives are:

$$\frac{\partial}{\partial E_i}\boldsymbol{K} = \frac{1}{E_i}\boldsymbol{K}_i \tag{6.15}$$

$$\frac{\partial^d}{\partial {E_i}^d}\boldsymbol{K} = 0 \quad \forall\, d \geq 2 \tag{6.16}$$

$$\frac{\partial}{\partial \rho_i}\boldsymbol{M} = \frac{1}{\rho_i}\boldsymbol{M}_i \tag{6.17}$$

$$\frac{\partial^d}{\partial {\rho_i}^d}\boldsymbol{M} = 0 \quad \forall\, d \geq 2 \tag{6.18}$$

$$\frac{\partial}{\partial t_i}\boldsymbol{K} = E_i(\boldsymbol{K}^m{}_i + 3t_i^2\boldsymbol{K}^b{}_i + \frac{\partial 1}{\partial t_i}\boldsymbol{K}^s{}_i(t)) \tag{6.19}$$

$$\frac{\partial^2}{\partial {t_i}^2}\boldsymbol{K} = E_i(6t_i\boldsymbol{K}^b{}_i + \frac{\partial^2 1}{\partial {t_i}^2}\boldsymbol{K}^s{}_i(t)) \tag{6.20}$$

$$\frac{\partial^3}{\partial {t_i}^3}\boldsymbol{K} = E_i(6\boldsymbol{K}^b{}_i + \frac{\partial^3 1}{\partial {t_i}^3}\boldsymbol{K}^s{}_i(t)) \tag{6.21}$$

$$\frac{\partial^d}{\partial {t_i}^d}\boldsymbol{K} = E_i(\frac{\partial^3 1}{\partial {t_i}^3}\boldsymbol{K}^s{}_i(t)) \quad \forall\, d \geq 4 \tag{6.22}$$

$$\frac{\partial}{\partial t_i}\boldsymbol{M} = \frac{1}{t_i}\boldsymbol{M}_i \tag{6.23}$$

$$\frac{\partial^d}{\partial {t_i}^d}\boldsymbol{M} = 0 \quad \forall\, d \geq 2 \tag{6.24}$$

$$\tag{6.25}$$

If transverse shear can be neglected and only up to second order derivatives shall be considered the derivative matrices $\frac{\partial}{\partial t_i}\boldsymbol{K}$ and $\frac{\partial^2}{\partial {t_i}^2}\boldsymbol{K}$ are linear combinations of $\boldsymbol{K}^m{}_i$ and $\boldsymbol{K}^b{}_i$. Therefore it is sufficient to consider only the later matrices when generating the derivative modes instead of explicitly calculating the $\frac{\partial^1}{\partial {t_i}^1}\boldsymbol{K}$ and $\frac{\partial^2}{\partial {t_i}^2}\boldsymbol{K}$, which requires a numerically more intensive process.

6.1.2 Derivatives of system matrices for beam elements

For beam elements the parts contributing to the stiffness are:

- *cross section area* A for axial forces
- *moment of inertia about axis 1* I_1 for bending

- *moment of inertia about axis 2* I_2 for bending
- *mixed moment of inertia* I_{12} for bending
- *torsional constant* T for torsion

For *truss* (or *rod*) elements like the ones used in the example shown in Fig. 2.1 no bending or torsional stiffness exist. If the beam elements are defined via geometrical beam cross sections the values A, I_1, I_2, I_{12}, T have to be calculated based on the geometric cross section values before the system matrices are built up.

The stiffness matrix of a part i consisting of beams (Bernoulli-Euler theory, see Bathe [10]) is given by:

$$\boldsymbol{K}_i = E_i(A_i\boldsymbol{K}^L{}_i + I_1\boldsymbol{K}^{I_1}{}_i + I_2\boldsymbol{K}^{I_2}{}_i + I_{12}\boldsymbol{K}^{I_{12}}{}_i + T\boldsymbol{K}^T{}_i) \tag{6.26}$$

$$E_i\text{: Young's modulus of part } i \tag{6.27}$$

The mass matrix for a beam part is given by:

$$\boldsymbol{M}_i = \rho_i(A_i\boldsymbol{M}^L{}_i) \tag{6.28}$$

$$\text{with index } L \text{ for defining the length of all concerned beams} \tag{6.29}$$

$$\rho_i\text{: mass density of part } i\text{'s material} \tag{6.30}$$

$$\tag{6.31}$$

This results in the following derivatives:

$$\frac{\partial}{\partial E_i}\boldsymbol{K} = \frac{1}{E_i}\boldsymbol{K}_i \tag{6.32}$$

$$\frac{\partial^d}{\partial {E_i}^d}\boldsymbol{K} = 0 \quad \forall\, d \geq 2 \tag{6.33}$$

$$\frac{\partial}{\partial X_i}\boldsymbol{K} = E_i\boldsymbol{K}^X{}_i \quad \text{with } X \in [A, I_1, I_2, I_{12}, T] \tag{6.34}$$

$$\frac{\partial^d}{\partial {X_i}^d}\boldsymbol{K} = 0 \quad \forall\, d \geq 2 \quad \text{and } X \in [A, I_1, I_2, I_{12}, T] \tag{6.35}$$

$$\frac{\partial}{\partial \rho_i}\boldsymbol{M} = A_i\boldsymbol{M}^L{}_i \tag{6.36}$$

$$\frac{\partial^d}{\partial {\rho_i}^d}\boldsymbol{M} = 0 \quad \forall\, d \geq 2 \tag{6.37}$$

$$\tag{6.38}$$

6.1.3 Derivatives of system matrices for mass elements

Mass elements contribute only to the mass matrix. Focusing on simple elements with no moments of inertia the mass matrix contribution of a mass element i is:

$$\boldsymbol{M}_i = m_i \boldsymbol{\delta}_{kj} \tag{6.39}$$

$$k\text{: degrees of freedom to which the mass is attached} \tag{6.40}$$

$$m\text{: mass of element } i \tag{6.41}$$

$$\tag{6.42}$$

This means for the resulting derivatives:

$$\frac{\partial}{\partial m_i}\boldsymbol{M} = \delta_{kj} \tag{6.43}$$

$$\frac{\partial^d}{\partial {m_i}^d}\boldsymbol{M} = 0 \quad \forall\, d \geq 2 \tag{6.44}$$

$$\tag{6.45}$$

An important fact especially in a later generation of the derivative modes is that a mass element is connected to only a few (in general 3 for $x-$, $y-$, $z-$) displacement degrees of freedom. Therefore the rank of the *derivative matrix* $\frac{\partial 1}{\partial m_1}\boldsymbol{M}$ is mostly 3.

6.1.4 Derivatives of system matrices for spring/damper elements

Spring/damper elements contribute only to the damping and stiffness matrix. For structural design problems these elements mostly connect only two degrees of freedom due to the fact that bushings are mostly non isotropic and have therefore different values for stiffness/damping in each direction. The damping/stiffness matrix contribution of a spring/damper element i is:

$$\boldsymbol{X}_i = x_i {\boldsymbol{X}^B}_i \tag{6.46}$$

$$x_i\text{: stiffness or damping value} \tag{6.47}$$

$${\boldsymbol{X}^B}_i\text{: (mostly rank 2) symmetric matrix} \tag{6.48}$$

$$\tag{6.49}$$

This means for the resulting derivatives:

$$\frac{\partial}{\partial x_i}\boldsymbol{X}_i = {\boldsymbol{X}^B}_i \tag{6.50}$$

$$\frac{\partial^d}{\partial {x_i}^d}\boldsymbol{X}_i = 0 \quad \forall\, d \geq 2 \tag{6.51}$$

$$\tag{6.52}$$

In analogy to the mass elements the stiffness/damping matrix of bushing elements has low rank.

6.2 Practical calculation of derivative modes

The derivative modes for structural models and the above mentioned design variables imply certain simplifications. A general derivative mode is given by Eq. (5.116):

$$\partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}} \boldsymbol{x} \tag{6.53}$$

Basically that means that the displacement vector $\boldsymbol{x}$ has to be differentiated by the design variables mentioned above. Using the equations of motion in matrix notation Eq. (4.89) and setting the general coordinates to the displacement vector $\boldsymbol{x}$ the general derivative mode is given by:

$$\partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}} \boldsymbol{x} = \partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}} [\boldsymbol{X}^{-1} \boldsymbol{F}] \tag{6.54}$$

$$\text{with} \quad \boldsymbol{X} = [-\omega^2 \boldsymbol{M} + i\omega \boldsymbol{B} + \boldsymbol{K}] \tag{6.55}$$

As the force does not depend on the design variables

$$\partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}} \boldsymbol{x} = [\partial_{\boldsymbol{\lambda}}^{\bar{\boldsymbol{\alpha}}} \boldsymbol{X}^{-1}] \boldsymbol{F} \tag{6.56}$$

is obtained. In order to calculate and discuss the $m-$th derivative of this matrix one has to start with the first order. The derivative for the $i-$th design variable is, using the derivative of the inverse given in Eq. (5.7).

$$\frac{\partial}{\partial \lambda_i} \boldsymbol{x} = \left[-\boldsymbol{X}^{-1} \left(\frac{\partial}{\partial \lambda_i} \boldsymbol{X} \right) \boldsymbol{X}^{-1} \right] \boldsymbol{F} \tag{6.57}$$

$$= -\boldsymbol{X}^{-1} \left(\frac{\partial}{\partial \lambda_i} \boldsymbol{X} \right) (\boldsymbol{X}^{-1} \boldsymbol{F}) \tag{6.58}$$

$$= -\boldsymbol{X}^{-1} \left[\frac{\partial}{\partial \lambda_i} \boldsymbol{X} \right] \boldsymbol{x} \tag{6.59}$$

The general second derivative of an inverse matrix is given by:

$$\frac{\partial^2}{\partial \lambda_i {\lambda_j}^2} \boldsymbol{X}^{-1} = \frac{\partial}{\partial \lambda_i} \left[\frac{\partial}{\partial \lambda_j} \boldsymbol{X}^{-1} \right] \tag{6.60}$$

$$= \frac{\partial}{\partial \lambda_i} \left[-\boldsymbol{X}^{-1} \left(\frac{\partial}{\partial \lambda_j} \boldsymbol{X} \right) \boldsymbol{X}^{-1} \right] \tag{6.61}$$

$$\begin{aligned} &= -\boldsymbol{X}^{-1} \left[\frac{\partial^2}{\partial \lambda_i {\lambda_j}^2} \boldsymbol{X} \right] \boldsymbol{X}^{-1} \\ &+ \boldsymbol{X}^{-1} \left(\frac{\partial}{\partial \lambda_j} \boldsymbol{X} \right) \boldsymbol{X}^{-1} \left(\frac{\partial}{\partial \lambda_i} \boldsymbol{X} \right) \boldsymbol{X}^{-1} \\ &+ \boldsymbol{X}^{-1} \left(\frac{\partial}{\partial \lambda_i} \boldsymbol{X} \right) \boldsymbol{X}^{-1} \left(\frac{\partial}{\partial \lambda_j} \boldsymbol{X} \right) \boldsymbol{X}^{-1} \end{aligned} \tag{6.62}$$

$$\tag{6.63}$$

An expansion to higher derivatives is straightforward. More important here is the relevance for structural problems. As seen in sections 6.1.1, 6.1.2, 6.1.3, 6.1.4 the second and higher derivatives of the system matrices vanish, if it is assumed that bending and membrane stiffness are separate but dependent design variables. Therefore what is left from the higher derivatives of the inverse are the mixed terms:

$$\frac{\partial^2}{\partial \lambda_i {\lambda_j}^2} \boldsymbol{X}^{-1} = \left[G_j G_i + G_i G_j \right] \boldsymbol{X}^{-1} \quad \text{using } G_i = \boldsymbol{X}^{-1} \left(\frac{\partial}{\partial \lambda_i} \boldsymbol{X} \right) \tag{6.64}$$

$$\rightarrow \frac{\partial^2}{\partial \lambda_i {\lambda_j}^2} \boldsymbol{x} = \left[G_j G_i + G_i G_j \right] \boldsymbol{x} \tag{6.65}$$

$$\tag{6.66}$$

The m-th derivative of the displacement vector $\boldsymbol{x}$ can be expressed, if assumed that the 2nd and higher derivatives of the original matrix $\boldsymbol{X}$ vanish, through:

$$\partial_{\boldsymbol{\lambda}}^{\overline{\boldsymbol{\alpha}}} \boldsymbol{x} = \left[\sum_{\text{permutations}} \underbrace{G_1 G_1 \dots G_1}_{\overline{\alpha}_1 times} \underbrace{G_2 G_2 \dots G_2}_{\overline{\alpha}_2 times} \dots \underbrace{G_d G_d \dots G_d}_{\overline{\alpha}_d times} \right] \boldsymbol{x} \tag{6.67}$$

Using the notation from Eq. (5.112) to Eq. (5.116).

6.3 Krylov sequence and Krylov subspaces

6.3.1 Dynamic steady state: frequency response

Assume there is only one parameter ω which represents the running frequency in a harmonic dynamic load case, see Eq. (6.55). Having a solution at $\omega = \omega_0$ for $\boldsymbol{X}$ solutions can be obtained at nearby ω's with the current method. If damping is neglected, the derivative modes

$$\frac{\partial^0}{\partial \omega^0}, \frac{\partial^1}{\partial \omega^1}, \frac{\partial^2}{\partial \omega^2}, \frac{\partial^3}{\partial \omega^3}, \dots \tag{6.68}$$

$$\tag{6.69}$$

become

$$\boldsymbol{x}, \boldsymbol{G}\boldsymbol{x}, \boldsymbol{G}^2\boldsymbol{x}, \boldsymbol{G}^3\boldsymbol{x}, \dots \tag{6.70}$$

with

$$\boldsymbol{G} = \boldsymbol{X}^{-1} \frac{\partial}{\partial \omega} \boldsymbol{X} \tag{6.71}$$

$$= \boldsymbol{X}^{-1} \boldsymbol{M} \tag{6.72}$$

Equation (6.70) defines actually a Krylov sequence (see also Parlett [107]), the base for the Lanczos procedure (see also Lanczos [78], Paige [105],[106] and Komzsik [77]), one of todays most efficient methods of calculating normal modes for large sparse systems. A comparison between a solution with the above described Krylov vectors and a classical modal solution with *normal modes* concerning accuracy and performance can be found in Arnold, Citerley, Chargin and Galant [6].

Also with multiple variables and higher derivatives the derivative modes are similar to Krylov vectors and therefore the embedding space of the solution manifold can be seen as a sort of a Krylov subspace. Nair shows in [96] the equivalence between the Krylov subspaces and Kirsch's [62] *Combined Approximations*. The relation between the *Combined Approximations* and the ARM approach is given in chapter 6.6.2.

6.3.2 Exact results and the rank of the Krylov subspace

As discussed earlier (see Eq. (5.1)) the generated derivative modes are not necessarily linear independent. Once one dependent vector appears, all the succeeding ones are also dependent (Eq. (5.39)). As we know now that the sequence of the derivative modes is a Krylov sequence (see Eq. (6.70)), at least if we have a one dimensional problem, we can assign the knowhow of Krylov subspaces to our derivative modes. Using the definition from Eq. (6.71) we know that $\boldsymbol{G}$ has at maximum the rank of the derivative matrix $\frac{\partial}{\partial\omega}\boldsymbol{X}$. A Krylov sequence can only generate as many independent vectors as it's generating matrix has distinct non zero eigenvalues (see also appendix C). This means that for parameters *wetting* only 2 degrees of freedom such as springs or dampers a maximum of 2 derivative modes is necessary to obtain exact results using the proposed approach. In this cases other *exact* approaches, such as *static condensation* (see Guyan [34]) and the use of substructures (see also Craig [13], Craig and Bampton [14], Craig and Chang [15] and Hurty [49]) can be considered inferior ways concerning efficiency once it comes to design problems. .

6.4 Derivative modes of the ten-bar truss problem

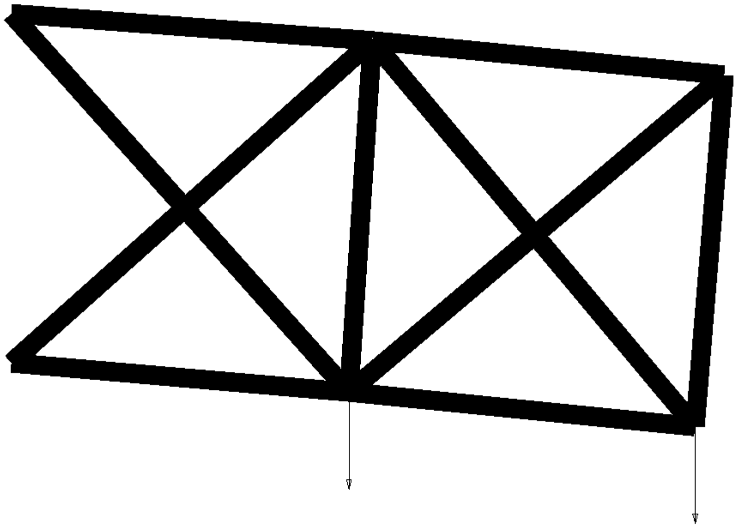


Figure 6.1: Original deflection $\boldsymbol{x}$ of the ten-bar truss

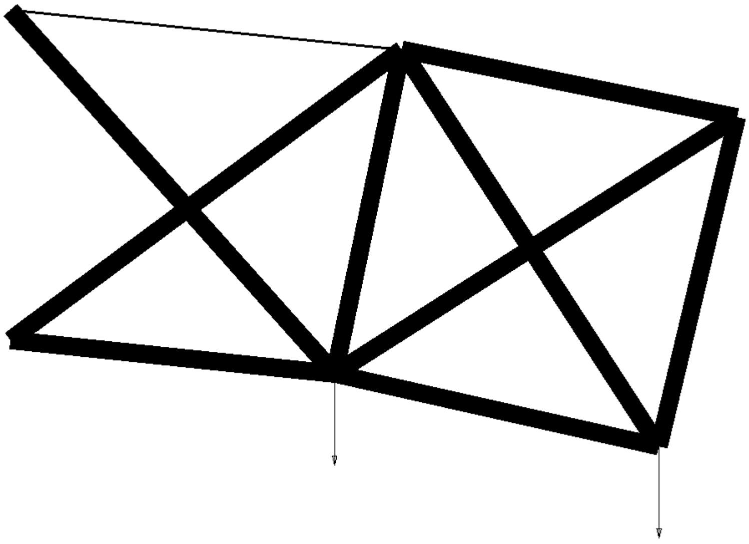


Figure 6.2: Derivative mode $\frac{\partial}{\partial \lambda_1}\boldsymbol{x}$ of the ten-bar truss

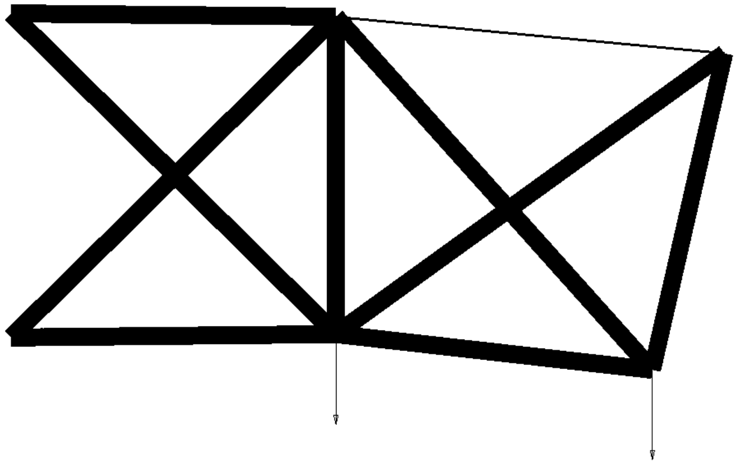


Figure 6.3: Derivative mode $\frac{\partial}{\partial \lambda_2}\boldsymbol{x}$ of the ten-bar truss

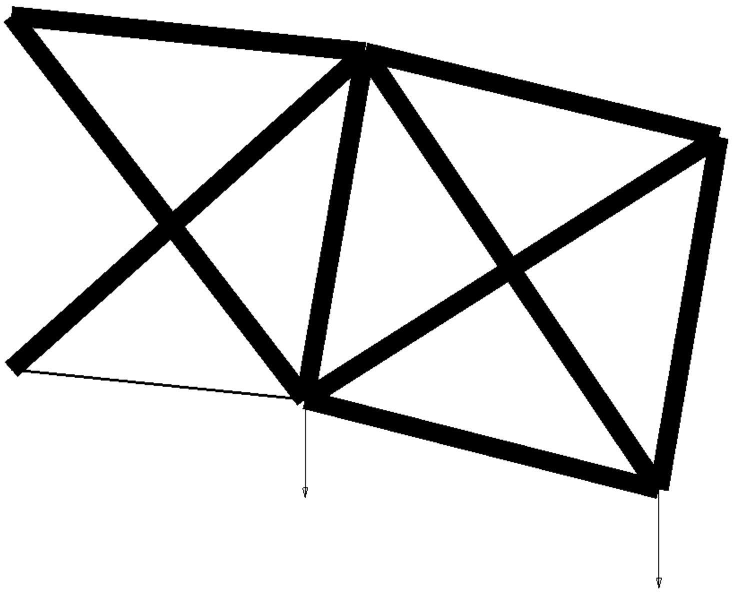


Figure 6.4: Derivative mode $\frac{\partial}{\partial\lambda_3}\boldsymbol{x}$ of the ten-bar truss

Applying the abstract context worked out in this chapter to the ten-bar truss problem in Fig. 2.1 it can actually be seen that the derivative modes are *similar* to the original deflection. This has the extremely positive effect that the space $\mathcal{D}$ saturates very fast and only few derivative modes are necessary for a rather accurate answer.

6.5 Data flow recipe

The necessary steps to approximate the *response manifold* for structural problems ($\boldsymbol{X} = -\omega^2\boldsymbol{M} + i\omega B + K$) are summed up as follows:

1. calculate the baseline analysis

$$\boldsymbol{X}\boldsymbol{x} = \boldsymbol{F} \longleftrightarrow \boldsymbol{x} = \boldsymbol{X}^{-1}\boldsymbol{F} \tag{6.73}$$

2. calculate the necessary derivatives

$$\frac{\partial}{\partial\lambda_i}\boldsymbol{X},\, i = 1, \ldots, d \tag{6.74}$$

3. calculate the factors for the Krylov sequence

$$\boldsymbol{G}_i = \boldsymbol{X}^{-1}[\frac{\partial}{\partial\lambda_i}\boldsymbol{X}] \tag{6.75}$$

4. calculate as many derivative modes $\partial_{\boldsymbol{\lambda}}^{\overline{\boldsymbol{\alpha}}}\boldsymbol{x}$ as desired, starting with first order derivatives. Stop criteria is a tradeoff consideration between accuracy and computational effort

$$\frac{\partial}{\partial\lambda_i}\boldsymbol{x} = \boldsymbol{G}\boldsymbol{x} \tag{6.76}$$

5. assemble all derivative modes into matrix $\mathbf{\Phi}$ and ortho-normalize it

$$\mathbf{\Phi} = [\frac{\partial}{\partial \lambda_1}\boldsymbol{x}, \frac{\partial}{\partial \lambda_2}\boldsymbol{x}, \ldots] \tag{6.77}$$

6. if the accuracy is not sufficient, add residual modes to $\mathbf{\Phi}$

$$\mathbf{\Phi} = [\mathbf{\Phi}, \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots] \tag{6.78}$$

7. transform all system matrices from the displacement degree of freedom coordinates to the manifold coordinates

$$\overline{\boldsymbol{X}} = \mathbf{\Phi}^t \boldsymbol{X} \mathbf{\Phi} \tag{6.79}$$

$$\overline{\boldsymbol{X}_i} = \mathbf{\Phi}^t \boldsymbol{X}_i \mathbf{\Phi} \tag{6.80}$$

$$\overline{\boldsymbol{F}} = \mathbf{\Phi}^t \boldsymbol{F} \tag{6.81}$$

$$\ldots \tag{6.82}$$

To evaluate a new design (new parameter setting) using the fast *approximate response manifold*, the following steps are necessary:

1. assemble the (manifold oriented) system matrices according to the desired parameter setting

$$\overline{\boldsymbol{X}}_{new} = \overline{\boldsymbol{X}} + \sum f(\lambda_i)\overline{\boldsymbol{X}_i} \tag{6.83}$$

2. solve the physical equation in manifold coordinates

$$\overline{\boldsymbol{x}}_{new} = \overline{\boldsymbol{X}}_{new}^{-1} \overline{\boldsymbol{F}} \tag{6.84}$$

3. transform the manifold coordinates back to displacement degrees of freedom using the transformation matrix $\mathbf{\Phi}$

$$\boldsymbol{x}_{new} = \mathbf{\Phi} \overline{\boldsymbol{x}}_{new} \tag{6.85}$$

4. if desired calculate strains and stresses based on the displacements

6.6 Related approaches and similar procedures in literature

6.6.1 Global approximation in design space

The early work of FOX and MIURA [29] can be seen as a special case of the current approach, if the modes forming $\mathbf{\Phi}$ are restricted to the mentioned *global* method defined in section 5.6. In their work 5 distinct points in the global design space were taken to form a basis with 5 residual vectors for approximating a design space with 4 parameters.

6.6.2 Combined approximation in the context

The *Combined Approximation* (CA) named procedure of Kirsch presented in numerous articles is bascially summarized in [56] and [63]. His work represents a major motivation for this thesis. The well converging method Kirsch presented in the above papers can basically be seen as a restriction of the present ARM method to a single variable with a first order parameter dependency as it appears using truss elements. Kirsch further examined other categories of optimization such as shape optimization in [69]. There he showed that the underlying procedure also had a reasonable convergence for changes concerning nodal coordinates. Topology optimization using the *Combined Approximation* was also adressed by Kirsch and Papalambros in their article [70]. Kirsch and Papalambros made suggestions to handle the decreasing amount of displacement degrees of freedom due to singularities in the stiffness matrix, which is a major problem for the reanalysis of topological modified structures due to singularity issues.

6.6.3 The pure natural modes approach

The *Vibro Acoustic Optimization* (VAO) approach presented by Mass [87], Witta [121] and Freymann [30] uses the *natural modes* generated by an eigenvalue analysis as the encapsulating space $\mathbf{\Phi}$ of the response manifold. This has the benefit that the infrastructure for computing the $\mathbf{\Phi}$ is already available. For modal frequency response analyses with moderate changes in parameters Mass [87] has shown that there is a reasonable prediction possible using the mentioned modal correction method. Nevertheless there are several factors that are disadvantageous compared to the sensitivity and Ritz-vector based encapsulating space $\mathbf{\Phi}$ of the ARM method. These are:

1. The $\mathbf{\Phi}$ matrix of VAO contains all natural modes up to a certain frequency regardless if and how much the corresponding natural mode contributes under the current loading condition. This is especially critical due to the fact that the number of column vectors in the reduced basis $\mathbf{\Phi}$ is the overall determining factor concerning computing time.

2. With *natural modes* alone there is no guarantee at all that more modes and therefore a larger $\mathbf{\Phi}$ and more computation effort leads to a higher accuracy.

3. Large shell models can have extremely high modal densities. A few thousand *natural modes* are not uncommon, see also Heiserer and Chargin [42]. This can totally destroy the computational advantage of a coordinate transformation, as seen later on in Fig. 7.13.

Therefore the VAO approach seems primarily efficient for areas with low modal densities and small parameter changes.

6.7 Mechanical interpretation

For a better understanding we restrict the general problem now to a static problem ($\omega = 0$).

6.7.1 Role of the strain energy

If we would not calculate any derivative or residual modes for the problem reduction we would be left with the current displacement vector $\boldsymbol{x}$:

$$\boldsymbol{\Phi} = \boldsymbol{x} \tag{6.86}$$

$$\tag{6.87}$$

The modal stiffness then basically represents the strain energy of the system:

$$\begin{aligned} \overline{\boldsymbol{K}} &= \boldsymbol{\Phi}^t \boldsymbol{K} \boldsymbol{\Phi} && (6.88) \\ &= \boldsymbol{x}^t \boldsymbol{K} \boldsymbol{x} && (6.89) \end{aligned}$$

Restricting the stiffness matrix to one design variable we get the strain energy of this variable contributing to the whole system:

$$\overline{\boldsymbol{K}_i} = \boldsymbol{x}^t \boldsymbol{K}_i \boldsymbol{x} \tag{6.90}$$

A new analysis with design variable i increased by α would lead to the following equation in reduced coordinates:

$$\begin{aligned} \boldsymbol{\xi} &= [\overline{\boldsymbol{K}} + (\alpha - 1)\overline{\boldsymbol{K}_i}]^{-1}[\boldsymbol{\Phi}^t \boldsymbol{F}] && (6.91) \\ \boldsymbol{x}_{\text{new}} &= \boldsymbol{\Phi}\boldsymbol{\xi} && (6.92) \\ &= \boldsymbol{x}\frac{\boldsymbol{x}^t \boldsymbol{F}}{\overline{\boldsymbol{K}} + (\alpha - 1)\overline{\boldsymbol{K}_i}} && (6.93) \end{aligned}$$

This is basically a scaling of the original deflection. The scaling factor is defined by the increase of the strain energy of the modified part.

6.7.2 Load path interpretation

As seen in the previous subsection with one vector in the reduction basis the estimation is restricted to scaling of the original design. If we add derivative modes or residual vectors to the $\boldsymbol{\Phi}$ matrix we open new load paths for the forces. The generation of a new derivative mode or *load path* was defined in Eq. (6.59) with:

$$\frac{\partial}{\partial \lambda_i}\boldsymbol{x} = -\boldsymbol{X}^{-1}\left[\frac{\partial}{\partial \lambda_i}\boldsymbol{X}\right]\boldsymbol{x} \tag{6.94}$$

The product $\left[\frac{\partial}{\partial \lambda_i}\boldsymbol{X}\right]\boldsymbol{x}$ can contain only high values if the extremely sparse matrix $\left[\frac{\partial}{\partial \lambda_i}\boldsymbol{X}\right]$ has high values at degrees of freedom where the displacement vector $\boldsymbol{x}$ has high deflection.

Therefore it is implicitly described by our formulation that stiffening the structure between displacement degrees of freedom with high deflection has the largest impact on the generation of new load paths. A well known and very intuitive method for engineers to stiffen structures.

7 Computational complexities and performance considerations

In this chapter we will derive an estimation of the amount of *floating point operations* (FLOPS) necessary to solve the equations given in the previous chapter. Based on these results we are also able to build an *a priori* interior estimation of the computing effort for an *approximate response manifold*. This will be a crucial part of judging the introduced ARM method for the desired area of application such as multi-criteria optimization in terms of performance and turnaround time.

7.1 Floating point operations and performance

For solving numerical problems we assume that we have a *central processing unit*, called *CPU*, which carries out the operations. In order to forecast the time required by a whole calculation the following components have to be considered:

- how much numerical data needs to be processed?
- where is the numerical data stored?
- how fast is the data transferred to the *CPU*?
- how fast can one unit (floating point number) be processed?
- how, how fast and how many resulting data has to be stored?

Throughout this thesis we measure the arithmetic requirements by counting the necessary floating point operations. We distinguish between the following floating point operations which all are counted as one FLOP:

- addition, referred as an ADD
- multiplication, referred as a MULT
- and for modern computing architectures a multiplication in combination with an addition, referred as a MPYADD

Many of the above mentioned steps are heavily dependent on the setup of the calculation engine and implementation of the software. Their timing efforts can many times only be measured during calculation and therefore be estimated empirically. But, as all of these steps have to be in a logical order, they can be executed only sequentially. Therefore having an accurate time measurement for one step we have at least an estimate for the minimum time required. Focusing on the CPU, which effectively carries out the calculation, the minimum time required can be calculated if it is known how many operations have to be done in total and how fast the CPU can operate them per time unit (FLOP-rate). As the estimation of the required FLOPS can be done analytically, we have a measurement independent from hard- and software.

Nevertheless we should not forget that this always underestimates the required time, because the process of shuffling the data from storage to CPU and back via different levels of hierarchic memory, such as disk storage, memory storage and different levels of cache, is a time consuming and complex task and is extremely dependent on the implementation of the program and the setup of the machine.

7.2 Floating point operation counts for basic matrix operations

For operations such as matrix addition and multiplication of full matrixes the counting of operations is shown. For the solution of a system of linear equations and for operations of sparse and complex matrices the necessary FLOPS are derived from George, Liu [31], Parlett [107], and Komzsik [76].

7.2.1 Matrix addition and subtraction

Matrix addition is an element by element operation.

$$C = A + B \quad \text{with } A, B, C \in \mathbb{R}^{m \times n} \tag{7.1}$$

$$c_{ij} = a_{ij} + b_{ij} \quad \forall \quad i = 1, \ldots, m; j = 1, \ldots, n \tag{7.2}$$

Examining Eq. (7.2) it is obviously that for the addition of two matrices the following amount of operations is needed:

FLOPS($C = A + B$)	full	sparse
real	mn	$mn\rho_C$
complex	$2mn$	$2mn\rho_C$

where ρ_C is the density of the resultant C, if resultant 0's are stored, but matching zero inputs are not added.

7.2.2 Matrix multiplication

Matrix multiplication is defined via:

$$C = A \cdot B \quad \text{with } A \in \mathbb{R}^{m \times l}, B \in \mathbb{R}^{l \times n}, C \in \mathbb{R}^{m \times n} \tag{7.3}$$

$$c_{ij} = \sum_{k=1}^{l} a_{ik} b_{kj} \quad \forall \quad i = 1, \ldots, m; j = 1, \ldots, n \tag{7.4}$$

This results in mnl MPYADD operations for full real matrices.

For complex operations we have to consider the multiplication of two complex numbers which require four MPYADDs:

$$z_1 = x_1 + \mathrm{i}{\cdot}y_1 \qquad x_1, y_1 \in \mathbb{R}, z_1 \in \mathbb{C} \tag{7.5}$$

$$z_2 = x_2 + \mathrm{i}{\cdot}y_2 \qquad x_2, y_2 \in \mathbb{R}, z_2 \in \mathbb{C} \tag{7.6}$$

$$z_3 = z_1 z_2 \tag{7.7}$$

$$= (x_1 x_2 - y_1 y_2) + \mathrm{i}{\cdot}(x_1 y_2 + x_2 y_1) \tag{7.8}$$

FLOPS($C = A \cdot B$)	full	sparse
real	lmn	$lnm\rho_C$
complex	$4lmn$	$4lnm\rho_C$

7.2.3 Solving symmetric systems

To solve the system of equations

$$AX = B \quad A \in R^{n \times n}, X, B \in R^{n \times m} \tag{7.9}$$

the inverse of A, A^{-1} is explicitly not computed. Instead a decomposition (DCMP)

$$A = LU \qquad \text{for non-symmetric matrices} \tag{7.10}$$

$$A = LDL^t \qquad \text{for symmetric matrices} \tag{7.11}$$

$$= CC^t \qquad \text{Cholesky form for symmetric positive definite matrices} \tag{7.12}$$

and a forward backward substitution (FBS) are carried out. L is called the lower triangular factor, U the upper triangular factor and C the Cholesky factor. Using Eq. (7.10) Eq. (7.9) becomes

$$LY = B \qquad \text{forward step} \tag{7.13}$$

$$UX = Y \qquad \text{backward step} \tag{7.14}$$

For the decomposition of a symmetric matrix A with front-width f, which will be defined in 7.3, the approximate complexity is [31]:

FLOPS(DCMP($A \rightarrow LU$))	full ($f = n$)	sparse
real	$\frac{1}{2}n^3$	$\frac{1}{2}nf^2$
complex	$2n^3$	$2nf^2$

The computational complexity of the forward backward substitution is approximately [31]:

FLOPS(FBS(B, L, U))	full ($c_\emptyset = \frac{1}{2}n^2$)	sparse
real	mn^2	$2mc_\emptyset$
complex	$4mn^2$	$8mc_\emptyset$

Where $c_\emptyset$ is the number of non-zeroes in the Cholesky factor C.

7.2.4 Gram-Schmidt ortho-normalization

The $[n \times s]$ matrix Φ defined in Eq. (5.78) must have full rank in order to serve as a basis for our design space. For reasons of numerical accuracy we use an ortho-normalized Φ. The standard procedure of ortho-normalization is, beside a singular value decomposition (see [111]), the Gram-Schmidt ortho-normalization (see KOMZSIK [75]) which we will use here. As the ortho-normalization is a performance critical task in the preparation phase of the ARM method we have to consider its computational complexity here in order to judge the new method. It must be noted that this is a lower estimate of the required FLOPS, due to the fact that standard Gram-Schmidt ortho-normalization can result in bad round-off errors due to finite precision arithmetic and *modified* Gram-Schmidt or other more expansive procedures must be used.

Theorem 7.1 *The computational complexity of Gram-Schmidt ortho-normalization is at most* $n(s^2 + s)$.

Proof 7.1 *The procedure of ortho-normalization takes the* $[n \times s]$ *matrix*

$$M = [m_1, m_2, \ldots, m_s] \tag{7.15}$$

of rank $t \leq s$*, where* m_i *are column vectors of size* n*, and produces the matrix*

$$\Phi = [\phi_1, \phi_2, \ldots, \phi_t] \tag{7.16}$$

with

$$< \phi_i, \phi_j >= \delta_{ij} \quad \textit{Kronecker delta} \tag{7.17}$$

For the proof we assume $t = s$.

$$for \quad i = 1 : s \tag{7.18}$$

$$\overline{\phi}_i = m_i - \sum_{k=1}^{i-1} < \phi_k, m_i > \phi_k \tag{7.19}$$

$$\phi_i = \frac{1}{\|\overline{\phi}_i\|} \overline{\phi}_i \tag{7.20}$$

$$endfor \qquad \Phi = [\phi_1, \ldots, \phi_i] \tag{7.21}$$

$$\tag{7.22}$$

Examining Eq. (7.19) we see a scalar product requiring n MPYADDs, and an additional n MPYADDs for the vector subtraction. This sums up to $2n$ FLOPS. This is done $i - 1$ times. The normalization takes another $2n$ FLOPS. So one loop requires $2n+(i-1)*2n = 2ni$ FLOPS. The total loop requires then

$$\sum_{i=1}^{s} 2ni = 2n \sum_{i=1}^{s} i \tag{7.23}$$

$$= 2n \frac{s(s+1)}{2} \tag{7.24}$$

$$= n(s^2 + s) \tag{7.25}$$

operations. q.e.d.

FLOPS(ORTHO(M))	full
real	$n(s^2 + s)$
complex	$4n(s^2 + s)$

7.3 Matrix properties for structural problems

Looking at the discretized equations for solving structural problems in Eq. (4.88) in the frequency domain we see 3 matrices $\boldsymbol{K}, \boldsymbol{B}$ and $\boldsymbol{M}$ which form the dynamic stiffness $\boldsymbol{X} = (-\omega^2\boldsymbol{M}+i\omega\boldsymbol{B}+\boldsymbol{K})$. The structure of $\boldsymbol{X}$ is in general dominated by $\boldsymbol{K}$, because $\boldsymbol{M}$ is in most cases nearly diagonal (lumped mass approach, see also ZIENKIEWICZ [125]) and $\boldsymbol{B}$ is mostly either extremely sparse or can have a similar pattern then $\boldsymbol{K}$.

In order to give reasonable estimations for the computational efforts, required by solving structural problems we have to know the following properties of a matrix $\boldsymbol{X}$ depending on the model size n:

- number of rows **nrow**(X) and columns **ncol**(X), both are n as the matrix is square and symmetric

- matrix density $\mathbf{density}(X)$, which is the number of non-zeroes divided by the matrix size $\mathbf{nrow}(X) \cdot \mathbf{ncol}(X)$
- number of non-zeroes per row i: $\mathbf{nnzpr}_i(X)$

  $$\mathbf{nnzpr}(X) = \max\{\mathbf{nnzpr}_i(X) \quad | \quad 1 \leq i \leq \mathbf{nrow}(X)\} \tag{7.26}$$

- bandwidth as defined in [17]:

  $$\mathbf{bandwidth}_i(X) = i - \min\{j \quad | \quad X_{ij} \neq 0\} \tag{7.27}$$

  $$\mathbf{bandwidth}(X) = \max\{\mathbf{bandwidth}_i(X) \quad | \quad 1 \leq i \leq \mathbf{nrow}(X)\} \tag{7.28}$$

- front-width, with

  $$\mathbf{frontwidth}_i(X) = |\{k \quad | \quad k > i \quad \text{and} \quad X_{kl} \neq 0 \quad \text{for some} \quad l \leq i\}| \tag{7.29}$$

  $$\mathbf{frontwidth}(X) = \max\{\mathbf{frontwidth}_i(X) \quad | \quad 1 \leq i \leq \mathbf{nrow}(X)\} \tag{7.30}$$

  Note that $\mathbf{frontwidth}_i(X)$ is simply the number of active rows at the i-th step in the factorization and $\mathbf{frontwidth}(X)$ is the frontwidth of the whole matrix. Further we have (George [31]):

  $$\mathbf{bandwidth}(X) \geq \mathbf{frontwidth}(X) \geq \mathbf{nnzpr}(X) \tag{7.31}$$

  Throughout this chapter we may also denote the front-width with the letter f for better readability.

- number of non-zeroes of the Cholesky factor $\mathbf{nnzchol}(X) = \mathbf{nnz}(C)$, where C is the Cholesky factor of X (see Eq. (7.12)). Throughout this chapter we may also denote the number of non-zeroes in the factor with the symbol $c_{\emptyset}$.

To obtain these properties for matrices used in structural analysis we look at the following simplified models representing the basic discretization approaches in the 3-dimensional space (see Fig. 7.1):

- 1-dimensional abstraction with beam elements, BEAM
- 2-dimensional abstraction with quadrilateral, QUAD, or triangular, TRIA, elements
- 3-dimensional model with pentahedral, PENTA or hexahedral, HEXA, elements.

An element with all edges of the same length we will call a *perfect* element. For each of these models we consequently increase the number of degrees of freedom n by subdividing each edge of the basic element with an increased number of new elements m of the same type. Therefore the amount of elements increases proportional to m for one dimensional elements (BEAM), $\approx m^2$ for two dimensional elements such as QUAD and TRIA elements and $\approx m^3$ for three dimensional ones like HEXA and PENTA elements.

We will not look at tetrahedral elements, because a *perfect* tetrahedral element cannot be split into multiple *perfect* tetrahedral elements. For the further discussion we will focus on elements with linear shape functions (see [125]) only.

The stiffness matrix K has in general, for a reasonable geometric topology, the following characteristics:

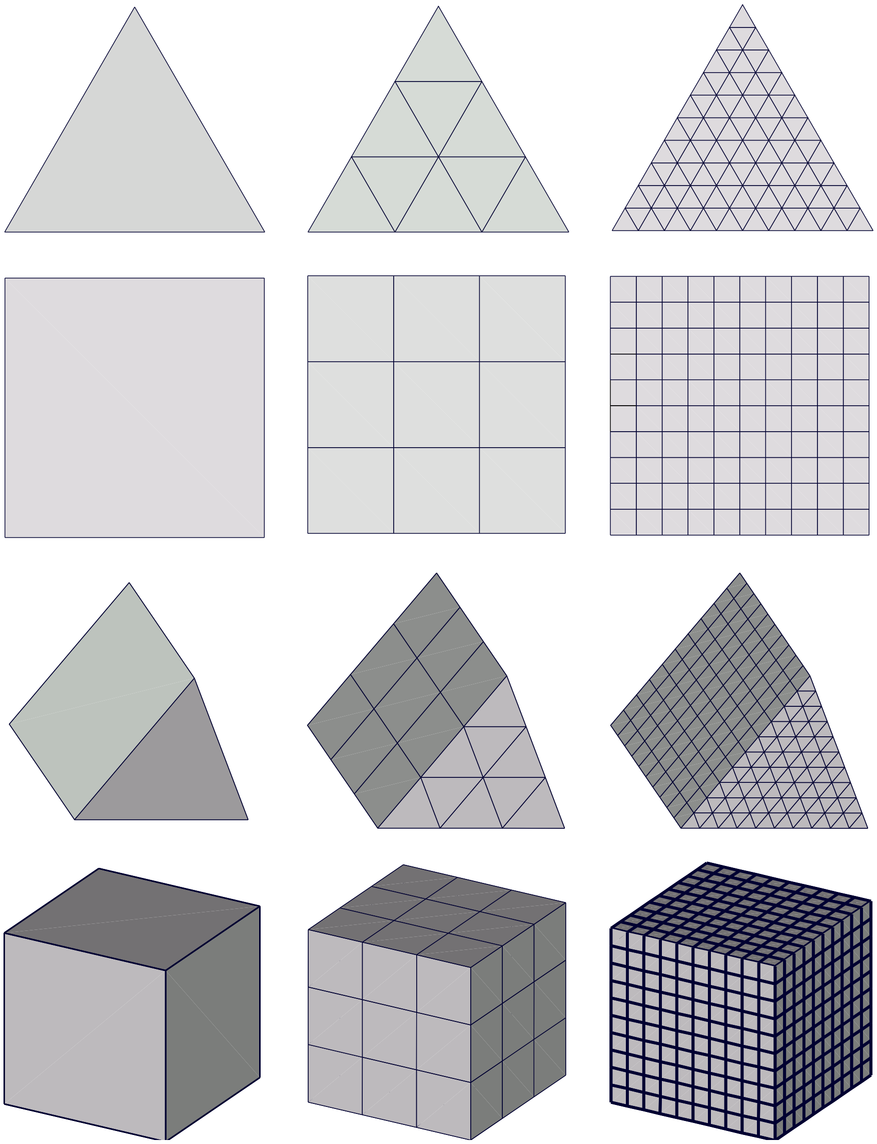


Figure 7.1: Representative models for the matrix property study including the 2D and 3D models for 1, 3 and 10 elements per edge

(linear) element-type (number of nodes)	$\mathbf{nnzpr}(K)$ [1]	$\mathbf{density}(K^{\text{type}})$ see Eq. (7.33)
BEAM (2)	5	$\mathbf{nnzpr}(K^{\text{BEAM}})/n$
TRIA (3)	21	$\mathbf{nnzpr}(K^{\text{TRIA}})/n$
QUAD (4)	24	$\mathbf{nnzpr}(K^{\text{QUAD}})/n$
PENTA (6)	63	$\mathbf{nnzpr}(K^{\text{PENTA}})/n$
HEXA (8)	77	$\mathbf{nnzpr}(K^{\text{HEXA}})/n$

With $\mathbf{nrow}(K) = \mathbf{ncol}(K) = n$ the total amount of terms in matrix K is n^2. Therefore we get an upper bound of the density of sparse stiffness matrices of:

$$\begin{aligned} \mathbf{density}(K^{\text{type}}) &\leq \frac{\mathbf{nnzpr}(K^{\text{type}})n}{n^2} && (7.32) \\ &\leq \frac{\mathbf{nnzpr}(K^{\text{type}})}{n} && (7.33) \end{aligned}$$

7.3.1 Approximation of front-width and non-zeroes in the Cholesky factor

K with $\mathbf{nrow}(K) = n$ will be denoted with K_n. In order to observe the behavior of $\mathbf{frontwidth}(K_n)$ and $\mathbf{nnzchol}(K_n)$ for large as well as small models, we introduce two new variables:

$$\xi = \frac{\log(\mathbf{frontwidth}(K_n))}{\log(n)} \tag{7.34}$$

$$\eta = \frac{\log(\mathbf{nnzchol}(K_n))}{\log(n)} \tag{7.35}$$

Table 7.3.1 shows the development of the amount of elements, the number of degrees of freedom, and the characteristic properties of the stiffness matrix such as $\mathbf{frontwidth}(K_n)$ and $\mathbf{nnzchol}(K_n)$ for these models and a selective large range of degrees of freedom.

elements per edge (m)	total number of elements	degrees of freedom (n) $\bar{n} = (m+1)$	**frontwidth**	(**nnzchol** ± 1000) /1000	ξ	η
type BEAM (6 DOF/node)	m	$6\bar{n}$				
1	1	12	4	1	0.56	2.78
3	3	24	6	1	0.56	2.17
10	10	66	22	1	0.74	1.65
31	31	192	12	2	0.47	1.45
100	100	606	28	8	0.52	1.40
316	316	1902	22	20	0.41	1.31
1000	1000	6006	34	91	0.41	1.31
3162	3162	18978	33	233	0.35	1.25

elements per edge (m)	total number of elements	degrees of freedom (n) $\bar{n} = (m+1)$	**frontwidth**	(**nnzchol** ±1000) /1000	ξ	η
10000	10000	60006	37	1062	0.33	1.26
31622	31622	189738	37	3639	0.30	1.24
100000	100000	600006	37	11770	0.27	1.22
316227	316227	1897368	37	37487	0.25	1.21
type TRIA (5 DOF/node)	m^2	$5/2 \cdot \bar{n} \cdot (\bar{n}+1)$				
1	1	15	9	1	0.81	2.55
2	4	30	12	1	0.73	2.03
4	16	75	30	2	0.79	1.76
8	64	225	51	6	0.73	1.61
16	256	765	87	32	0.67	1.56
32	1024	2805	117	155	0.60	1.51
64	4096	10725	216	793	0.58	1.46
128	16384	41925	420	3957	0.57	1.43
256	65536	165765	6912	19051	0.57	1.39
1000	1000000	2507505	3393	401499	0.55	1.34
2073	4297329	10758875	8931	2144215	0.56	1.33
type QUAD (5 DOF/node)	m^2	$5\bar{n}^2$				
1	1	20	12	1	0.83	2.31
2	4	45	15	1	0.71	1.81
4	16	125	33	3	0.72	1.66
8	64	405	57	15	0.67	1.60
16	256	1445	99	75	0.63	1.54
32	1024	5445	195	399	0.61	1.50
64	4096	21125	321	1958	0.58	1.45
128	16384	83205	729	9554	0.58	1.42
256	65536	330245	1323	45969	0.57	1.39
512	262144	1315840	2913	221332	0.57	1.36
type PENTA (3 DOF/node)	m^3	$3/2 \cdot \bar{n}^2 \cdot (\bar{n}+1)$				
1	1	18	18	1	1.00	2.39
2	8	54	27	1	0.83	1.73
3	27	120	48	4	0.80	1.73
4	64	378	102	20	0.78	1.67
6	216	864	186	78	0.77	1.67

elements per edge (m)	total number of elements	degrees of freedom (n) $\bar{n} = (m+1)$	**frontwidth**	(**nnzchol** ±1000) /1000	ξ	η
8	512	2808	417	455	0.76	1.64
12	1728	9234	957	2724	0.75	1.62
18	5832	27378	2085	14700	0.75	1.62
26	17576	91260	5757	85438	0.76	1.60
39	59319	202878	8562	260079	0.74	1.59
51	132651	521850	17316	1038091	0.74	1.58
type HEXA (3 DOF/node)	m^3	$3\bar{n}^3$				
1	1	24	24	1	1.00	2.17
2	8	81	36	3	0.82	1.82
3	27	192	72	9	0.81	1.73
5	125	648	168	56	0.79	1.69
8	512	2187	351	350	0.76	1.66
11	1331	5184	648	1244	0.76	1.64
17	4913	17496	1458	7564	0.75	1.62
27	19683	65856	4176	56559	0.75	1.61
38	54872	177957	8079	233701	0.74	1.59
57	185193	585336	21294	1427257	0.75	1.59

Table 7.1: Table of matrix properties for the simplified models

7.3.1.1 Behavior of the front-width

The bandwidth and with it the front-width depend extremely on the ordering of the degrees of freedom and therefore on the numbering scheme of the nodes in the model. As highly sophisticated reordering schemes, such as Metis [52], used in most solvers today, are not topic of this work we will use a straight forward numbering scheme and can give therefore only an upper estimate of the asymptotic behavior of the front-width. Nevertheless we will show that for our simplified models this numbering scene is close to an optimum concerning a minimum front-width. Without restricting any generality we can focus for each dimension (1-D: BEAM, 2-D:QUAD, 3-D:HEXA) on a model with 2 elements (3 nodes) per edge. This is due to the fact that the intermediate node has a full connection to the neighbors. As seen in Fig. 7.11 and Fig. 7.12 the front-width for the TRIA model shows almost the same behavior as the QUAD model. This is also true for the PENTA model and the HEXA model.

Looking at the BEAM model in Fig. 7.2 we see that the grid with the number ID is connected with the grids $ID-1$ and $ID+1$ giving a bandwidth of:

$$\mathbf{bandwidth}_{ID}(K_n^{\text{BEAM}}) = const. \cdot (ID + 1 - (ID - 1)) \tag{7.36}$$

$$\mathbf{bandwidth}_{ID}(K_n^{\text{BEAM}}) = const. \tag{7.37}$$

For the two dimensional model we see a connectivity between ID ,$ID-m-1$, and $ID+m+1$ (Fig. 7.2), where m is the number of grids per edge. This leads to a bandwidth of:

$$\mathbf{bandwidth}_{ID}(K_n^{\text{QUAD}}) = const. \cdot (ID + m + 1 - (ID - m - 1)) \tag{7.38}$$

$$\mathbf{bandwidth}_{ID}(K_n^{\text{QUAD}}) = const. \cdot m + const. \tag{7.39}$$

For the three dimensional model we have to increase the offset of the nodes with m^2 (Fig. 7.3):

$$\mathbf{bandwidth}_{ID}(K_n^{\text{HEXA}}) = const. \cdot (ID + m^2 + m + 1 - (ID - m^2 - m - 1)) \tag{7.40}$$

$$\mathbf{bandwidth}_{ID}(K_n^{\text{HEXA}}) = const. \cdot m^2 + const. m + const. \tag{7.41}$$

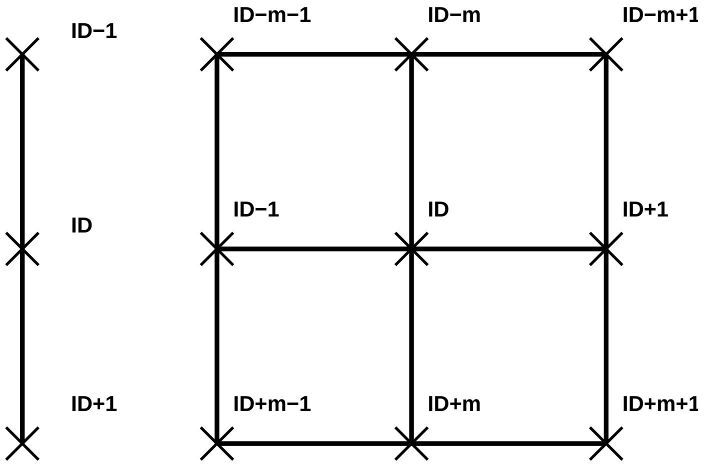


Figure 7.2: Numbering scheme for the 1D BEAM and 2D QUAD model.

Using Eq. (7.37) to Eq. (7.39) and Eq. (7.31) we can estimate the complexity of the front-width for the demonstrated models. Using the data in tabular 7.3.1 we then fit the mathematical models for the front-width in a least square sense.

The BEAM model's front-width does also not depend on the matrix size n at all. Therefore the front-width is:

$$\mathbf{frontwidth}(K_n^{\text{BEAM}}) = c \quad \text{with } c \in \mathbb{R} \tag{7.42}$$

For the QUAD model we saw a linear dependency on m which is the number of grids per edge. For a 2 dimensional model this means that $m \approx \sqrt{n}$ because the amount of degrees of freedom is proportional to the total number of grids in the model. Again respecting Eq. (7.31) we assume

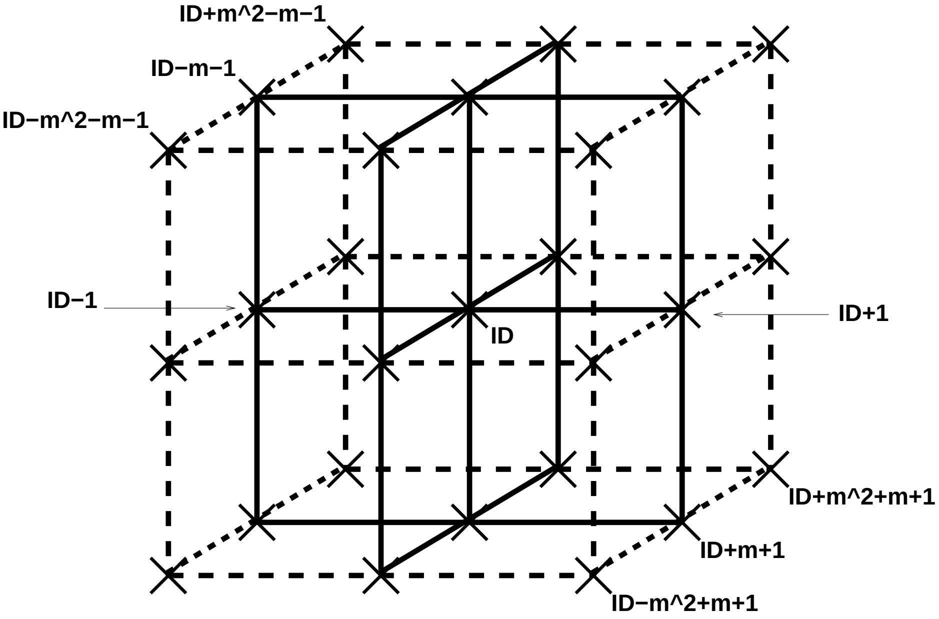


Figure 7.3: Numbering scheme for the 3D HEXA model. $m\hat{}2$ means m^2.

the same complexity for the front-width as for the bandwidth we get the following test-function for the front-width of the QUAD model:

$$\mathbf{frontwidth}(K_n^{\text{QUAD}}) = b \cdot \sqrt{n} + c \quad \text{with } b, c \in \mathbb{R} \tag{7.43}$$

Applying the same procedure on the HEXA model we see a quadratic dependency on $m \approx \sqrt[3]{n}$ according to the 3D nature of this model. For the test function we therefore apply the following function:

$$\mathbf{frontwidth}(K_n^{\text{HEXA}}) = a \cdot n^{2/3} + b \cdot n^{1/3} + c \quad \text{with } a, b, c \in \mathbb{R} \tag{7.44}$$

In summary we get for the polynomial pre factors to fit the front-width:

	a	b	c
1D/BEAM			37.000
2D/QUAD		2.587	0.261
3D/HEXA	2.803	-9.663	28.288

The observed and the fitted data can be seen in Fig. 7.4, Fig. 7.5 and Fig. 7.6.

7.3.1.2 Behavior of the number of non-zeroes in the Cholesky factor

For the number of non-zeros in the Cholesky factor, $\mathbf{nnzchol}(K_n)$, we observe in a log/log plot (Fig. 7.7) a linear behavior between the amount of DOF n and the $\mathbf{nnzchol}(K_n)$.

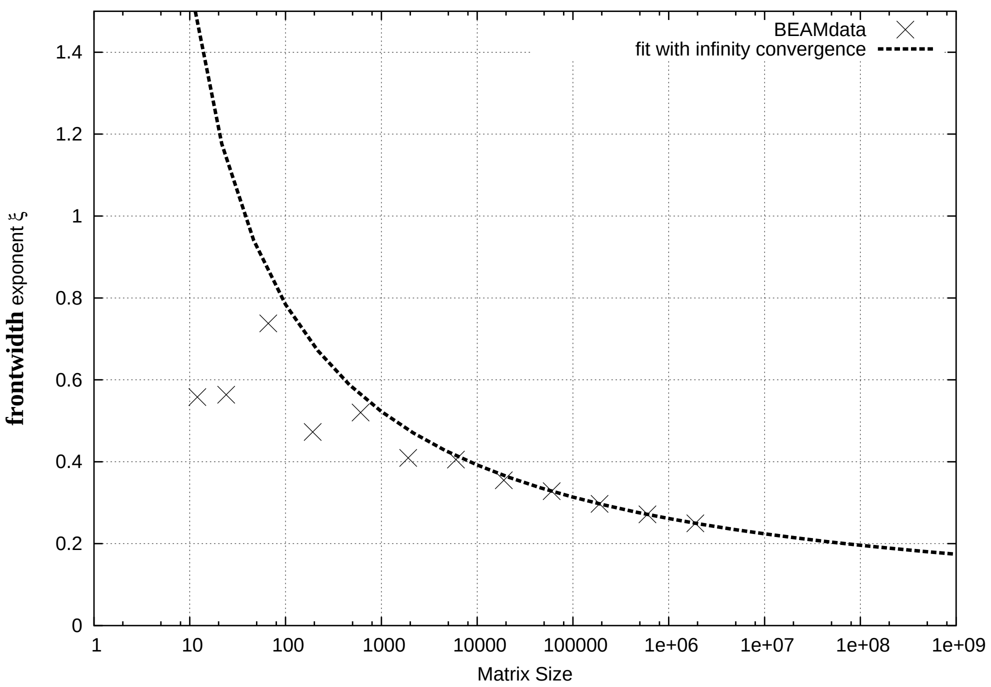


Figure 7.4: Front-width data of the artificial model and the fitted front-width for the BEAM model.

$$\log\left(\mathbf{nnzchol}(K_n)\right) = d\log\left(n\right) + \bar{e} \tag{7.45}$$

$$\leftrightarrow \mathbf{nnzchol}(K_n) = \exp(d\log(n) + \bar{e}) \tag{7.46}$$

$$\leftrightarrow \mathbf{nnzchol}(K_n) = en^d \quad \text{with } e = exp(\bar{e}) \tag{7.47}$$

Here d and e are model specific constants. The artificial data and the fitted curves can be seen in Fig. 7.8, Fig. 7.9, and Fig. 7.10.

In summary we get for the constants to fit the number of non-zeroes in the Cholesky factor:

	d	e
1D/BEAM	1.105	4.86
2D/QUAD	1.180	13.80
3D/HEXA	1.492	3.56

7.3.1.3 Behavior of real world models

In addition to the study with the simplified models we can also see the properties of the stiffness matrix for *real world models* in Fig. 7.11 and Fig. 7.12. This data represents characteristic stiffness matrices, mostly dominated by QUAD and TRIA elements, of $\approx 50,000$ finite element

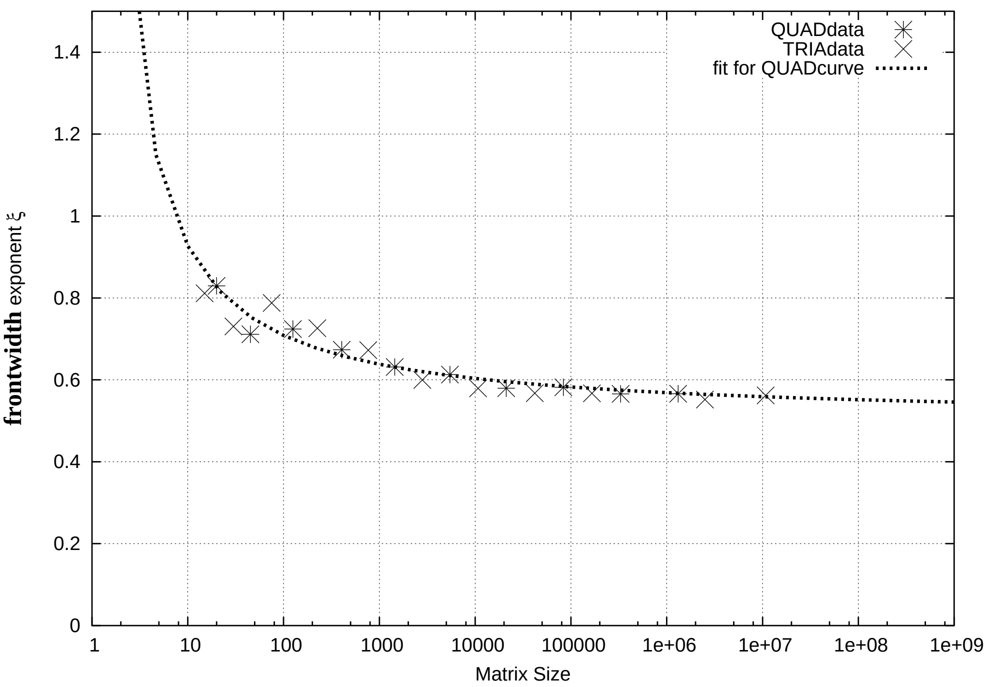


Figure 7.5: Front-width data of the artificial model and the fitted front-width for the 2D models.

simulations. The data was produced during the development process of automotive structures over several years at a major car company [2].

It can be clearly seen that the values for ξ and η for these models are close to these of the 2D simplified models. The difference is explained by the following properties, which are not covered within the simplified models:

- mixture of 1D, 2D and 3D element types
- different topology of the underlying geometry inserting holes and closures in the topology of the mesh
- single and multi-point constraints caused by rigid elements and other boundary conditions, which increase the matrix density, by eliminating degrees of freedom

---

[2] data is courtesy of BMW AG, collected over the years 1998-2003 in the development and research center FIZ

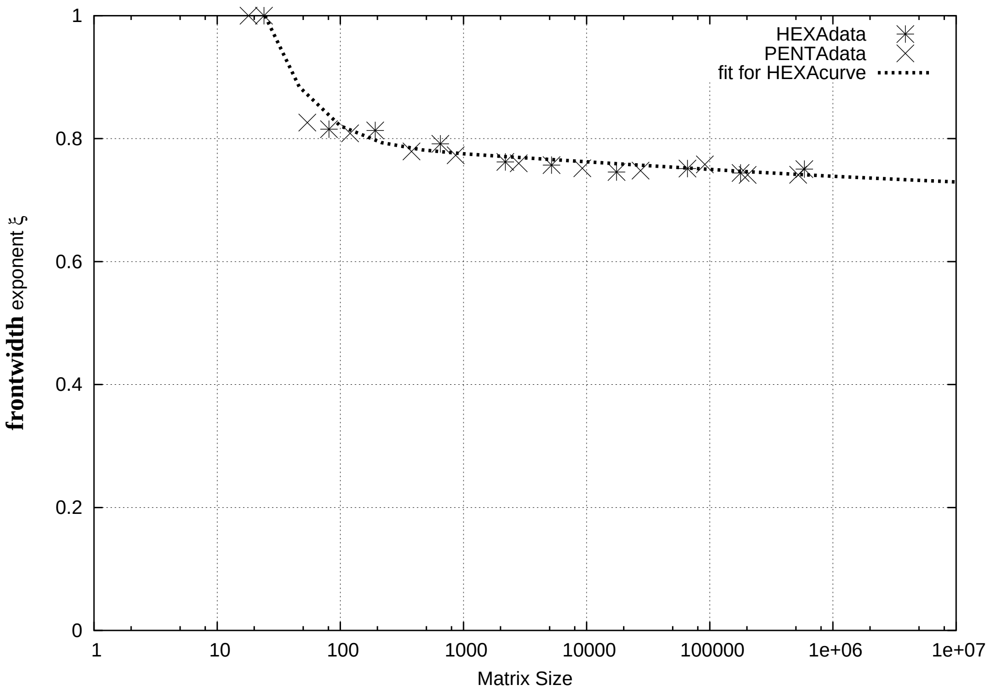


Figure 7.6: Front-width data of the artificial model and the fitted front-width for the 3D models.

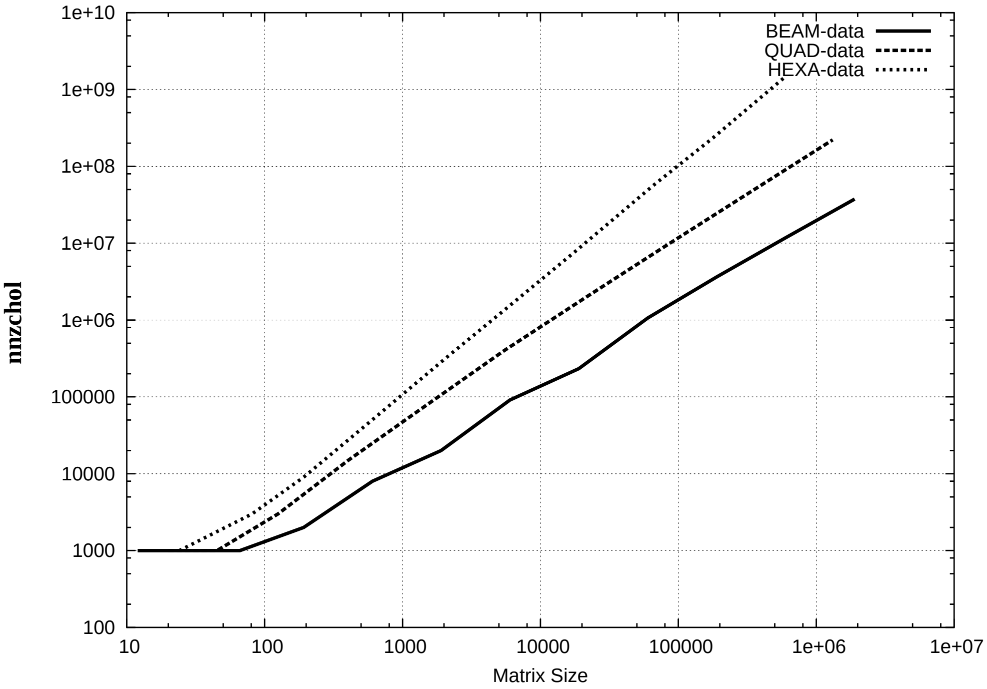


Figure 7.7: The log/log plot between the degrees of freedom and the number of non-zeros in the Cholesky factor shows a linear dependency for larger amounts of degrees of freedom in the log scale.

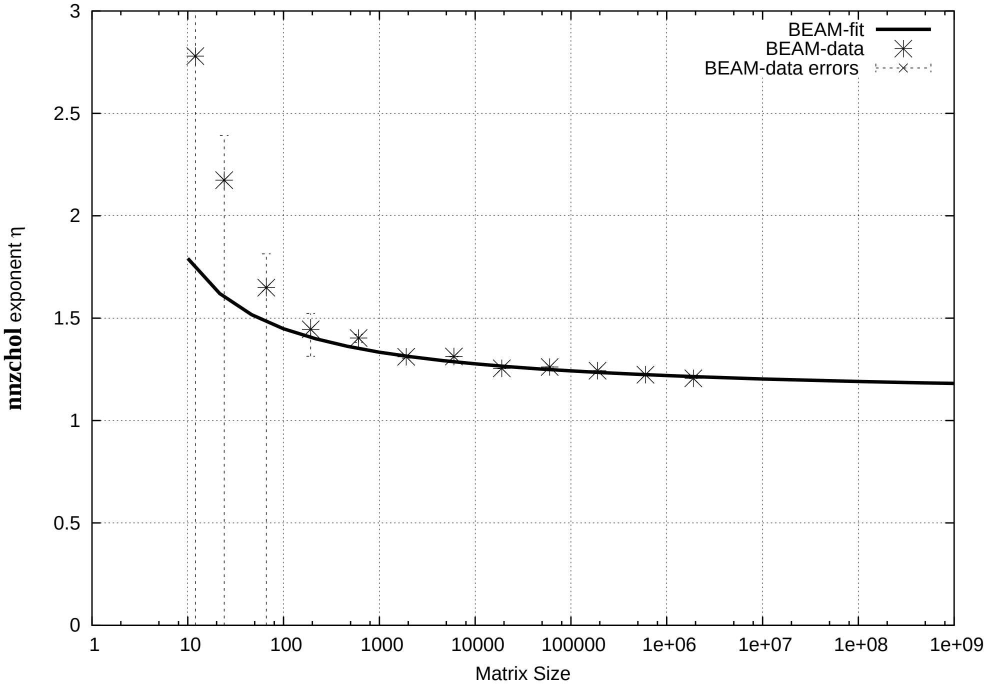


Figure 7.8: Number of non-zeros in the Cholesky factor of the artificial model and the fitted data for the 1D model.

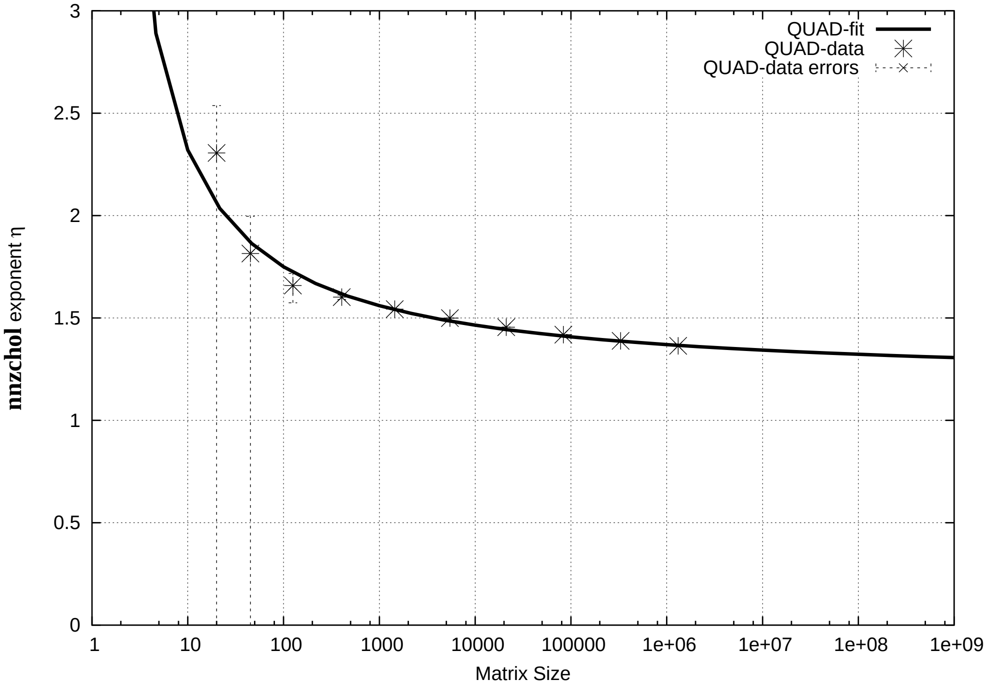


Figure 7.9: Number of non-zeros in the Cholesky factor of the artificial model and the fitted data for the 2D model.

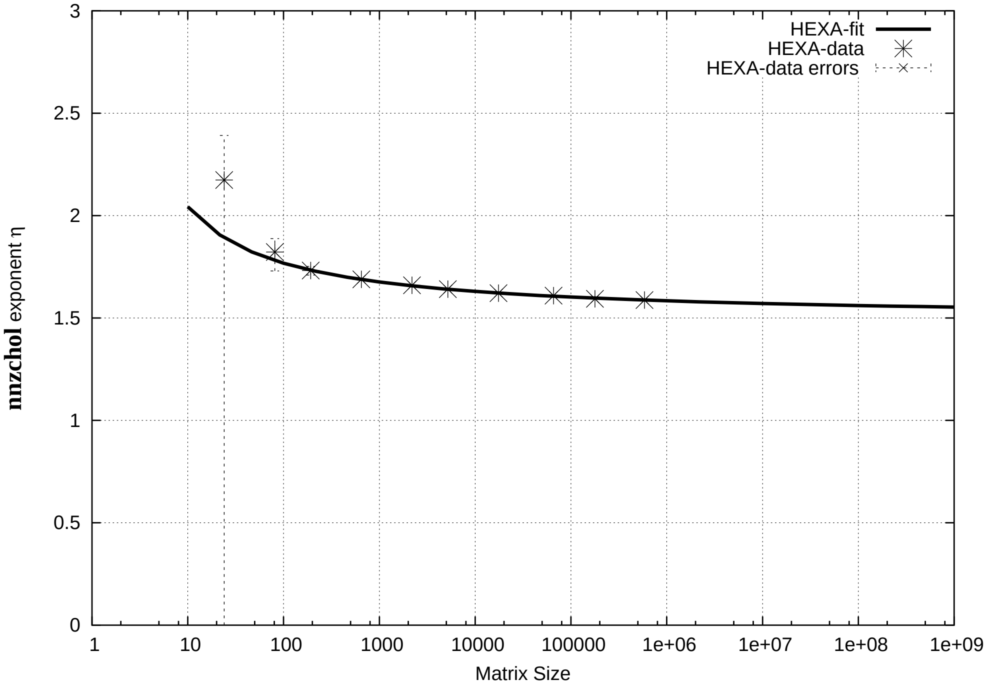


Figure 7.10: Number of non-zeros in the Cholesky factor of the artificial model and the fitted data for the 3D model.

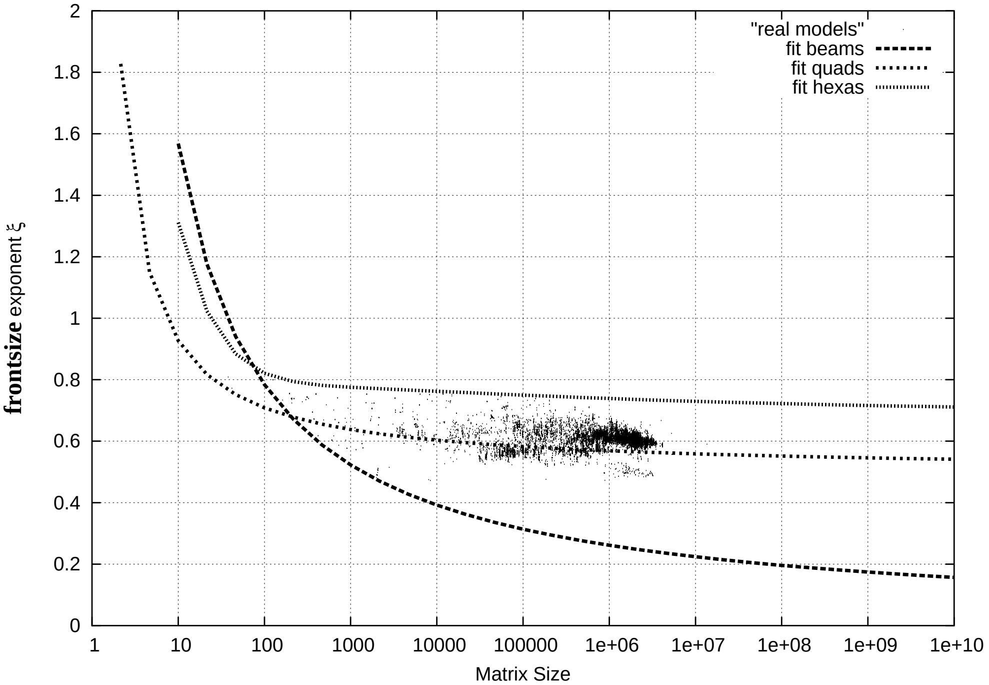


Figure 7.11: $\xi = \log(\mathbf{frontwidth}(K_n))/\log(n)$ describing the front-width depending on n. The real model data, mainly consisting of quadrilateral and triangular elements and some 3D elements, is, as predicted, close to the curve for the 2D elements.

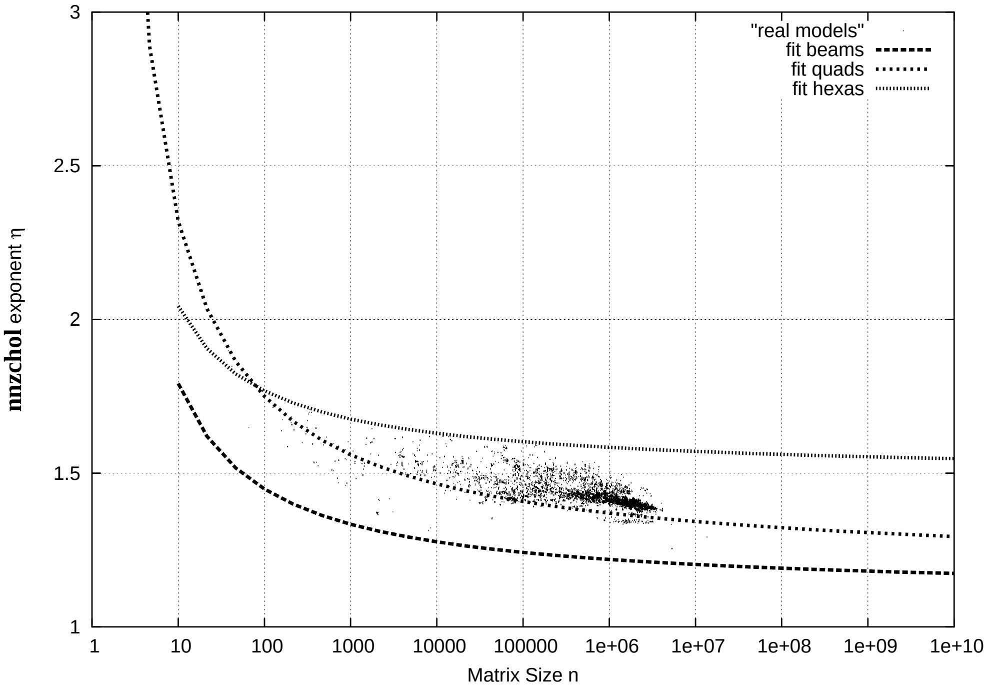


Figure 7.12: $\eta = \log(\mathbf{nnzchol}(K_n))/\log(n)$ describing the amount of non-zeros in the Cholesky factor depending on n. The real model data, mainly consisting of quadrilateral and triangular elements and some 3D elements, is, as predicted, close to the curve for the 2D elements.

7.4 Computational complexity for a structural optimization problem

To judge the ARM method's computational efficiency we have to add up the FLOPS of all the steps of the solution procedure. The (linear) solution of a finite element model, can be described basically with the following steps:

- element matrix generation: $K_i, M_i, B_i \quad \text{for} \quad i = 1, \ldots, N_{ele}$
- element matrix Assembly: $K_g = \sum_{i=1}^{N_{ele}} K_i, M_g = \sum_{i=1}^{N_{ele}} M_i, B_g = \sum_{i=1}^{N_{ele}} C_i$,
- load generation
- single- and multi-point elimination, $K_g \rightarrow K$, and respectively M and B
- decomposition of dynamic stiffness matrix $X = -\omega^2 M + \mathrm{i} \cdot \omega B + K$
- forward- and backward substitution

In general the decomposition and the forward-backward substitution are the performance critical tasks.

For the optimization process we assume that we need σ_{ng} function evaluations for non gradient based search algorithms or σ_g iterations including τ gradient evaluations at each iteration for gradient based optimization algorithms. d shall be the number of *parameters*.

7.4.1 Standard approach

7.4.1.1 Non-sensitivity based solution

Counting only the more time consuming tasks like DCMP and FBS, further assuming one right-hand side and the complexities defined in 7.2.3 and 7.2.3, the necessary floating point operations are:

Operation	FLOPS
decomposition of stiffness matrix	$\frac{1}{2}nf^2$
forward-backward substitution (1 right-hand side)	$2c_\emptyset$
Total	$\frac{1}{2}nf^2 + 2c_\emptyset$

7.4.1.2 Gradient based solution

For gradient based solutions we have to calculate d gradients for each function evaluation. As defined in Eq. (6.59) the gradient of the displacements, according to the i-th variable is given by:

$$\frac{\partial 1}{\partial \lambda_i}\boldsymbol{x} = \boldsymbol{X}^{-1}\left[\frac{\partial 1}{\partial \lambda_i}\boldsymbol{X}\right]\boldsymbol{x} \tag{7.48}$$

As the decomposed $\boldsymbol{X}$ is already available from the initial solution of the system, the procedure of calculating the derivative of the displacements requires one matrix vector operation and one forward-backward substitution only. The amount of floating point operations required by the matrix vector operation depends on the density of the differential matrix $\frac{\partial 1}{\partial \lambda_i}\boldsymbol{X}$, cannot be predicted in advance. Nevertheless some assumptions can be made, which are reasonable for many cases.

The structure of the system matrix X has the following pattern:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}^i_{11} & \boldsymbol{X}^i_{12} & 0 \\ \boldsymbol{X}^i_{21} & \boldsymbol{X}^i_{22} + \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ 0 & \boldsymbol{R}_{21} & \boldsymbol{R}_{22} \end{pmatrix} \tag{7.49}$$

Where $\boldsymbol{X}^i$ denotes the sub matrix of $\boldsymbol{X}$ belonging to part i and $\boldsymbol{R}$ represents the rest of the structure defined by the other parts.

The derivative of the system matrix according to the design variable i has then the following sparsity pattern:

$$\mathrm{Sparsity}[\frac{\partial}{\partial \mathrm{\lambda_i}}\mathrm{X}] = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{7.50}$$

Here 1 denotes entries in the matrix.

So let:

$$n_i \quad \text{be the amount of DOF affected by variable } i \tag{7.51}$$

$$l^i_\emptyset \quad \text{be the number of non-zeros per row in the mean for variable } i \tag{7.52}$$

$$\overline{n} = \frac{1}{d}\sum_{i=1}^{d} n_i \quad \text{the average amount of DOF affected by each design variable} \tag{7.53}$$

$$\overline{l_\emptyset} = \frac{\sum_{i=1}^{d} n_i l^i_\emptyset}{\sum_{i=1}^{d} n_i} \quad \text{the mean amount of non-zeroes per row of each design variable} \tag{7.54}$$

Theorem 7.2 *The amount of floating point operations to calculate all d right-hand side vectors for the FBS in Eq. (7.48) is:*

$$d\overline{n}\overline{l_\emptyset} \tag{7.55}$$

Proof 7.2 *We basically will carry out one forward backward substitution to calculate the gradients in one shot:*

$$\left[\frac{\partial 1}{\partial \lambda_1}\boldsymbol{x}, \frac{\partial 1}{\partial \lambda_2}\boldsymbol{x}, \ldots, \frac{\partial 1}{\partial \lambda_2}\boldsymbol{x}\right] = \boldsymbol{X}^{-1}\boldsymbol{Y} \tag{7.56}$$

$$\textit{with} \quad Y = \left[\left(\frac{\partial 1}{\partial \lambda_1}\boldsymbol{X}\right), \left(\frac{\partial 1}{\partial \lambda_1}\boldsymbol{X}\right), \ldots, \left(\frac{\partial 1}{\partial \lambda_1}\boldsymbol{X}\right)\right] \tag{7.57}$$

As x is a $n \times 1$ vector the matrix product $[\partial_i X]U$ requires $n_i \cdot l^i_\emptyset$ FLOPS. Therefore the number of FLOPS to form Y is:

$$\sum_{i=1}^{d} n_i l^i_\emptyset \tag{7.58}$$

Based on Eq. (7.54) we have:

$$\overline{l_\emptyset} = \frac{\sum_{i=1}^{d} n_i l^i_\emptyset}{\sum_{i=1}^{d} n_i} \tag{7.59}$$

$$\longleftrightarrow \left(\sum_{i=1}^{d} n_i\right)\overline{l_\emptyset} = \sum_{i=1}^{d} n_i l^i_\emptyset \tag{7.60}$$

$$\longleftrightarrow d\left(\frac{1}{d}\sum_{i=1}^{d} n_i\right)\overline{l_\emptyset} = \sum_{i=1}^{d} n_i l^i_\emptyset \tag{7.61}$$

$$\longleftrightarrow d\overline{n}\overline{l_\emptyset} = \sum_{i=1}^{d} n_i l^i_\emptyset \quad \textit{using Eq. (7.53)} \tag{7.62}$$

.q.e.d

Operation	FLOPS
decomposition of general stiffness matrix	$\frac{1}{2}nf^2$
forward-backward substitution	$2c_\emptyset$
matrix vector products for all i $\left[\frac{\partial 1}{\partial \lambda_i}\boldsymbol{X}\right]\boldsymbol{x}$	$d\overline{n}\overline{l_\emptyset}$
d forward-backward substitutions	$2dc_\emptyset$
Total	$\frac{1}{2}nf^2 + 2(1+d)c_\emptyset + d\overline{n}\overline{l_\emptyset}$

7.4.2 *Approximate Response Manifold* (ARM) based solution

As shown in section 6.5 the ARM based solution is a two stage procedure. In the first stage the embedding space of the response manifold is generated. In the second step the solution is carried out in this space. It was demonstrated that the solution vectors of the ARM based solution are equivalent to the shape functions of the element formulation in terms of generalized coordinates. The only difference is that the solution manifold embedding space $\mathcal{D}$ is considered as the solution space instead of the discretization space $\mathcal{S}$. Therefore our modal basis Φ can be seen as the *shape functions* of an *approximated response space.*

Concerning the discretization using finite elements, there is no exact definition of the *right* or *perfect* discretization. The discretization is more based on general rules, experience concerning the gained accuracy and available computing resources. The in many cases valid statement *the smaller the elements, the more accurate the solution* is outweighed by the computing power and the needed *time to solution.* A better geometric approximation through smaller elements is only one factor for improving accuracy. In the same manner as we assume a general desired element edge length for discretizing our geometry we will assume the same amount of derivative modes for each design variable when constructing the manifold encapsulating subspace $\mathcal{D}$.

The numerical procedures for generating the base of an approximated manifold and their respective computational complexities are, for a static structural problem and assuming that we do not add additional residual modes:

	Operation	FLOPS O(n,s)
	calculation of the fix points	
1	decomposition of general stiffness matrix	$\frac{1}{2}nf^2$
1	forward- and backward substitution	$2c_\emptyset$
	creating the derivative modes	
d	preparation of Ritz vectors, d matrix vector products $Y_i = \left[\frac{\partial 1}{\partial \lambda_i} X\right] x$	$\overline{n}\overline{l_\emptyset}$
d	generation of Ritz vectors, forward-backward substitution $\phi_i = X^{-1} Y_i$	$2c_\emptyset$
	generation and projection into the embedding solution space $\mathcal{D}$	$s = d + 1$
1	generation of the solution manifold $\mathcal{M}$ embedding solution space $\mathcal{S}$, vector assembly and ortho-normalization $\Phi = ortho([x, \phi_1, \phi_2, \ldots, \phi_d])$ see Eq. (7.1)	$n(s^2 + s)$
d	transformation of the design part matrices from discretization space $\mathcal{S}$ to the embedding solution space $\mathcal{D}$, triple matrix product $\overline{X}_j = \Phi^t [\overline{X}_j] \Phi$	$s\overline{n}(s + \overline{l_\emptyset})$
1	transformation of the general stiffness matrix from discretization space $\mathcal{S}$ to the embedding solution space $\mathcal{D}$, triple matrix product $\overline{X} = \Phi^t X \Phi$ assuming **density**$(K) \approx \frac{n\overline{l_\emptyset}}{n^2}$	$sn(s + \overline{l_\emptyset})$
$\sum$		$\approx \frac{1}{2}nf^2 + 2c_\emptyset s + (s-1)\overline{n}\overline{l_\emptyset} + n(s^2+s) + (s-1)s\overline{n}(s+\overline{l_\emptyset}) + sn(s+\overline{l_\emptyset}))$

7.4.2.1 Non-sensitivity based solution

Based on the above projection scheme from the discretization space into the response space we have to solve the equation in a $s-$ and not in $n-$dimensional space. As a disadvantage the matrices are not sparse any more and therefore the front-size f will be s. Also the number of non-zeroes becomes $c_\emptyset = \frac{1}{2}s^2$. For that reason also forming of the resultant (dynamic) stiffness matrix $\overline{X}$ has to be considered, as it is now not a minor task compared to the DCMP and the FBS. (The α_i denote the pre-factors based on the new parameter setting).

	Operation	FLOPS
s	forming new stiffness matrix $\overline{X} = \sum_i^s \alpha_i \overline{X}_i$	s^2
1	decomposition of new stiffness matrix	$\frac{1}{2}s^3$
1	forward- and backward substitution	s^2
	$\sum$	$\frac{3}{2}s^3 + s^2$

7.4.2.2 Gradient based solution

The accounting of the flops for a gradient based ARM solution works in full analogy to 7.4.2.1 and 7.4.1.2.

	Operation	FLOPS
s	forming new stiffness matrix $\overline{X} = \sum_i^s \alpha_i \overline{X}_i$	s^2
1	decomposition of stiffness matrix	$\frac{1}{2}s^3$
1	forward- and backward substitution	s^2
d	matrix vector product $[\frac{\partial 1}{\partial \lambda_i} X]\overline{x}$	s^2
d	forward- and backward substitution	s^2
	$\sum$	$\frac{7}{2}s^3 - s^2$

7.4.3 Discussion and comparison of numerical efficiency

Based on the results derived above the ARM method can now be compared with the traditional way. Depending on the dimension s of $\mathcal{D}$ the calculation of the solution might be much faster then in the original $\mathcal{S}$-space. Nevertheless the effort for generating $\mathcal{D}$ has to be invested. Therefore it is a question of how many reanalysis solutions will be carried out in order to make the calculation in the $\mathcal{D}$-space more efficient then in the (discretization) $\mathcal{S}$-space. Figure 7.13 shows this principle. Due to the ortho-normalization the dimension of $\mathcal{D}$ is always lower or equal then the dimension of $\mathcal{S}$. However, a function evaluation in $\mathcal{D}$, based on full matrices, can take longer than one in $\mathcal{S}$, which benefits from sparse matrices. This would mean that there could be no *break even point* for efficiency considerations. Practically the solution in $\mathcal{D}$ is faster then in $\mathcal{S}$, due to the dramatically reduced dimension. If we have a break even point it is given by:

$$\sigma := \quad \text{number of re-analyses} \tag{7.63}$$

$$FLOPS_{\text{total}-\mathcal{S}} = \sigma \cdot FLOPS_{\text{solve}-\mathcal{S}} \quad \text{flops for the general method} \tag{7.64}$$

$$FLOPS_{\text{total}-\mathcal{D}} = FLOPS_{\text{generate}-\mathcal{D}} + \sigma \cdot FLOPS_{\text{solve}-\mathcal{D}} \quad \text{flops for the ARM method} \tag{7.65}$$

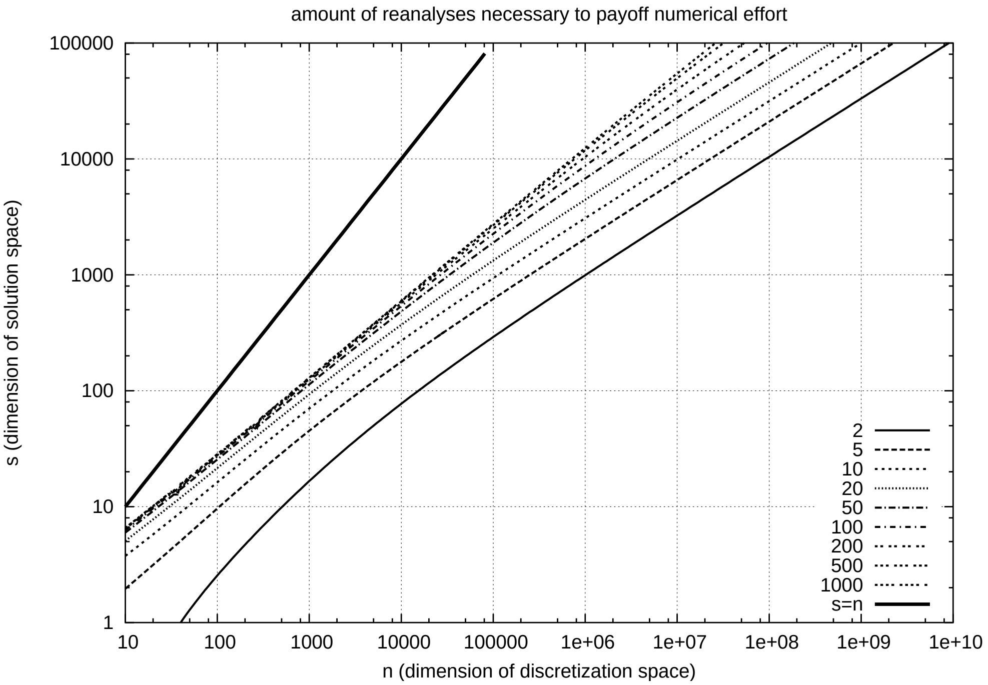


Figure 7.13: The amount of reanalysis necessary to make the ARM method more efficient according to the number of floating point operations then the generic approach. The underlying structural model is the 2D (QUAD) model discussed before in this chapter.

The total amount of FLOPS must be now equalized to obtain the number of structural analyses σ where both methods take the same amount of time:

$$FLOPS_{\text{total}-\mathcal{D}} = FLOPS_{\text{total}-\mathcal{S}} \tag{7.66}$$

$$\longleftrightarrow \sigma = \frac{FLOPS_{\text{generate}-\mathcal{D}}}{FLOPS_{\text{total}-\mathcal{S}} - FLOPS_{\text{total}-\mathcal{D}}} \tag{7.67}$$

$$\tag{7.68}$$

If the total number of analyses, for example in an optimization project, multi-parametric study, robust design analysis or other project, is larger then σ then the ARM-method will be less time consuming. If the planned number of function evaluations is lower then σ then the ARM method is less efficient according to the total number of floating point operations. If there is a need to have a very fast reanalysis, for example for quick design decisions, it might be worth to generate the $\mathcal{D}$-space anyway.

Figure 7.13 shows the amount of function evaluations, at which the ARM method pays off according to the total number of floating point operations depending on n and s.

7.4.3.1 Megaflop examples and computing time

To give some estimates for computing time we estimate the number of FLOPS for sparse and full models according to the derived scheme. As the sparse model we take the QUAD model from table 7.1 with 1315840 degrees of freedom. The FLOPS for the full models (for the ARM approach) for sizes of $10, 100, 1000$ and 10000 are taken from the derivations in this chapter.

Model size	Estimated Gigaflops for one solution (FBS+DCMP)
1315840 sparse	5583.3
10 full	1.6e-6
100 full	0.0015
1000 full	1.50
10000 full	1500.1

Assume we have a 1 Gigaflop computer, which means that this computer is able to perform 10^9 floating point operations per second. Then we need at least the same amount of seconds as we have Gigaflops to perform the described operations. A 1 Gigaflop computer can be either a machine with a 1 Gigahertz CPU and 1 floating point operation per cycle or a 500 Megahertz CPU with 2 floating point operations per cycle.

However, as mentioned in beginning of this chapter our estimates are lower estimates concerning computing times. In practice there will be significant differences due to not considered hardware and software features such as memory hierarchies and basic instruction code differences.

Nevertheless the above example shows that we could perform 3722 full solutions with a reduced basis of size 1000 instead of one sparse solution with 1315840 degrees of freedom.

This demonstrates the capabilities of the proposed approach.

8 Applications to structural design problems

8.1 Uni parametric study: Pareto optimal stiffness of engine hood

The target of this section is to show the convergence radius of a real world example with a one dimensional parameter space. The second aspect will be the comparison with a traditional straight forward *response surface method*.

Concerning the observed model the influence of glue stiffness (see Fig. 8.1) on the global static stiffness of a car engine hood is examined. While increasing the glue stiffness the weight of the interior shell will be reduced with the constraint, that certain static stiffness values do not fall beyond their original settings.

This is carried out in two ways. First the ARM method is used for an efficient Pareto optimization (see also Das [18] and Pietrzak [108]). During increasing the glue stiffness the sheet thickness of the interior sheet will be decreased so far that the initial stiffness values are maintained. As a comparison in a second step the stiffness is settled at a certain value and the weight is reduced using topology optimization of the interior sheet. As one result the potential of a sizing and a topology optimization are compared.

8.1.1 Model and loading conditions

The following load-cases where considered:

- cross member stiffness (load-case- 1)
- longitudinal member stiffness (load-case 2)
- corner-bending stiffness (load-case 3)

The corresponding finite element model consists of 115506 unconstrained *displacement degrees of freedom* (DOF) and can be seen in Fig. 8.1.

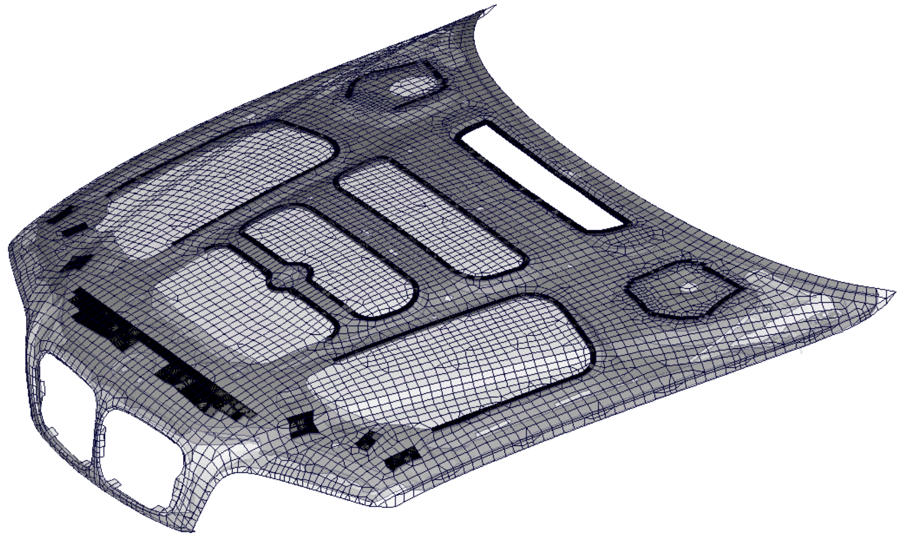


Figure 8.1: Model of engine hood with 115506 displacement degrees of freedom. The glue connecting the inner sheet (gray) with the outer sheet (transparent, mesh style) is displayed black.

8.1.2 Parametric task

The following procedures were carried out:

- Show the dependency of the stiffnesses of the 3 load-cases from the shear modulus of the material (Fig. 8.1), which connects the inner and the outer sheet, in the range [0.1-70,000 $\frac{N}{mm^2}$]. This shall show the stiffness potential due to different glue mixtures and even due to a welding. The baseline material was a composition rubber based glue with shear modulus of 4 $\frac{N}{mm^2}$, e.g. *Totalseal* from *Le Joint Francais.*
- Conversion of the stiffness gain at a shear modulus of G= 800 $\frac{N}{mm^2}$ (using epoxy based glue, e.g. *Betamate 1496* from *DOW Automotive AG*) into a weight potential by
  - decreasing the thickness t of the interior sheet.
  - removing material from the interior sheet using topology optimization.

8.1.3 Procedure

The governing equation in this case is:

$$(\boldsymbol{K}_r + \boldsymbol{K}_g(G) + \boldsymbol{K}_i(t))\,\boldsymbol{x} = \boldsymbol{F} \tag{8.1}$$

where $\boldsymbol{K}_g(G)$ is the stiffness matrix of the glue and $\frac{\partial}{\partial \lambda_1}\boldsymbol{K}_g(G) = \frac{1}{G}\boldsymbol{K}_g$. Parameter $\lambda_1 = G$ represents the shear modulus of the glue. $\boldsymbol{K}_i(t)$ is the stiffness matrix of the inner shell and shall be composed of

$$\boldsymbol{K}_i(t) = t\boldsymbol{K}_i^m + t^3\boldsymbol{K}_i^b \tag{8.2}$$

neglecting transverse shear effects (m stands for *membrane* and b for *bending* stiffness). $\boldsymbol{K}_r$ represents the rest of the structure.

Using the approach for a one dimensional parametric problem the ortho-normalized $\{\boldsymbol{x}, \boldsymbol{A}_0\boldsymbol{x}, \boldsymbol{A}_0^2\boldsymbol{x}, \ldots\}$ with $\boldsymbol{A}_0 = \boldsymbol{K}^{-1}\boldsymbol{K}_g$ represent the global shape functions and form the basis for the new solution space. Due to the fact that the glue stiffness undergoes relative changes of over 5 orders of magnitude additional residual modes were inserted. These are formed by the displacement vectors at the lower and upper range of the glue stiffness, respectively $\boldsymbol{x}_l$ and $\boldsymbol{x}_u$. This takes care that the results at the upper and lower bounds can be predicted *exactly*. Additionally Taylor series terms concerning the changes in the interior shell $\boldsymbol{A}_1 = \boldsymbol{K}^{-1}\boldsymbol{K}_i^b$ and $\boldsymbol{A}_2 = \boldsymbol{K}^{-1}\boldsymbol{K}_i^m$ were inserted. The resulting global shape functions therefore are:

$$\boldsymbol{\Phi} = \mathbf{orthog}([\boldsymbol{x}, \boldsymbol{x}_l, \boldsymbol{x}_u, \boldsymbol{A}_1\boldsymbol{x}, \boldsymbol{A}_2\boldsymbol{x}, \boldsymbol{A}_0\boldsymbol{x}, \boldsymbol{A}_0^2\boldsymbol{x}, \ldots]) \tag{8.3}$$

8.1.4 The solution manifold

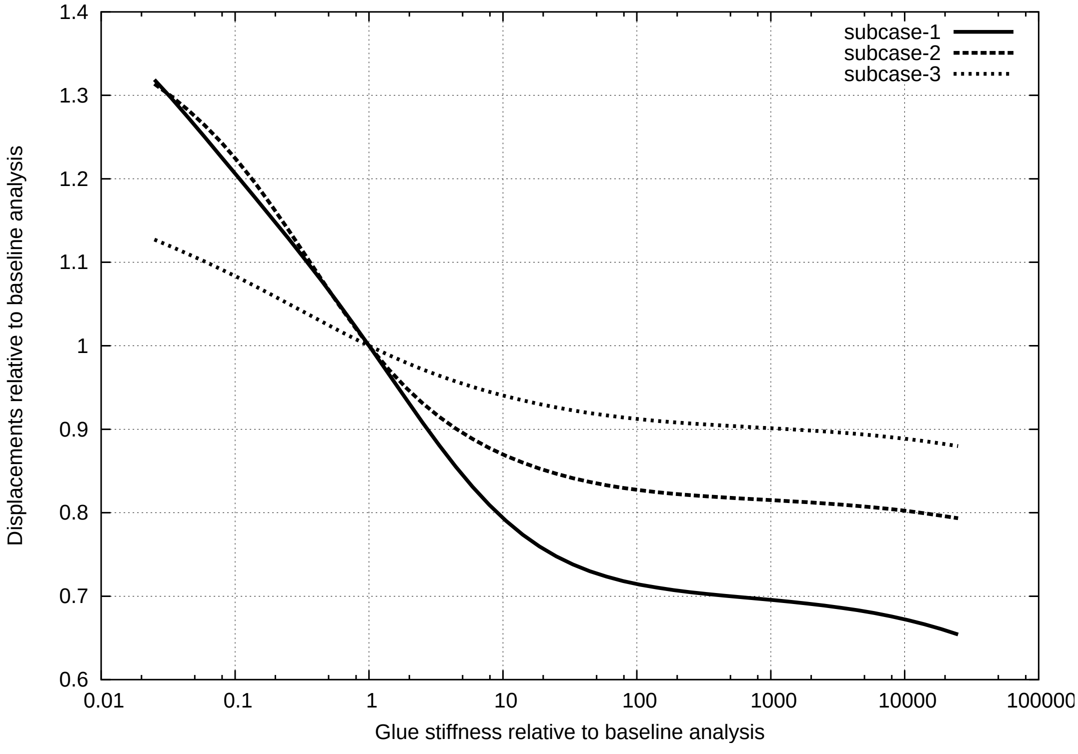


Figure 8.2: 1-dimensional solution manifold of the glue stiffness.

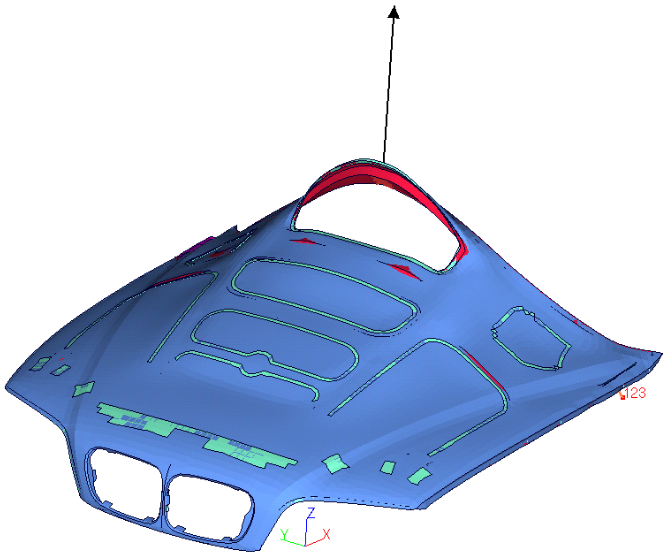


Figure 8.3: Displacement plot of load-case-1

The graphs of the three 1-dimensional solution manifolds are shown in Fig. 8.2. Figure 8.3, Fig. 8.4, and Fig. 8.5 show the corresponding displacement plots.

8.1.5 Comparison of *approximate response manifold* with *response surface method*

Due to the similar behavior of the stiffness of all load-cases the focus is on the first subcase for the comparison of the *exact* solution and the solution given by the *approximate* models. The comparison focuses on *local* and *global* behavior.

8.1.5.1 The *approximate response manifold*

For the local behavior of the one-dimensional manifold the derivative modes were used, developed for this case in chapter 6.2. In Fig. 8.6 the behavior using the global shape functions

$$\Phi = \mathbf{orthog}([\boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}^2\boldsymbol{x}, \ldots, \boldsymbol{A}^j\boldsymbol{x}]) \tag{8.4}$$

can be seen for different j's. It can be clearly seen in Fig. 8.6 that in the range of changing the stiffness by a factor 3 up or down an excellent matching with the *exact* results can be obtained using only one derivative mode. This means that the Taylor series of derivative mode generation can be stopped after the first differentiation.

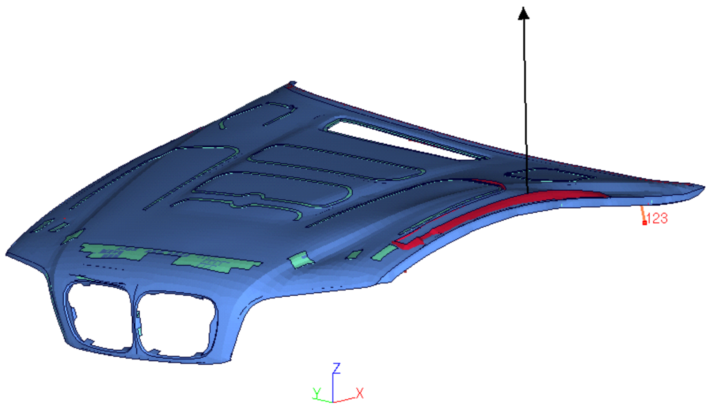


Figure 8.4: Displacement plot of load-case-2

It can be also seen that *without* any derivative mode, using the baseline analysis displacement vector only, a reasonable result can be obtained. This benign behavior is gained, in spite of the fact that the manifold encapsulating space $\mathcal{D}$ is only one dimensional, because the ARM-method uses the same *underlying physical equation* as the original analysis. Figure 8.7 shows exactly the same choice of global shape functions but in a global behavior. It is observed that the derivative modes can predict accurate results for changes in the parameters up to a factor 3. For larger changes residual modes have to be considered. These results can be seen in Fig. 8.8. The *lower* residual mode $\boldsymbol{x}_l$ takes care of convergence at the lower end while the additionally added *upper* one, $\boldsymbol{x}_u$, fixes the *approximated response manifold* to the upper boundary. The usage of only the baseline mode and the two residuals may be suggested when viewing this figure. Nevertheless Fig. 8.9 shows that their local behavior is inferior compared to the derivative modes. Adding only the first derivative mode, the 4-dimensional manifold encapsulating space $\mathcal{D}$ predicts very good local and global behavior for parametric changes up to five orders of magnitude.

8.1.5.2 The *response surface method*

As this thesis is about a new surrogate model developed for parametric analysis it will be now compared to one of the most commonly used methods used in this area. There are multiple approaches for the *response surface method*. A deep overview is given in the book of Myers and Montgomery [95]. However there are two basic facts that influence the surrogate model of the *response surface method*:

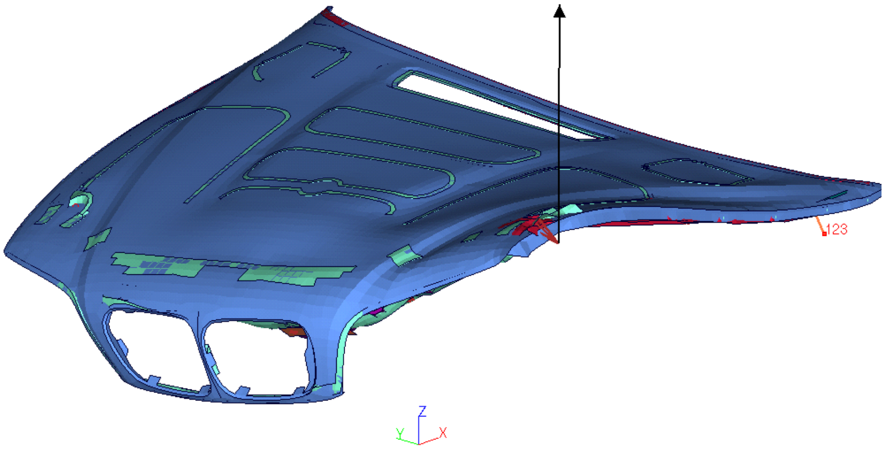


Figure 8.5: Displacement plot of load-case-3

- the underlying mathematical model
- the distribution of the data samples for fitting the model
- fitting of the free parameters

When knowing nothing about the underlying model the use of a *second order Taylor series* model is the standard choice. Higher ordered *Taylor series models* for multi parameters are mostly useless for practical applications due to their *insatiable* need for fitting data. (Needed complexity is $O(d^k)$, for d dimensions and k-th order Taylor series).

To construct the response surface model several attempts are made:

- the parameters are chosen in a way that the result for the baseline parameter setting is exact together with the first and second derivative: $RSM(a, b, c; [1, f'(1), f''(1)]) = a + bx + cx^2$
- the parameters are chosen in a way that the result at the baseline setting and at the minimum and maximum parameter setting is exact: $RSM(a, b, c; [min(x), f(1), max(x)]) = a + bx + cx^2$
- as seen in the Fig. 8.10 the above approaches are of limited prediction value. The fitting of the curve is now done using all previously calculated data points: $RSM(a, b, c; allpoints) = a + bx + cx^2$
- as the results for the large range Fig. 8.11 was still not meaningful the curve will be fit now in the logarithmic scale to selected data points. $RSM(a, b, c; [min(x), f(1), max(x)]) = a + b\log(x) + c\log(x)^2$

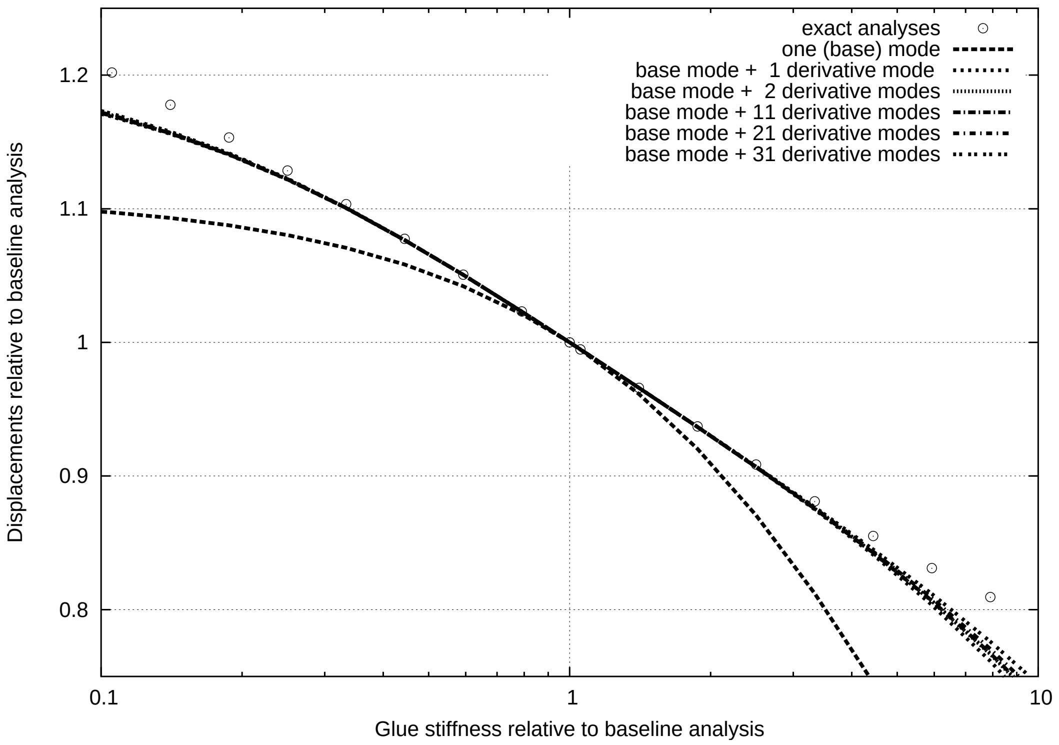


Figure 8.6: ARM based local approximation using derivative modes

- the same underlying model is fit now with all available data points: $RSM(a, b, c; allpoints) = a + b\log(x) + c\log(x)^2$

For fitting the data the general approach was used:

$$\sum_{j=1}^{N}(y_j - RSM(a, b, c, x_j))^2 \overset{!}{=} \text{Minimum} \quad \text{where the } j \text{ represent the data points} \tag{8.5}$$

As seen in the Fig. 8.10 and Fig. 8.11 additional information had to be put into the surrogate model of *response surface* to get reasonable predictions. However the best global results were achieved with the logarithmic scale of the abscissa parameters. This was necessary to the previously chosen spacing of the data points. This leads to the impression that arbitrary distributed data points can make the response surface model useless depending on the fitting algorithm and on the weighting of the different data points during the fitting algorithm. Due to this reason the *design of experiment*, a method describing the sampling of data points, occupies much space in the standard literature of the *response surface method*. See Myers and Montogmery in [95].

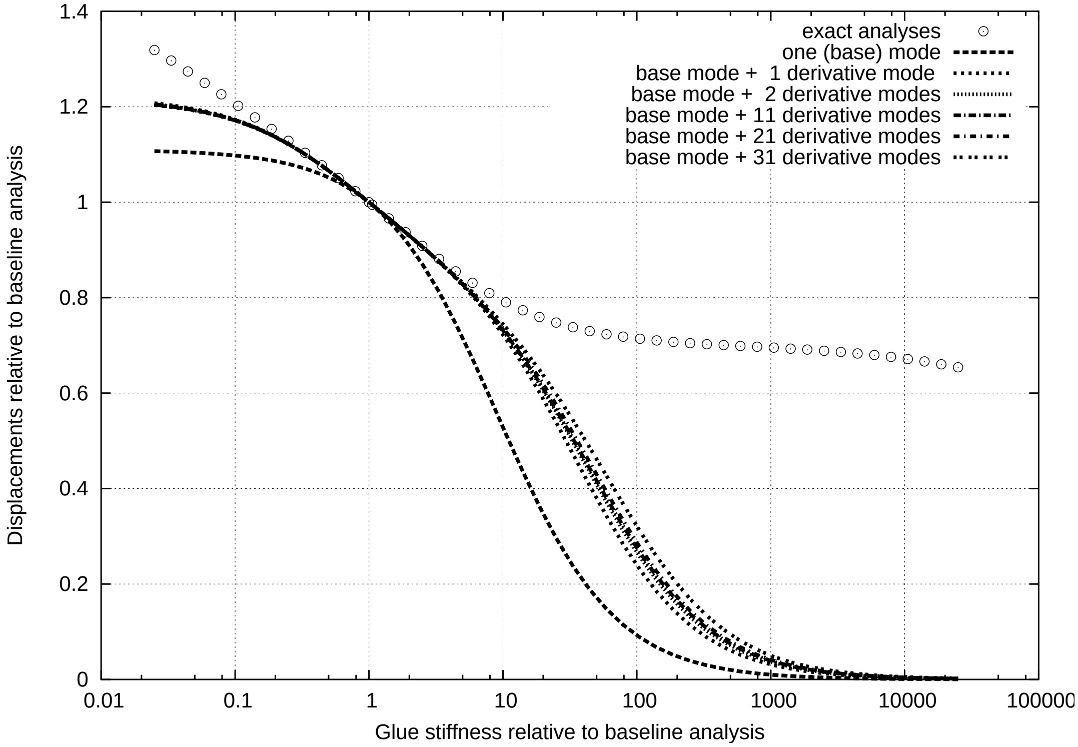


Figure 8.7: ARM based global approximation using derivative modes

8.1.5.3 *Approximate response manifold* versus *response surface method*

Both methods require additional knowledge of the system. For the ARM method it is at least necessary to define the amount and number of derivative modes and the use of residual modes. The current example showed for the ARM method very good local approximation with one derivative mode for relative changes up to factor 3. For sufficient approximation of larger changes and a global approximation, residual modes had to be picked. However the calculation of residual modes is numerically expensive compared to the generation of derivative modes. The amount of desired/needed residual modes can significantly increase with more then one parameter.

The RSM needs more user input to provide reasonable results. The choice of the underlying mathematical model, the sampling of the data points, and the provided fitting criteria determined entirely the usability of the surrogate model. According to numerical effort the least expensive RSM (amount of data points equals amount of free RSM-parameters) is about as cheap as an ARM approach. However the ARM model is in this case more accurate then the RSM model. This can be seen in the Fig. 8.12 where the local approximation is displayed. For both approaches the most accurate is displayed. The Fig. 8.13 shows the accuracy of the global approximation. Only the ARM method could simultaneously approximate local and global behavior.

An alternative mathematical model for the response surface could be chosen. It would be obvious to chose a mathematical model which has the same parametric behavior as the underlying physical model. Obviously the *best* mathematical model would be the *correct* physical equation.

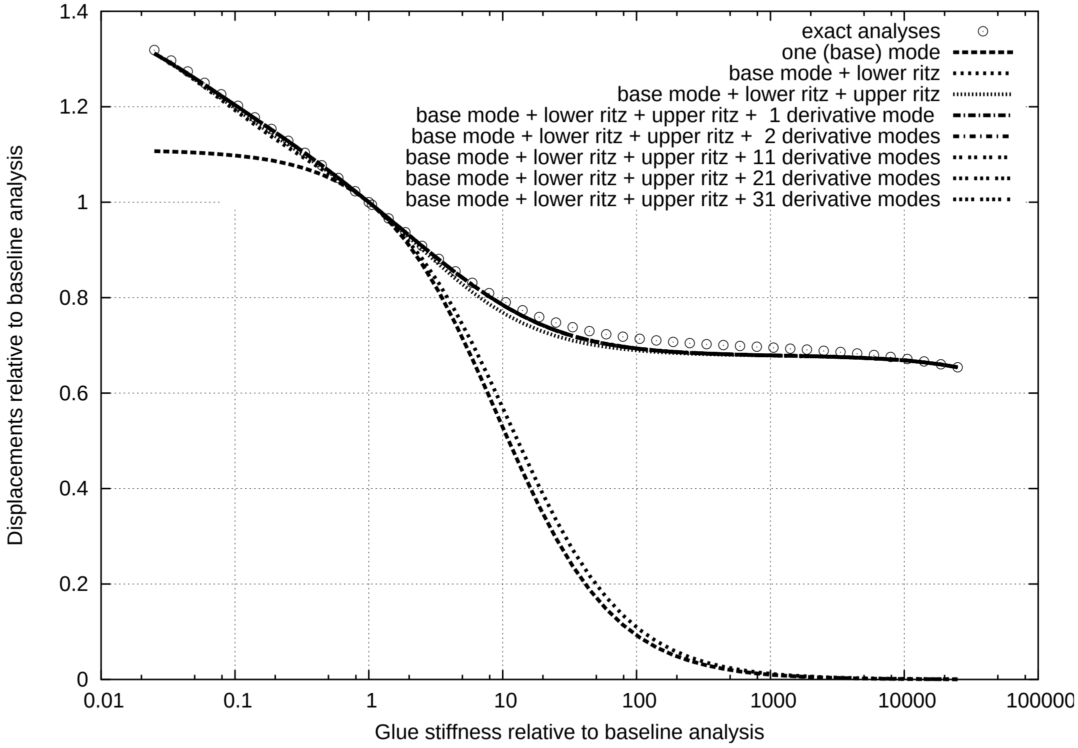


Figure 8.8: ARM based global approximation using residual and derivative modes

After calculating the data samples and fitting the model parameters one would more or less end up with a model similar to the ARM method, but without using its straight forward way of generation.

In summary it can be stated that it makes not much sense to use a *response surface method* instead of the demostrated *approximate response manifold*. From the engineering point of view the best *equation* for approximating an solution is the *correct* physical equation, as used in the ARM, despite to a surrogate mathematical replacement model which is used by the RSM method. As long as the ARM method has, as shown for the underlying examples, a performance advantage it should be the preferred approach. The RSM is more suited to approximate engineering problems to which the ARM is not or cannot be expanded or to fit experimental results.

8.1.6 Corresponding weight potential of the interior sheet using Pareto optimization

In addition to the derivative modes of the glue, derivative modes of the interior sheet concerning it's thickness were calculated. This allows parametric changes in the glue stiffness as well as in the thickness of the interior sheet. While increasing the glue stiffness a globally stiffer model for the three load-cases is gained. While there is no need for additional stiffness, the weight can be reduced by decreasing the thickness of the interior sheet simultaneously in a way that the

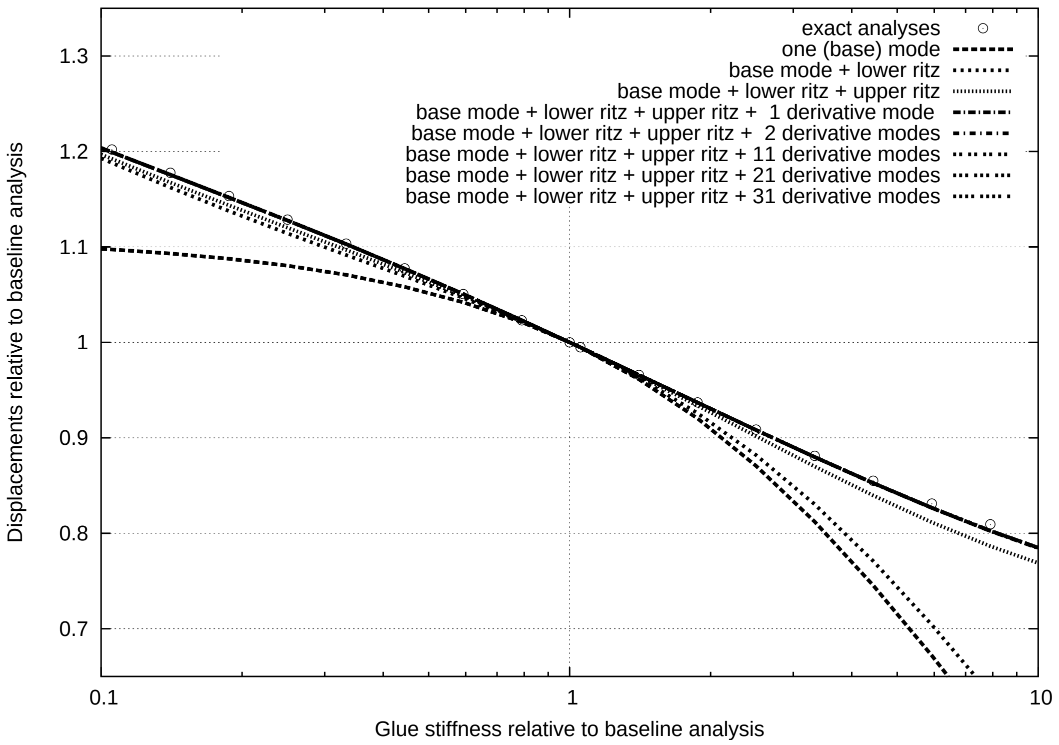


Figure 8.9: ARM based local approximation using residual and derivative modes

global stiffness does not fall below the original values. This results in a certain weight gain per added glue stiffness. This trade off curve between glue stiffness and interior sheet thickness was carried out using the ARM method. The first run with 1000 analyses scanned the parameter region and determined a feasible and infeasible area for pareto optimality. Due to the fact that the manifold encapsulating space $\mathcal{D}$ was only of dimension 7 these analyses could be carried out in a few seconds. After processing the results of this first run and creating the feasible and infeasible pareto front another sample of 1000 shots between the two fronts was created in order to illuminate the real boundary between feasibility and infeasiblity. This was repeated two more times. The pareto curve in Fig. 8.14 shows us that the initial thickness of 0.9mm at a shear modulus of 4 $\frac{N}{mm^2}$ can be dropped down to 0.73mm at a shear modulus of 800 $\frac{N}{mm^2}$ without loosing stiffness. This represents a weight reduction of 530g.

The total run time for all 4000 data samples was below 3 minutes on a low end workstation (Intel(R) Xeon(TM) CPU 2.40GHz). The initial full analysis in $\mathcal{N}$-space took 1 minute 39 seconds. The generation of the subspace $\mathcal{D}$ took 3 minutes on a HP-PA-RISC compute server running at 750Mhz. So after the second analysis the break even point was reached. The disagreement with the prediction of Fig. 7.13 indicating less then 2 analyses would be needed to reach the break even point is explained at the beginning of chapter 7.1.

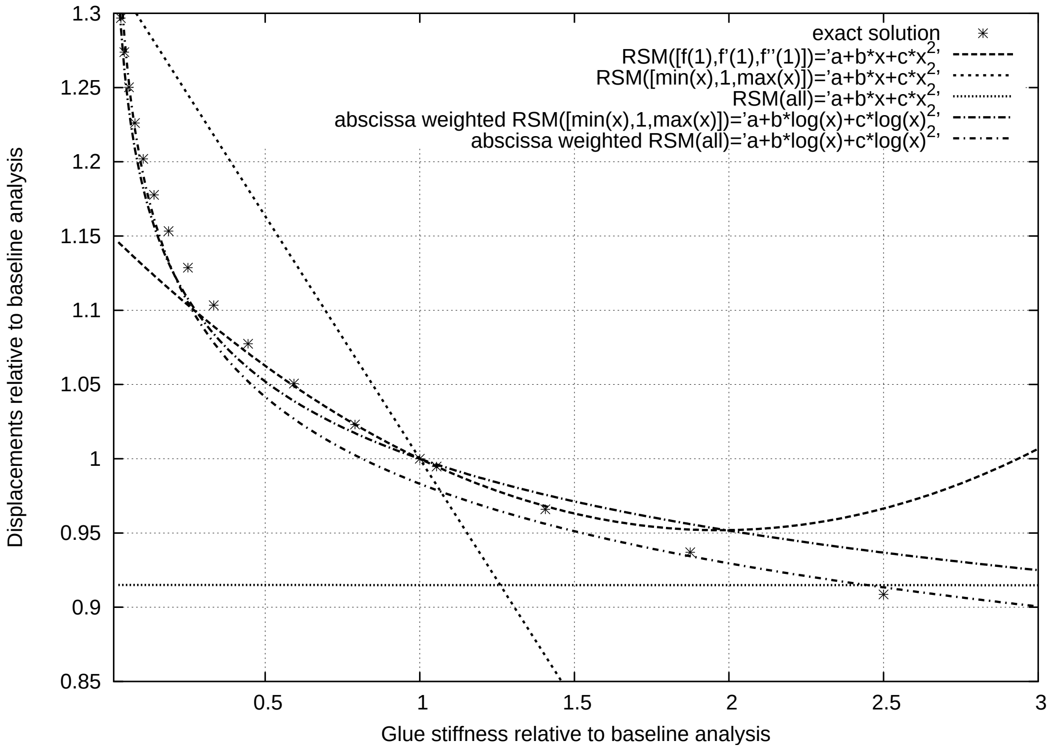


Figure 8.10: RSM based local approximation

8.1.7 Corresponding weight potential of the interior sheet using topology optimization

Based on these results topology optimization can be used to trim the interior sheet by maintaining the global stiffnesses using an epoxy based glue (e.g. *Betamate 1496* from *DOW Automotive AG*) with a shear modulus of $G = 800\frac{N}{mm^2}$ instead of a composition rubber based glue (e.g. *Totalseal* from *Le Joint Francais*) with a shear modulus of $4\frac{N}{mm^2}$.

The Fig. 8.15 shows the deleted elements. The dark elements show the relevant load-path, while the light gray elements can be deleted. The medium gray elements are in an *inter-medium stage*. They cannot be deleted by the topology optimizer with the current discretization. Neglecting the light gray elements leads to a weight reduction from 2.8kg to 1.8kg and helps therefore to save 1kg of material.

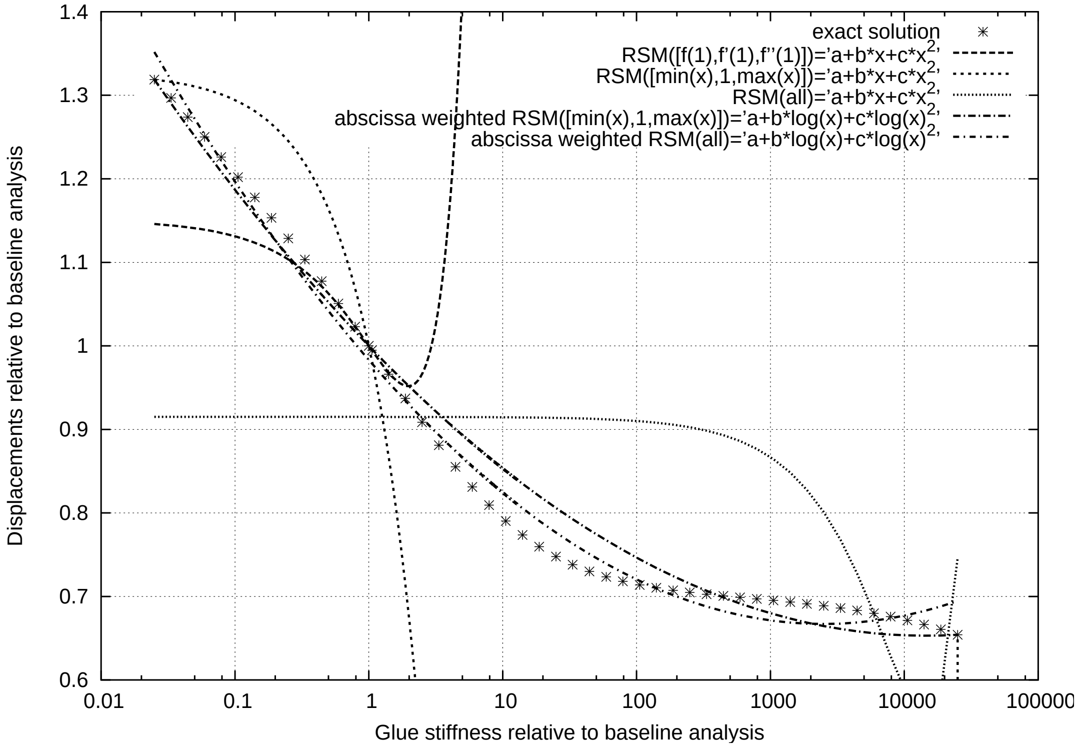


Figure 8.11: RSM based global approximation

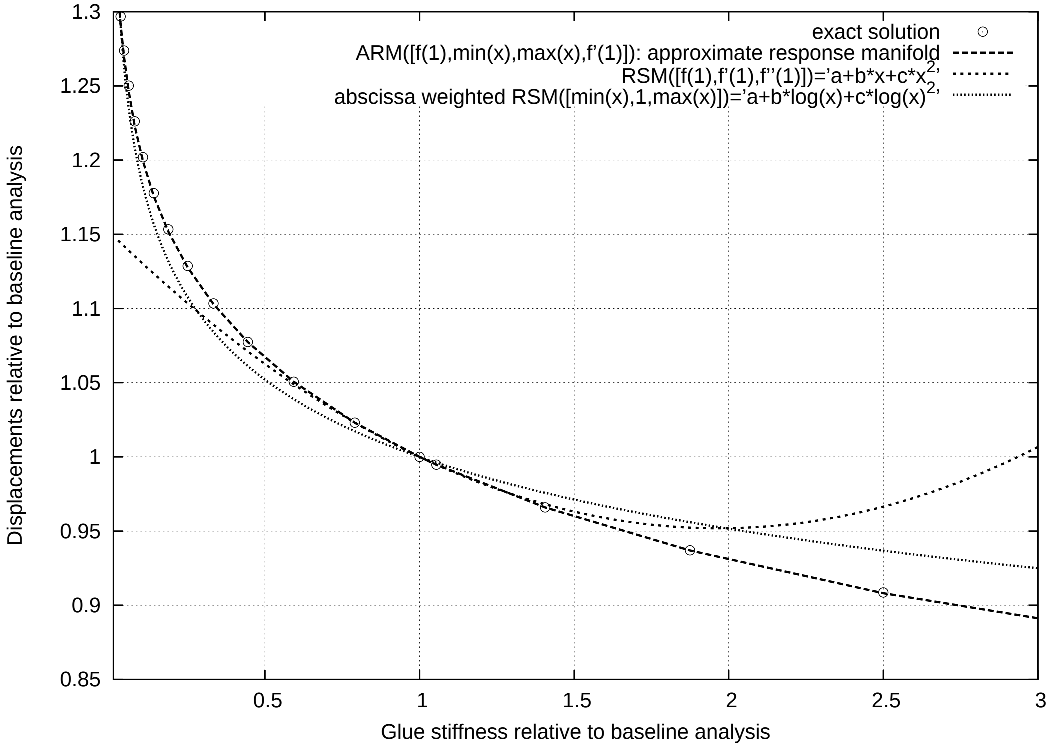


Figure 8.12: ARM/RSM comparison for local approximation

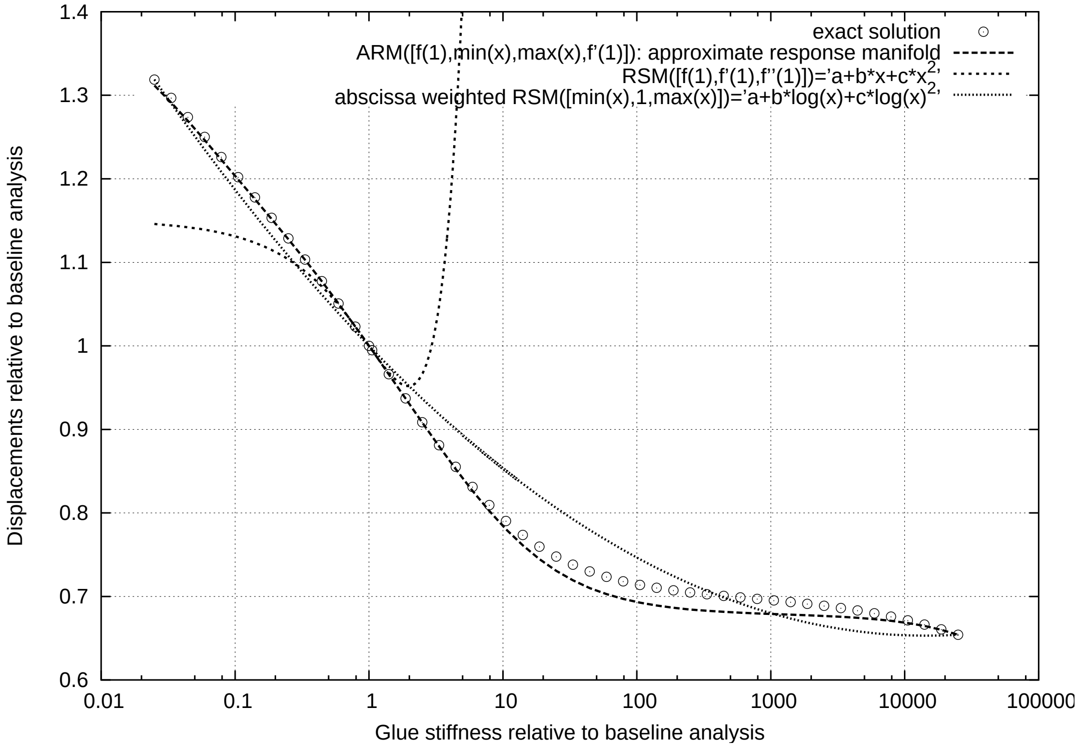


Figure 8.13: ARM/RSM comparison for global approximation

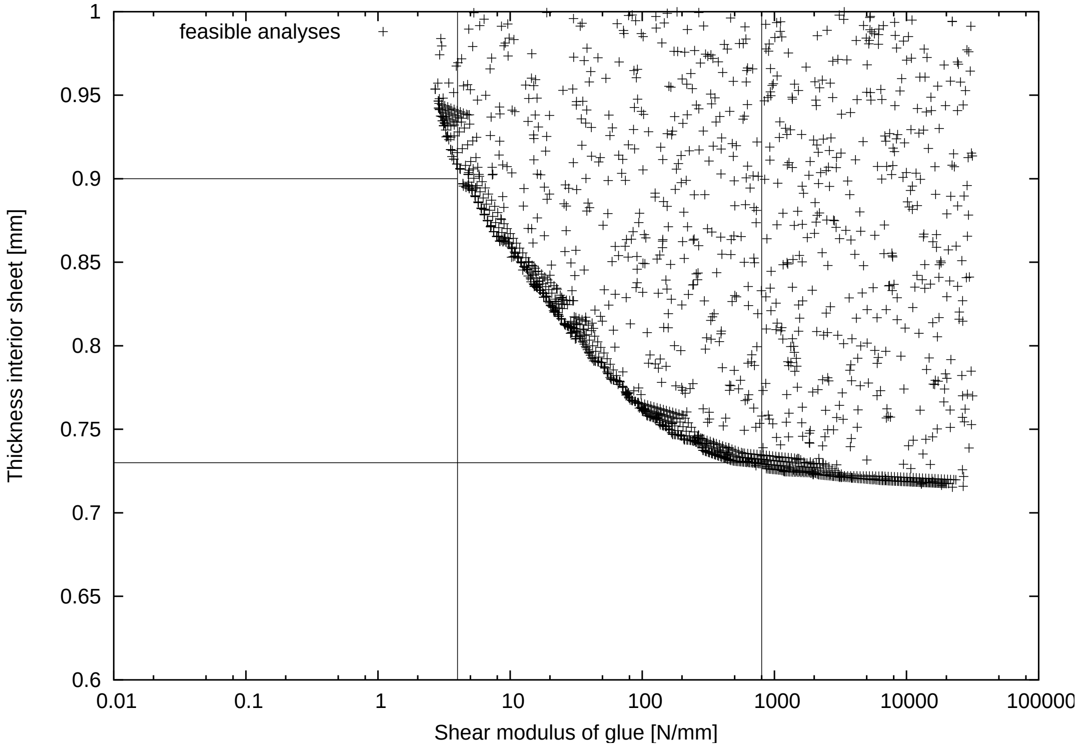


Figure 8.14: Pareto for weight reduction. Only feasible solutions displayed. 4000 runs in total.

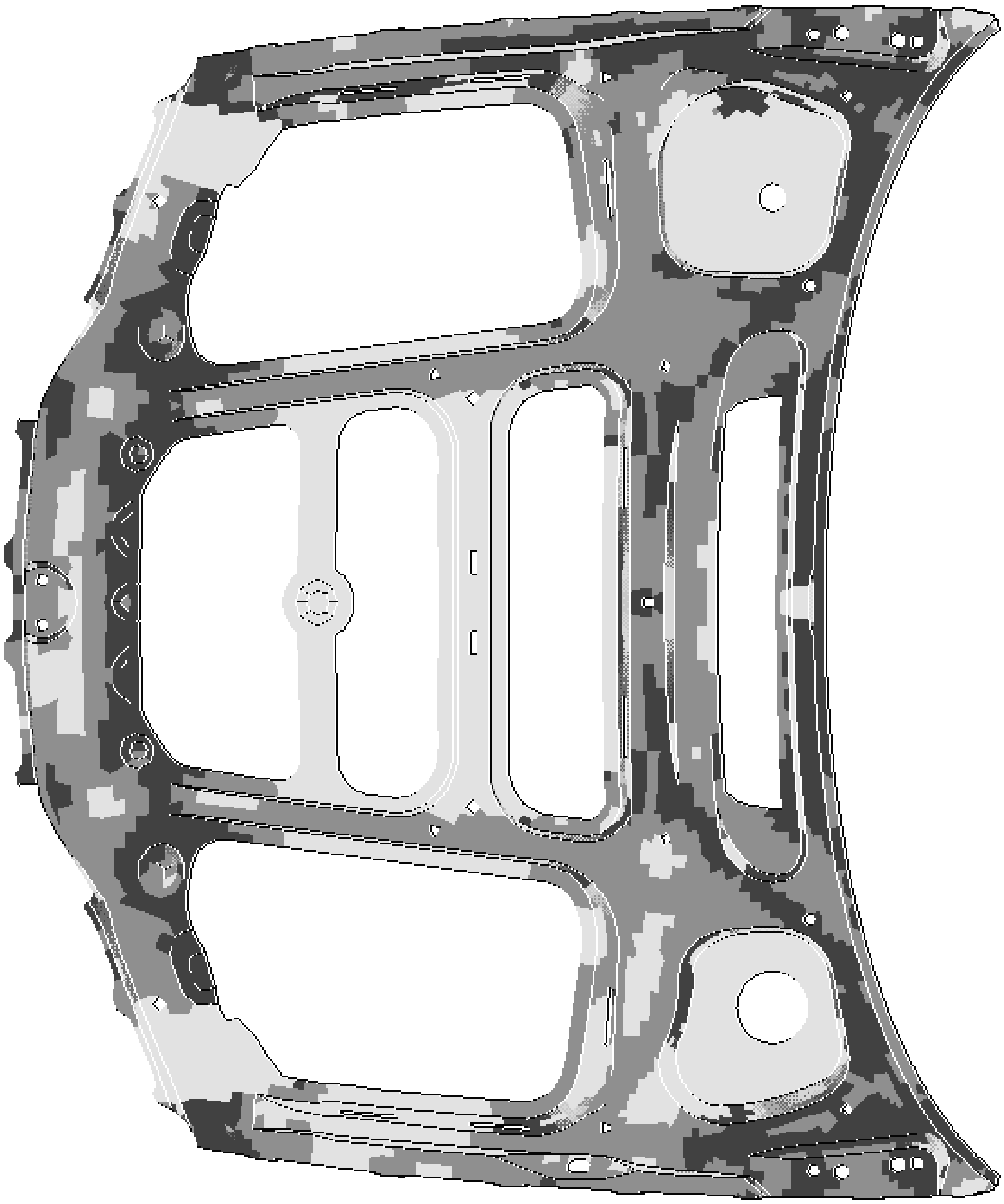
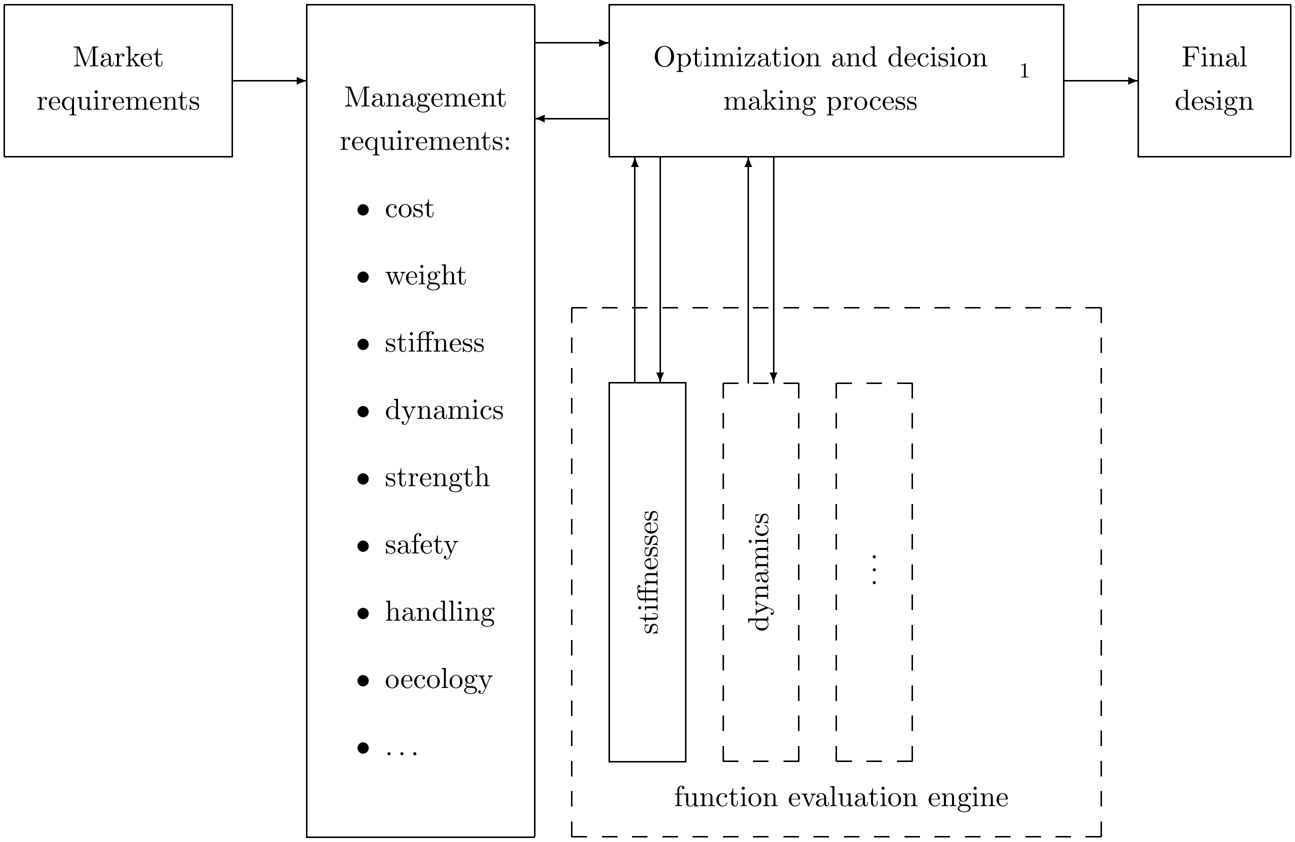


Figure 8.15: Last cycle of topology optimization concerning the inner sheet. The elements indicated for removal are displayed light gray. Dark gray areas show main load paths.

8.2 Multi dimensional parametric study: static stiffnesses of a complete car body

The following model represents a part of a typical multidisciplinary optimization problem in the automotive industry (see also HEISERER and BAIER [38]). The following scheme outlines a decision making process and the involved calculation engines concerning the functional requirements for a body in white.



In realistic problems there exist not only a few but many variables which influence the design. In this example the focus will be on a larger model with a corresponding high number of design variables. The goal is to show the accuracy and efficency of the approach for these kind of problems and to demonstrate the ability to serve as a high speed calculation and reanalysis engine.

8.2.1 Model and loading conditions

The underlying finite element model consists of 232272 displacement degrees of freedom (DOF) representing a full body in white. In order to show the load independent accuracy of the method 5 typical different load cases will be examined:

- 2 torsional load cases

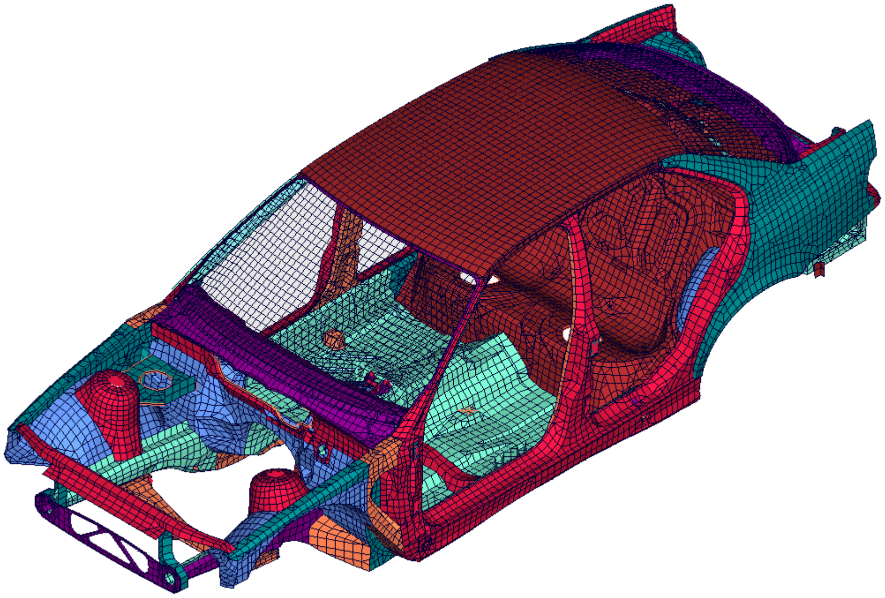


Figure 8.16: Example Model of a body in white

- 3 bending load cases

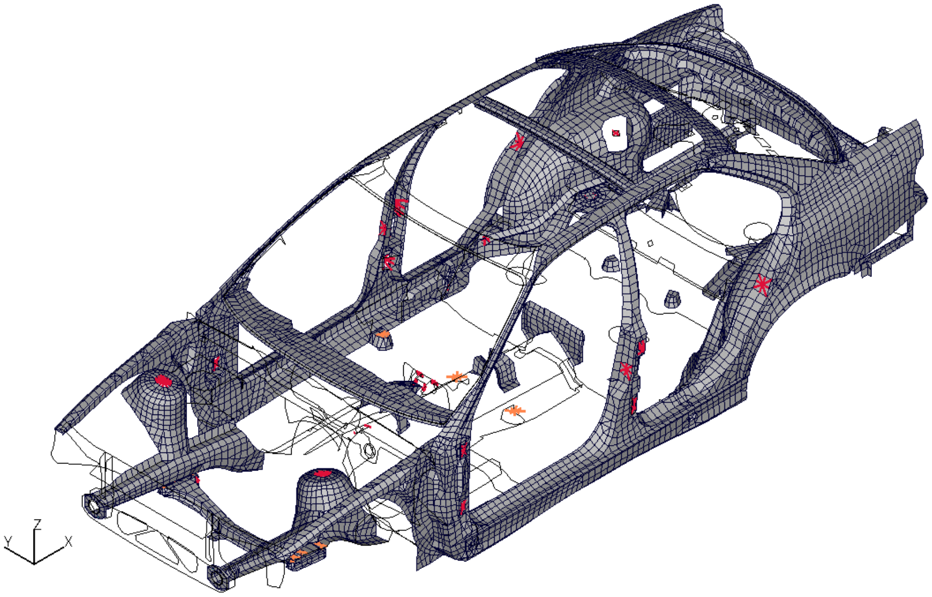


Figure 8.17: Example design-1: Linking of several design variables (thicknesses of parts) into one for parametric studies.

The design model consisted of 106 design variables, the thicknesses of 103 shell parts and the cross section diameters of 3 beam parts. To ensure that the comparison of the ARM method concerning accuracy does not neglect certain critical design variables all design variables were considered.

Also, the evaluation of multiple load cases and different output degrees of freedom is necessary, see Fig. 8.18. To avoid unnecessary amount of data, three new design parameters were introduced which lead to three parametric design studies. For each new design parameter a certain number of design variables, in this case shell thicknesses of certain parts, were linked together. Each of the three new parameters *design-1*, *design-2* and *design-3* contained approximately 50% of the original DOF and also 50% of the area of all designed parts (see Fig. 8.17). The selection of these parts and also their contribution factor to the new parameters (design-i) where chosen purely randomly with the only boundary condition of being *close* to the mentioned 50%. The reason for combining each time approximately the half of the model into one variable is that derivative modes of variables *wetting* $\approx$ 50% of the DOF have mostly the worst convergence ratio. So the examples shown here represent by statistical choice an upper limit of the approximation error.

Linked parts/variables for design studies:

	% of total parts	% of total area
design-1	47.170	47.657
design-2	49.686	56.082
design-3	44.025	53.365

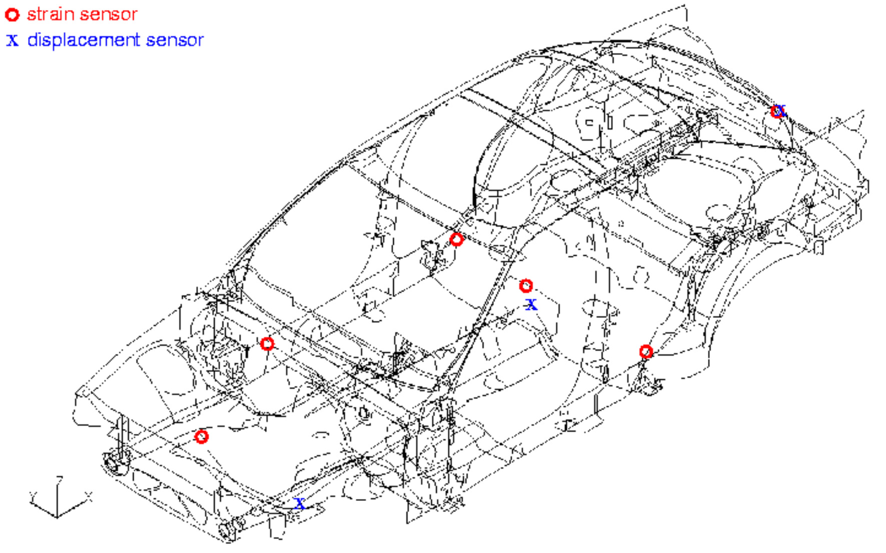


Figure 8.18: Example of sensors for stiffness and strain measurement

8.2.2 ARM specific formulation

The system can be described with:

$$\boldsymbol{K}\boldsymbol{x} = \boldsymbol{F} \tag{8.6}$$

$$\tag{8.7}$$

The stiffness matrices K_i for shell elements (without transverse shear) are divided into the following stiffness components:

- membrane stiffness component $\boldsymbol{K}_i^m$
- bending stiffness component $\boldsymbol{K}_i^b$

Also the stiffness matrix for beam elements (without transverse shear) is divided into the

- area part $\boldsymbol{K}_i^A$
- area moment of inertia for bending in plane 1 $\boldsymbol{K}_i^{I_1}$
- area moment of inertia for bending in plane 2 $\boldsymbol{K}_i^{I_2}$
- area product of inertia $\boldsymbol{K}_i^{I_{12}}$
- torsional stiffness $\boldsymbol{K}_i^J$

These separated stiffnesses can simply be added to represent the total stiffness as long as transverse shear is negligible.

$$\boldsymbol{K} = \boldsymbol{K}_r + \sum_{i=1}^{d_{shell}} (t_i \boldsymbol{K}_i^m + t_i^3 \boldsymbol{K}_i^b) \tag{8.8}$$

$$+ \sum_{j=1}^{d_{beam}} (A_j \boldsymbol{K}_j^A + I_{1j} \boldsymbol{K}_j^{I_1} + I_{2j} \boldsymbol{K}_j^{I_2} + I_{12j} \boldsymbol{K}_j^{I_{12}} + J_j \boldsymbol{K}_j^J) \tag{8.9}$$

The separation of the variables for beams also allows total freedom in changing the cross section shapes of the beams. For each part, and each component stiffness, first order derivative modes are created according to chapter 5.5 to form the basis for the reduction:

$$\boldsymbol{\phi}_i = \boldsymbol{K}^{-1} \left[\frac{\partial}{\partial \lambda_i} \boldsymbol{K} \right] \boldsymbol{x} \tag{8.10}$$

These vectors are assembled into a matrix and then ortho-normalized, as described in 5.127, to form the basis of the approximate response manifold (reduced basis):

$$\Phi = \mathbf{orthog}([\boldsymbol{x}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \boldsymbol{\phi}_3, \ldots]) \tag{8.11}$$

Then all necessary system matrices are transformed into the subspace:

$$\overline{\boldsymbol{X}} = \boldsymbol{\Phi}^t \boldsymbol{X} \boldsymbol{\Phi} \tag{8.12}$$

The analysis carried out in the reduced space is then:

$$\overline{\boldsymbol{K}} = \overline{\boldsymbol{K}_r} + \sum_{i=1}^{d_{shell}} \left(t_i \overline{\boldsymbol{K}_i^m} + t_i^3 \overline{\boldsymbol{K}_i^b} \right) \tag{8.13}$$

$$+ \sum_{j=1}^{d_{beam}} \left(A_j \overline{\boldsymbol{K}_j^A} + I_{1j} \overline{\boldsymbol{K}_j^{I_1}} + I_{2j} \overline{\boldsymbol{K}_j^{I_2}} + I_{12j} \overline{\boldsymbol{K}_j^{I_{12}}} + J_j \overline{\boldsymbol{K}_j^J} \right) \tag{8.14}$$

$$\overline{\boldsymbol{x}} = \overline{\boldsymbol{K}}^{-1} \overline{F} \tag{8.15}$$

$$\boldsymbol{x}_{\text{new}} = \Phi \overline{\boldsymbol{x}} \tag{8.16}$$

The $\boldsymbol{x}_{\text{new}}$ represents the estimated displacement vector. Strains are derived from $\boldsymbol{x}_{\text{new}}$ using the specific differential operator in Eq. (4.61).

In this case the rank of the reduced basis (per load case) was 222 (initial vector + 2x103 shell vectors + 3x5 beam vectors). For each load case displacements (global stiffness) and strains (local stiffness) were evaluated and compared to the *exact* analysis. [2]

8.2.3 Comparison of the exact solution to the ARM method

Figure 8.19 compares the exact results with the ARM results. The parametric study was carried out in 25 logarithmic steps $1.1^j \quad j = -12, \ldots, 12$. The parameter ranged from $1.1^{-12} = 0.31863$ to $1.1^{12} = 3.1384$. In order to give a better illustration of the error introduced by the ARM method, the relation between the ARM-analysis and the exact analysis is plotted in Fig. 8.20. over the relative change of the exact analysis compared to the baseline (initial) analysis. The Fig. 8.20 shows that a 400% change in the displacements can be predicted with an error of 10%. E.g. a 360% change in the displacements is predicted and 400% is achieved.

For local stiffness comparison, the elastic strain of certain shell elements was evaluated. The comparison is shown in Fig. 8.21. Also, a relative comparison for the strains is shown in Fig. 8.22.

It can be observed that the prediction of strain is not as good as for the displacements. Furthermore, the error trend for strain prediction is not convex as it is almost the case for displacements.

The reason is that strain evaluation requires a derivation of the displacements, as shown in Eq. (4.61). A maximum error for displacements therefore can become quadratic for strains. Nevertheless, most of the strains show a very good approximation using the ARM method. The strains which are very badly predicted are those which are close to the marked region in Fig. 8.22. Further investigation shows that these strains are very close to zero and are almost unaffected by the design change of the specific parameter. These facts lead to bad relative errors. However

---

[2] The software used was MSC.Nastran

these facts are not crucial for design optimization, as low strains and strains with small changes concerning parameter changes are in general no subject of parametric design.

8.2.4 Numerical effort

As seen in chapter 7 the ARM method is a two step procedure consisting of *data preprocessing* (to create the reduced basis, which has to be done once) and an *analysis* (using the reduced basis), while the *exact* solution can be seen as a one step process.

The pure theoretical flop counts based Fig. 7.13 shows that the break even point for the ARM method concerning this model would be approximately **two** baseline analyses.

A previous discussion 7.1 already stressed the point that the pure flop counting gives only a lower estimate of the required computing time, due to fact that disk I/O and storage were not considered. For this example more computing effort was needed to generate the approximate manifold basis $\mathcal{D}$ then the two baseline analyses predicted by Fig. 7.13. This is mainly due to the fact that the used implementation consists of interpreted code which cannot compete with the over decades optimized and also compiled code of the software vendor.

Nevertheless the reanalysis in the reduced basis of the manifold is extremely fast and needs only seconds on low end hardware. This makes it a perfect tool serving as an efficient calculation engine for the proposed multi-criteria optimization task or robust design analyses.

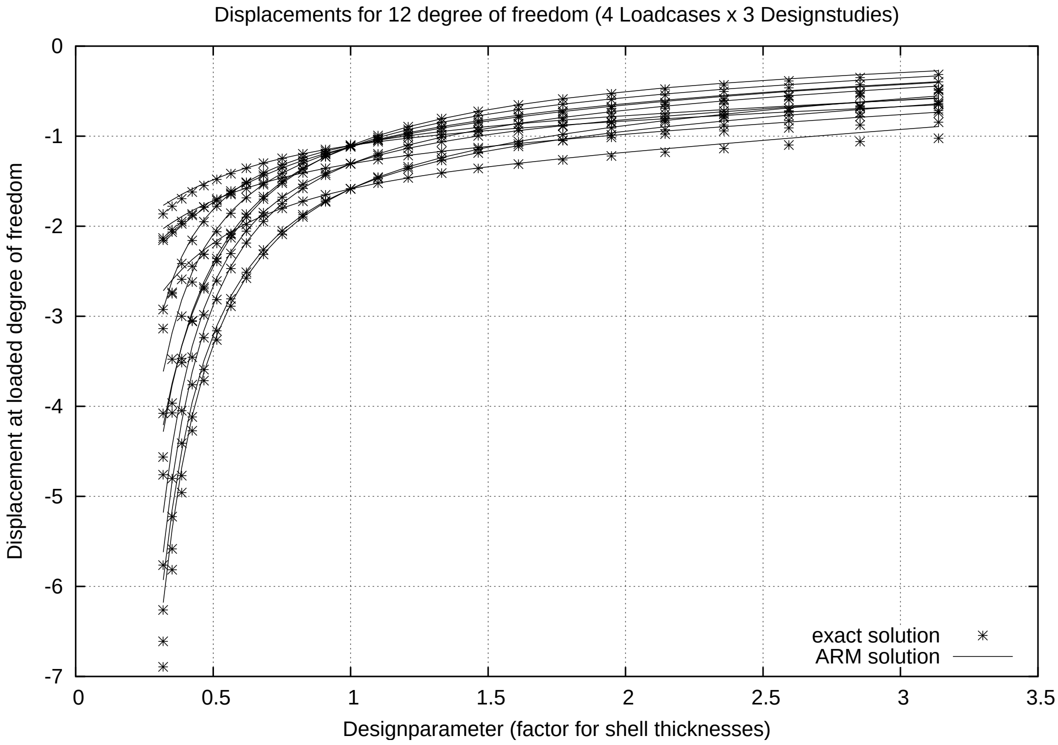


Figure 8.19: Displacement prediction accuracy of the ARM method. The design factor ranged from 0.32 to 3.14, where 1 represents the baseline design.

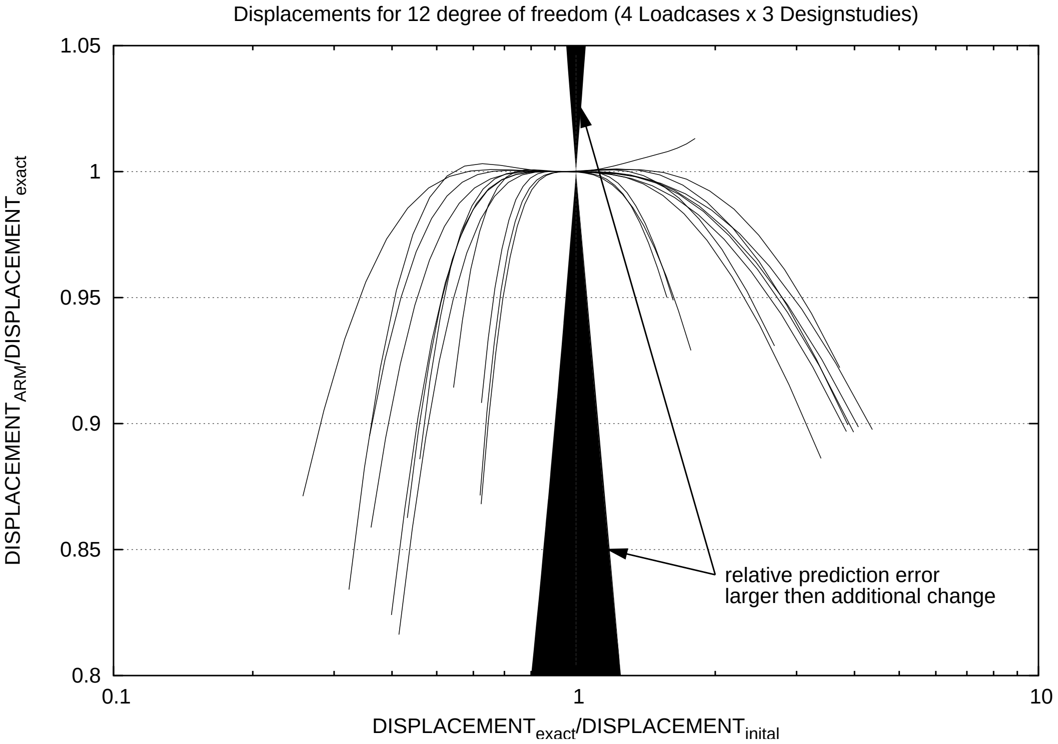


Figure 8.20: Displacement prediction accuracy of the ARM method. The abscissa shows the displacement factor growth relative to the changes in the design variables, while the ordinate indicates the relative error of the ARM method compared to the exact solution. The design factor ranged from 0.32 to 3.14, where 1 represents the baseline design.

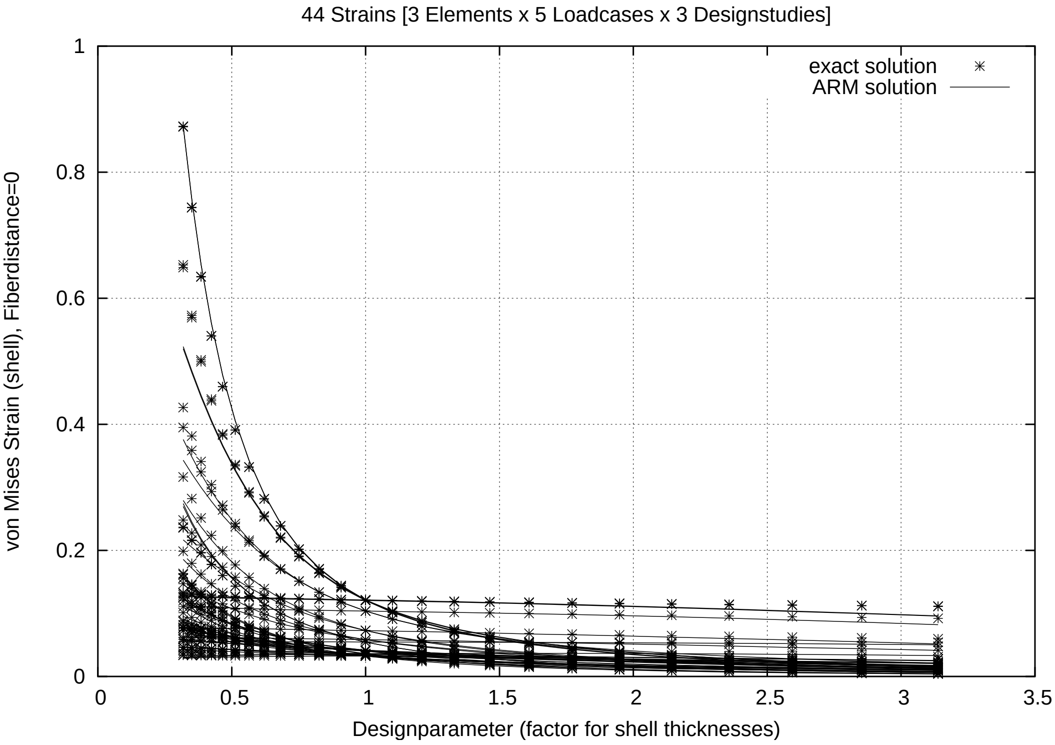


Figure 8.21: Elastic strain prediction accuracy of the ARM method. The design factor ranged from 0.32 to 3.14, where 1 represents the baseline design.

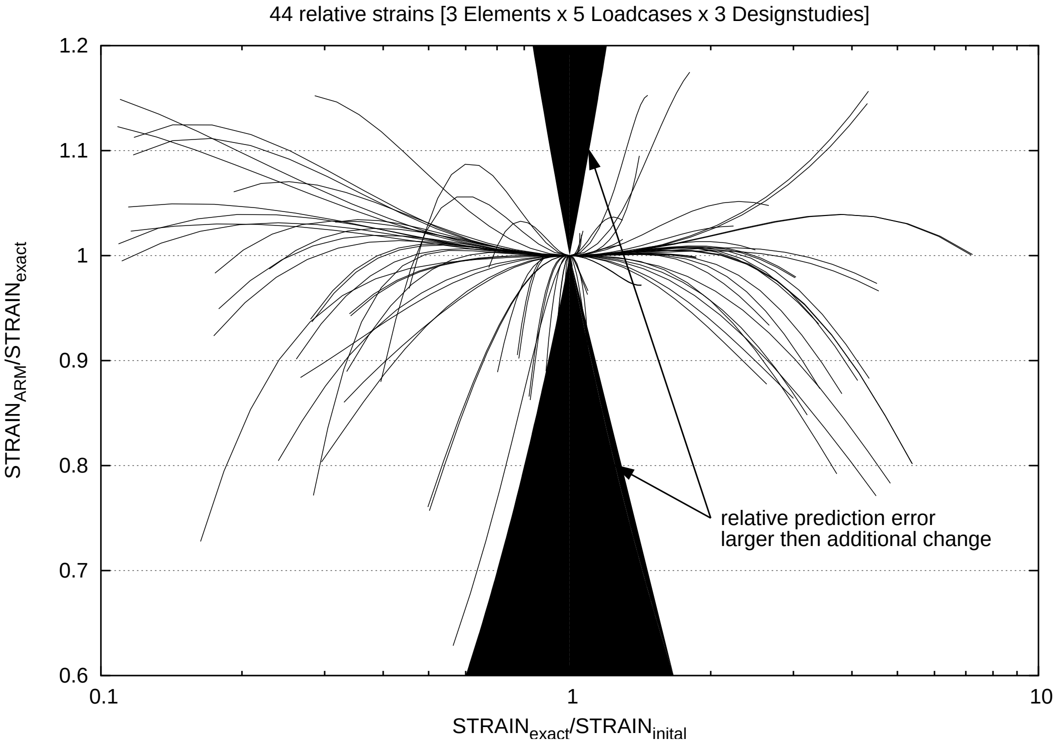


Figure 8.22: Elastic strain prediction accuracy of the ARM method. The abscissa shows the strain factor growth relative to the changes in the design variables, while the ordinate indicates the relative error of the ARM method compared to the exact solution. The design factor ranged from 0.32 to 3.14, where 1 represents the baseline design.

9 Summary

Numerical optimization and reliability analysis have an extremely high demand on computing capacities, because of the high number of required reanalyses for different parameter settings. Especially more sophisticated approaches such as trade off studies, the evaluation of Pareto optima and stochastic optimization are, at least for complex and large real world models beyond the scope of structural simulation today. The capacity of steady growing computing resources is mostly used up by the ever larger models, which are in the focus of the simulation. The solution space of parametric models which is the target of the above mentioned approaches however does primarily depend on the amount of free parameters in the model and not on the numerical degrees of freedom.

After an introduction into structural mechanics and the abstract concept of a manifold, which represents the solution space of parametric models, the thesis demonstrated how this manifold can be practically embedded in a much lower dimensional subspace, starting with the tangent space of the manifold at the baseline solution. It was shown that the response manifold can be approximately embedded in that smaller subspace. Coherent to the derived classical solution of a structural problem using the *Hamiltonian principle* in the displacement degree of freedom space an analysis can be carried out in the embedding subspace. Instead of using local shape functions, describing displacements of grids, the generalized coordinates are now global shape functions, which are derivative modes of the baseline solution.

The dimension of the response manifold embedding Euclidian space does only depend on the amount of free parameters and the desired accuracy. However this means that for todays approach of topology optimization, by treating each element as a design variable, the method will be extremely inefficient if it is applied formally. If the method should be applied to topology optimization problems other forms of reduction vectors as the described derivative modes have to be developed in the future.

Although the theory is general enough to hold also for other non-iterative implicit system of equations the approach for structural systems is given including a ready to code recipe.

The whole benefit of dramatically reducing computation time is supported by an intensive study about the computational complexities of the presented ARM method and the standard approach of using full analyses. It is shown for which amount of displacement degrees of freedom and which amount of free parameters a computing time reduction can be expected. This also demonstrates that, as long as the amount of desired parameters does not dramatically increase, future growth of model size does not restrict the optimization capabilities. Instead the process of design evaluations will accelerate in the future due to faster computing capabilites.

Two larger real world examples demonstrate the capabilities and effectiveness of the approach.

1. A trade off study between glue stiffness and shell thickness of a engine hood's inner sheet, targeting weight and stiffness of a typical limousine, was the target of the first example. The high accuracy of a huge parameter range was demonstrated in detail. The comparison with the surrogate model of the response surface method showed advantage of the developed approach.
2. A large scaled model of a full body in white of a sedan was subject to the second example. There it was shown that also a larger amount of design variables (103) does not harm the accuracy of displacement and strain prediction for reasonable parametric changes (factor 3).

The efficiency and mightiness, as demonstrated on real word examples, recommends further investigation in the following subjects:

- Within the work of this thesis there exists no experience concerning the radius of convergence (necessary amount of derivative modes) for shape optimization problems.
- The ratio of efficiency and accuracy for steady state dynamic load cases was not investigated and should be of interest in further studies. Especially the eigenvalue spectrum of the krylov subspace generating matrix is significantly different compared to static problems. This might restrict the approach to either small frequency ranges or lead to inefficient large subspaces.

A Matrix Utility Operations

Not all physical degrees of freedom (DOF) introduced by the discretization step are mathematically independent. In order to respect this feature these dependencies have to be resolved. This is either done by using *Lagrange multipliers* with increasing the amount of DOF or by eliminating the dependencies by matrix operations. Examples for these degrees of freedom are those with boundary conditions, such as fixed displacements (single point constraints, SPC), or multi-point constraints (MPC) which are defined by rigid body elements or constraint elements. The current chapter defines these sets and shows the basic operations necessary to create an equation of motion without dependencies between degrees of freedoms.

A.1 Physical sets introduced by discretization

For the following deliberations we have to define certain sets of degrees of freedom (DOF). (See also [94]). The g-set defines all degree of freedom. It consists of the mutually exclusive m-set, and n-set. The m-set contains all *dependent* degrees of freedom introduced by multi-point constraints. The n-set represents the independent DOF. Itself consists of a k-set, which describes independent DOF referenced by those in the m-set, and others which are not referenced by any elements creating multi-point constraints. The n-set can also be split into an s-set and an a-set. The s-set defines degrees of freedom containing boundary conditions (SPC), fixed or prescribed displacements. After eliminating the m-set and s-set the resulting a-set remains as the solution set for solving the equations of motion. For sub-models we introduce further the $p-$set containing all degrees of freedom belonging to a certain part. As well as the $q-$set which contains the intersection of the $p-$set and the $n-$set plus degrees of freedom on which certain $p-$set DOFs depended on.

- *g-set*, all DOF of whole model
- *n-set*, independent DOF of whole model
- *m-set*, dependent DOF of whole model, $g = n \cup m$, $n \cap m = \emptyset$
- *s-set*, fixed DOF of whole model
- *a-set*, free DOF of whole model, $n = a \cup s$, $a \cap s = \emptyset$

- *p-set*, *inside* DOF. DOF of a part $p \subseteq g$
- *h-set*, DOF referenced (masters of dependent one) by a part
- *u-set*, dependent DOF in part $u = m \cap p$
- *r-set*, union of *inside* DOF and referenced DOF $r = p \cup h$
- *q-set*, reduction of *r-set* to independent DOF $r = q \cup u$, $q \cap u = \emptyset$
- *b-set*, reduction of *q-set* by eliminating fixed (*s-set*) DOF $b = q \ominus s$, with $\ominus$ meaning *without.*

A.2 Matrix Notations

Matrices are in this chapter noted with capital letters. A lower index containing a $\times$ indicates the size of the matrix. For example $X_{m\times n}$ contain the following information:

- X, the actual name of the matrix
- m the number of rows of the matrix
- n the number of columns of the matrix
- $[X_{m\times n}]_{ij}$ denotes the element in the $i-$th row and $j-$th column.

A Capital letter with ˆ, for example $\hat{X}_g^m$ denotes a partition vector. A partition vector is a column vector which is used for partitioning/merging. In this case $\hat{X}$ has g rows and m entries of these rows are nonzero, $g - m$ are zero.

Matrix-partitioning is denoted by

$$\begin{bmatrix} A_{(g-m)\times(h-n)} & B_{(g-m)\times n} \\ C_{m\times(h-n)} & D_{m\times n} \end{bmatrix} := Partn(X_{g\times h}, \hat{E}_g^m, \hat{F}_h^n) \tag{A.1}$$

$\hat{E}$ is the row partitioning vector, $\hat{F}$ the column partitioning vector.

- $A_{(g-m)\times(g-n)}$ contains the rows and columns of $X_{g\times h}$ where $\hat{E}$ and $\hat{F}$ have zeros.
- $B_{m\times(g-n)}$ contains the rows and columns of $X_{g\times h}$ where $\hat{E}$ has non-zeros and $\hat{F}$ has zeros.
- $C_{(g-m)\times n}$ contains the rows and columns of $X_{g\times h}$ where $\hat{E}$ has zeros and $\hat{F}$ has non-zeros.
- $D_{m\times n}$ contains the rows and columns of $X_{g\times h}$ where $\hat{E}$ and $\hat{F}$ have non-zeros.

Matrix-merging is similar, denoted by

$$G = \begin{bmatrix} A_{(g-m)\times(h-n)} & B_{(g-m)\times n} \\ C_{m\times(h-n)} & D_{m\times n} \end{bmatrix} := Merge(A_{(g-m)\times(h-n)}, B_{(g-m)\times n}, C_{m\times(h-n)}, D_{m\times n}, \hat{E}_g^m, \hat{F}_h^n) \tag{A.2}$$

With $\hat{E}$ as the row merging vector and $\hat{F}$ as the column merging vector.

A.3 Multi-point Constraint Elimination

A.3.1 Multi-point Constraint Elimination for full sized matrices

Assume we have a system matrix in *g-size* $X_{g\times g}$. In the matrix $R_{m\times g}$ each row defines a constraint equation. $R_{m\times g}$ will now be partitioned into $\bar{R}_{m\times n}$ and $\bar{R}_{m\times m}$ using the partition vector $\hat{A}^m_g$ ($\hat{A}$ has m non-zeros).

$$[\bar{R}_{m\times n}\bar{R}_{m\times m}] \quad = Partn(R_{m\times g}, 0, \hat{A}^m_g) \quad \text{wiht } m+n=g \tag{A.3}$$

$$R_{m\times n} \quad = -[\bar{R}_{m\times m}]^{-1}\bar{R}_{m\times n} \tag{A.4}$$

Now the system matrix will be symmetrically partitioned according to the m-set and n-set and then the constraints will be eliminated.

$$\begin{bmatrix} \bar{X}_{n\times n} & \bar{X}_{n\times m} \\ \bar{X}_{m\times n} & \bar{X}_{m\times m} \end{bmatrix} = Partn(X_{g\times g}, \hat{A}^m_g, \hat{A}^m_g) \tag{A.5}$$

$$X_{n\times n} = \bar{X}_{n\times n} + [R_{m\times n}]^t\bar{X}_{m\times n} + \bar{X}_{n\times m}R_{m\times n} \tag{A.6}$$

$$+ [R_{m\times n}]^t\bar{X}_{m\times m}R_{m\times n} \tag{A.7}$$

The remaining $X_{n\times n}$ is free of constraints.

A.3.2 Multi-point Constraint Elimination for sub-models (components)

A system operator like the stiffness matrix has in general full rank. In general it is a linear combination of its d components (sub-models):

$$X_{g\times g} \quad = \sum_{i=1}^{d} X^i_{g\times g} \tag{A.8}$$

where $X^i_{g\times g}$ is the corresponding system matrix of the i-th component having full system size g. For numerical efficiency and generality in adding/removing components it is suitable not storing the $X^i_{g\times g}$, which contain lots of zero columns and rows, but therefore only the nonzero part. Let's assume we have p nonzero columns/rows in one of the $X^i_{g\times g}$. We will store $X^i_{g\times g}$, then as $X^i_{p\times p}$ and a merge vector ${}^i\hat{P}^p_g$ with p nonzero entries indicating the columns and rows where $X^i_{p\times p}$ has to be inserted to build $X^i_{g\times g}$. We will derive now the process of Multi-point Constraint Elimination (MCE) on a sub-model basis using partition vectors (seeking $X^i_{q\times q}$ and the partition vector $\hat{Q}^p_n$ with q non-zeros).

The following combinations of sets exist for our case:

$$(110000) \quad = \quad [\hat{A}^m_g]^t : \text{indicates the } m \text{ deepended DOF} \tag{A.9}$$

$$(011100) \quad = \quad [\hat{X}^p_g]^t : \text{indicates the } p \text{ DOF contributed by a part} \tag{A.10}$$

$$(\quad 0110) \quad = \quad [\hat{C}^h_n]^t : \text{indicates the } h \text{ master DOF referenced by} \tag{A.11}$$

$$\text{depended DOF of the part } X_{p\times p} \tag{A.12}$$

$$(000110) \quad = \quad [\hat{C}^h_g]^t : \text{indicates the } h \text{ master DOF referenced by} \tag{A.13}$$

$$\text{depended DOF of the part } X_{p\times p} \tag{A.14}$$

$$(011110) \quad = \quad [\hat{D}^q_g]^t : \text{the } \textit{union} \text{ of } \hat{X}^p_g \text{ and } \hat{C}^k_g, \text{ containing nonzeros of both} \tag{A.15}$$

$$(110000) \quad = \quad [\hat{E}^p_q]^t : \text{the interscection of } \hat{D}^q_g \text{ and } \hat{X}^p_g \tag{A.16}$$

$$(110000) \quad = \quad [\hat{F}^u_q]^t : \text{the interscection of } \hat{D}^q_g \text{ and } \hat{A}^p_g \tag{A.17}$$

$$\begin{bmatrix} 0 & 0 \\ 0 & X_{p\times p} \end{bmatrix}_{g\times g} = Merge(0, 0, 0, X_{p\times p}, \hat{x}^p_g, \hat{X}^p_g) \tag{A.18}$$

We partition now $\hat{X}^p_g$ using $\hat{A}^m_g$:

$$\begin{bmatrix} \hat{X}^{p-u}_{(g-m)} \\ \hat{X}^u_m \end{bmatrix}_{g\times 1} = Partn(\hat{X}^p_g, \hat{A}^m_g, 0) \tag{A.19}$$

If $u = \emptyset$ our sub-model contains no depended degrees of freedom and we are done by setting:

$$\begin{aligned} q &= \quad p \\ \hat{X}^q_n &= \quad \hat{X}^p_n \\ X_{q\times q} &= \quad X_{p\times p} \end{aligned} \tag{A.20}$$

Assume now $u \neq \emptyset$. We use now $\hat{X}^u_m$ to partition $R_{m\times n}$ and get the depended degrees of freedom of our sub-model, then we blow up the partition vector to g-size:

$$\begin{bmatrix} R_{(m-u)\times n} \\ R_{u\times n} \end{bmatrix}_{m\times n} = \quad Partn(R_{m\times n}, \hat{X}^u_m, 0) \tag{A.21}$$

$$[\hat{C}^h_n]_{i\in n} = \begin{cases} 0 & \text{if } [R_{u\times n}]_{ji} = 0, \forall j \in u \\ 1 & \text{else} \end{cases} \tag{A.22}$$

$$\hat{C}^h_g = \quad Merge(\hat{C}^h_n, 0, 0, 0, \hat{A}^m_g, 0) \tag{A.23}$$

$\hat{C}^h_g$ has h non-zeros. Now we merge the partition vectors $\hat{C}^h_g$ and $\hat{X}^p_g$ in order to get the new r-size ($r = p \cup h$), a temporary size which contains all DOFs of the part model as well as the

DOF which the part depends on. The system matrix $X_{p\times p}$ has now to be expanded to fill this size:

$$\hat{D}_g^r = \hat{C}_g^h \cup \hat{X}_g^p \tag{A.24}$$

$$\begin{bmatrix} \hat{X}_{g-r}^0 \\ \hat{X}_r^p \end{bmatrix}_{g\times 1} = Partn(\hat{X}_g^p, \hat{D}_g^r, 0) \tag{A.25}$$

$$X_{r\times r} = \begin{bmatrix} 0 & 0 \\ 0 & X_{p\times p} \end{bmatrix}_{r\times r} \tag{A.26}$$

$$= Merge(0, 0, 0, X_{p\times p}, \hat{X}_r^p, \hat{X}_r^p) \tag{A.27}$$

Partitioning $\hat{A}_g^m$ using $\hat{D}_g^r$ will result in the partition vector $\hat{A}_r^u$ which now gives us the u depended DOF of our part in the r-size.

$$\begin{bmatrix} \hat{A}_{g-r}^{m-u} \\ \hat{A}_r^u \end{bmatrix}_{g\times 1} = Partn(\hat{A}_g^m, \hat{D}_g^r, 0) \tag{A.28}$$

Now we can split $X_{r\times r}$ into a depended part and an independent part:

$$\begin{bmatrix} Y_{q\times q} & Y_{q\times u} \\ Y_{u\times q} & Y_{u\times u} \end{bmatrix}_{r\times r} = Partn(X_{r\times r}, \hat{A}_r^u, \hat{A}_r^u) \tag{A.29}$$

To cut down $R_{u\times n}$ in Eq. (A.21) to our new part size we have to get the partition vector:

$$\begin{bmatrix} \hat{D}_n^q \\ \hat{D}_m^u \end{bmatrix}_{g\times 1} = Partn(\hat{D}_g^r, \hat{A}_g^m, 0) \tag{A.30}$$

Now we can partition $R_{u\times n}$ (Eq. (A.21)) down to the new independent part size q using Eq. (A.4):

$$[R_{u\times(n-q)} \qquad R_{u\times q}] = Partn(R_{u\times n}, 0, \hat{D}_n^q) \tag{A.31}$$

The final part matrix and it's partition vector for n-size we get using Eq. (A.7):

$$\begin{aligned} X_{q\times q} &= Y_{q\times q} + [R_{u\times q}]^t Y_{u\times q} + Y_{q\times u} R_{u\times q} \\ &\quad + [R_{u\times q}]^t Y_{u\times u} R_{u\times q} \end{aligned} \tag{A.32}$$

The corresponding merge vector is $\hat{X}_n^q = \hat{D}_n^q$.

A.4 Single-point Constraint Elimination

Single-point constraint elimination can be seen as a special case of multi-point point constraint elimination, with the $R_{m\times n} = 0$.

$$\begin{bmatrix} \hat{X}_a^{q-v} \\ \hat{X}_s^v \end{bmatrix}_{n\times 1} = Partn(\hat{X}_n^q, \hat{S}_n^s, 0) \tag{A.33}$$

If $v \equiv 0$ (no single point constraints in the component) we are done by setting:

$$\begin{aligned} b &= q \\ \hat{X}_a^b &= \hat{X}_a^{q-v} \\ X_{b\times b} &= X_{q\times q} \end{aligned} \tag{A.34}$$

If $v \neq 0$ we derive the DOF which have to be eliminated in the component by first partitioning:

$$\begin{bmatrix} \hat{S}_{n-q}^b \\ \hat{S}_q^v \end{bmatrix}_{n\times 1} = Partn(\hat{S}_n^s, \hat{X}_n^q, 0) \tag{A.35}$$

and then we eliminate the constrained DOF:

$$\begin{bmatrix} X_{b\times b} & X_{b\times v} \\ X_{v\times b} & X_{v\times v} \end{bmatrix}_{q\times q} = Partn(X_{q\times q}, \hat{S}_q^v, \hat{S}_q^v) \tag{A.36}$$

The partition vector is then also $\hat{X}_a^b$.

B Synthesis of system matrices

The ARM method requires that the system matrices for a current configuration, given e.g. by a different setting of parameters and design variables, are generated for the solution of the system. The general way to generate the system matrices for a given configuration would be to generate the matrix in displacement coordinates $\mathcal{S}$ and then transform it into modal coordinates $\mathcal{D}$. This transformation is however a time consuming task and therefore not suitable for a fast reanalysis.

As long as the system matrix is analytical concerning the design variables, the presented approach is that the analytical components are transformed during the *generation phase* of the response manifold. The assembling of the system matrices in the response manifold coordinates obey then to the same analytical synthesis as in displacement coordinates. However for the general case there might be no finite analytical description of this behavior or it simply might be unknown.

This can be the case for example for shell elements with a large influence of *transverse shear*, *shape variables* or simply any other unknown parametricity.

The subject of this chapter is to describe how to deal with them.

B.1 Analytical description

As discussed before we need an analytical description of the behavior of the system matrices concerning the design variables. If we do not know this behavior we define one which is sufficient by our needs. This can be done with one of the following approaches:

- for variables available only in discrete gauges, e.g. such as shell thicknesses we could provide all of these (limited) gauges
- we can use Taylor series expansion of the system matrices according to the design variable
- we can use any interpolation scheme such as *cubic splines* to interpolate between some discrete points

B.1.1 Example for shell elements with transverse shear

The stiffness of shell elements without transverse shear can be simply synthesized by:

$$\boldsymbol{K}(E,t) = E(t\boldsymbol{K}^m + t^3\boldsymbol{K}^b) \tag{B.1}$$

where m represents *membrane* and b *bending* stiffness.

There exists obviously a significant influence of transverse shear if the system matrix generated by this approach differs significantly from the one generated using transverse shear using the thickness of the baseline run. If the influence of transverse shear is significant, but still small we can make the following assumption:

$$\boldsymbol{K}(E,t) = E(t\boldsymbol{K}^m + t^3\boldsymbol{K}^{bs}) \tag{B.2}$$

Here the bs index indicates that we use a mixed transverse-shear-bending matrix. This approach ensures that we get exactly the same result for the baseline run regardless of the coordinates (displacement, arm). $\boldsymbol{K}^{bs}$ is defined via the membrane stiffness $\boldsymbol{K}^m$ and the total stiffness defined by the shell thickness of the baseline run ($t = t_0, E = E_0$):

$$\boldsymbol{K}(E_0,t_0) = E_0(t_0\boldsymbol{K}^m + {t_0}^3\boldsymbol{K}^{bs}) \tag{B.3}$$

$$\boldsymbol{K}^{bs} = \frac{1}{{t_0}^3}(\frac{1}{E_0}\boldsymbol{K}(E_0,t_0) - t_0\boldsymbol{K}^m) \tag{B.4}$$

The resulting stiffness matrix of arbitrary settings of E, t is then given by:

$$\boldsymbol{K}(E,t) = E[t\boldsymbol{K}^m + t^3(\frac{1}{{t_0}^3}(\frac{1}{E_0}\boldsymbol{K}(E_0,t_0) - t_0\boldsymbol{K}^m))] \tag{B.5}$$

$$\boldsymbol{K}(E,t) = E(t - \frac{t^3}{{t_0}^3}t_0)\boldsymbol{K}^m + (\frac{Et^3}{E_0{t_0}^3})\boldsymbol{K}(E_0,t_0) \tag{B.6}$$

This satisfies the following requirements:

- exact results for the baseline run
- reasonable stiffness behavior for moderate transverse shear influence
- simple and short analytical description

C Properties of Krylov subspaces

The Krylov subspace $K_r(x, A)$ plays an important role in approximating our response manifolds. We will give some properties here which are used within the thesis.

C.1 Krylov subspaces

Let $x \in \mathbb{C}^n$ and $x \in \mathbb{C}^{n \times n}$. A Krylov subspace $K_r(x, A)$ is defined by:

$$K_r(x, A) = \mathbf{span}\{x, Ax, A^1x, A^2x, \ldots, A^{r-1}x\} \tag{C.1}$$

Let further be

$$\mathbb{C}^n = B_1 \oplus B_2 \oplus \ldots \oplus B_m \tag{C.2}$$

the direct sum of subspaces. Where B_i is the eigenspace to the i-th of m distinct eigenvalues λ_i of A. Without restricting generality we can assume that the B_i are labeled in that way that $|\lambda_i| < |\lambda_{i+1}|$. Note that the B_i are not necessarily one-dimensional. Furthermore they are spanned up by all eigenvectors with eigenvalue λ_i. Due to the orthogonality and completeness of the B_i we can express x in terms of $b_i \subset B_i$, with $\|b_i\| = 1$:

$$x = \sum_{i=1}^{m} \beta_i b_i \quad \beta_i \in \mathbb{C} \tag{C.3}$$

Lets denote x_i the i-th Krylov vector $A^i x$. Then the following states:

Theorem C.1

1. *all* x_i*, with* $i > 0$ *are orthogonal to* $\mathbf{kernel}(A)$
2. $\lim_{i\to\infty} x_i = \alpha b_m$ *with* $\alpha \in \mathbb{C}$

Proof C.1

1. *Lets assume that* $\lambda_1 = 0$, *otherwise the* **kernel**$(A) = 0$ *and the proof is trivial, then*

$$x_0 = \sum_{i=1}^{m} \beta_i b_i \tag{C.4}$$

$$\rightarrow x_1 = A \sum_{i=1}^{m} \beta_i b_i \tag{C.5}$$

$$\rightarrow x_1 = \sum_{i=1}^{m} \beta_i A b_i \tag{C.6}$$

$$\rightarrow x_1 = \sum_{i=1}^{m} \beta_i \lambda_i b_i \tag{C.7}$$

$$\rightarrow x_1 = \sum_{i=2}^{m} \beta_i \lambda_i b_i \tag{C.8}$$

as all b_i *are orthogonal,* $x_i, i > 0$ *must be orthogonal to* b_1 *(kernel(A))*
q.e.d.

2. *from Eq. (C.7) follows:*

$$x_j = \sum_{i=1}^{m} \beta_i \lambda_i^j b_i \tag{C.9}$$

$$\rightarrow x_j = \lambda_m^j \left[\left(\sum_{i=1}^{m-1} \beta_i \left(\frac{\lambda_i}{\lambda_m} \right)^j b_i \right) + \beta_m b_m \right] \tag{C.10}$$

$$\rightarrow \lim_{j \to \infty} x_j = \lambda_m^j \left[\beta_m b_m \right] \tag{C.11}$$

$$\rightarrow \lim_{j \to \infty} x_j = \alpha b_m \quad \textit{with } \alpha = \lambda_m^j \beta_m \tag{C.12}$$

$$\tag{C.13}$$

List of Figures

2.1 Ten-bar truss 7

5.1 Schematic display of discretization error 30

6.1 Deflection of ten-bar truss 50
6.2 Derivative mode 1 of the ten-bar truss 51
6.3 Derivative mode 2 of the ten-bar truss 51
6.4 Derivative mode 2 of the ten-bar truss 52

7.1 2- and 3-dimensional test models 63
7.2 Numbering scheme for the 1d and 2d model 67
7.3 Numbering scheme for the 3d model 68
7.4 Front-width of 1d artificial model 69
7.5 Front-width of 2d artificial model 70
7.6 Front-width of 3d artificial model 71
7.7 Nonzeroes in the Cholesky factor 71
7.8 Non-zeros in the Cholesky factor of 1d model 72
7.9 Non-zeros in the Cholesky factor of 2d model 72
7.10 Non-zeros in the Cholesky factor of 3d model 73
7.11 Front-width prediction compared to real world data 73
7.12 Prediction of non-zeros in the Cholesky factor compared to real world data . . . 74
7.13 Estimation of amount of reanalyses for computational breakeven point 80

8.1 Engine hood example 84
8.2 Parametric solution of glue stiffness 85
8.3 Displacements of engine hood of load-case-1 86
8.4 Displacements of engine hood of load-case-2 87

8.5 Displacements of engine hood of load-case-1 88
8.6 ARM based approximation using derivative modes: local 89
8.7 ARM based approximation using derivative modes: global 90
8.8 ARM based approximation using residual and derivative modes: global 91
8.9 ARM based approximation using residual and derivative modes: local 92
8.10 RSM based approximation: local 93
8.11 RSM based approximation: global 94
8.12 Comparison between RSM and ARM: local 94
8.13 Comparison between RSM and ARM: global 95
8.14 Pareto front between glue stiffness and sheet thickness 95
8.15 Result topology optimization of engine hood 96
8.16 Example of a body in white 98
8.17 Example of design variables for body in white 98
8.18 Stiffness and strain sensors of body in white 99
8.19 Displacement prediction accuracy of the ARM method 103
8.20 Displacement accuracy of the ARM method: relative 103
8.21 Strain prediction accuracy of the ARM method 104
8.22 Strain prediction accuracy of the ARM method: relative 104

Bibliography

[1] E. Aarts and J. Korst. *Simulated annelaing and Boltzmann machines.* Whiley-Interscience series in discrete mathematics and optimization. Whiley, Chichester, 1989.

[2] R. Abraham and J.E. Marsden. *Foundations of Mechanics.* The Benjamin/Cummings Publishing Company, London, 2. edition, 1978.

[3] M. A. Akgun, R.T. Haftka, and J.H. Garcelon. Fast exact static structural reanalysis in relation to the Sherman-Morrison-Woodbury equation. *ISSMO/NASA/AIAA First Internet Conference on Approximation and Fast Reanalysis Techniques in Engineering Optimization*, 14, June 14-27 1998.

[4] A. Aktas and F. Moses. Reduced basis eigenvalue solutions for damaged structure. *Mech. Struc. Mach.*, 26:63–79, 1987.

[5] B.O. Almroth, P. Stern, and F.A. Brogan. Automatic choice of global shape functions in structural analysis. *AIAA Journal*, 16(5):525–528, 1978.

[6] R.R. Arnold, R.L. Citerley, M. Chargin, and D. Galant. Application of ritz vectors for dynamic analysis of large structures. *Computers and Structures*, 21(3):461–467, 1985.

[7] A.C. Arora. Survey of structural reanalysis techniques. *Jouernal of Structural Division ASCE*, 102(4):783–802, 1976.

[8] H. Baier, C. Seesselberg, and B. Specht. *Optimierung in der Strukturmechanik.* Vieweg, Braunschweig, 1994.

[9] J-F.M. Barthelemy and R. T. Haftka. Approximation concepts for optimum structural design - a review. *Structural optimization*, 5:129–144, 1993.

[10] K.J. Bathe. *Finite Elemente Methoden.* Springer-Verlag, Berlin, 1990.

[11] S. Chen, C. Huang, and Z. Liu. Structural approximate reanalysis for topological modifications by finite element systems. *AIAA*, 36:1760–1762, 1998.

[12] R.W. Clough and J. Penzien. *Dynamics of structures.* McGraw-Hill International Editions, 175.

[13] R.R Craig. *Structural Dynamics–An Introduction to Computer Methods.* John Wiley & Sons, New York, 1981.

[14] R.R. Craig and M.C.C. Bampton. Coupling of substructures for dynamic analysis. *AIAA Journal*, 6(7):1313–1319, 1968.

[15] R.R. Craig and C-J. Chang. On the use of attachment modes in substructure coupling for dynamic analysis. In *Proceedings of the AIAA/ASME 18th Struct., Struct. Dyn., and Matrials Conference*, Paper 77–405, San Diego, Ca, 1977.

[16] R.R. Craig and Z. Ni. Component mode synthesis for modal order reduction of nonclassical damped systems. *AIAA Journal of Guidance, Control and Dynamics*, 12(4), 1989.

[17] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. *Proc. 24th Nat. conf. Assoc. Comput. Mach., ACM*, pages 157–172, 1969.

[18] I. Das. On characterizing the knee of the pareto curve based on normal-boundary intersection. *Structural Opotimization*, 18:107–115, 1999.

[19] L. Davis. *Genetic algorithms and simulated annealing.* Research Notes in artificial intelligence. Pitman, London, 1988.

[20] H. Ding and R.H. Gallagher. Approximate force method reanalysis techniques in structural optimization. *Int. J. Numer. Meht. Engrg.*, 21:1253–1271, 1985.

[21] I.S. Duff, A.M. Erisman, and J.K. Reid. *Monographs on numerical analysis.* Clarendon, 1986.

[22] H. Eschenauer, J. Koski, and A. Osyczka. *Multicriteria Desgin Optimization.* Springer-Verlag, Berlin, 1. edition, 1990.

[23] J.B. Fink and W.C. Rheinboldt. On the discretization error of parametrized nonlinear equations. *SIAM J: Numer. Anal.*, 20:732–746, 1983.

[24] J.B. Fink and W.C. Rheinboldt. On the error behavior of the reduced basis technique for nonlinear finite element approximations. *ZAMM*, 63:21–28, 1983.

[25] J.B. Fink and W.C. Rheinboldt. Solution manifolds and submanifolds of parametrized equations and their discretization errors. *Numerische Mathematik*, 45:323–343, 1984.

[26] R. Fletcher and C. M. Reeves. *Function Minimization by Conjugate Gradients*, volume 7. Br. Computer J., 1964.

[27] O. Forster. *Analysis 2, Differentialrechnung im* $\mathbb{R}^n$*, Gewöhnliche Differentialgleichungen.* Vieweg, Braunschweig, 1984.

[28] O. Forster. *Analysis 3, Integralrechnung im* $\mathbb{R}^n$*.* Vieweg, Braunschweig, 1984.

[29] R.I. Fox and H. Miura. An approximate analysis technique for design calculations. *AIAA Journal*, 9(1):177–179, January 1972.

[30] R. Freymann. *Advanced numerical and experimental methods in the field of vehicle structural-acoustics*. PhD thesis, Habilitationsschrift, Technische Universität München, 2000.

[31] A. George and J. W. Liu. *Computer Solutions of Large Sparse Positive Definite Systems*. Prentice Hall, 1981.

[32] M. A. Gockel. *Handbook for Dynamic Analysis*. The MacNeal-Schwendler Corporation, 1983.

[33] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Pub Co, 1989.

[34] R.J. Guyan. Reduction of stiffness and mass matrices. *AIAA Journal*, 3(2):380, 1965.

[35] Udo Haenle. *Modellierung und Berechnung dynamisch beanspruchter elastischer Strukturen bei endlichen Rotationen*. PhD thesis, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart, 1995.

[36] W. W. Hager. Updating the inverse of a matrix. *SIAM Review*, Vol. 31(No. 2):221–239, 1989.

[37] S.W. Hawking and G.F.R. Ellis. *The Large Scale Structure of Space–Time*. Cambridge University Press, 1973.

[38] D. F. X. Heiserer and H. Baier. Applied reanalysis techniques for large scale structural mechanics multidisciplinary optimization in automotive industry. In *4th World Congress of Structural and Multidisciplinary Optimization, 4.-8. June, Dalian, China*, 2001.

[39] D. F. X. Heiserer and M. Chargin. High performance, process oriented, weld spot approach. In *MSC 1st Worldwide Automotive Users Conference, Munich*, 1999.

[40] D. F. X. Heiserer and M. Chargin. Optimization of engine mounts for NVH response using global search algorithms. In *ASME Optimization in Industry-II, Banff Center for Conferecnes June 6-11, Canada 1999*, 1999.

[41] D. F. X. Heiserer and M. Chargin. Design study for problems involving responses over multiple disciplines. In *MSC 2nd Worldwide Automotvie Users Conference, Detroit*, 2000.

[42] D. F. X. Heiserer and M. Chargin. Full car NVH simulations using a massive parallel paradigm - feasibility study on a Cray SV1. In *Mannheim Supercomputing, 8.-14. June 2000*. Mateo, Mannheim., 2000. ISBN 3-932178-15-7.

[43] D. F. X. Heiserer and M. Chargin. Optimierung von Motorlagern bezüglich des Vibrationskomforts unter Verwendung globaler Suchalgorithmen. In *VDI Kongress, Berechnung und Simulation im Fahrzeugbau, Würzburg 2000*, 2000.

[44] D. F. X. Heiserer and M. Chargin. Optimierung von Schweisspunkten unter Fertigungsgesichtspunkten mittels Verwendung von Neuronalen Netzen. In *VDI Kongress, Berechnung und Simulation im Fahrzeugbau, Würzburg 2002*, 2000.

[45] D. F. X. Heiserer, A. Irrgang, and J. Sielaff. Stabile Auslegung des dynamischen Verhaltens einer Fahrzeugkarosserie unter Brücksichtigung vieler Ausstattungsvarianten. In *VDI Kongress, Berechnung und Simulation im Fahrzeugbau, Würzburg 1998*, 1998.

[46] J. Holinicki-Szulc and J.T. Gierlinkski. Structural modifications simulated by virtual distortions. *Int. J. Numer. Meht. Engrg.*, 28:645–666, 1989.

[47] C. Huang and G. Verchery. An exact structural static reanalysis method. *Comm. Num. Meth. Engrg.*, 13, 103-112 1997.

[48] W.C. Hurty. Vibrations of structural systems by component modal synthesis. *Proc. ASCE 86, EM 4*, pages 51–69, 1960.

[49] W.C. Hurty. Dynamic analysis of structural systems using component modes. *AIAA Journal*, 3(4):678–685, 1965.

[50] L. Ingber. Very fast simulated re-annealing. *J Mathl. Comput. Modelling*, 12:967–973, 1989.

[51] L. Ingber. Simulated annelaing: Practice versus theory. *J. Mathl. Comput. Modelling*, 18(11):29–57, 1993.

[52] G. Karypis and V. Kumar. Metis, a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, version 3.0.3. *University of Minnesota, Department of computer sciences/ARMY HPC Research Center, Minneapolis, MN*, 1997, November 5.

[53] K.T. Kavangh. An approximate algorithm for the reanalysis of strucurtes by the finite element method. *Comp. & Struct.*, 2:713–722, 1972.

[54] U. Kirsch. *Optimum structural design.* New York, McGraw Hill, 1981.

[55] U. Kirsch. Optimal topologies of structures. *Applied Mechanics Reviews*, (42):223–239, 1989.

[56] U. Kirsch. Reduced basis approximations of structural displacements for optimal design. *AIAA Journal*, 29:1751–1758, 1991.

[57] U. Kirsch. Efficient reanalysis for topological optimization. *Structural optimization*, 6(143-150), 1993.

[58] U. Kirsch. *Structural optimization: status and promise*, chapter Approximate reanalysis methods. AIAA, 1993.

[59] U. Kirsch. *Structural optimizations, fundamentals and applications.* Springer-Verlag, Heidelbrg, 1993.

[60] U. Kirsch. Effective sensitivity analysis for structural optimization. *Comp. Meth. Appl. Mech. Engrg.*, 117:143–156, 1994.

[61] U. Kirsch. Effective reanalysis of structures, concepts and implementation. Technical report, Technio, Department of Civil Engineering, Haifa, Israel, January 1996.

[62] U. Kirsch. Efficient, accurate approximations for structural optimization. *AIAA Journal*, 37(12):1663–1669, 1999.

[63] U. Kirsch. Combined approximations - a general reanalysis approach for structural optimization,. *Structural Optimization*, 20(2):97–106, 2000.

[64] U. Kirsch. *Design oriented analysis of structures.* Kluwer Academic Publishers, 2002.

[65] U. Kirsch and S. Liu. Exact structural reanalysis by a first-order reduced basis approach. *Structural Optimization*, Vol. 10:153–158, 1995.

[66] U. Kirsch and S. Liu. Improved stiffness-based first-order approximations for structural optimization. *AIAA Journal*, 33:143–150, 1995.

[67] U. Kirsch and S. Liu. Structural reanalysis for general layout modifications. *AIAA Journal*, Vol. 35(No.2):382–388, 1997.

[68] U. Kirsch and P.Y. Papalambros. Accurate displacement derivatives for structural optimization using approximate reanalysis. *Computer Methods in Applied Mechanics and Engineering*, 190:3945–3956, 2001.

[69] U. Kirsch and P.Y. Papalambros. Exact and accurate reanalysis of structures for geometrical changes. *Engineering with Computers*, 17(4):363–372, december 2001. ISSN:0177-0667.

[70] U. Kirsch and P.Y. Papalambros. Structural reanalysis for topological modifications. *Structural and Multidisciplinary Optimization*, 21(5):333–344, July 2001. ISSN:1615-147X.

[71] U. Kirsch and M. F. Rubinstein. Reanalysis for limited structural modifications. *J. Eng. Mech. Div; ASCE; Proc. Paper 8706*, 98:61–70, 1972.

[72] U. Kirsch and M. F. Rubinstein. Structural reanalysis by iteration. *Computers and Structures*, 2:497–510, 1972.

[73] K.A. Kline. Dynamic analysis using a reduced basis of exact modes and Ritz vectors. *AIAA Journal*, 24(12):2022–2029, 1986.

[74] E. Klingbeil. *Tensorrechnung für Ingenieure.* Verlag Bibliographisches Institut & F.A. Brockhaus AG, 1989.

[75] L. Komzsik. *MSC/NASTRAN 2001 - Numerical methods Users Guide.* MSC.Software Corporation.

[76] L. Komzsik. *Handbook for Numerical Methods.* The Mac-Neal Schwendler Corporation, 1990.

[77] L. Komzsik. *The Lanczos Method: Evolution and Application (Software, Environments, and Tools).* SIAM, Society for Industrial and Applied Mathematics, 2003.

[78] C. Lanczos. *An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators*, volume 45. Journal of the Research of the National Bureau of Standards, 1950.

[79] L.D. Landau and E.M. Lifschitz. *Lehrbuch der theoretischen Physik, Band 1 Mechanik.* Akademie-Verlag, 1979.

[80] L.D. Landau and E.M. Lifschitz. *Lehrbuch der theoretischen Physik*, volume 4: Elastizitätstheorie. Akademie–Verlag, Berlin, 1989.

[81] Meei-Yow Lin Lee. Estimation of the error in the reduced basis method solution of differential algebraic equation systems. *SIAM J. Numer. Aanal.*, 28(2):512–528, April 1991.

[82] C-W. Leu, L-J.and Huang. A reduced basis method for geometric nonlinear analysis of structures. *IASS Journal*, 39:71–75, 1998.

[83] R. Levy, U. Kirsch, and S. Liu. Reanalysis of trusses using modified initial designs. *Structural Optimization*, 19, 2001.

[84] R.H. MacNeal. A hybrid method of component mode synthesis. *Computers & Structures*, 1:581–601, 1971.

[85] H. Mang and G. Hofstetter. *Festigkeitslehre.* Springer, 2000.

[86] J.E. Marsden and T.J.R. Hughes. *Mathematical Foundation of Elasticity.* Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1983. ISBN 0-486-67865-2.

[87] D. Mass. *Schnelle rechnerische Komfortoptimierung von Kraftfahrzeugen mittels modaler Korrektur.* PhD thesis, Technische Universität München, 2001.

[88] L. Meirovitch and M.K. Kwak. Inclusion principle for the Rayleigh–Ritz based substructure synthesis. *AIAA Journal*, 30(5):1344–1351, 1992.

[89] M. Meywerk. Optimierung in der Fahrzeugindustrie: Methoden und Anwendungen. In *Berechnung und Simulation im Fahrzeugbau.* VDI-Berichte 1701, 2002.

[90] R.D. Mindlin. Influence of rotary inertia and shear of flexural motion of isotropic elastic plates. *J. Appl. Mech.*, Vol. 12:31–38, 1951.

[91] G.J. Moore. *MSC/NASTRAN Design Sensitivity and optimization.* The MacNeal-Schwendler Corporation, 1992.

[92] MSC/NASTRAN. *DMAP Module Dictionary*, 1994.

[93] MSC.Software Corporation, Santa Ana, CA 92707 USA. *MSC.NASTRAN 2004 Quick Reference Guide.*

[94] MSC.Software Corporation, 2 MacArthur Place, Santa Ana, CA 92707 USA. *MSC.NASTRAN 2004 Reference Manual.*

[95] R.H. Myers and D.C. Montogmery. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments.* Wiley, 2002.

[96] P.B. Nair. Equivalence between the combined approximations technique and krylov subspace methods. *AIAA Journal*, 40(5):1021–1032, 2002.

[97] A.K. Noor. Recent advances in reduction methods for nonlinear problems. *Computers and Structures*, 13:31–44, 1981.

[98] A.K. Noor. On making nonlinear problems small. *Computer Methods in Applied Mechanics and Engineering*, 34:955–985, 1982.

[99] A.K. Noor. Recent advances and applications of reduction methods. *Appl. Mech. Rev.*, 47(5):125–146, 1994.

[100] A.K. Noor and H. E. Lowder. Structural reanalysis via a mixed method. *Computers and Structures*, 5(1):9–12, 1975.

[101] A.K. Noor and H.E. Lowder. Approximate techniques of structural reanalysis. *Computers and Structures*, 4(4):801–812, 1974.

[102] A.K. Noor and H.E. Lowder. Approximate reanalysis techniques with substructuring. *ASCE Journal of the Structural Division*, 101(8):1687–1698, August 1975.

[103] A.K. Noor and J.M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA Journal*, 18(4):455–462, 1980.

[104] M. Paas, H. Ippen, and R. Schilling. Anwendung von genetischen Algorithmen zur Optimierung von Karosseriekomponenten und Identifizeriung von Materialmodellen. In *Berechnung und Simulation im Fahrzeugbau.* VDI-Berichte 1701, 2002.

[105] C.C. Paige. *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices.* PhD thesis, University of London, 1971.

[106] C.C. Paige. Computational variants of the lanczos method for the eigenproblem. *J. Inst. Math. Appl.*, 10:373–381, 1972.

[107] B. Parlett. *The symmetric eigenvalue problem.* SIAM, 1998.

[108] J. Pietrzak. A systematic search for pareto optimum solutions. *Structural Optimization*, 17:79–81, 1999.

[109] T.A. Porsching. Estimation of the error in the reduced basis method solution of nonlinear equations. *Mathematics of Computation*, 45(172):487–496, 1985.

[110] T.A. Porsching and M. L. Lee. The reduced basis method for initial value problems. *SIAM Journal of Numerical Analysis*, 24:1277–1287, 1987.

[111] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes in C.* Cambridge University Press, 1992.

[112] I. Raasch and A. Irrgang. Shape optimization with MSC/NASTRAN. In *MSC/NASTRAN European Users conference, Rome*, 1988.

[113] W. Ritz. über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik. *J. Reine Angew. Math.*, 135:1–61, 1909.

[114] S. Rubin. Improved component–mode representation for structural dynamic analysis. *AIAA Journal*, 13(8):995–1006, 1975.

[115] F. Scheck. *Mechanik.* Springer-Verlag, second edition, 1990.

[116] K. Schittkowski. NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems. *Annals of Operations Research*, 5:485–500, 1986.

[117] L.A. Schmit and B. Farshi. Some approximation concepts for structural synthesis. *AIAA Journal*, 11:489–494, 1974.

[118] J. Sherman and W. J. Morrison. Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix. *Ann. Math. Statist.*, 20:621, 1949.

[119] S. Timoshenko and S. Woinowski-Krieger. *Theory of plates and shells.* McGraw-Hill Book Company, New York, N.Y., 1959.

[120] H.G. Vopel and J. Hillmann. Berücksichtigung der Schweisspunkte bei der FE-Modellierung von Karosserien. *VDI Berichte 1283*, pages 171–181, 1996.

[121] L. Witta. *Entwurf und Realisierung interaktiver modaler Berechnungs- und Optimierverfahren für gekoppelte Struktur-Fluid-Systeme.* PhD thesis, Technische Universität München, 2001.

[122] M. Woodbury. Inverting modified matrices. Technical report, Memorandum Report 42, Statistical research group, Princeton University, Princeton, NJ., 1950.

[123] P. Wynn. The rational approximation of functions which are formally defined by a power series expansion. *Mathematics of Computation*, 14:147–186, 1960.

[124] G. Zhang and B. Richter. Fe-basierte, rechnerische Betriebsfestigkeitsuntersuchung von punktgeschweissten Strukturen. *VDI Berichte Nr. 1411*, pages 245–270, 1998.

[125] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*, volume 1: Basic Formulations and Linear Problems. McGraw–Hill, London, 4. edition, 1989.

[126] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method II*, volume 2: Solid and Fluid Mechanics, Dynamics and Non-Linearity. McGraw–Hill, London, 4. edition, 1989.

Curriculum Vitae

	Daniel Franz Xaver Heiserer
3rd December 1970	born in Haag/Obb., Germany, as first son of Sebastian and Gudrun Heiserer, both teachers, single
1977-1981	Grundschule Heldenstein
1981-1990	Ruperti-Gymnasium Mühldorf am Inn Reifezeugnis 29th June 1990
1988	Industrial placement at SIEMENS AG, *division of telecommunication*
1990-1996	Study of Physics at the Technical University Munich. Diploma-thesis *Beurteilung des dynamischen Verhaltens eines Fahrzeugs* at the Chair of Mechanical engineering (Prof. Pfeiffer). Diploma 15th November 1996
1994	working student at the *institute of energy conversion*, faculty of physics, Technical University Munich
1990-1991	Industrial placement at GRUNDIG OHG, in the area of metal processing
1996	since December 1996 employed as scientific engineer at BMW, research and development center (FIZ) in Munich: - member working group for car interior noise - co-development of finite element based spot-weld approach - responsible for the *Full vehicle Assembly* project - responsible for structural and multi criteria optimization development - responsible for shape optimization