Fakultät für Maschinenwesen
Lehrstuhl für Angewandte Mechanik

# Model Order Reduction for Nonlinear Structural Dynamics

## *Simulation-free Approaches*

**Johannes Baptist Rutzmoser**

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

**Doktor-Ingenieurs (Dr.-Ing.)**

genehmigten Dissertation.

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

**Prüfer der Dissertation:**

1. Prof. dr. ir. Daniel Rixen
2. Ass. Prof. Dr. Joaquín Alberto Hernández Ortega

Die Dissertation wurde am 19.09.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 12.02.2018 angenommen.

# Zusammenfassung

Im Zuge der kontinuierlichen Verbesserung von dynamisch belasteten Bauteilen und Systemen sind effiziente Simulationsmethoden für strukturdynamische Fragestellungen von hoher Bedeutung. Insbesondere die Anforderung von geringem Gewicht bei zugleich hoher Festigkeit und Lebensdauer erfordert häufig Lösungen, in denen Phänomene wie geometrische Nichtlinearität, die durch große Rotationen verursacht wird, eine oft dominierende Rolle spielen. Da die Berücksichtigung von Nichtlinearitäten in Simulationsmodellen jedoch einen erheblich gesteigerten Rechenaufwand nach sich zieht, ist die Modellreduktion, die Approximation von rechenaufwändigen Modellen durch deutlich reduzierte Modelle, eine attraktive Option.

Die meisten der etablierten nichtlineare Reduktionsverfahren basieren auf der Analyse von sogenannten Trainings-Simulationen des nicht reduzierten, mit sehr großem Rechenaufwand assoziierten Modells. Diese Herangehensweise steht jedoch in einem gewissen Widerspruch zur Idee von Modellreduktion, die genau die Reduktion des Modells anstrebt. Daher werden in der vorliegenden Arbeit sogenannte simulationsfreie Methoden diskutiert und entwickelt, die nicht oder so wenig wie möglich auf Trainings-Simulationen basieren.

Die effiziente Reduktion von geometrisch nichtlinearen Modellen erfordert die Lösung zweier unterschiedlicher Probleme. Das erste Problem ist die Bestimmung einer Reduktionsbasis, die die Parametrisierung des kinematisch zulässigen Bewegungsraumes darstellt. Im Rahmen dieser Arbeit werden verschiedene Verfahren vorgestellt und untersucht, die etablierte Reduktionsverfahren für lineare Systeme für geometrisch nichtlineare Systeme mit Hilfe von Perturbationsansätzen erweitern. Zusätzlich werden weiterführende Verfahren wie eine nichtlineare Projektion auf eine quadratische Mannigfaltigkeit sowie eine Vergleichsmethode für Projektionsbasen vorgestellt und diskutiert.

Das zweite Problem für die effiziente Reduktion von geometrisch nichtlinearen strukturdynamischen Systemen ist die Approximation der Nichtlinearität, genannt Hyper-Reduktion. Ein Verfahren, das neben anderen im Rahmen dieser Arbeit diskutiert wird, ist die ECSW-Methode, die die Anzahl der Elemente im Netz des reduzierten Modells verringert. Für diese Methode werden jedoch Training-Sets benötigt, die bisher durch eine Simulation des vollen, unreduzierten Modells generiert werden. In dieser Arbeit werden zwei Ansätze vorgestellt, die eine deutlich günstigere Generierung von Training-Sets ermöglichen. Des Weiteren werden Hyper-Reduktionsmethoden basierend auf der polynomialen Struktur der Nichtlinearität sowie basierend auf einem Kollokationsansatz diskutiert und auf die Brauchbarkeit in strukturdynamischen Anwendungen untersucht.

Abschließend wird die Leistungsfähigkeit und Praktikabilität der vorgeschlagenen Methoden anhand eines industriellen Modells einer LKW-Blattfeder demonstriert.

# Abstract

Due to the continuous improvement of mechanical parts and systems in dynamic applications, efficient simulation methods for structural dynamics problems are of high importance. Especially the requirement of low weight and at the same time high strength and durability leads to designs where geometrical nonlinearity caused by large rotations is often dominating. However, since considering nonlinearities leads to significantly higher computational costs, model order reduction, the approximation of large models with significantly smaller ones, is an attractive option.

Most of the established nonlinear reduction methods are based on the analysis of so-called training simulations of the full, unreduced model which is associated with high computational costs. This approach, however, to a certain extent contradicts the idea of model order reduction, which attempts to reduce the computational effort. Hence, the following thesis discusses and develops so-called simulation-free methods which are not, or as little as possible, based on training simulations.

The efficient reduction of geometrically nonlinear models requires the solution of two distinct problems. The first problem is the computation of the reduced basis which forms the parametrization of the kinematically admissible space for the displacements. In this work, different methods are presented and investigated which extend the established reduction techniques for linear systems to geometrically nonlinear systems by using perturbation approaches. In addition, further methods like a nonlinear projection on a quadratic manifold and a comparison technique for reduced bases are introduced and discussed.

The second problem for the efficient reduction of geometrically nonlinear structural dynamics systems is the approximation of the nonlinear terms, which is referred to as Hyper-Reduction. One approach, which is discussed amongst others, is the ECSW method which reduces the number of elements in the mesh of the reduced model. This technique, however, requires training sets that are hitherto generated using the full, unreduced model. In this thesis, two approaches are introduced which allow for a clearly cheaper generation of the training sets. Furthermore, approaches exploiting the polynomial structure of the nonlinearity or using a collocation technique are discussed and investigated for their suitability in the context of structural dynamics.

Finally, the performance and practicability of the proposed methods are demonstrated on an industrial model of a truck leaf spring.

# Danksagung

Die vorliegende Arbeit entstand während meiner Zeit von 2011 bis 2017 als wissenschaftlicher Mitarbeiter am Lehrstuhl für Angewandte Mechanik der Technischen Universität München. Den vielen Personen, die mich während dieser Zeit unterstützt und gefördert haben, möchte ich meinen ganz herzlichen Dank aussprechen.

Mein erster Dank gilt meinem Doktorvater Prof. dr. ir. Daniel Rixen für das in mich gesetzte Vertrauen, die wohlwollende Begleitung und Unterstützung. Er gab mir die Möglichkeit, an einem spannenden und vielseitigen Thema zu arbeiten und stand mit Rat und Tat zur Seite. Sowohl die intensive Betreuung und die fachliche Unterstützung, als auch die großen Freiräume waren und sind für mich von unschätzbarem Wert.

Des Weiteren möchte ich mich bei Prof. Dr.-Ing. Michael W. Gee für die Übernahme des Vorsitzes der Prüfungskommission sowie bei Prof. Dr. Joaquín Alberto Hernández Ortega für das Interesse an meiner Arbeit und die Übernahme des Zweitgutachtens bedanken.

Die Zeit am Lehrstuhl wird mir vor allem auch wegen der Kollegen in bester Erinnerung bleiben. Ihnen danke ich für die offene und hilfsbereite Atmosphäre, den spannenden Austausch und die zahlreichen und regen fachlichen und nicht-fachlichen Diskussionen. Insbesondere möchte ich mich bei Andreas Bartl, Bastian Eselfeld, Alexander Ewald, Fabian Gruber und Christian Meyer für die kollegiale und intensive Zusammenarbeit in Lehre und Forschung herzlich bedanken.

Auch bei den Studenten möchte ich mich bedanken, die in Form von Studienarbeiten einen wertvollen Beitrag zum Gelingen dieser Arbeit beigetragen haben. Nicht nur durch die fleißige Arbeit, sondern auch durch die Diskussionen bei der Betreuung gab es stets neue Denkanstöße. Besonders danken möchte ich dabei Amandine Desjardins, Gabriel Gruber, Anton Mayr, Sebastian Otten, Mikhail Pak, Daniel Scheffold und Xuwei Wu. Bedanken möchte ich mich auch bei Pascal Reuß, der mich dazu ermutigte, die entwickelten Methoden auf ein industrielles Beispiel anzuwenden. Christian Meyer und Daniel Scheffold gilt mein Dank für die wertvollen Hinweise bei der Durchsicht des Manuskripts.

Ganz herzlich möchte ich meiner Familie danken, die mich an so vielen Stellen gefördert und unterstützt hat, insbesondere meinen Eltern. Schade, dass mein Vater den Abschluss dieser Arbeit nicht mehr erleben durfte.

Nicht zuletzt gilt meine aufrichtigste Dankbarkeit meiner Frau Nathalie, die mich in allen Belangen in ihrer unnachahmlichen Art unterstützt und mir den Rücken frei gehalten hat. Ohne ihre Ausdauer, ihre Geduld und ihr Verständnis wäre diese Arbeit undenkbar.

Garching, im März 2018

# Table of Contents

# Chapter 1

# Introduction

The development process in structural dynamics engineering has dramatically changed in the past decades. With the numerical analysis tools available, the behavior of complex structures and systems can be predicted as efficiently and effectively as never before. However, despite the exponential growth of the hardware performance for over half a century, which has already been predicted by Gordon Moore [Moo65] back in 1965, the demand for faster and more accurate simulation tools is still unbroken.

To overcome the computational limitations set by the hardware development, better numerical algorithms are strongly required, which provide both high accuracy and moderate computational costs. One approach to achieve the contradicting goals of high accuracy and low computational costs is model order reduction. In this framework, one approximates a large model with a reduced model of less computational complexity. The idea is to invest computational effort, the offline cost, to reduce the large model and thus obtain the reduced model, as depicted in Figure 1.1. The online cost, i.e., the effort for the actual computation yielding the desired results, is then clearly smaller for the reduced model compared to the full model.

The advantages of the reduced online costs and hence of model order reduction itself are effective in several scenarios. In control engineering, for instance, little online costs are of high interest for real time controllers while the offline costs in the design process barely matter. In various design analyses in engineering, model order reduction speeds up the computation, if both online and offline costs to reduce and run the system are lower than the online costs of the full system. In the analysis of multiple load cases or in optimizations, reduction methods are often used when the offline costs of the reduction have to be invested only once and pay off with multiple online runs. Also in substruc-

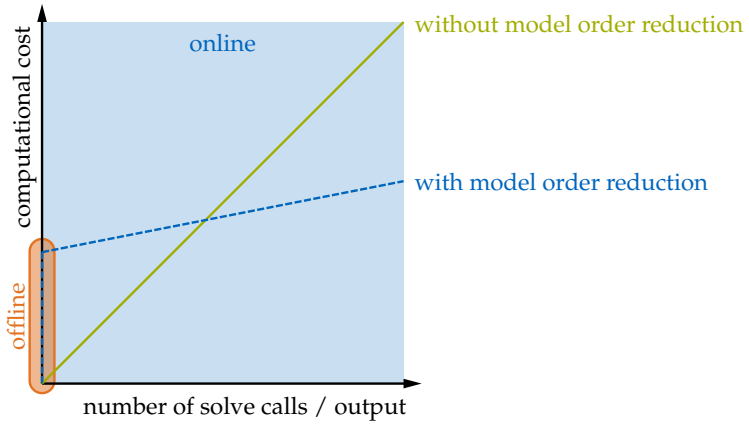


**Figure 1.1:** Offline and online costs in model order reduction.

**Figure 1.2:** Wing displacements on ground (top), flight with 1 g (middle) and maximum load (bottom) of the Boeing 787 Dreamliner. (source: [Dod].)

turing, where subcomponents are modeled and then assembled in a subsequent step, the offline costs usually pay off.

In structural dynamics, the large model is commonly constructed using the finite element method. It allows for the discretization of arbitrary geometries with excellent accuracy. However, this method is computationally too demanding for many nonlinear applications, especially when transient solutions over larger time spans are desired. Hence, model order reduction is mandatory to make the nonlinear dynamical systems available for broader analyses.

The need for simulating structural dynamics in the nonlinear regime is prominent, since lightweight structures with high stiffness to weight ratios are prevalent in modern engineering. They are sometimes operated in the nonlinear regime to achieve the goals of performance and efficiency. The maximum load case wing flex of the Boeing 787 Dreamliner depicted in Figure 1.2, for instance, is clearly geometrically nonlinear yielding large displacements and rotations which cannot be captured by linear structural dynamics theory. The analysis of geometrically nonlinear structures in the aerospace industry is crucial in many applications such as wing design and flutter analysis (cf. [LL04; XX15]). Also in the automotive industry, elastic parts are operated in the geometrically nonlinear regime like the leaf spring used in trucks (cf. [Sug+06; Kon+13]). The accurate simulation is crucial in the prediction of the vehicle dynamics, fatigue and also relevant for comfort including noise and vibrations. In Chapter 15, a geometrically nonlinear leaf spring is chosen as industrial real-life example to prove the applicability of the proposed methods. Among further industries where geometric nonlinearities play a crucial role are manufacturers of microelectromechanical systems (cf. [You11; Che+04]) or wind turbines (cf. [BV10; LHB04]). Common to all applications is the presence of large displacements and rotations or the existence of cable or membrane-effects, which make linear analyses

insufficient and require geometrically nonlinear investigations.

## 1.1 Objective

Model order reduction is well established in linear structural dynamics. Various methods exist, which rely on intrinsic properties such as eigenmodes and hence are backed up by system theoretic properties. For nonlinear systems, however, these properties do either not exist or are very hard to compute, so that the linear methods are in general not successfully applicable to nonlinear systems. Thus, the state-of-the-art approach in nonlinear model order reduction is a detour over statistical methods, where so-called training snapshots computed using the full, unreduced system are used to build the reduced models. This approach, however, is to a certain extent contradictory to the concept of model order reduction, since the large, unreduced system needs to be solved first in order to reduce exactly this system. Depending on the system and the computational hardware available, these offline costs associated with the solution of the full, unreduced system are inconvenient in the best case. In the worst case, model order reduction with simulation-based methods requiring the large training snapshots is infeasible when the computational hardware for the full system is not available.

Simulation-free methods solve this issue by circumventing the necessity of full systems's training sets for generating nonlinear reduced models. They identify the nonlinear effects by other means like Taylor expansions or cheaper static training sets with pseudo-dynamic forces. Thereby, the task of model order reduction for nonlinear systems can be subdivided into the reduced basis problem and the hyper-reduction problem. In the former, the kinematic space of motion is reduced yielding a reduced set of generalized coordinates. The latter deals with the accelerated evaluation of the nonlinearities in the reduced equations of motion. Hence, *the objective of this thesis is to discuss already known and introduce new simulation-free methods for both reduced basis methods and hyper-reduction methods in the context of nonlinear structural dynamics*.

Not discussed are methods based on the proper generalized decomposition (PGD), which take a completely different approach. While the reduced basis methods have proven to be excellent procedures for the reduction of linear and nonlinear structural dynamical systems, it is still unclear if the PGD is applicable in this regime, see for instance Boucinha et al. [BGA13; Bou+14].

## 1.2 Scientific Contributions

This thesis gives an overview of simulation-free reduced basis and hyper-reduction methods. Some approaches presented are novel scientific contributions of the author:

- A nonlinear projection technique named quadratic manifold approach introduced in Chapter 7, of which parts are published in [RR14b; RRT14; Jai+17; Rut+17],
- a clarification and generalization of the static derivatives and how they fit into the framework of simulation-free nonlinear reduction, which is published in [Rut+17],
- a comparison technique of reduction bases using subspace angles discussed in Chapter 8, which is published in [RGR15],
- the alternative computation of the polynomial tensors used for polynomial tensors hyper-reduction based on numerical differentiation as well as the speedup technique to exploit the symmetry of the polynomial tensors both addressed in Chapter 11, and

- the generation of almost simulation-free training snapshots named NSKTS and NSMTS for the ECSW hyper-reduction proposed in Chapter 13, published in [RR17].

Furthermore, numerical experiments for the relevant methods are reported. Some of them reveal novel results, which are discussed in the respective chapter. To enable the application of the advanced reduction techniques to academic and industrial examples, a non-linear structural dynamics research code named AMFE, based on finite elements and written in Python and Fortran, was entirely developed by the author in the context of this thesis.

## 1.3 Outline

After this introductory chapter, the fundamentals of the finite element method are introduced in Chapter 2 with special focus on geometrical nonlinearity. Closely related is Chapter A in the Appendix, where the nonlinear finite element formulation is covered in detail together with the structure of the code AMFE.

Then, the thesis is subdivided in three parts. Part I is devoted to the reduced basis approach which is introduced in Chapter 3. In the following four chapters, different approaches to the construction of the reduced basis are discussed. Chapter 4 deals with the reduction of linear structural dynamics systems. The two main methods proposed are embedded in the substructuring context, which is very common in linear model order reduction, as well as in the system theoretics context, which provides more underlying theory.

The reduction of nonlinear systems is addressed by the three subsequent chapters. Chapter 5 introduces the Proper Orthogonal Decomposition (POD), the state-of-the-art method for the reduction of nonlinear systems, which is simulation-based, though. Thus, Chapter 6 investigates the simulation-free reduced basis methods. These methods extend the linear reduction methods introduced in Chapter 4 with static and modal derivatives, making them suitable for geometrically nonlinear systems. Several variants of the static and modal derivative approach are discussed as well as techniques for selecting specific basis vectors. In numerical experiments, these procedures are applied to academic examples revealing certain novel patterns of the methods.

A fundamentally different approach for projective model order reduction is taken in Chapter 7, where a nonlinear mapping technique is introduced. In this simulation-free method, the projectional subspace is constantly altered based on a quadratic manifold which is constructed using basis vectors and quadratic expansion vectors. These are either chosen as modal derivatives or static derivatives which result from the so-called force compensation approach. This technique provides a minimal number of degrees of freedom (dofs), however it is limited to special cases. For deeper insights, the method is discussed in the context of von Karman kinematics, where it evolves in a nonlinear Guyan reduction scheme. A broad numerical investigation is conducted in the applications section. It unveils the pattern, that the quadratic manifold approach is successfully applicable to structures where the nonlinearity is associated with small rotations, as it is the case for membrane-like structures.

Chapter 8 addresses the comparison of bases using subspace angles and principal vectors. They form a powerful and insightful tool for comparing reduced bases, which is exemplarily applied to one of the examples used throughout this thesis. In this numerical experiment it is shown that the linearization of rotations is the main reason for the failure of linear reduction methods in the context of geometrically nonlinear finite element systems. Part I is concluded with a brief summary given in Chapter 9.

Part II is devoted to the hyper-reduction problem emerging from the reduced basis reduction of nonlinear systems. In Chapter 10, the hyper-reduction problem is explained,

which is then addressed in the following three chapters. Chapter 11 deals with the polynomial tensors hyper-reduction. This method is based on the fact that the internal nonlinear forces are third order polynomials for systems modeled with St. Venant-Kirchhoff material. To obtain the polynomial coefficients, different non-intrusive methods are investigated. Furthermore, an approach for exploiting the symmetry of the polynomial coefficient arrays is proposed for both efficient storing and efficient multiplication.

A different hyper-reduction approach named Discrete Empirical Interpolation Method (DEIM) is discussed in Chapter 12. It approximates the nonlinear internal forces in the unreduced domain using an empirically computed force basis and a collocation scheme, which however breaks the symmetry and with it the stability of the reduced system. Various improvements of the method are discussed, which however cannot alleviate the fundamental flaw of lost symmetry cast into the method. A brief empirical study investigates why the DEIM fails for what concerns both accuracy and stability when applied in the geometrically nonlinear finite element context.

Chapter 13 is committed to the Energy Conserving Mesh Sampling and Weighting (ECSW) hyper-reduction method. It is similar to the DEIM, as it also relies on a reduced evaluation of the internal forces, but resolves the fundamental flaws of the DEIM allowing for the construction of accurate and stable hyper-reduced models. The main issue of the method in the context of simulation-free reduced bases is the need for training snapshots, which are commonly obtained from a training simulation of the full, unreduced system. In order to make the ECSW applicable in a simulation-free context and fill the gap between simulation-free reduced bases and the ECSW hyper-reduction, the novel Nonlinear Stochastic Krylov Training Sets (NSKTS) and the Nonlinear Stochastic Modal Training Sets (NSMTS) are proposed. In a detailed study, the applicability and robustness of these lean training sets is demonstrated. In Chapter 14, a concise summary of Part II ist given.

Part III of this thesis concludes the two previous parts. In Chapter 15, the simulation-free framework of building a reduced basis and applying a hyper-reduction is demonstrated on a real-life example of a leaf spring of a truck. The applicability, accuracy and great speedup factors are confirmed in a thorough investigation. Chapter 16 summarizes the thesis with a conclusion of the main results and with topics suitable for future research.

## 1.4 Remarks on Notation

Throughout this thesis, the symbols are mostly used in a consistent manner. If not explicitly denoted otherwise, non-boldface symbols refer to scalar values, lowercase boldface symbols are column vectors and uppercase boldface symbols are matrices. Calligraphic letters, e.g., $\mathcal{E}$, refer to sets.

All numerical experiments are conducted with AMFE, the Python module written in the context of this thesis. Hence, the indexing and slicing convention follows the zero-based style of Python, i.e., $\boldsymbol{a}[0]$ selects the first element of vector $\boldsymbol{a}$, $\boldsymbol{a}[:5]$ the first five elements of $\boldsymbol{a}$ and $\boldsymbol{A}[:, :3]$ the first three columns of the matrix $\boldsymbol{A}$. Furthermore, in the ranges used in for-loops of algorithms, the upper value is not included in the range, i.e., **for** $i := 0 : 10$ yields ten ascending values for $i$ in every iteration with the first being zero and the last being nine.

# Chapter 2

# Nonlinear Finite Elements

The key technology in structural dynamics is the finite element method, which allows for the discretization of the continuous elasto-dynamic problem. Since the method is so widespread and successful, it has an enormous body of literature. In this section, only the basic concepts are briefly introduced, which are necessary for the understanding of the reduction methods in the following chapters. For introductory textbooks on the topic, the reader is referred to, e.g., Bonet and Wood [BW97], Belytschko et al. [BLM00], Reddy [Red04], Simo and Hughes [SH06] or Crisfield et al. [C+12].

Different textbooks come with different notations. Throughout this thesis, the notation is kept close to Belytschko et al. [BLM00] using the Total Lagrangian framework. Several modifications are done in order to keep the notation consistent with the software AMFE, which is discussed in Chapter A in the Appendix.

## 2.1 Continuum Mechanics

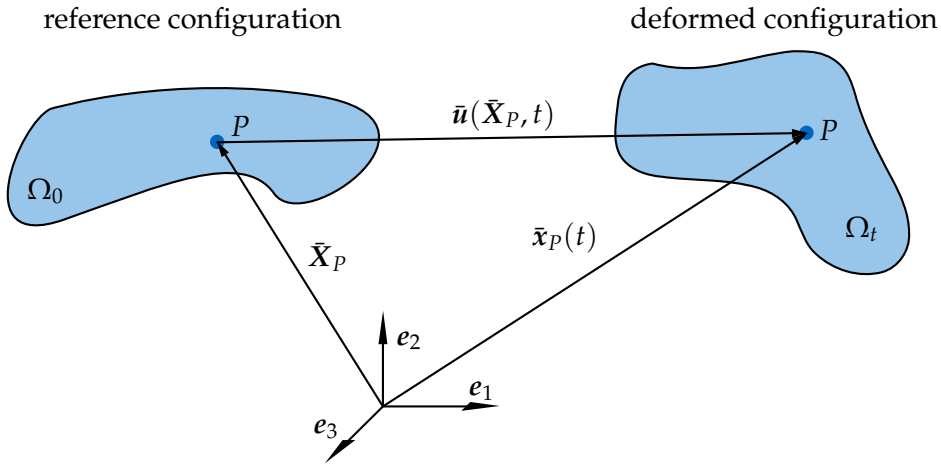


**Figure 2.1:** Reference configuration $\Omega_0$ and deformed configuration $\Omega_t$ at time $t$ of an elastic deformable continuous body.

First, the kinematics of a deformable continuum three dimensional body depicted in Figure 2.1 is discussed, although the derivation is also valid for two dimensions. The body has a reference configuration $\Omega_0$ at $t = 0$ and deforms over time. At time $t$ the body occupies the configuration $\Omega_t$. All material points of the body in the undeformed configuration $\Omega_0$ are described by the vector $\bar{\boldsymbol{X}} \in \mathbb{R}^3$ and all points of the deformed body at time $t$ are described by the vector $\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}}(\bar{\boldsymbol{X}}, t) \in \mathbb{R}^3$. Quantities referring to the reference

domain are expressed with uppercase letters, whereas quantities in the deformed domain are expressed with lowercase letters. The bar over the letters $\boldsymbol{X}$ and $\boldsymbol{x}$ denotes that these variables are continuous in space in contrary to the nodal quantities of discretized finite elements, see Section 2.3. The displacement of a particle is then expressed by the displacement vector $\bar{\boldsymbol{u}}$ describing the position of the deformed configuration relative to the initial configuration:

$$\bar{\boldsymbol{u}}(\bar{\boldsymbol{X}}, t) = \bar{\boldsymbol{x}}(\bar{\boldsymbol{X}}, t) - \bar{\boldsymbol{X}}. \tag{2.1}$$

The deformation is measured with the *deformation gradient* $\boldsymbol{F} \in \mathbb{R}^{3\times3}$. It is the partial derivative of the deformed configuration with respect to the initial configuration and can also be expressed in terms of the displacement vector $\bar{\boldsymbol{u}}$ using the identity matrix $\boldsymbol{I} \in \mathbb{R}^3$:

$$\boldsymbol{F} = \frac{\partial \bar{\boldsymbol{x}}}{\partial \bar{\boldsymbol{X}}} = \frac{\partial \bar{\boldsymbol{u}}}{\partial \bar{\boldsymbol{X}}} + \boldsymbol{I}. \tag{2.2}$$

The deformation gradient describes the mapping of an infinitesimal vector $\mathrm{d}\bar{\boldsymbol{X}}$ from the reference domain to the current domain $\mathrm{d}\bar{\boldsymbol{x}} = \boldsymbol{F}\mathrm{d}\bar{\boldsymbol{X}}$, which is also denoted the push forward operation. Since the mapping is bijective, $\mathrm{d}\bar{\boldsymbol{x}}$ can also be uniquely mapped to $\mathrm{d}\bar{\boldsymbol{X}}$ with $\mathrm{d}\bar{\boldsymbol{X}} = \boldsymbol{F}^{-1}\mathrm{d}\bar{\boldsymbol{x}}$ in the pull back operation. The deformation gradient $\boldsymbol{F}$, which is not symmetric in general, accounts for both stretching and rotation.

For describing strains in a large deformation context, a strain measure should not feature strains for pure rigid body rotations, but should be *rotation-invariant* or *objective*. Before introducing an objective strain measure, the deformation gradient $\boldsymbol{F}$ is investigated further. The mapping of $\boldsymbol{F}$ can be decomposed mathematically with the singular value decomposition (SVD) to

$$\boldsymbol{F} = \boldsymbol{U}_{\mathrm{svd}}\boldsymbol{\Sigma}_{\mathrm{svd}}\boldsymbol{V}_{\mathrm{svd}}^T, \tag{2.3}$$

where $\boldsymbol{U}_{\mathrm{svd}} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{V}_{\mathrm{svd}} \in \mathbb{R}^{3\times3}$ are orthogonal matrices and $\boldsymbol{\Sigma}_{\mathrm{svd}} \in \mathbb{R}^{3\times3}$ is a diagonal matrix formed by the singular values. In the geometric interpretation of the SVD, the orthogonal matrices represent rotation operators in the 3D space, while the diagonal matrix represents a stretching along the main axes. Hence, the mapping $\boldsymbol{F}$ from the reference configuration to the deformed configuration can be split into a rotation performed by $\boldsymbol{V}_{\mathrm{svd}}^T$ followed by a stretch along the main axes performed by $\boldsymbol{\Sigma}_{\mathrm{svd}}$ followed by another rotation performed by $\boldsymbol{U}_{\mathrm{svd}}$. Since $\boldsymbol{\Sigma}_{\mathrm{svd}}$ is a diagonal matrix, the stretch operation is performed along the main axes of the intermediate coordinate system, into which the rotation $\boldsymbol{V}_{\mathrm{svd}}^T$ was transforming. The stretch operation performed by $\boldsymbol{\Sigma}_{\mathrm{svd}}$ can also be expressed as a stretch operation not along the principal axes but along axes different from them, leading to the symmetric material stretch tensor $\boldsymbol{U}$:

$$\boldsymbol{U} = \boldsymbol{V}_{\mathrm{svd}}\boldsymbol{\Sigma}_{\mathrm{svd}}\boldsymbol{V}_{\mathrm{svd}}^T. \tag{2.4}$$

If $\boldsymbol{F}$ is to be expressed as a stretch operation with the material stretch operator $\boldsymbol{U}$, the rotation

$$\boldsymbol{R} = \boldsymbol{U}_{\mathrm{svd}}\boldsymbol{V}_{\mathrm{svd}}^T \tag{2.5}$$

completes the mapping of $\boldsymbol{F}$ since $\boldsymbol{V}_{\mathrm{svd}}^T\boldsymbol{V}_{\mathrm{svd}} = \boldsymbol{I}$, leading to the polar decomposition of the deformation gradient $\boldsymbol{F}$:

$$\boldsymbol{F} = \boldsymbol{R}\boldsymbol{U}. \tag{2.6}$$

The orthogonal matrix $\boldsymbol{R}$ represents the rotation, which is performed after the material stretching performed by the symmetric tensor $\boldsymbol{U}$.

Since the mapping of $\boldsymbol{F}$ can be decomposed into a rotational part and a deformation part, a strain measure can be defined allowing for cancelling out the rotational part in order to make the strain measure *objective*. The strain measure used in the Total Lagrangian framework is the *Green-Lagrange* strain $\boldsymbol{E} \in \mathbb{R}^{3\times3}$, which is defined as

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}). \tag{2.7}$$

The Green-Lagrange strain $\boldsymbol{E}$ is rotation-invariant, since $\boldsymbol{R}$ is orthogonal yielding $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$:

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}) = \frac{1}{2}(\boldsymbol{U}^T\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{U} - \boldsymbol{I}) = \frac{1}{2}(\boldsymbol{U}^T\boldsymbol{U} - \boldsymbol{I}). \tag{2.8}$$

The Green-Lagrange strain tensor $\boldsymbol{E}$ is symmetric by construction and ignores the rigid body motion of a body. Other objective strain definitions exist, which are not discussed here. For further readings the relevant textbooks, e.g., Bonet and Wood [BW97], are recommended.

The interpretation of the Green-Lagrange strain is not as straightforward as for the deformation gradient. Whereas the deformation gradient $\boldsymbol{F}$ describes the change of an infinitesimal element $\mathrm{d}\bar{\boldsymbol{X}}$, the Green-Lagrange strain describes the change of their squares:

$$\frac{1}{2}\left(\mathrm{d}\bar{\boldsymbol{x}}^T\mathrm{d}\bar{\boldsymbol{x}} - \mathrm{d}\bar{\boldsymbol{X}}^T\mathrm{d}\bar{\boldsymbol{X}}\right) = \mathrm{d}\bar{\boldsymbol{X}}^T\boldsymbol{E}\,\mathrm{d}\bar{\boldsymbol{X}}. \tag{2.9}$$

It can be interpreted as the quadratic change measure of the length of the infinitesimal element $\mathrm{d}\bar{\boldsymbol{X}}$.

Complementary to a strain measure, a corresponding stress measure is necessary to form an energy or work-conjugate pair allowing the application of energy or work principles. The Green-Lagrange strain tensor is given in the reference domain. The true Cauchy stress $\boldsymbol{\sigma}$, however, is defined in the deformed or material domain, as it returns the traction vector $\boldsymbol{t}$ for a given normal vector $\boldsymbol{n}$ of the cutting face:

$$\boldsymbol{t} = \boldsymbol{\sigma}\boldsymbol{n}. \tag{2.10}$$

To make the Cauchy stress $\boldsymbol{\sigma}$ work-conjugate to the Green-Lagrange strain $\boldsymbol{E}$, it needs to be pulled back leading to the *second Piola-Kirchhoff stress tensor* $\boldsymbol{S}$, which is also symmetric by construction:

$$\boldsymbol{S} = \boldsymbol{F}^{-1}\boldsymbol{\sigma}\boldsymbol{F}^{-T}\det(\boldsymbol{F}). \tag{2.11}$$

In comparison to the *true* Cauchy stress $\sigma$, the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$ does not have an intuitive interpretation. However, it can be regarded as the stress expressed in the reference domain leading to the work-conjugate associated to the Green-Lagrange strain tensor $\boldsymbol{E}$. Hence, the internal virtual work done by $\boldsymbol{S}$ in a body filling the domain $\Omega_0$ in the reference domain is

$$\delta W_{int} = \int_{\Omega_0} \boldsymbol{S} : \delta\boldsymbol{E}\,\mathrm{d}\Omega_0 \tag{2.12}$$

with the variation of the strain $\delta\boldsymbol{E}$. Since $\boldsymbol{E}$ is computed with the deformation gradient $\boldsymbol{F}$ (cf. (2.7)), the variation of the Green-Lagrange strain tensor is given as

$$\delta\boldsymbol{E} = \frac{1}{2}(\delta\boldsymbol{F}^T\boldsymbol{F} + \boldsymbol{F}^T\delta\boldsymbol{F}). \tag{2.13}$$

The relation between strain and stress is defined by the constitutive law expressing the elastic behavior of the material. In the following section, the modeling of nonlinear materials is discussed.

## 2.2 Nonlinear Material

The relation between stress and strain is determined by the material. If the stress tensor is solely a function of the strain and not dependent on the deformation history or the rate of deformation, the material is denoted as *hyperelastic*. It is characterized uniquely by the so-called energy density function $W(\boldsymbol{E})$, which describes the stored potential energy in the material due to the applied Green-Lagrange strain $\boldsymbol{E}$. The stress strain relation is then given as

$$\boldsymbol{S} = \frac{\partial W(\boldsymbol{E})}{\partial \boldsymbol{E}}, \tag{2.14}$$

i.e., the partial derivative of the energy density function with respect to the Green-Lagrange strain yields the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$.

The hyperelastic material mimicking the linear-elastic behavior as known in linear elasticity is the *St. Venant-Kirchhoff material*. It has the quadratic energy density function

$$W_{SV-K} = \frac{\lambda}{2}\text{trace}(\boldsymbol{E})^2 + \mu \boldsymbol{E} : \boldsymbol{E}, \tag{2.15}$$

where $\lambda$ and $\mu$ are the so-called Lamé constants, which are related to the Young's modulus $E$ and Poisson's ratio $\nu$ known from linear elastic materials:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)}. \tag{2.16}$$

The relationship between the second Piola-Kirchhoff stress and the Green-Lagrange strain in the St. Venant-Kirchhoff material is linear, since $W_{SV-K}$ is quadratic and differentiated once with respect to $\boldsymbol{E}$ (cf. (2.14)). Hence, the St. Venant-Kirchhoff material can be interpreted as the hyperelastic representation of the linear material behavior. Though it does not represent common materials for large strains, it is an often used material for geometric nonlinearities, when the strains remain small and in the linear-elastic range, as it often appears for instance in metallic structures.

For other materials exposing hyperelastic behavior like rubber, more advanced energy density functions are employed. For example, the compressible Neo-Hookean material is characterized by

$$W_{NH} = \frac{\mu}{2}(\text{trace}(\boldsymbol{E})) - \mu \ln(J) + \frac{\lambda}{2}(\ln(J))^2, \tag{2.17}$$

leading to a nonlinear strain-stress relationship. $J$ is the determinant of the deformation gradient and thus a measure of the material compression. Many more elaborate material laws exist, like the Mooney-Rivlin or the Ogden material, which can be used for the approximation of the nonlinear material behavior of rubber materials or organic tissues. For further information on nonlinear materials, the reader is referred to the textbooks of the field, e.g., Holzapfel [Hol00], Kim and Sankar [KS09] or Ogden [Ogd97].

## 2.3 Finite Element Discretization

Having defined the kinematics of the continuous deformable body and the constitutive law, the equations of motion can be derived with a variational principle. In mechanics, the principle of virtual work extended to dynamics by d'Alembert's principle yields the weak form of the equations of motion:

$$\delta W = \underbrace{\int_{\Omega_0} \delta\bar{\boldsymbol{u}}^T \rho_0 \ddot{\bar{\boldsymbol{u}}} \, \mathrm{d}\Omega_0}_{\delta W_{kin}} + \underbrace{\int_{\Omega_0} \boldsymbol{S} : \delta\boldsymbol{E} \, \mathrm{d}\Omega_0}_{\delta W_{int}} - \underbrace{\int_{\partial\Omega_0} \delta\bar{\boldsymbol{u}}^T \boldsymbol{t} \, \mathrm{d}\partial\Omega_0 - \int_{\Omega_0} \delta\bar{\boldsymbol{u}}^T \boldsymbol{b} \, \mathrm{d}\Omega_0}_{\delta W_{ext}} = 0 \tag{2.18}$$

with the kinematically admissible variation of the displacements $\delta\bar{\boldsymbol{u}}$, the accelerations of the particles $\ddot{\bar{\boldsymbol{u}}}$ being the second time derivative of the displacements $\bar{\boldsymbol{u}}$ defined in (2.1), the density $\rho$ of the body, the traction $\boldsymbol{t}$ at the boundary of the body $\partial\Omega_0$ and the external forces $\boldsymbol{b}$ acting in the body domain $\Omega_0$.

### 2.3.1 Approximation with Shape Functions

The equation (2.18) above can be solved analytically only for special domains $\boldsymbol{\Omega}_0$, tractions $\boldsymbol{t}$ and volume loads $\boldsymbol{b}$. To solve the equation in a systematic and approximate manner, the continuous space can be discretized using the concept proposed by Ritz [Rit09]. There, the key idea is the expression of the field variables $\bar{\boldsymbol{u}}$ and its derived quantities like $\boldsymbol{E}$ and $\boldsymbol{S}$ as a sum of a number of trial or shape functions $\boldsymbol{N}(\bar{\boldsymbol{X}})$ and their corresponding amplitudes $\hat{\boldsymbol{u}}_e(t)$. Hence the displacements are written as

$$\bar{\boldsymbol{u}}(\boldsymbol{X}, t) = \sum_{i=1}^{n_n} N_i(\boldsymbol{X}) \hat{\boldsymbol{u}}_{e,i}(t), \tag{2.19}$$

where the a priori defined shape functions $N_i(\boldsymbol{X})$ are purely space dependent and the amplitudes $\hat{\boldsymbol{u}}_{e,i}(t)$ are purely time dependent and are the variables which are generally sought in the solution procedure.

In the finite element framework, the shape functions are defined and evaluated on subdomains $\Omega_{0,e}$ dividing the global domain $\Omega_0$ into a set $\mathcal{E}$ of geometrical primitives:

$$\Omega_0 \approx \bigcup_{e \in \mathcal{E}} \Omega_{0,e}. \tag{2.20}$$

The primitives are referred to as *finite elements* responsible for the method's name. They are commonly tetrahedrons or hexahedrons in $\mathbb{R}^3$ or triangles or quadrangles in $\mathbb{R}^2$. For every element, a set of low order polynomial functions $N_{e,i}$ is typically used as shape function set which is defined in a local element coordinate system. For a Hexahedron element, for instance, the local coordinates are $\boldsymbol{\xi} = (\xi, \eta, \zeta)^T$, which are defined in $[-1, 1]$ for every component.

The shape functions are usually chosen as Lagrange polynomials, since they can be hinged on nodes like corners or center points of edges of the geometric primitive. With this choice, all discrete nodal values are interpolated by the Lagrangian polynomial shape functions leading to the isoparametric concept. Hence, all field variables including the reference configuration are expressed by the shape functions $\boldsymbol{N}(\boldsymbol{\xi})$ and the nodal coordinates $\hat{\boldsymbol{X}}_e$, $\hat{\boldsymbol{x}}_e$ or $\hat{\boldsymbol{u}}_e$:

$$\bar{\boldsymbol{X}}(\boldsymbol{\xi}) = \sum_{i=1}^{n_n} N_i(\boldsymbol{\xi}) \hat{\boldsymbol{X}}_{e,i} = \hat{\boldsymbol{X}}_e^T \boldsymbol{N}(\boldsymbol{\xi}), \tag{2.21}$$

$$\bar{\boldsymbol{x}}(\boldsymbol{\xi}) = \sum_{i=1}^{n_n} N_i(\boldsymbol{\xi}) \hat{\boldsymbol{x}}_{e,i} = \hat{\boldsymbol{x}}_e^T \boldsymbol{N}(\boldsymbol{\xi}), \tag{2.22}$$

$$\bar{\boldsymbol{u}}(\boldsymbol{\xi}) = \sum_{i=1}^{n_n} N_i(\boldsymbol{\xi}) \hat{\boldsymbol{u}}_{e,i} = \hat{\boldsymbol{u}}_e^T \boldsymbol{N}(\boldsymbol{\xi}). \tag{2.23}$$

The vector $\boldsymbol{N}(\boldsymbol{\xi}) \in \mathbb{R}^{n_n}$ contains the shape functions defined for the element $e$ with $n_n$ being the number of nodes of the element, $\hat{\boldsymbol{X}}_e \in \mathbb{R}^{n_n \times 3}$ is the matrix of the nodal

coordinates in the reference configuration, $\hat{\boldsymbol{x}}_e \in \mathbb{R}^{n_n \times 3}$ the matrix of the nodal coordinates in the deformed configuration and $\hat{\boldsymbol{u}}_e \in \mathbb{R}^{n_n \times 3}$ is the matrix of the nodal displacements. For simplified notation, the displacement matrix $\hat{\boldsymbol{u}}_e$ is commonly reshaped to the column vector $\boldsymbol{u}_e \in \mathbb{R}^{3 \cdot n_n}$ describing the nodal displacements of an element. In the remainder of this thesis, all displacements denoted with a plain $\boldsymbol{u}_e$ for an element or $\boldsymbol{u}$ for a full system are given in this column vector notation. For more details, see Section A.2 in the Appendix.

To ensure $C^0$ continuity of the field variables across the boundary of elements, the nodal variables of neighboring elements have to be shared. This is implicitly accounted for by applying the assembly process discussed in Subsection 2.3.2, since local variables belonging to multiple elements are picked from a unique set of global variables.

With the kinematic approximations (2.21) to (2.23), all derived field tensors can be computed accordingly (cf. Section A.2). The space derivatives like the deformation gradient $\boldsymbol{F}$ are carried out on the shape functions in the element coordinates $\boldsymbol{\xi}$. The mapping between the element coordinates and the reference domain is performed using the element Jacobian $\boldsymbol{J} = \partial \bar{\boldsymbol{X}} / \partial \boldsymbol{\xi}$, yielding the finite element discretization of the deformation gradient $\boldsymbol{F}$:

$$\begin{aligned} \boldsymbol{F} =& \boldsymbol{I} + \frac{\partial \bar{\boldsymbol{u}}}{\partial \bar{\boldsymbol{X}}} = \boldsymbol{I} + \hat{\boldsymbol{u}}_e^T \frac{\partial \boldsymbol{N}}{\partial \bar{\boldsymbol{X}}} = \boldsymbol{I} + \hat{\boldsymbol{u}}_e^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \bar{\boldsymbol{X}}} = \boldsymbol{I} + \hat{\boldsymbol{u}}_e^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} \boldsymbol{J}^{-1} \\ =& \boldsymbol{I} + \hat{\boldsymbol{u}}_e^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} \left( \hat{\boldsymbol{X}}_e^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} \right)^{-1}. \end{aligned} \tag{2.24}$$

Obviously, the deformation gradient is a function of the nodal displacements $\hat{\boldsymbol{u}}_e$ and the initial configuration of the element expressed in the initial positions $\hat{\boldsymbol{X}}_e$. The spatial derivatives are carried out on the shape functions $\boldsymbol{N}$ defined for the element.

The virtual work of the inertia forces $\delta W_{kin,e}$ of element $e$ is obtained by applying the principle of virtual work using the discretizations (2.21) to (2.23):

$$\delta W_{kin,e} = \int_{\Omega_{0,e}} \delta \bar{\boldsymbol{u}}^T \rho_0 \ddot{\bar{\boldsymbol{u}}} \, \mathrm{d}\Omega_{0,e} = \int_{\Omega_{0,e}} \boldsymbol{N}^T \delta \hat{\boldsymbol{u}}_e \ddot{\hat{\boldsymbol{u}}}_e^T \boldsymbol{N} \rho_0 \, \mathrm{d}\Omega_{0,e}. \tag{2.25}$$

The last term leads to the element mass matrix $\boldsymbol{M}_e$ of element $e$ by evaluating the integral and expressing the quantities in matrix notation:

$$\delta W_{kin,e} = \delta \boldsymbol{u}_e^T \boldsymbol{M}_e \ddot{\boldsymbol{u}}_e. \tag{2.26}$$

The internal virtual work of element $e$ can be expressed similarly leading to the internal force vector $\boldsymbol{f}_e$ of element $e$:

$$\delta W_{int,e} = \int_{\Omega_{0,e}} \boldsymbol{S} : \delta \boldsymbol{E} \, \mathrm{d}\Omega_{0,e} = \delta \boldsymbol{u}_e^T \boldsymbol{f}_e(\boldsymbol{u}_e). \tag{2.27}$$

The internal force vector $\boldsymbol{f}_e$ is a nonlinear function of the elemental displacements $\boldsymbol{u}_e$. For the computations of $\boldsymbol{M}_e$ and $\boldsymbol{f}_e$, the evaluation of the integrals is commonly performed with a quadrature rule. Thereby, the continuous integral is substituted by a weighted sum of $m$ discrete values of the integrand at distinct *integration points* or *Gauss points* $\boldsymbol{\xi}_i$ with corresponding weights $w_i$. Exemplarily, the quadrature of the integrand $\boldsymbol{i}(\xi)$ for a one-dimensional integral in the domain of $\xi_i \in [-1, 1]$ is written as:

$$\int_{-1}^{1} \boldsymbol{i}(\xi) \, \mathrm{d}\xi \approx \sum_{i=1}^{m} w_i \boldsymbol{i}(\xi = \xi_i). \tag{2.28}$$

The proper choice of geometric primitive, shape functions and Gauss points leads to well-known element types. In the field of *Element Technology*, many combinations of

these three ingredients have been tried and studied to build elements showing good approximation properties while keeping the computational costs low. For further information, the interested reader is referred to the literature in this topic, e.g., Belytschko et al. [BLM00], Zienkiewicz and Taylor [ZT13] and for more fundamental studies Strang and Fix [SF73] and Zienkiwicz et al. [ZTZ13]. The formulation to derive and evaluate the field variables in (2.25) and (2.27) is given in the Appendix Section A.2.

### 2.3.2 Assembly

The virtual work $\delta W$ of the whole domain $\Omega_0$ is composed of the contributions of every element $e$ forming the domain. To compute a solution on the full domain, a global displacement vector $\boldsymbol{u} \in \mathbb{R}^N$ containing all nodal displacements is used. The nodal displacements $\boldsymbol{u}_e \in \mathbb{R}^{3n_n}$ of an element $e$ are then expressed by the global displacement vector $\boldsymbol{u}$ using the mapping provided by the Boolean localization matrix $\boldsymbol{L}_e \in \mathbb{R}^{3n_n \times N}$:

$$\boldsymbol{u}_e = \boldsymbol{L}_e \boldsymbol{u}. \tag{2.29}$$

The virtual work $\delta W_{kin}$ of the inertia forces of the whole domain $\Omega_0$ is then expressed as the sum of the virtual work of the inertia forces of all finite elements in the element set $\mathcal{E}$. With $\delta \boldsymbol{u}_e = \boldsymbol{L}_e \delta \boldsymbol{u}$ one obtains:

$$\delta W_{kin} = \sum_{e \in \mathcal{E}} \delta W_{kin,e} = \sum_{e \in \mathcal{E}} \delta \boldsymbol{u}^T \boldsymbol{L}_e^T \boldsymbol{M}_e \boldsymbol{L}_e \ddot{\boldsymbol{u}}, \tag{2.30}$$

leading to the assembled mass matrix $\boldsymbol{M}$:

$$\boldsymbol{M} = \sum_{e \in \mathcal{E}} \boldsymbol{L}_e^T \boldsymbol{M}_e \boldsymbol{L}_e. \tag{2.31}$$

Similarly, the internal virtual work $\delta W_{int}$ is assembled by the sum of all elemental contributions:

$$\delta W_{int} = \sum_{e \in \mathcal{E}} \delta W_{int,e} = \sum_{e \in \mathcal{E}} \delta \boldsymbol{u}^T \boldsymbol{L}_e^T \boldsymbol{f}_e(\boldsymbol{L}_e \boldsymbol{u}), \tag{2.32}$$

leading to the assembled nonlinear internal force vector $\boldsymbol{f}$:

$$\boldsymbol{f} = \sum_{e \in \mathcal{E}} \boldsymbol{L}_e^T \boldsymbol{f}_e(\boldsymbol{L}_e \boldsymbol{u}). \tag{2.33}$$

Also the tangential stiffness matrix $\boldsymbol{K}_e = \partial \boldsymbol{f}_e / \partial \boldsymbol{u}_e$ of element $e$ is assembled in the same way as the mass matrix:

$$\boldsymbol{K} = \sum_{e \in \mathcal{E}} \boldsymbol{L}_e^T \, \boldsymbol{K}_e(\boldsymbol{L}_e \boldsymbol{u}) \, \boldsymbol{L}_e. \tag{2.34}$$

The assembly is written here as products of sparse Boolean matrices with the elemental quantities. In efficient implementations, the assembly is performed using index-operations addressing the dofs in the global vectors and matrices.

### 2.3.3 Equations of Motion

The virtual work expressed in (2.18) yields with the finite element discretization above

$$\delta W = \delta \boldsymbol{u}^T (\boldsymbol{M} \ddot{\boldsymbol{u}} + \boldsymbol{f}(\boldsymbol{u}) - \boldsymbol{g}) = 0, \tag{2.35}$$

with the external generalized forces $\boldsymbol{g}$ stemming from the external forces $\boldsymbol{b}$ in the domain $\Omega_0$ and the traction forces $\boldsymbol{t}$ onto the boundary $\partial\Omega_0$. Since the variations $\delta \boldsymbol{u}$ are

arbitrary, the term in brackets has to be equal to zero leading to the semi-discretized equations of motion

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{g}. \tag{2.36}$$

With the finite element procedure, the spatial domain is discretized using the shape functions of the elements while the time domain is still in its continuous form. The equations of motion are nonlinear since the internal forces responsible for the elastic restoration of the initial position are nonlinear, whereas the inertia forces $\boldsymbol{M}\ddot{\boldsymbol{u}}$ are linear with respect to the accelerations. In the typical setup, the external forces $\boldsymbol{g}$ are time-dependent and sometimes also displacement-dependent, especially if follower-forces are considered.

If the displacements are small, the nonlinear forces $\boldsymbol{f}$ can be considered linear. They are then replaced by the linearized internal forces given as the product of the displacements $\boldsymbol{u}$ with the linear stiffness matrix $\boldsymbol{K}$, which is the Jacobian of the internal forces with respect to the displacements $\boldsymbol{u}$ evaluated at $\boldsymbol{u} = \boldsymbol{0}$:

$$\boldsymbol{K}(\boldsymbol{u}) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\,, \qquad \boldsymbol{K} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}=\boldsymbol{0}}\,. \tag{2.37}$$

The linear stiffness matrix $\boldsymbol{K}$ is equivalent to the tangential stiffness matrix $\boldsymbol{K}(\boldsymbol{u})$ evaluated at $\boldsymbol{u} = \boldsymbol{0}$. To clarify the notation, $\boldsymbol{K}$ denotes the linear stiffness matrix which is equal to the tangential stiffness matrix evaluated at $\boldsymbol{u} = \boldsymbol{0}$. If the tangential stiffness matrix $\boldsymbol{K}(\boldsymbol{u})$ as a function of the displacements is addressed, the dependence on the displacements is explicitly given.

The linearized equations of motion are then written as

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{g}, \tag{2.38}$$

which is a second order linear ordinary differential equation (ODE). The homogeneous solution of this equation is an eternal oscillation, since no viscous forces are considered in the system. To approximate the physical behavior of structures which exhibit commonly decaying oscillations, viscous damping is often added to the finite element model expressed by the damping matrix $\boldsymbol{C}$ leading to the damped linear equations of motion:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{g}. \tag{2.39}$$

The damping matrix $\boldsymbol{C}$ is often defined as a weighted sum of the mass and stiffness matrix known as Rayleigh damping with the weighting factors $\alpha$ and $\beta$:

$$\boldsymbol{C} = \alpha\boldsymbol{M} + \beta\boldsymbol{K}. \tag{2.40}$$

This purely phenomenological damping approximation is also often used in nonlinear finite elements leading to the damped nonlinear equations of motion

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{g}. \tag{2.41}$$

By now all equations above are semi-discrete equations of motion forming ODEs, which stem from the PDE cast in the variational form (2.18). To solve the ODEs for a given set of initial values, an appropriate time discretization technique is necessary, which is addressed next.

## 2.4 Time Integration

The semi-discretized equations of motion (2.41) or the counterpart without damping (2.36) are ODEs, where only the spatial domain is discretized according to the finite element methodology. For solving the equations of motion for given initial displacements $\boldsymbol{u}_0 = \boldsymbol{u}(t=0)$ and velocities $\boldsymbol{v}_0 = \dot{\boldsymbol{u}}(t=0)$, suitable time integration schemes are necessary. They transform the ODE to sets of algebraic equations, for which the solutions are computed using appropriate algorithms. Thereby, the continuous time interval $t \in [t_0, t_{end}]$ is discretized leading to a set of discrete time steps $\mathcal{T} = \{t_0, t_1, \ldots, t_{end}\}$, at which the solution $\boldsymbol{u}$ is sought for.

For the solution of ODEs, many time integration methods have been developed. In the realm of finite elements, second order one-step integration methods based on the Newmark scheme have turned out to be most suited, since they exploit the second order structure of the ODEs (2.41).

The basic second order one step scheme was proposed by Newmark [New59]. Several enhancements of the method were proposed, of which the HHT-$\alpha$ scheme by Hilber et al. [HHT77] and the generalized-$\alpha$ scheme by Chung and Hulbert [CH93] are the most popular. The advantage of these schemes is the ability to control the numerical dissipation, which is a crucial property for the use in the finite element context. Finite element models suffer from high eigenfrequencies which are artifacts from the spatial discretization. Time integration schemes like the HHT-$\alpha$ and the generalized-$\alpha$ scheme allow for the numerical damping of these spurious high frequency content while low frequency motions are barely affected with numerical damping. In both schemes mentioned, the damping of the high frequency content can be adjusted by a single parameter making the schemes very handy.

In the second order time step schemes, two main approximations are performed. The first is the approximation of the nodal displacements and velocities of the $j+1$-th time step in terms of all quantities of the preceding $j$-th time step and the unknown accelerations at the $j+1$-th time step:

$$\dot{\boldsymbol{u}}_{j+1} = \dot{\boldsymbol{u}}_j + (1-\gamma)h\ddot{\boldsymbol{u}}_j + \gamma h\ddot{\boldsymbol{u}}_{j+1}, \tag{2.42}$$

$$\boldsymbol{u}_{j+1} = \boldsymbol{u}_j + h\dot{\boldsymbol{u}}_j + \left(\frac{1}{2} - \beta\right) h^2\ddot{\boldsymbol{u}}_j + \beta h^2\ddot{\boldsymbol{u}}_{j+1}. \tag{2.43}$$

The value $h$ is the time step size. The parameters $\gamma \in [0,1]$ and $\beta \in [0, \frac{1}{2}]$ are responsible for the weighting of the previous and the current time step for the approximation of the displacements and the velocities and determine the stability and accuracy of the time integration scheme. For $\gamma = \beta = 0$, the time discretization is completely explicit, so that the displacements and velocities can be determined directly from all quantities of the previous time step. The time integration with this choice of $\gamma$ and $\beta$, though, is always unstable [GR14, pp. 530 ff].

The equations (2.42) and (2.43) have three unknowns: The displacements, the velocities and the accelerations of the $j+1$-th time step. Hence, they can be rearranged to express the velocities and the accelerations of the $j+1$-th time step in terms of the unknown displacements $\boldsymbol{u}_{j+1}$:

$$\dot{\boldsymbol{u}}_{j+1} = \frac{\gamma}{\beta h}(\boldsymbol{u}_{j+1} - \boldsymbol{u}_j) - \frac{\gamma-\beta}{\beta}\dot{\boldsymbol{u}}_j - \frac{\gamma-2\beta}{2\beta}h\ddot{\boldsymbol{u}}_j, \tag{2.44}$$

$$\ddot{\boldsymbol{u}}_{j+1} = \frac{1}{\beta h^2}(\boldsymbol{u}_{j+1} - \boldsymbol{u}_j) - \frac{1}{\beta h}\dot{\boldsymbol{u}}_j - \frac{1-2\beta}{2\beta}\ddot{\boldsymbol{u}}_j. \tag{2.45}$$

The second approximation consists in the time point at which the equilibrium of the equations of motion (2.41) is enforced. In the Newmark scheme, this time point is chosen

to be the $j+1$-th time step. The generalized-$\alpha$ method shifts this time points with the two alpha-parameters $\alpha_f$ and $\alpha_m$ between the $j$-th time step for $\alpha_f = \alpha_m = 0$ to the $j+1$-th time step for $\alpha_f = \alpha_m = 1$. The displacements, velocities, accelerations and external forces are thus evaluated at the shifted time step:

$$\boldsymbol{u}_{j+1-\alpha_f} = (1-\alpha_f)\boldsymbol{u}_{j+1} + \alpha_f \boldsymbol{u}_j, \tag{2.46}$$

$$\dot{\boldsymbol{u}}_{j+1-\alpha_f} = (1-\alpha_f)\dot{\boldsymbol{u}}_{j+1} + \alpha_f \dot{\boldsymbol{u}}_j, \tag{2.47}$$

$$\ddot{\boldsymbol{u}}_{j+1-\alpha_m} = (1-\alpha_m)\ddot{\boldsymbol{u}}_{j+1} + \alpha_m \ddot{\boldsymbol{u}}_j, \tag{2.48}$$

$$\boldsymbol{g}_{j+1-\alpha_f} = (1-\alpha_f)\boldsymbol{g}_{j+1} + \alpha_f \boldsymbol{g}_j, \tag{2.49}$$

and then inserted into the semi-discretized equations of motion (2.41) yielding

$$\boldsymbol{M}\ddot{\boldsymbol{u}}_{j+1-\alpha_m} + \boldsymbol{C}\dot{\boldsymbol{u}}_{j+1-\alpha_f} + \boldsymbol{f}(\boldsymbol{u}_{j+1-\alpha_f}) = \boldsymbol{g}_{j+1-\alpha_f}\,. \tag{2.50}$$

The time integration methods mentioned above differ in the choice of the shift parameters $\alpha_m$, $\alpha_f$ and the choice of the parameters $\beta$ and $\gamma$ for the time discretization.

The generalized-$\alpha$ method expresses the set of parameters in terms of the spectral radius $\rho_\infty$ as:

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \quad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1}, \quad \gamma = \frac{1}{2} - \alpha_m + \alpha_f, \quad \beta = \frac{1}{4}(1 - \alpha_m + \alpha_f)^2. \tag{2.51}$$

The spectral radius $\rho_\infty$ is the adjustment parameter for the numerical dissipation. For $\rho_\infty = 1$ the numerical damping is zero and can be increased by setting $\rho_\infty$ smaller than 1.

The other time integration schemes can be expressed with the framework introduced above. The classical Newmark scheme, for instance, is given for $\alpha_f = \alpha_m = 0$. The HHT-$\alpha$ scheme is given for $\alpha_m = 0$ and $\alpha_f = \alpha \in [0, 0.3]$ where the damping is adjusted with $\alpha$ instead of $\rho_\infty$.

For a given parametrization of the time integration scheme, the solution process involves the solution of the algebraic balance equation (2.50), which is nonlinear if $\boldsymbol{f}$ is a nonlinear function. The solution process of the equation is performed iteratively with a Newton-Raphson solver, where the residual of equation (2.50) is the function for the root searching algorithm:

$$\boldsymbol{r}(\boldsymbol{u}_{j+1}) = \boldsymbol{M}\ddot{\boldsymbol{u}}_{j+1-\alpha_m} + \boldsymbol{C}\dot{\boldsymbol{u}}_{j+1-\alpha_f} + \boldsymbol{f}(\boldsymbol{u}_{j+1-\alpha_f}) - \boldsymbol{g}_{j+1-\alpha_f}\,. \tag{2.52}$$

In the Newton-Raphson loop searching for $\boldsymbol{r}(\boldsymbol{u}_{j+1}) = \boldsymbol{0}$, the residual is linearized around the displacements $\boldsymbol{u}^i_{j+1}$ of the current iteration step $i$ and a correction $\Delta\boldsymbol{u}^i_{j+1}$ is computed. Then the correction is added to the current step yielding the subsequent $i+1$-th iteration step. This procedure is repeated until convergence is gained. Commonly, the convergence is achieved, when the $L^2$-norm of the residual is below a certain, carefully chosen tolerance.

The linearization of the residual equation (2.52) yields the linear system with the Jacobian matrix $\boldsymbol{K}_{dyn}$ of the residual:

$$\underbrace{\left.\frac{\partial \boldsymbol{r}(\boldsymbol{u}_{j+1})}{\partial \boldsymbol{u}_{j+1}}\right|_{\boldsymbol{u}_{j+1}=\boldsymbol{u}^i_{j+1}}}_{\boldsymbol{K}^i_{dyn,j+1}} \Delta\boldsymbol{u}^i_{j+1} = -\boldsymbol{r}(\boldsymbol{u}^i_{j+1}). \tag{2.53}$$

The update routine is written as

$$\boldsymbol{u}^{i+1}_{j+1} = \boldsymbol{u}^i_{j+1} + \Delta\boldsymbol{u}^i_{j+1} \tag{2.54}$$

yielding the displacements for the subsequent iteration step. If the underlying equations of motion are nonlinear, the iteration matrix $\boldsymbol{K}_{dyn}$ changes with every iteration step. This is due to the fact, that $\boldsymbol{K}_{dyn}$ involves the Jacobian of the nonlinear forces which is also known as the tangential stiffness matrix $\boldsymbol{K}(\boldsymbol{u})$ (cf. (2.37)):

$$\boldsymbol{K}^i_{dyn,j+1} = \left.\frac{\partial \boldsymbol{r}(\boldsymbol{u}_{j+1})}{\partial \boldsymbol{u}_{j+1}}\right|_{\boldsymbol{u}^i_{j+1}} = \left(\frac{1-\alpha_m}{\beta h^2}\boldsymbol{M} + \frac{(1-\alpha_f)\gamma}{\beta h}\boldsymbol{C} + (1-\alpha_f)\boldsymbol{K}(\boldsymbol{u}^i_{j+1-\alpha_f})\right). \tag{2.55}$$

In the Newton-Raphson procedure above, the linear system of equations has to be solved in every iteration step. Since the dimension of the system is associated with the dofs of the finite element discretization, fine and accurate meshes involve computer intensive solutions. One approach to circumvent the expensive solution is the use of explicit integration schemes. However, since the finite element system has very high frequencies due to the spatial discretization, explicit integration schemes require extremely small time steps, which are only efficient for scenarios, where shock wave propagations are the relevant dynamics in the system (cf. [GR14]).

In other applications, where the overall global motion of the system is dominating, implicit time integration methods are indispensable. Thereby, the Newton-Raphson method can be substituted by secant methods, quasi-Newton methods or other variants of it. However, the appeal of the Newton method is the quadratic convergence in the vicinity of the solution, which is not gained with other methods.

When static systems instead of dynamical systems are addressed, the same procedure as described above is applied. In the nonlinear balance equation defining the residual (2.52), the inertia and damping terms drop out and the residual is composed of the balance of internal forces $\boldsymbol{f}$ and external forces $\boldsymbol{g}$. For the solution of these type of systems, the external force is usually stepwise increased using pseudo time steps or more advanced techniques like arc-length continuation methods are used. Since the thesis is mainly about the reduction of dynamical systems and the specific solution technique is secondary, the interested reader is referred to literature on this topic, e.g., Bathe [Bat06], Wriggers [Wri08] and Kim and Sankar [KS09].

In dynamical systems, the iteration matrix is a weighted sum of $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{K}$. Since these matrices are not fully populated, they are treated as sparse matrices where only the non-zero entries are stored. They are commonly identified in a pre-allocation step, in which an empty matrix with the sparsity pattern of the problem is built. The assembly routine described in Subsection 2.3.2 uses this matrix for efficiently adding the elemental contributions without allocating new memory by changing the sparsity pattern of the system. Figure 2.2, left, shows exemplarily the sparsity pattern of the cantilever given in the next section.

For efficient computations, not only the assembly is relevant but also the solution of the sparse system. The strategies are basically divided in two classes: direct and indirect solvers. The direct solvers use elimination to solve the linear system. Thereby, the sparsity of the system is heavily exploited yielding huge advantages over classical solution techniques of dense systems. The computational cost is usually determined not only by the dimension of the matrix, but also by its band width, which is also shown exemplarily in Figure 2.2, right. The indirect solvers seek a solution in an iterative and approximate manner. They are the method of choice for very large systems, where direct methods are not feasible any more. They need very problem-specific preconditioning techniques which go far beyond the topic of this thesis. The interested reader is hence referred to the literature in that field, e.g., Quarteroni et al. [QSS10] or Golub and van Loan [GV12].

In all examples in this thesis, direct solvers are used. They show excellent accuracy and good speed for the problems addressed in the thesis. The solver package used in the

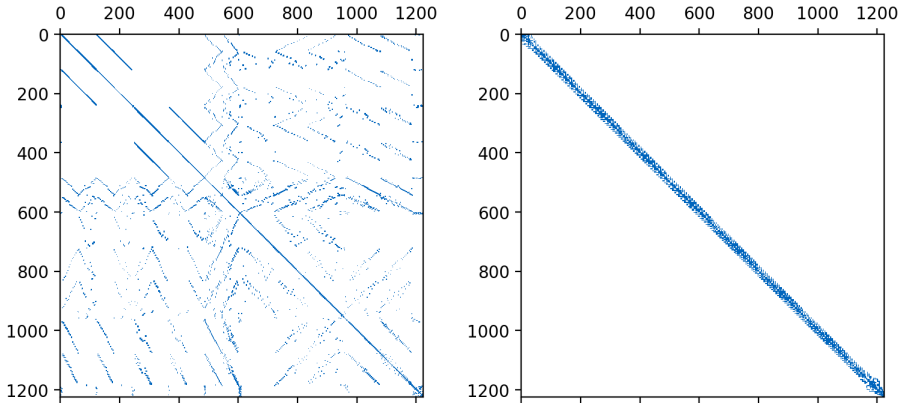


**Figure 2.2:** Sparsity pattern of the iteration matrix $\boldsymbol{K}_{dyn}$ of the cantilever depicted in Figure 2.4. Every dot represents a non-zero entry in the matrix. Left the original matrix, right the reordered matrix (reordered using the reverse Cuthill-McKee algorithm [CM69]) revealing the bandwidth of the problem.

AMFE code is the PARDISO package [Sch+10] which has excellent execution speeds and proved to be very efficient for PDE based nonlinear solutions (cf. eg. [Sch+01]).

## 2.5 Applications: Large Deformations and Geometric Nonlinearity

To illustrate the effect of geometric nonlinearity, the nonlinear finite element methodology outlined above is applied to two beam examples. In Figure 2.3, the discretization, the loading and the observed nodes of the two examples are depicted, the cantilever (Figure 2.3 top) and the clamped-clamped beam (Figure 2.3 bottom). Both are modeled with a St. Venant-Kirchhoff material and discretized with triangular Tri6 elements with quadratic shape functions. The external forces are applied in 20 equally spaced load steps using a forcing factor from 0.05 to 1 representing the scaling of the external load.

Figure 2.4 depicts both the linearized and the nonlinear static solution of both beams. Obviously, the linearized solution depicted with the green colored mesh differs clearly from the blue colored mesh depicting the geometrically nonlinear solution. Basically, two effects can be observed. First, the displacements in the linear setup are larger than in the nonlinear setup. This is also illustrated in Figure 2.5, where the displacements of the observation nodes are drawn over the forcing factor for both the linearized and nonlinear solution. In both the cantilever and the fixed-fixed setup, the displacements of the linear system are proportional to the external forces. In the geometrically nonlinear setup, the displacements are not proportional to the loading for larger forces, but a stiffening effect is visible producing smaller displacements.

Second, the rotations involved in the larger displacements produce a growth of the elements in the linear case. This leads to unphysical behavior of the mesh with a blow up of the volume. Whilst large translational displacements do not produce nonlinear behavior a priori, large rotations are doomed to fail in a linearized description, as illustrated in Figure 2.6.

Geometric nonlinear systems can exhibit many more effects including geometric instabilities caused by buckling or snap-through behavior, dynamic instabilities in rotations and many more. Since a detailed study of all effects goes far beyond the topic of this thesis, the nonlinear examples are all in the domain of large deformations, as they are common for geometrically nonlinear structural dynamics.

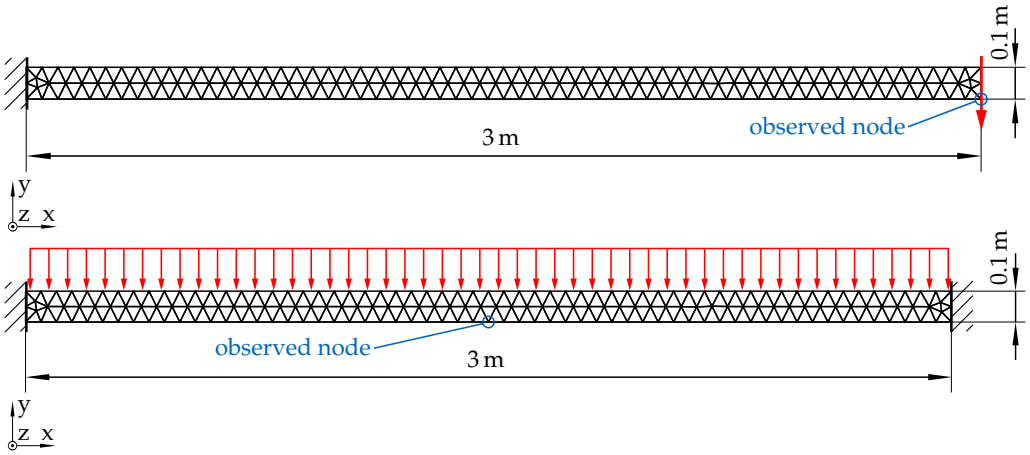


**Figure 2.3:** Discretization, loading and observed nodes of the cantilever (top) and the clamped-clamped beam (bottom).

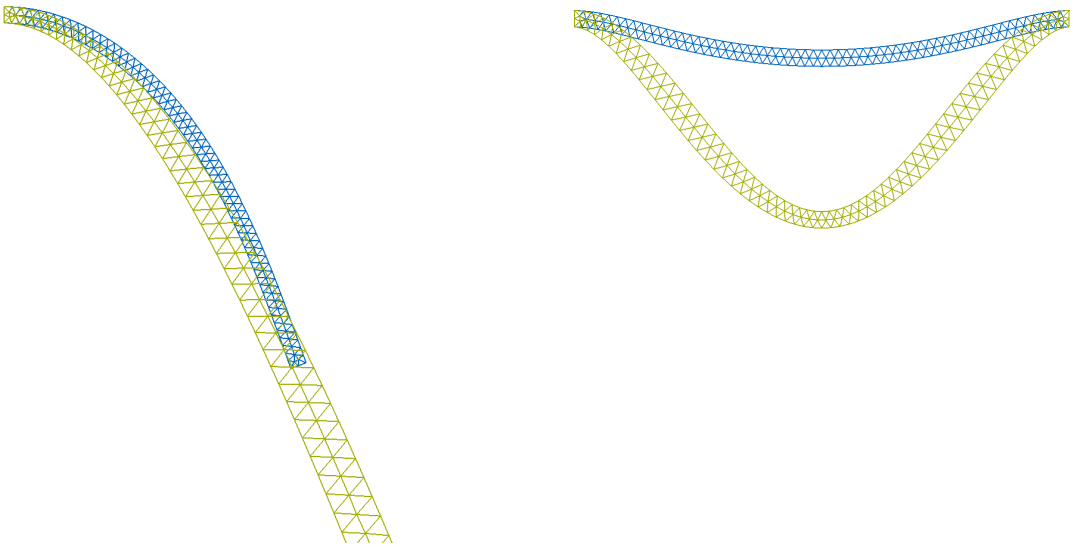

**Figure 2.4:** Linearized (green) and geometrically nonlinear (blue) static displacements of the cantilever (left) and the clamped-clamped beam (right).

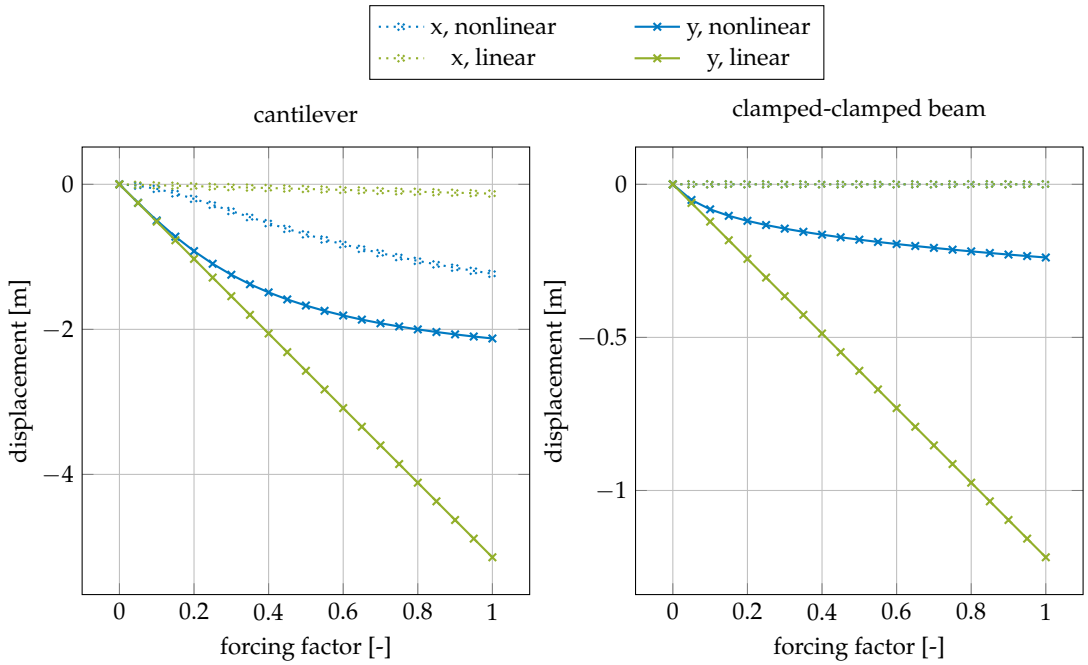


**Figure 2.5:** Displacements of the observed node in $x$ and $y$ direction of the linearized and geometrically nonlinear cantilever (left) and clamped-clamped beam (right).

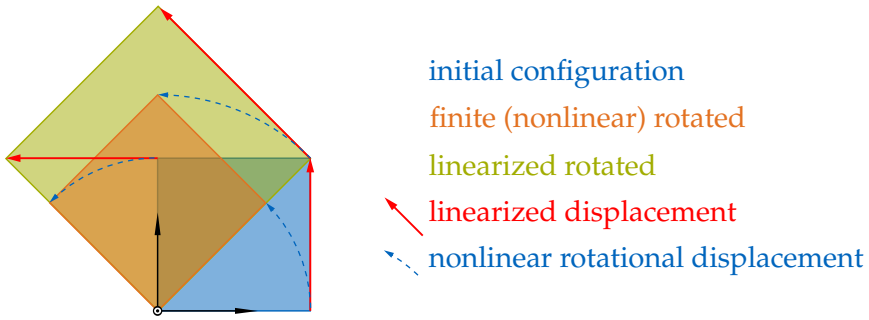


**Figure 2.6:** Comparison of rotation of a single square element about 45° and the linearized version of the rotation. The linearization of the rotation increases the area of the element in a non-physical manner.

# Part I

# Reduced Basis

# Chapter 3

# Model Order Reduction using Subspace Projection

Projective model order reduction is the key concept in reducing the number of dofs of a dynamical system from a large number $N$ to a small number $n$. Since the projection can be interpreted as a Ritz or Galerkin method, it shares the same underlying theory as the discretization of the finite element method, which has been discussed in the previous chapter. In contrary to the finite element method, where a continuous problem of dimension infinity is reduced to a discrete problem of dimension $N$, the projective model order reduction repeats the procedure to reduce $N$ further to the reduced dimension $n$. It is, however, not limited to finite element systems but can be applied to various systems. In this context, though, it is derived and discussed for nonlinear finite element systems which have been introduced in Chapter 2.

## 3.1 Fundamentals of Projective Model Order Reduction

The point of departure are the semi-discretized equations of motion, which might stem from a finite element system (cf. (2.41)):

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{g}. \tag{3.1}$$

The matrix $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ is the constant mass matrix, $\boldsymbol{C} \in \mathbb{R}^{N\times N}$ is the damping matrix and $\boldsymbol{f} \in \mathbb{R}^N$ denotes the nonlinear restoring forces. The vector $\boldsymbol{g} \in \mathbb{R}^N$ represents the external forces and $\boldsymbol{u} \in \mathbb{R}^N$ is the vector of generalized displacements. If the equation above stems from the finite element discretization, the generalized displacement vector $\boldsymbol{u}$ represents the nodal displacements. The dimension $N$ of the equations of motion represents directly the resolution of the finite element mesh, i.e., a coarse mesh results in a smaller dimension of (3.1) compared to a finer mesh and vice versa.

The number of unknowns $N$ is typically very high for industrial finite element models, since complicated geometries require fine meshes. The dimension of the dynamic problem, though, is often much smaller than $N$, i.e., the resulting displacements $\boldsymbol{u}$ of the mechanical problem are bound to a small subspace forming the set of all possible configurations. This fact is exploited by projective model order reduction.

If the subspace is known, in which the solution $\boldsymbol{u}$ of (3.1) is assumed to live, it can be spanned by a set of $n$ basis vectors. These basis vectors can be arranged in a matrix $\boldsymbol{V} \in \mathbb{R}^{N\times n}$ and the physical displacement vector $\boldsymbol{u}$ can then be approximated by a linear combination of these basis vectors. When the amplitudes of the basis vectors are stored in $\boldsymbol{q} \in \mathbb{R}^n$, the physical displacements $\boldsymbol{u}$ can be expressed in terms of the basis $\boldsymbol{V}$ and the new generalized coordinates gathered in $\boldsymbol{q}$ as:

$$\boldsymbol{u} = \boldsymbol{V}\boldsymbol{q}. \tag{3.2}$$

This approximation is the *key concept* in projective model order reduction. The matrix $\boldsymbol{V}$ is referred to as *reduced basis* spanning the subspace onto which the problem is projected. If (3.2) is inserted in the equations of motion, a residual $\boldsymbol{r} \in \mathbb{R}^N$ occurs, since the $N$ equations cannot be satisfied in general by the $n$ unknowns in $\boldsymbol{q}$:

$$\boldsymbol{M}\boldsymbol{V}\ddot{\boldsymbol{q}} + \boldsymbol{C}\boldsymbol{V}\dot{\boldsymbol{q}} + \boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}) = \boldsymbol{g} + \boldsymbol{r}. \tag{3.3}$$

This residual has to be treated in order to make the equation above uniquely determined. The common concept to handle the residual $\boldsymbol{r}$ is to keep it orthogonal to the column space of $\boldsymbol{V}$, which describes the space of all possible motions of $\boldsymbol{u}$ in (3.2). Hence, with the condition

$$\boldsymbol{V}^T\boldsymbol{r} = \boldsymbol{0}, \tag{3.4}$$

the residual can be projected out of (3.3) by premultiplying with $\boldsymbol{V}^T$, leading to the reduced equations of motion of dimension $n$ instead of $N$ in (3.1):

$$\underbrace{\boldsymbol{V}^T\boldsymbol{M}\boldsymbol{V}}_{\boldsymbol{M}_r}\ddot{\boldsymbol{q}} + \underbrace{\boldsymbol{V}^T\boldsymbol{C}\boldsymbol{V}}_{\boldsymbol{C}_r}\dot{\boldsymbol{q}} + \underbrace{\boldsymbol{V}^T\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q})}_{\boldsymbol{f}_r(\boldsymbol{q})} = \underbrace{\boldsymbol{V}^T\boldsymbol{g}}_{\boldsymbol{g}_r}. \tag{3.5}$$

The matrix $\boldsymbol{M}_r \in \mathbb{R}^{n\times n}$ is the reduced mass matrix, $\boldsymbol{C}_r \in \mathbb{R}^{n\times n}$ the reduced damping matrix, $\boldsymbol{f}_r \in \mathbb{R}^n$ the reduced internal force vector and $\boldsymbol{g}_r \in \mathbb{R}^n$ the reduced external force vector.

Similarly, the reduced equations of motion can also be retrieved by the principle of virtual work, since the kinematically admissible displacements $\delta\boldsymbol{u}$ are defined with (3.2) to

$$\delta\boldsymbol{u} = \boldsymbol{V}\delta\boldsymbol{q}. \tag{3.6}$$

The reduced equations of motion are then equivalent to the derivation above

$$\delta W = \delta\boldsymbol{q}^T\boldsymbol{V}^T(\boldsymbol{M}\boldsymbol{V}\ddot{\boldsymbol{q}} + \boldsymbol{C}\boldsymbol{V}\dot{\boldsymbol{q}} + \boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}) - \boldsymbol{g}) = 0, \tag{3.7}$$

$$\boldsymbol{V}^T\boldsymbol{M}\boldsymbol{V}\ddot{\boldsymbol{q}} + \boldsymbol{V}^T\boldsymbol{C}\boldsymbol{V}\dot{\boldsymbol{q}} + \boldsymbol{V}^T\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}) = \boldsymbol{V}^T\boldsymbol{g}. \tag{3.8}$$

The condition (3.4) for the residual is hence equivalent to the application of the principle of virtual work. This principle, which holds for all mechanical systems, results in the orthogonal projection of the residual onto the column space of $\boldsymbol{V}$ (cf. (3.4)). For ODEs in a domain, where the principle of virtual work does not hold like, e.g., heat transfer problems, the space for the left sided projection can be chosen differently to the approximation space of the primal variable $\boldsymbol{u}$. Then, (3.3) is premultiplied by a matrix $\boldsymbol{W}$ different from $\boldsymbol{V}$ leading to the Petrov-Galerkin approach. The symmetric projection of the matrices $\boldsymbol{M}$ and $\boldsymbol{C}$ is referred to as Galerkin or Bubnov-Galerkin approach, which is in accordance with the principle of virtual work.

The reduction of the semi-discretized equations of motion with the reduced basis $\boldsymbol{V}$ is conceptually the same step as the finite element discretization in Subsection 2.3.1. In the latter, the continuous problem (2.18) living in the infinite function space is projected onto the function space spanned by the shape functions (2.19). This projection is inherently performed by the principle of virtual work, which leads to the symmetric Bubnov-Galerkin projection of the linear and nonlinear operators in (2.18). Due to this symmetric projection, in which function space and trial space are equal, the resulting matrices $\boldsymbol{M}$ and $\boldsymbol{K}$ are symmetric. They can be interpreted as the discrete representation of linear operators defined in the function space spanned by the shape functions. The same holds

also for the reduced matrices $\boldsymbol{M}_r$ and $\boldsymbol{K}_r$, where the function space is not spanned by local shape functions defined on the element level but global function spaces of piecewise polynomials.

When the reduction of the semi-discrete equations of motion is performed according to (3.5) or (3.7), the number of dofs is reduced from $N$ to $n$. If $n$ is much smaller than $N$, the computational costs for the time integration as explained in Section 2.4 or other analysis processes reduce drastically. This is usually the case, since the factorization of a matrix of dimension $n$ is much cheaper if $n \ll N$, even though the high dimensional matrices are sparse. Since the reduced matrices are generally very small, they do not need to be sparse, though they could be made sparse by a special choice of $\boldsymbol{V}$.

Besides the reduced order, the reduced system does not represent the original system accurately. Since the displacements, the velocities and the accelerations are forced to live in the space spanned by $\boldsymbol{V}$, the reduced system contains additional constraints. They limit the motion of the system, as they reduce the number of dofs. Consequently the reduced system is equivalent to the original system of dimension $N$ with the constraints (3.4) enforced.

## 3.2 Problem Statement

The key question in projective model order reduction is to find a set of basis vectors forming the reduced basis $\boldsymbol{V}$. Thereby, the reduced basis $\boldsymbol{V}$ should span the subspace, in which the high dimensional displacements $\boldsymbol{u}$ evolve. Since the reduced system is constrained to the subspace spanned by the reduced basis, the proper choice of $\boldsymbol{V}$ is elementary to obtain a reduced system, which is a good approximation of the full, unreduced system.

In the construction of the reduced basis, the specific parametrization of $\boldsymbol{V}$ is arbitrary. It can be built by combining various displacement vectors into the matrix $\boldsymbol{V}$. Solely for numerical reasons, it is beneficial for $\boldsymbol{V}$ having a low condition number. A poorly conditioned $\boldsymbol{V}$ can be orthogonalized with orthogonalization schemes like Gram-Schmidt or Householder procedures, so that the numerical procedures are stable. The results of the reconstructed $\boldsymbol{u} = \boldsymbol{V}\boldsymbol{q}$, however, only depend on the spanned subspace and not on the column vectors in $\boldsymbol{V}$.

## 3.3 Measurement of Reduction Error

The ability of a reduced system to represent the full system is a key requirement in model order reduction. Hence, the accuracy of a reduction method is measured by comparing the displacement field obtained from the reduced system with the displacement field of the full, unreduced model, which serves as a reference. To quantify this approximation accuracy, the relative error measure $RE$ is used:

$$RE = \frac{\sqrt{\sum_{t\in\mathcal{T}} \Delta\boldsymbol{u}(t)^T \Delta\boldsymbol{u}(t)}}{\sqrt{\sum_{t\in\mathcal{T}} \boldsymbol{u}_{\text{ref}}(t)^T \boldsymbol{u}_{\text{ref}}(t)}} \cdot 100\% \quad \text{with} \quad \Delta\boldsymbol{u}(t) = \boldsymbol{u}(t) - \boldsymbol{u}_{\text{ref}}(t). \tag{3.9}$$

The vector $\boldsymbol{u}_{\text{ref}}$ represents the displacement field of the full, unreduced system and $\boldsymbol{u} = \boldsymbol{V}\boldsymbol{q}$ the restored displacements of the reduced system. The time step set $\mathcal{T} = \{t_0, t_1, \ldots, t_{end}\}$ contains the time steps of the time integration.

The $RE$ as a global error measure considers the error in the full time interval of the time steps in $\mathcal{T}$. Since errors in a time integration are cumulated, the error measures of different time spans are different even when the systems are equal but are run for different time frames. Consequently, the $RE$ error is only a valid comparative measure for different reduction methods, when the time frame is kept equal.

# Chapter 4

# Model Order Reduction in Linear Structural Dynamics

The finite element method was developed for linear elastic systems first and was subsequently extended to nonlinear continuum mechanics. As the computational resources were more limited in the early days of the development of the finite element method, reduction and substructuring methods were simultaneously invented in the field of structural dynamics. In the 1960s and 1970s, the most prominent linear substructuring techniques were developed which are still state of the art like the Guyan Reduction method [Guy65], the Craig-Bampton method [CB68], or later on the methods by MacNeal [Mac71] or Rubin [Rub75].

Model order reduction did not only grow popular in structural dynamics. Also in the field of systems theory and control, the demand for fast and compact linear models drove the development of linear reduction techniques. Since the linear equations of motion can be investigated with the methods known from both structural dynamics and systems theory, many methods were developed independently. Furthermore, these two branches still face independent developments and improvements, and were linked, if so, not from the early days on. Since the task of discussing linear reduction methods either in linear structural dynamics or systems theory goes far beyond the scope of the thesis, only a short introduction to the common concepts is given. Exhaustive overviews for the system theoretics based methods can be found in the textbook of Antoulas [Ant09] and the references therein. Model order reduction in the realm of structural dynamics is discussed broadly in the textbook of Craig and Kurdilla [Cra81] and of Géradin and Rixen [GR14].

## 4.1 Key Idea: Subspace Projection of Linear Operators

In linear structural dynamics systems, the internal forces are linearized around a point of equilibrium. If the zero-displacements are chosen such, that the selected point of equilibrium is at $\boldsymbol{u} = \boldsymbol{0}$, the governing equation is given as

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{g} \tag{4.1}$$

with the mass matrix $\boldsymbol{M}$, the damping matrix $\boldsymbol{C}$, the stiffness matrix $\boldsymbol{K}$ being the Jacobian of the internal forces $\partial \boldsymbol{f} / \partial \boldsymbol{u}|_{\boldsymbol{u}=\boldsymbol{0}}$ and the external force vector $\boldsymbol{g}$. Since all operators are linear, the reduced system is obtained as

$$\underbrace{\boldsymbol{V}^T\boldsymbol{M}\boldsymbol{V}}_{\boldsymbol{M}_r}\ddot{\boldsymbol{q}} + \underbrace{\boldsymbol{V}^T\boldsymbol{C}\boldsymbol{V}}_{\boldsymbol{C}_r}\dot{\boldsymbol{q}} + \underbrace{\boldsymbol{V}^T\boldsymbol{K}\boldsymbol{V}}_{\boldsymbol{K}_r}\boldsymbol{q} = \underbrace{\boldsymbol{V}^T\boldsymbol{g}}_{\boldsymbol{g}_r} \tag{4.2}$$

with the reduced mass, damping and stiffness matrices $\boldsymbol{M}_r$, $\boldsymbol{C}_r$ and $\boldsymbol{K}_r$, which can be precomputed. These reduced matrices characterize the reduced system and can be interpreted as the projection of the linear operators onto the subspace spanned by $\boldsymbol{V}$.

Since the linear mechanical system is entirely characterized by the mass, damping and stiffness matrix, system theoretic properties and also rather intuitive physical insights form the basis for the reduction methods. They are discussed in the following.

## 4.2 Modal Truncation

The oldest model order reduction technique is the modal truncation. It relies on the principle, that the dynamical system is approximated by a superposition of modal displacements. This idea goes back at least to Lord Rayleigh [Lor94], where the vibration modes were computed in an analytical fashion for simple structures like bars, beams and plates even before the finite element method was developed.

The basic idea is the analysis of the free motion of the undamped homogeneous system:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{0} \tag{4.3}$$

which can be analytically solved with the ansatz

$$\boldsymbol{u}(t) = \boldsymbol{\phi}_i \cos(\omega_i t + \alpha). \tag{4.4}$$

Inserting this ansatz into (4.3) and cancelling out the time dependent part $\cos(\omega_i t + \alpha)$, one obtains the generalized eigenvalue problem

$$\omega_i^2 \boldsymbol{M}\boldsymbol{\phi}_i = \boldsymbol{K}\boldsymbol{\phi}_i \tag{4.5}$$

where the solutions $\omega_i$ are the undamped eigenfrequencies and $\boldsymbol{\phi}_i$ are the corresponding eigenvectors of the system characterized in (4.3).

The eigenvectors $\boldsymbol{\phi}_i$, which are also denoted as vibration modes, eigenmodes or mode shapes, are the spatial solutions of the homogeneous undamped system (4.3), which oscillate with the associated frequency $\omega_i$ according to the ansatz (4.4).

The solutions of the eigenvalue problem (4.5) are not unique, since the norm of the eigenvector $\boldsymbol{\phi}_i$ is not determined. Consequently, an additional condition can be applied to define the norm of $\boldsymbol{\phi}_i$. Most commonly, a mass normalization is employed which fixes the length of the eigenvector $\boldsymbol{\phi}_i$ to one in the $\boldsymbol{M}$-norm:

$$\boldsymbol{\phi}_i^T \boldsymbol{M}\boldsymbol{\phi}_i = 1. \tag{4.6}$$

The vibration modes $\boldsymbol{\phi}_i$ share a further property. They are mass and stiffness orthogonal, i.e., in both the $\boldsymbol{M}$ and $\boldsymbol{K}$-norm, the modes are orthogonal to each other. For a derivation of this property, the interested reader is referred to Ewins [Ewi84]. If all eigenmodes are gathered in the matrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N]$, the mass and stiffness matrix are diagonalized by the transformation given by $\boldsymbol{\Phi}$. While the transformed mass matrix yields identity, the stiffness matrix results in the squared diagonal matrix $\boldsymbol{\Omega} = \operatorname{diag}(\omega_1, \ldots, \omega_N)$ which contains the undamped eigenfrequencies $\omega_i$ of the system:

$$\boldsymbol{\Phi}^T \boldsymbol{M}\boldsymbol{\Phi} = \boldsymbol{I}, \qquad \boldsymbol{\Phi}^T \boldsymbol{K}\boldsymbol{\Phi} = \boldsymbol{\Omega}^2 = \begin{pmatrix} \omega_1^2 & & 0 \\ & \ddots & \\ 0 & & \omega_N^2 \end{pmatrix}. \tag{4.7}$$

Model order reduction using modal truncation is realized by composing the basis $\boldsymbol{V}$ of a selected set of vibration modes $\boldsymbol{\phi}_i$. When the damped system (4.1) is considered, the

system can only be decoupled if the damping matrix $\boldsymbol{C}$ is diagonalized by the projection onto the modal coordinates. When Rayleigh damping is applied (cf. Subsection 2.3.3), $\boldsymbol{C}$ is diagonalized since $\boldsymbol{M}$ and $\boldsymbol{K}$ are diagonalized as above. Using the full modal basis $\boldsymbol{\Phi}$ as reduction basis $\boldsymbol{V}$, the system (4.1) decouples into

$$\begin{pmatrix} \ddot{q}_1 \\ \vdots \\ \ddot{q}_N \end{pmatrix} + 2 \begin{pmatrix} \zeta_1\omega_1 & & \\ & \ddots & \\ & & \zeta_N\omega_N \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_N \end{pmatrix} + \begin{pmatrix} \omega_1^2 & & \\ & \ddots & \\ & & \omega_N^2 \end{pmatrix} \begin{pmatrix} q_1 \\ \vdots \\ q_N \end{pmatrix} = \begin{pmatrix} \boldsymbol{\phi}_1^T \\ \vdots \\ \boldsymbol{\phi}_N^T \end{pmatrix} \boldsymbol{g}, \tag{4.8}$$

leading to a system of $N$ decoupled equations with the modal amplitudes $q_i$ gathered in $\boldsymbol{q} = (q_1, \ldots, q_N)$ and the damping ratio $\zeta_i$. The excitation of mode $i$ is given by the projection of the external forces $\boldsymbol{g}$ onto the mode yielding $\boldsymbol{\phi}_i^T \boldsymbol{g}$.

As (4.8) illustrates, the modal basis $\boldsymbol{\Phi}$ decouples the equations of motion (4.1) into $N$ independent ODEs. In modal truncation, the full basis $\boldsymbol{\Phi}$ is truncated, so that $\boldsymbol{V}$ is only composed of a selection of vibration modes $\boldsymbol{\phi}_i$. Since the equations of motion are decoupled in the modal space, the truncation of a mode does not influence the result of another mode, which is left in the basis $\boldsymbol{V}$.

Basically two criteria exist for the selection of the modes $\boldsymbol{\phi}_i$ as members for the basis $\boldsymbol{V}$. The first criterion is based on the eigenfrequency $\omega_i$ associated with the eigenmode $\boldsymbol{\phi}_i$. According to Géradin and Rixen [GR14], as a rule of thumb, the modes with eigenfrequencies up to twice as high as the frequency range of interest are recommended to be taken into the basis. Hence, when the $n$-th eigenfrequency $\omega_n$ is about two times the maximum frequency of interest, the reduced basis of size $n$ is constructed as:

$$\boldsymbol{V} = (\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n). \tag{4.9}$$

The second criterion is based on the excitation of one modal dof. Since the excitation of the $i$-th mode is given by $\boldsymbol{\phi}_i^T \boldsymbol{g}$, different variants of the so-called modal dominance criterion can be applied. In this criterion, the modes where the term $\boldsymbol{\phi}_i^T \boldsymbol{g}$ is high and which are thus strongly excited are chosen for the basis $\boldsymbol{V}$, whereas the modes with small excitation are left out. In practical applications, usually both concepts are combined to obtain a good approximation while keeping the size $n$ of the reduced basis small.

The modal truncation approach has multiple advantages. Since the vibration modes are associated with an eigenfrequency, they are a good choice for harmonic excitations. Furthermore, modal reduction is commonly used when experimental data of the structure area available. Then, the modal decoupling allows for assigning modal damping ratios obtained from the measurements to every mode. One further advantage of modal reduction is, that the system is decoupled when transformed into modal space, as illustrated in (4.8). Due to the decoupling, the truncation of modes as performed in the modal reduction does not affect the motion of the retained modes. That means, that the reduction error is limited to the space of the truncated modes in the metric of the $\boldsymbol{M}$-norm. As a consequence, the displacements of a modal reduced system are equal to the displacements of a full, unreduced system, when observed in the modal space spanned by $\boldsymbol{V}$ in the $\boldsymbol{M}$-norm.

The drawbacks are the computational costs associated with the computation of the eigenmodes. For large systems, this is a computer intensive task, for which several methods have been developed for the efficient computation of eigenmodes in a certain frequency range. Most prominently, power iteration methods are used in combination with Lanczos iterations. However, other methods have been developed like the Jacobsen or the FEAST algorithm proposed by Bai et al. [Bai+00] and Polizzi [Pol09], which has not

been used in the realm of structural dynamics yet. For established strategies for the computation of large eigenvalue problems, the reader is referred to the textbooks of Craig [CK06], Géradin and Rixen [GR14] and Golub and van Loan [GV12].

## 4.3 Perturbation of Eigenmodes

When the linearized system (4.1) is a parametric system and the system matrices $\boldsymbol{M}$ and $\boldsymbol{K}$ depend on a parameter $p$, the change of the vibration modes $\boldsymbol{\phi}_i$ in dependence of $p$ are often of interest. This sensitivity is expressed by the so-called modal derivative $\partial\boldsymbol{\phi}_i/\partial p$ yielding the perturbation of mode $\boldsymbol{\phi}_i$ with respect to the parameter $p$. In the following, the modal derivatives are derived based on the concept of Nelson [Nel76], which is transformed to second order systems here. For the sake of completeness it is mentioned that further techniques exist like , e.g., Seyranian [Sey93], Akgun [Akg94] and Zhangping and JinWu [ZJ07]. The interested reader is referred to the extensive overview papers of Tortorelli and Michaleris [TM94] and Van Keulen et al. [VHK05] in that field.

With $\boldsymbol{M} = \boldsymbol{M}(p)$ and $\boldsymbol{K} = \boldsymbol{K}(p)$, the eigenvalue problem (4.5) becomes dependent on the parameter $p$. The derivation of the eigenvalue problem with respect to this parameter leads to

$$\frac{\partial}{\partial p}\left(-\omega_i^2\boldsymbol{M}(p)+\boldsymbol{K}(p)\right)\boldsymbol{\phi}_i = \boldsymbol{0} \tag{4.10}$$

$$\left(-\frac{\partial\omega_i^2}{\partial p}\boldsymbol{M}-\omega_i^2\frac{\partial\boldsymbol{M}}{\partial p}+\frac{\partial\boldsymbol{K}}{\partial p}\right)\boldsymbol{\phi}_i+\left(-\omega_i^2\boldsymbol{M}+\boldsymbol{K}\right)\frac{\partial\boldsymbol{\phi}_i}{\partial p} = \boldsymbol{0} \tag{4.11}$$

with the modal derivative $\partial\boldsymbol{\phi}_i/\partial p$ of mode $\boldsymbol{\phi}_i$ with respect to the parameter $p$. To keep the notation uncluttered, the dependence of $\boldsymbol{M}$ and $\boldsymbol{K}$ on $p$ is not given explicitly. The derivatives $\partial\boldsymbol{M}/\partial p$ and $\partial\boldsymbol{K}/\partial p$ are determined by the parametric mechanical system. They can be computed via an analytic scheme, where the dependence of $p$ is explicitly accounted for or an implicit finite difference scheme. However, the derivative of the squared $i$-th eigenfrequency cannot be obtained directly. To obtain $\partial\omega_i^2/\partial p$ from the equation above, (4.11) is premultiplied by $\boldsymbol{\phi}_i^T$. With the definition of the eigenvalue problem in its transposed form $\boldsymbol{\phi}_i^T(\omega_i^2\boldsymbol{M}+\boldsymbol{K}) = \boldsymbol{0}^T$ and $\boldsymbol{\phi}_i^T\boldsymbol{M}\boldsymbol{\phi}_i = 1$, one obtains the desired

$$\frac{\partial\omega_i^2}{\partial p} = \boldsymbol{\phi}_i^T\left(-\omega_i^2\frac{\partial\boldsymbol{M}}{\partial p}+\frac{\partial\boldsymbol{K}}{\partial p}\right)\boldsymbol{\phi}_i. \tag{4.12}$$

With $\partial\omega_i^2/\partial p$ given in the equation above, the modal derivative can be determined by solving the rearranged (4.11) with the right hand side *rhs*:

$$\left(-\omega_i^2\boldsymbol{M}+\boldsymbol{K}\right)\frac{\partial\boldsymbol{\phi}_i}{\partial p} = \underbrace{\left(\frac{\partial\omega_i^2}{\partial p}\boldsymbol{M}+\omega_i^2\frac{\partial\boldsymbol{M}}{\partial p}-\frac{\partial\boldsymbol{K}}{\partial p}\right)\boldsymbol{\phi}_i}_{rhs}. \tag{4.13}$$

The right hand side *rhs* is known, since $\partial\boldsymbol{M}/\partial p$ and $\partial\boldsymbol{K}/\partial p$ are given by the mechanical system and $\partial\omega_i^2/\partial p$ is computed according to (4.12). The coefficient matrix $(-\omega_i^2\boldsymbol{M}+\boldsymbol{K})$ is singular, though. This means, that the solution of the equation above is not determined in the direction of the null space, which is formed by the vibration mode $\boldsymbol{\phi}_i$. Consequently, an additional constraint is necessary for determining the component of the modal derivative in the null space of the coefficient matrix. Requiring, that the norm of the vibration mode (4.6) remains unchanged is one of the manners to impose such a condition. With differentiating (4.6) with respect to $p$, one obtains:

$$\frac{\partial}{\partial p}\left(\boldsymbol{\phi}_i^T\boldsymbol{M}\boldsymbol{\phi}_i\right) = 0, \tag{4.14}$$

$$\left(\frac{\partial \boldsymbol{\phi}_i}{\partial p}\right)^T \boldsymbol{M}\boldsymbol{\phi}_i + \boldsymbol{\phi}_i^T \frac{\partial \boldsymbol{M}}{\partial p}\boldsymbol{\phi}_i + \boldsymbol{\phi}_i^T \boldsymbol{M}\frac{\partial \boldsymbol{\phi}_i}{\partial p} = 0, \tag{4.15}$$

and with the symmetry of $\boldsymbol{M}$, the condition for the modal derivative $\partial\boldsymbol{\phi}_i/\partial p$ is:

$$\left(\frac{\partial \boldsymbol{\phi}_i}{\partial p}\right)^T \boldsymbol{M}\boldsymbol{\phi}_i = -\frac{1}{2}\boldsymbol{\phi}_i^T \frac{\partial \boldsymbol{M}}{\partial p}\boldsymbol{\phi}_i. \tag{4.16}$$

The condition above gives an additional constraint to the modal derivative. To solve the singular system (4.13), one dof affecting the null space of the coefficient matrix is fixed in order to make the coefficient matrix invertible. The system is partitioned such, that the index 2 denotes the index of the fixed dof:

$$\begin{pmatrix} -\omega_i^2\boldsymbol{M}_{11} + \boldsymbol{K}_{11} & 0 & -\omega_i^2\boldsymbol{M}_{13} + \boldsymbol{K}_{13} \\ 0 & 1 & 0 \\ -\omega_i^2\boldsymbol{M}_{31} + \boldsymbol{K}_{31} & 0 & -\omega_i^2\boldsymbol{M}_{33} + \boldsymbol{K}_{33} \end{pmatrix} \begin{pmatrix} \boldsymbol{v}_1 \\ v_2 \\ \boldsymbol{v}_3 \end{pmatrix} = \begin{pmatrix} rhs_1 \\ 0 \\ rhs_3 \end{pmatrix}. \tag{4.17}$$

The system is non-singular, if the pivot index $v_2$ is chosen well. Since the null space of the coefficient matrix is $\boldsymbol{\phi}_i$, a good choice to affect the subspace is to pick the maximum absolute value of the eigenvector $\boldsymbol{\phi}_i$. The solution of (4.13) is given with the particular solution $\boldsymbol{v} = \left(\boldsymbol{v}_1^T, 0, \boldsymbol{v}_3^T\right)^T$ of (4.17) and the null space solution which is $\boldsymbol{\phi}_i$:

$$\frac{\partial \boldsymbol{\phi}_i}{\partial p} = \boldsymbol{v} + c\boldsymbol{\phi}_i. \tag{4.18}$$

The amplitude of the null space solution is then determined with (4.16) yielding the component $c$ for the null space solution in the equation above:

$$\left(\boldsymbol{v} + c\boldsymbol{\phi}_i\right)^T \boldsymbol{M}\boldsymbol{\phi}_i = -\frac{1}{2}\boldsymbol{\phi}_i^T \frac{\partial \boldsymbol{M}}{\partial p}\boldsymbol{\phi}_i, \tag{4.19}$$

$$c = -\frac{1}{2}\boldsymbol{\phi}_i^T \frac{\partial \boldsymbol{M}}{\partial p}\boldsymbol{\phi} - \boldsymbol{v}^T\boldsymbol{M}\boldsymbol{\phi}_i. \tag{4.20}$$

With the procedure described above, the derivatives of vibration modes can be determined analytically. Other methods exist to incorporate the constraint (4.16) to the singular system (4.13). However, with the proposed approach, neither the sparsity nor the symmetry of the coefficient matrix is spoiled, whereas several other approaches suffer from these drawbacks. Note, that the procedure is only valid for two distinct eigenfrequencies $\omega_i$. If multiple eigenfrequencies exist, they have to be accounted for as discussed, e.g., in Slaats et al. [SdS95].

The derivatives of the mass and stiffness matrices have to be computed within the finite element program. This can either be done directly, so that the derivatives $\partial\boldsymbol{K}/\partial p$ and $\partial\boldsymbol{M}/\partial p$ are computed on the element level and then assembled as described in Subsection 2.3.2 and discussed in Idelsohn and Cardona [IC85b]. Other approaches are the computation via finite differences of the full assembled matrices. However, these procedures can lead to numerical errors, if one-sided difference schemes are used instead of central differences with tuned step width. For speeding up finite difference schemes, hybrid approaches as proposed by [VD98] or the refined version [DV00] can be used, as they allow to obtain good accuracy with one-sided difference schemes by a special treatment of the rigid body modes of every element. They require a modification of the element routines, though.

## 4.4 Krylov Subspace Reduction

The modal truncation method as described in Section 4.2 provides a good basis for various dynamic problems. It relies on the homogeneous equations of motion, where the force excitation location is ignored. However, often many vibration modes have to be taken into the basis to gain accurate results (cf. [CK06, p. 349]), since the external forces are not accounted for in the method.

The Krylov subspace method, on the contrary, builds a basis with the knowledge of the force input locations. In many cases the excitation forces act in a small subspace of the physical vector space, since the external forces act only on few points or surfaces. Then, the external force vector $\boldsymbol{g}$ can be expressed as a product of the constant location matrix $\boldsymbol{G} \in \mathbb{R}^{N\times p}$ describing the subspace, in which the external excitation lives and the time dependent amplitude $\hat{\boldsymbol{g}}(t) \in \mathbb{R}^p$ representing the variation of the forces with respect to time:

$$\boldsymbol{g} = \boldsymbol{G}\hat{\boldsymbol{g}}. \tag{4.21}$$

The key idea of the Krylov subspace method is to build the basis $\boldsymbol{V}$ from static displacement fields and higher order approximations. The starting point are the linearized equations of motion with neglected damping:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{G}\hat{\boldsymbol{g}}. \tag{4.22}$$

Let $\boldsymbol{v}_1$ be the first approximation of the displacement field. If $\boldsymbol{u} = \boldsymbol{v}_1$ is inserted into the equation above, one obtains

$$\boldsymbol{M}\ddot{\boldsymbol{v}}_1 + \boldsymbol{K}\boldsymbol{v}_1 = \boldsymbol{G}\hat{\boldsymbol{g}}. \tag{4.23}$$

As a first approximation of the solution, the acceleration forces are neglected and $\ddot{\boldsymbol{v}}_1$ is dropped from the equation leading to the first order approximation for $\boldsymbol{v}_1$:

$$\boldsymbol{v}_1 = \boldsymbol{K}^{-1}\boldsymbol{G}\hat{\boldsymbol{g}}. \tag{4.24}$$

The time dependent amplitudes $\hat{\boldsymbol{g}}$ are arbitrary. However, independently of $\hat{\boldsymbol{g}}$, $\boldsymbol{v}_1$ lives in the subspace spanned by $\boldsymbol{K}^{-1}\boldsymbol{G}$ leading to the first set of vectors $\boldsymbol{v}_{1,\mathrm{kry}} \in \mathbb{R}^{N\times p}$:

$$\boldsymbol{v}_{1,\mathrm{kry}} = \boldsymbol{K}^{-1}\boldsymbol{G}. \tag{4.25}$$

Obviously the equation above states the solution of a static problem. Hence $\boldsymbol{v}_{1,\mathrm{kry}} = \boldsymbol{K}^{-1}\boldsymbol{G}$ spans the subspace of all static solutions, which can be reached by $\boldsymbol{G}\hat{\boldsymbol{g}}$. Since $\boldsymbol{v}_1$ is only the first order approximation of the desired subspace, in which the displacements $\boldsymbol{u}$ live, one can add a further vector $\boldsymbol{v}_2$ to approximate the displacement field yielding $\boldsymbol{u} = \boldsymbol{v}_1 + \boldsymbol{v}_2$. Inserting this equation into (4.22) results in

$$\boldsymbol{M}(\ddot{\boldsymbol{v}}_1 + \ddot{\boldsymbol{v}}_2) + \boldsymbol{K}(\boldsymbol{v}_1 + \boldsymbol{v}_2) = \boldsymbol{G}\hat{\boldsymbol{g}} \tag{4.26}$$

Once again, the acceleration $\ddot{\boldsymbol{v}}_2$ of the unknown $\boldsymbol{v}_2$ is dropped. Furthermore, $\boldsymbol{K}\boldsymbol{v}_1$ on the left hand side and $\boldsymbol{G}\hat{\boldsymbol{g}}$ cancel out leading to the second order approximation vector $\boldsymbol{v}_2$:

$$\boldsymbol{v}_2 = -\boldsymbol{K}^{-1}\boldsymbol{M}\ddot{\boldsymbol{v}}_1. \tag{4.27}$$

The acceleration $\ddot{\boldsymbol{v}}_1$ is forced to live in the subspace of the displacements $\mathrm{span}(\boldsymbol{v}_1)$. Hence, the second order subspace approximation is

$$\boldsymbol{v}_{2,\mathrm{kry}} = \boldsymbol{K}^{-1}\boldsymbol{M}\boldsymbol{v}_{1,\mathrm{kry}} = \boldsymbol{K}^{-1}\boldsymbol{M}\boldsymbol{K}^{-1}\boldsymbol{G}. \tag{4.28}$$

This procedure can be repeated $m$ times leading to the raw Krylov basis spanning the Krylov subspace:

$$\boldsymbol{V}_{\text{kry,raw}} = \left(\boldsymbol{K}^{-1}\boldsymbol{G}, \boldsymbol{K}^{-1}\boldsymbol{M}\boldsymbol{K}^{-1}\boldsymbol{G}, \ldots, (\boldsymbol{K}^{-1}\boldsymbol{M})^{m-1}\boldsymbol{K}^{-1}\boldsymbol{G}\right). \tag{4.29}$$

The linear independence of the raw Krylov basis is in general very poor. Hence, an orthogonalization needs to be performed in every iteration step, so that $\boldsymbol{v}_{i,\text{kry}}$ is orthogonalized with respect to all previous Krylov vectors, which were recursively themselves orthogonalized with respect to their previous ones. This procedure is referred to as Arnoldi or Lanczos iteration (cf. [Str07] or [GR14]). It produces the orthogonal and well conditioned Krylov basis $\boldsymbol{V}_{\text{kry}}$, which spans exactly the subspace spanned by the raw Krylov basis $\boldsymbol{V}_{\text{kry,raw}}$.

In the derivation above, the static stiffness matrix $\boldsymbol{K}$ is used for the computation of the Krylov subspace. The first Krylov vectors $\boldsymbol{v}_{1,\text{kry}}$ form the static solutions of arbitrary inputs in the space spanned by $\boldsymbol{G}$. Nevertheless, it is possible to run the Krylov iteration not for a static solution but for a dynamic solution at a certain frequency $\omega_{\text{kry}}$. Then the stiffness matrix changes to a dynamic stiffness matrix $\boldsymbol{K}_{dyn} = -\omega_{\text{kry}}^2\boldsymbol{M} + \boldsymbol{K}$ and the first Krylov vectors $\boldsymbol{v}_{1,\text{kry},dyn}$ form the dynamic solutions for arbitrary inputs in the space spanned by $\boldsymbol{G}$ at frequency $\omega_{\text{kry}}$. The frequency-shifted Krylov subspace approximates the responses in the neighborhood of the shift frequency $\omega_{\text{kry}}$ more accurately than the Krylov subspace with regular $\boldsymbol{K}$, which can be regarded as a shifted Krylov subspace around $\omega_{dyn} = 0$.

With the shifted frequency Krylov subspace, the basis $\boldsymbol{V}$ can be tailored more specifically to the problem, if the excitation frequency of the system is known. Furthermore, it can be used for floating structures, since the shift eliminates the semi-definiteness of the stiffness matrix stemming from the rigid body modes, as for instance in the use for space structures in Ricles and Leger [RL93]. However, for non-floating problems exhibiting unknown excitations or excitations in the low frequency range, the regular Krylov subspace without frequency shift is commonly used.

When comparing the modal truncation method with the Krylov subspace method, one difference is obvious. The modal truncation method relies solely on the system and is independent of the excitation. The vibration modes describe the motion of the isolated system and share thus solely system properties. The Krylov subspace method, on the other hand, exploits the property of the input locations or, from the vector space point of view, the excitation subspace. Hence, the subspace gathered in that manner knows of the input locations and tailors the basis specially to these input locations.

Consequently both methods have their advantages and disadvantages. To combine the advantages of both methods, several variants of mixing modal truncation and Krylov subspace vectors were proposed like for example Dickens and Pool [DP92] or Rixen [Rix01]. These methods are referred to as Modal Truncation Augmentation. The basic idea is to combine the basis of modal truncation and a form of Krylov subspace resulting in the basis $\boldsymbol{V}_{MTA}$:

$$\boldsymbol{V}_{MTA} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_q, \boldsymbol{v}_{\text{kry},1}, \ldots, \boldsymbol{v}_{\text{kry},r}]. \tag{4.30}$$

## 4.5 Component Mode Synthesis and Substructuring

The reduction strategies proposed above are designed to reduce a structural dynamical system as a whole. Many applications and work flows though require the coupling of dynamical subsystems. Similar to single finite elements, which are a discretized version of a continuous dynamical system, it is often desirable to reduce a part of a mechanical

system, which is commonly referred to as substructure, while allowing the coupling with other substructures.

The most common approach to handle substructures is the primal assembly, which is the conceptual continuation of the finite element method. A finite element has boundary nodes which are coupled with the neighboring elements' nodes in the assembly process. With this process, the continuity of the displacements as well as the balance of forces between the elements are implicitly accounted for.

A finite element can also exhibit internal nodes describing dofs, which do not require a direct coupling with the neighboring elements. These dofs are commonly referred to as bubble dofs since the displacement fields associated with these dofs have to be zero at the boundary of the element but may have arbitrary shapes inside the element, which however often exhibit a bubble-like shape.

The idea of substructuring and component mode synthesis is the application of this principle to already discretized finite element substructures. Thereby the nodes are partitioned in boundary nodes referred to with index $b$ and internal nodes with index $i$. With this partitioning, the undamped linear equations of motion can be recast to

$$\begin{pmatrix} \boldsymbol{M}_{bb} & \boldsymbol{M}_{bi} \\ \boldsymbol{M}_{ib} & \boldsymbol{M}_{ii} \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{u}}_b \\ \ddot{\boldsymbol{u}}_i \end{pmatrix} + \begin{pmatrix} \boldsymbol{K}_{bb} & \boldsymbol{K}_{bi} \\ \boldsymbol{K}_{ib} & \boldsymbol{K}_{ii} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{g}_b \\ \boldsymbol{g}_i \end{pmatrix} \tag{4.31}$$

with $\boldsymbol{u}_b$ being the boundary dofs and $\boldsymbol{u}_i$ being the internal dofs of the finite element system. In this partition, the boundary displacements $\boldsymbol{u}_b$ have to be preserved to enable the coupling with the adjacent substructures, while the internal dofs can be reduced in order to reduce the number of dofs.

### 4.5.1 Guyan Reduction

The substructuring technique which was proposed independently by Irons [Iro65] and Guyan [Guy65] eliminates all internal dofs by forcing them to follow the external dofs statically. When the inertia terms and the external forces acting on the internal dofs are neglected, (4.31) yields

$$\begin{pmatrix} \boldsymbol{K}_{bb} & \boldsymbol{K}_{bi} \\ \boldsymbol{K}_{ib} & \boldsymbol{K}_{ii} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{g}_b \\ \boldsymbol{0} \end{pmatrix}. \tag{4.32}$$

Since the dynamic effects of the internal dofs are neglected, they follow statically the boundary dofs. The second line of (4.32) can then be solved for the internal static displacements $\boldsymbol{u}_i$ in dependence of the boundary dofs $\boldsymbol{u}_b$

$$\boldsymbol{K}_{ii}\boldsymbol{u}_i + \boldsymbol{K}_{ib}\boldsymbol{u}_b = \boldsymbol{0} \quad \leftrightarrow \quad \boldsymbol{u}_i = -\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}\boldsymbol{u}_b \tag{4.33}$$

and inserted then into the first line of (4.32)

$$\left(\boldsymbol{K}_{bb} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}\right)\boldsymbol{u}_b = \boldsymbol{g}_b \tag{4.34}$$

yielding the Schur complement of the internal dofs in the boundary dofs as additional stiffness for the boundary dofs.

This reduction, in which the internal dofs follow the external dofs, is denoted as *Guyan reduction*, *Guyan-Irons reduction* or *static condensation*. It can be achieved with the following basis, if the dofs are partitioned in boundary dofs and internal dofs as in (4.31):

$$\boldsymbol{V}_{\text{guyan}} = \begin{pmatrix} \boldsymbol{V}_b \\ \boldsymbol{V}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} \end{pmatrix}. \tag{4.35}$$

As pointed out above, with this reduced basis the inertia effects of the internal dofs are not considered with its own dynamics. However, the following extensions allow for the consideration of the dynamics of the internal dofs.

### 4.5.2 Craig-Bampton Reduction

In the Guyan reduction, the inertia effects of the internal dofs are only projected onto the boundary dofs via the Schur complement, but no internal dynamics is considered. In the substructuring technique proposed by Craig and Bampton [CB68], the static condensation method is extended with so-called bubble modes accounting for the dynamics of the internal dofs. These bubble modes are computed as the eigenvectors of the substructure, on which the boundary dofs are fixed. Since the fixation of dofs can be achieved by deleting the dofs from the system, the internal eigenvalue problem is written as

$$\omega_{ii,j} \boldsymbol{M}_{ii} \boldsymbol{\phi}_j = \boldsymbol{K}_{ii} \boldsymbol{\phi}_j \tag{4.36}$$

with the mass matrix $\boldsymbol{M}_{ii}$ and the stiffness matrix $\boldsymbol{K}_{ii}$ of the internal dofs retrieved from the partitioned system (4.31) and the eigenfrequencies $\omega_{ii,j}$ with the corresponding eigenvectors $\boldsymbol{\phi}_j$ forming the bubble modes. The Craig-Bampton reduction basis is then given as the reduced basis of the Guyan method (4.35) augmented with a selection of the bubble modes $\boldsymbol{\phi}_j$ forming the displacement fields for the internal dofs:

$$\boldsymbol{V}_{CB} = \begin{pmatrix} \boldsymbol{V}_b \\ \boldsymbol{V}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ -\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} & \boldsymbol{\phi}_1 & \dots & \boldsymbol{\phi}_m \end{pmatrix}. \tag{4.37}$$

The internal bubble modes $\boldsymbol{\phi}_j$ are usually selected according to the corresponding eigenfrequency. A common approach is that the bubble modes with eigenfrequencies up to twice the eigenfrequencies which want to be approximated in the full system are stacked into $\boldsymbol{V}_{CB}$.

Besides the Guyan and the Craig-Bampton substructuring technique, various other techniques proposed by and named after MacNeal[Mac71] or Rubin [Rub75] have been developed to solve the problem of reducing substructures, in which the boundary dofs are retained. The method proposed by Ocallahan et al. [OAR89] is specially pointed out, since it allows to partition an arbitrary given basis $\boldsymbol{V}$ into boundary dofs and internal dofs. For the sake of completeness it is mentioned that not only a primal coupling for substructuring is possible, but also dual methods exist. In these methods, the displacement continuity is not enforced implicitly with the assembly process, but explicitly with the use of constraints enforced with Lagrange multipliers. Further information on the dual substructuring methods can be retrieved in Rixen [Rix04] and Gruber et al. [GRR16]. For substructuring itself, the reading of the extensive textbooks of Géradin and Rixen [GR14] and Craig [Cra81] and of the references therein is recommended.

Since substructuring goes hand in hand with model order reduction, many developments in model order reduction were transferred to substructuring applications and vice versa. In this thesis, the focus is the reduction of nonlinear uncoupled structures, which are not coupled with other structures. Many concepts in nonlinear model order reduction and especially hyper-reduction can be applied with substructuring or transferred to it. This task goes, however, beyond the scope of this thesis.

## 4.6 System Theoretic Approaches

Model order reduction is also broadly in use in the realm of control, since fast and accurate models are necessary for real-time controllers. However, the common formulations of dynamical systems and also the requirements of these systems differ to the ones in structural dynamics. As a consequence, the concepts of model order reduction in systems theory and structural dynamics differ. Nonetheless, many methods of the two fields have their counterpart or at least a link to a method of the other field. Hence, one link is drawn to the Krylov subspace method discussed in Section 4.4.

### 4.6.1 Conceptual Differences

In control theory, dynamical systems are typically described and analyzed differently than in structural dynamics. In the latter, the dynamics of the system is described using a second order ordinary differential equation with symmetric and positive (semi) definite mass, damping and stiffness matrices and the displacements as the primal variable (cf. (4.1)). In control theory, the dynamical system is commonly described as a state-space system, where both displacements and velocities form the unknown variables of the state vector $\boldsymbol{x}$:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_{ss}\boldsymbol{x} + \boldsymbol{B}_{ss}\boldsymbol{u}_{ss}, \tag{4.38}$$

$$\boldsymbol{y}_{ss} = \boldsymbol{C}_{ss}\boldsymbol{x}, \tag{4.39}$$

which can be realized as

$$\begin{aligned} \boldsymbol{x} &= \begin{pmatrix} \boldsymbol{u} \\ \dot{\boldsymbol{u}} \end{pmatrix} \in \mathbb{R}^{2N}, & \boldsymbol{A}_{ss} &= \begin{pmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{C} \end{pmatrix} \in \mathbb{R}^{2N\times 2N}, \\ \boldsymbol{B}_{ss} &= \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1}\boldsymbol{G} \end{pmatrix} \in \mathbb{R}^{2N\times p}, & \boldsymbol{u}_{ss} &= \hat{\boldsymbol{g}} \in \mathbb{R}^{p}. \end{aligned} \tag{4.40}$$

The subscript $ss$ denotes state-space matrices and vectors. The system matrix $\boldsymbol{A}_{ss}$ describes the dynamics and can be constructed using the mass, damping and stiffness matrix of the second order mechanical system. It is not symmetric and non positive definite. The input locations are expressed with the input location matrix $\boldsymbol{B}_{ss}$ and the external excitation $\boldsymbol{u}_{ss}$ is equal to the external forcing $\hat{\boldsymbol{g}}$.

The state-space description in the use of structural dynamics has the drawback that the physical interpretation of a state-space vector is lost to a certain extent. While the displacements $\boldsymbol{u}$ are associated to a displacement field, the state-space vector $\boldsymbol{x}$ describes both a displacement field with a velocity field, which makes general interpretations difficult.

The reduction of a state-space system is performed with the expression of the state-space vector $\boldsymbol{x}$ by a combination of basis vectors gathered in $\boldsymbol{V}_{ss} \in \mathbb{R}^{2N\times 2n}$ and the reduced state-space coordinates $\boldsymbol{x}_r \in \mathbb{R}^{2n}$ with $n \ll N$:

$$\boldsymbol{x} = \boldsymbol{V}_{ss}\boldsymbol{x}_r. \tag{4.41}$$

Similar to (3.3), the application of the transformation to the dynamics equations leads to a residual $\boldsymbol{r}_{ss}$:

$$\boldsymbol{V}_{ss}\dot{\boldsymbol{x}}_r = \boldsymbol{A}_{ss}\boldsymbol{V}_{ss}\boldsymbol{x}_r + \boldsymbol{B}_{ss}\boldsymbol{u}_{ss} + \boldsymbol{r}_{ss}. \tag{4.42}$$

Again, as in (3.4), a constraint for the residual has to be found in order to make the equations above determined. However, the physical units of the equations above are not of one type like in second order systems, where the equations of motion are force equations. With the realization (4.40), for instance, the units in (4.38) are both velocities and accelerations. Since the constraints have to be applied to both units, the handling of constraints is more complicated for first order state-space systems than for second order systems. In the latter, the principle of virtual work requires the residual forces to be orthogonal to the kinematically admissible displacements resulting in the symmetric projection discussed in Section 3.2. In the former, the orthogonality is enforced with a second matrix $\boldsymbol{W}_{ss}$, which is defined by the chosen first order reduction method:

$$\boldsymbol{W}_{ss}^{T}\boldsymbol{r}_{ss} = \boldsymbol{0}. \tag{4.43}$$

With applying (4.43) to (4.42), the following reduced state-space system is obtained:

$$\dot{\boldsymbol{x}}_r = \boldsymbol{A}_{ss,r}\boldsymbol{x}_r + \boldsymbol{B}_{ss,r}\boldsymbol{u} \tag{4.44}$$

$$\boldsymbol{y}_{ss} = \boldsymbol{C}_{ss,r}\boldsymbol{x}_r \tag{4.45}$$

with

$$\boldsymbol{A}_{ss,r} = (\boldsymbol{W}_{ss}^T\boldsymbol{V}_{ss})^{-1}\boldsymbol{W}_{ss}^T\boldsymbol{A}_{ss}\boldsymbol{V}_{ss} \in \mathbb{R}^{2n\times 2n}, \tag{4.46}$$

$$\boldsymbol{B}_{ss,r} = (\boldsymbol{W}_{ss}^T\boldsymbol{V}_{ss})^{-1}\boldsymbol{W}_{ss}^T\boldsymbol{B}_{ss} \in \mathbb{R}^{2n\times p}, \tag{4.47}$$

$$\boldsymbol{C}_{ss,r} = \boldsymbol{C}_{ss}\boldsymbol{V}_{ss} \in \mathbb{R}^{m\times 2n}. \tag{4.48}$$

Note, that in general $\boldsymbol{V}_{ss}$ and $\boldsymbol{W}_{ss}$ are not equal. In the projective model order reduction for second order mechanical systems, the principle of virtual work requires the left and right sided projection to be equal. When the principle of virtual work is not violated, the mechanical system properties like stability, passivity and symmetry are preserved. This makes second order reduction an easy task, since any non-singular basis $\boldsymbol{V}$ preserves all properties associated with mechanical systems. In a reduction scheme for first order systems, however, the preservation of the mechanical properties like stability or passivity is not genuinely given and has to be investigated and proved for every method.

In control theory, commonly the input-output behavior of a system is of interest while the approximation of the internal state variable $\boldsymbol{x}$ is not important. In structural dynamics, on the contrary, the full displacement field is usually of interest, since very often stress distributions are sought, e.g., for fatigue analyses where the critical location is not known beforehand. Consequently, it is not the approximation of the input-output behavior which matters most in structural dynamics, but the accurate reconstruction of the full displacement field of the system.

Due to these differences, most projective reduction techniques from system theory cannot be directly transferred to the realm of structural dynamics. However, several links exist, for which the link between the moment matching technique and the Krylov subspace technique introduced in Section 4.4 is drawn in the next subsection.

### 4.6.2 Moment Matching and Krylov Subspaces

As discussed above, the input-output behavior is the key property in the system theory reduction methods. The transfer function $\boldsymbol{H}(s)$ describes the input-output behavior in the Laplace domain. For the state-space system (4.38) it is defined as

$$\boldsymbol{H}(s) = \frac{\boldsymbol{Y}_{ss}(s)}{\boldsymbol{U}_{ss}(s)} = \boldsymbol{C}_{ss}(s\boldsymbol{I} - \boldsymbol{A}_{ss})^{-1}\boldsymbol{B}_{ss} \tag{4.49}$$

with the Laplace transform $\boldsymbol{U}_{ss}(s)$ of the input vector $\boldsymbol{u}_{ss}(t)$ and the Laplace transform $\boldsymbol{Y}_{ss}(s)$ of the output vector $\boldsymbol{y}_{ss}(t)$.

The transfer function can be expanded as a power series around the point $s_0$ in the Laplace domain (cf. [Ant09, p. 345]) yielding the so-called *moments* of the transfer function:

$$\boldsymbol{H}(s) = \boldsymbol{H}(s_0) + \boldsymbol{H}^{(1)}(s_0)\frac{(s-s_0)}{1!} + \cdots + \boldsymbol{H}^{(k)}(s_0)\frac{(s-s_0)^k}{k!} + \ldots \tag{4.50}$$

$$= \boldsymbol{T}_0 + \boldsymbol{T}_1(s-s_0) + \cdots + \boldsymbol{T}_k(s-s_0)^k + \ldots \tag{4.51}$$

The polynomial coefficients $\boldsymbol{H}(s_0) = \boldsymbol{T}_0(s_0)$, $\boldsymbol{H}^{(1)}(s_0) = \boldsymbol{T}_1(s_0), \ldots, \boldsymbol{H}^{(k)}/k! = \boldsymbol{T}_k(s_0)$ are called zeroth, first or $k$-th Moment of the transfer function about the point $s_0$. They can be derived as [Ant09, p. 345]:

$$\boldsymbol{T}_k = \boldsymbol{C}_{ss}(s_0\boldsymbol{I} - \boldsymbol{A}_{ss})^{-(k+1)}\boldsymbol{B}_{ss}. \tag{4.52}$$

The goal of model order reduction using moment matching is that the first $k$ moments of the reduced system about $s_0$ are equal to the first $k$ moments about $s_0$ of the full, unreduced system. Grimme [Gri97] showed that the first $k$ moment are matched when the left and right sided projection matrices $\boldsymbol{W}_{ss}$ and $\boldsymbol{V}_{ss}$ span the subspace of the following Krylov subspace:

$$\boldsymbol{V}_{ss,\text{raw}} = \left((s_0\boldsymbol{I} - \boldsymbol{A}_{ss})^{-1}\boldsymbol{B}_{ss}, \ldots, (s_0\boldsymbol{I} - \boldsymbol{A}_{ss})^{-(k+1)}\boldsymbol{B}_{ss}\right) \tag{4.53}$$

$$\boldsymbol{W}_{ss,\text{raw}} = \left((s_0\boldsymbol{I} - \boldsymbol{A}_{ss}^T)^{-1}\boldsymbol{C}_{ss}^T, \ldots, (s_0\boldsymbol{I} - \boldsymbol{A}_{ss}^T)^{-(k+1)}\boldsymbol{C}_{ss}^T\right) \tag{4.54}$$

The subspace spanned by $\boldsymbol{V}_{ss}$ is the so-called input Krylov subspace and the subspace spanned by $\boldsymbol{W}_{ss}$ is the so-called output Krylov subspace.

The concept of moment matching can also be applied to second order systems without damping. When the full displacements $\boldsymbol{u}$ of the second order system are of interest, the transfer function of the input vector $\hat{\boldsymbol{g}}$ to the displacement field $\boldsymbol{u}$ is given as

$$\boldsymbol{H}(s) = \left(s^2\boldsymbol{M} + \boldsymbol{K}\right)^{-1}\boldsymbol{G} \tag{4.55}$$

$$= \sum_{i=0}^{\infty} \underbrace{\left[(\boldsymbol{K} + s_0^2\boldsymbol{M})^{-1}\boldsymbol{M}\right]^i (\boldsymbol{K} + s_0^2\boldsymbol{M})^{-1}\boldsymbol{G}}_{\boldsymbol{T}_i(s_0)}(s^2 - s_0^2)^i \tag{4.56}$$

with the force input location matrix $\boldsymbol{G}$. The power series above yield the moments of the second order system, which are matched (cf. [Bes+13]) when the second order projection is performed with the following Krylov subspace:

$$\boldsymbol{V} = \left((\boldsymbol{K} + s_0^2\boldsymbol{M})^{-1}\boldsymbol{G}, \ldots, \left[(\boldsymbol{K} + s_0^2\boldsymbol{M})^{-1}\boldsymbol{M}\right]^k (\boldsymbol{K} + s_0^2\boldsymbol{M})^{-1}\boldsymbol{G}\right). \tag{4.57}$$

Note, that this subspace is equal to the Krylov subspace stemming from the consideration in Section 4.4, when $s_0 = i\omega_0$. Hence, the Krylov subspace technique discussed in Section 4.4 matches the first $2k$ moments of the transfer function of the undamped mechanical system. In the presence of damping, second order methods to build an appropriate Krylov subspace exist, e.g., Salimbahrami and Lohmann [SL06], Bai and Su [BS05] or Lehner and Eberhard [LE06].

For building the Krylov subspace, the expansion point $s_0$ can be chosen. In structural dynamics, it is common to choose $s_0 = 0$ (cf. [Rix01], [GR14]), since the static approximation and the low frequency range is of great interest. When the input-output behavior is addressed, however, the accuracy of the Krylov reduction basis can be increased by a better choice of the expansion points $s_0$. One popular strategy is the so-called IRKA method proposed by Gugercin et al. [GAB08], which has been modified and improved in various ways. The interested reader is referred to the work of Panzer [Pan14], Wolf [Wol14], Beattie and Gugercin [BG16] and the references therein.

### 4.6.3 Further Approaches

Besides the Krylov subspace method, further system theory inspired methods have been developed. The most prominent method is probably the Balanced Truncation method first proposed by Mullis and Roberts [MR76] and then generalized by Moore [Moo81] for general linear, time invariant systems, where the concept of controllability and observability is exploited to rank the state variables. The reduction basis is then built with the states which are both well controllable and observable, and the remaining states are truncated.

This concept has been successfully applied to systems in control, elastic multibody systems and other domains like electrical engineering, e.g., [FE10], [Now+13], [GA04]. With the high computational costs associated with the solution of a Lyapunov equation, this method is limited to small size systems, where computations of systems with more than 2,500 displacement dofs and hence 5,000 states become tough [Wol14, pp. 7f.].

Even though the method was extended to second order systems by Chahlaoui et al. [Cha+06] and Reis and Stykel [RS08], it is merely used in structural dynamics. This might also be due to the fact, that the modal basis yields systems, which are called almost balanced, see Jonckheere [Jon84], Davis and Skelton [DS84], Gawronski and Lim [GL96] or Gawronski [Gaw97; Gaw04; Gaw06]. The Balanced Truncation method converges to the Modal reduction method when the system is very lightly damped, the eigenfrequencies are separated enough and the input and output locations are collocated, which is the case for many metallic structures.

As mentioned above, many other model order reduction methods like the Hankel-Norm approximation or Krylov-SVD methods were developed in the realm of systems theory. The interested reader is referred to the textbook of Antoulas [Ant05] and the references therein.

# Chapter 5

# Proper Orthogonal Decomposition (POD)

In the previous chapter, a few reduction methods for linear structural dynamical systems are discussed. Since these systems are linear, they are characterized by a mass, stiffness and damping matrix and sometimes by input and output locations. The reduction methods are hence explicitly or implicitly based on system theoretic concepts. For instance, the decoupling of states results in modal truncation, the approximation of the transfer function results in the Krylov subspace methods with moment matching, or the concept of controllability and observability results in Balanced Truncation or – for slightly damped systems – in modal truncation with modal dominance ranking.

Nonlinear systems lack these systems theoretic concepts in most cases. They are usually not defined for nonlinear systems or, if they are defined, they are extremely expensive to compute. Hence, the concept of finding a reduced basis $\boldsymbol{V}$ for a nonlinear system is usually conceptually different to the strategies of finding a reduced basis for a linear system. Since nonlinear systems cannot be fully described by system matrices, a detour over data-driven methods is a common approach. This means that training simulations of the full, high dimensional system are run in a first step, of which the results are analyzed in a second step to build the projection matrix $\boldsymbol{V}$. The most prominent method is the Proper Orthogonal Decomposition (POD), which is discussed in this chapter.

According to Berkooz et al. [BHL93], the POD was developed independently by a number of scientists, e.g., by Karhunen [Kar46] and Loève [Loè48], after whom the procedure is given the name *Karhunen-Loève decomposition*, but also Kosambi [Kos43], Pougachev [Pug53] and Obukhov [Obu54]. It was first used widely in fluid dynamics to identify coherent structures [Sir87] and is now widely used in the field of model order reduction for turbulent flows, see, e.g., [HTD00; WP02; Epu03; Lum07; Hol12]. Furthermore, it is used in the control of dynamical systems [Rav00; AM02; BCB05], damage detection [DG03; LD06; Muj+11], finite element model updating [HD01; LKG03; KAS07] and of course in model order reduction of structural dynamics, e.g., [AV99; KF99; AV01; KLM01; ST01; Lia+02b; AP03; TWS05; GSD08], besides many other domains. For a more detailed overview of the POD and its history, the publications of Kerschen et al. [Ker+05], Liang et al. [Lia+02a] and Berkooz [Ber92] are recommended.

## 5.1 Key Idea: Principal Component Analysis

The POD is a data driven method, which identifies the optimal subspace from a set of displacement snapshots $\boldsymbol{u}_s$. This task is related to the principal component analysis (PCA) used in statistics, where the so-called principal components are sought, which minimize the Euclidean distance to a set of data points. In the POD, the displacement snapshots $\boldsymbol{u}_s$ are the data points, for which the subspace $\boldsymbol{V}$ is sought to fulfill the following minimization problem:

$$\min \sum_{i=1}^{m} ||\boldsymbol{u}_s(t_i) - \boldsymbol{V}\boldsymbol{q}(t_i)||_2. \tag{5.1}$$

In this minimization problem, the subspace $\boldsymbol{V} \in \mathbb{R}^{N\times n}$ of predefined dimension $n$ is searched for, while the vector of amplitudes $\boldsymbol{q}(t_i) \in \mathbb{R}^n$ is adapted to minimize the expression above. The solution of this minimization problem can be obtained with the singular value decomposition (SVD), the underlying mathematical operation of the PCA and the POD. It identifies the subspace $\boldsymbol{V}$ fulfilling (5.1) when applied to the snapshot matrix $\boldsymbol{S}_d \in \mathbb{R}^{N\times m}$, in which the displacements of all $m$ sampled time steps $\boldsymbol{u}_s$ are gathered:

$$\boldsymbol{S}_d = \big(\boldsymbol{u}_s(t_1), \dots, \boldsymbol{u}_s(t_m)\big) = \boldsymbol{U}_{\text{svd}}\boldsymbol{\Sigma}\boldsymbol{V}_{\text{svd}}^T. \tag{5.2}$$

The SVD of the snapshot matrix $\boldsymbol{S}_d$ yields the orthogonal matrix $\boldsymbol{U}_{\text{svd}} \in \mathbb{R}^{N\times m}$ containing the left singular vectors, the diagonal matrix $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_m)$ containing the singular values $\sigma_i$, which are ordered decreasingly with $\sigma_i \geq \sigma_{i+1}$ and the orthogonal matrix $\boldsymbol{V}_{\text{svd}} \in \mathbb{R}^{m\times m}$ containing the right singular vectors. The declared dimensions are only valid for the case, in which the number of snapshots $m$ is smaller than the number of dofs $N$. This is the case in most reduction settings, when large models are considered. In the other case for $m > N$, the dimensions of the matrices are $\boldsymbol{U}_{\text{svd}} \in \mathbb{R}^{N\times N}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{V}_{\text{svd}} \in \mathbb{R}^{N\times m}$.

The left singular vectors $\boldsymbol{u}_{\text{svd},i}$ form the principal components of the displacement field set gathered in $\boldsymbol{S}_d$. For a given dimension $n$, the first $n$ principal components fulfill the optimality condition (5.1) and minimize the Euclidean error. Consequently, the POD builds the basis $\boldsymbol{V}$ using the first $n$ left sided singular vectors $\boldsymbol{u}_{\text{svd},i}$. Since both the left and the right singular vectors are normalized, the amplitude for reconstructing the displacement snapshots is stored in the singular values $\sigma_i$. Thus, the importance of the $i$-th principal component $\boldsymbol{u}_{\text{svd},i}$ is associated with the value of the corresponding singular value $\sigma_i$.

While the left singular vectors $\boldsymbol{u}_{\text{svd},i}$ carry the information of the dominating spatial displacement fields, the right singular vectors $\boldsymbol{v}_{\text{svd},i}$ composing $\boldsymbol{V}_{\text{svd}}$ describe the temporal evolution of them. Consequently, the SVD of the snapshots does not only provide the dominant subspace of the motion but also the temporal information, which can be used in applications such as model updating as reported by Hemez and Deobling [HD01] or Lenaerts et al. [LKG03].

The minimization (5.1) can also be expressed in matrix notation, where the SVD identifies an optimal low rank approximation $\boldsymbol{S}_{d,\text{approx}}$ with rank $n$ of the snapshot matrix $\boldsymbol{S}_d$. This optimal approximation is obtained when the $L^2$ norm or the Frobenius norm of the difference between $\boldsymbol{S}_d$ and $\boldsymbol{S}_{d,\text{approx}}$ is minimal. The $L^2$ error $\epsilon_2$ and the Frobenius error $\epsilon_f$ are given as:

$$\epsilon_2 = ||\boldsymbol{S}_{d,\text{approx}} - \boldsymbol{S}_d||_2, \qquad \epsilon_f = ||\boldsymbol{S}_{d,\text{approx}} - \boldsymbol{S}_d||_f. \tag{5.3}$$

According to the Eckart-Young-Mirsky theorem (cf. [GV12]), both errors $\epsilon_2$ and $\epsilon_f$ are minimized at the same time, when the approximation matrix $\boldsymbol{S}_{d,\text{approx}}$ is built from a sum of $n$ rank one matrices. Each of them is defined as the outer product of the normal vectors $\boldsymbol{u}_{\text{svd},i}$ and $\boldsymbol{v}_{\text{svd},i}$ and a weighting factor $\sigma_i$ being exactly the left and right singular vectors and the corresponding singular value stemming from the SVD:

$$\boldsymbol{S}_{d,\text{approx}} = \sum_{i=1}^{n} \sigma_i \boldsymbol{u}_{\text{svd},i}\boldsymbol{v}_{\text{svd},i}^T. \tag{5.4}$$

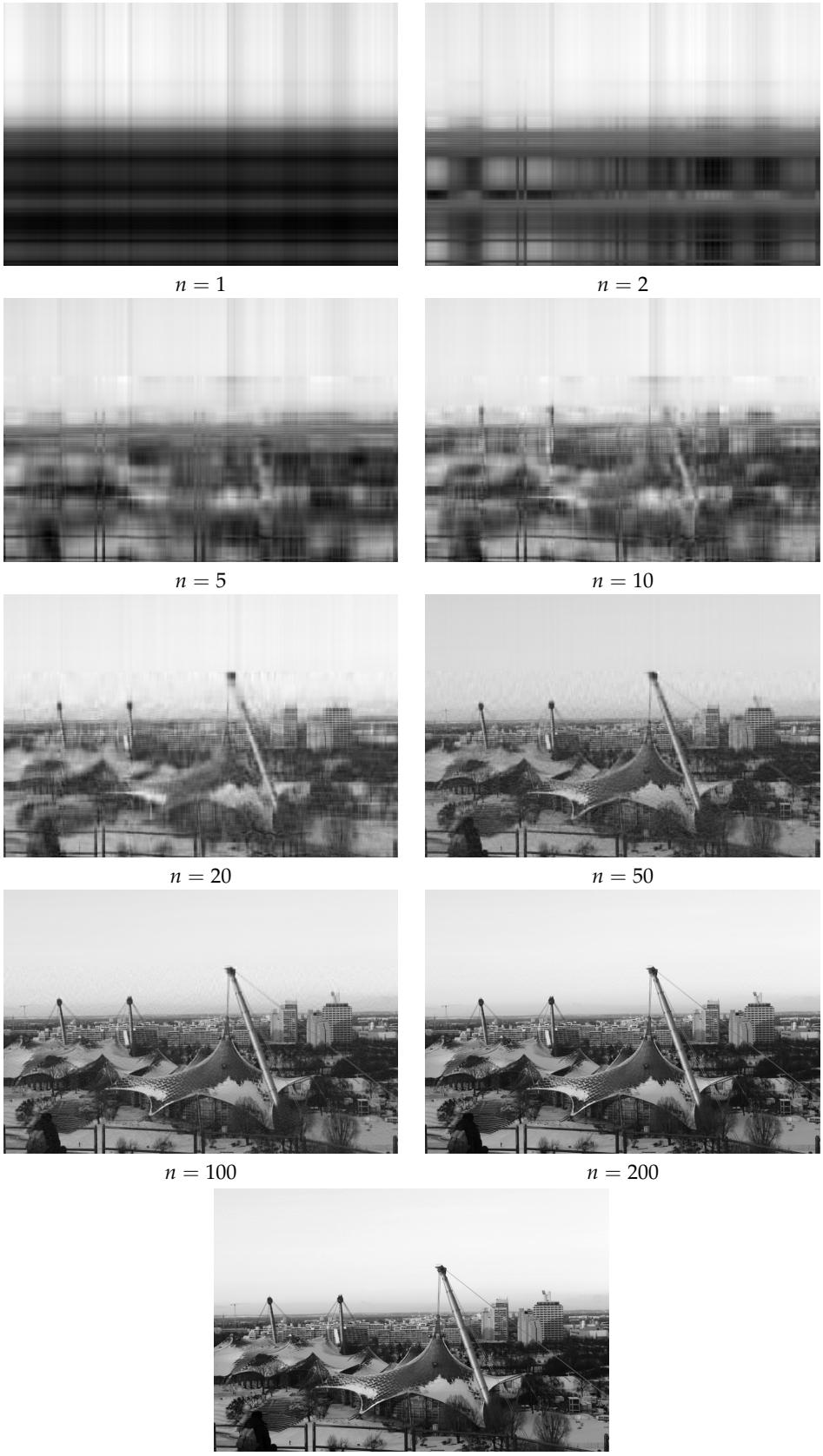


**Figure 5.1:** Low rank approximation of the $600 \times 900$ bitmap picture with different ranks $n$. Picture from the author.

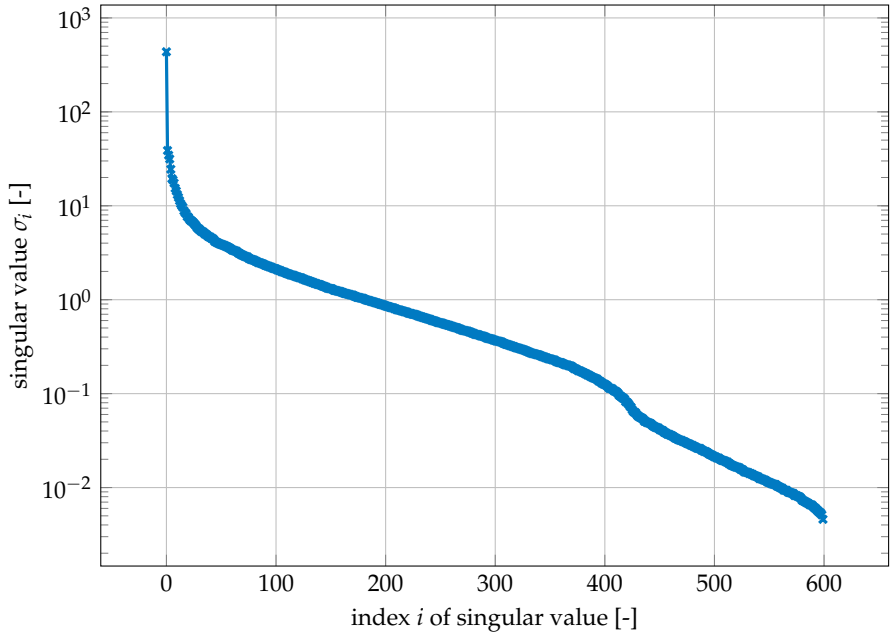


**Figure 5.2:** Singular values of the picture shown in Figure 5.1

This minimality principle is illustrated with a bitmap picture depicted in Figure 5.1, inspired by Antoulas et al. [ASG06]. The $600 \times 900$ bitmap picture represents the snapshot matrix $\boldsymbol{S}_d$ with $N = 600$ dofs and $m = 900$ snapshots. The columns of the picture represent the single snapshots $\boldsymbol{u}_s$ and the grayscale value of a pixel represents the displacement magnitude of a single dof. With the SVD of $\boldsymbol{S}_d$ stemming from the picture, a low rank approximation is built for different $n$ according to (5.4). Figure 5.1 depicts the low rank approximations for $n = 1, 2, 5, 10, 20, 50, 100$ and 200. The singular values of the picture are depicted in Figure 5.2. They indicate the convergence of the series (5.4), i.e., quantitatively indicate, how well the original picture is matched with the low rank approximation. This decay of the singular values in the context of the POD is an excellent indicator of the complexity of the system's dynamics. A rapid decay indicates that few singular vectors can represent the relevant motion of the high dimensional system and consequently the size $n$ of the basis $\boldsymbol{V}$ can be chosen small. On the contrary, when the decay of the singular values is slow, many basis vectors are necessary to span the subspace of the relevant motion requiring $n$ being relatively large. Hence, the analysis of the singular values provides a good measure of how the size of the reduced order $n$ can be chosen.

## 5.2 Variants and Improvements

The SVD carries the property that the left singular vectors $\boldsymbol{u}_{\text{svd},i}$ of the snapshot matrix $\boldsymbol{S}_d$ and the squared singular values $\sigma_i^2$ are equivalent to the eigenvectors and eigenvalues of the covariance matrix of the snapshot matrix $\boldsymbol{S}_d$:

$$\boldsymbol{S}_d \boldsymbol{S}_d^T \boldsymbol{u}_{\text{svd},i} = \sigma_i^2 \boldsymbol{u}_{\text{svd},i}. \tag{5.5}$$

Since, by construction, the covariance matrix is symmetric and positive semi-definite, the eigenvectors are orthogonal and the singular values $\sigma_i$ are greater or equal to zero. Hence, the POD can be seen as the solution of the eigenvalue problem of the covariance matrix of the displacements. With this interpretation in mind, two variants of the POD are discussed in the following.

### 5.2.1 Smooth Orthogonal Decomposition

The first variant is the smooth orthogonal decomposition (SOD) proposed by Chelidze and Zhou [CZ06]. In this approach, not only the displacement snapshots collected in $\boldsymbol{S}_d$ are analyzed but also the velocity snapshots $\boldsymbol{S}_v$. They are obtained from the displacement snapshots using a finite difference scheme, which can be expressed as $\boldsymbol{S}_v = \boldsymbol{D}\boldsymbol{S}_d$ with the finite difference matrix $\boldsymbol{D}$ for equally spaced time increments $\Delta t$ between the snapshots:

$$\boldsymbol{D} = \frac{1}{\Delta t}\begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix}. \tag{5.6}$$

The smooth orthogonal modes (SOMs) $\boldsymbol{u}_{\text{som}}$ are then defined as the generalized eigenvalue problem of the covariance matrix of the displacements and the covariance matrix of the velocities with the smooth orthogonal values $\lambda_i$:

$$\boldsymbol{S}_d\boldsymbol{S}_d^T\boldsymbol{u}_{\text{som},i} = \lambda_i\boldsymbol{S}_v\boldsymbol{S}_v^T\boldsymbol{u}_{\text{som},i}. \tag{5.7}$$

Whereas in the classical POD, the regular eigenvalue problem of the displacement covariance matrix identifies the dominating displacements, an additional property is accounted for in the smooth POD. In this generalized eigenvalue problem, the solution vectors associated with large $\lambda_i$ are both dominant in the displacement covariance matrix and non dominant in the velocity covariance matrix. This non-dominance in the velocity matrix can be interpreted as smoothness meaning little fluctuation with respect to time. Consequently, the smooth POD ranks the smooth orthogonal modes according to both its dominance in the displacements and at the same time to its smoothness with time.

For linear slightly damped systems, the smooth orthogonal modes $\boldsymbol{u}_{\text{som},i}$ correspond to the vibration modes of the undamped system and the smooth orthogonal values $\lambda_i$ correspond to the inverse squared eigenfrequencies $1/\omega_i^2$. Hence, the smooth orthogonal decomposition is an attractive tool for the modal analysis of linear systems, especially measurements where the mass matrix is not known. For example, Farooq and Feeny [FF08a] use this method for the linear modal analysis of randomly excited structures and Rezaee et al. [RSB13] for the identification of modal parameters in suspension systems.

In the field of nonlinear model order reduction, however, the performance is in general not superior compared to the regular POD, as reported by Przekop et al. [PGR12] and Lülf et al. [LTO13]. The computation of the smooth orthogonal decomposition is clearly more expensive than the POD, since a dense generalized eigenvalue problem has to be solved instead of a regular eigenvalue problem. Probably, this is the reason why the popularity of the method in the realm of nonlinear model order reduction is moderate compared to the classical POD.

### 5.2.2 Weighted POD

The second variant of the POD discussed here is a form of a weighted POD, as used, e.g., by Guo [Guo11]. As stated in (5.1), the POD identifies the subspace minimizing the error of the snapshots in the Euclidean norm. This norm is not always optimal, since it overestimates areas where the nodes of the mesh are dense and underestimates coarse areas. Furthermore, it is not appropriate for displacement fields comprised of different physical units like displacements and rotations. Hence, different norms for enforcing the optimality in the low rank approximation are often attractive.

One popular choice, for instance, is the mass norm

$$||\boldsymbol{u}||_M = \sqrt{\boldsymbol{u}^T \boldsymbol{M} \boldsymbol{u}} \tag{5.8}$$

which eliminates the mentioned drawbacks of the Euclidean norm. The idea of the weighted POD is the substitution of the $L^2$ norm in (5.1) with a norm like the $\boldsymbol{M}$-norm above.

With the Cholesky factorization of the mass matrix $\boldsymbol{M}$

$$\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T, \tag{5.9}$$

the physical displacements $\boldsymbol{u}$ can be transformed to the displacements

$$\widetilde{\boldsymbol{u}} = \boldsymbol{L}^T \boldsymbol{u}. \tag{5.10}$$

This transformation changes the $\boldsymbol{M}$ norm to the $L^2$ norm

$$\boldsymbol{u}^T \boldsymbol{M} \boldsymbol{u} = \boldsymbol{u}^T \boldsymbol{L}\boldsymbol{L}^T \boldsymbol{u} = \widetilde{\boldsymbol{u}}^T \widetilde{\boldsymbol{u}} \tag{5.11}$$

enabling the computation of the standard SVD with the transformed snapshot matrix $\widetilde{\boldsymbol{S}}_d$:

$$\boldsymbol{L}^T \boldsymbol{S}_d = \widetilde{\boldsymbol{S}}_d = \widetilde{\boldsymbol{U}}_{\text{svd}} \widetilde{\boldsymbol{\Sigma}} \widetilde{\boldsymbol{V}}_{\text{svd}}. \tag{5.12}$$

The mass scaled left sided singular vectors $\boldsymbol{U}_{\text{svd},M}$ are obtained by the transformation back to the physical coordinates, which are obtained with the backward substitution of the Cholesky factorization $\boldsymbol{L}$ of the mass matrix:

$$\boldsymbol{L}^T \boldsymbol{U}_{\text{svd},M} = \widetilde{\boldsymbol{U}}_{\text{svd}}. \tag{5.13}$$

In the weighted POD, the reduced basis $\boldsymbol{V}$ is built with the first $n$ mass scaled left sided singular vectors $\boldsymbol{u}_{\text{svd},M,i}$. They are ranked with the mass-weighted singular values $\sigma_{M,i}$, which represent the mass associated with the displacements $\boldsymbol{u}_{\text{svd},M,i}$ in the snapshot matrix. Since the mass scaling equalizes the dofs of different units and different discretizations and gives it a model-based physical meaning, it is the recommended method for inhomogeneous models. It comes with the additional cost of the factorization of the mass matrix. For the sake of generality, the associated eigenvalue problem is given as

$$\boldsymbol{L}^T \boldsymbol{S}_d \boldsymbol{S}_d^T \boldsymbol{L} \boldsymbol{u}_{\text{svd},M,i} = \sigma_{M,i}^2 \boldsymbol{L}^T \boldsymbol{u}_{\text{svd},M,i}, \tag{5.14}$$

where once again the strategy of the transformation of the Euclidean space to the $\boldsymbol{M}$ space is clearly visible. The procedure is, of course, not limited to the $\boldsymbol{M}$-norm but can be applied to any other norm, as long as the matrix $\boldsymbol{M}$ is positive definite.

## 5.3 Advantages and Drawbacks

The POD is, as mentioned above, a data-driven method and hence independent of the underlying system. Thus, the POD offers a maximum of flexibility and applicability, since it can be applied to any model of any physical domain where representative simulations of the high dimensional model are possible. This is one of the reasons why the POD is so widespread also beyond applications in structural dynamics. To mention a few applications, it has been applied in fluid dynamics and aero-elasticity [TDH03; CDM15; AF08; XXD14; XX15], electrochemical applications [CW09], welding [CCI14; BMS16], and elasto-plastic problems [RR14a; CDM15]. Furthermore, the singular values offer a good insight into the complexity of the model, allowing for an estimation of the error that can be assumed to correspond with the singular values which are not included.

The POD is a method which is also not only applied to nonlinear systems. As mentioned in the subsection above, especially the smooth POD is a great method for system

identification of linear systems in both experimental and simulation setups. Intrinsic properties of linear systems like vibration modes can be identified under certain circumstances. Literature discussing the relation of linear system properties and the POD are for instance the publications of Feeny and Kappaguntu [FK98], Kim [Kim98], Kerschen and Golinval [KG02], Feeny and Liang [FL03] or Feeny and Farooq [FF08b].

The necessity to require a high dimensional training simulation is one of the major drawbacks though. In many applications, these immense offline costs are not affordable or even not feasible if the computational resources are not available. Especially in single query scenarios, where the number of reduced runs is limited, the POD is often not attractive. However, in cases, where the offline costs do not matter like applications in control or scenarios with many reduced runs, the POD is a very competitive choice.

Another drawback is associated with the inherent nature of the method as a statistical method. Since only the output data of the high dimensional model are analyzed, only the states which are triggered in the training simulation are incorporated into the reduced basis. If model parameters, boundary conditions or excitations change, the snapshots obtained by the changed model can be different. Hence, the POD is very sensitive to the parameters and especially the excitations of the training simulation. As often different load cases compared to the trained ones should be simulated with the reduced model, it is necessary to design training simulations capturing the full dynamics, which should be represented by the reduced model. Hence, the generation of good snapshots remains an open topic. The drawbacks of the POD mentioned above are motivation for so-called *simulation-free* reduction methods, which are not based on training sets requiring full simulation runs. They are discussed in the following chapter.

# Chapter 6

# Simulation-Free Approaches

The previous two chapters discuss the construction of reduced bases for linear and nonlinear systems. The methods for generating a reduced basis for linear systems use invariant physical properties characterizing the system like the mass and the stiffness matrix. On the contrary, the state-of-the-art POD method uses the data generated by training simulations, since invariant and meaningful physical properties are generally hard to retrieve for arbitrary nonlinear systems.

For nonlinear structural systems, especially for geometric nonlinearity, the generation of reduced bases in a non-POD fashion is possible, though. The key idea is the exploitation of the physical structure of the system similar to the linear reduction methods, however with capturing the information about the nonlinearity. Hence, these methods rely on the linearized system and extend the operation to the nonlinear regime via information from perturbation techniques.

The construction of reduced bases for nonlinear structural dynamics goes back to Noor and Peters [NP80] and Noor [Noo82], where so-called path derivatives are proposed and applied to static problems. The subspace obtained by triggering the nonlinearity using displacement fields as perturbation parameters is successfully used for the augmentation of the basis obtained from linear methods. Since then, several approaches have been proposed which differ often only in details. Idelsohn and Cardona [IC85b] propose the so-called modal derivatives, where the eigenvalue problem is perturbed to obtain the derivatives of vibration modes of a nonlinear system, which are applied successfully for the reduction of nonlinear dynamical problems. In [IC85a] they extend the method to other types of vectors. Perturbation techniques for reduction in the realm of structural dynamics are further used by, e.g., Chang and Engblom [CE91], Jacob and Ebecken [JE92], Noor et al. [NAP93], Bauchau and Guernsey [BG93], Slaats et al. [SdS95], Tiso and Rixen [TR11], Tiso et al. [TJA11] and Weeger et al. [WWS14; WWS16]. Recently, methods based on this principle were extended to substructuring applications by Wennecker and Tiso [WT14], to localized nonlinearities by Witteveen and Pichler [WP14] and to elastic multibody dynamics by Tiso and Wu [WT16b]. Selection strategies are proposed by Tiso in [Tis11]. The concept of derivatives is further successfully used in computer graphics, e.g., by Barbič and James [BJ05] or Hildebrand et al. [Hil+11]. In the former, the concept of modal derivatives is combined with a polynomial hyper-reduction technique to obtain further speedups. Tycowicz et al. [Tyc+13] propose an alternative method to obtain spatial augmentation modes based on rotational properties, which are similar in performance to modal derivatives.

In the following, the necessity for augmentation of linear bases in the use for structural dynamics is illustrated and the concept of static and modal derivatives is discussed. Furthermore, strategies to handle the rank deficiency, the quadratically increasing number of derivatives and the computation of the derivatives are given before the methods

are applied to two selected examples for performance assessment.

## 6.1 Motivating Example

A naive approach to reduce a nonlinear system is to linearize the system, apply a linear reduction method and use the reduced basis obtained this way to reduce the nonlinear system. These reduced bases perform generally very poorly for geometrically nonlinear systems as is shown for the cantilever introduced in Section 2.5 and depicted in Figure 2.3. This model is linearized for obtaining the reduced basis using both the modal truncation and the Krylov subspace method with a reduced basis of $n = 5$.

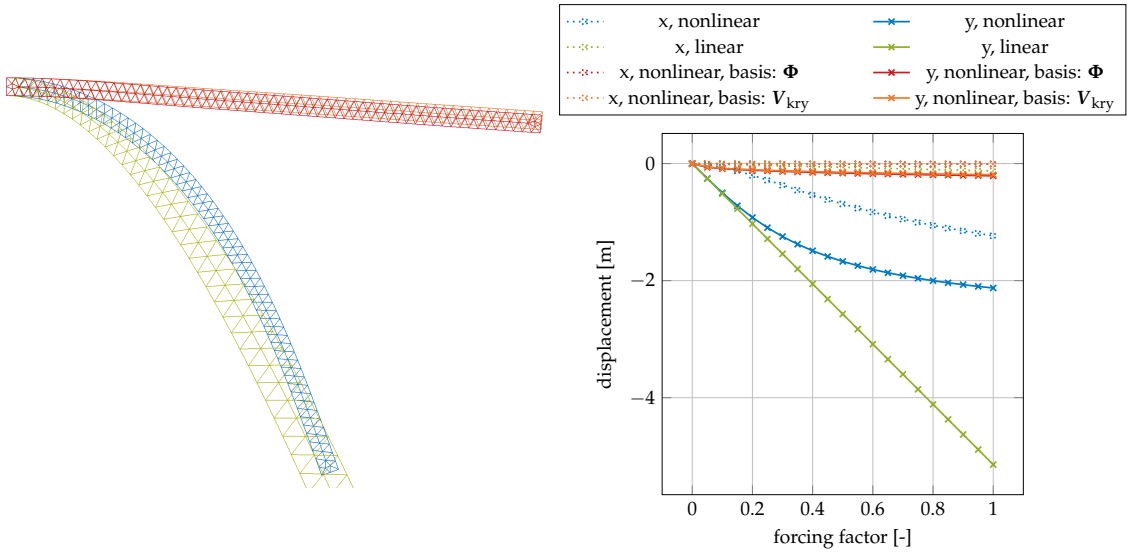


**Figure 6.1:** Static displacement of linear and geometrically nonlinear beam as well as the nonlinear beam reduced with 5 vibration modes and 5 Krylov modes. The displacement of the reduced nonlinear beam is severely underestimated.

Then this basis is used for reducing the nonlinear system. The nonlinear system is loaded statically with the same boundary conditions as in the static experiment in Section 2.5. The results of both reduced nonlinear models are given together with the results of the numerical experiments in Section 2.5 in Figure 6.1. They indicate a very poor behavior, since the displacements are severely underestimated or, from the other point of view, the naive reduced system is orders of magnitude stiffer. This problem of stiffening is well-known in the finite element domain, where it is referred to with the term *locking*. It occurs when the shape functions provided in the element cannot represent the relevant deformation and, in consequence, the element exhibits a stiffer behavior.

In projective reduction, the same effect appears when the subspace spanned by the reduced basis does not capture the deformation sufficiently well. In the example given above, the subspace spanned by the basis $\boldsymbol{V}_{\text{kry}}$ or by the truncated modes $\boldsymbol{\Phi}$ is not sufficient to represent the motion of the geometrically nonlinear beam. Consequently, since the reduced model is a constrained version of the original model as discussed in Chapter 3, the constraints limit the motion of the reduced model resulting in a severe stiffening of the system. Since in the example both the vibration modes and the Krylov modes are transverse modes, i.e., modes with motion perpendicular to the beam axes in the undeformed configuration, the reduced system is forced to only exhibit displacements in the transverse direction. When the beam is forced at the tip, it cannot move along the curved trajectory as depicted in Figure 2.4 but has to move straight downwards, since the motion

is constrained to the reduced basis. This, however, causes unphysical in-plane strains, which cause the stiffening of the structure.

This academic example shows that the basis has to capture the full motion of the nonlinear system. Otherwise, unphysical stiffening effects may occur making the approximation of the reduced model extremely poor. Thereby, the subspace identified with a linear reduction method of the linearized system is generally too small and does not capture directions in the displacement vector space, which are necessary to represent the full nonlinear motion. So even adding more basis vectors from the linear reduction technique does not solve the problem, since they are based on the linearized system and do not consider the nonlinearity triggered.

However, there exist techniques to identify possible directions, which are discussed in the following. They investigate the nonlinearity and identify directions, which are necessary to represent the nonlinear motion.

## 6.2 Key Idea: Augmentation of Reduction Basis

A way to account for the nonlinearities ignored by the linear reduction techniques is the augmentation of the reduced basis $\boldsymbol{V}_{lin}$ stemming from the linearized system with basis vectors $\boldsymbol{\theta}$ 'knowing' the nonlinearity. Hence the linear basis composed of $m$ modes is augmented with $o$ basis vectors $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_o$ to receive the augmented basis $\boldsymbol{V}$ as:

$$\boldsymbol{V} = \left(\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_m, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_o\right). \tag{6.1}$$

In the following, the basis stemming from the linear system is called *lin basis* $\boldsymbol{V}_{lin}$ and the augmentation vectors capturing the nonlinearity are given with the symbol $\boldsymbol{\Theta} = \left(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_o\right)$, independent of the technique applied to generate them. The basis for the reduction of the nonlinear problem is referred to as $\boldsymbol{V}$.

### 6.2.1 Modal Derivatives

When a modal basis is used for the reduction of the geometrically nonlinear system, the modes are based on the linearized system. Thereby the point of linearization is commonly the equilibrium position, for which both the internal and the external forces are zero.

The idea of modal derivatives (MDs) for use in nonlinear reduction is to introduce the point of linearization as a parameter $p$ for the perturbation of the eigenvalue problem (4.11) in Section 4.3. If the nonlinear system (2.41) is linearized about a point in the displacement space, the resulting system can be considered as both linear and parametric, allowing for the modal derivative computation as outlined in Section 4.3. The displacements form an $N$ dimensional parameter space, though. To obtain a single parameter $p$ to compute the perturbation of mode $\boldsymbol{\phi}_i$, $p$ is chosen as the amplitude of a given displacement field, namely the $j$-th vibration mode $\boldsymbol{\phi}_j$. Hence, the stiffness matrix is treated as a parametric stiffness matrix $\boldsymbol{K}(p)$, where the parameter $p = \eta_j$ is the amplitude of the displacement field formed by $\boldsymbol{\phi}_j$, which is the point of linearization:

$$\boldsymbol{K}(p) = \boldsymbol{K}(\boldsymbol{u} = \boldsymbol{\phi}_j \eta_j) = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}=\boldsymbol{\phi}_j\eta_j} \tag{6.2}$$

Since the mass matrix $\boldsymbol{M}$ of the nonlinear equations of motion (2.36) is constant, the parameter dependency of the linearized system is only in the stiffness matrix $\boldsymbol{K}$. The modal derivative $\boldsymbol{\theta}_{ij} = \partial \boldsymbol{\phi}_i / \partial \eta_j$ is then the derivative of mode $\boldsymbol{\phi}_i$ with respect to the amplitude $\eta_j$ of mode $\boldsymbol{\phi}_j$, see (4.13):

$$\left(-\omega_i^2 \boldsymbol{M} + \boldsymbol{K}\right) \frac{\partial \boldsymbol{\phi}_i}{\partial \eta_j} = \left( \boldsymbol{\phi}_i^T \left. \frac{\partial \boldsymbol{K}(\boldsymbol{u} = \boldsymbol{\phi}_j \eta_j)}{\partial \eta_j} \right|_{\eta_j = 0} \boldsymbol{\phi}_i \boldsymbol{M} - \left. \frac{\partial \boldsymbol{K}(\boldsymbol{u} = \boldsymbol{\phi}_j \eta_j)}{\partial \eta_j} \right|_{\eta_j = 0} \right) \boldsymbol{\phi}_i \quad (6.3)$$

As in (4.13), the coefficient matrix $\left(-\omega_i^2 \boldsymbol{M} + \boldsymbol{K}\right)$ of the linear problem above is singular and is solved with the additional constraint (4.16) as given in Section 4.3. Since the mass matrix is constant, the constraint is

$$\frac{\partial \boldsymbol{\phi}_i}{\partial \eta_j} \boldsymbol{M} \boldsymbol{\phi}_i = 0. \quad (6.4)$$

Hence, any MD $\boldsymbol{\theta}_{ij} = \partial \boldsymbol{\phi}_i / \partial \eta_j$ is $\boldsymbol{M}$-orthogonal with respect to the vibration mode $\boldsymbol{\phi}_i$.

The augmentation of the lin basis (6.1) is then performed with the $o = n^2$ modal derivatives $\boldsymbol{\theta}_{ij} = \partial \boldsymbol{\phi}_i / \partial \eta_j$ corresponding to the $n$ vibration modes of the lin basis. Since the raw reduced basis, where both the vibration modes and the modal derivatives are gathered, is not guaranteed to be well conditioned, a deflation as given in Subsection 6.2.3 is necessary to obtain a reduced basis with a low condition number.

### 6.2.2 Static Derivatives

The computation of the modal derivatives introduced in the previous section involve the solution of a singular system, which is not desirable. Hence, in many applications, the mass terms in (6.3) are neglected leading to the linear system of equations

$$\boldsymbol{K} \left. \frac{\partial \boldsymbol{\phi}_i}{\partial \eta_j} \right|^s = - \left. \frac{\partial \boldsymbol{K}(\boldsymbol{u} = \boldsymbol{\phi}_j \eta_j)}{\partial \eta_j} \right|_{\eta_j = 0} \boldsymbol{\phi}_i \quad (6.5)$$

which defines the so-called *static* Modal Derivative (SMD) $\boldsymbol{\theta}_{s,ij} = \partial \boldsymbol{\phi}_i / \partial \eta_j |^s$ with the super or subscript $s$ indicating the static version. The computation of the SMDs is related to smaller computational costs, since the coefficient matrix of (6.5) is the stiffness matrix $\boldsymbol{K}$, which is nonsingular when the system is not floating. Furthermore, the coefficient matrix is constant, so that only one factorization of the matrix $\boldsymbol{K}$ is necessary to compute all SMDs of a number of modes.

These benefits make the SMDs very popular in the literature for the reduction of geometrically nonlinear systems. However, they are seldom distinguished from the classical MDs as defined in (6.3). Consequently, the modal perturbations computed according to (6.3) are referred to as *modal derivatives* (MDs) with the symbol $\boldsymbol{\theta}_{ij}$, whereas the perturbations with neglected mass effects as given in (6.5) are called *static modal derivatives* (SMDs) (cf. [SdS95]) and referred to with the symbol $\boldsymbol{\theta}_{s,ij}$.

The MDs $\boldsymbol{\theta}_{ij} = \partial \boldsymbol{\phi}_i / \partial \eta_j$ as well as the SMDs capture the change of the modes with respect to a changed configuration. Hence, they capture the displacement fields, which are necessary to complement the basis vectors from the linearized system to account for the nonlinearity. This property is illustrated in Figure 6.2, where the SMDs of the first three modes of a cantilevered plate are given.

The first three vibration modes of the plate are transverse modes: $\boldsymbol{\phi}_1$ is the first bending mode, $\boldsymbol{\phi}_2$ is the second bending mode and $\boldsymbol{\phi}_3$ is the first torsion mode. These modes can represent the transverse motion of a linearized system but not the in-plane motion which is triggered by the geometric nonlinearity as illustrated in the motivating example in the section before. The (static) modal derivatives account exactly for these in-plane motions. When the vibration modes are transverse modes, the corresponding (S)MDs turn out to be in-plane modes representing the in-plane stretching which is triggered when the transverse modes are applied to the nonlinear system. Consequently, the (S)MDs are the

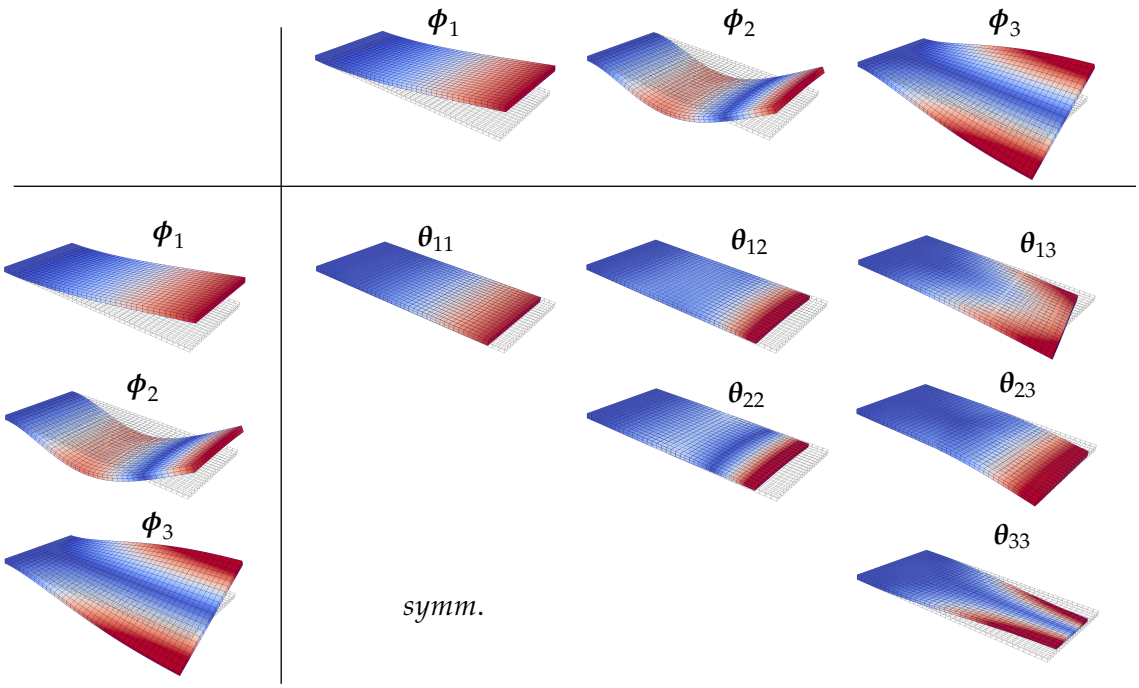


**Figure 6.2:** Static modal derivatives of the first three vibration modes of a cantilevered plate. The figure is inspired by [Tis11].

complementary ingredients to the modal basis which are necessary to allow the curved trajectory in the example given in Section 6.1.

When comparing the constitutive formulas for the computation of the MDs (6.3) with the SMDs (6.5), two main differences are apparent. First, the coefficient matrix for the SMDs, i.e. the stiffness matrix $\boldsymbol{K}$, is independent of the vibration mode $\boldsymbol{\phi}_i$. The coefficient matrix for the MDs, however, contains the information of the eigenfrequency of mode $\boldsymbol{\phi}_i$, since it is $\boldsymbol{K}$ shifted with $-\omega_i^2 \boldsymbol{M}$ leading to the singular coefficient matrix. Second, the right hand side of (6.5) is symmetric with respect to $i$ and $j$. This is due to the fact that the tangential stiffness matrix $\boldsymbol{K}(\boldsymbol{u})$ is the derivative of the internal forces with respect to the displacements. In the derivation, the displacement field $\boldsymbol{u}$ is expressed in terms of the modes $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ and the corresponding amplitudes $\eta_i$ and $\eta_j$ as $\boldsymbol{u} = \boldsymbol{\phi}_i \eta_i + \boldsymbol{\phi}_j \eta_j$. Then the right hand side of (6.5) can be rewritten as

$$\left.\frac{\partial \boldsymbol{K}(\boldsymbol{u} = \boldsymbol{\phi}_j \eta_j)}{\partial \eta_j}\right|_{\eta_j=0} \boldsymbol{\phi}_i = \left.\frac{\partial^2 \boldsymbol{f}(\boldsymbol{u})}{\partial \eta_j \partial \boldsymbol{u}}\right|_{\eta_j=0} \frac{\partial \boldsymbol{u}}{\partial \eta_i} = \frac{\partial^2 \boldsymbol{f}(\boldsymbol{u})}{\partial \eta_i \partial \eta_j} \tag{6.6}$$

since $\partial \boldsymbol{u} / \partial \eta_i = \boldsymbol{\phi}_i$. The right hand side of (6.5) is hence the second derivative of the nonlinear forces with respect to the modal amplitudes $\eta_i$ and $\eta_j$. Consequently, since the partial derivative is a commutative linear operation, $\partial^2 \boldsymbol{f} / \partial \eta_i \partial \eta_j = \partial^2 \boldsymbol{f} / \partial \eta_j \partial \eta_i$ holds. Thus, the SMDs are symmetric with respect to the indices $i$ and $j$:

$$\boldsymbol{\theta}_{s,ij} = \left.\frac{\partial \boldsymbol{\phi}_i}{\partial \eta_j}\right|^s = \left.\frac{\partial \boldsymbol{\phi}_j}{\partial \eta_i}\right|^s = \boldsymbol{\theta}_{s,ji}. \tag{6.7}$$

With the derivation above, the SMDs can be interpreted as the negative linear static response to the second derivative of the nonlinear forces:

$$\boldsymbol{\theta}_{s,ij} = -\boldsymbol{K}^{-1} \frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{\eta}_i \partial \boldsymbol{\eta}_j}. \tag{6.8}$$

With this in mind, it becomes clear that the SMDs 'know' the nonlinearity of the system. They capture the quadratic nonlinearities triggered and extend the basis with components, the linear reduction techniques are not aware of.

With the interpretation of the SMDs as the displacement fields triggered by the second order derivatives of the nonlinear forces, the concept of SMDs is not limited to vibration modes. This concept can be applied to all types of displacement fields used for linear reduction. These modes are referred to as *static derivatives* (SDs), since they do not involve any dynamical effects but are based on the static higher order analysis of the nonlinear internal forces. This generalization was first proposed in Idelsohn and Cardona [IC85a] for Krylov subspace vectors, though lacking a sound theory. A deeper physical interpretation can be achieved with the concept of Quadratic Manifold, which is introduced in the following chapter.

Similar to the MDs, the SMDs or SDs form the augmentation vectors in (6.1). Since the SMDs and SDs are symmetric with $\boldsymbol{\theta}_{s,ij} = \boldsymbol{\theta}_{s,ji}$, $o = n(n+1)/2$ distinct S(M)Ds for $n$ vibration modes exist. Like the MDs, the SMDs do not form a linearly independent basis when combined with the vibration modes. Hence, the raw basis of the concatenated modes and (S)MDs has to be processed further to build a basis which is numerically stable.

### 6.2.3 Deflation and Orthogonalization

When the basis $\boldsymbol{V} = (\boldsymbol{V}_{lin}, \boldsymbol{\Theta})$ is built according to (6.1), the condition number of $\boldsymbol{V}$ becomes in general very high, when the size of $\boldsymbol{V}_{lin}$ and $\boldsymbol{\Theta}$ increases. This can cause numerical errors, since the reduced system is projected onto this basis resulting in poorly conditioned reduced mass, damping and tangential stiffness matrices. Since the matrices $\boldsymbol{M}_r = \boldsymbol{V}^T\boldsymbol{M}\boldsymbol{V}$, $\boldsymbol{C}_r = \boldsymbol{V}^T\boldsymbol{C}\boldsymbol{V}$ and $\boldsymbol{K}_r(\boldsymbol{q}) = \boldsymbol{V}^T\boldsymbol{K}(\boldsymbol{V}\boldsymbol{q})\boldsymbol{V}$ are multiplied by $\boldsymbol{V}$ twice, the condition number of $\boldsymbol{V}$ gets generally squared leading to very ill-conditioned systems in the time integration causing poor convergence and even instability.

The potentially high condition number of the raw basis constructed in (6.1) stems from the fact, that the (S)MDs or the SDs are based on perturbations of vibration modes or the nonlinear forces and are not necessarily linear independent with respect to the other basis vectors. In the reduced basis technique, though, only the subspace spanned by the basis $\boldsymbol{V}$ is of interest, while the parametrization of the basis is arbitrary. Consequently, a Gram-Schmidt like orthogonalization technique can be used to obtain a low condition number for the basis. However, for the sake of efficiency, it is better to deflate the raw basis, i.e., to remove the linear dependent vectors from the basis and capture the subspace spanned by the raw basis. Then, the reduced dimension of the model is equivalent to the rank of $\boldsymbol{V}$.

To obtain the subspace spanned by $\boldsymbol{V}_{lin}$ and $\boldsymbol{\Theta}$, an SVD can reveal the continuous rank decay associated with the (S)MDs. Therefore, the basis vectors are normed and then gathered in the raw matrix $\boldsymbol{R}_{\text{raw}}$:

$$\boldsymbol{R}_{\text{raw}} = \left( \frac{\boldsymbol{v}_1}{|\boldsymbol{v}_1|} \quad \cdots \quad \frac{\boldsymbol{v}_m}{|\boldsymbol{v}_m|} \quad \frac{\boldsymbol{\theta}_{11}}{|\boldsymbol{\theta}_{11}|} \quad \cdots \quad \frac{\boldsymbol{\theta}_{1m}}{|\boldsymbol{\theta}_{1m}|} \quad \frac{\boldsymbol{\theta}_{21}}{|\boldsymbol{\theta}_{21}|} \quad \cdots \quad \frac{\boldsymbol{\theta}_{mm}}{|\boldsymbol{\theta}_{mm}|} \right). \tag{6.9}$$

Note, that for the non-symmetric case of the MDs, all MDs are gathered in $\boldsymbol{R}_{\text{raw}} \in \mathbb{R}^{m(m+1)}$. For the symmetric S(M)Ds, only the unique basis vectors are collected in $\boldsymbol{R}_{\text{raw}} \in \mathbb{R}^{\frac{m(m+3)}{2}}$. The SVD of $\boldsymbol{R}_{\text{raw}}$ results in

$$\boldsymbol{R}_{\text{raw}} = \boldsymbol{U}_{\text{svd}}\boldsymbol{\Sigma}_{\text{svd}}\boldsymbol{V}_{\text{svd}} \tag{6.10}$$

with the orthogonal matrices $\boldsymbol{U}_{\text{svd}}$ and $\boldsymbol{V}_{\text{svd}}$ and the diagonal matrix $\boldsymbol{\Sigma}_{\text{svd}}$ composed of the singular values arranged in decreasing order $\sigma_1 \geq \sigma_2 \geq \dots$. These singular values

indicate the continuous rank decay of the raw basis $\boldsymbol{R}_{\text{raw}}$. An orthogonal basis spanning the relevant subspace of $\boldsymbol{R}_{\text{raw}}$ can be built with the $n$ left singular vectors, which are associated with the $n$ largest singular values. One possible way to identify the numerical rank of $\boldsymbol{R}_{\text{raw}}$ is the introduction of the tolerance $\varepsilon$, so that only singular vectors associated with singular values smaller than $\varepsilon\,\sigma_1$ are dismissed:

$$\boldsymbol{V} = \left(\boldsymbol{u}_{\text{svd},1}, \ldots, \boldsymbol{u}_{\text{svd},n}\right), \quad \text{with} \quad \varepsilon\,\sigma_1 > \sigma_{n+1}. \tag{6.11}$$

The tolerance $\varepsilon$ gives the accuracy, up to which threshold the subspace spanned by $\boldsymbol{R}_{\text{raw}}$ is kept. In the applications in this work, $\varepsilon$ is chosen to be $10^{-8}$ and thus small enough to capture the full rank of $\boldsymbol{R}_{\text{raw}}$ in a numerically stable way.

In the literature, refinements for the identification of the subspace are presented in Barbič and James [BJ05], where the procedure above is altered in two places. First, a different norming of the raw matrix $\boldsymbol{R}_{\text{raw}}$ in (6.9) is possible. The norming might be based on the eigenfrequencies of the modes, the un-normed length of the (S)MDs or other properties. The second place is the computation of the SVD in (6.10), where a weighted SVD as explained in Subsection 5.2.2 can be used. Especially the mass-weighted SVD is a proper choice, since it alleviates the issues of different mesh densities and different physical coordinates.

### 6.2.4 Selection Criteria for Modal Derivatives

The augmentation approach presented in the previous subsections has one fundamental drawback: the quadratic growth of the basis with respect to the size of the lin basis. This flaw is due to the fact, that the (S)MDs are quantities based on the interaction of modes and hence the number of interaction possibilities rises quadratically. Though deflation strategies as proposed in the previous subsection are able to extract the relevant subspace of all modes and the corresponding (S)MDs and thus may reduce the size of the quadratically growing basis, this course of dimensionality is prohibitive for models exhibiting complex motions, for which many linear modes are necessary.

To alleviate this issue, selection techniques have been developed in order to retain only a few (S)MDs necessary to represent the motion properly. These techniques rely on heuristic rankings of the (S)MDs, where either a cheap linear test run is performed with the given external excitation or where system-specific properties of the linearized system are exploited like the eigenfrequencies of the system. All these methods can be generalized to a method, where a weighting matrix $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ is built. This matrix, where the row and column indices stand for the indices of the (S)MD, is filled by the methods with importance factors so that the entry $W_{ij}$ represents the importance of MD $\theta_{ij}$ or SMD $\theta_{s,ij}$. The value of these factors give then the ranking of the (S)MDs to be selected for the basis $\boldsymbol{V}$.

The first proposed method is inspired by Barbič and James [BJ05], where the reduced basis is used for real-time computer graphics. It is named here frequency weighting (FW) as it selects the SMDs based on the eigenfrequency of the two modes involved in the SMDs. Since in the weighting matrix $\boldsymbol{W}$ higher values are ranked higher, the inverse of the product of both eigenfrequencies is used as weighting factor:

$$W_{fw,ij} = \frac{1}{\omega_i \omega_j}. \tag{6.12}$$

This method selects the SMDs independently of the excitation, since the eigenfrequencies are properties obtained from the homogeneous linearized system. One method considering the excitation is the Maximum Modal Interaction (MMI) scheme proposed by Tiso [Tis11]. It relies on a training simulation of the linearized system reduced with

modal truncation. The amplitudes $\eta_i$ of the modal coordinates are evaluated to identify the modes which are triggered simultaneously in the linearized system:

$$W_{mmi,ij} = \int_0^T |\eta_i(t)\eta_j(t)| \mathrm{d}t. \tag{6.13}$$

When the training simulation is run with a time integration scheme with evenly spaced time steps, the weighting matrix is obtained as

$$W_{mmi,ij} = \sum_{t \in \mathcal{T}} |\eta_i(t)\eta_j(t)| \tag{6.14}$$

with the set $\mathcal{T}$ of all time steps. The correction with $\Delta t$ to fulfill the approximation of the integral in (6.13) is not necessary, since the weights are ranked and constant factors do not change this ranking.

Another technique named Modal Virtual Work (MVW) proposed by Jain et al. [Jai+17] assesses the virtual work of mode $i$ done on mode $j$ based on the linear training set as in the method before:

$$t_{max,i} = \arg\max |\eta_i(t)| \tag{6.15}$$

$$W_{mvw,ij} = |\boldsymbol{\phi}_j^T \boldsymbol{f}(\boldsymbol{\phi}_i \eta_i(t_{max}))|. \tag{6.16}$$

Since the MVW method is not symmetric, it is specially suited for the MDs, which are also not symmetric. However, if the symmetric SMDs are used, the symmetric Modal Virtual Work (SMVW) is proposed with the weighting coefficients given as:

$$t_{max,ij} = \arg\max(\eta_i(t)\eta_j(t))^2 \tag{6.17}$$

$$W_{smvw,ij} = \sqrt{\left(\boldsymbol{\phi}_j^T \boldsymbol{f}\left(\boldsymbol{\phi}_i \eta_i\left(t_{max,ij}\right)\right)\right)^2 + \left(\boldsymbol{\phi}_i^T \boldsymbol{f}\left(\boldsymbol{\phi}_j \eta_j\left(t_{max,ij}\right)\right)\right)^2}. \tag{6.18}$$

This method captures the approximated virtual work of a mode pair $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ in a symmetric fashion.

All selection methods are purely heuristic methods which are based on the linearized system. They rank the (S)MDs by importance according to the heuristic scheme and do not suggest a selection of vibration modes. As they do not consider the nonlinearity, their performance is very problem-dependent. Hence, a non-exhaustive investigation of the performance of these methods is given in the application part of this chapter in Section 6.4.

## 6.3 Numerical Differentiation

The computation of the (S)MDs and the SDs involves the derivative of the tangent stiffness matrix with respect to a displacement direction. This derivative has to be carried out either intrusively within the finite element framework or non-intrusively via finite differences or an identification technique.

In the intrusive computation, the derivative with respect to the displacement field is computed analytically on the level of the finite element, as for instance given in the appendix of Barbič and James [BJ05]. This requires special implementation for all element types and is often not available in general finite element codes. However, it enjoys the benefit of accuracy up to machine precision. The non-intrusive methods, on the other hand, do not require the access to special implementations within the finite element routines but allow the determination of the derivatives with multiple evaluations of tangential stiffness matrices or nonlinear forces. However, when these methods are used, special care has to be taken in order to obtain accurate results.

The most prominent method to obtain the derivatives is the finite difference method, where the derivative is computed as a discrete difference of two evaluations. The difference can either be computed with forward, backward or central differences. For the evaluation of the right hand side of the SMDs, the derivative of the tangential stiffness matrix $\boldsymbol{K}(\boldsymbol{u})$ with respect to the amplitude $\eta_j$ of mode $\boldsymbol{\phi}_j$ is approximated with forward differences as

$$\left.\frac{\partial \boldsymbol{K}(\boldsymbol{u}=\boldsymbol{\phi}_j\eta_j)}{\partial \eta_j}\right|_{\eta_j=0}^{fd} = \frac{\boldsymbol{K}(\boldsymbol{u}=\boldsymbol{\phi}_j\cdot h)-\boldsymbol{K}(\boldsymbol{u}=\boldsymbol{0})}{h}, \tag{6.19}$$

with backward differences as

$$\left.\frac{\partial \boldsymbol{K}(\boldsymbol{u}=\boldsymbol{\phi}_j\eta_j)}{\partial \eta_j}\right|_{\eta_j=0}^{bd} = \frac{\boldsymbol{K}(\boldsymbol{u}=\boldsymbol{0})-\boldsymbol{K}(\boldsymbol{u}=-\boldsymbol{\phi}_j\cdot h)}{h} \tag{6.20}$$

and with central differences as

$$\left.\frac{\partial \boldsymbol{K}(\boldsymbol{u}=\boldsymbol{\phi}_j\eta_j)}{\partial \eta_j}\right|_{\eta_j=0}^{cd} = \frac{\boldsymbol{K}(\boldsymbol{u}=\boldsymbol{\phi}_j\cdot h)-\boldsymbol{K}(\boldsymbol{u}=-\boldsymbol{\phi}_j\cdot h)}{2h}. \tag{6.21}$$

The forward and backward differences share the benefit, that the linear stiffness matrix $\boldsymbol{K}=\boldsymbol{K}(\boldsymbol{u}=\boldsymbol{0})$ can be reused and hence only one evaluation of the tangential stiffness matrix is necessary for the computation of one directional derivative.

The finite difference scheme is an approximation, which should come as close to the original derivative as possible. To achieve good accuracy, both the choice of the scheme and the step width $h$ have to be chosen correctly. As measure for the accuracy of the finite difference scheme, the symmetry of the SMDs can be used. Since for the computation of $\boldsymbol{\theta}_{ij}$ the derivative of $\boldsymbol{K}_r(\boldsymbol{q})$ with respect to $\eta_j$ is evaluated and for $\boldsymbol{\theta}_{ji}$ the derivative is taken with respect to $\eta_i$, numerical errors in the finite difference scheme result in deviations between $\boldsymbol{\theta}_{ij}$ and $\boldsymbol{\theta}_{ji}$. Hence, the error measure $\epsilon_{\text{symm}}$ is a good indicator for the quality of the numerical differentiation scheme:

$$\epsilon_{\text{symm}} = \frac{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{\theta}_{ij}-\boldsymbol{\theta}_{ji})^T(\boldsymbol{\theta}_{ij}-\boldsymbol{\theta}_{ji})}}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\boldsymbol{\theta}_{ji}^T\boldsymbol{\theta}_{ij}}} \tag{6.22}$$

In Figure 6.3, the relative symmetry error $\epsilon_{\text{symm}}$ for the cantilevered beam example from Section 6.1 is depicted. The forward and backward finite difference schemes show almost the same symmetry error for the given step width. They perform poorly, since their minimal symmetry error is in the range of $\epsilon_{\text{symm}} = 10^{-5}$, while the central finite difference scheme achieves a minimal error in the range below $\epsilon_{\text{symm}} = 10^{-11}$. The lowest symmetry error and hence the highest accuracy of the one-sided schemes is in the range of the step width $h = 10^{-6}$. This is in accordance with the common step width for floating point computations as recommended for instance in Gill et al. [GMW81], which is given in the range $\sqrt{\varepsilon}$, where $\varepsilon$ is the machine precision which is approximately $2.2 \cdot 10^{-16}$ for a 64 bit double precision floating point number. The central difference scheme, however, has its minimum in the range of $h = 10^0 - 10^1$. In the author's experience, the optimal step width with the lowest symmetry error is always in this range. However, a detailed investigation revealing a reason or a pattern for the optimal step width is a topic for future research.

The first reason for the poor performance of the one-sided finite difference schemes lies in the rotational rigid body modes of the elements. They can spoil the accuracy of

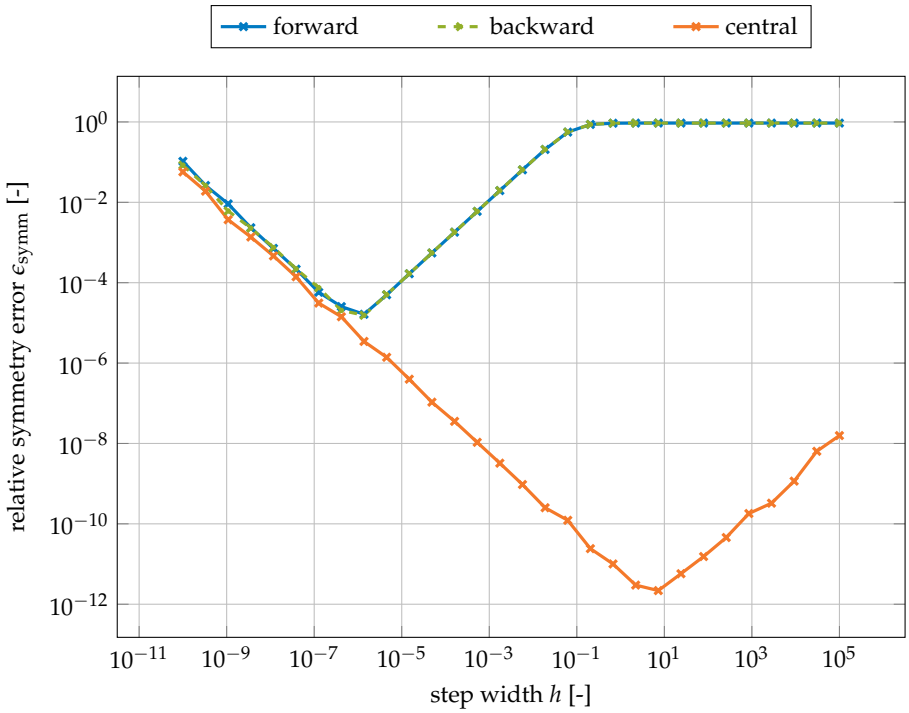


**Figure 6.3:** Relative symmetry error of the SMDs computed with different schemes and different step widths $h$ for the cantilever example. Only the central finite difference scheme yields acceptable results.

the finite difference scheme, since the linearized rigid body rotations cause a change of the volume. As illustrated in Figure 2.6 in Section 2.5, the linearized rotation distorts the element in an unphysical manner causing large elastic forces to compensate for the volume growth. This results in bad approximations in the numerical differentiation scheme. One way to overcome this limitations is the semi-analytical computation of derivatives as proposed by van Keulen and de Boer [VD98; DV00], where the contribution of the rigid body modes is computed analytically while the derivative of the deformation is obtained via a one-sided finite difference scheme. These methods require a special implementation, however they accelerate the finite difference computation compared to central finite differences. They, on the other hand, require more evaluations and thus higher computational efforts, but they allow for high accuracy close to machine precision, when the step width $h$ is well tuned. In this work, solely central finite difference schemes with tuned step widths are used.

The second reason for the sensitivity of the SMDs with respect to the step width are the right hand side pseudo forces in (6.5), which are depicted exemplarily for the SMD $\boldsymbol{\theta}_{11}$ of the first mode $\boldsymbol{\phi}_1$ in the upper part of Figure 6.4. The SMD $\boldsymbol{\theta}_{11}$ is an in-plane mode realizing a contraction of the beam similar to $\boldsymbol{\theta}_{11}$ in Figure 6.2. This contraction is acting in both the vertical an the horizontal direction and compensates the volume growth due to the linearization of the rotation.

From the computational perspective, this compression is caused by the right hand side pseudo forces of (6.5), which are depicted in the lower part of Figure 6.4. These forces acting on the surface nodes with increasing amplitude towards the right hand side are responsible for the compression displacement field $\boldsymbol{\theta}_{11}$. If the forces on the top and bottom face are not perfectly balanced due to numerical errors, they cause large displacements of the beam in the vertical direction, even though the error of the forces is in the range several magnitudes below.

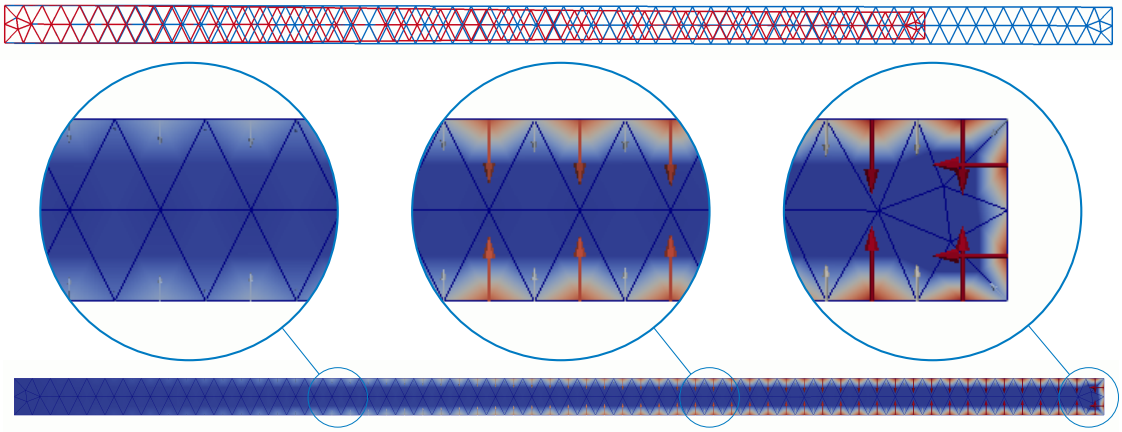

**Figure 6.4:** SMD $\boldsymbol{\theta}_{s,11}$ (top) and corresponding right hand side pseudo forces (bottom). The pseudo forces cause an in-plane compression of the beam as well as a transversal compression.

In general, the solution of the static problem (6.5) can amplify errors immensely, leading to deteriorated SMDs. Hence, a proper tuning of the step width $h$ is necessary before the computation of the SMDs. Thereby the symmetry of the SMDs can be used as cross validation indicator for the quality of the approximation. In the numerical experiments, the optimal step width $h$ is identified by the evaluation of the relative symmetry error $\epsilon_{\text{symm}}$ for the selected modes forming the linear basis. When the computation of $\epsilon_{\text{symm}}$ for various step width is tedious due to the number of modes and the size of the finite element model, only a small number of modes spanning the frequency range of the modes in the linear basis are used for the evaluation of $\epsilon_{\text{symm}}$. In the numerical experiments, commonly five modes were chosen. Once the optimal step width is determined for a model, the MDs are computed with this tuned step width, too.

## 6.4 Applications

The capability of the simulation free augmentation vectors introduced above is demonstrated on two examples. The first is the cantilever from the motivating example in Section 6.1, which is loaded dynamically on the tip with a load given in Table B.1 together with the other physical properties of the system and the time integration used.

First, the first ten vibration modes and the corresponding MDs and SMDs are computed. To assess the consistency of the MDs and the SMDs, both are stacked in a matrix $\boldsymbol{\Theta} \in \mathbb{R}^{N\times 100}$ and $\boldsymbol{\Theta}_s \in \mathbb{R}^{N\times 100}$ in an ordered manner, so that the first ten (S)MDs are all derivatives of the first mode $\boldsymbol{\phi}_1$ and so on:

$$\begin{aligned} \boldsymbol{\Theta} &= \left(\boldsymbol{\theta}_{11}, \boldsymbol{\theta}_{12}, \ldots, \boldsymbol{\theta}_{1n}, \boldsymbol{\theta}_{21}, \ldots, \boldsymbol{\theta}_{nn}\right) \quad \forall n \in 1, \ldots, 10 \\ \boldsymbol{\Theta}_s &= \left(\boldsymbol{\theta}_{s,11}, \boldsymbol{\theta}_{s,12}, \ldots, \boldsymbol{\theta}_{s,1n}, \boldsymbol{\theta}_{s,21}, \ldots, \boldsymbol{\theta}_{s,nn}\right) \quad \forall n \in 1, \ldots, 10 \end{aligned} \tag{6.23}$$

Then, the similarity of the MDs and the SMDs is checked. Therefore, the modal assurance criterion (MAC) is used, see e.g. Allemang [All03], which is a quadratic correlation measure of two displacement fields $\boldsymbol{\phi}_i$ and $\boldsymbol{\psi}_j$. It is defined as:

$$MAC_{ij} = \frac{(\boldsymbol{\phi}_i^T \boldsymbol{\psi}_j)^2}{\boldsymbol{\phi}_i^T \boldsymbol{\phi}_i \; \boldsymbol{\psi}_j^T \boldsymbol{\psi}_j}. \tag{6.24}$$

When applying the MAC to measure the similarity of MDs and SMDs, $\boldsymbol{\phi}_i$ is the $i$-th column of the matrix $\boldsymbol{\Theta}$ and $\boldsymbol{\psi}_j$ is $j$-th column of the matrix $\boldsymbol{\Theta}_s$ defined in (6.23). Then the

MAC as defined above represents the correlation of every MD with every SMD. If both the MD with index $i$ and the SMD with index $j$ coincide, the values of the MAC is one representing a perfect fit. If the displacement fields are barely correlated, the MAC value is close to zero.

Since the MAC values are correlation coefficients between two variables, they can be displayed in a table or matrix. If both variables correlate, the diagonal values are all one, since equal vectors correlate perfectly. The off-diagonal entries indicate though, how much vectors with different indices correlate with each other.

In Figure 6.5 the MAC tables are graphically depicted for three cases. On the top left, the so-called auto-MAC of the MDs with respect to the MDs are given. The diagonal is consistently one, while the off-diagonal entries are small indicating, that every MD is different from the other. Only few index combinations show high MAC values indicating a closeness of MDs with different indices. The depiction of the auto-MAC of the SMDs given in the top right is different, though. Despite the consistent diagonal, many more off-diagonal entries exist with high values. One reason for this phenomenon is the symmetry of the SMDs, which is discussed in Subsection 6.2.2. However, despite that fact, more SMDs with different indices are correlated than MDs. Especially the SMDs from 40-49 and from 80-89 have a strong correlation among each other. This is due to the fact, that both modes $\boldsymbol{\phi}_4$ and $\boldsymbol{\phi}_8$ forming the parents of these SMDs are in-plane modes, so that the corresponding SMDs are transverse displacements sharing similarity to the other transverse modes.

The MAC criterion of the MDs with respect to the SMDs are given in the bottom part of Figure 6.5. There, the diagonal entries are close to one only for the lower modes and are small for the higher modes. This indicates, that the SMDs are an approximation of the MDs only for lower frequency modes, whereas they diverge for higher frequency modes. Furthermore, the SMDs corresponding to the in-plane-modes $\boldsymbol{\phi}_4$ and $\boldsymbol{\phi}_8$ are fully uncorrelated. Consequently, the MDs and SMDs seem to capture different subspaces, which will be discussed in the following, when (S)MDs are used as reduced bases.

For assessing the reduction capability of the (S)MDs, the linear basis is built from vibration modes and Krylov subspace vectors. Then the corresponding MDs, SMDs and SDs are computed and a deflated linear basis is constructed according to (6.9) and (6.10). A time integration scheme is run and the relative error $RE$ is measured according to (3.9). The results of the reduction error are given in Figure 6.6. Thereby the left plot depicts the relative error over the size $m$ of the linearized system, i.e., the number of modes used for computing the (S)MDs and SDs. The right plot displays the same information over the dimension $n$ of the reduced system after deflation. For comparison, the RE of a POD basis is also given. It is, however, trained with the full solution of the model, which is not a realistic scenario, since the training simulation usually deviates from the solution sought. Nonetheless, the POD basis serves as a benchmark of the best basis possible, since it is the optimal basis tailored to exactly this model, load case and integration time.

The left plot of Figure 6.6 indicates that the MDs are indeed not symmetric, since they cover a larger subspace which performs better for the same number $m$ of modes from the linearized system. For this example, the MDs build the best reduced basis for mostly all dimensions $n$ compared to the simplified SMDs. Only for small reduction dimensions $n$, the SMDs perform slightly better. The Krylov SDs, though, do not exhibit the accuracy of the MDs or SMDs.

The performance of the selection criteria given in Subsection 6.2.4 is further investigated. Therefore, the weighting matrices $\boldsymbol{W}_{fw}$ for frequency weighting, $\boldsymbol{W}_{mmi}$ for maximum modal interaction, $\boldsymbol{W}_{mvw}$ for maximum virtual work and $\boldsymbol{W}_{smvw}$ for symmetric virtual work are computed. They are depicted in Figure 6.9. Commonly, all weighting schemes indicate a tendency to weight lower frequencies higher than higher frequencies.

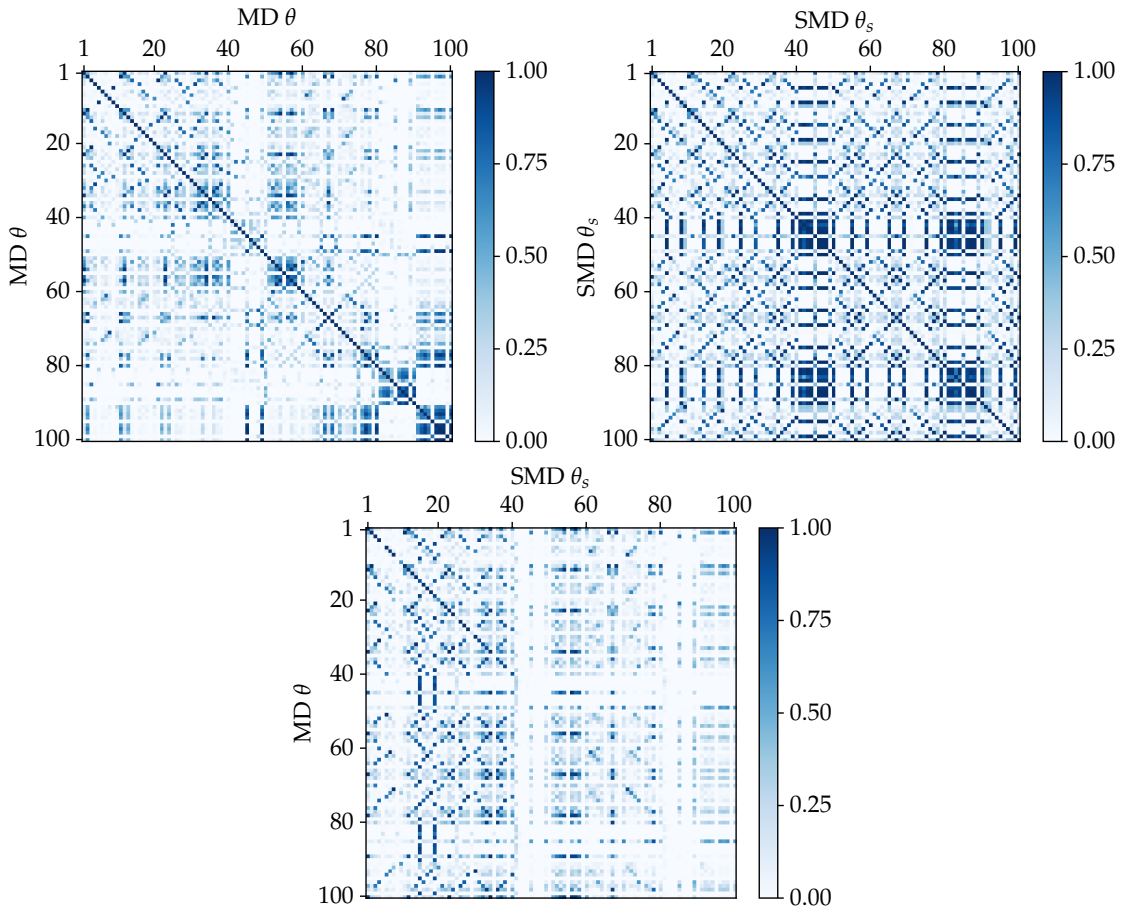


**Figure 6.5:** Auto-MAC of the MDs (top left), auto-MAC of the SMDs (top right) and MAC between MDs and SMDs (bottom) for the cantilever example.

However, the in-plane modes 5, 9, 13, 16 and 19 exhibit clearly lower weights in the MMI and MVW ranking, since they are barely triggered by the linear excitation or produce no work on the transverse modes. Furthermore, both virtual work rankings clearly emphasize the diagonal terms and weight (S)MDs which are derivatives with respect to itself higher.

For investigating the performance of the ranking schemes, for $m$ linear modes the $m + m(m+1)/6$ highest ranked (S)MDs are added to the raw basis $\boldsymbol{R}_{\text{raw}}$ according to (6.9). The raw basis is deflated using (6.10) and (6.11) and used as reduced basis for the reduction of the cantilever example.

The investigation of the displacements over time are given exemplarily for the size $m = 6$ of the linearized system in Figure 6.8. In this plot, no difference between the reference model and the reduced models is visible, though the system behaves strongly nonlinearly, as the displacement snapshots in Figure 6.7 indicate. The reduced bases obtained with selection techniques above have the dimension $n$ of 18 and 19. The SMD and Krylov SD basis has 27 dofs and the MD basis has 41 dofs.

A more detailed assessment is given in Figure 6.10. The relative error of the reduced models with respect to the number $m$ of basis vectors from the linearized system is given in the left plot. In the right plot the same error is given with respect to the dimension $n$ of the reduced model. Both plots indicate, that the MMI selection technique does not perform well compared to the other methods. Overall, the MDs tend to achieve better results than the SMDs, especially for higher reduced dimensions. Furthermore, the symmetric MVW applied to SMDs performs very poorly, since the error rises with a larger

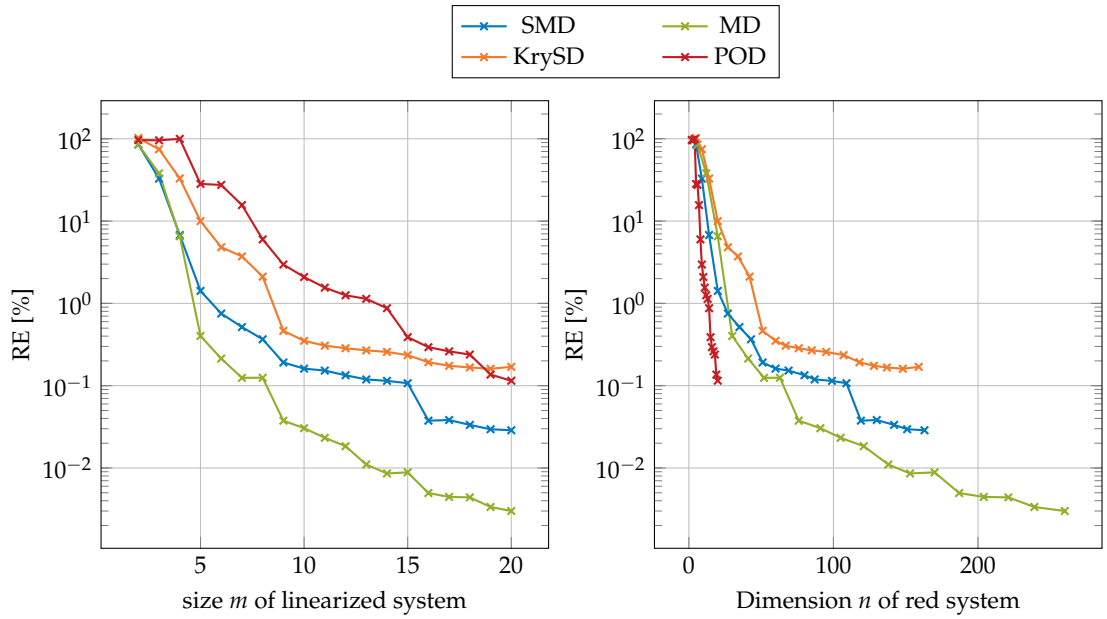


**Figure 6.6:** Error of different reduction methods with deflation for the cantilever example. The relative error is given over the dimension of the linear basis (left), and over the dimension of the reduced model (right). The POD reduction was trained on the full solution.

dimension of the linearized system. This is due to the fact, that SMDs associated with higher frequency modes are preferred to lower frequency mode interactions resulting in a poor reduction basis.

Interestingly, the frequency weighting technique, which is the computationally cheapest, produces the best results for both, SMDs and SDs. It seems to favor the (S)MDs in the most efficient fashion.

The cantilever is a special structure, which does not represent many phenomena. It comprises a decoupling of in-plane and transverse motion, which is represented by the modes which can be distinguished in these two categories. Hence the second example c-shape is investigated, since it has both a slender structure but no separation in in-plane and transverse modes.

First, the MDs and SMDs are computed. The same MAC investigation as with the cantilever is given in Figure 6.11. Thereby, the trends in the difference between MDs and SMDs are even stronger compared to the cantilever example. While most of the MDs show a distinct decoupling against each other, the SMDs exhibit an extreme correlation with many off-diagonal terms being close to one. Consequently, the SMDs are very often similar displacement fields while the MDs are mostly distinct besides in the top left corner, where the MDs related to $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ show clear closeness.

The MAC between MDs and SMDs is far from exhibiting a dominant diagonal, indicating that MDs and SMDs are different displacement fields. Given the lack of correlation between MDs and SMDs, the SMDs are barely an approximation of the MDs for the c-shape example. Hence, the approximation of MDs with SMDs might hold for simple structures exhibiting decoupling effects as in straight or slightly curved beams (cf. the examples in [IC85a]), but does not hold for complex models or modes in the higher frequency range.

Next, the reduction capability is assessed for the c-shape example. Therefore, as in the beam example before, the basis constructed of modes and (S)MDs is deflated and the relative error is measured of the reduced models for different reduction orders. In Figure 6.12 the RE is given over the size $m$ of the linearized system and the order $n$ of the

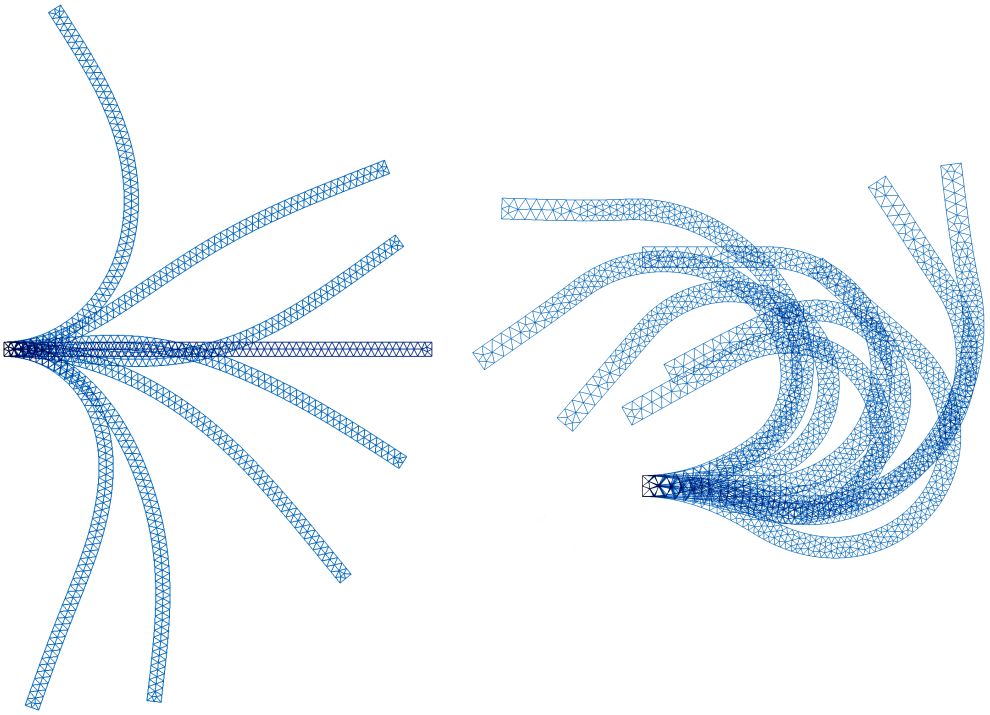

**Figure 6.7:** Displacement snapshots of the two benchmark examples cantilever and c-shape at random time instances. The motion of both is clearly geometrically nonlinear.

reduced system. The graph basically shows the same results as the cantilever example: The Krylov SDs perform the worst while the MDs perform best. Also the MDs span the larger subspace than the other derivatives, since they lack the symmetry. The errors are also in the similar order of magnitude indicating a good approximation of the nonlinear motion. As in the cantilever example, the tip displacements of reduced models with $m = 6$ depicted in Figure 6.15 overlap with the full solution.

Also the selection strategies show a similar trend in Figure 6.14 compared to the cantilever example. Once again, the MDs outperform the SDs for larger reduced dimensions while the methods exhibit similar performance. Also the trend of the SMVW selection to perform worse for MDs with a larger number of vibration modes is clearly visible.

To summarize, both the MDs as well as the SMDs and SDs based on Krylov vectors are generally suitable for augmenting the so-called *lin basis* stemming from linearized systems like vibration modes or Krylov subspace modes. However, since the (S)MDs grow quadratically with respect to the number of vibration modes or Krylov subspace modes, the large size of the basis to achieve good results might be limiting. There exist good techniques to rank the derivative vectors and augment the basis only with few, however, they barely are as efficient as well-trained POD basis vectors. Nonetheless, the (S)MDs are an attractive choice to reduce geometrically nonlinear models in a simulation-free fashion. The issue of the quadratic growth of the basis is tackled in the following chapter by a projection technique different from the linear projection technique.

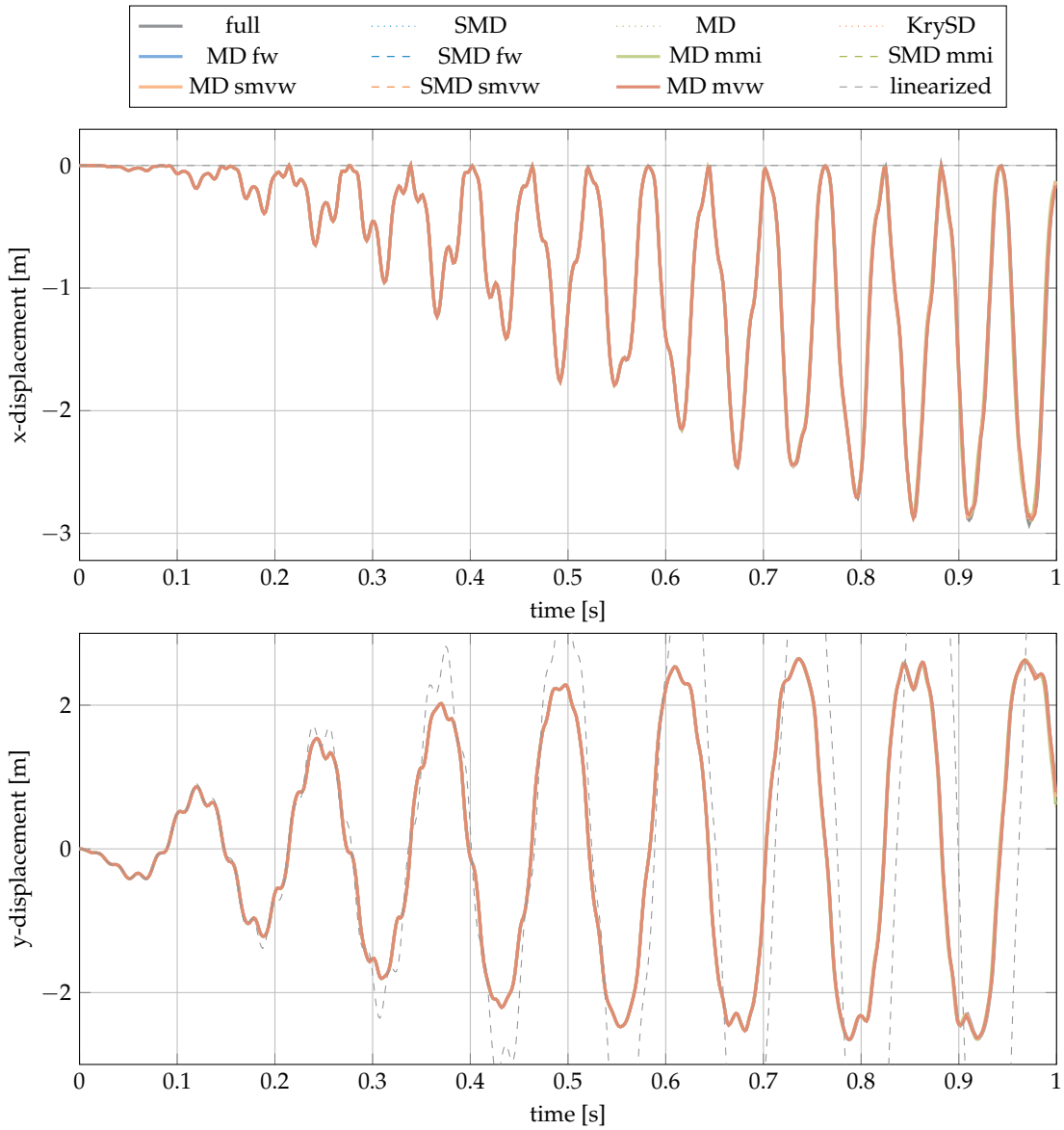


**Figure 6.8:** Displacement of the tip of the cantilever example for different simulation free reduction methods for $m = 6$. The dimension $n$ of the reduced models are between 18 and 41. The abbreviations for the selection strategies in the legend stand for: fw – frequency weighting, mmi - maximum modal interaction, smvw - symmetric modal virtual work, mvw – modal virtual work.

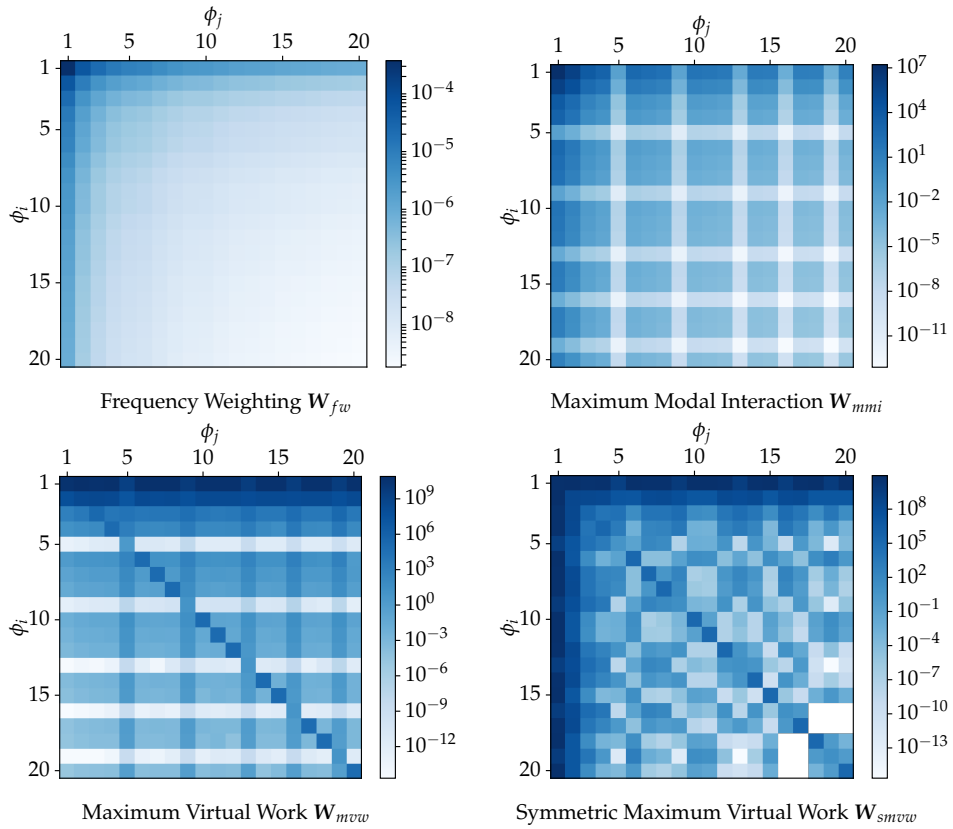


**Figure 6.9:** Weighting matrices for different weighting schemes applied to the cantilever example.

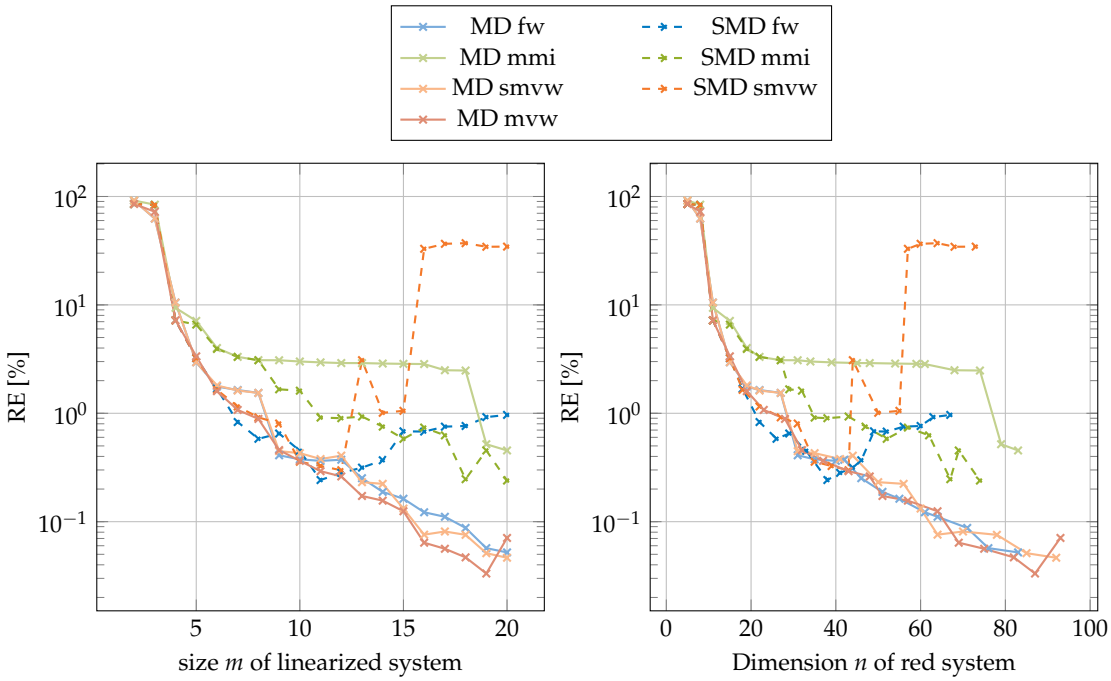


**Figure 6.10:** Comparison of different selection methods with deflation for the cantilever example. The relative error is given over the dimension of the linear basis (left), and over the dimension of the reduced model (right).

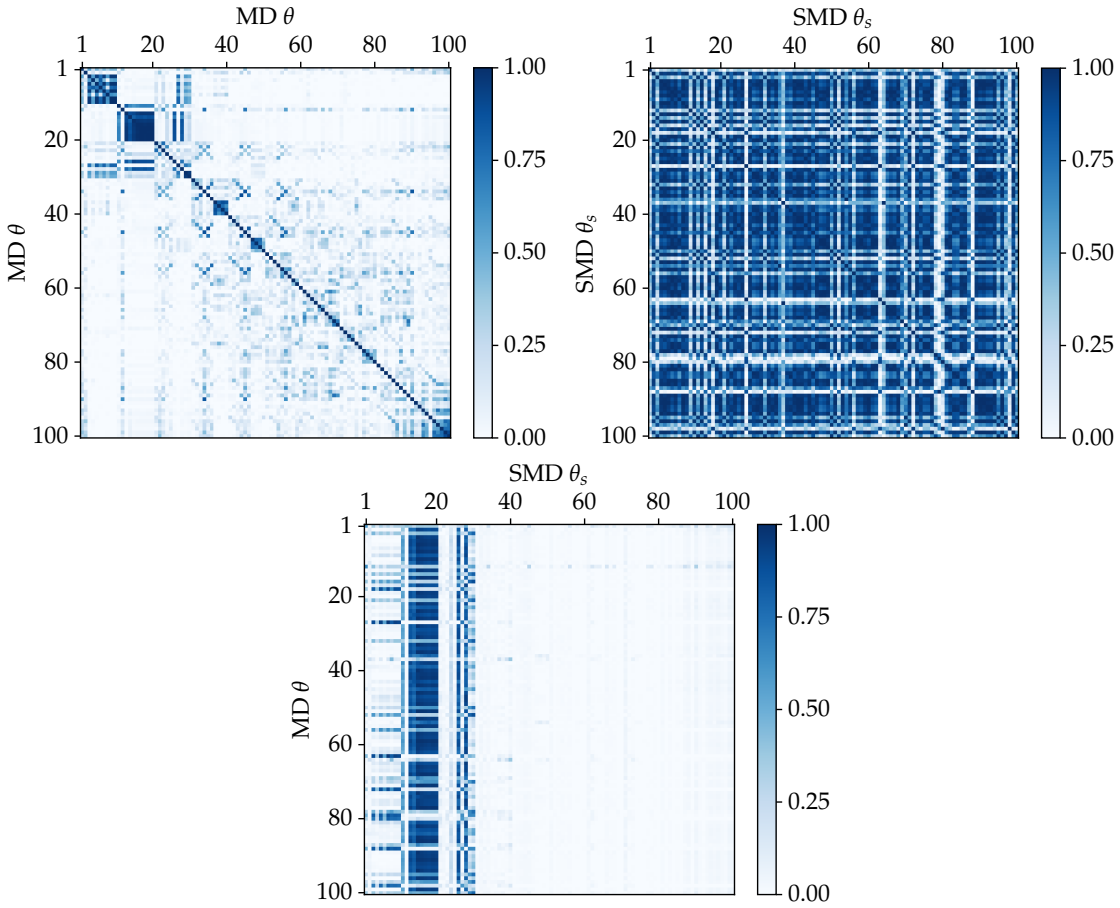


**Figure 6.11:** Auto-MAC of the MDs (top left), auto-MAC of the SMDs (top right) and MAC between MDs and SMDs (bottom) for the c-shape example.

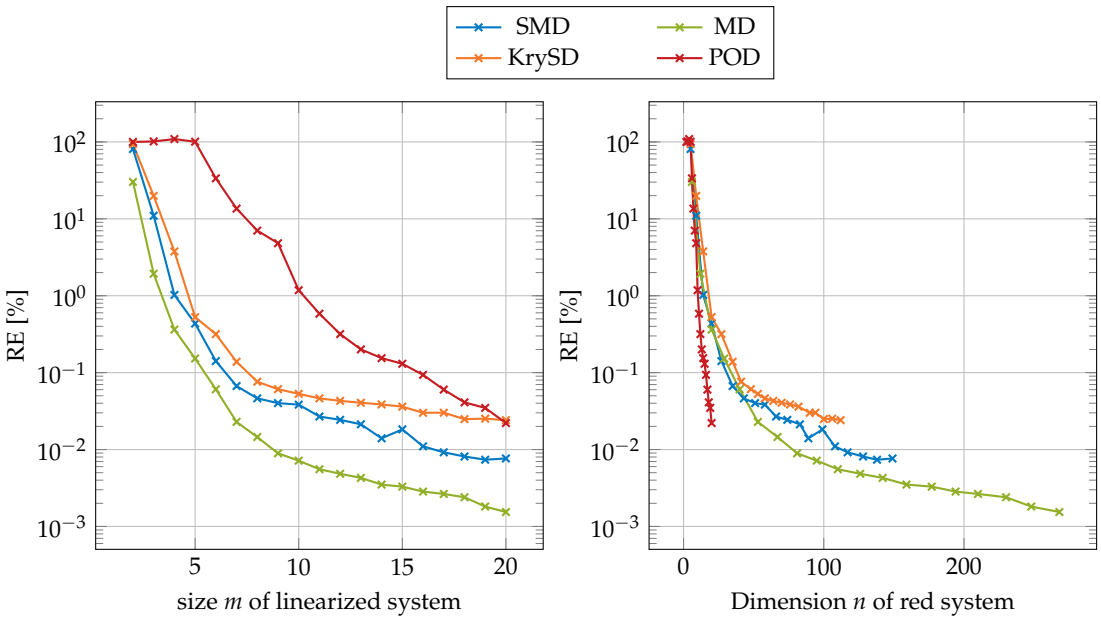


**Figure 6.12:** Error of different reduction methods with deflation for the c-shape example. The relative error is given over the dimension of the linear basis (left), and over the dimension of the reduced model (right). The POD reduction was trained on the full solution.

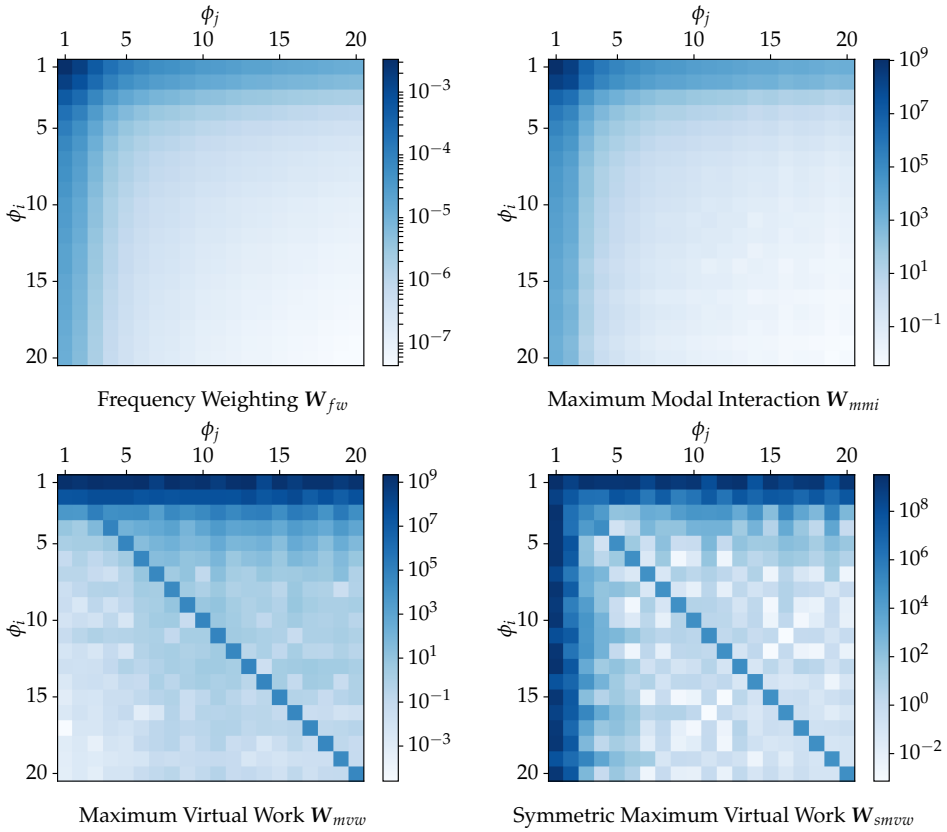


**Figure 6.13:** Weighting matrices for the c-shape example.

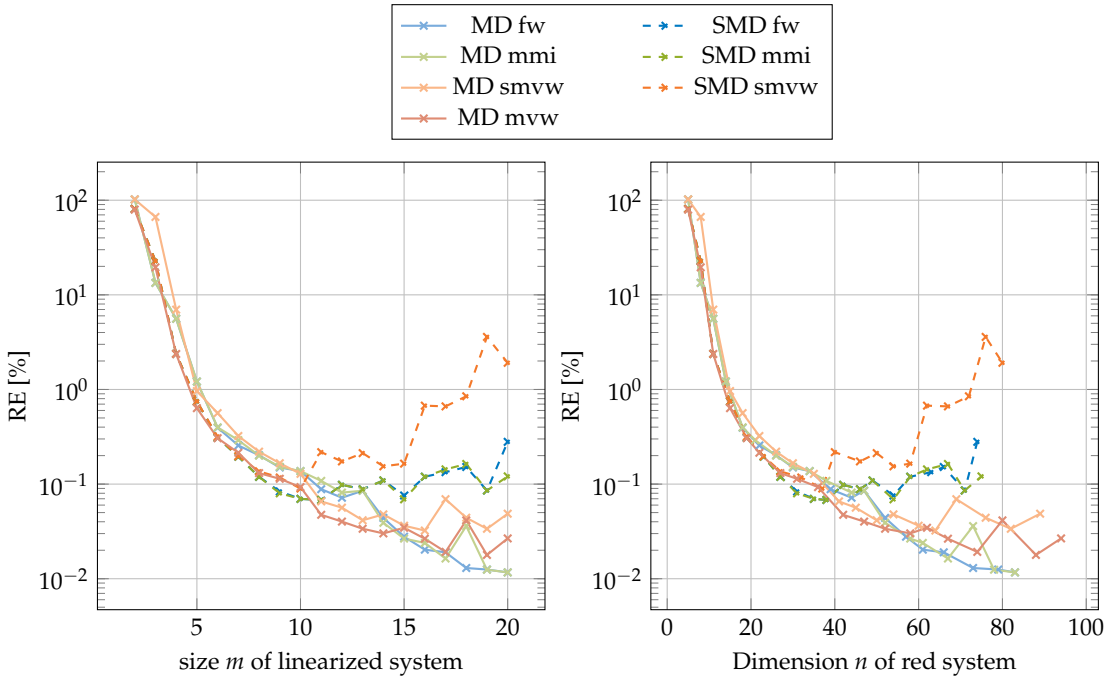


**Figure 6.14:** Comparison of different selection methods with deflation for the c-shape example. The relative error is given over the dimension of the linear basis (left), and over the dimension of the reduced model (right).

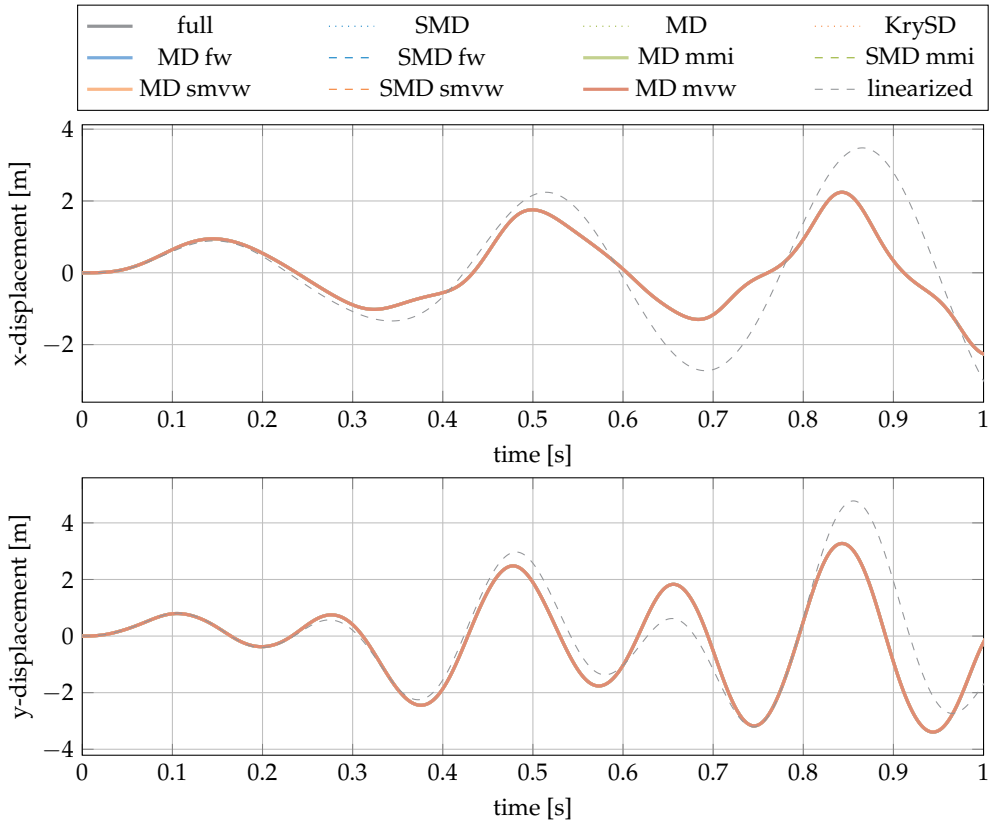


**Figure 6.15:** Displacement of the tip of the c-shape example for different simulation free reduction methods for $m = 6$.

# Chapter 7

# Quadratic Manifold

In the previous chapter, simulation-free reduction techniques were addressed. In these techniques the basis is formed of two ingredients: the so-called lin basis stemming from a linearized system and the augmentation vectors like (S)MDs, which account for the nonlinearity. One issue of these techniques is, that the number of (S)MDs grows quadratically with the size of the lin basis. This leads to very large bases when the lin basis is large, which limit the gain in computational time of the reduced basis projection.

One approach to tackle the problem is the use of selection criteria as discussed in Subsection 6.2.4 and Section 6.4. Another approach to alleviate the issue of the quadratically growing basis is the concept of quadratic manifolds, where the (S)MDs are not treated as independent dofs but are quadratically coupled with the modes of the lin basis. The number of unknowns is then reduced to a minimum. Then, the mapping from the generalized coordinates to the physical coordinates is nonlinear, leading to a different structure in the equations of motion.

However, the reduction technique proposed in this chapter is not suitable for all geometrically nonlinear problems. It is an attractive option for beam and shell structures, where the von Karman kinematic assumption is used for approximating the Green-Lagrange strain. Hence, in Section 7.3 the von Karman beam theory is introduced and the relation to the static condensation method discussed in Subsection 4.5.1 is studied in Subsection 7.3.2.

Nonlinear projection with the aim of model order reduction is already used in the Global Modal Parametrization technique introduced by Aarts and Jonker [AJ02] and extended by Brüls et al. [BDG07] and Naets et al. [Nae+11; Nae+12], where elastic multibody systems are reduced to a set of minimal coordinates using a database of configuration-specific mappings. A similar approach is also used by Tamarozzi et al. [THD14] and Blockmans et al. [Blo+15] to update the basis for contact problems, where the contact location is changing. A further approach to reduce mechanical systems using nonlinear projections is proposed by Millan and Arroyo [MA13], where the nonlinear projection is trained in a machine-learning style from a set of training snapshots. With the method proposed in this work, however, the manifold is defined to be quadratic leading to a simple mathematical structure not requiring database operations.

In this chapter first the nonlinear projection framework is introduced and then specified further to the quadratic manifold approach. After the discussion of a stabilization technique and the application of time integration schemes, the application to models using von Karman beams is discussed. The chapter ends with applications to various structures using von Karman beam elements and solid elements to illustrate the potential and the limits of this method. The quadratic manifold approach is published in [RR14b], [Jai+17] and [Rut+17], on which this chapter bases. Furthermore it is mentioned, that Wu and Tiso [WT16a] propose an approach of the quadratic manifold approach in the context

of nonlinear substructuring.

## 7.1 Key Idea: Nonlinear Projection

The key in the projectional reduction is the linear mapping (3.2) expressing the physical displacements $\boldsymbol{u}$ in terms of the reduced generalized coordinates $\boldsymbol{q}$ as $\boldsymbol{u} = \boldsymbol{V}\boldsymbol{q}$. In the framework of nonlinear projection, the physical displacements $\boldsymbol{u}$ are expressed as a nonlinear function

$$\boldsymbol{u} = \boldsymbol{\Gamma}(\boldsymbol{z}) \tag{7.1}$$

with the reduced generalized coordinates $\boldsymbol{z} \in \mathbb{R}^n$. $\boldsymbol{\Gamma} : \mathbb{R}^n \to \mathbb{R}^N$ is a nonlinear mapping from the reduced generalized coordinates to the physical coordinates, which is to be defined later. Since $\boldsymbol{\Gamma}$ is not an explicit function of time, the velocities and accelerations are then given as

$$\dot{\boldsymbol{u}} = \frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{z}}\dot{\boldsymbol{z}}, \qquad \ddot{\boldsymbol{u}} = \frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{z}}\ddot{\boldsymbol{z}} + \frac{\partial^2 \boldsymbol{\Gamma}}{\partial \boldsymbol{z}^2}\dot{\boldsymbol{z}}\dot{\boldsymbol{z}} \tag{7.2}$$

with the Jacobian $\partial\boldsymbol{\Gamma}/\partial\boldsymbol{z} \in \mathbb{R}^{N\times n}$ and the second derivative tensor $\partial^2\boldsymbol{\Gamma}/\partial\boldsymbol{z}^2 \in \mathbb{R}^{N\times n\times n}$. When the nonlinear transformation (7.1) and its derivatives (7.2) are inserted into the nonlinear equations of motion (2.41), one obtains the residual $\boldsymbol{r}$ similar to (3.3). As in Chapter 3, the principle of virtual work leads to the equations of motion for a reduced set of generalized coordinates, since the residual is forced to be orthogonal to the kinematically admissible displacements and is thus projected out. The variation of the displacement field $\boldsymbol{u}$ with the nonlinear mapping is given as

$$\delta\boldsymbol{u} = \frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{z}}\delta\boldsymbol{z} = \boldsymbol{P}_\Gamma \delta\boldsymbol{z} \tag{7.3}$$

with the tangent projector $\boldsymbol{P}_\Gamma = \partial\boldsymbol{\Gamma}/\partial\boldsymbol{z}$ being the Jacobian of the nonlinear mapping, which spans the tangent subspace of the kinematically admissible displacements $\delta\boldsymbol{u}$. Applying the principle of virtual work results in the nonlinear projected reduced equations of motion

$$\boldsymbol{P}_\Gamma^T\boldsymbol{M}\boldsymbol{P}_\Gamma\ddot{\boldsymbol{z}} + \boldsymbol{P}_\Gamma^T\boldsymbol{M}\frac{\partial^2\boldsymbol{\Gamma}}{\partial\boldsymbol{z}^2}\dot{\boldsymbol{z}}\dot{\boldsymbol{z}} + \boldsymbol{P}_\Gamma^T\boldsymbol{C}\boldsymbol{P}_\Gamma\dot{\boldsymbol{z}} + \boldsymbol{P}_\Gamma^T\boldsymbol{f}(\boldsymbol{\Gamma}(\boldsymbol{z})) = \boldsymbol{P}_\Gamma^T\boldsymbol{g}, \tag{7.4}$$

which can be rearranged to

$$\widetilde{\boldsymbol{M}}_r\ddot{\boldsymbol{z}} + \widetilde{\boldsymbol{p}} + \widetilde{\boldsymbol{C}}_r\dot{\boldsymbol{z}} + \widetilde{\boldsymbol{f}}_r(\boldsymbol{z}) = \widetilde{\boldsymbol{g}}_r(\boldsymbol{z}, t). \tag{7.5}$$

The reduced mass matrix $\widetilde{\boldsymbol{M}}_r = \boldsymbol{P}_\Gamma^T\boldsymbol{M}\boldsymbol{P}_\Gamma \in \mathbb{R}^{n\times n}$ is state dependent, $\widetilde{\boldsymbol{C}}_r = \boldsymbol{P}_\Gamma^T\boldsymbol{C}\boldsymbol{P}_\Gamma \in \mathbb{R}^{n\times n}$ is the reduced damping matrix, $\widetilde{\boldsymbol{f}}_r(\boldsymbol{z}) = \boldsymbol{P}_\Gamma^T\boldsymbol{f}(\boldsymbol{\Gamma}(\boldsymbol{z})) \in \mathbb{R}^n$ the reduced internal force vector and $\widetilde{\boldsymbol{g}}_r = \boldsymbol{P}_\Gamma^T\boldsymbol{g}$ the reduced external force vector. The term $\widetilde{\boldsymbol{p}} = \boldsymbol{P}_\Gamma^T\boldsymbol{M}\frac{\partial^2\boldsymbol{\Gamma}}{\partial\boldsymbol{z}^2}\dot{\boldsymbol{z}}\dot{\boldsymbol{z}}$ can be interpreted as a convective term which is state dependent, proportional to the squared velocities and proportional to the curvature of the nonlinear mapping $\boldsymbol{\Gamma}(\boldsymbol{z})$. The structure of equation (7.5) is well-known from multibody dynamics and finite elements, where frames are rotating and thus the mapping of the derivative of the generalized coordinates to the velocities involves a state depending mapping similar to (7.2).

The equations of motion (7.5) describe the dynamics of a reduced system for an arbitrary nonlinear mapping $\boldsymbol{\Gamma}(\boldsymbol{z})$. If the mapping is linear, as in the common projective model order reduction, the question of reduction boils down to the question of the subspace spanned by the basis $\boldsymbol{V}$. However, for a nonlinear mapping, first the structure of the nonlinearity is to be chosen before the parameters of the mapping are determined.

## 7.2 Mapping on Quadratic Manifold

In the Quadratic Manifold (QM) approach, the mapping $\boldsymbol{\Gamma}(\boldsymbol{z})$ is chosen to be a quadratic function of $\boldsymbol{z}$. This allows to keep the dimension of the reduced generalized coordinates $\boldsymbol{z}$ small while still incorporating the information of MDs or SMDs. However, first the general framework of quadratic projection is addressed before the incorporation of MDs, SMDs and SDs is discussed. The quadratic mapping can be expressed as

$$\boldsymbol{\Gamma} = \boldsymbol{V}\boldsymbol{z} + \frac{1}{2}(\boldsymbol{\Theta}\boldsymbol{z})\boldsymbol{z} \tag{7.6}$$

with the linear part of the transformation $\boldsymbol{V} \in \mathbb{R}^{N\times n}$ and the quadratic part $\boldsymbol{\Theta} \in \mathbb{R}^{N\times n\times n}$. In index notation, the quadratic mapping (7.6) is written as

$$u_i = V_{ij}z_j + \frac{1}{2}\Theta_{ijk}z_jz_k. \tag{7.7}$$

To make the mapping unique, $\boldsymbol{\Theta}$ has to be symmetric with respect to the last two indices, since the anti symmetric part is eliminated by the quadratic form of $\boldsymbol{z}$. To illustrate this, an anti-symmetric three dimensional tensor $\boldsymbol{\Lambda} \in \mathbb{R}^{N\times n\times n}$ is added to the symmetric $\boldsymbol{\Theta}$. If $\Lambda_{ijk} = -\Lambda_{ikj}$, then the mapping (7.6) yields

$$\begin{aligned}
\Gamma_i &= V_{ij}z_j + \frac{1}{2}(\Theta_{ijk} + \Lambda_{ijk})z_jz_k \\
&= V_{ij}z_j + \frac{1}{2}\Theta_{ijk}z_jz_k + \frac{1}{2}\Lambda_{ijk}z_jz_k \\
&= V_{ij}z_j + \frac{1}{4}\Theta_{ijk}z_jz_k + \frac{1}{4}\Lambda_{ijk}z_jz_k + \frac{1}{4}\Lambda_{ijk}z_jz_k \\
&= V_{ij}z_j + \frac{1}{4}\Theta_{ijk}z_jz_k + \frac{1}{4}\Lambda_{ijk}z_jz_k - \frac{1}{4}\Lambda_{ikj}z_jz_k \\
&= V_{ij}z_j + \frac{1}{4}\Theta_{ijk}z_jz_k + \frac{1}{4}\Lambda_{ijk}z_jz_k - \frac{1}{4}\Lambda_{ijk}z_kz_j \\
&= V_{ij}z_j + \frac{1}{4}\Theta_{ijk}z_jz_k
\end{aligned} \tag{7.8}$$

eliminating the anti-symmetric part of the quadratic mapping (7.6). Consequently, $\boldsymbol{\Theta}$ has only $N \cdot n \cdot (n+1)/2$ independent entries, while the remaining entries are defined with the symmetry constraint.

The velocities and the accelerations of the physical dofs are then expressed as

$$\dot{\boldsymbol{u}} = \dot{\boldsymbol{\Gamma}} = \boldsymbol{P}_\Gamma\dot{\boldsymbol{z}}, \quad \ddot{\boldsymbol{u}} = \ddot{\boldsymbol{\Gamma}} = \boldsymbol{P}_\Gamma\ddot{\boldsymbol{z}} + \boldsymbol{\Theta}\dot{\boldsymbol{z}}\dot{\boldsymbol{z}}, \quad \boldsymbol{P}_\Gamma = \boldsymbol{V} + \Theta\boldsymbol{z}. \tag{7.9}$$

with the tangent projector $\boldsymbol{P}_\Gamma \in \mathbb{R}^{N\times n}$ being a function depending linearly on the generalized quadratic coordinates $\boldsymbol{z}$. If the mapping is chosen to be quadratic, not only the linear part $\boldsymbol{V}$ but also the nonlinear part $\boldsymbol{\Theta}$ of the mapping need to be defined. In the following, two options are presented to form the quadratic part, if the basis is filled with vibration modes. Furthermore, the force compensation method is presented to build $\boldsymbol{\Theta}$ for an arbitrary linear basis $\boldsymbol{V}$.

### 7.2.1 Modal Derivatives

If the linear part of the quadratic mapping (7.6) is composed of vibration modes, the quadratic part $\boldsymbol{\Theta}$ should capture the change of the vibration modes with respect to a change of the geometrical configuration. This change is expressed by MDs, which are

constructed by the perturbation of the eigenvalue problem and hence capture the change of one mode with respect to a geometry change in the direction of another mode shape.

Consequently, the quadratic part $\boldsymbol{\Theta}$ can be composed of MDs when the linear part $\boldsymbol{V}$ is built using vibration modes. However, since the MDs are not symmetric with respect to the last two indices $\theta_{ij} \neq \theta_{ji}$, the quadratic tensor $\boldsymbol{\Theta}$ is to be composed of the symmetric part of the tensor $\boldsymbol{\Omega} \in \mathbb{R}^{N\times n\times n}$ which is filled with the MDs $\boldsymbol{\theta}_{ij} = \partial \boldsymbol{\phi}_i / \partial \eta_j$:

$$\boldsymbol{\Omega}[:, i, j] = \boldsymbol{\theta}_{ij} \tag{7.10}$$

The quadratic tensor $\boldsymbol{\Theta}$ of the mapping is then given as the symmetric part of $\boldsymbol{\Omega}$ yielding

$$\Theta_{ijk} = \frac{1}{2}\left(\Omega_{ijk} + \Omega_{ikj}\right). \tag{7.11}$$

The approach using MDs as the quadratic extension for a linear part $\boldsymbol{V}$ composed of vibration modes is motivated by the conceptual idea of MDs. Since they represent the perturbation of modes, it seems to be a reasonable quadratic extension to the linear mapping of modes. The concept of using MDs as quadratic extension is purely heuristic. The suitability of this approach is illustrated in the Applications in Section 7.4.

The SMDs also capture the change of modes in a different fashion than the MDs. Hence, they can also be regarded as reasonable extensions for the quadratic mapping $\boldsymbol{\Theta}$.

### 7.2.2 Static Modal Derivatives

As a second approach, the quadratic part $\boldsymbol{\Theta}$ can be composed of SMDs if the linear part $\boldsymbol{V}$ is built using vibration modes. Since the SMDs are symmetric with respect to the last two indices, they form directly the second order tensor $\boldsymbol{\Theta}$:

$$\boldsymbol{\Theta}[:, i, j] = \boldsymbol{\theta}_{s,ij} \tag{7.12}$$

with $\boldsymbol{\theta}_{s,ij} = \partial \boldsymbol{\phi}_i / \partial \eta_j |^s$. It should be pointed out, that the MDs and the SMDs are built using different assumptions. While the MDs stem from the perturbation of an eigenvalue problem, where the nonlinear system is made both linear and parametric, the concept of the SMDs is different. They are solely based on the quadratic part of the nonlinearity of the forces which can be given a different interpretation in the context of a QM mapping. This other route to define SDs named force compensation method or force compensation approach is given next.

### 7.2.3 Force Compensation Method

In the previous two sections, the quadratic part $\boldsymbol{\Theta}$ of the nonlinear mapping was built either from MDs or SMDs in the case, that $\boldsymbol{V}$ is built from vibration modes. When using SMDs in the quadratic part, however, a condition is satisfied which can be generalized to the force compensation method. The motivation of this approach is that the quadratic nonlinearity of the internal forces is swapped to the quadratic manifold, so that the internal forces have not quadratic terms on the manifold.

When only the linear part $\boldsymbol{V}$ is given as vibration modes or any other type of modes, the quadratic part can be uniquely determined, when the second derivative of the nonlinear forces with respect to the generalized quadratic coordinate $\boldsymbol{z}$ is forced to be zero:

$$\left.\frac{\partial^2 \boldsymbol{f}(\boldsymbol{\Gamma}(\boldsymbol{z}))}{\partial \boldsymbol{z}^2}\right|_{\boldsymbol{z}=\boldsymbol{0}} = \boldsymbol{0}. \tag{7.13}$$

This condition defines the quadratic manifold such that it compensates the quadratic part of the nonlinear forces with respect to the generalized coordinates $\boldsymbol{z}$. Then, the quadratic tensor $\boldsymbol{\Theta}$ can be derived from (7.13) in the following manner:

After applying the chain rule yielding

$$\left[\left(\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{u}^2}\cdot\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{z}}\right)\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{z}}+\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\frac{\partial^2 \boldsymbol{u}}{\partial \boldsymbol{z}^2}\right]_{\boldsymbol{z}=\boldsymbol{0}}=\boldsymbol{0}, \tag{7.14}$$

with the first and second derivative of the quadratic mapping given as

$$\left.\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{z}}\right|_{\boldsymbol{z}=\boldsymbol{0}}=\boldsymbol{V},\qquad \left.\frac{\partial^2 \boldsymbol{u}}{\partial \boldsymbol{z}^2}\right|_{\boldsymbol{z}=\boldsymbol{0}}=\boldsymbol{\Theta}, \tag{7.15}$$

and the definition of the stiffness matrix $\boldsymbol{K}=\left.\frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\boldsymbol{u}}\right|_{\boldsymbol{u}=\boldsymbol{0}}$, one obtains the equation for the third order tensor $\boldsymbol{\Theta}$ as

$$\left(\left.\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{u}^2}\right|_{\boldsymbol{u}=\boldsymbol{0}}\cdot\boldsymbol{V}\right)\boldsymbol{V}+\boldsymbol{K}\cdot\boldsymbol{\Theta}=\boldsymbol{0}. \tag{7.16}$$

From (7.16), the tensor $\boldsymbol{\Theta}$ is uniquely defined if $\boldsymbol{K}$ is not rank deficient. In other words, there exists a unique quadratic tensor $\boldsymbol{\Theta}$ compensating the nonlinear forces to meet condition (7.13) for *any* linear basis $\boldsymbol{V}$. Thereby, $\boldsymbol{V}$ does not have to be composed of vibration modes, but can be composed of any suitably chosen set of linearly independent vectors. Furthermore, $\boldsymbol{\Theta}$ is symmetric with respect to the last two indices, i.e., $\Theta_{ijk}=\Theta_{ikj}$.

For further analysis (7.16) is recast to index notation yielding

$$\left(\left.\frac{\partial^2 f_k}{\partial u_l \partial u_m}\right|_{\boldsymbol{u}=\boldsymbol{0}}\right)V_{li}V_{mj}+K_{kl}\boldsymbol{\theta}_{lij}=0, \tag{7.17}$$

where $k,l,m\in\{1,\ldots,N\}$ are the indices of the physical domain, and $i,j\in\{1,\ldots,n\}$ are of the reduced domain. Since (7.16) is evaluated at $\boldsymbol{u}=\boldsymbol{0}$, the matrix $\boldsymbol{V}$ can be interpreted as the linear mapping of the reduced, linear generalized coordinates $\boldsymbol{q}$ to the full displacements $\boldsymbol{u}=\boldsymbol{V}\boldsymbol{q}$ as in the linear basis projection discussed in Chapter 3. Then the partial derivative $\partial u_m/\partial q_j|_{\boldsymbol{u}=\boldsymbol{0}}$ yields $V_{mj}$ and one can rewrite (7.17) using $\partial f_k/\partial u_l|_{\boldsymbol{u}=\boldsymbol{0}}=K_{kl}$ to

$$\frac{\partial}{\partial u_m}\left(\frac{\partial f_k}{\partial u_l}\right)\left.\frac{\partial u_m}{\partial q_j}\right|_{\boldsymbol{u}=\boldsymbol{0}}V_{li}+K_{kl}\boldsymbol{\theta}_{lij}=0 \tag{7.18}$$

$$\Longrightarrow\ \frac{\partial}{\partial u_m}\left(K_{kl}\right)\left.\frac{\partial u_m}{\partial q_j}\right|_{\boldsymbol{u}=\boldsymbol{0}}V_{li}+K_{kl}\boldsymbol{\theta}_{lij}=0. \tag{7.19}$$

With the chain rule, (7.19) can be simplified to

$$\frac{\partial K_{kl}}{\partial q_j}V_{li}+K_{kl}\boldsymbol{\theta}_{lij}=0. \tag{7.20}$$

The expression above can be rewritten in matrix-vector notation with $l$ being the index of column vectors and $k$ being the row-index of matrices as

$$\frac{\partial \boldsymbol{K}}{\partial q_j}\boldsymbol{v}_i+\boldsymbol{K}\boldsymbol{\theta}_{ij}=\boldsymbol{0}\quad\Leftrightarrow\quad\boldsymbol{K}\boldsymbol{\theta}_{ij}=-\frac{\partial \boldsymbol{K}}{\partial q_j}\boldsymbol{v}_i. \tag{7.21}$$

The solution of (7.21) which yields the Static Derivatives (SDs) $\boldsymbol{\theta}_{s,ij}$ is equivalent to the definition of the SMDs (7.21), if the linear part $\boldsymbol{V}$ of the quadratic mapping is chosen to

consist of vibration modes: $\boldsymbol{V} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_n]$. Hence, SMDs are obtained as a special case of the general framework described by the *force compensation approach* using (7.13).

The approach presented above yields the same static derivatives as presented in Subsection 6.2.2. In the force compensation approach, the nonlinearity is mapped from the internal forces to the quadratic part of the basis, leading to exactly the static derivatives in the quadratic part. The fact that the two interpretations are equivalent for vibration modes enables one to extend the concept of quadratic manifolds to bases different from vibration modes, such as Krylov subspace vectors or other linear reduction techniques. Since both concepts are interrelated, a physical interpretation of the SDs obtained within the QM framework should be given, which is also illustrated in Figure 7.1.

For two given linear displacement fields $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, where for instance $\boldsymbol{v}_1$ is a static displacement mode of a unit force at the tip of the beam and $\boldsymbol{v}_2$ is the second vibration mode, there exists a unique corresponding force distribution $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ such that $\boldsymbol{f}_i = \boldsymbol{K}\boldsymbol{v}_i : i \in \{1, 2\}$. If both force distributions are combined and applied to the *nonlinear system*, the resulting displacements can be split in two contributions: the linear displacements $\boldsymbol{v}_1 + \boldsymbol{v}_2$, and a nonlinear correction resulting from the combination of both forces and the nonlinearity. As is shown below, the quadratic part of this nonlinear contribution contains *exactly* all SDs associated to the displacement fields $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$.

The nonlinear forces can be expanded using a Taylor expansion up to the quadratic part as

$$\boldsymbol{f} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\boldsymbol{u} + \frac{1}{2}\frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{u}^2}\boldsymbol{u}\boldsymbol{u} + \mathcal{O}(u^3) = \boldsymbol{K}\boldsymbol{u} + \boldsymbol{K}^{(2)}\boldsymbol{u}\boldsymbol{u} + \mathcal{O}(||\boldsymbol{u}||^3) \tag{7.22}$$

with the linear stiffness matrix $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ and the second order stiffness tensor $\boldsymbol{K}^{(2)} \in \mathbb{R}^{N\times N\times N}$. The external forces $\boldsymbol{g}$ in Figure 7.1 are the response of the linearized system which is perturbed with the displacements $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, i.e.,

$$\boldsymbol{g} = \epsilon \boldsymbol{K}(\boldsymbol{v}_1 + \boldsymbol{v}_2), \tag{7.23}$$

where $\epsilon > 0$ is a load scaling factor. Setting the static equilibrium of the external forces $\boldsymbol{g}$ with the Taylor expanded internal forces (7.22), one obtains

$$\boldsymbol{g} = \boldsymbol{f} \tag{7.24}$$

$$\implies \epsilon \boldsymbol{K}(\boldsymbol{v}_1 + \boldsymbol{v}_2) = \boldsymbol{K}\boldsymbol{u} + \frac{1}{2}\boldsymbol{K}^{(2)}\boldsymbol{u}\boldsymbol{u} + \mathcal{O}(||\boldsymbol{u}||^3)\,, \tag{7.25}$$

with the unknown displacement field $\boldsymbol{u}$. When $\epsilon$ is assumed to be small, the solution can be expanded with the bookkeeping notation as

$$\mathbf{u} = \epsilon \boldsymbol{u}_{(1)} + \epsilon^2 \boldsymbol{u}_{(2)} + \ldots\,. \tag{7.26}$$

Substituting (7.26) into (7.25) and comparing coefficients of different powers of $\epsilon$, one obtains the leading order coefficient to the solution $\boldsymbol{u}$ as

$$\boldsymbol{u}_{(1)} = \boldsymbol{v}_1 + \boldsymbol{v}_2, \tag{7.27}$$

and the second order approximation as

$$\boldsymbol{u}_{(2)} = -\frac{1}{2}\boldsymbol{K}^{-1}\boldsymbol{K}^{(2)}(\boldsymbol{v}_1 + \boldsymbol{v}_2)(\boldsymbol{v}_1 + \boldsymbol{v}_2)\,. \tag{7.28}$$

Now, the SD $\boldsymbol{\theta}_{ij}$ is defined according to (7.17) as

$$\boldsymbol{\theta}_{ij} = -\boldsymbol{K}^{-1}\boldsymbol{K}^{(2)}\boldsymbol{v}_i\boldsymbol{v}_j. \tag{7.29}$$

Thus, by using the fact that the SDs are symmetric, i.e., $\boldsymbol{\theta}_{s,12} = \boldsymbol{\theta}_{s,21}$, $\boldsymbol{u}_{(2)}$ from (7.28) can be expressed in terms of SDs as

$$\boldsymbol{u}_{(2)} = \frac{1}{2}(\boldsymbol{\theta}_{s,11} + \boldsymbol{\theta}_{s,12} + \boldsymbol{\theta}_{s,21} + \boldsymbol{\theta}_{s,22}) = \boldsymbol{\theta}_{s,12} + \frac{1}{2}(\boldsymbol{\theta}_{s,11} + \boldsymbol{\theta}_{s,22}). \tag{7.30}$$

Consequently, the displacement field is composed of all three static derivatives, $\boldsymbol{\theta}_{s,11}$, $\boldsymbol{\theta}_{s,22}$ and $\boldsymbol{\theta}_{s,12}$.

### 7.2.4 Stabilization Through Orthogonalization

In some cases, the tangent projector $\boldsymbol{P}_\Gamma = \boldsymbol{V} + \boldsymbol{\Theta}\boldsymbol{z}$ is not well conditioned. As discussed in Subsection 6.2.3, the subspace spanned by the linear part $\boldsymbol{V}$ formed by vibration modes or other basis modes and the quadratic part $\boldsymbol{\Theta}$ formed by MDs or SDs is not necessarily distinct. Then, for certain generalized coordinates $\boldsymbol{z}$, the tangent projector $\boldsymbol{P}_\Gamma$ might be poorly conditioned leading to poor or even no convergence in the Newton-Raphson loop of the time integration scheme. To solve this issue, a Gram-Schmidt like orthogonalization of the quadratic part with respect to the linear part can be applied, which alleviates the convergence issue. The orthogonalization can be written as

$$\boldsymbol{\Theta}_{\perp \boldsymbol{V}} = (\boldsymbol{I} - \sum_{i=1}^{n} \boldsymbol{v}_i \boldsymbol{v}_i^T)\boldsymbol{\Theta}, \quad \text{with} \quad \boldsymbol{v}_i^T \boldsymbol{v}_i = 1 \tag{7.31}$$

so that the orthogonality condition of $\boldsymbol{\Theta}_{\perp \boldsymbol{V}}$ and $\boldsymbol{V}$ is fulfilled:

$$\boldsymbol{V}^T \boldsymbol{\Theta}_{\perp \boldsymbol{V}} = \boldsymbol{0}. \tag{7.32}$$

This orthogonalization, however, breaks the quadratic manifold. If $\boldsymbol{\Theta}$ is built using SDs for instance, the condition (7.13) is violated. In the other case, when the QM is constructed with MDs, the MDs are deteriorated so that the projection is performed on a different manifold which is stabilized but not the original one.

The effect of the orthogonalization is commonly dependent on the size of the linear part $\boldsymbol{V}$ of the mapping. In the extreme case of $\boldsymbol{V}$ being square and having full rank, the orthogonalization (7.31) leads to $\boldsymbol{\Theta}_{\perp \boldsymbol{V}} = \boldsymbol{0}$. However, if $\boldsymbol{V}$ is only composed of few modes, the deterioration effect on the quadratic mapping due to the orthogonalization is generally mild.

### 7.2.5 Time Integration

For solving the equations of motion (7.4) for a given set of initial reduced displacements $\boldsymbol{z}_0 = \boldsymbol{z}(t = 0)$ and velocities $\dot{\boldsymbol{z}}_0 = \dot{\boldsymbol{z}}(t = 0)$, a time stepping integration scheme like the generalized-$\alpha$ method as outlined in Section 2.4 is necessary.

The generalized-$\alpha$ integration scheme has two different balancing time shifts $\alpha_m$ for the acceleration forces and $\alpha_f$ for the internal, external and damping forces. However, since in the QM approach the basis is changed with the current displacement state, the idea of balancing, which leads to an optimized numerical damping behavior, is hardly physically reasonable for state dependent bases. In order to keep the time integration variationally consistent, so that both the acceleration forces as well as the internal, external and damping forces are acting in the same tangential subspace $\boldsymbol{P}_\Gamma$, the time integration recommended and used in the examples is the Newmark scheme, where the balancing time shifts are set to zero, i.e. $\alpha_f = \alpha_m = 0$. As a consequence, the tangential subspace is consistent with the reconstructed displacement field $\boldsymbol{u}_{j+1} = \boldsymbol{\Gamma}(\boldsymbol{q}_{j+1})$. The numerical damping can be adjusted by the damping constant $\alpha \geq 0$, which determines the parameters $\beta$ and $\gamma$ as

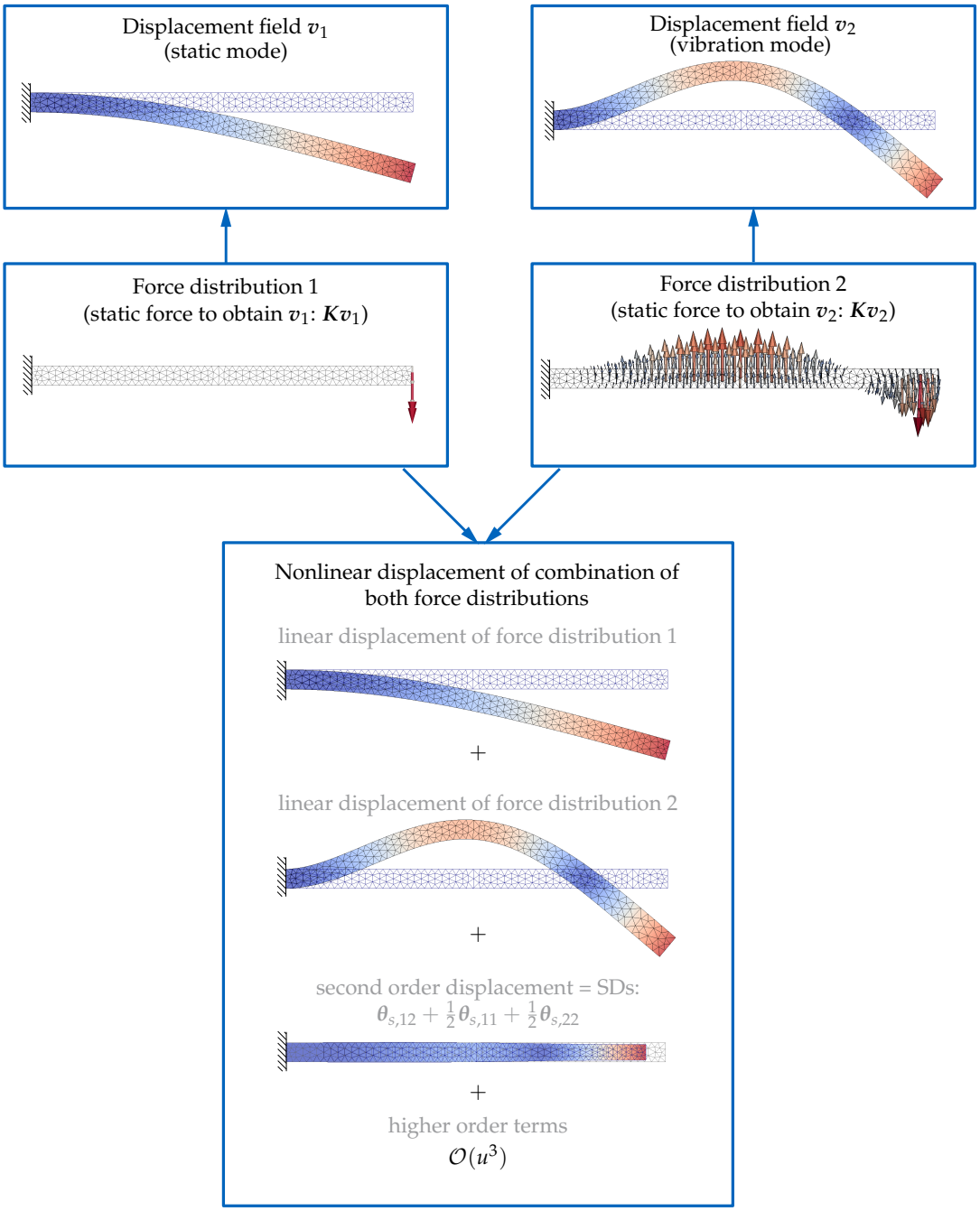


**Figure 7.1:** Physical interpretation of the static derivatives (SDs): Two force distributions yielding a linear displacement mode are applied to the nonlinear static problem. The result is a combination of the two linear displacement modes, the SDs in a second order expansion and higher order terms ($\mathcal{O}(u^3)$).
Note that for this beam example, both parent modes feature transverse displacements only, while the SD features only an axial field.

$$\gamma = \frac{1}{2} + \alpha, \qquad \beta = \frac{1}{4}(1+\alpha)^2, \qquad 0 \leq \alpha \leq 0.3. \tag{7.33}$$

Numerical damping is introduced by choosing $\alpha > 0$. However, this choice destroys the second order accuracy of the integration scheme, since the Newmark scheme is only second order accurate for $\alpha = 0$. In the author's experience, the QM approach yields best results with the Newmark integration scheme. Hence, this scheme using $\alpha_f = \alpha_m = 0$ is derived in the following.

Similar to the balance equation (2.50), the balance equation for the $j+1$-th time step of the QM projected system is written as

$$\widetilde{\boldsymbol{M}}_{r,j+1}\ddot{\boldsymbol{z}}_{j+1} + \widetilde{\boldsymbol{p}}_{j+1} + \widetilde{\boldsymbol{C}}_{r,j+1}\dot{\boldsymbol{z}}_{j+1} + \widetilde{\boldsymbol{f}}_{r,j+1}(\boldsymbol{z}_{j+1}) = \widetilde{\boldsymbol{g}}_{r,j+1}(\boldsymbol{z}_{j+1}, t_{j+1}). \tag{7.34}$$

The reduced quantities are obtained with the tangent projector $\boldsymbol{P}_\Gamma$ given with the reduced generalized displacements $\boldsymbol{z}_{j+1}$ at the $j+1$-th time step:

$$\boldsymbol{P}_\Gamma = \boldsymbol{V} + \boldsymbol{\Theta}\boldsymbol{z}_{j+1}. \tag{7.35}$$

Consequently, the residual of the force balance equation (7.34) yields

$$\begin{aligned} \boldsymbol{r}(\boldsymbol{z}_{j+1}) =& \boldsymbol{P}_\Gamma^T \left( \boldsymbol{M}\boldsymbol{P}_\Gamma\ddot{\boldsymbol{z}}_{j+1} + \boldsymbol{M}\boldsymbol{\Theta}\dot{\boldsymbol{z}}_{j+1}\dot{\boldsymbol{z}}_{j+1} + \boldsymbol{C}\boldsymbol{P}_\Gamma\dot{\boldsymbol{z}}_{j+1} + \boldsymbol{f}(\boldsymbol{\Gamma}(\boldsymbol{z}_{j+1})) - \boldsymbol{g}_{j+1} \right) && (7.36) \\ =& \boldsymbol{P}_\Gamma^T \boldsymbol{r}_{\text{full}}(\boldsymbol{z}_{j+1}), && (7.37) \end{aligned}$$

which is solved for in every iteration step with a Newton-Raphson iteration. The Jacobian $\boldsymbol{K}^i_{dyn,j+1} = \partial \boldsymbol{r}(\boldsymbol{z}_{j+1})/\partial \boldsymbol{z}_{j+1}$ necessary for the iteration is given exactly as

$$\begin{aligned} \boldsymbol{K}_{dyn,j+1} = \; & \frac{1}{\beta h^2}\boldsymbol{P}_\Gamma^T\boldsymbol{M}\boldsymbol{P}_\Gamma + \frac{\gamma}{\beta h}\left( \boldsymbol{P}_\Gamma^T\boldsymbol{C}\boldsymbol{P}_\Gamma + 2\boldsymbol{P}_\Gamma^T\boldsymbol{M}\boldsymbol{\Theta}\dot{\boldsymbol{z}}_{j+1} \right) + \boldsymbol{P}_\Gamma^T\boldsymbol{K}\left(\boldsymbol{\Gamma}(\boldsymbol{z}_{j+1})\right)\boldsymbol{P}_\Gamma \\ & + \boldsymbol{P}_\Gamma^T\left( \boldsymbol{M}\boldsymbol{\Theta}\ddot{\boldsymbol{z}}_{j+1} + \boldsymbol{C}\boldsymbol{\Theta}\dot{\boldsymbol{z}}_{j+1} \right) + \boldsymbol{\Theta}^T\boldsymbol{r}_{\text{full}}(\boldsymbol{z}_{j+1}) \end{aligned} \tag{7.38}$$

with the residual of the full system $\boldsymbol{r}_{\text{full}}(\boldsymbol{z}_{j+1})$ from (7.37). The transposed third order tensor multiplication $\boldsymbol{\Theta}^T$ in the last term is expressed in index notation as

$$\boldsymbol{\Theta}^T\boldsymbol{r}_{\text{full}} = \Theta_{kil}\, r_{\text{full},k} \tag{7.39}$$

with $i$ and $l$ being the row and column index of the resulting matrix.

The Jacobian given in (7.38) is the exact Jacobian leading to optimal quadratic convergence in the Newton-Raphson iteration. However, it involves several terms arising due to the change of the tangential projector $\boldsymbol{P}_\Gamma$, which can be neglected. The approximate Jacobian $\boldsymbol{K}_{dyn,j+1} \approx \boldsymbol{K}_{dyn,\text{approx},j+1}$ then yields

$$\boldsymbol{K}_{dyn,\text{approx},j+1} = \frac{1}{\beta h^2}\boldsymbol{P}_\Gamma\boldsymbol{M}\boldsymbol{P}_\Gamma + \frac{\gamma}{\beta h}\boldsymbol{P}_\Gamma\boldsymbol{C}\boldsymbol{P}_\Gamma + \boldsymbol{P}_\Gamma^T\boldsymbol{K}(\boldsymbol{\Gamma}(\boldsymbol{z}_{j+1}))\boldsymbol{P}_\Gamma. \tag{7.40}$$

which can also be used in the Newton-Raphson iteration instead of $\boldsymbol{K}_{dyn,j+1}$ from (7.38). The convergence rate is not as good as with the exact Jacobian, though. In the author's experience it is problem dependent, if the approximate or the full Jacobian yields to faster computations. In the given examples, the time integration is performed with full Jacobians to benefit of the optimal convergence behavior.

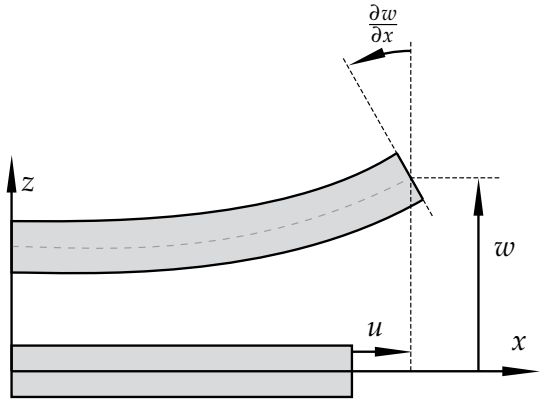


**Figure 7.2:** Kinematics of an Euler-Bernoulli beam.

## 7.3 Quadratic Manifold for von Karman Structures

The quadratic mapping introduced in Section 7.2 enslaves the displacement fields gathered in $\boldsymbol{\Theta}$ quadratically with the displacement fields in $\boldsymbol{V}$. Hence, the QM approach is only expedient, when the dominating nonlinearity is of quadratic nature. This is the case in beam and shell formulations using the von Karman kinematic assumption. This assumption is basically a simplification of the Green-Lagrange strain tensor, which is valid for moderate displacements. In order to investigate the QM approach for von Karman beams, the kinematic formulation and the polynomial structure of a beam with von Karman assumption is investigated in the following. Furthermore, the relation to the Guyan reduction technique described in Subsection 4.5.1 is discussed.

### 7.3.1 Kinematic Formulation and Polynomial Structure

For the derivation of the von Karman beam formulation (cf. [C+12, pp. 119 f.]), a beam as depicted in Figure 7.2 is considered. The neutral axis, which is the set of all centroids of the cross sections along the beam, is aligned with the $x$-axis in the initial configuration. The vector of the reference configuration $\bar{\boldsymbol{X}} = \begin{pmatrix} x & z \end{pmatrix}^T$ is given with the $x$ and $z$ position of the neutral axis in the reference configuration.

According to the Euler-Bernoulli beam theory, the neutral axis has to stay orthogonal to the cross section of the beam. Then the displacement field $\bar{\boldsymbol{u}}$ of the cross section can be expressed as

$$\bar{\boldsymbol{u}} = \begin{pmatrix} u - z\frac{\partial w}{\partial x} \\ w \end{pmatrix} \tag{7.41}$$

with the slope $\partial w / \partial \bar{x}$ of the beam. Note, that $u$ and $w$ are the displacements of the neutral axis. The deformation gradient $\boldsymbol{F}$ is given as:

$$\boldsymbol{F} = \frac{\partial \bar{\boldsymbol{u}}}{\partial \bar{\boldsymbol{X}}} + \boldsymbol{I} = \begin{pmatrix} \frac{\partial u}{\partial x} - z\frac{\partial^2 w}{\partial x^2} & \frac{\partial w}{\partial x} \\ \frac{\partial w}{\partial x} & 0 \end{pmatrix} + \boldsymbol{I}. \tag{7.42}$$

The strain is expressed with the quadratic Green-Lagrange strain $\boldsymbol{E}$. However, since the initial configuration of the beam is aligned with the $x$-axis, only the $xx$-component of the Green-Lagrange strain tensor is of interest yielding

$$E_{xx} = \frac{\partial u}{\partial x} - z\frac{\partial^2 w}{\partial x^2} + \frac{1}{2}\left(\frac{\partial u}{\partial x} - z\frac{\partial^2 w}{\partial x^2}\right)^2 + \frac{1}{2}\left(\frac{\partial w}{\partial x}\right)^2. \tag{7.43}$$

The von Karman assumption (cf. [FT01; C+12]) claims that the axial strains and the curvature are small compared to the bending rotation. This condition is stated as

$$\frac{1}{2}\left(\frac{\partial u}{\partial x} - z\frac{\partial^2 w}{\partial x^2}\right)^2 = 0 \tag{7.44}$$

resulting in the quadratic von Karman strain expression

$$E_{x,\text{Karman}} = \frac{\partial u}{\partial x} - z\frac{\partial^2 w}{\partial x^2} + \frac{1}{2}\left(\frac{\partial w}{\partial x}\right)^2. \tag{7.45}$$

When this strain is used with a linear constitutive law and the finite element procedure is applied, one obtains a polynomial system of equations for the Euler-Bernoulli Beam theory applied with von Karman strain assumption. The dofs can be partitioned in membrane dofs $u$ acting in $x$-direction and bending dofs $w$ acting in $z$-direction resulting in two coupled equations, which are given in index notation following Einstein's summation convention over repeated indices:

$$M_{m,ij}\ddot{u}_j + K^{(1)}_{m,ij}u_j + K^{(2)}_{mb,ijk}w_j w_k = g_{m,i} \tag{7.46}$$

$$M_{b,ij}\ddot{w}_j + K^{(1)}_{b,ij}w_j + K^{(2)}_{bm,ijk}w_j u_k + K^{(3)}_{b,ijkl}w_j w_k w_l = g_{b,i} \tag{7.47}$$

The subscripts $b$ and $m$ stand for bending and membrane respectively. The equations (7.46) and (7.47) are the coupled equations of motion for a von Karman beam for the membrane and bending dofs, respectively. The first equation shows, that the membrane dynamics is linear but coupled nonlinearly with the bending displacements, which trigger the membrane forces quadratically. The bending restoring forces are linear and cubic with respect to the bending displacements and coupled with the membrane displacements in a bilinear fashion. The equations (7.46) and (7.47) can be rearranged to highlight the structure of the coupling as

$$\begin{pmatrix} \boldsymbol{M}_m & \\ & \boldsymbol{M}_b \end{pmatrix}\begin{pmatrix} \ddot{\boldsymbol{u}} \\ \ddot{\boldsymbol{w}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K}^{(1)}_m & \\ & \boldsymbol{K}^{(1)}_b \end{pmatrix}\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K}^{(2)}_{mb}\boldsymbol{w}\boldsymbol{w} \\ \boldsymbol{K}^{(2)}_{bm}\boldsymbol{w}\boldsymbol{u} + \boldsymbol{K}^{(3)}_b\boldsymbol{w}\boldsymbol{w}\boldsymbol{w} \end{pmatrix} = \begin{pmatrix} \boldsymbol{g}_m \\ \boldsymbol{g}_b \end{pmatrix}. \tag{7.48}$$

### 7.3.2 Nonlinear Static Condensation

Consider a more general nonlinear system with separated nonlinear terms and neglected damping for the sake of simplicity

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} + \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{g}(t) \tag{7.49}$$

and assume that the nonlinearity is such, that the equations can be written in partitioned form as

$$\begin{pmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} \end{pmatrix}\begin{pmatrix} \ddot{\boldsymbol{u}}_1 \\ \ddot{\boldsymbol{u}}_2 \end{pmatrix} + \begin{pmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} \end{pmatrix}\begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{pmatrix} + \begin{pmatrix} \boldsymbol{f}_1(\boldsymbol{u}_1, \boldsymbol{u}_2) \\ \boldsymbol{f}_2(\boldsymbol{u}_1) \end{pmatrix} = \begin{pmatrix} \boldsymbol{g}_1(t) \\ \boldsymbol{g}_2(t) \end{pmatrix}. \tag{7.50}$$

Note, that in (7.49) and (7.50) a more general case is investigated which translates to (7.48) with $\boldsymbol{u}_1 = \boldsymbol{w}$ and $\boldsymbol{u}_2 = \boldsymbol{u}$. In the partitioned equation (7.50), the nonlinearity is coupled only in one way, so that the nonlinear forces $\boldsymbol{f}_2$ are only dependent on $\boldsymbol{u}_1$ and not $\boldsymbol{u}_2$. If the inertia forces acting on $\boldsymbol{u}_2$ are set to zero:

$$\boldsymbol{M}_{21}\ddot{\boldsymbol{u}}_1 + \boldsymbol{M}_{22}\ddot{\boldsymbol{u}}_2 = \boldsymbol{0}, \tag{7.51}$$

the second line of (7.50) is an algebraic equation. It yields for the excitation forces $\boldsymbol{g}_2(t)$ set to zero:

$$\boldsymbol{K}_{21}\boldsymbol{u}_1 + \boldsymbol{K}_{22}\boldsymbol{u}_2 + \boldsymbol{f}_2(\boldsymbol{u}_1) = \boldsymbol{0}. \tag{7.52}$$

This algebraic equation can be solved for $\boldsymbol{u}_2$ resulting in

$$\boldsymbol{u}_2 = -\boldsymbol{K}_{22}^{-1}(\boldsymbol{K}_{21}\boldsymbol{u}_1 + \boldsymbol{f}_2(\boldsymbol{u}_1)). \tag{7.53}$$

If this equation (7.53) is inserted into the first line of (7.50), one obtains the static condensed equations of motion, where $\boldsymbol{u}_2$ is eliminated under the condition (7.51).

Exactly the same result is obtained, if the nonlinear transformation

$$\begin{pmatrix}\boldsymbol{u}_1\\ \boldsymbol{u}_2\end{pmatrix} = \begin{pmatrix}\boldsymbol{u}_1\\ -\boldsymbol{K}_{22}^{-1}(\boldsymbol{K}_{21}\boldsymbol{u}_1 + \boldsymbol{f}_2(\boldsymbol{u}_1))\end{pmatrix} = \boldsymbol{\Gamma}(\boldsymbol{u}_1), \tag{7.54}$$

in which $\boldsymbol{u}_1$ is enslaved in terms of $\boldsymbol{u}_1$, is applied to the linear and nonlinear restoring forces in (7.50).

However, the mapping (7.54) can be applied not only to the restoring forces, as in the nonlinear static condensation approach above, but also to the full system as in Section 7.1. Then, the acceleration forces in the $\boldsymbol{u}_2$ direction as stated in (7.51) are not forced to be zero, since this is only the constraint for the kinematic mapping but not for the resulting dynamic equation.

The procedure of nonlinear static condensation and full nonlinear mapping is subsequently applied to the von Karman beam, which exhibits the decoupling structure as given in (7.50).

### 7.3.3 Application to the von Karman Beam

Now, the theory developed in the previous section is applied to the von Karman beam characterized by the partitioned system (7.48) with $\boldsymbol{u}_1 = \boldsymbol{w}$ and $\boldsymbol{u}_2 = \boldsymbol{u}$. The assumption of the static condensation approach is, that the acceleration forces in the in-plane direction are zero:

$$M_{m,ij}\ddot{u}_j = 0. \tag{7.55}$$

Then the second line in (7.48) becomes an algebraic equation, which can be solved for the membrane displacements $u$ with the external forcing $g_{m,i}$ set to zero:

$$u_j = -\left(K^{(1)}_{m,ij}\right)^{-1} K^{(2)}_{mb,ikl} w_k w_l \tag{7.56}$$

This equation can be inserted into (7.47), yielding the equations of motion for the membrane dofs with the bending dofs condensed out:

$$M_{b,ij}\ddot{w}_j + K^{(1)}_{b,ij} w_j - K^{(2)}_{bm,ijk} w_j \left(K^{(1)}_{m,lk}\right)^{-1} K^{(2)}_{mb,lno} w_n w_o + K^{(3)}_{b,ijkl} w_j w_k w_l = g_{b,i}. \tag{7.57}$$

It can be recast to the condensed equation with all cubic terms gathered in $\widetilde{K}^{(3)}_{b,ijkl}$:

$$M_{b,ij}\ddot{w}_j + K^{(1)}_{b,ij} w_j + \widetilde{K}^{(3)}_{b,ijkl} w_j w_k w_l = g_{b,i} \tag{7.58}$$

In this equation, the nonlinear beam is modeled solely with bending displacements $w$, whereas the membrane displacements $u$ are forced to follow the bending displacements in a static fashion. This equation is based, however, on the negligence of the membrane dynamics by setting the inertia forces in $x$-direction to zero (7.55).

As in the previous section, the static condensation approach can also be expressed in terms of a nonlinear basis. The membrane dofs are then enslaved in terms of the bending dofs by the following nonlinear mapping applied to (7.48):

$$\begin{pmatrix} u \\ w \end{pmatrix} = \Gamma(w) = \begin{pmatrix} -\left(K_m^{(1)}\right)^{-1} K_{mb}^{(2)} ww \\ w \end{pmatrix} = \begin{pmatrix} \Theta_{mb} ww \\ w \end{pmatrix} \tag{7.59}$$

with the bending dofs $w$ being the generalized quadratic coordinates $z$. If the mapping (7.59) is applied only on the internal forces in (7.48) and the acceleration forces in membrane direction are set to zero, i.e. $M_m \ddot{u} = 0$, the resulting equations are exactly the condensed equations of motion (7.57). The quadratic manifold is formed by the symmetric second oder tensor $\Theta_{mb} = -(K_m^{(1)})^{-1} K_{mb}^{(2)}$. Note that in the static condensation approach, the quadratic mapping is only applied on the internal restoring forces and not on the acceleration terms. Next, the force compensation approach is applied to the von Karman beam and the similarities are pointed out.

### 7.3.4 Force Compensation Approach

Up to now, only the dofs of the beam are partitioned in bending and membrane dofs. If a QM reduction is applied using the force compensation approach, first the linear basis is defined. Assuming that the linear basis $V$ is built using modes which are solely bending modes, like the first vibration modes or Krylov subspace modes due to pure bending forces, the linear basis $V$ can be partitioned into

$$V = \begin{pmatrix} V_m \\ V_b \end{pmatrix} \tag{7.60}$$

with $V_m$ describing the membrane components and $V_b$ the bending components of the linear basis $V$. Since the basis consists only of bending modes, the membrane components are zero, leading to

$$V = \begin{pmatrix} 0 \\ V_b \end{pmatrix}. \tag{7.61}$$

When the force compensation approach (cf. Subsection 7.2.3 is applied, the third order tensor $\Theta$ is defined in accordance with (7.16) and with $\hat{u} = (w^T, u^T)^T$ as

$$\theta_{ij} = K^{-1} \left.\frac{\partial^2 f}{\partial \hat{u}^2}\right|_{\hat{u}=0} v_i v_j. \tag{7.62}$$

With the partitioning in (7.48), one finds

$$\Theta = \begin{pmatrix} \left(K_m^{(1)}\right)^{-1} & 0 \\ 0 & \left(K_b^{(1)}\right)^{-1} \end{pmatrix} \begin{pmatrix} K_{mb}^{(2)} V_b V_b \\ K_{bm}^{(2)} V_b V_m \end{pmatrix} \tag{7.63}$$

with the second order derivative $\partial^2 f / \partial u^2$ of the internal forces forming the quadratic stiffness terms $K_{mb}^{(2)}$ and $K_{bm}^{(2)}$. Since $V_m = 0$, (7.63) simplifies to

$$\Theta = \begin{pmatrix} \left(K_m^{(1)}\right)^{-1} K_{mb}^{(2)} V_b V_b \\ 0 \end{pmatrix} \tag{7.64}$$

and the full quadratic mapping $\Gamma$ is then given as

$$\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{pmatrix} = \begin{pmatrix} \left[ \left( \boldsymbol{K}_m^{(1)} \right)^{-1} \boldsymbol{K}_{mb}^{(2)} \boldsymbol{V}_b \boldsymbol{V}_b \right] \boldsymbol{z}\boldsymbol{z} \\ \boldsymbol{V}_b \boldsymbol{z} \end{pmatrix} \tag{7.65}$$

$$= \underbrace{\begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{V}_b \end{pmatrix}}_{\boldsymbol{V}} \boldsymbol{z} + \underbrace{\begin{pmatrix} \left( \boldsymbol{K}_m^{(1)} \right)^{-1} \boldsymbol{K}_{mb}^{(2)} \boldsymbol{V}_b \boldsymbol{V}_b \\ \boldsymbol{0} \end{pmatrix}}_{\boldsymbol{\Theta}} \boldsymbol{z}\boldsymbol{z}. \tag{7.66}$$

This quadratic mapping is the result of the force compensation approach applied to a von Karman beam. This mapping shows clearly, that the membrane displacements $\boldsymbol{u}$ are enslaved quadratically to the bending displacements $\boldsymbol{w}$. Note that the mapping (7.65) is equivalent to (7.59) with $\boldsymbol{w} = \boldsymbol{V}\boldsymbol{z}$ substituted. This means that the force compensation approach yields exactly the same basis as the static condensation approach for the von Karman beam, given that the linear basis $\boldsymbol{V}$ is composed of bending displacements only.

### 7.3.5 Relation between QM Approach and Static Condensation

Using the mapping (7.59) solely on the internal forces and not the acceleration terms results in the static condensation (7.57). However, when using the quadratic mapping (7.59) in the nonlinear projection framework (7.4), the resulting equations of motion are written as:

$$\begin{aligned} &[(\boldsymbol{\Theta}_{mb}\boldsymbol{w})^T \boldsymbol{M}_m \boldsymbol{\Theta}_{mb}\boldsymbol{w} + \boldsymbol{M}_b]\ddot{\boldsymbol{w}} + (\boldsymbol{\Theta}_{mb}\boldsymbol{w})^T \boldsymbol{M}_m \boldsymbol{\Theta}_{mb}\dot{\boldsymbol{w}}\dot{\boldsymbol{w}} \\ &+ \boldsymbol{K}_b^{(1)}\boldsymbol{w} + \boldsymbol{K}_{bm}^{(2)}\boldsymbol{w}\left( \boldsymbol{K}_m^{(1)} \right)^{-1} \boldsymbol{K}_{mb}^{(2)}\boldsymbol{w}\boldsymbol{w} + \boldsymbol{K}_b^{(3)}\boldsymbol{w}\boldsymbol{w}\boldsymbol{w} = \boldsymbol{g}_b + (\boldsymbol{\Theta}_{mb}\boldsymbol{w})^T \boldsymbol{g}_m \end{aligned} \tag{7.67}$$

Or recast and written in index notation:

$$\begin{aligned} &\left[ \widetilde{M}_{m,ijkl}^{(3)} w_k w_l + M_{b,ij} \right] \ddot{w}_j + \widetilde{M}_{m,ijkl}^{(3)} w_j \dot{w}_k \dot{w}_l \\ &+ K_{b,ij}^{(1)} w_j + \widetilde{K}_{b,ijkl}^{(3)} w_j w_k w_l = g_{b,i} + \Theta_{jik} w_k g_{m,j} \end{aligned} \tag{7.68}$$

with the membrane mass tensor $\widetilde{M}_{m,ijkl}^{(3)} = M_{m,no}\Theta_{nij}\Theta_{okl}$ and $\Theta_{ijk} = (K_{m,li}^{(1)})^{-1} K_{mb,ljk}^{(2)}$.

When comparing the nonlinear projected equations of motion (7.68) with the equations of motion with static condensation (7.57), the difference between the nonlinear projection and the static condensation lies in the additional acceleration terms associated with $\widetilde{M}_m$, which are only in place in the nonlinear projection. They represent the inertia effect in the membrane direction. They are only nonzero if the bending displacement $\boldsymbol{w}$ is nonzero leading to a tangential motion of the nonlinear beam which carries a component in membrane direction. This additional inertia is often negligible making the static condensation a good choice in general. However, in special cases, they can play a non-negligible role as illustrated in [RR14b].

Compared to static condensation, the QM approach carries the advantage that it is applicable to systems where the decomposition in master and slave dofs is not possible or convenient. Hence, the nonlinear mapping using a quadratic manifold opens the field to other systems not comprising these properties.

## 7.4 Applications

To show the feasibility of the QM approach as well as its limits, it is applied to several examples. The QM approach is an extension of the reduced basis approach introduced in Chapter 3, the goal is the reduction of the number of dofs. Since, as will be discussed in Part II of this thesis, the bottleneck of reduced models is very often the evaluation of

the nonlinear, internal forces, decent speedup rates require the use of hyper-reduction techniques. Hence, as the focus of this section is the kinematic fit of the QM assumption for different applications, the computational times are not reported here.

### 7.4.1 Approach to Investigation of the Proposed Methods

The motivation of the QM reduction framework is the reduction of the number of dofs when using (S)MDs for enriching the lin basis. The key idea is the quadratic enslavement of the amplitudes of the (S)MDs to the master coordinates $\boldsymbol{z}$ representing the amplitudes of the linear basis $\boldsymbol{V}$. Consequently, the QM approach is compared to the simulation-free approaches proposed in Chapter 6, where the (S)MDs are added to the basis $\boldsymbol{V}$ by not enslaving them but giving them additional dofs.

However, when building a basis with modes and the corresponding (S)MDs, the linear basis has to be treated with an orthogonalization technique or a deflation technique as proposed in Subsection 6.2.3 to keep the condition number of the linear basis in an acceptable range. A similar issue can also occur in the QM approach, where a stabilization technique as proposed in Subsection 7.2.4 can be used to deteriorate the QM such, that the quadratic part is orthogonalized with respect to the linear part and hence the system remains stable.

As a result, basically three projection techniques are available which are investigated here. First, the QM approach as defined in Section 7.2, where the quadratic part stems directly from the MDs or from the SDs, which is equivalent to the force compensation approach proposed in Subsection 7.2.3. Second, the QM approach, where the quadratic part is orthogonalized with respect to the linear part in order to stabilize the reduced system. And third, the approach proposed in the previous chapter, where the (S)MDs are given independent dofs.

In addition to the projection techniques, different basis techniques are investigated. Since they are mostly independent of the projection techniques, they can be combined with all three projection techniques. An overview of the combinations are given in Table 7.1, where the different choices of bases are combined with different projection techniques and each combination is given a tag to identify the reduction method. Solely the MD reduction technique is not combined with the QM-orthogonalized method, since this combination does have stability issues in the numerical experiments. The basis techniques are

- MD: Modal derivatives. The linear part of the basis $\boldsymbol{V}$ is composed of vibration modes $\boldsymbol{\phi}_i$ and complemented with modal derivatives $\partial\boldsymbol{\phi}_i/\partial\eta_j$ as described in Subsection 6.2.1.

- SMD: Static modal derivatives. The linear part of the basis $\boldsymbol{V}$ is composed of vibration modes $\boldsymbol{\phi}_i$ and complemented with static modal derivatives $\partial\boldsymbol{\phi}_i/\partial\eta_j|^s$ which are computed as given in Subsection 6.2.2.

- KrySD: Static Derivatives using Krylov subspace vectors. The linear part of the basis $\boldsymbol{V}$ is composed of Krylov-subspace vectors as given in Section 4.4. They are complemented with the SDs corresponding to the basis.

- KrySD-SMD: Static Derivatives using both, a combination of vibration modes and Krylov subspace vectors as linear part $\boldsymbol{V}$ of the basis. They are complemented with the SDs corresponding to the basis.

In the following, different reduction techniques as given in Table 7.1 are investigated on several examples. Some examples are discretized with beam elements, where the displacement vector also contains rotational dofs additionally to the displacement dofs.

**Table 7.1:** Combination of projection type (columns) and basis type (rows) for the numerical experiments. The declarations are used to indicate the experiments.

| | **QM** | **QM-orthogonalized** | **Linear Basis (LB)** |
|---|---|---|---|
| **MD** | QM-MD | — | LB-MD |
| **SMD** | QM-SMD | QM-SMD-orth | LB-SMD |
| **KrySD** | QM-KrySD | QM-KrySD-orth | LB-KrySD |
| **KrySD-SMD** | QM-KrySD-SMD | QM-KrySD-SMD-orth | LB-KrySD-SMD |

Since the magnitude of the rotations is a matter of scaling, the relative error is expressed with $RE_M$ instead of $RE$ as defined in (3.9), since the mass norm compensates for the different kinematic quantities:

$$RE_M = \frac{\sqrt{\sum_{t\in\mathcal{T}} \Delta\boldsymbol{u}(t)^T \boldsymbol{M} \Delta\boldsymbol{u}(t)}}{\sqrt{\sum_{t\in\mathcal{T}} \boldsymbol{u}_{\text{ref}}(t)^T \boldsymbol{M} \boldsymbol{u}_{\text{ref}}(t)}} \cdot 100\% \quad \text{with} \quad \Delta\boldsymbol{u}(t) = \boldsymbol{u}(t) - \boldsymbol{u}_{\text{ref}}(t) \tag{7.69}$$

The mass matrix $\boldsymbol{M}$ is of the unreduced system, $\boldsymbol{u}_{\text{ref}}$ is the reference displacement field which is obtained using a full, unreduced simulation and $\boldsymbol{u} = \boldsymbol{\Gamma}(\boldsymbol{z})$ is the restored full displacement field of the reduced system, which is compared to the reference solution.

In all numerical experiments, the time integration is performed using the Newmark scheme as outlined in Subsection 7.2.5. In the Newton-Raphson loop for solving the nonlinear system of equations for the $j+1$-th time step, the number of iterations is limited to 30. If no convergence is gained within this number, the time integration is aborted and declared as failed.

### 7.4.2 Clamped-Clamped Beam

The first example is the clamped-clamped beam as depicted in Figure 7.3. It is loaded with a constant, time varying pressure on a part of the top side. The investigation is conducted with two models: One model is discretized using von Karman beam elements, whereas the other is discretized using solid elements with quadratic shape functions. The parameters of both models are listed in Table 7.2.

First, the von Karman beam is investigated. A time integration is run for $\boldsymbol{V}$ composed of five modes for the various reduction techniques outlined in Subsection 7.4.1. The displacement results of the observer point depicted in Figure 7.3 are given in the two top plots of Figure 7.5. The displacements of the reduced models show very good agreement with the full, unreduced reference solution, independently of the reduction method employed. Since the QM approach fits very well into the polynomial structure of the von Karman assumption, this experiment is a verification of the theory employed in Section 7.3.

Second, the clamped-clamped beam discretized with triangular elements as depicted in Figure 7.3 is investigated. The reduction methods of Table 7.1 are applied and time integrations of the reduced and the full models are conducted. The displacements of the observed node depicted in Figure 7.5 show also an excellent match of the reduced models with the reference solution for five modes. To compare the von Karman discretization with the solid element discretization, the displacement of the observed node for both full models is depicted in Figure 7.4. The trajectories of both full models show clearly an excellent agreement in the $y$-direction indicating, that the von Karman assumption is valid for the given clamped-clamped beam example.

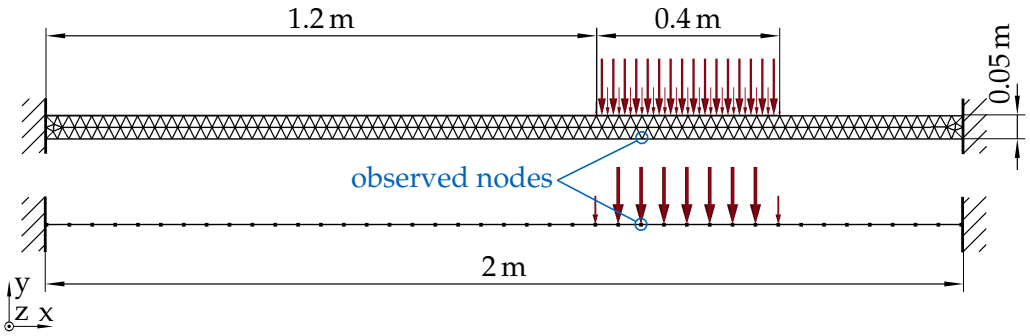


**Figure 7.3:** Beam model with external forcing for solid beam (top) and von Karman beam (bottom).

**Table 7.2:** Settings for the numerical experiments for the clamped-clamped beam. The properties of von Karman beam and solid discretized beam are equal.

| **Property** | **solid beam** | **von Karman beam** |
|---|---|---|
| no of elements | 326 | 40 |
| no of dofs | 1614 | 117 |
| element type | Tri6 | Beam Elements (Karman) |
| Stress configuration | plane stress | plane stress |
| Young's modulus $E$ | 210 GPa | 210 GPa |
| Poisson ratio $\nu$ | 0.3 | 0.3 |
| time step size $\Delta t$ | $1 \cdot 10^{-4}$ s | $1 \cdot 10^{-4}$ s |
| $t_{end}$ | 0.2 s | 0.2 s |
| $\alpha$ | 0.1 | 0.1 |
| excitation | $\sin(72 \cdot 2\pi t) + \sin(100 \cdot 2\pi t)$ | $\sin(72 \cdot 2\pi t) + \sin(100 \cdot 2\pi t)$ |
| magnitude | $2 \cdot 10^6$ N/m | $2 \cdot 10^6$ N/m |

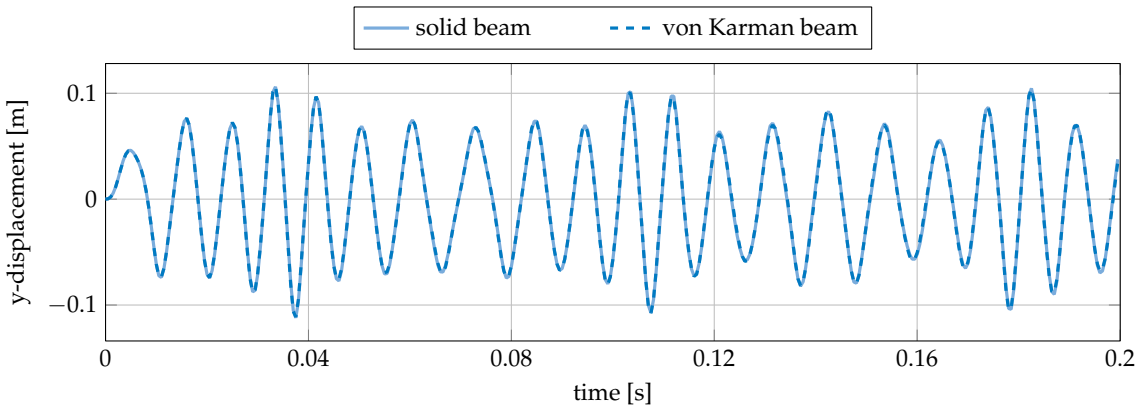


**Figure 7.4:** Displacements at the observed node for the clamped-clamped beam discretized with solid elements and with von Karman beam elements.

Next the number of modes is varied and the relative error $RE_M$ is computed for the different reduction techniques given in Table 7.1. The error measures for both models, the solid and von Karman discretized beam are given in Figure 7.6. Only successful numerical experiments are shown in the graph, whereas no markers are drawn when the time integration failed. The following observations can be drawn from these results:

- In general, the quadratic enslavement of the dofs yields good results. This is in accordance with the von Karman beam theory claiming, that the in-plane dofs are related quadratically to the transverse dofs.
- The reduction error for the von Karman model test is in general lower than for solid beam. Since the solid beam expresses the motion of every element nodes, the motion can be considered as more complex making it difficult to approximate the motion in order to obtain very low $RE_M$ values.
- Instability can be observed for the force compensation approach, i.e., using SDs in the QM framework, for a higher number of modes. This is due to the fact, that in-plane modes (modes number 7, 11, 14, 18) destroy the purely transverse linear basis, which is then augmented with the SDs being completely in-plane motions. Since both transverse and in-plane modes are in the linear basis $\boldsymbol{V}$, the quadratic part $\boldsymbol{\Theta}$ consists also of both in-plane and transverse modes causing potentially ill-conditioned tangential projectors $\boldsymbol{P}_\Gamma$.
- The LB approach as introduced in Chapter 6 yields the best results, however at the price of high number of dofs. As in Section 6.4, the accuracy of MD and SMD is better than for KrylovSD bases.

To summarize, the QM approach works for the simple case of a straight beam which is clamped on both sides. Even though the beam does not have to be modeled with von Karman beam elements, the quadratic enslavement makes sense. However, the in-plane modes can destroy the linear independence of the tangential subspace $\boldsymbol{P}_\Gamma$ making an orthogonalization technique necessary to stabilize the problem. Nonetheless, the stabilization destroys the accuracy of the QM mapping leading to a decreasing accuracy of the reduced model even though the number of dofs increases. The MD approach by contrast is always stable.

The success of the QM approach for the simple clamped-clamped beam is strictly supported by the von Karman beam theory, which is valid for this setup. However, in the following, more complicated examples are investigated.

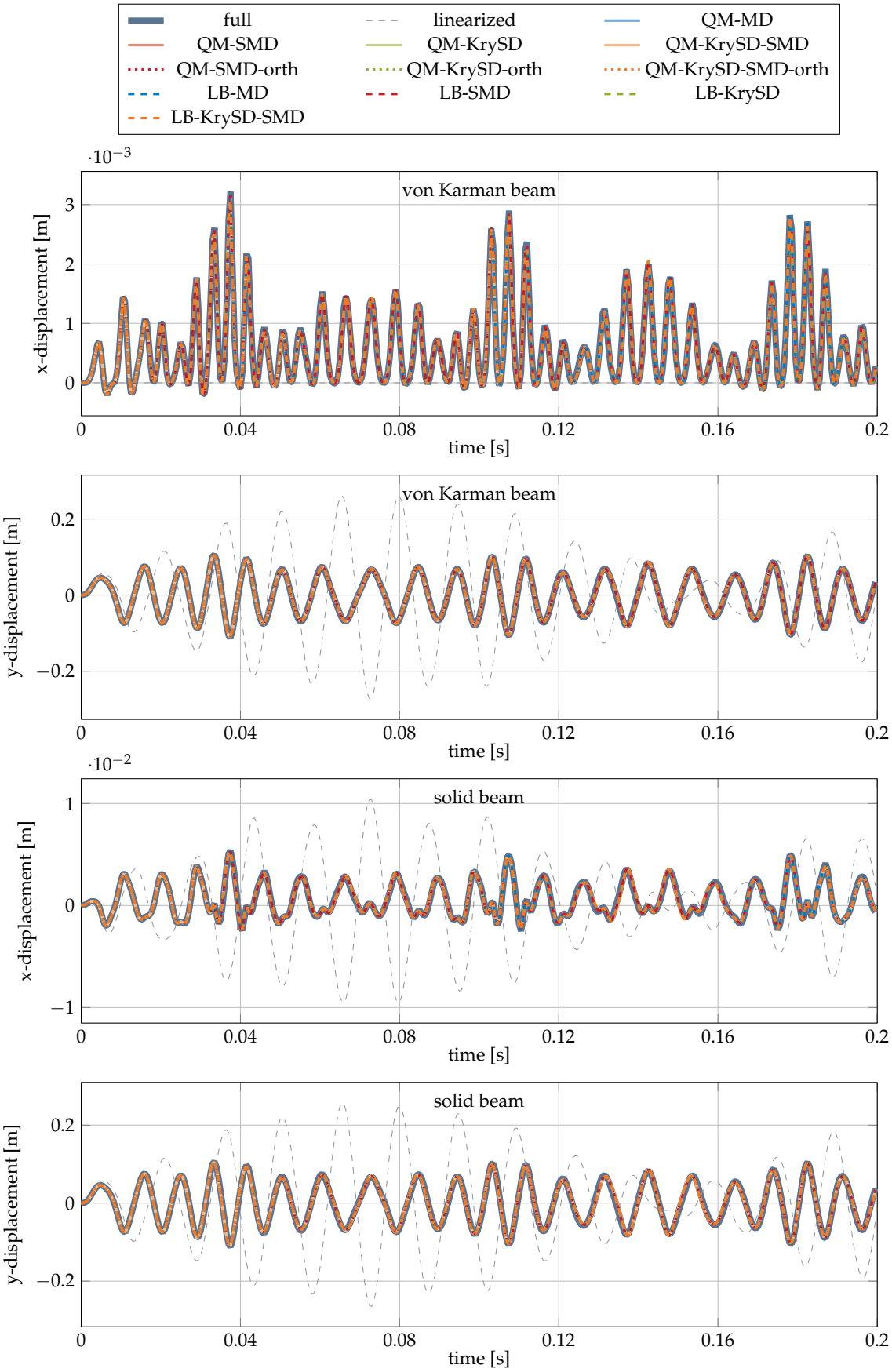


**Figure 7.5:** Displacements at the observed node for the clamped-clamped beam discretized with von Karman beam elements (top plots) and continuum elements (bottom plots) for different reduction techniques using 5 modes.

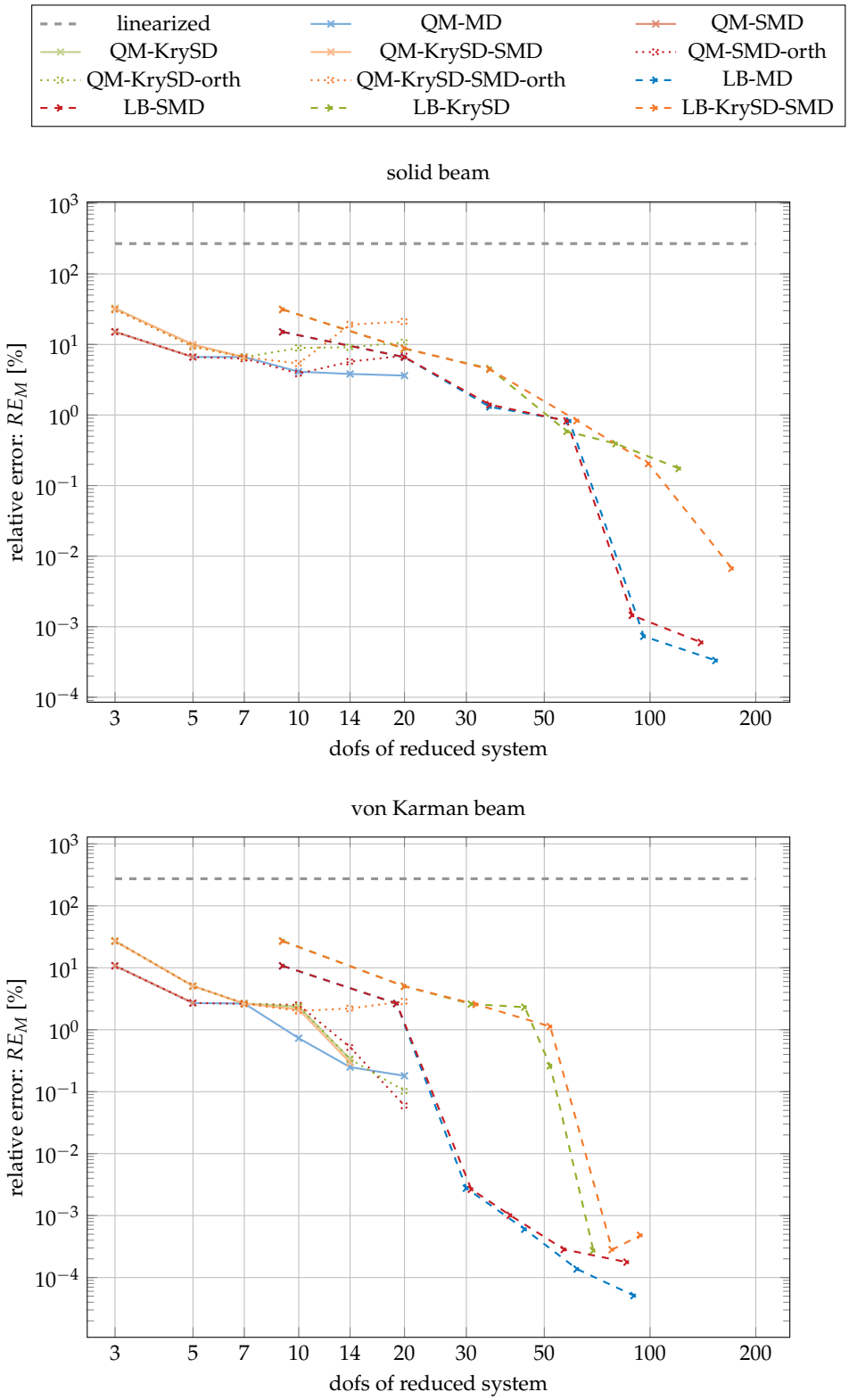


**Figure 7.6:** Relative error of the clamped-clamped beam discretized with solid elements (top) von Karman beam elements (bottom) for different reduction rates.

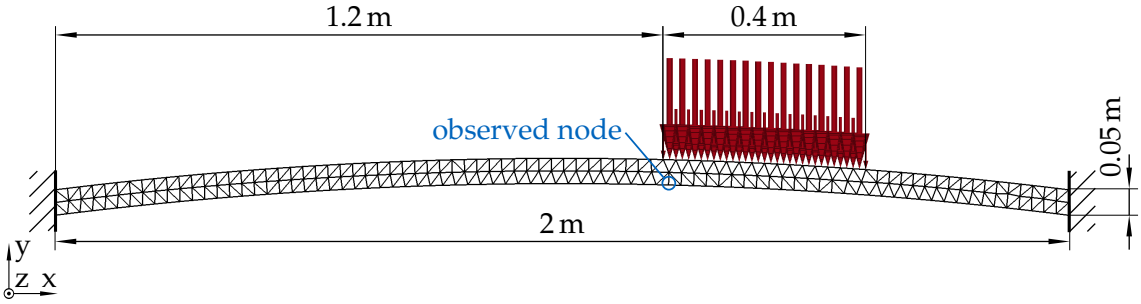


**Figure 7.7:** Mesh of the arch with external forcing.

### 7.4.3 Arch

Next a slightly more complicated model compared to the clamped-clamped beam from the section before is investigated. It is an arch carrying the same properties as the clamped-clamped beam, however it is curved with a radius of $R = 8\,\mathrm{m}$. The mesh of the arch, which is discretized also with triangular solid elements with quadratic shape functions, is depicted in Figure 7.7. The properties of the model are given in Table 7.3.

Similar to the previous example, a time integration is performed and the displacement results for the different reduction techniques using five modes for the linear part $\boldsymbol{V}$ are investigated. The displacements of the observed node depicted in Figure 7.8 indicate that some QM reduction techniques are successful, while others do not capture the motion or do not gain convergence in the Newton-Raphson loop of the time integration. The $RE_M$ measure overview for different reduction orders is given in Figure 7.9. The first observation is, that all plain SD reduction techniques without orthogonalization fail to converge. On the other hand, the QM-MD approach leads to incorrect results which do not capture the motion of the reference solution. This deviation from the reference solution can be clearly seen in the time series plot in Figure 7.8. However, the SD approach with stabilization through orthogonalization yields good results with few dofs. As in the clamped-clamped example before, the accuracy decreases with a higher number of dofs, since the orthogonalization destroys the quadratic mapping. Hence, the best QM results are obtained with a moderate number of dofs. The greatest accuracy, though, is obtained with the LB approach, similar as in the previous example. Once again, this accuracy comes at the price of a clearly larger reduction base.

### 7.4.4 3D Membrane Structure

Next, a 3D membrane structure depicted in Figure 7.10 is investigated. The model is a solid part which is thicker in the middle and clamped on the outer circular rim which is colored in Figure 7.10. It is loaded at the bottom face with a constantly distributed force acting in $y$-direction. The observed node is located nearly in the middle of the bottom face. Figure 7.11 depicts the displacements in the $y$ direction of this node for different reduction schemes using five modes. Apparently, the QM-SMD and the QM-KrySD-SMD projection techniques are not stable and diverge until the Newton-Raphson loop does not converge within 30 iterations. But also the QM-Kry-SD does not follow the reference solution. On the other hand, the orthogonalized QM approaches, especially the QM-KrySD-orth and the QM-KrySD-SMD-orth represent the motion of the full system fairly well.

The more general investigation of the $RE_M$ is given in Figure 7.12. The QM approaches are mostly unstable and fail to converge for almost all reduction orders. However, the orthogonalization of the quadratic part with respect to the linear part stabilizes the procedure leading to fairly good results. In this application, the Krylov subspace

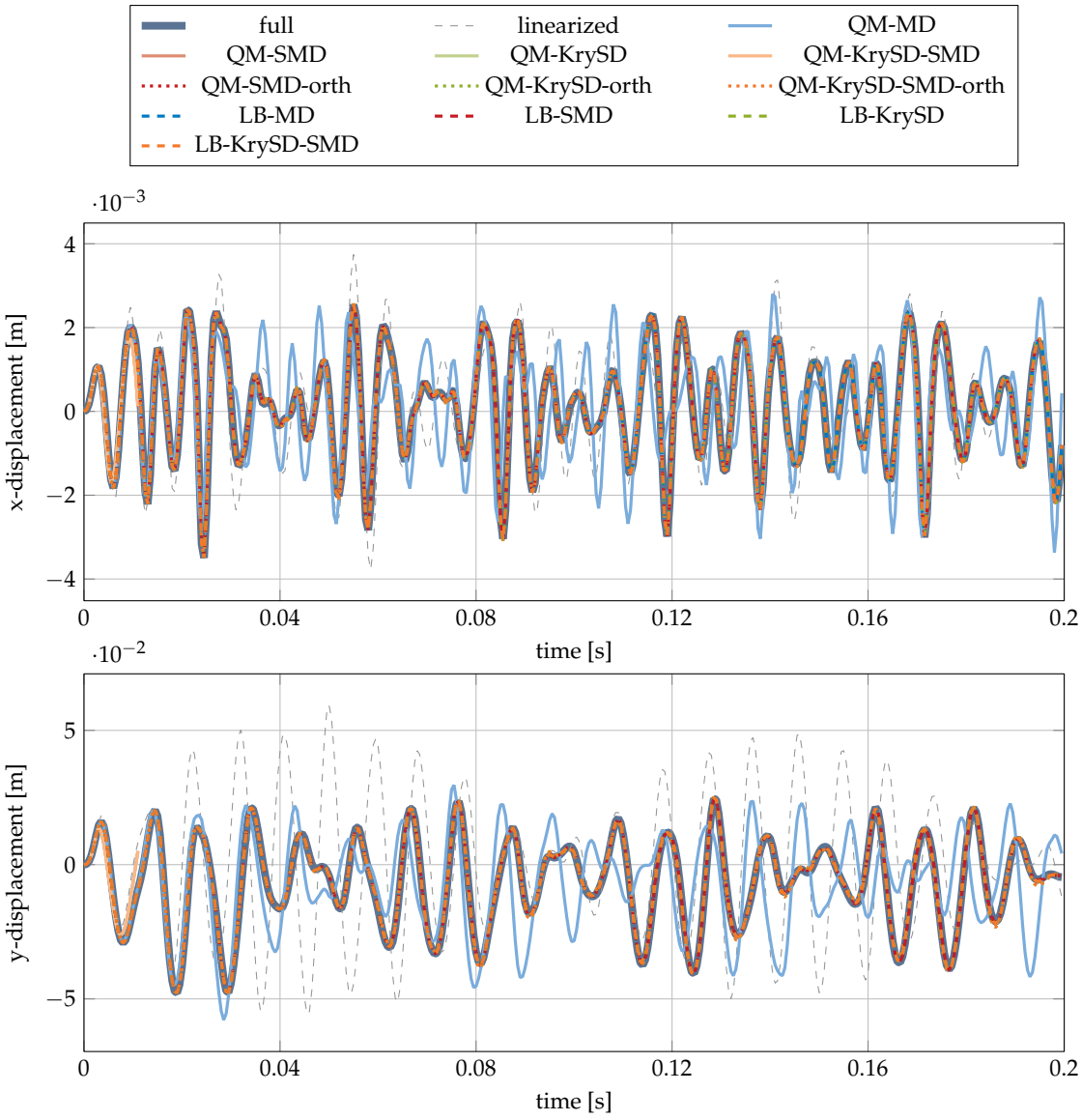


**Figure 7.8:** Displacements at the observed node of the arch example for different reduction techniques using 5 modes.

bases perform better than the modal bases. This is due to the symmetry of the membrane structure, which is also symmetrically loaded. The expected response of the structure is also symmetric, and hence only symmetric displacement fields are good ingredients in the linear part $V$ of the quadratic mapping. Since the Krylov subspace technique accounts for the symmetry in both the structure and the loading, the Krylov vectors are also symmetric yielding the increase of accuracy compared to the modal approaches, where several anti-symmetric modes are present. A second observation from the results given in Figure 7.12 is that the orthogonalization in the QM framework does not lead to an increase of the error with an increasing number of dofs. However, it is to be expected, that the loss of accuracy would appear for higher number of dofs not displayed in Figure 7.12. As in all examples before, the best accuracy is obtained with the LB approach. Again, this comes with the price of large bases.

**Table 7.3:** Settings for the numerical experiments for the arch and 3D membrane structure.

| Property | Arch | 3D Membrane Structure |
|---|---|---|
| no of elements | 325 | 9774 |
| no of dofs | 1616 | 52362 |
| element type | Tri6 | Tet10 |
| Stress configuration | plane stress | — |
| Young's modulus $E$ | 70 GPa | 70 GPa |
| Poisson ratio $\nu$ | 0.3 | 0.3 |
| time step size $\Delta t$ | $1 \cdot 10^{-4}$ s | $4 \cdot 10^{-6}$ s |
| $t_{end}$ | 0.2 s | 0.01 s |
| $\alpha$ | 0.1 | 0.1 |
| excitation | $\sin(115 \cdot 2\pi t) + \sin(150 \cdot 2\pi t)$ | $\sin(500 \cdot 2\pi t) + \sin(1030 \cdot 2\pi t)$ |
| magnitude | $1 \cdot 10^6$ N/m | $1 \cdot 10^7$ N/m$^2$ |

### 7.4.5 Cantilever

The last application for the QM approach is the cantilever beam depicted in Figure 7.13. As in the clamped-clamped beam, the cantilever is once discretized with von Karman beam elements and once with quadratic solid elements. The cantilever is loaded with a force at the tip. First, the von Karman discretization is investigated. For the given load the tip displacements of all QM approaches using five modes perform extremely well, as Figure 7.14 indicates. However, also the linearized system which is described by the dark-green dashed line represents the motion in $y$-direction very well. This indicates that besides the membrane effect responsible for the displacements in $x$-direction, the transverse dynamics can be captured very well with a linearized system.

Secondly, the solid discretized beam is investigated. As Figure 7.15 clearly shows, the QM approach fails to capture the dynamics of the beam for all types of bases, while the LB approach and the linearized system capture the dynamics of the beam pretty well. Obviously the QM reduced system of the solid beam exhibits a stiffening effect as displayed in Figure 6.1, while the von Karman discretized approach as well as the linearized approach shows a good behavior. To investigate this issue further, both cantilever models are loaded statically with a constantly increasing force to examine the static displacement behavior. Since the nonlinearity is only in the internal restoring forces, the nonlinearity can be revealed by this static investigation.

In Figure 7.16, the responses to the static load on the tip are displayed in the left part of the figure. While the displacements of the von Karman beam coincide with the solid discretized beam for small displacements, they deviate for larger displacements. This is not surprising, since the von Karman assumption neglects a term in the Green-Lagrange strain, which is only admissible for small displacements. However, since not only the displacement amplitude differs, but also the trajectory of the tip, which is depicted for both examples in the right part of Figure 7.16, the mapping between in-plane and transverse displacement differs in the two discretizations of the cantilever.

The von Karman beam exhibits a clear quadratic relation between the in-plane and the transverse forces, as explained in Section 7.3. However, if the solid beam does not follow the shape of the trajectory of the von Karman beam, the mapping of the internal transverse forces with the in-plane forces is obviously different and not quadratic.

The first vibration modes of the cantilever are transverse modes. Consequently, as shown in Figure 6.2 and Figure 7.1, the SMDs are in-plane displacement fields. In the QM approach, the in-plane SMDs are forced to follow the transverse vibration modes

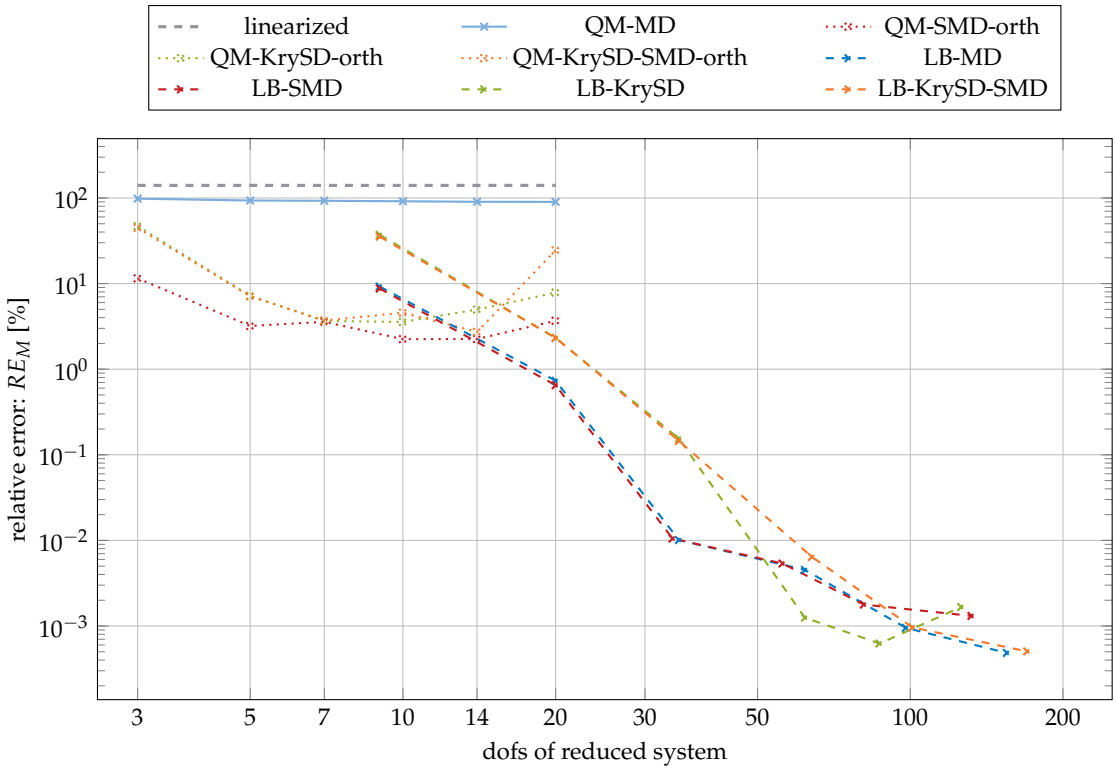


**Figure 7.9:** Relative error of the arch for different reduction rates.

quadratically. However, if the internal forces violate the quadratic relationship which is enforced by the kinematic relationship, a locking effect as illustrated in Section 6.1 arises. Then artificial strains arise due to the mismatch of kinematically admissible displacements and internal forces.

In Figure 7.16, the tip trajectory of the von Karman cantilever and the solid cantilever diverge for a displacement at about 0.5 m. At exactly this displacement value, the solid cantilever starts to lock in the time series depicted in Figure 7.15 leading to the divergence of the reference solution and the QM reduced solution. Consequently, the QM approach fails to represent the motion of the solid cantilever. The von Karman cantilever, on the other hand, has an internal relationship between the in-plane and transverse forces cast into the formulation leading to accurate results for the cantilever example.

For the sake of completeness, the $RE_M$ errors of both cantilever discretizations are given in Figure 7.17. They indicate a clear failure of the QM approach of the solid discretization while the QM reduction for the von Karman discretization works fairly well. However, the orthogonalization destroys the QM mapping behavior for higher number of modes leading to large errors. This is due to the fact that in-plane modes are taken into the linear basis $\boldsymbol{V}$ leading either to instabilities due to bad conditioning of $\boldsymbol{P}_\Gamma$ or to a disruption of the QM relation if the quadratic part is orthogonalized with respect to the modes.

## 7.5 Discussion

The QM approach proposed in this chapter is twofold. On the one hand, it allows for the reduction of the dofs to a minimum number. On the other hand, the approach is not very robust for the applications to arbitrary structures. It is very suitable to structures where the quadratic enslavement of certain dofs is backed by the polynomial structure of the

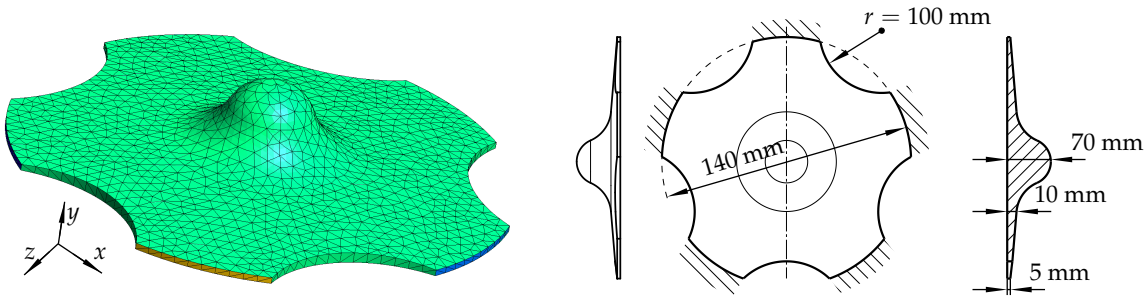


**Figure 7.10:** Mesh and size of the 3D membrane structure.

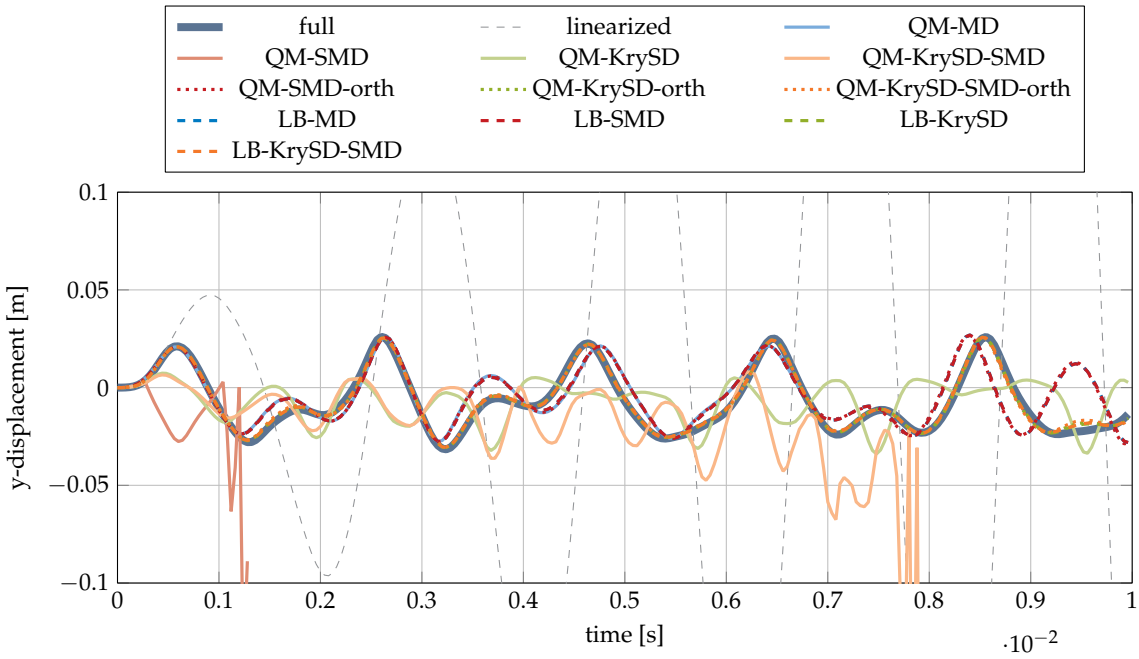


**Figure 7.11:** Displacements in the $y$-direction at a node at the bottom face of the 3D membrane structure for different reduction techniques using 5 modes.

**Table 7.4:** Settings for the numerical experiments for the cantilever.

| **Property** | **Cantilever solid** | **Cantilever von Karman** |
|---|---|---|
| no of elements | 326 | 40 |
| no of dofs | 1624 | 120 |
| element type | Tri6 | Beam Elements (Karman) |
| Stress configuration | plane stress | plane stress |
| Young's modulus $E$ | 70 GPa | 70 GPa |
| Poisson ratio $\nu$ | 0.3 | 0.3 |
| time step size $\Delta t$ | $1 \cdot 10^{-4}$ s | $1 \cdot 10^{-4}$ s |
| $t_{end}$ | 0.2 s | 0.2 s |
| $\alpha$ | 0.1 | 0.1 |
| excitation | $\sin(20 \cdot 2\pi t) + \sin(48 \cdot 2\pi t)$ | $\sin(20 \cdot 2\pi t) + \sin(48 \cdot 2\pi t)$ |
| magnitude | $3 \cdot 10^6$ N/m | $1.5 \cdot 10^5$ N |

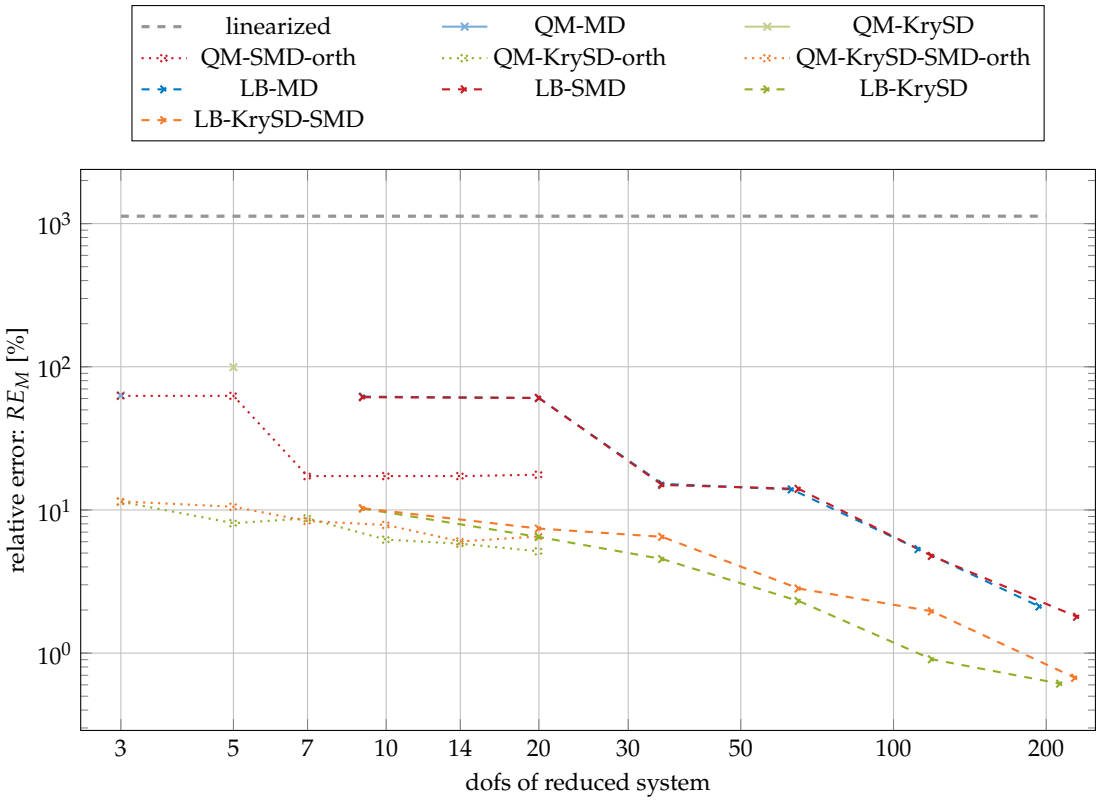


**Figure 7.12:** Relative error of the 3D membrane structure for different reduction rates and techniques.

equations of motion, as it is the case for flat structures like beams and shells exhibiting the von Karman kinematic assumption. Then, the QM approach can be seen as an extension to the static condensation, where the inertia effects are consistently considered by the projection onto a QM.

The range is however limited, when the structures are discretized without kinematic assumptions. Then, the QM approach is reasonable when nonlinearities are triggered while the displacements are moderate. This is the case in applications where the so-called cable or membrane effect is dominating. Then, the triggering nonlinearity is of quadratic nature and is well captured by the QM projection technique.

One further issue is the stability of the manifold projection. In many cases, the tangential basis $\boldsymbol{P}_\Gamma$ of the quadratic manifold is ill-conditioned, leading to no convergence in the Newton-Raphson loop of the time integration. This issue, however, can be solved efficiently by an orthogonalization technique proposed in Subsection 7.2.4, where the quadratic part is orthogonalized with respect to the linear part. This procedure, though, destroys the QM mapping leading to larger error rates, especially for larger linear bases.

In the cases where the QM approach fails, the ingredients of the QM basis commonly suit the nonlinear problem very well. If the lin basis modes and the corresponding (S)MDs or SDs are given independent dofs which are not linked by a quadratic relationship, the reduced models are commonly very accurate, though equipped with many dofs. This idea is already extensively discussed in the simulation free approaches of the previous chapter.

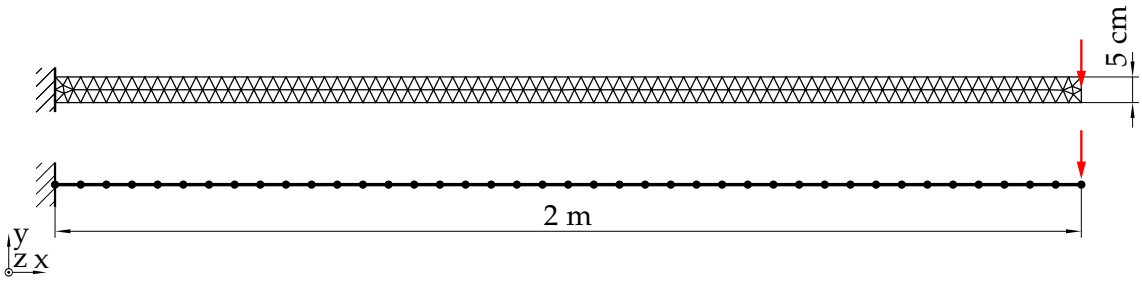


**Figure 7.13:** Cantilever discretized with triangular solid elements (top) and beam elements using von Karman kinematics (bottom). The cantilever is loaded with a tip force in $y$-direction.

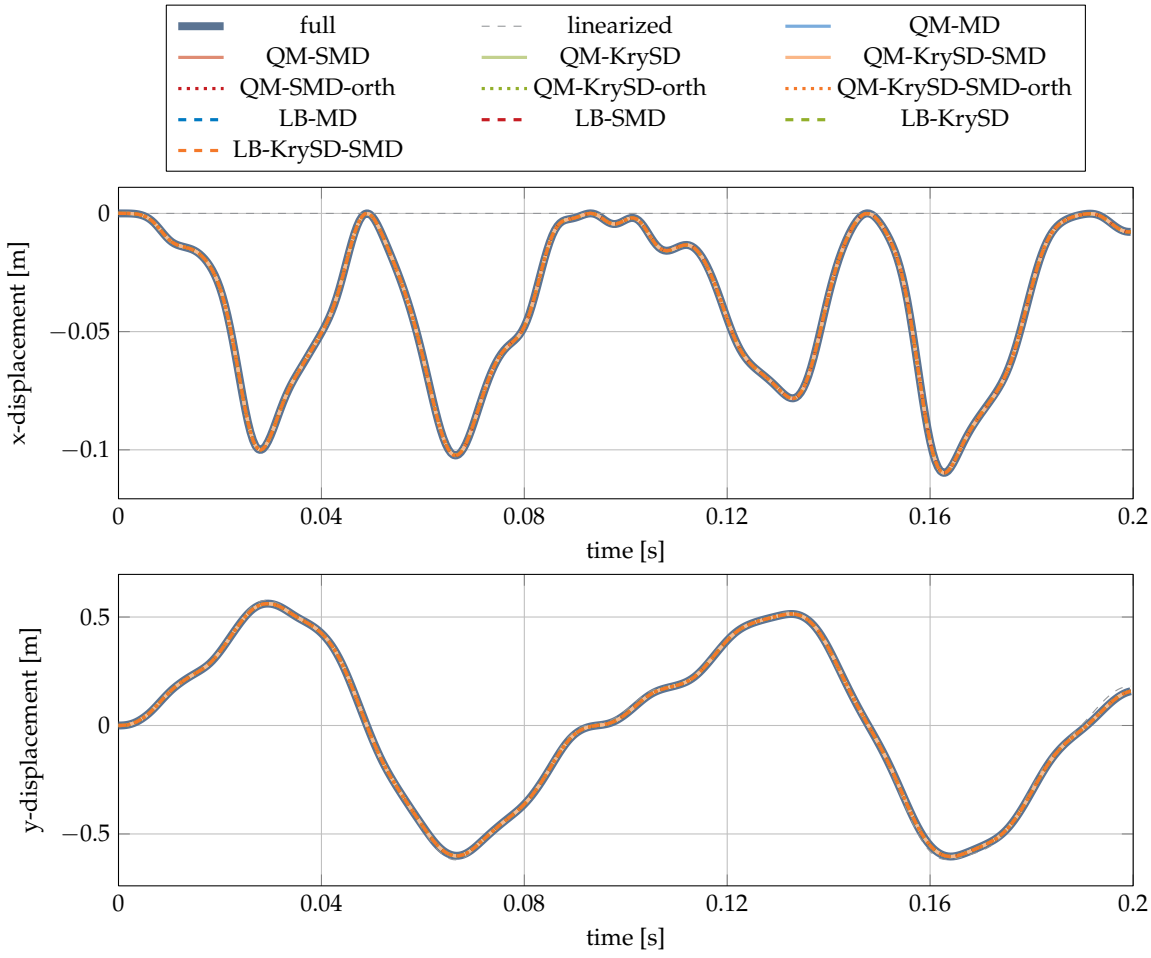


**Figure 7.14:** Displacements at the marked node for the cantilever discretized with von Karman beam elements for different reduction techniques using 5 modes.

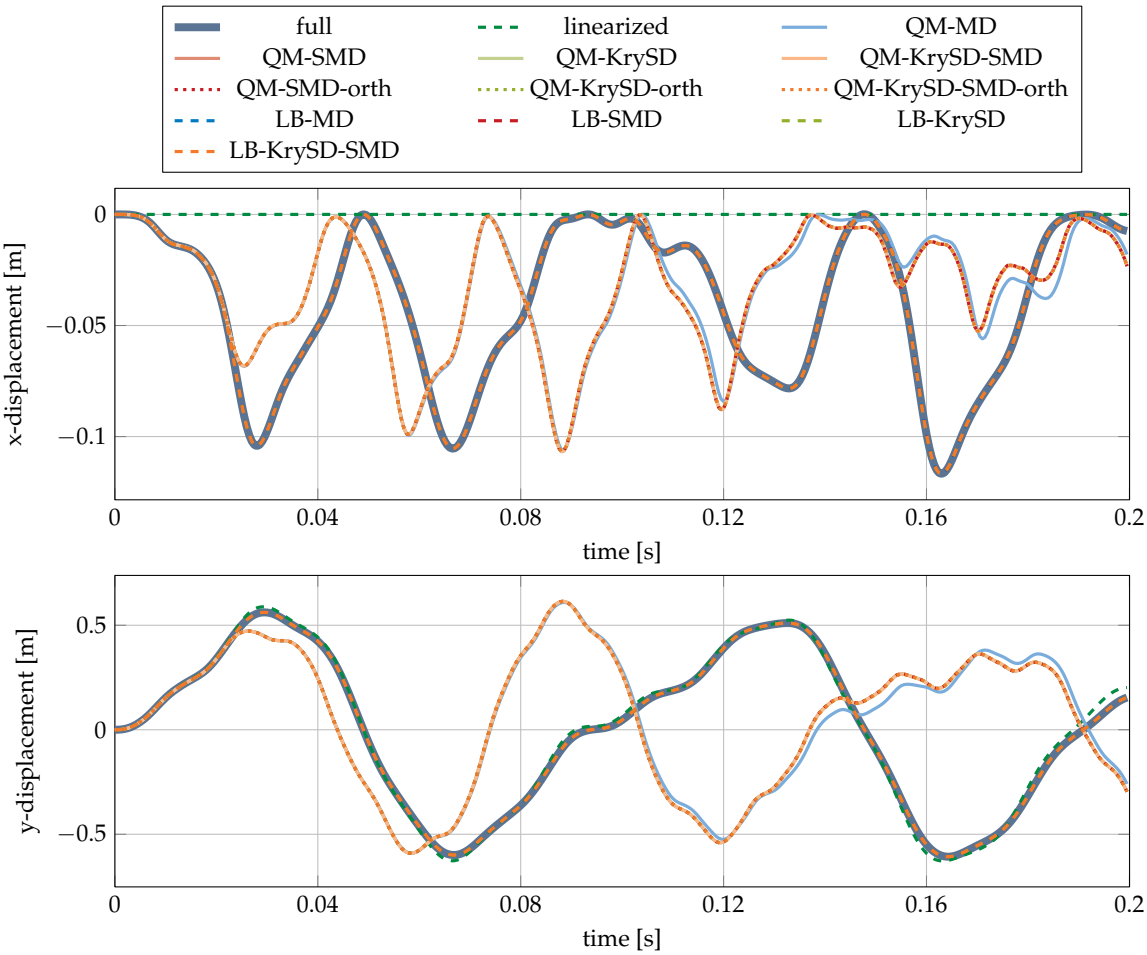


**Figure 7.15:** Displacements at the marked node for the cantilever discretized with solid elements for different reduction techniques using 5 modes.

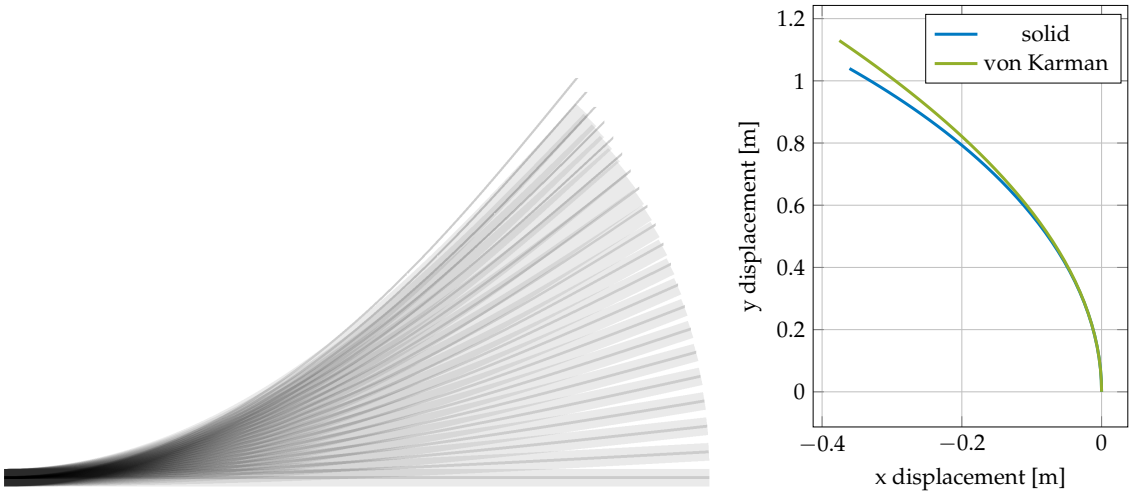


**Figure 7.16:** Static response of the cantilever discretized with von Karman beam elements and triangular solid elements to an increasing tip load.

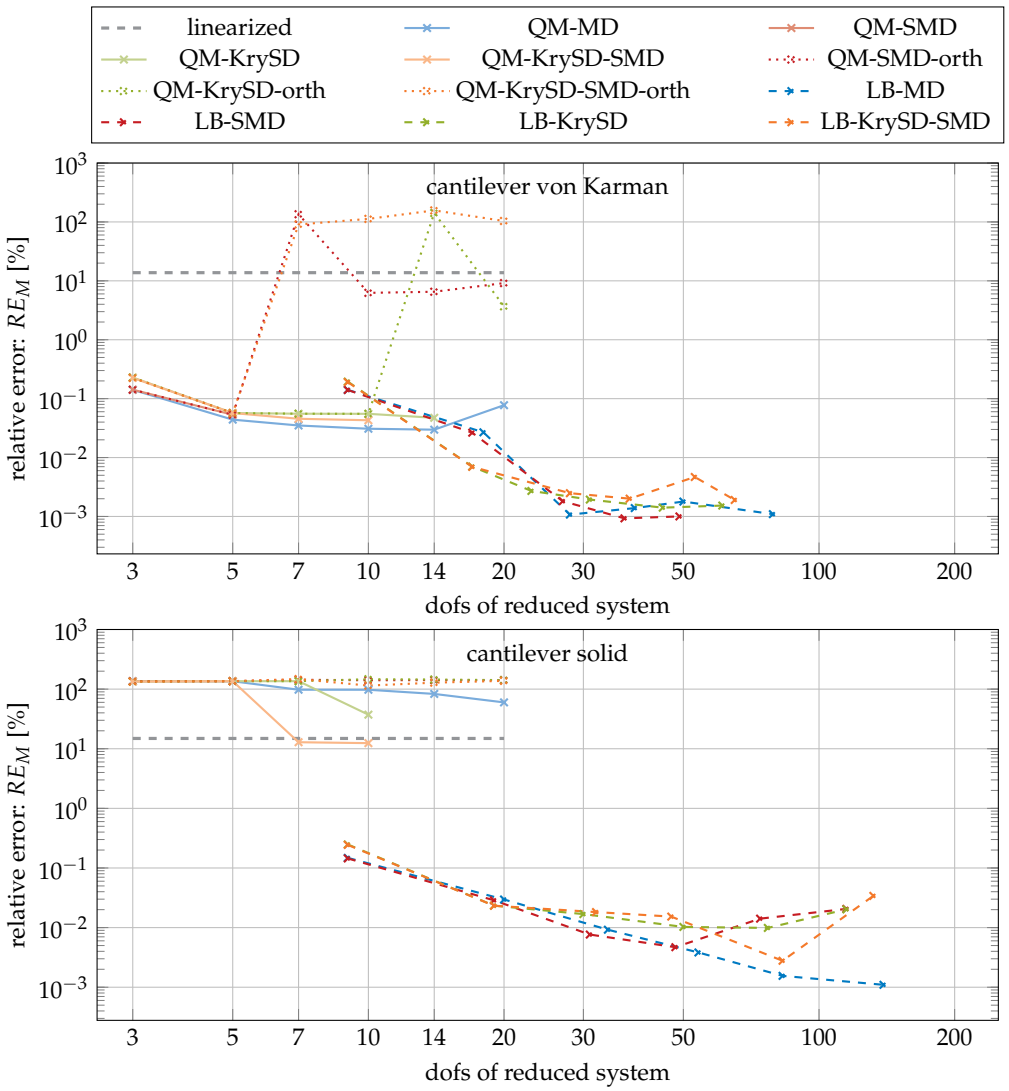


**Figure 7.17:** Relative error of the cantilever discretized with von Karman beam elements (top) and solid elements (bottom) for different reduction techniques and different reduction rates.

# Chapter 8

# Comparison of Bases: Subspace Angles

When a system is reduced using a reduced basis as described in Chapter 3, the subspace spanned by the reduction basis $\boldsymbol{V}$ is the essential contribution to the characteristics of the reduced order model. This subspace can be parametrized in arbitrary variants, since an infinite number of bases $\boldsymbol{V}$ cover exactly the same subspace. ROMs reduced with different bases however covering the same subspace exhibit the same dynamics and lead to exactly the same results, when the physical state is reconstructed. Hence, these equivalent models can be seen as different realizations of the same physical model.

In this chapter, a method to compare different subspaces by the concept of *subspace angles* is presented. It allows comparing different subspaces in a way that is not depending on the specific vectors used to represent them. This chapter is based on the publication [RGR15].

## 8.1 The Basis Problem in Nonlinear Reduction

The issue that an infinite number of systems have the same dynamics, is well-known in systems theory as the *realization problem* [Ant09; Bro15]. To make the comparison of systems independent of the realization, which is in its parts arbitrary, invariant system theoretic properties can be used to describe the characteristics of the reduced system. For linear systems, the most popular tool for investigation is the analysis of the transfer function. This concept works, when only a few inputs and outputs are of interest as it is most often the case in control theory. Hence, many reduction techniques rely on this concept and minimize different error measures on the transfer function, mostly in the $\mathcal{H}_2$- or the $\mathcal{H}_\infty$-norm, (cf. [Ant09]).

In nonlinear systems, however, these system theoretic concepts are either not existent or hard to compute. Hence, most error analyses rely on the comparison of the system response to certain inputs requiring great computational effort. In nonlinear structural dynamics, Nonlinear Normal Modes (NNMs) have been used for the characterization of the reduction quality [KBA14; KA16]. They require the computation of the NNMs, which is commonly performed with a shooting method, where a time integration is necessary. Hence, these methods are also associated with high computational costs.

As a consequence, it is desirable to have a tool to compare the reduction bases for nonlinear systems which do not require a full simulation of the system. In the following a concept is proposed, where the reduction bases are compared in a mathematical rigorous way. It allows only to compare reduction bases and not measuring errors. As a consequence, it cannot be directly applied to assess the quality of reduced bases. However, it can be used to assess the difference between one basis and another basis which is known to be optimal, like the POD basis obtained for a given excitation. Then, for instance simulation-free bases can assessed by the comparison against the optimal basis

without requiring a full simulation.

## 8.2 Principal Angles of Subspaces: Measurements of Bases

The key idea of the comparison of bases is the concept of principal angles of subspaces. This concept is well-known and used in numerical linear algebra, e.g., Betcke and Trefethen [BT05], statistics, e.g., van Overschee and de Moor [VD93] or machine learning, e.g., Dhillon and Modha [DM01]. In model order reduction, it is used as interpretation of the matrix interpolation used in parametric model order reduction, e.g., in Lieu and Lesoinne [LL04], Lieu and Farhat [LF07] and Amsallem [Ams10, p. 93]. It has also been used in damage detection in combination with the POD by de Boe and Golinval [DG03] and sensor validation by Friswell and Inman [FI99] and Kerschen et al. [Ker+04].

In the concept of subspace angles, the $n$-dimensional subspace $\mathcal{S}_1$ spanned by the basis $\boldsymbol{V}_1 \in \mathbb{R}^{N\times n}$ and the $m$ dimensional subspace $\mathcal{S}_2$ spanned by $\boldsymbol{V}_2 \in \mathbb{R}^{N\times m}$ embedded in the $N$ dimensional vector space are compared. If the subspace $\mathcal{S}_1$ is not of higher dimension than $\mathcal{S}_2$, i.e., $n \leq m$, then $n$ so-called principal angles $\gamma_1, \ldots, \gamma_n$ exist, which are recursively defined as

$$\begin{aligned} \underset{\boldsymbol{f}_i, \boldsymbol{g}_i}{\text{maximize}} \quad & \cos\gamma_i = \boldsymbol{f}_i^T \boldsymbol{g}_i \\ \text{subject to} \quad & \boldsymbol{f}_i^T \left[\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{i-1}\right] = \boldsymbol{0}, \quad \boldsymbol{g}_i^T \left[\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{i-1}\right] = \boldsymbol{0}, \\ & \boldsymbol{f}_i \in \mathcal{S}_1, \quad \boldsymbol{g}_i \in \mathcal{S}_2, \quad |\boldsymbol{f}_i|_2 = 1, \quad |\boldsymbol{g}_i|_2 = 1 \end{aligned} \tag{8.1}$$

with the $i$-th principal vector $\boldsymbol{f}_i$ of subspace $\mathcal{S}_1$ and $\boldsymbol{g}_i$ of subspace $\mathcal{S}_2$. The principal vector pair $\boldsymbol{f}_i$ and $\boldsymbol{g}_i$ are the closest possible vectors with unit length which are both members of the corresponding subspaces and orthogonal to the previous principal vectors of the same subspace. The subspace angles are the angles between the two vectors of the principal vector pair indicating the difference between the two subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$.

Two relevant properties arise from the recursive definition (8.1). First, the subspace angles are bound to the interval between zero and $\pi/2$:

$$0 \leq \gamma_i \leq \pi/2 \quad \forall \quad i \in \{1, \ldots, n\}, \tag{8.2}$$

since the inner product of two unit vectors cannot be larger than one and the subspace angles are defined to be positive. Secondly, the subspace angles are defined to be monotonically increasing

$$\gamma_i \leq \gamma_{i+1} \quad \forall \quad i \in \{1, \ldots, n-1\}, \tag{8.3}$$

so that the first subspace angle $\gamma_1$ is the smallest and the last subspace angle $\gamma_n$ is the largest subspace angle.

The concept of subspace angles identifies the separated and overlapping dimensions of two subspaces. For instance, if $\mathcal{S}_1$ and $\mathcal{S}_2$ are equal and of same dimension, then all subspace angles are zero. If $\gamma_n = 0$ and $\mathcal{S}_2$ is of higher dimension than $\mathcal{S}_1$ ($m > n$), $\mathcal{S}_1$ is fully included into $\mathcal{S}_2$. On the contrary, if all subspace angles are equal to $\pi/2$, i.e., $\gamma_0 = \pi/2$, then the two subspaces are fully orthogonal and do not share any vector. In the case between the two aforementioned cases, principal vectors associated with small principal angles indicated directions in the subspaces, where $\mathcal{S}_1$ and $\mathcal{S}_2$ overlap, and on the contrary, principal vectors associated with large principal angles indicate directions, where $\mathcal{S}_1$ and $\mathcal{S}_2$ are different.

This concept is illustrated in Figure 8.1 where two two-dimensional subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ are depicted in $\mathbb{R}^3$. $\mathcal{S}_1$ is realized by the basis $\boldsymbol{V}_1 = (\boldsymbol{v}_{11}, \boldsymbol{v}_{12})$ and $\mathcal{S}_2$ is realized by $\boldsymbol{V}_2 = (\boldsymbol{v}_{21}, \boldsymbol{v}_{22})$. Both subspaces are planes in three dimensional space, which have to

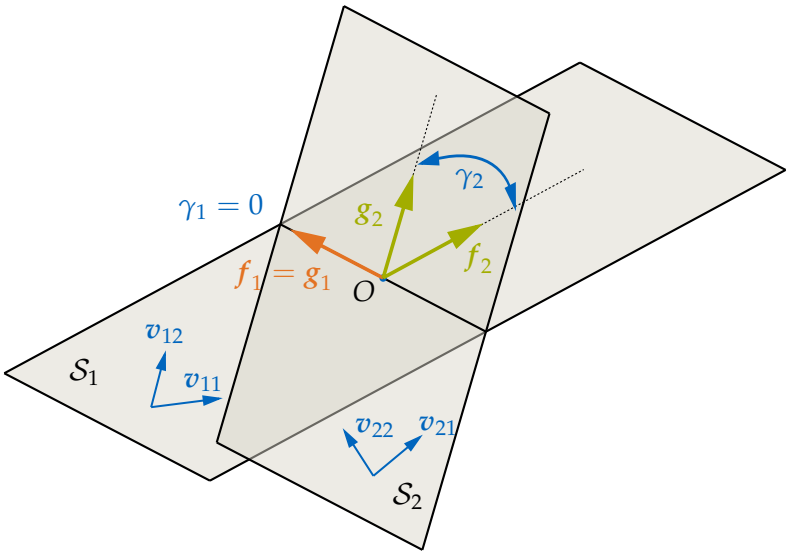


**Figure 8.1:** Illustration of the concept of subspace angles

include the origin $O$. The first principal vectors $f_1$ and $g_1$ are equal and yield the intersection of the two planes, which is a line. The corresponding principal angle $\gamma_1$ is equal to zero. The second principal vectors $f_2$ and $g_2$ are the vectors of $\mathcal{S}_1$ and $\mathcal{S}_2$ perpendicular to the intersecting line. They yield the angle between the two planes, which is the second principal angle $\gamma_2$. Whereas in this example the first principal angle $\gamma_1$ is always zero, the second principal angle can have all values between 0 and $\pi/2$. For $\gamma_2 = 0$, both planes coincide, for $\gamma_2 = \pi/2$ both planes are orthogonal.

In the illustration of a three dimensional vector space, the concept of subspace angles of subspaces with different dimensions can also be interpreted. The principal angle of a one dimensional subspace compared to a two dimensional subspace is the angle between a line and a plane. In this setup, the full separation of subspaces is possible, if the line is perpendicular to the plane and the principal angle is hence $\pi/2$. If two one dimensional subspaces are compared in $\mathbb{R}^3$, the principal angle is the angle between the two lines through the origin.

The computation of the principal angles and the principal vectors is given in Algorithm 1. It involves solely linear algebra manipulations, which is computationally no big burden, if the dimensions of the subspaces $n$ and $m$ are moderate. Further insights into the concept of subspace angles are given in the exhaustive textbook of Golub and van Loan [GV12]. The computation of the principal angle using the arccosine operation in line 4 in Algorithm 1 can lead to numerical errors when the principal angles are small. However, Björck and Golub [BG73] discuss a way to circumvent this issue by computing the small angles by an arcsin function. If the principal angles and vectors should not be computed in the Euclidean $L^2$ norm but in the $\boldsymbol{M}$-norm, since e.g. different displacement variables like displacements and rotations are used, the subspace angles can be computed according to the algorithms proposed in [KA02]. In the following applications, the subspace angles are computed according to the straightforward algorithm given in Algorithm 1, since the computation did not exhibit any numerical issues for these cases.

**Algorithm 1:** Algorithm for computing the principal vectors and subspace angles of two bases $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$, based on [GV12]

**Input** : Reduced bases $\boldsymbol{V}_1 \in \mathbb{R}^{N\times n}$ and $\boldsymbol{V}_2 \in \mathbb{R}^{N\times m}$ spanning $\mathcal{S}_1$ and $\mathcal{S}_2$
**Output** : Vector of subspace angles $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)$,
principal vectors $\boldsymbol{F} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n)$, $\boldsymbol{G} = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n)$

1 $\boldsymbol{V}_1 = \boldsymbol{Q}_1\boldsymbol{R}_1, \quad \boldsymbol{V}_2 = \boldsymbol{Q}_2\boldsymbol{R}_2$ // Orthogonalize reduced bases $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ via QR decomposition
2 $\boldsymbol{C} = \boldsymbol{Q}_1^T\boldsymbol{Q}_2$ // Compute the correlation matrix of the two orthogonalized subspaces
3 $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{C}$ with $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma)$ // Compute the SVD of the correlation matrix
4 $\boldsymbol{\gamma} = \arccos(\sigma)$ // Compute the subspace angles from the singular values
5 $\boldsymbol{F} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n) = \boldsymbol{Q}_1\boldsymbol{U}$ // Compute the principal vectors, if necessary
6 $\boldsymbol{G} = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n) = \boldsymbol{Q}_2\boldsymbol{V}$

## 8.3 Applications

To show the usefulness of the subspace angles and the corresponding principal vectors, the POD reduction basis and the simulation free reduction basis are investigated with this tool on the c-shape example of Section 6.4. The model is run with exactly the same model parameters as there, however the simulation time is taken longer from 2 s to 3 s.

First, a POD is run. In the case when no parameters in the model are changed, the POD basis is the optimal basis as discussed in Chapter 5. The first 100 singular values of the POD are depicted in Figure 8.2. They indicate that the model is of limited complexity, since the decay in the first 10 to 20 POD modes is over 3 to 4 orders of magnitude. To check the results, one simulation is run with 10 POD vectors yielding a relative error of $RE = 8.7\,\%$ and with 20 POD vectors yielding $RE = 0.12\,\%$. The first 10 POD modes can be considered as necessary to obtain good results while the first 20 POD modes are necessary for very good results. For the sake of better presentability, the first 10 POD modes are used as benchmark modes to be captured by a simulation free basis.

The naive basis composed solely of vibration modes has been shown to fail in capturing the nonlinear effects, as illustrated with the static example in Section 6.1. A similar investigation is performed by comparing purely modal bases of different sizes against the ten dimensional POD basis using subspace angles. The results of this investigation are depicted in Figure 8.3. As expected, a basis of 10 vibration modes is not equal to the POD basis and consequently many subspace angles are fairly large. However, if the number of vibration modes in the basis is increased, the subspace angles with greater indices decrease only slowly, seemingly converging to a fairly large value.

To investigate this issue further, the *principal vectors* for the subspaces spanned by 40 vibration modes versus ten POD modes are computed and depicted as displacement fields of the mesh in Figure 8.4. They confirm the observations already made. In accordance with Figure 8.3, the first four subspace angles are comparably low, and no difference can be seen in the principal vectors, which are plotted as displacement field. However, the principal vectors with larger indices show some deviation, so that the blue principal vectors of the modal subspace do not cover fully the green principal vectors of the POD subspace. A closer look at the difference reveals that the main difference between the modal subspace and the POD subspace is the contraction and the expansion of the mesh.

As discussed in Chapter 6, the linear basis fails to capture the nonlinear effects of the large rotation. These large rotations cause an enlargement of the mesh when they are linearized. Consequently, the linear bases fail to capture the displacement fields necessary for displaying nonlinear large rotations. However, the (static) modal derivatives are able to capture them.

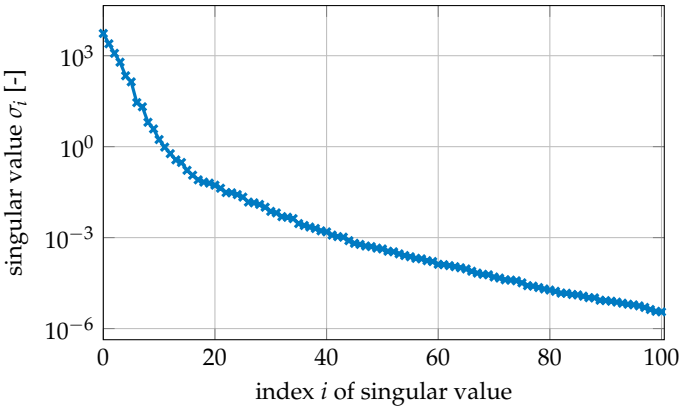


**Figure 8.2:** Singular values of the snapshot matrix built by a full training simulation. The left singular vectors associated with the first ten singular values were used as POD-basis.

Hence, the subspace angles between the first ten POD modes and a simulation-free basis composed of the first 6 vibration modes and 14 SMDs are computed. The SMDs are selected based on the frequency weighting algorithm introduced in Subsection 6.2.4. The principal angles of this computation are also given in Figure 8.3 together with the principal angles of the purely modal bases. The subspace angles of the simulation-free basis composed of SMDs are all fairly small indicating a good match with the POD basis. Also the principal vectors of the two subspaces, which are depicted in Figure 8.5, show an extremely good overlap of the simulation-free basis and the POD basis. The principal vectors of the simulation-free basis (blue) practically cover the principal vectors of the POD basis.

The similarity of the simulation-free subspace and the POD subspace is also validated with a time integration of the simulation-free reduced system. It exhibits a relative error of $RE = 2.0\,\%$ indicating that the subspace is chosen well with the simulation free technique. To recall, the relative errors of the POD reduced system were $8.7\,\%$ for 10 POD vectors and $0.12\,\%$ for 20 POD vectors. Given that the simulation free technique does not know the excitation, which the POD method does, the simulation-free basis turns out to be very effective for these types of geometrically nonlinear problems.

The concept of subspace angles thereby turns out to be a valuable tool to compare the subspaces spanned by the bases. Furthermore, the principal vectors can be used to identify the similarities and differences of subspaces. Hence, they are an ideal tool for post-processing, debugging, subspace identification of certain nonlinear methods and the design of new methods.

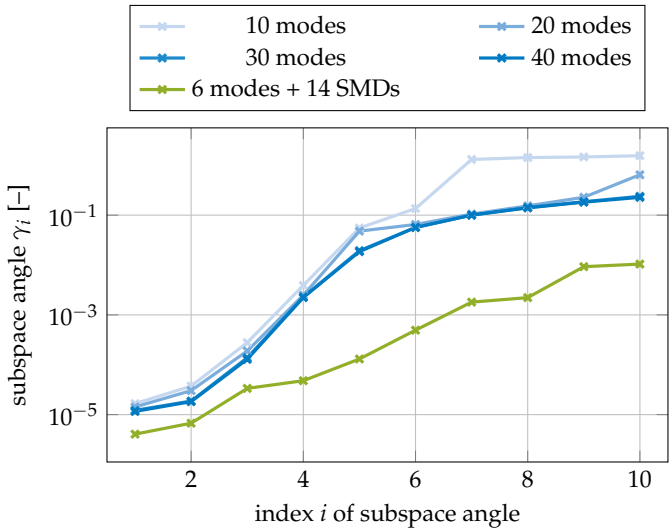


**Figure 8.3:** Subspace angles of 10 POD modes and a number of vibration modes (blue lines) for the c-shape example. The green line denotes the subspace angles of the POD basis versus a basis composed of 6 vibration modes + 14 SMDs selected with the frequency weighting technique as explained in Subsection 6.2.4. The corresponding principal vectors are depicted in Figure 8.4 and Figure 8.5.

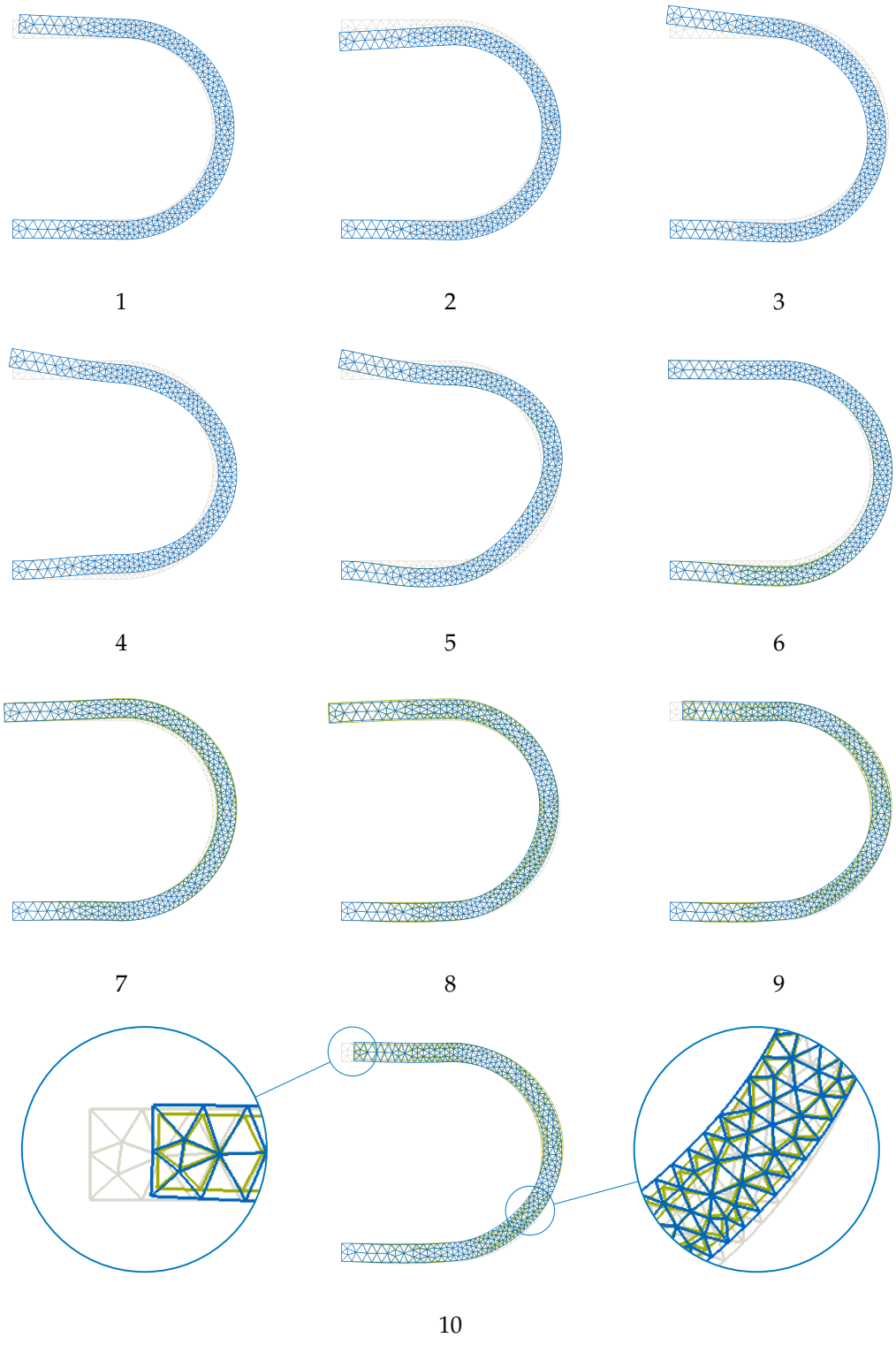


**Figure 8.4:** Principal vectors $f_i$ and $g_i$ of the two subspaces spanned by the first 10 POD vectors (green) and the first 40 vibration modes (blue) for the c-shape example.

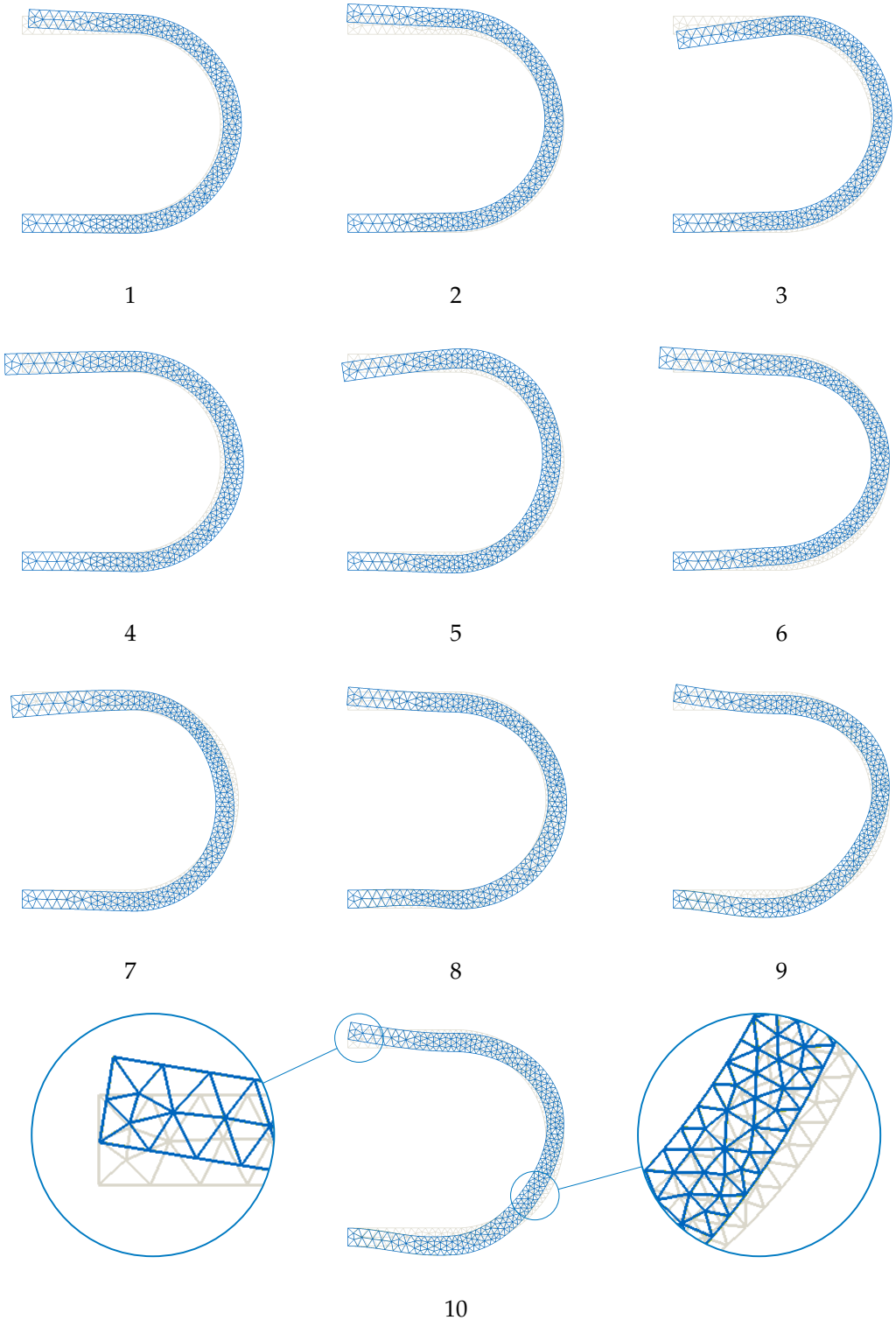


**Figure 8.5:** Principal angles of the two subspaces spanned by the first 10 POD vectors and the simulation free basis containing 6 vibration modes and 14 SMDs selected by frequency weighting for the c-shape example.

# Chapter 9

# Summary of Part I

In the previous six chapters, various aspects of the reduced basis method have been discussed, which are summarized in this chapter. The main question in projective model order reduction is the determination of a reduced basis $\boldsymbol{V} \in \mathbb{R}^{N \times n}$ spanning the subspace, in which the relevant dynamics of the system of dimension $N$ happens. The projection of the system onto the subspace spanned by $\boldsymbol{V}$ is equivalent to a coordinate transform with a subsequent truncation, resulting into a system on $n$ reduced equations.

In linear structural dynamics, the established reduction methods for determining $\boldsymbol{V}$ rely on system intrinsic properties, which are often supported by system-theoretic concepts. This is due to the fact, that the linear system is fully characterized by its mass, stiffness and damping matrix, which can be analyzed by system theoretic procedures relying on linear algebra operations. The two methods covered in Chapter 4 are modal truncation, which relies on the decoupling of states, and the Krylov subspace method, where the higher order displacement fields of the force inputs are regarded. It is further shown, that the Krylov subspace method exists also in systems theory, where the method matches the moments of the transfer function.

In nonlinear systems, the characterization of the system solely with matrices is not possible, though, since the nonlinear operators cannot be cast into matrix-vector operations. As a consequence, the state-of-the–art reduction method for nonlinear systems, the Proper Orthogonal Decomposition (POD) discussed in Chapter 5, takes a detour over a statistical evaluation of training sets obtained with expensive training simulations of the full, unreduced system. While this method builds an optimal basis for the given training sets and is not limited to structural dynamics systems, the requirement of the full training simulation is very prohibitive in many contexts, where the costs of the full simulation are not affordable or even infeasible. Furthermore, the method is very sensitive with respect to the training sets requiring them to be representative and not too different for the cases for the reduced system.

The disadvantages of training simulations associated with the construction of a reduced basis are the motivation for the so-called *simulation-free* reduced basis methods addressed in Chapter 6. There, extensions to the so-called *lin basis* obtained from a linear reduction method applied to the linearized system are addressed. As exemplarily shown, these lin bases exhibit severe locking when applied to geometrically nonlinear systems making them useless for the reduction of these systems. However, they form the starting point for the established modal derivatives (MDs) and static derivatives (SDs), which capture the nonlinear effects of the system. The concatenated lin basis and the corresponding MDs or SDs form an excellent simulation-free basis suited for geometrically nonlinear problems. The MDs are based on the perturbation of vibration modes and require the solution of multiple singular systems in order to obtain the augmentation vectors for the lin basis. Hence, this technique is limited to a lin basis composed of vibration

modes. The SDs, on the contrary, base on the perturbation of the nonlinear internal forces and are hence applicable to every lin basis. Furthermore, they are less computational expensive, since only one non-singular system needs to be solved for multiple right hand sides.

The computation of both the MDs and the SDs requires derivatives of the nonlinear internal forces. For this computation, a numerical differentiation approach is suggested, which requires solely the computation of tangential stiffness matrices. The differentiation scheme, however, is sensitive with respect to the step width requiring a tuning of this parameter. This is achieved by exploiting the fact, that the SDs are symmetric allowing for tracking the accuracy of the differentiation by evaluating the symmetry error. The quadratic growth of the number of the augmentation vectors with respect to the lin basis is a further issue of the simulation-free reduction schemes covered. Hence, selection strategies which heuristically rank the importance of the SDs or MDs are discussed. In the numerical experiments, reduced bases built with the different simulation-free methods are investigated. Thereby, the MDs turn out to be excellent ingredients for the reduced basis. The SDs perform also well, especially when combined with a selection strategy. For the examples which are investigated, the frequency weighting strategy in combination with static modal derivatives turns out to be the best, since it provides good performance with computational ease.

A second attempt to address the quadratic growth of the number of SDs and SMDs with respect to the size of the lin basis is given in Chapter 7. In this novel approach named Quadratic Manifold (QM), the basis projection is not linear but nonlinear, projecting the equations of motion onto the tangent space of the nonlinear transformation. The nonlinear function is chosen to be quadratic, enslaving the modal derivatives with the vibration modes which form the linear part of the transformation. Alternatively, when the linear part is built from basis vectors different from vibration modes, the so-called *force compensation approach* allows to transfer the quadratic part of the nonlinear forces to the quadratic transformation. Interestingly, it turns out that the resulting quadratic part is composed of the SDs. Hence, the QM approach provides a further theoretic foundation to the SDs.

The QM approach allows to keep the number of unknowns minimal, since the MDs or SDs are not treated as independent dofs. Especially when dealing with flat structures using von Karman kinematic assumptions within beam or shell elements, a sound theory is derived supporting the good performance for applications with these structures. The QM approach, however, turns out to be fairly unstable for discretizations using solid elements. This can be solved by an orthogonalization procedure, which destroys the original form of the quadratic mapping, though. Since sufficient accuracy is gained only for membrane-dominated motions, as shown in the extensive numerical investigation, the applicability of this method is fairly limited.

The key in projective model order reduction is the subspace spanned by the reduced basis $\boldsymbol{V}$. However, since the subspace is always realized by a parametrization leading to the projection basis $\boldsymbol{V}$, the comparison of subspaces is a non-trivial task. In Chapter 8, subspace angles and principal vectors are suggested as main tools for the comparison of reduced bases. Thereby, the angles give an insight to the proximity of the two investigated subspaces and detect intersections and differences. But not only the principal angles, but especially the principal vectors can be used to identify the differences of the subspaces making physically-based insights possible. In the context of geometrically nonlinear reduction it is shown, that even a large reduction basis composed of vibration modes does not capture the displacement fields necessary to describe rigid body rotations of elements. It is further shown that the augmentation vectors in the simulation free approaches like static derivatives capture these displacement fields.

Part I of this thesis discusses extensively the reduced basis approach which reduces the dimension of the system from a large number $N$ to the small number $n$. As a consequence, the solution of the system of linear equations necessary in the Newton-Raphson loop is accelerated. However, since the reduced nonlinear internal forces cannot be precomputed a priory, the computational bottleneck of the reduced basis system is the construction of the internal forces and, in implicit time integrations, the tangential stiffness matrix. This issue, which is addressed with hyper-reduction, is the topic of the second part of this thesis to follow.

# Part II

# Hyper-Reduction

# Chapter 10

# The Need for Hyper-Reduction in Nonlinear Model Order Reduction

The reduced basis technique applied to nonlinear systems reduces the size of the semi-discretized equations of motion from dimension $N$ to the much smaller dimension $n$. As a consequence, the number of unknowns and hence the computational cost associated with the solution of linearized systems of equations is drastically reduced. However, after accelerating the linear systems solver necessary in implicit time integration schemes by reducing the number of unknowns, a new computational bottleneck emerges, which is still associated with the size of the full, unreduced system: the calculation of the internal nonlinear forces. For accelerating the computation of these nonlinear internal forces, the term hyper-reduction was created by Ryckelynck [Ryc05], even though attempts to address this issue were undertaken before.

In the following section, the problem of hyper-reduction is formulated and different approaches are discussed in the following three chapters. They are applied to rather academic problems in order to give an insight to the methods. In Chapter 15, they are applied to a real-life system to demonstrate the applicability in an industrial context.

## 10.1 Problem Statement

For linear systems, all members in the equations of motion are linear functions: The inertia forces are a linear function of the generalized accelerations, the damping forces are a linear function of the generalized velocities and the restoring forces are linear with respect to the displacements. Hence all these functions can be expressed as matrix-vector products. For projective model order reduction, this is of eminent benefit, since the matrices can be projected on the reduced basis as depicted in Figure 10.1 resulting in reduced matrices which can be precomputed.

In nonlinear systems, some functions in the equations of motion are nonlinear by definition. In contrast to the linear functions, no underlying matrices can be precomputed. Since the nonlinearity of the restoring forces in geometrically nonlinear finite elements are formulated on the element level, the reduced internal force vector still requires the computation of the full, unreduced, nonlinear internal force vector, which is projected onto the reduction basis afterwards.

This computational sequence depicted in Figure 10.1 is the new computational bottleneck in both static and dynamic simulations. It requires the expansion of the reduced coordinates $\boldsymbol{q}$ to the full displacement field $\boldsymbol{u} = \boldsymbol{V}\boldsymbol{q}$, the computation of the full nonlinear internal force vector $\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q})$ and the projection onto the reduced basis $\boldsymbol{V}$ to obtain $\boldsymbol{f}_r = \boldsymbol{V}^T\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q})$. While the number of equations is reduced to $n \ll N$ and hence the burden of the original dimension $N$ of the problem is overcome in the solution, it is still

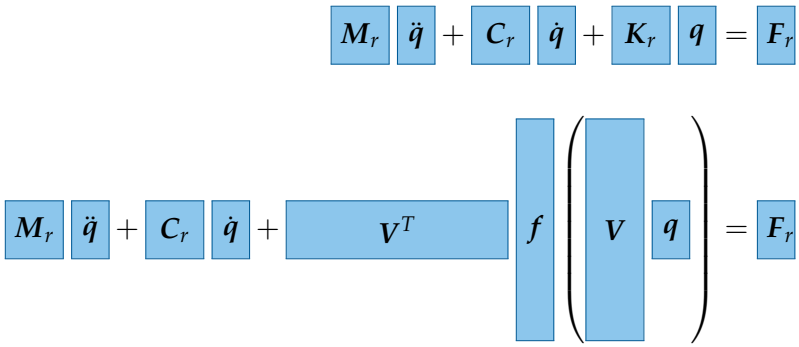


**Figure 10.1:** Matrix-vector operations for a linear reduced system (top) and a nonlinear reduced system (bottom). The nonlinear system requires the evaluation of the nonlinearity in the physical, unreduced domain.

present in the evaluation of the nonlinearity. Even worse, since the physical states of the unreduced system have to be reconstructed from the reduced generalized coordinates and projected back onto the subspace, the computation of the reduced nonlinear forces is even more costly than the plain evaluation of the nonlinear forces in the unreduced, high dimensional domain.

In nonlinear static and implicit time integration scenarios, it is not only the reduced, nonlinear internal force vector which has to be computed in this computational sequence, but also the reduced tangential stiffness matrix. They have to be computed in every single step of the Newton-Raphson loop of static or dynamic computations, since both the reduced internal force vector and the reduced tangential stiffness matrix are functions of the reduced generalized coordinates.

As a consequence, hyper-reduction methods have been developed to reduce the computational cost associated with the computation of the reduced forces. Their key idea is the approximation of the reduced internal force vector $\boldsymbol{f}_r(\boldsymbol{q})$ and the tangential stiffness matrix $\boldsymbol{K}_r(\boldsymbol{q})$ by the hyper-reduced internal force vector $\boldsymbol{f}_{r,hr}$ and tangential stiffness matrix $\boldsymbol{K}_{r,hr}$:

$$\boldsymbol{f}_r(\boldsymbol{q}) = \boldsymbol{V}^T \boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}) \approx \boldsymbol{f}_{r,hr}\,, \qquad \boldsymbol{K}_r(\boldsymbol{q}) = \boldsymbol{V}^T \boldsymbol{K}(\boldsymbol{V}\boldsymbol{q})\boldsymbol{V} \approx \boldsymbol{K}_{r,hr}\,. \tag{10.1}$$

The resulting equations of motion of the hyper-reduced system are then the reduced basis system with the hyper-reduced internal forces:

$$\boldsymbol{M}_r\ddot{\boldsymbol{q}} + \boldsymbol{C}_r\dot{\boldsymbol{q}} + \boldsymbol{f}_{r,hr} = \boldsymbol{g}_r. \tag{10.2}$$

The matrix $\boldsymbol{M}_r = \boldsymbol{V}^T\boldsymbol{M}\boldsymbol{V}$ is the reduced mass matrix, $\boldsymbol{C}_r = \boldsymbol{V}^T\boldsymbol{C}\boldsymbol{V}$ the reduced damping matrix and $\boldsymbol{g}_r = \boldsymbol{V}^T\boldsymbol{g}$ the vector of the reduced external force.

As (10.2) and (10.1) clearly indicate, hyper-reduction is an approximation of the reduced internal forces *on top* of a reduced basis method. Thereby, hyper-reduction can be realized in different ways. The approaches named DEIM and ECSW, which are discussed in Chapters 12 and 13, exploit the sum-structure of the finite element assembly, see Subsection 2.3.2, and provide an approximation of the reduced forces based on less summands. The computational speedup is realized as less elements of the mesh are evaluated. The polynomial tensors approach discussed in Chapter 11 takes a different tour and exploits the polynomial structure of the nonlinear function for the special case of St. Venant-Kirchhoff materials. There, the computational gain is associated with the smaller costs of evaluating a multidimensional polynomial for small $n$ compared to the evaluation of the reduced internal forces as $\boldsymbol{V}^T\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q})$.

## 10.2 Measurement of Hyper-Reduction Error

Since the hyper-reduction is applied to a reduced basis system, two errors are cumulated when a hyper-reduced model is compared to the full, unreduced high dimensional model. The first error is due to the reduced basis, as discussed in Section 3.3. The second error is due to the actual hyper-reduction approximation.

In order to assess the hyper-reduction methods properly, two error measures are used. The full relative error $RE_f$ gives the relative error $RE$ of the hyper-reduced model relative to the full, unreduced model. This error is caused by both the reduced basis approximation of the displacements and the hyper-reduction approximation of the nonlinearity. To recall the relative error $RE$ from (3.9):

$$RE_{f/hr} = \frac{\sqrt{\sum_{t\in\mathcal{T}} \Delta\boldsymbol{u}(t)^T \Delta\boldsymbol{u}(t)}}{\sqrt{\sum_{t\in\mathcal{T}} \boldsymbol{u}_{\text{ref}}(t)^T \boldsymbol{u}_{\text{ref}}(t)}} \cdot 100\% \quad \text{with} \quad \Delta\boldsymbol{u}(t) = \boldsymbol{u}(t) - \boldsymbol{u}_{\text{ref}}(t). \tag{10.3}$$

Hence, for the $RE_f$, $\boldsymbol{u}_{\text{ref}}$ is the displacement field of the full, unreduced simulation and $\boldsymbol{u} = \boldsymbol{V}\boldsymbol{q}_{hr}$ is the reconstructed full displacement field of the hyper-reduced model. For the second error measure $RE_{hr}$, the relative error measuring the error due to the hyper-reduction approximation, $\boldsymbol{u}_{\text{ref}} = \boldsymbol{V}\boldsymbol{q}$ is the reconstructed full displacement field of the reduced basis model and $\boldsymbol{u} = \boldsymbol{V}\boldsymbol{q}_{hr}$ is the reconstructed displacement field of the hyper-reduced model. In the numerical experiments of hyper-reduced systems, both error measures are reported.

In the following chapters, several numerical experiments are conducted, of which the wall times for the computation are reported. All numerical experiments are run with the nonlinear finite element package AMFE, which is presented in detail in Chapter A. It is written mostly in Python with the time critical parts implemented in Fortran in order to achieve computation times in the range of commercial finite element codes. All experiments were conducted on a workstation with 32 GB RAM and an Intel Xeon processor operating at 3.6 GHz.

# Chapter 11

# Polynomial Tensors

Hyper-reduction, as pointed out in the previous chapter, is about the approximation of the nonlinearity in the equations of motion. The common concept to achieve this is the exploitation of the structure of the nonlinear terms being a sum, since the nonlinear term is obtained by adding up the contributions of all elements in the mesh. Hyper-reduction using polynomial tensors is different. It is not the sum structure of the nonlinearity, which is exploited, but the polynomial structure of the nonlinearity, which is approximated using a Taylor series. When the Taylor series converges quickly, a reduction is achieved by storing the coefficients of the Taylor series of the reduced system, making the computation of every elemental contribution unnecessary.

This can be a huge benefit in many cases, since the elemental formulation is not explicitly called in the computations of the reduced system. Furthermore, this allows for an export of the reduced model in a standardized way, since only the polynomial tensors need to be exported similar to the export of the mass and linear stiffness matrix. Hence, no element libraries have to be called for time integration. However, the polynomial tensors technique is only limited to models with a particular nonlinearity like finite element systems with St. Venant-Kirchhoff materials.

The approach to hyper-reduce geometrically nonlinear systems by using polynomial expressions can be traced back at least to Nash [Nas78] and Almroth et al. [ASB78], where geometrically nonlinear, flat and curved beams, plates and shells are reduced using polynomial coefficients. They are computed by directly using the finite element formulation of the elements. Similar approaches are taken in Przekop et al. [Prz+04] and Shi and Mei [SM96]. Since these techniques require specialized finite element codes, indirect methods for the use with black-box commercial finite element codes are proposed by Muravyov and Rizzi [MR03] and Kim et al. [Kim+13]. These methods compute the polynomial coefficients by evaluating the nonlinear forces resulting from prescribed displacements. An enhancement to these methods is proposed by Perez et al. [PWM14] and Phlipot [Phl+14], where the tangential stiffness matrix instead of the internal force vector is evaluated leading to clearly reduced offline costs, especially for larger models. Another approach proposed by McEwan et al. [McE+01] called implicit condensation identifies the polynomial coefficients by prescribing external loads and evaluating the resulting displacement fields of the nonlinear system.

The Implicit Condensation and Expansion (ICE) method proposed in Hollkamp and Gordon [HG08] addresses the two tasks of computing augmentation vectors for the reduced basis as discussed in Chapter 6 and determining the polynomial coefficients at once. Thereby, this method evaluates the displacement fields of the nonlinear system triggered by prescribed external loads.

The polynomial tensor approach is widely used for geometrically nonlinear applications like aeroelasticity, see for instance Abdel-Motaglay et al. [ACM99], Guo and Mei

[GM03] and Ghoman and Azzouz [GAM09] or computer graphics, e.g., Capell et al. [Cap+02] or Barbič and James [BJ05]. In the latter, the polynomial tensor approach is used in combination with the static modal derivatives discussed in Subsection 6.2.2. Furthermore, substructuring techniques have been developed using polynomial expansion, e.g., Kuether and Allen [KA14] and Kuether et al. [KAH15], as well as methods for use with uncertainty, e.g., Capiez-Lernout et al. [CSM14]. For an overview of the different methodologies and applications, the interested reader is referred to Mignolet et al. [Mig+13].

## 11.1 Key Idea: Taylor Expansion

The key idea in polynomial tensor hyper-reduction is the approximation of the nonlinear restoring forces $\boldsymbol{f}_r(\boldsymbol{q}) \in \mathbb{R}^n$ as a Taylor expansion around the point of equilibrium. As in linear systems, the displacements $\boldsymbol{q} \in \mathbb{R}^n$ are zero at the point of equilibrium, so that the constant part is zero. The Taylor expansion up to the $k$-th order is then:

$$\boldsymbol{f}_r(\boldsymbol{q}) = \frac{\partial \boldsymbol{f}_r}{\partial \boldsymbol{q}}\boldsymbol{q} + \frac{1}{2}\frac{\partial^2 \boldsymbol{f}_r}{\partial \boldsymbol{q}\partial \boldsymbol{q}}\boldsymbol{q}\boldsymbol{q} + \frac{1}{6}\frac{\partial^3 \boldsymbol{f}_r}{\partial \boldsymbol{q}^3}\boldsymbol{q}\boldsymbol{q}\boldsymbol{q} + \cdots + \frac{1}{k!}\frac{\partial^k \boldsymbol{f}_r}{\partial \boldsymbol{q}^k}\boldsymbol{q}^k + O(\boldsymbol{q}^{k+1}) \tag{11.1}$$

$$= \boldsymbol{K}^{(1)}\boldsymbol{q} + \frac{1}{2}\boldsymbol{K}^{(2)}\boldsymbol{q}\boldsymbol{q} + \frac{1}{6}\boldsymbol{K}^{(3)}\boldsymbol{q}\boldsymbol{q}\boldsymbol{q} + \cdots + \frac{1}{k!}\boldsymbol{K}^{(k)}\boldsymbol{q}^k + O(\boldsymbol{q}^{k+1}). \tag{11.2}$$

Since the internal force vector is a multidimensional function, the Taylor coefficients are symmetric arrays of dimension $k+1$. They can be interpreted as higher order stiffness arrays $\boldsymbol{K}^{(1)}, \ldots, \boldsymbol{K}^{(k)}$, of which the first array $\boldsymbol{K}^{(1)} = \partial \boldsymbol{f}_r / \partial \boldsymbol{q}$ yields the reduced stiffness matrix $\boldsymbol{K}_r \in \mathbb{R}^{n\times n}$. The second term is the reduced quadratic stiffness tensor $\boldsymbol{K}^{(2)} = \partial^2 \boldsymbol{f}_r / \partial \boldsymbol{q}^2 \in \mathbb{R}^{n\times n\times n}$ and so on. Throughout this thesis, the polynomial tensors $\boldsymbol{K}^{(k)}$ are defined as the $k$-th derivative of the reduced internal force vector $\partial^k \boldsymbol{f}_r / \partial \boldsymbol{q}^k$ without the coefficient $1/k!$ of the Taylor series. It is pointed out, that in several publications the coefficients $1/k!$ are added to the tensors.

The size of the coefficient arrays grows exponentially with the number of members in the Taylor series. As the higher order polynomial terms are very inefficient for both storing in memory and evaluating through multiplication, a polynomial tensors hyper-reduction is only efficient when only few members of the Taylor series (11.1) are necessary to approximate the nonlinear forces accurately. Hence, the convergence of the series is of extreme importance. To assess this, the mathematical structure of the internal forces of a St. Venant-Kirchhoff material, which mimics the linear-elastic behavior for large deformations, is investigated next.

## 11.2 Polynomial Structure for St. Venant-Kirchhoff materials

For analyzing the polynomial structure of the restoring forces of a hyperelastic St. Venant-Kirchhoff material, the Total Lagrangian formulation discussed in Section 2.1 is used. The deformation is characterized by the deformation gradient $\boldsymbol{F}$ which is the spatial derivative of the continuous displacement field $\bar{\boldsymbol{u}}$ with respect to the undeformed configuration $\bar{\boldsymbol{X}}$. In the finite element approximation, the continuous displacement field is expressed by the nodal displacements $\boldsymbol{u}$, which are the amplitudes of the spatial shape functions $\boldsymbol{N}$. As a consequence, $\bar{\boldsymbol{u}}$ is a linear function of $\boldsymbol{u}$. As the spatial derivative of the deformation gradient acts only on the shape functions (cf. (2.24)), $\boldsymbol{F}$ is a function of polynomial order one with respect to the nodal displacements $\boldsymbol{u}$:

$$\boldsymbol{F} = \frac{\partial \bar{\boldsymbol{u}}}{\partial \bar{\boldsymbol{X}}} + \boldsymbol{I} \quad \Rightarrow \quad \boldsymbol{F} = \mathcal{O}(\boldsymbol{u}^1). \tag{11.3}$$

This relationship holds only for elements with nodal displacement dofs in three dimensions and rotational dofs in two dimensions. The polynomial order of the shape functions $\boldsymbol{N}$ is irrelevant, as long as the dofs are solely nodal displacements. On the contrary, if the displacements are expressed as three dimensional finite rotations, as it is common for shell elements, the mathematical structure depends on the parametrization of rotation and is more complicated.

The strain measure used in the Total Lagrangian framework is the Green-Lagrange strain, which is a quadratic strain measure. Thus, it is a quadratic function of the nodal displacements $\boldsymbol{u}$:

$$\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}\right) \quad \Rightarrow \quad \boldsymbol{E} = \mathcal{O}(\boldsymbol{u}^2). \tag{11.4}$$

The potential energy $\Pi$ of an elastic deformable body occupying the domain $\Omega_0$ exhibiting a St. Venant-Kirchhoff material can be expressed as a square function of the strain measure with the material properties gathered in the fourth order material tensor $\boldsymbol{C}_{SE}$ expressing the energy density function (2.15). It is given as

$$\Pi = \int_{\Omega_0} \boldsymbol{E} : \boldsymbol{C}_{SE} : \boldsymbol{E} \, \mathrm{d}\Omega_0 \quad \Rightarrow \quad \Pi = \mathcal{O}(\boldsymbol{u}^4). \tag{11.5}$$

As a consequence, the potential energy of the elastic deformation is a quartic function of the nodal displacements $\boldsymbol{u}$. Finally, the internal forces of a conservative system are the derivative of the negative potential with respect to the dofs yielding:

$$\boldsymbol{f}(\boldsymbol{u}) = -\frac{\partial \Pi}{\partial \boldsymbol{u}} \quad \Rightarrow \quad \boldsymbol{f} = \mathcal{O}(\boldsymbol{u}^3). \tag{11.6}$$

Consequently, the internal forces of any elastic system exhibiting a St. Venant-Kirchhoff material, which is discretized with a Galerkin scheme, are cubic polynomials with respect to the displacements $\boldsymbol{u}$. Since the displacements $\boldsymbol{u}$ are linear functions of the reduced generalized displacements $\boldsymbol{q}$, the resulting reduced equations of motion can be written in the form

$$\boldsymbol{M}_r\ddot{\boldsymbol{q}} + \boldsymbol{C}_r\dot{\boldsymbol{q}} + \boldsymbol{K}^{(1)}\boldsymbol{q} + \frac{1}{2}\boldsymbol{K}^{(2)}\boldsymbol{q}\boldsymbol{q} + \frac{1}{6}\boldsymbol{K}^{(3)}\boldsymbol{q}\boldsymbol{q}\boldsymbol{q} = \boldsymbol{g}, \tag{11.7}$$

where the vector multiplications with the higher order arrays $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ are multiplications with all axes of the tensor besides the first one. In index notation (11.7) yields

$$M_{r,ij}\ddot{q}_j + C_{r,ij}\dot{q}_j + K^{(1)}_{ij}q_j + \frac{1}{3}K^{(2)}_{ijk}q_jq_k + \frac{1}{6}K^{(3)}_{ijkl}q_jq_kq_l = g_i. \tag{11.8}$$

To summarize the derivation above: When the linear St. Venant-Kirchhoff material is used, the Taylor series (11.1) is completely converged after the third member. To accomplish the polynomial tensors hyper-reduction, the tensors $\boldsymbol{K}^{(1)}$, $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ have to be computed in an explicit form, so that the nonlinear force vector is computed using these tensors explicitly as in (11.7) instead of the classical reduced formulation (3.5), where the nonlinear forces are computed on the element level.

The polynomial tensors are derivatives of the elastic potential $\Pi$ with respect to the generalized coordinates $\boldsymbol{q}$. Thus, the tensors are fully symmetric with respect to its indices, i.e., the indices $i, j, k$ and $l$ are fully interchangeable. For $\boldsymbol{K}^{(1)}$ this means, that the stiffness matrix is symmetric ($K_{ij} = K_{ji}$), which is well known. For the higher order arrays, this holds also, as given exemplarily for $\boldsymbol{K}^{(2)}$:

$$K^{(2)}_{ijk} = K^{(2)}_{ikj} = K^{(2)}_{jik} = K^{(2)}_{jki} = K^{(2)}_{kij} = K^{(2)}_{kji} \quad \forall i, j, k \in \{1, \ldots, n\}. \tag{11.9}$$

The symmetry of the coefficient arrays is convenient for the computation as well as the storage and the multiplication to obtain the internal reduced forces. These topics are discussed in the following.

## 11.3 Computation of Coefficients

As shown in the previous section, the internal forces of a St. Venant-Kirchhoff material are cubic polynomials. To hyper-reduce a reduced basis system using polynomial tensors, $\boldsymbol{K}^{(1)}$, $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ have to be present in the explicit form. Hence, different strategies for obtaining the polynomial tensors of a given (reduced) system are discussed in the following subsections.

### 11.3.1 Direct Computation

The most accurate way of computing the tensors is the direct computation of the coefficients in the element formulation. This can be done by formulating the internal potential $\Pi_e$ of an element $e$ similar to (11.5) on the element level. The polynomial tensors are then first obtained on the element level by differentiating the potential $\Pi_e$ of the element multiple times:

$$\boldsymbol{K}_e^{(1)} = \frac{\partial^2 \Pi}{\partial \boldsymbol{u}_e^2}, \quad \boldsymbol{K}_e^{(2)} = \frac{\partial^3 \Pi}{\partial \boldsymbol{u}_e^3}, \quad \boldsymbol{K}_e^{(3)} = \frac{\partial^4 \Pi}{\partial \boldsymbol{u}_e^4}. \tag{11.10}$$

The elemental tensors are then assembled with the elemental localization matrix $\boldsymbol{L}_e$ and the reduced basis $\boldsymbol{V}$ to

$$\begin{aligned}
\boldsymbol{K}^{(1)} &= \sum_{e \in \mathcal{E}} \boldsymbol{V}^T \boldsymbol{L}_e^T \boldsymbol{K}_e^{(1)} \boldsymbol{L}_e \boldsymbol{V}, \\
\boldsymbol{K}^{(2)} &= \sum_{e \in \mathcal{E}} \boldsymbol{V}^T \boldsymbol{L}_e^T \left[ \boldsymbol{K}_e^{(2)} (\boldsymbol{L}_e \boldsymbol{V}) \right] : (\boldsymbol{L}_e \boldsymbol{V}), \\
\boldsymbol{K}^{(3)} &= \sum_{e \in \mathcal{E}} \boldsymbol{V}^T \boldsymbol{L}_e^T \left\{ \left[ \boldsymbol{K}_e^{(3)} (\boldsymbol{L}_e \boldsymbol{V}) \right] : (\boldsymbol{L}_e \boldsymbol{V}) \right\} : (\boldsymbol{L}_e \boldsymbol{V}),
\end{aligned} \tag{11.11}$$

or in index notation with implicit summation over indices as

$$\begin{aligned}
K_{ij}^{(1)} &= \sum_{e \in \mathcal{E}} (VL_e)_{ki} \, K_{e,kl}^{(1)} \, (VL_e)_{lj}, \\
K_{ijk}^{(2)} &= \sum_{e \in \mathcal{E}} (VL_e)_{li} \, K_{e,lmn}^{(2)} \, (VL_e)_{mj} \, (VL_e)_{nk}, \\
K_{ijkl}^{(3)} &= \sum_{e \in \mathcal{E}} (VL_e)_{mi} \, K_{e,mnop}^{(3)} \, (VL_e)_{nj} \, (VL_e)_{ok} \, (VL_e)_{pl}.
\end{aligned} \tag{11.12}$$

Both, the element formulation and the assembly have to be formulated and implemented for all elements in the element library of a finite element code. Especially for codes with many different element types this can be tedious task, since the terms in the formulation can become lengthy and cumbersome, see for instance Capell et al. [Cap+02]. In commercial packages, this feature is commonly not implemented. Hence, this approach is only attractive in cases where the source code is available and only few different elements are used. For instance, Barbič and James [BJ05] used this technique in computer graphics, since all geometries were rendered in voxels and hence only polynomial tensors for one element had to be formulated. As a consequence, so-called *non-intrusive* methods have been developed to circumvent the derivation and implementation efforts of this method.

### 11.3.2 Determining the Tensor Expansion by Numerical Differentiation

In most finite element codes, the computation of higher order derivatives of the internal forces is not implemented. Since in implicit time integrations solely the internal force vector $\boldsymbol{f}(\boldsymbol{u})$ and the tangential stiffness matrix $\boldsymbol{K}(\boldsymbol{u})$ is needed, only these two functions can

be evaluated analytically. This leads to the concept of non-intrusive methods where the coefficients $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ are obtained without having access to higher order derivatives.

The most straightforward way to compute the higher order derivatives $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ is the use of numerical differentiation using a finite difference scheme. When the tangential stiffness matrix $\boldsymbol{K}(\boldsymbol{u})$ is available in the code, the higher order derivatives can be computed as derivatives of the reduced tangential stiffness matrix $\boldsymbol{K}_r(\boldsymbol{q})$:

$$\boldsymbol{K}^{(2)} = \frac{\partial \boldsymbol{K}_r}{\partial \boldsymbol{q}}, \quad \boldsymbol{K}^{(3)} = \frac{\partial^2 \boldsymbol{K}_r}{\partial \boldsymbol{q}^2}. \tag{11.13}$$

When evaluating the partial derivatives with central finite differences, the higher order derivatives are:

$$\boldsymbol{K}^{(1)} = \boldsymbol{K}_r = \boldsymbol{V}^T \boldsymbol{K} \boldsymbol{V}, \tag{11.14}$$

$$\boldsymbol{K}^{(2)}_{..i} = \boldsymbol{V}^T \frac{\boldsymbol{K}(h\boldsymbol{v}_i) - \boldsymbol{K}(-h\boldsymbol{v}_i)}{2h} \boldsymbol{V}, \tag{11.15}$$

$$\boldsymbol{K}^{(3)}_{..ij} = \boldsymbol{V}^T \frac{\boldsymbol{K}(h\boldsymbol{v}_i + h\boldsymbol{v}_j) - \boldsymbol{K}(-h\boldsymbol{v}_i + h\boldsymbol{v}_j) - \boldsymbol{K}(h\boldsymbol{v}_i - h\boldsymbol{v}_j) + \boldsymbol{K}(-h\boldsymbol{v}_i - h\boldsymbol{v}_j)}{4h^2} \boldsymbol{V}. \tag{11.16}$$

The dots in the indices denote the row and column dofs of the right hand side matrix. The computation of the finite differences above requires multiple evaluations of the tangential stiffness matrix $\boldsymbol{K}(\boldsymbol{u})$ for different displacements. Since the higher oder derivatives forming the polynomial tensors are symmetric with respect to all indices, as discussed in Section 11.2, only the finite difference quotients (11.16) for $i \geq j$ have to be evaluated while the remaining cases $i < j$ can be restored using the symmetry of $\boldsymbol{K}^{(3)}$. With this technique, the computation of $\boldsymbol{K}^{(1)}$, $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ requires $n(n+1)/2 + n + 1$ evaluations of the tangential stiffness matrix with the dimension $n$ of the reduced system.

In Section 6.3, a finite difference scheme is also applied to the tangential stiffness matrix in order to obtain the static or modal derivatives. Consequently, the same considerations given there are also valid for the numerical differentiation to obtain $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$. Hence, the step width $h$ has to be chosen appropriately to obtain good accuracy. Thereby, the symmetry of $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ with respect to all indices can be used as an indicator of the accuracy making a proper adjustment of the step width $h$ possible. Furthermore, non-symmetric finite difference schemes like the forward or backward finite differences lack of accuracy, and hence they are not suitable for the computation of the higher order tensors. These techniques can be stabilized as, e.g., discussed by de Boer and van Keulen [DV00], however requiring additional implementation effort on the element level.

### 11.3.3 Determining the Tensor Expansion by Identification

Another class of techniques is based on the fact that the polynomial structure of the nonlinear forces is known. Hence the coefficients can be identified by multiple evaluations of the nonlinear forces for different displacements. To develop the strategy, recall the polynomial cubic structure of the nonlinear internal forces:

$$f_{r,i}(\boldsymbol{q}) = K^{(1)}_{ij} q_j + \frac{1}{2} K^{(2)}_{ijk} q_j q_k + \frac{1}{6} K^{(3)}_{ijkl} q_j q_k q_l. \tag{11.17}$$

The coefficients $K^{(1)}_{ij}$, $K^{(2)}_{ijk}$ and $K^{(3)}_{ijkl}$ can be computed by setting up algebraic systems of equations, see e.g. [Mig+13]. Therefore, the internal force vector has to be evaluated for different vectors $\boldsymbol{q}$. One systematic way to obtain the coefficients is the evaluation with unit vectors in the reduced space. Let $\boldsymbol{e}_j \in \mathbb{R}^n$ be a unit vector in the $j$-th direction, i.e., $\boldsymbol{e}_j$ is

the $j$-th column vector of the identity matrix $\boldsymbol{I} \in \mathbb{R}^{n\times n}$. Then, for three randomly chosen different displacement amplitudes $a_1$, $a_2$ and $a_3$, the internal force vector is evaluated for the displacement fields $a_1\boldsymbol{e}_j$, $a_2\boldsymbol{e}_j$ and $a_3\boldsymbol{e}_j$. In this case, no summation over repeated indices is assumed:

$$\begin{aligned} f_{r,i}(\boldsymbol{q} = a_1\boldsymbol{e}_j) &= a_1 K_{ij}^{(1)} + a_1^2\frac{1}{2}K_{ijj}^{(2)} + a_1^3\frac{1}{6}K_{ijjj}^{(3)}, \\ f_{r,i}(\boldsymbol{q} = a_2\boldsymbol{e}_j) &= a_2 K_{ij}^{(1)} + a_2^2\frac{1}{2}K_{ijj}^{(2)} + a_2^3\frac{1}{6}K_{ijjj}^{(3)}, \\ f_{r,i}(\boldsymbol{q} = a_3\boldsymbol{e}_j) &= a_3 K_{ij}^{(1)} + a_3^2\frac{1}{2}K_{ijj}^{(2)} + a_3^3\frac{1}{6}K_{ijjj}^{(3)}. \end{aligned} \tag{11.18}$$

The resulting system of equations allows determining the unknown coefficients $K_{ij}^{(1)}$, $K_{ijj}^{(2)}$ and $K_{ijjj}^{(3)}$, which are obtained in a subsequent step by solving the linear system of equations:

$$\begin{pmatrix} a_1 & \frac{a_1^2}{2} & \frac{a_1^3}{6} \\ a_2 & \frac{a_2^2}{2} & \frac{a_2^3}{6} \\ a_3 & \frac{a_3^2}{2} & \frac{a_3^3}{6} \end{pmatrix} \begin{pmatrix} K_{ij}^{(1)} \\ K_{ijj}^{(2)} \\ K_{ijjj}^{(3)} \end{pmatrix} = \begin{pmatrix} f_{r,i}(\boldsymbol{q} = a_1\boldsymbol{e}_j) \\ f_{r,i}(\boldsymbol{q} = a_2\boldsymbol{e}_j) \\ f_{r,i}(\boldsymbol{q} = a_3\boldsymbol{e}_j) \end{pmatrix}. \tag{11.19}$$

This procedure is performed $n$ times for $j = 1 \dots n$ to obtain all coefficients for all combinations for $i$ and $j$. In the next step, displacement fields composed from two unit displacement vectors $\boldsymbol{e}_j$ and $\boldsymbol{e}_k$ are addressed. They have to be evaluated for three different amplitudes $a_1$, $a_2$ and $a_3$ and are exemplarily given for $a_1$:

$$\begin{aligned} f_{r,i}(\boldsymbol{q} = a_1(\boldsymbol{e}_j + \boldsymbol{e}_k)) =& a_1\left(K_{ij}^{(1)} + K_{ik}^{(1)}\right) + a_1^2\frac{1}{2}\left(K_{ijj}^{(2)} + K_{ikk}^{(2)} + 2K_{ijk}^{(2)}\right) \\ &+ a_1^3\frac{1}{6}\left(K_{ijjj}^{(3)} + 3K_{ijjk}^{(3)} + 3K_{ijkk}^{(3)}\right). \end{aligned} \tag{11.20}$$

The equation above has three new unknowns $K_{ijk}^{(2)}$ and $K_{ijjk}^{(3)}$ and $K_{ijkk}^{(3)}$ which can be determined with three different amplitudes $a_1$, $a_2$ and $a_3$ and the coefficients obtained from the previous step (11.18). Note that in this step the symmetry is exploited, since $K_{ijjk} = K_{ijkj} = K_{ikjj}$ is summarized by $K_{ijjk}$. Consequently, only $3n(n-1)/2$ evaluations of (11.20) have to be performed, as only combinations with $k < j$ have to be evaluated and the other ones with $k > j$ are obtained using symmetry. In the last step, the displacement field with three different unit displacements $\boldsymbol{e}_j$, $\boldsymbol{e}_k$, $\boldsymbol{e}_l$ are evaluated yielding:

$$\begin{aligned} f_{r,i}(\boldsymbol{q} = a_1(\boldsymbol{e}_j + \boldsymbol{e}_k + \boldsymbol{e}_l)) =& a_1(K_{ij}^{(1)} + K_{ik}^{(1)} + K_{il}^{(1)}) \\ &+ a_1^2\frac{1}{2}(K_{ijj}^{(2)} + K_{ikk}^{(2)} + K_{ill}^{(2)} + 2K_{ijk}^{(2)} + 2K_{ijl}^{(2)} + 2K_{ikl}^{(2)}) \\ &+ a_1^3\frac{1}{6}(K_{ijjj}^{(3)} + 3K_{ijjk}^{(3)} + 3K_{ijkk}^{(3)} + 3K_{ijll}^{(3)} \\ &\quad + 3K_{iljj}^{(3)} + 3K_{iklI}^{(3)} + 3K_{ilkk}^{(3)} + 6K_{ijkl}^{(3)}), \end{aligned} \tag{11.21}$$

which can be directly solved for $K_{ijkl}$. The symmetry is also exploited here so that only $n(n-1)(n-2)/6$ cases for $j < k < l$ have to be evaluated. The other terms are obtained using the symmetry, as above.

The method described above evaluates the nonlinear internal forces. However, the same information can be obtained by evaluating the reduced tangential stiffness matrix $\boldsymbol{K}_r(\boldsymbol{q})$ for different displacement fields, see Perez et.al. [PWM14]. The reduced tangential stiffness matrix is a quadratic function:

$$K_{r,ij}(\boldsymbol{q}) = K_{ij}^{(1)} + K_{ijk}^{(2)} q_k + \frac{1}{2} K_{ijkl}^{(3)} q_k q_l. \tag{11.22}$$

Similarly, the coefficients can be obtained by evaluating the tangential stiffness matrix for the unit displacement with two amplitudes $a_1$ and $a_2$. As in (11.18), no summation over repeated indices is assumed:

$$\begin{aligned} K_{r,ij}(\boldsymbol{q} = a_1 \boldsymbol{e}_k) &= K_{ij}^{(1)} + a_1 K_{ijk}^{(2)} + a_1^2 \frac{1}{2} K_{ijkk}^{(3)}, \\ K_{r,ij}(\boldsymbol{q} = a_2 \boldsymbol{e}_k) &= K_{ij}^{(1)} + a_2 K_{ijk}^{(2)} + a_2^2 \frac{1}{2} K_{ijkk}^{(3)}. \end{aligned} \tag{11.23}$$

With the equations above, the unknown coefficients $K_{ijk}^{(2)}$ and $K_{ijkk}^{(3)}$ can be identified in the same fashion as in (11.19). Then, in a subsequent step, the coefficients with different indices can be obtained by solving

$$K_{r,ij}(\boldsymbol{q} = a_1(\boldsymbol{e}_k + \boldsymbol{e}_l)) = K_{ij}^{(1)} + a_1 \left( K_{ijk}^{(2)} + K_{ijl}^{(2)} \right) + a_1^2 \frac{1}{2} \left( K_{ijkk}^{(3)} + K_{ijll}^{(3)} + 2K_{ijkl}^{(3)} \right) \tag{11.24}$$

for the unknown coefficient $K_{ijkl}^{(3)}$. The identification technique using the stiffness matrix needs only $2n$ computations of the tangential stiffness matrix in order to determine terms with similar indices and $n(n-1)/2$ evaluations for the terms with different indices, since the symmetry can be exploited. As a consequence, the number of operations is clearly smaller compared to the force identification technique.

In both techniques described above, the choice of the amplitudes $a_1$, $a_2$ and $a_3$ for the unit displacements is to be specified. When the material of the underlying model is a St. Venant-Kirchhoff material which exhibits a third order polynomial structure for the internal forces, the choice of the parameters is arbitrary as long as they are sufficiently different from each other leading to invertible matrices in (11.19). However, the inversion of the matrix can be simplified, if the coefficients are chosen in a certain pattern like $a_1 = 1$, $a_2 = -1$ and $a_3 = 2$. Then, the solution of the system like (11.19) can be obtained directly by algebraic operations.

### 11.3.4 Other Approaches

All approaches presented above are approaches where the polynomial coefficients of a reduced system are computed and hence a hyper-reduction is performed. Thereby the reduction basis is chosen a priori and the coefficients are independent of the method used. However, for the sake of completeness it is mentioned, that a whole group of so-called force-based methods exist. In these methods, instead of prescribing the displacements as in the methods above for obtaining the nonlinear forces or tangential stiffness matrices, the external force vector is prescribed on the full model and the nonlinear problem is solved iteratively.

From the resulting displacement fields of the high dimensional, unreduced model, different approximations can be taken. In the so-called Implicit Condensation (IC) method proposed by Hollkamp et al. [HGS05], the nonlinear effects, which are not considered in the reduction basis, are accounted for in the polynomial tensors. In the extension of the IC method named Implicit Condensation and Expansion (ICE), see Kim et al. [Kim+13] and Mignolet et al. [Mig+13], the so-called dual modes similar to the SMDs introduced in Subsection 6.2.2 are identified empirically.

These methods require multiple nonlinear static solutions of the full model. Furthermore, they require a careful selection of the forcing amplitudes, which have to be manually controlled on certain points on the mesh requiring elaborate preprocessing, see

Guerin et al. [GKA16]. These methods are attractive in scenarios where the solutions of the nonlinear problem are used for both, the basis construction and the identification of the polynomial coefficients. These methods have mainly been applied to flat or slightly curved shell and beam problems [GKA16; KA16; Wan+09; Mig+13] as well as more elaborate airplane structures [Phl+14] with moderate rotations. How they perform for arbitrary geometries exhibiting large rotations is still an open question.

## 11.4 Efficient Treatment of Symmetric Arrays

The multidimensional arrays $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ are highly symmetric. Hence, if they are stored and treated naively as regular dense high dimensional arrays, a remarkable amount of storage is used redundantly. Especially for $\boldsymbol{K}^{(3)}$, the amount of RAM can increase quickly when the basis becomes larger. As an example, for a basis of $n = 20$, $\boldsymbol{K}^{(3)}$ requires 1.28 MB of storage, if 64 bit float numbers with double precision are used. However, when only the unique values are stored, this number reduces to 0.071 MB. This ratio becomes even more dramatic, if $n$ becomes larger. For instance for $n = 50$, the naive storage of $\boldsymbol{K}^{(3)}$ requires 50 MB vs. 2.34 MB in efficient storage or 800 MB vs. 35.4 MB for $n = 100$. Thus, efficient storage of the higher order tensors is necessary, if the memory is limited.

However, not only for storage reasons, a special treatment of the high dimensional arrays is necessary. The computation of the internal forces $\boldsymbol{f}_r$ and the tangential stiffness matrix $\boldsymbol{K}_r(\boldsymbol{q})$ become extremely costly for growing reduction dimension $n$, since the number of multiplications and hence for floating point operations grows with order $n^4$. This makes the polynomial multiplication unattractive for larger $n$, if it is performed directly as given in (11.8). So not only an efficient storage strategy, but also an efficient multiplication strategy is necessary to achieve good speedup factors for polynomial tensor hyper-reduced models.

### 11.4.1 Efficient Storage

First, the efficient storage of $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ is addressed. Since both multidimensional arrays are symmetric with respect to all axes, it is sufficient to store entries with one index combination only once. Hence, for efficient storage, only the unique, non-redundant entries of $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ are stored in the vectors $\boldsymbol{k}^{(2)}$ and $\boldsymbol{k}^{(3)}$ in a prescribed sequence:

$$\begin{aligned} \boldsymbol{k}^{(2)} = \Bigg( &\left(\boldsymbol{k}_{iii}^{(2)}\right)^T, \left(\boldsymbol{k}_{iij}^{(2)}\right)^T, \left(\boldsymbol{k}_{ijj}^{(2)}\right)^T, \left(\boldsymbol{k}_{ijk}^{(2)}\right)^T \Bigg)^T \\ &\forall \quad \{i,j,k \in \{1,\dots,n\} \mid i > j > k\}, \end{aligned} \tag{11.25}$$

$$\begin{aligned} \boldsymbol{k}^{(3)} = \Bigg( &\left(\boldsymbol{k}_{iiii}^{(3)}\right)^T, \left(\boldsymbol{k}_{iiij}^{(2)}\right)^T, \left(\boldsymbol{k}_{iijj}^{(3)}\right)^T, \left(\boldsymbol{k}_{ijjj}^{(3)}\right)^T, \left(\boldsymbol{k}_{iijk}^{(3)}\right)^T, \left(\boldsymbol{k}_{ijjk}^{(3)}\right)^T, \left(\boldsymbol{k}_{ijkk}^{(3)}\right)^T, \left(\boldsymbol{k}_{ijkl}^{(3)}\right)^T \Bigg)^T \\ &\forall \quad \{i,j,k,l \in \{1,\dots,n\} \mid i > j > k > l)\}. \end{aligned} \tag{11.26}$$

The vectors $\boldsymbol{k}^{(2)} \in \mathbb{R}^{n(n+1)(n+2)/6}$ and $\boldsymbol{k}^{(3)} \in \mathbb{R}^{n(n+1)(n+2)(n+3)/24}$ are column vectors containing all entries corresponding to the index combination of the higher dimensional arrays. For instance, $\boldsymbol{k}_{iii}^{(2)}$ contains all diagonal entries of $\boldsymbol{K}^{(2)}$, i.e., $\boldsymbol{k}_{iii}^{(2)} = \left(K_{111}^{(2)}, \dots, K_{nnn}^{(2)}\right)$. Similarly, the vectors containing combinations of different indices are constructed with the first index $i$ changing slowest and $l$ changing fastest like in nested loops. For a deeper explanation, the construction of the vector $\boldsymbol{k}^{(2)}$ is exemplarily given in Algorithm 2. The construction of $\boldsymbol{k}^{(3)}$, which is more lengthy, follows the same principle.

With the construction of the vectors $\boldsymbol{k}^{(2)}$ and $\boldsymbol{k}^{(3)}$, the memory consumption of storing the polynomial tensors is reduced considerably. This is, however, not the only benefit. Since the tensors are used in every evaluation of the internal forces and the tangential stiffness matrix, the memory needs to be addressed leading to slow multiplication routines, if a large block of memory needs to be addressed. Consequently, the reduction of memory space by transforming $\boldsymbol{K}^{(2)}$ into $\boldsymbol{k}^{(2)}$ and $\boldsymbol{K}^{(3)}$ into $\boldsymbol{k}^{(2)}$ can also reduce the time for the multiplication. However, besides that, even more computational savings can be achieved with an efficient multiplication technique.

**Algorithm 2:** Build the unique vector $\boldsymbol{k}^{(2)}$ from the highly redundant symmetric array $\boldsymbol{K}^{(2)}$

**Input** : Symmetric tensor $\boldsymbol{K}^{(2)} \in \mathbb{R}^{n \times n \times n}$
**Output** : Vector of unique tensor entries $\boldsymbol{k}^{(2)} \in \mathbb{R}^{(n^3+3n^2+2n)/6}$

```
c := 0                                                  // Initialize counter c
k^(2) ∈ R^((n^3+3n^2+2n)/6)                             // initialize k^(2)
for i := 0 : n                                          // Equal indices: iii
 do
    k^(2)[c] := K^(2)[i,i,i]
    c := c + 1
end
for i := 0 : n                                          // Two different indices: iij
 do
    for j := 0 : i − 1 do
        k^(2)[c] := K^(2)[i,i,j]
        c := c + 1
    end
end
for i := 0 : n                                          // Two different indices: ijj
 do
    for j := 0 : i − 1 do
        k^(2)[c] := K^(2)[i,j,j]
        c := c + 1
    end
end
for i := 0 : n                                          // Three different indices: ijk
 do
    for j := 0 : i − 1 do
        for k := 0 : j − 1 do
            k^(2)[c] := K^(2)[i,j,k]
            c := c + 1
        end
    end
end
```

### 11.4.2 Efficient Multiplication

In an implicit time integration, the internal forces $\boldsymbol{f}_r(\boldsymbol{q})$ and the tangential stiffness matrix $\boldsymbol{K}_r(\boldsymbol{q})$ are needed. In the polynomial setup for St. Venant-Kirchhoff materials, these two

**Algorithm 3:** Build the intermediate matrix $\bar{\boldsymbol{K}}^{(2)}$ using the unique vector $\boldsymbol{k}^{(2)}$

**Input :** Vector of unique tensor entries $\boldsymbol{k}^{(2)} \in \mathbb{R}^{(n^3+3n^2+2n)/6}$ representing $\boldsymbol{K}^{(2)} \in \mathbb{R}^{n\times n\times n}$, reduced vector of generalized coordinates $\boldsymbol{q} \in \mathbb{R}^n$

**Output :** Intermediate matrix $\bar{\boldsymbol{K}}^{(2)}$ equivalent to $\boldsymbol{K}^{(2)}\boldsymbol{q}$

$c := 0$ // Initialize counter $c$
$\bar{\boldsymbol{K}}^{(2)} := \boldsymbol{0} \in \mathbb{R}^{n\times n}$ // initialize $\bar{K}^{(2)}$ to zero
**for** $i := 0 : n$ // Equal indices: *iii*
**do**
  $\bar{\boldsymbol{K}}^{(2)}[i,i] := \boldsymbol{k}^{(2)}[c]\boldsymbol{q}[i]$
  $c := c + 1$
**end**
**for** $i := 0 : n$ // Two different indices: *iij*
**do**
  **for** $j := 0 : i - 1$ **do**
    $\bar{\boldsymbol{K}}^{(2)}[i,i] := \bar{\boldsymbol{K}}^{(2)}[i,i] + \boldsymbol{k}^{(2)}[c]\boldsymbol{q}[j]$
    $\bar{\boldsymbol{K}}^{(2)}[i,j] := \bar{\boldsymbol{K}}^{(2)}[i,j] + \boldsymbol{k}^{(2)}[c]\boldsymbol{q}[i]$
    $c := c + 1$
  **end**
**end**
**for** $i := 0 : n$ // Two different indices: *ijj*
**do**
  **for** $j := 0 : i - 1$ **do**
    $\bar{\boldsymbol{K}}^{(2)}[j,j] := \bar{\boldsymbol{K}}^{(2)}[j,j] + \boldsymbol{k}^{(2)}[c]\boldsymbol{q}[i]$
    $\bar{\boldsymbol{K}}^{(2)}[i,j] := \bar{\boldsymbol{K}}^{(2)}[i,j] + \boldsymbol{k}^{(2)}[c]\boldsymbol{q}[j]$
    $c := c + 1$
  **end**
**end**
**for** $i := 0 : n$ // Three different indices: *ijk*
**do**
  **for** $j := 0 : i - 1$ **do**
    **for** $k := 0 : j - 1$ **do**
      $\bar{\boldsymbol{K}}^{(2)}[i,j] := \bar{\boldsymbol{K}}^{(2)}[i,j] + \boldsymbol{k}^{(2)}[c]\boldsymbol{q}[k]$
      $\bar{\boldsymbol{K}}^{(2)}[i,k] := \bar{\boldsymbol{K}}^{(2)}[i,k] + \boldsymbol{k}^{(2)}[c]\boldsymbol{q}[j]$
      $\bar{\boldsymbol{K}}^{(2)}[j,k] := \bar{\boldsymbol{K}}^{(2)}[j,k] + \boldsymbol{k}^{(2)}[c]\boldsymbol{q}[i]$
      $c := c + 1$
    **end**
  **end**
**end**
**for** $i := 0 : n$ // Fill lower half of matrix
**do**
  **for** $j := 0 : i - 1$ **do**
    $\bar{\boldsymbol{K}}^{(2)}[j,i] := \bar{\boldsymbol{K}}^{(2)}[i,j]$
  **end**
**end**

quantities are given as

$$\boldsymbol{f}_r(\boldsymbol{q}) = \boldsymbol{K}^{(1)}\boldsymbol{q} + \frac{1}{2}\boldsymbol{K}^{(2)}\boldsymbol{q}\boldsymbol{q} + \frac{1}{6}\boldsymbol{K}^{(3)}\boldsymbol{q}\boldsymbol{q}\boldsymbol{q} \tag{11.27}$$

$$\boldsymbol{K}_r(\boldsymbol{q}) = \boldsymbol{K}^{(1)} + \boldsymbol{K}^{(2)}\boldsymbol{q} + \frac{1}{2}\boldsymbol{K}^{(3)}\boldsymbol{q}\boldsymbol{q}. \tag{11.28}$$

If both quantities need to be computed at the same time, it is most efficient to precompute the matrices $\bar{\boldsymbol{K}}^{(2)}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ and $\bar{\boldsymbol{K}}^{(3)}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ as intermediate values defined as:

$$\bar{\boldsymbol{K}}^{(2)} = \boldsymbol{K}^{(2)}\boldsymbol{q}, \qquad \Leftrightarrow \qquad \bar{K}^{(2)}_{ij} = K^{(2)}_{ijk}q_k, \tag{11.29}$$

$$\bar{\boldsymbol{K}}^{(3)} = \boldsymbol{K}^{(3)}\boldsymbol{q}\boldsymbol{q}, \qquad \Leftrightarrow \qquad \bar{K}^{(3)}_{ij} = K^{(3)}_{ijkl}q_k q_l. \tag{11.30}$$

Then $\boldsymbol{f}_r(\boldsymbol{q})$ and $\boldsymbol{K}_r(\boldsymbol{q})$ are computed using $\bar{\boldsymbol{K}}^{(2)}$ and $\bar{\boldsymbol{K}}^{(3)}$:

$$\boldsymbol{f}_r(\boldsymbol{q}) = \left(\boldsymbol{K}^{(1)} + \frac{1}{2}\bar{\boldsymbol{K}}^{(2)} + \frac{1}{6}\bar{\boldsymbol{K}}^{(3)}\right)\boldsymbol{q}, \tag{11.31}$$

$$\boldsymbol{K}_r(\boldsymbol{q}) = \boldsymbol{K}^{(1)} + \bar{\boldsymbol{K}}^{(2)} + \frac{1}{2}\bar{\boldsymbol{K}}^{(3)}. \tag{11.32}$$

The question for an efficient multiplication is consequently the fast and efficient construction of the two intermediate matrices $\bar{\boldsymbol{K}}^{(2)}$ and $\bar{\boldsymbol{K}}^{(3)}$. For a fast construction, only the necessary multiplications should be performed.

For the computation of $\bar{\boldsymbol{K}}^{(2)}$, the symmetry can be exploited to reduce the number of multiplications. The direct multiplication $K^{(2)}_{ijk}q_k$ requires $n^3$ multiplications, since every entry in the resulting matrix $\bar{K}^{(2)}_{ij}$ needs $n$ multiplications over the last index $k$. However, since $\bar{K}^{(2)}_{ij}$ is symmetric, it is sufficient to perform the multiplications only for the lower half of the matrix, i.e., $i \geq j$, so that only the $n(n+1)/2$ unique entries of $\bar{\boldsymbol{K}}^{(2)}$ are filled. The upper half of $\bar{\boldsymbol{K}}^{(2)}$ is then reconstructed from the lower half with copy operations. As a consequence, this accelerated multiplication requires only $n^2(n+1)/2$ instead of $n^3$ multiplication operations.

Similarly, the idea can be used for the computation of $\bar{\boldsymbol{K}}^{(3)}$. There, two symmetries can be exploited for the efficient multiplication of $\bar{K}^{(3)}_{ij} = K^{(3)}_{ijkl}q_k q_l$. The direct approach requires $n^4$ multiplications, since every index goes from 1 to $n$. However, the same idea for the efficient multiplication of $\bar{\boldsymbol{K}}^{(2)}$ can be used. Since also the resulting matrix $\bar{K}^{(3)}_{ij}$ is symmetric, only the entries with $i \geq j$ are computed and the remaining entries with $i < j$ are filled with copy operations from the known ones. So for the first indices only $n(n+1)/2$ instead of $n^2$ entries need to be filled.

The multiplication of the last two entries in $K^{(3)}_{ijkl}q_k q_l$ follows the same pattern. In the direct multiplication approach, there exist $n^2$ combinations of $q_k$ and $q_l$. However, since the multiplication is commutative, it is sufficient to perform the multiplication only for $k \geq l$ similar to the first two indices. Then, only $n(n+1)/2$ multiplications for the indices $k$ and $l$ are necessary for every *i-j* index pair. The multiplications for the indices $k < l$ can be considered by multiplying the corresponding combination with $k > l$ with a factor of 2. As a consequence, the full multiplication $K^{(3)}_{ijkl}q_k q_l$ can be performed with $(n(n+1)/2)^2 = n^2(n+1)^2/4$ multiplication operations instead of $n^4$ operations for the direct approach.

One example of the efficient multiplication of $\bar{\boldsymbol{K}}^{(2)}$ is given in Algorithm 3. Note, that in this algorithm both the efficient storage of the higher order array and the efficient multiplication are considered. With the combination of both the compact storage

and the accelerated multiplication the computational effort can be clearly reduced. On the one hand, the efficient multiplication requires less floating point operations to obtain the results, as discussed above. On the other hand, the efficient storage of the higher order tensors minimizes the memory reading effort, since the values are called subsequently and are not scattered in the memory. Since both effects promise to clearly reduce the computational effort, this technique is applied to an academic but insightful example in the subsequent section. An overview of the polynomial tensors hyper-reduction technique using the accelerated multiplication is given in Figure 11.1. Note, that in the overview and in the numerical experiments, first the symmetric higher order tensors $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ are computed before they are transformed to the lean vectorial representation $\boldsymbol{k}^{(2)}$ and $\boldsymbol{k}^{(3)}$. It is, however, possible to store the results of the coefficient computation like the finite difference scheme or the identification directly into $\boldsymbol{k}^{(2)}$ and $\boldsymbol{k}^{(3)}$ without the detour over $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$. This enhancement which requires more sophisticated indexing is specially important for larger reduced models, since the size of $\boldsymbol{K}^{(2)}$ and especially $\boldsymbol{K}^{(3)}$ becomes quickly prohibitive for larger $n$.

## 11.5 Applications

The following applications are run on the cantilever example introduced in Section 6.4. The reduced basis model of the cantilever is constructed with a simulation-free basis built with five vibration modes and all corresponding SMDs yielding a reduced model of dimension $n = 20$. The parameters of the model and for the time integration are equal to the experiments in Section 6.4.

### 11.5.1 Comparison of Identification Techniques

First, the different methods for computing the polynomial tensors are compared. Therefore, a relative error measure similar to the relative error given in (3.9) is used. With the error measure $\epsilon_{rel}$, the multidimensional arrays $\boldsymbol{K}^{(2)}$ and $\widetilde{\boldsymbol{K}}^{(2)}$, or $\boldsymbol{K}^{(3)}$ and $\widetilde{\boldsymbol{K}}^{(3)}$ respectively, are compared. The tensors $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ are the reference tensors and $\widetilde{\boldsymbol{K}}^{(2)}$ and $\widetilde{\boldsymbol{K}}^{(3)}$ are the tensors to be measured against the reference:

$$\begin{aligned}
\epsilon_{rel} &= \frac{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\left(K_{ijk}^{(2)} - \widetilde{K}_{ijk}^{(2)}\right)^2}}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\left(K_{ijk}^{(2)}\right)^2}}, \\
\epsilon_{rel} &= \frac{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\left(K_{ijkl}^{(3)} - \widetilde{K}_{ijkl}^{(3)}\right)^2}}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\left(K_{ijkl}^{(3)}\right)^2}}.
\end{aligned} \tag{11.33}$$

The computation of $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ via finite differences, as shown in Subsection 11.3.2, requires the choice of a proper step width $h$. Similar to the computation of the quadratic tensor $\boldsymbol{\Theta}$ in Section 6.3, the symmetry can be used to control the error, as the higher oder tensors should remain the same when the indices are swapped. The symmetry of the higher order tensors is assessed with the relative error $\epsilon_{rel}$ given above. Thereby, the tensors $\widetilde{\boldsymbol{K}}^{(2)}$ and $\widetilde{\boldsymbol{K}}^{(3)}$ are equal to $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$, however with swapped indices:

$$\widetilde{K}_{ijk}^{(2)} = K_{ikj}^{(2)}, \qquad \widetilde{K}_{ijkl}^{(3)} = K_{ikjl}^{(3)}. \tag{11.34}$$

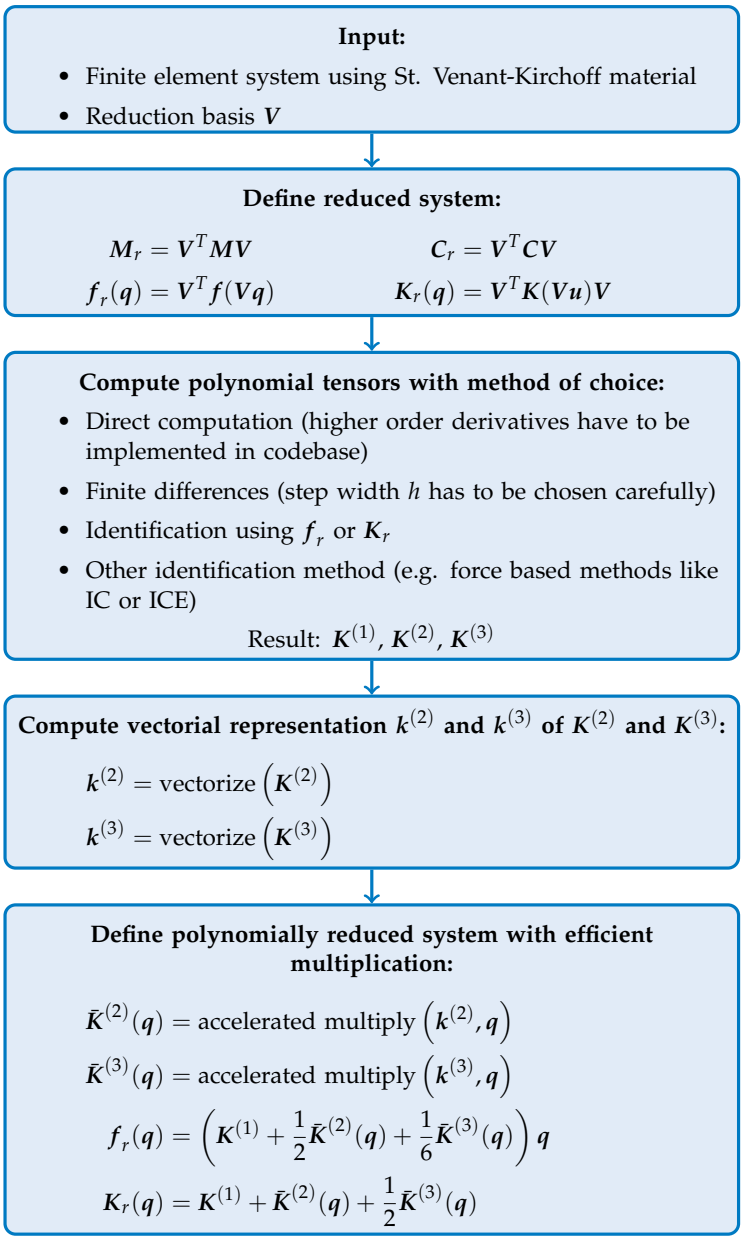


**Figure 11.1:** Overview of the reduction process using polynomial tensors and accelerated multiplication.

The relative error $\epsilon_{rel}$ for assessing the symmetry is evaluated with $\widetilde{K}^{(2)}$ and $\widetilde{K}^{(3)}$ given in (11.34) for the cantilever example for different step width $h$, see Figure 11.2. The relative symmetry error has its minimum at the step width $h \approx 1$ similar to the finite difference scheme used for $\boldsymbol{\Theta}$. The error is in the range of machine epsilon indicating, that the finite difference scheme is very exact for the correct step width. The error for the tensor $K^{(2)}$ responsible for the quadratic contributions in the forces rises after this optimal step width. The error for $K^{(3)}$ responsible for the cubic forces, however, remains in the order of machine epsilon. This is due to the fact, that for larger step width, the secant error rises, whereas for small step width the round-off errors dominate, see Griewank and Walter [GW08]. Since the tangential stiffness matrix is a purely quadratic function, the secant error is zero for the second order derivative, since the central difference scheme is of second order accuracy (cf. [Str07, pp. 15]). Consequently, a larger step width $h \geq 1$ is

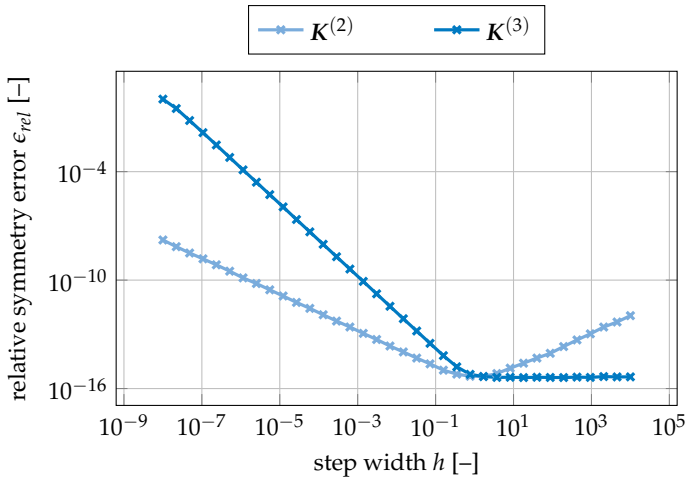


**Figure 11.2:** Symmetry error of the higher order polynomial tensors computed with finite difference scheme for different step widths $h$ for the cantilever example. The optimal step width is in the range of $h \approx 1$.

appropriate for the correct computation of the cubic stiffness part.

Next, the accuracy of the different methods to compute the polynomial tensors are investigated. Therefore, the finite difference scheme with $h = 1$, the identification method using the nonlinear forces (ID $\boldsymbol{f}$) and the identification method using the tangential stiffness matrix (ID $\boldsymbol{K}$) are used to compute $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$. They are compared against each other using the relative error (11.33) yielding the cross errors depicted in Table 11.1. All errors are in the range of $10^{-8}$ except the diagonal terms of the cross error, which are by definition zero, since equal tensors are compared in this case. The range of the error indicates, that all three investigated methods are of equal accuracy. Consequently, the choice of the method for computing the higher order tensors is up to the user. However, since all three methods take different approaches, the offline costs for computing $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ are of interest.

The offline costs for the three different methods finite differences, identification using $\boldsymbol{f}$ and identification using $\boldsymbol{K}$ are measured for different reduction orders $n$. The wall time for all three methods are given in Figure 11.3 using the same Intel Xeon Machine working at 4.0 GHz with 32 GB RAM as in the previous experiments. The offline costs for the identification method using $\boldsymbol{K}$ are the lowest, as expected. Since this method needs only $(n^2 + 3n)/2 + 1$ evaluations of the tangential stiffness matrix, it is more efficient than the finite difference scheme needing $4n(n+1) + n + 1$ evaluations of $\boldsymbol{K}_r(\boldsymbol{q})$. However, the finite differences are more efficient for larger $n$ than the identification using the internal forces, since it requires $n(n^2 + 6n - 1)/6$ evaluations of $\boldsymbol{f}_r(\boldsymbol{q})$. The higher polynomial order of the force identification method is particularly slow, when reduced models of higher order are reduced. Then, as expected, the methods using the tangential stiffness matrix are in clear favor.

To assess both the speedup and the accuracy of the polynomial hyper-reduction, the cantilever example and the reduction basis used above are used. To recall, the reduction basis is formed of five vibration modes plus the corresponding SMDs yielding a basis of $n = 20$ reduced dofs.

A time integration of the full system, the reduced system and a polynomial tensors hyper-reduced system is run with the equal settings from the example in Section 6.4 given in Table B.1. Since two multiplication schemes, the direct multiplication using the full tensors and the accelerated multiplication using the minimal representation $\boldsymbol{k}^{(2)}$ and $\boldsymbol{k}^{(3)}$ are

**Table 11.1:** Relative errors ($RE$) between the different methods for computing $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$. The finite difference scheme is performed with $h = 1$.

$\boldsymbol{K}^{(2)}$:

| | finite differences | identification $\boldsymbol{K}$ | identification $\boldsymbol{f}$ |
|---|---|---|---|
| finite differences | 0.0 | $2.38 \cdot 10^{-8}$ | $4.75 \cdot 10^{-8}$ |
| identification $\boldsymbol{K}$ | $2.38 \cdot 10^{-8}$ | 0.0 | $4.80 \cdot 10^{-8}$ |
| identification $\boldsymbol{f}$ | $4.75 \cdot 10^{-8}$ | $4.80 \cdot 10^{-8}$ | 0.0 |

$\boldsymbol{K}^{(3)}$:

| | finite differences | identification $\boldsymbol{K}$ | identification $\boldsymbol{f}$ |
|---|---|---|---|
| finite differences | 0.0 | $3.62 \cdot 10^{-8}$ | $7.53 \cdot 10^{-8}$ |
| identification $\boldsymbol{K}$ | $3.62 \cdot 10^{-8}$ | 0.0 | $7.63 \cdot 10^{-8}$ |
| identification $\boldsymbol{f}$ | $7.53 \cdot 10^{-8}$ | $7.63 \cdot 10^{-8}$ | 0.0 |

proposed in Subsection 11.4.2, a system with each method is run. Table 11.2 shows the wall times, the full relative error $RE_f$ and the hyper-reduction error $RE_{hr}$ of the time integrations carried out. The hyper-reduction error $RE_{hr}$ is in the range of $10^{-5}$ %. Since this order of magnitude is in the range of the tolerances of the integration scheme, both the reduced system and the polynomial hyper-reduced system are equal up to tiny numerical differences. The accumulation of tiny round-off errors are also the explanation why the different multiplication techniques yield different $RE_{hr}$ values in a very low regime. The full error $RE_f$ of the experiments stem solely from the reduction basis. However, the polynomially reduced system yields dramatically greater speedups compared to the reduced basis system. To highlight the speed especially of the polynomial system with accelerated multiplication, the wall time for the linearized system is also given. It can be expected, that the main part of time integration resources of the linear system are devoted to overhead costs like writing the output of the current displacements, which occur independently of the mathematical operations in the time integration scheme.

Since the polynomials systems exhibit such great speedups, the run time of this hyper-reduction technique is investigated further in the next section.

### 11.5.2 Accelerated Multiplication

According to Table 11.1, the accelerated multiplication described in Subsection 11.4.2 yields remarkable speedups compared to the direct approach, where the tensors are multiplied ignoring the symmetry. In the computational framework AMFE used for all experiments, the direct multiplication is performed in Python using the LAPACK routines written in Fortran. For the fast execution of the accelerated multiplication, a compiled function has to be run in order to achieve comparable speed. In the implementation used in this section, the algorithms are implemented in *Numba* (cf. [Oli12] and [LPS15]), which is based on the Low-Level-Virtual-Machine (LLVM) compilation technique. In many benchmarks, Numba achieves execution speed similar to compiled implementations, see Vanderplaas [Vana; Vanb]), so that the comparisons of the speed are fair.

To assess the computational speedup when using the accelerated multiplication, the computation of the internal force vector and the tangential stiffness matrix is measured for different reduction orders. Thereby, the direct approach, where the internal forces and the tangential stiffness matrix are computed according to (11.27) and (11.28), is compared against the accelerated multiplication given in Subsection 11.4.2 and illustrated in Figure 11.1. Figure 11.4 depicts the wall time taken for 10,000 evaluations of $\boldsymbol{K}_r(\boldsymbol{q})$ and

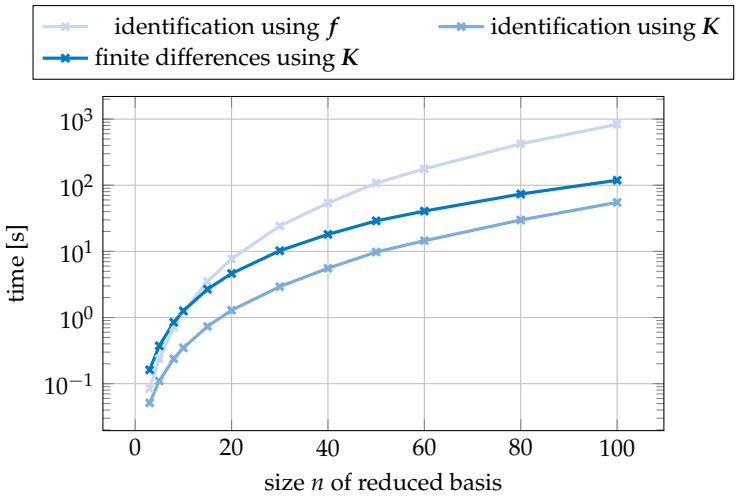


**Figure 11.3:** Wall time for computing $\boldsymbol{K}^{(1)}$, $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ using different methods.

$\boldsymbol{f}_r(\boldsymbol{q})$ using these two methods. For any reduction size $n$, the accelerated multiplication outperforms the direct multiplication. Probably due to some overhead costs in the testing routine, the computational time increases barely for reduced orders $n$ up to ten. With increasing $n$, the direct method consumes clearly more time than the accelerated multiplication routine in the range of over an oder of magnitude. As a consequence, the multiplication using the accelerated method is better for any reduction order.

Next, the runtime for full simulation runs is investigated. Therefore a POD basis is built to enable reduced bases of arbitrary size, which are physically reasonable and do not cause unwanted effects like different convergence rates in the Newton-Raphson scheme. Then, a full simulation, a reduced simulation, a polynomial tensors hyper-reduced simulation with direct multiplication and a polynomial tensors hyper-reduced simulation with accelerated multiplication are run for different reduction orders $n$. The wall times of the time integrations are depicted in Figure 11.5. The accelerated multiplication, as expected, is faster than the direct multiplication for all reduced dimensions. This is in accordance with the observations of the previous experiment, where the accelerated multiplication was also faster in any case. Furthermore, Figure 11.5 illustrates that the hyper-reduction effect of the polynomial tensor technique is only effective for small $n$. With the accelerated multiplication technique, the hyper-reduced model breaks even with the reduced model for $n = 80$. For greater $n$, the hyper-reduced model is slower than the reduced model. Even worse, for $n$ larger than 100, the original model is faster than the hyper-reduced model with polynomial tensors. When using the direct multiplication, the break even point of the hyper-reduced model versus the reduced basis model is even below $n = 60$ and versus the full model below $n = 80$.

Even though the benchmark example is of academic nature and has a small mesh with 246 elements and 1224 dofs, the weak point of the polynomial tensors hyper-reduction becomes obvious. While the technique is very efficient and fast for a small number of generalized coordinates, it becomes more and more inefficient for larger numbers. This is based in the polynomial nature of the tensor multiplication, where the computation of the nonlinear forces and the tangential stiffness matrix is associated with $\mathcal{O}(n^4)$. Even though the evaluation of the nonlinearity is not associated with the original dimension $N$ of the high dimensional model, it becomes prohibitive for reduced models with larger reduced dimensions.

**Table 11.2:** Wall times and errors of the polynomial-tensor hyper-reduced cantilever using the simulation-free basis of 5 vibration modes plus all corresponding SDs. The linearized run is given for comparison

| **Method** | **dofs** | **elements** | $RE_f$ [%] | $RE_{hr}$ [%] | $t_w$ [s] | **speedup** [−] |
|---|---|---|---|---|---|---|
| full | 1224 | 246 | – | – | 97.02 | 1.00 |
| reduced basis | 20 | 246 | 1.42 | – | 45.92 | 2.11 |
| poly direct mult | 20 | – | 1.42 | $1.50 \cdot 10^{-5}$ | 9.66 | 10.04 |
| poly fast mult | 20 | – | 1.42 | $4.84 \cdot 10^{-5}$ | 4.46 | 21.75 |
| linearized | 1224 | 246 | 123.35 | – | 1.95 | 49.75 |

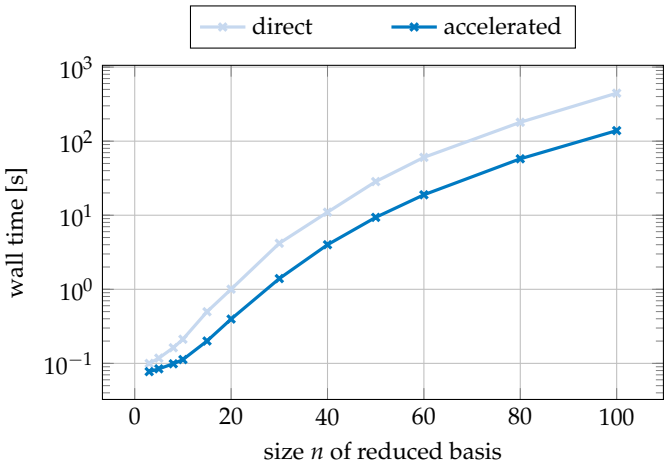


**Figure 11.4:** Wall time for 10,000 evaluations of the nonlinear force vector $\boldsymbol{f}(\boldsymbol{q})$ and tangential stiffness matrix $\boldsymbol{K}(\boldsymbol{q})$ with direct and accelerated multiplication method for different reduction orders $n$.

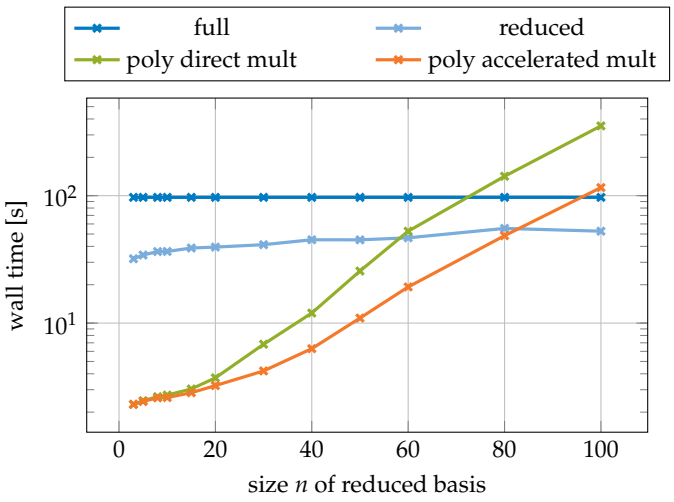


**Figure 11.5:** Wall times for the time integration of the cantilever example for different reduction orders $n$ using a POD basis.

# Chapter 12

# Discrete Empirical Interpolation Method (DEIM)

This chapter discusses the Discrete Empirical Interpolation Method (DEIM), a hyper-reduction method to approximate the high dimensional nonlinear force vector using an empirical force basis. In contrary to the polynomial tensors hyper-reduction discussed in the previous section, the reduction is achieved by an evaluation of the nonlinear forces only at selected points of the mesh.

The key idea of the DEIM goes back at least to the gappy POD method proposed in Everson and Sirovich [ES95], where facial images are reconstructed using "gappy", partial data. This idea is used in several approaches to hyper-reduce nonlinear dynamical systems such as fluid dynamics, e.g., in Bui et al. [BDW03], Willcox [Wil06] and Astrid et al. [Ast+08], or process simulations in Astrid [Ast04].

The Empirical Interpolation Method (EIM) proposed by Barrault et al. [Bar+04] and in a slightly different version by Ryckelynck [Ryc05] uses the same idea, however at the continuous level. It can be seen as the underlying theoretical framework of the Discrete Empirical Interpolation Method (DEIM) proposed by Chaturantabut et al. [CS10]. This method distinguishes itself from the others by the ease of computation, since a greedy algorithm is used for the selection of the gappy data points, the collocation points. Approaches similar to the DEIM are the best point interpolation method of Nguyen et al. [NPP08] and the method proposed by Nguyen and Peraire [NP08].

Further improvements for the DEIM were proposed, as to mention a few, the adaptation to finite element systems by Tiso and Rixen [TR13] called unassembled DEIM (UDEIM), a localized DEIM approach proposed by Peherstorfer et al. [Peh+14], strategies to handle parametric systems as proposed in Antil et al. [AHS14] as well as improved selection strategies in Drmac and Gugercin [DG16]. Adaptations to parametric systems are discussed in the overview paper on parametric model order reduction of Benner et al [BGW15]. Radermacher and Reese [RR16] applied the DEIM to geometrically and material nonlinear static systems.

## 12.1 Key Idea: Interpolation and Collocation

The idea of the DEIM is to approximate the high dimensional nonlinear force $\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q})$ with a force basis $\boldsymbol{U}_f$, similar to approximation of the full displacement field $\boldsymbol{u}$ using the reduced basis $\boldsymbol{V}$ (cf. Chapter 3). Hence, $\boldsymbol{f}$ is approximated as linear combination of the column vectors of the force basis $\boldsymbol{U}_f \in \mathbb{R}^{N\times m}$ with the time dependent amplitudes collected in $\boldsymbol{c}(\boldsymbol{V}\boldsymbol{q}) \in \mathbb{R}^m$:

$$\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}) = \boldsymbol{U}_f \boldsymbol{c}(\boldsymbol{V}\boldsymbol{q}) + \boldsymbol{r}. \tag{12.1}$$

Equation (12.1) has more equations than unknowns. Hence, the residual $\boldsymbol{r}$ occurs, as the basis $\boldsymbol{U}_f$ can only represent forces in the subspace spanned by $\boldsymbol{U}_f$. To solve the

equation above for the unknown reduced force vector $\boldsymbol{c}(\boldsymbol{V}\boldsymbol{q})$, a constraint for the residual is necessary for projecting the residual out.

In the derivation of the basis projection in Chapter 3, the residual was made orthogonal to the subspace spanned by the basis leading to the symmetric Galerkin projection. In the DEIM, the residual is treated in a different way. It is made zero only for certain dofs, which are the non-zero entries in the columns of the Boolean matrix $\boldsymbol{P} \in \mathbb{R}^{N \times m}$. Every column of $\boldsymbol{P}$ is a column of the identity matrix $\boldsymbol{I} \in \mathbb{R}^{N \times N}$. With enforcing

$$\boldsymbol{P}^T \boldsymbol{r} = \boldsymbol{0}, \tag{12.2}$$

the residual is hence set to zero only for the dofs corresponding to the non-zero coefficients in $\boldsymbol{P}$. When the residual is zero at certain dofs, (12.1) is collocated at exactly these dofs. Thus, the collocation of (12.1) is achieved by the premultiplication with the transposed collocation matrix $\boldsymbol{P}^T$. This leads to a determined system of equations, which can be solved for the unknown force amplitudes gathered in $\boldsymbol{c}$:

$$\boldsymbol{P}^T \boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}) = \boldsymbol{P}^T \boldsymbol{U}_f \boldsymbol{c}(\boldsymbol{V}\boldsymbol{q}) \tag{12.3}$$

$$\boldsymbol{c}(\boldsymbol{V}\boldsymbol{q}) = \left(\boldsymbol{P}^T \boldsymbol{U}_f\right)^{-1} \boldsymbol{P}^T \boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}). \tag{12.4}$$

It can be inserted in the approximation (12.1), and with accepting the residual being the error of the approximation, one gains the collocated reduced force vector

$$\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}) \approx \boldsymbol{f}_{\mathrm{DEIM}}(\boldsymbol{V}\boldsymbol{q}) = \boldsymbol{U}_f \left(\boldsymbol{P}^T \boldsymbol{U}_f\right)^{-1} \boldsymbol{P}^T \boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}). \tag{12.5}$$

The collocated internal force vector approximating the internal forces is the DEIM reduced internal force vector $\boldsymbol{f}_{\mathrm{DEIM}}(\boldsymbol{V}\boldsymbol{q})$. It can be computed cheaper than the full force vector $\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q})$, since the collocation matrix $\boldsymbol{P}$ is a sparse matrix with many zero entries. To evaluated the product $\boldsymbol{P}^T \boldsymbol{f}$ in (12.5), only the entries corresponding to a non-zero row in $\boldsymbol{P}$ have to be evaluated. The computation of all other entries in $\boldsymbol{f}$ corresponding to zero-rows in $\boldsymbol{P}$ can be omitted leading to a reduction of the computational costs.

For the design of the DEIM approximation of the nonlinear forces (12.5), two ingredients have to be determined: The force basis $\boldsymbol{U}_f$ and the Boolean collocation matrix $\boldsymbol{P}$. For the force basis, the issues discussed for kinematic bases in Chapter 5 hold also for finding an appropriate force basis: The lack of system theoretic properties allowing for proper nonlinear force estimations. Hence, the force basis $\boldsymbol{U}_f$ is computed with a POD of the force snapshots of a full, unreduced simulation similar to the POD reduced basis method discussed in Chapter 5.

To compute the Boolean matrix $\boldsymbol{P}$ containing the collocation dofs, a greedy method listed in Algorithm 4 is used, see also Chaturantabut and Sorensen [CS10]. This method iterates over the column vectors of the force basis $\boldsymbol{U}_f$. For every column, a residual $\boldsymbol{r}$ is computed representing the gap in the forces, which cannot be represented using both the force basis $\boldsymbol{U}_f$ and the collocation matrix $\boldsymbol{P}$ of the previous iteration step. Then the maximum value of the residual is picked as new collocation dof and added to the collocation dof set $\wp_{tmp}$. As the dof is associated with a node of the finite element mesh, it is an option to manipulate the selected collocation dof. For instance, it is reasonable to pick all dofs corresponding to the node of the selected dof. With the collocation point set $\wp_{tmp}$ and the collocation points of the former iteration step, the Boolean collocation matrix $\boldsymbol{P}$ is built for the next iteration step.

When the collocation point set $\wp_{tmp}$ is manipulated, as for instance all dofs of the selected node are collocated and not only the dof with the maximum residual, the number $c$ of collocation points is larger than the number $m$ of force basis vectors. Since then $\boldsymbol{P} \in \mathbb{R}^{N \times c}$ and $\boldsymbol{U}_f \in \mathbb{R}^{N \times m}$ are of different dimensions with $c > m$, the collocation (12.3)

**Algorithm 4:** Greedy collocation algorithm used in DEIM [CS10]

**Input** : Force basis $\boldsymbol{U}_f \in \mathbb{R}^{N \times n}$
**Output** : Collocation projection matrix $\boldsymbol{P} \in \mathbb{R}^{N \times |\wp_n|}$

$\wp_0 := \{\}$ // Initialize empty set of collocation points
**for** $i := 0 : n$ // Greedy loop over all force basis vectors
**do**
  // compute residual $\boldsymbol{r}$
  **if** $i = 0$ **then**
    $\boldsymbol{r} := \boldsymbol{U}_f[:, 0]$
  **else**
    $\boldsymbol{r} := \boldsymbol{U}_f[:, i] - \boldsymbol{U}_f[:, : i-1](\boldsymbol{P}_{i-1}^T \boldsymbol{U}_f[:, : i-1])^+ \boldsymbol{P}_{i-1}^T \boldsymbol{U}_f[:, i]$
  **end**
  $\wp_{tmp} := \arg\max(\boldsymbol{r})$ // Pick maximum residual
  Process collocation points $\wp_{tmp}$ based on collocation technique (e.g. add all dofs of selected node: $\wp_{tmp}$ := all dofs of current node)
  $\wp_i := \wp_{i-1} \cup \wp_{tmp}$ // update collocation point set $\wp_i$
  $\boldsymbol{P}_i := \boldsymbol{0} \in \mathbb{R}^{N \times |\wp_i|}$ // Build projector $\boldsymbol{P}_i$ from collocation set $\wp_i$
  **for** $j := 0 : |\wp_i|$ **do**
    $\boldsymbol{P}_i[\wp_i[j], j] := 1$
  **end**
**end**
$\boldsymbol{P} := \boldsymbol{P}_n$

is overdetermined and cannot be fulfilled exactly leading to another residual $\tilde{\boldsymbol{r}}$ after applying the collocation matrix $\boldsymbol{P}$:

$$\boldsymbol{P}^T \boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}) = \boldsymbol{P}^T \boldsymbol{U}_f \boldsymbol{c}(\boldsymbol{V}\boldsymbol{q}) + \tilde{\boldsymbol{r}}. \tag{12.6}$$

The equation above cannot be solved directly for the reduced forces $\boldsymbol{c}(\boldsymbol{V}\boldsymbol{q})$, since the residual is also unknown. However, it can be solved in a least-square sense, so that the residual $\tilde{\boldsymbol{r}}$ is minimized in the $L^2$-norm. Then the reduced forces $\boldsymbol{c}(\boldsymbol{V}\boldsymbol{q})$ are given as:

$$\boldsymbol{c}(\boldsymbol{V}\boldsymbol{q}) \approx \left(\boldsymbol{P}^T \boldsymbol{U}_f\right)^+ \boldsymbol{P}^T \boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}), \tag{12.7}$$

with $(\boldsymbol{P}^T \boldsymbol{U}_f)^+$ being the Moore-Penrose pseudo inverse of $\boldsymbol{P}^T \boldsymbol{U}_f$. The DEIM approximated internal force vector $\boldsymbol{f}_{\text{DEIM}}$ for $c > m$, i.e., more collocation points than force basis vectors, is then:

$$\boldsymbol{f}_{\text{DEIM}}(\boldsymbol{V}\boldsymbol{q}) = \boldsymbol{U}_f \left(\boldsymbol{P}^T \boldsymbol{U}_f\right)^+ \boldsymbol{P}^T \boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}). \tag{12.8}$$

The formula above includes both cases $c = m$ and $c > m$, since the Moore-Penrose pseudo inverse yields the inverse for full rank square matrices. As (12.8) is more general and includes (12.5), the former is used from now on to describe the DEIM independent of the collocation technique employed. In Subsection 12.2.2, further collocation techniques are discussed, especially for the UDEIM introduced in Subsection 12.2.1.

### 12.1.1 Oblique Projection

The approximation of the forces according to the DEIM (12.5) allows for an interpretation in terms of projection. Thereby, the two cases of collocation have to be distinguished. In the case with $c = m$, both subspaces, the force subspace spanned by $\boldsymbol{U}_f$ and the

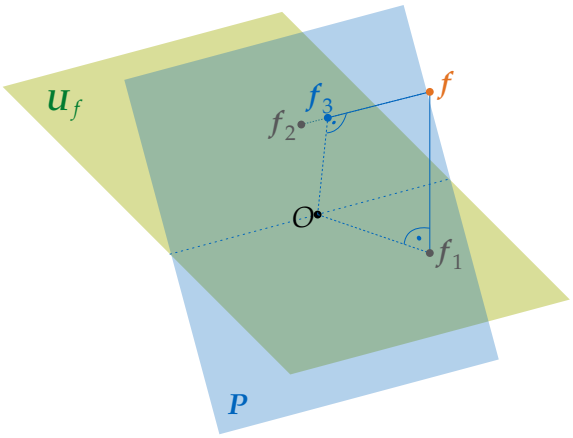


**Figure 12.1:** Oblique projection of the force vector $f$ onto the subspace spanned by $U_f$. The orthogonal projection of $f$ onto $U_f$ results in $f_1$; The orthogonal projection of $f$ onto $P$ results in $f_3$; The oblique projection of $f$ onto $U_f$ perpendicular to $P$ results in $f_2$.

collocation subspace spanned by $P$ are of equal size. They can be depicted as two dimensional planes in three dimensional space, as in Figure 12.1. There, the internal force vector $f$ is depicted as a point in the three dimensional vector space. The expression $U_f(P^T U_f)^{-1} P^T$ performs an *oblique projection* onto the subspace spanned by $U_f$, where the error of the projection is forced to be orthogonal to the subspace spanned by $P$. This oblique projection results in the point $f_2$, which is an element of the space spanned by $U_f$. It is, however, different from the perpendicular projection of $f$ onto $U_f$ which is $f_1$, if $\text{span}(P) \neq \text{span}(U_f)$. The perpendicular projection of $f$ onto $U_f$ is optimal in the sense of the $L^2$ norm. This means, that the distance between $f$ and $f_1$ is minimal and that the residual is orthogonal to $U_f$. However, in the oblique projection, this is not the case. There, the oblique projected force vector $f_2$ is, depending on the space spanned by $P$, not equal or close to the optimal projection $f_1$. Hence, the subspace spanned by $P$ has a great influence on the result of the oblique projection.

In the second case when more collocation points than force modes exist, i.e., $c > m$, the projection of the internal force vector (12.8) can be interpreted as a minimization problem. There, the force subspace spanned by $U_f$ is of lower dimension than the collocation subspace spanned by $P$. Hence, $U_f$ is depicted as a one dimensional subspace, i.e., a line, and $P$ as a two dimensional subspace, a plane, in three dimensional space. This analogy is depicted in Figure 12.2. The oblique projection of the force vector $f$ onto $U_f$ with the error being orthogonal to $P$ is not possible in general. This is equivalent to the projection of the point $f$ onto the line $U_f$ with the projection being orthogonal to the plane $P$. Since the line forming the projection of $f$ onto $P$ is skew to the line $U_f$ in general, this is generally not possible. The projection operation $U_f(P^T U_f)^{+} P^T$ using the Moore-Penrose pseudo inverse, however, performs a minimization by projecting the point $f$ onto the subspace spanned by $U_f$ by minimizing the $L^2$ norm of the residual $\tilde{r}$. This residual can be interpreted as the orthogonal projection of the connecting line between $f$ and the resulting force vector $f_2$ onto the space spanned by $P$. The two end points of the connecting line $f$ and $f_2$ are projected onto $P$ resulting in $f_4$ and $f_3$. These two points form the end points of the residual $\tilde{r}$, which is minimized.

Consequently, the operator $U_f(P^T U_f)^{+} P^T$ performs the best oblique projection of $f$ onto $U_f$ while keeping the error in the space spanned by $P$ minimal.

Both illustrations of the projection indicate that the internal force vector $f$ is treated in a non-orthogonal way to achieve the approximation $f_{\text{DEIM}}$. It is motivated by the sparse evaluation of the nonlinear force vector $f$. However, the loss of orthogonality comes with

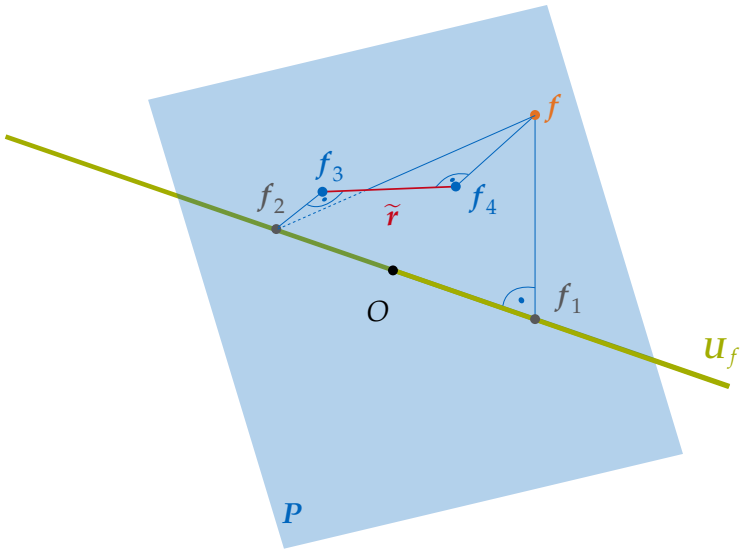


**Figure 12.2:** Oblique projection of the force vector $\boldsymbol{f}$ onto the subspace spanned by $\boldsymbol{U}_f$ with $c > m$. The orthogonal projection of $\boldsymbol{f}$ onto $\boldsymbol{U}_f$ results in $\boldsymbol{f}_1$; With this oblique projection, the residual $\widetilde{\boldsymbol{r}}$ being the gap between $\boldsymbol{f}$ and $\boldsymbol{f}_2$ projected onto $\boldsymbol{P}$ is minimized.

drawbacks which are discussed next.

### 12.1.2 Loss of Stability and Stabilization

The general concept of the DEIM comes with a fundamental drawback: The destruction of the symmetry of the system. With it, desirable properties like stability, variational consistency and an underlying energy function fitting into the Lagrangian framework are lost. The loss of all these properties is clearly visible, when the DEIM-reduced tangential stiffness matrix $\boldsymbol{K}_{\text{r,DEIM}}$ is expressed in terms of the symmetric tangential stiffness matrix $\boldsymbol{K}(\boldsymbol{u})$ of the underlying high dimensional system. It is derived as:

$$\begin{aligned}
\boldsymbol{K}_{\text{r, DEIM}} &= \frac{\partial}{\partial \boldsymbol{q}} \left( \boldsymbol{V}^T \boldsymbol{U}_f \left( \boldsymbol{P}^T \boldsymbol{U}_f \right)^+ \boldsymbol{P}^T \boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}) \right) \\
&= \boldsymbol{V}^T \boldsymbol{U}_f \left( \boldsymbol{P}^T \boldsymbol{U}_f \right)^+ \boldsymbol{P}^T \frac{\partial \boldsymbol{f}(\boldsymbol{u} = \boldsymbol{V}\boldsymbol{q})}{\partial \boldsymbol{u}} \boldsymbol{V} \\
&= \boldsymbol{V}^T \boldsymbol{U}_f \left( \boldsymbol{P}^T \boldsymbol{U}_f \right)^+ \boldsymbol{P}^T \boldsymbol{K}(\boldsymbol{V}\boldsymbol{q}) \boldsymbol{V}.
\end{aligned}$$

The DEIM reduced tangential stiffness matrix is computed from the full stiffness matrix $\boldsymbol{K}(\boldsymbol{u})$ by a left sided projection $\boldsymbol{V}^T\boldsymbol{U}_f(\boldsymbol{P}^T\boldsymbol{U}_f)^{-1}\boldsymbol{P}^T$ and the right sided projection $\boldsymbol{V}$. As both projectors are not equal, the symmetry of $\boldsymbol{K}$ is destroyed by the DEIM hyper-reduction.

The loss of symmetry leads to a loss of stability, which is often observed in Newton-Raphson loops in static solutions or in time integration schemes. In an attempt to improve the stability of non-symmetric oblique DEIM projection, the linear and the nonlinear part of the restoring forces are split apart (cf. [CS10; TR13]). The linear part is then symmetrically reduced as it is done in linear systems and only on the purely nonlinear part the whole DEIM procedure is applied. That means, that the force basis $\boldsymbol{U}_f$ stems from the SVD of the purely nonlinear part of the internal forces. The collocation scheme Algorithm 4 is then applied on this purely nonlinear force basis to obtain $\boldsymbol{P}$. The DEIM hyper-reduced nonlinear force vector is then:

$$\boldsymbol{f} = \boldsymbol{K}\boldsymbol{u} + \boldsymbol{f}_{nl} \quad \Leftrightarrow \quad \boldsymbol{f}_{nl} = \boldsymbol{f} - \boldsymbol{K}\boldsymbol{u}, \tag{12.9}$$

$$\boldsymbol{f}_r = \boldsymbol{V}^T\boldsymbol{K}\boldsymbol{V}\boldsymbol{q} + \boldsymbol{V}^T\boldsymbol{f}_{nl}, \tag{12.10}$$

$$\boldsymbol{f}_{r,\mathrm{DEIM}} = \boldsymbol{V}^T\boldsymbol{K}\boldsymbol{V}\boldsymbol{q} + \boldsymbol{V}^T\boldsymbol{U}_f\left(\boldsymbol{P}^T\boldsymbol{U}_f\right)^+\boldsymbol{P}^T(\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}) - \boldsymbol{K}\boldsymbol{V}\boldsymbol{q}) \tag{12.11}$$

$$= \underbrace{\boldsymbol{V}^T\left(\boldsymbol{I} - \boldsymbol{U}_f\left(\boldsymbol{P}^T\boldsymbol{U}_f\right)^+\boldsymbol{P}^T\right)\boldsymbol{K}\boldsymbol{V}}_{\boldsymbol{K}_{\mathrm{r,lin,DEIM}}}\boldsymbol{q} + \boldsymbol{V}^T\boldsymbol{U}_f\left(\boldsymbol{P}^T\boldsymbol{U}_f\right)^+\boldsymbol{P}^T\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}). \tag{12.12}$$

The matrix $K_{\mathrm{r,lin,DEIM}} = \boldsymbol{V}^T(\boldsymbol{I} - \boldsymbol{U}_f(\boldsymbol{P}^T\boldsymbol{U}_f)^{-1}\boldsymbol{P}^T)\boldsymbol{K}\boldsymbol{V}$ is the pseudo linear stiffness matrix for the DEIM procedure, which can be precomputed. For the implicit solution of static or dynamic problems, the tangential stiffness matrix is also necessary. It is given as

$$\begin{aligned}\boldsymbol{K}_{r,\mathrm{DEIM}}(\boldsymbol{V}\boldsymbol{q}) &= \frac{\partial \boldsymbol{f}_{\mathrm{r,DEIM}}}{\partial \boldsymbol{q}} \\ &= \underbrace{\boldsymbol{V}^T\left(\boldsymbol{I} - \boldsymbol{U}_f\left(\boldsymbol{P}^T\boldsymbol{U}_f\right)^+\boldsymbol{P}^T\right)\boldsymbol{K}\boldsymbol{V}}_{\boldsymbol{K}_{\mathrm{r,lin,DEIM}}} + \underbrace{\boldsymbol{V}^T\boldsymbol{U}_f\left(\boldsymbol{P}^T\boldsymbol{U}_f\right)^+\boldsymbol{P}^T\boldsymbol{K}(\boldsymbol{V}\boldsymbol{q})\boldsymbol{V}}_{\boldsymbol{K}_{\mathrm{r,nl,DEIM}}}.\end{aligned} \tag{12.13}$$

As above, the pseudo linear stiffness matrix can be precomputed while the nonlinear part has to be evaluated at every iteration step. The computational saving is effective, as only the subset $\widetilde{\mathcal{E}}$ of the element set $\mathcal{E}$ is necessary to obtain the DEIM hyper-reduced nonlinear forces:

$$\boldsymbol{f}_{r,\mathrm{DEIM}} = \sum_{e\in\widetilde{\mathcal{E}}} \boldsymbol{V}^T\boldsymbol{U}_f\left(\boldsymbol{P}^T\boldsymbol{U}_f\right)^+\boldsymbol{P}^T\boldsymbol{L}_e^T\boldsymbol{f}_e(\boldsymbol{L}_e\boldsymbol{V}\boldsymbol{q}) + \boldsymbol{K}_{r,\mathrm{lin,DEIM}}\boldsymbol{q}. \tag{12.14}$$

The reduced element set $\widetilde{\mathcal{E}}$ is defined as:

$$\widetilde{\mathcal{E}} = \{e \in \mathcal{E} \mid \boldsymbol{P}^T\boldsymbol{L}_e^T \neq \boldsymbol{0}\}. \tag{12.15}$$

All other elements $e$ in the element set $\mathcal{E}$, which are not associated with a collocation dof, i.e., $\boldsymbol{P}^T\boldsymbol{L}_e^T = \boldsymbol{0}$, do not contribute to the nonlinear DEIM forces and are hence not in the reduced element set $\widetilde{\mathcal{E}}$.

In the DEIM procedure, the oblique projection operator and the reduced basis is constant. Hence, a compact auxiliary matrix $\boldsymbol{H} \in \mathbb{R}^{n\times c}$ can be precomputed, so that the resulting expression for the nonlinear force vector $\boldsymbol{f}_{r,\mathrm{DEIM}}$ and the tangential stiffness matrix $\boldsymbol{K}_{r,\mathrm{DEIM}}$ is:

$$\boldsymbol{H} = \boldsymbol{V}^T\boldsymbol{U}_f\left(\boldsymbol{P}^T\boldsymbol{U}_f\right)^+, \tag{12.16}$$

$$\boldsymbol{f}_{r,\mathrm{DEIM}} = \sum_{e\in\widetilde{\mathcal{E}}} \boldsymbol{H}\boldsymbol{P}^T\boldsymbol{L}_e^T\boldsymbol{f}_e(\boldsymbol{L}_e\boldsymbol{V}\boldsymbol{q}) + \boldsymbol{K}_{r,\mathrm{lin,DEIM}}\boldsymbol{q}, \tag{12.17}$$

$$\boldsymbol{K}_{r,\mathrm{DEIM}} = \sum_{e\in\widetilde{\mathcal{E}}} \boldsymbol{H}\boldsymbol{P}^T\boldsymbol{L}_e^T\boldsymbol{K}_e(\boldsymbol{L}_e\boldsymbol{V}\boldsymbol{q})\boldsymbol{L}_e\boldsymbol{V} + \boldsymbol{K}_{r,\mathrm{lin,DEIM}}. \tag{12.18}$$

The operator $\boldsymbol{P}^T\boldsymbol{L}_e^T$ is a Boolean matrix for every element $e$ expressing the dof selection of the elemental contributions $\boldsymbol{f}_e$ acting onto the oblique projector. In computer implementations it can be realized efficiently with index operations.

## 12.2 Variants and Improvements

### 12.2.1 Unassembled DEIM (UDEIM)

The classical DEIM as first proposed in Chaturantabut and Sorensen [CS10] is designed for schemes where so-called vector valued functions are present. This means, that the evaluation of a force at one single dof involves only the displacements of the chosen dof and its neighbors and can be computed efficiently. For example, this is the case in finite difference schemes, where the derivatives are computed using the finite differences of the chosen dof and the adjacent dofs directly.

In the finite element framework, though, the computation of the nonlinear force at a single dof involves the evaluation of all adjacent elements of the chosen dof. Compared to finite difference schemes, the cost of evaluating a single nonlinear force at a specific dof becomes quickly expensive, since one node has usually multiple adjacent elements, especially in 3D meshes. The classical DEIM reduction can quickly become inefficient since many more elements have to be evaluated than collocation dofs exist.

To solve this issue, *unassembled* DEIM (UDEIM) has been proposed by Tiso and Rixen [TR13]. The key idea is to apply DEIM to the unassembled finite element mesh, so that only one element has to be evaluated when one collocation dof is chosen. The assembly is then performed after the unassembled DEIM approximation.

The assembly operation (2.31) is a sum over all elements and can be expressed as a matrix vector product, since the summation is a linear operation. It can be performed with the linear assembly operator represented by the Boolean matrix $\boldsymbol{B}_A \in \mathbb{R}^{N_e \cdot |\mathcal{E}| \times N}$, which is the linear transformation operator between the unassembled and the assembled domain. $N_e$ is the number of dofs per element, $|\mathcal{E}|$ the number of elements and $N$ the number of assembled dofs. The unassembled force vector $\boldsymbol{f}_u \in \mathbb{R}^{N_e \cdot |\mathcal{E}|}$ and unassembled stiffness matrix $\boldsymbol{K}_u \in \mathbb{R}^{N_e \cdot |\mathcal{E}| \times N_e \cdot |\mathcal{E}|}$ are defined as

$$\boldsymbol{f}_u = \left( \boldsymbol{f}_1^T, \dots, \boldsymbol{f}_{|\mathcal{E}|}^T \right)^T, \tag{12.19}$$

$$\boldsymbol{K}_u = \operatorname{diag}\left( \boldsymbol{K}_1, \dots, \boldsymbol{K}_{|\mathcal{E}|} \right), \tag{12.20}$$

with the internal force vector $\boldsymbol{f}_e \in \mathbb{R}^{N_e}$ and the tangential stiffness matrix $\boldsymbol{K}_e \in \mathbb{R}^{N_e \times N_e}$ of the element $e$. The unassembled quantities are assembled with the Boolean assembly matrix $\boldsymbol{B}_A \in \mathbb{R}^{N_e \cdot |\mathcal{E}| \times N}$:

$$\boldsymbol{B}_A = \left( \boldsymbol{L}_1^T, \dots, \boldsymbol{L}_{|\mathcal{E}|}^T \right)^T \tag{12.21}$$

yielding the assembled force vector $\boldsymbol{f}$ and stiffness matrix $\boldsymbol{K}$:

$$\boldsymbol{f} = \boldsymbol{B}_A^T \boldsymbol{f}_u, \tag{12.22}$$

$$\boldsymbol{K} = \boldsymbol{B}_A^T \boldsymbol{K}_u \boldsymbol{B}_A. \tag{12.23}$$

As UDEIM is equivalent to DEIM but operating in the unassembled domain, the force basis $\boldsymbol{U}_f$ has to be computed for the unassembled domain yielding $\boldsymbol{U}_{f,u}$. It cannot be derived directly from $\boldsymbol{U}_f$ but has to be computed with a POD on the unassembled force snapshots of the training displacement sets.

The UDEIM hyper-reduced nonlinear forces are computed similar to (12.12) but with the collocation operator $\boldsymbol{P}_u$ applied directly to the unassembled quantities. The assembly is done directly before the projection onto the reduced basis $\boldsymbol{V}$:

$$\begin{aligned} \boldsymbol{f}_{\text{r,UDEIM}} = & \underbrace{\boldsymbol{V}^T \boldsymbol{B}_A^T \left( \boldsymbol{I} - \boldsymbol{U}_{f,u} \left( \boldsymbol{P}_u^T \boldsymbol{U}_{f,u} \right)^+ \boldsymbol{P}_u^T \right) \boldsymbol{K}_u \boldsymbol{B}_A \boldsymbol{V}}_{\boldsymbol{K}_{\text{r,lin,UDEIM}}} \boldsymbol{q} \\ & + \boldsymbol{V}^T \boldsymbol{B}_A^T \boldsymbol{U}_{f,u} \left( \boldsymbol{P}_u^T \boldsymbol{U}_{f,u} \right)^+ \boldsymbol{P}_u^T \boldsymbol{f}_u(\boldsymbol{V}\boldsymbol{q}), \end{aligned} \tag{12.24}$$

$$\begin{aligned}\boldsymbol{K}_{\mathrm{r,UDEIM}}(\boldsymbol{V}\boldsymbol{q}) = &\underbrace{\boldsymbol{V}^T\boldsymbol{B}_A^T\left(\boldsymbol{I}-\boldsymbol{U}_{f,u}\left(\boldsymbol{P}_u^T\boldsymbol{U}_{f,u}\right)^+\boldsymbol{P}_u^T\right)\boldsymbol{K}_u\boldsymbol{B}_A\boldsymbol{V}}_{\boldsymbol{K}_{\mathrm{r,lin,UDEIM}}}\\ &+\underbrace{\boldsymbol{V}^T\boldsymbol{B}_A^T\boldsymbol{U}_{f,u}\left(\boldsymbol{P}_u^T\boldsymbol{U}_{f,u}\right)^+\boldsymbol{P}_u^T\boldsymbol{K}_u(\boldsymbol{V}\boldsymbol{q})\boldsymbol{B}_A\boldsymbol{V}}_{\boldsymbol{K}_{\mathrm{r,nl,UDEIM}}}.\end{aligned} \tag{12.25}$$

For UDEIM, a compact auxiliary matrix $\boldsymbol{H}_u \in \mathbb{R}^{n\times c}$ can be precomputed similar to the DEIM. The reduced assembly procedure is then a sum over the reduced element set $\widetilde{\mathcal{E}}$ as defined in (12.15):

$$\boldsymbol{H}_u = \boldsymbol{V}^T\boldsymbol{B}_A^T\boldsymbol{U}_{f,u}\left(\boldsymbol{P}^T\boldsymbol{U}_{f,u}\right)^+, \tag{12.26}$$

$$\boldsymbol{f}_{r,\mathrm{UDEIM}} = \sum_{e\in\widetilde{\mathcal{E}}}\boldsymbol{H}_u\boldsymbol{P}_u^T\boldsymbol{L}_e^T\boldsymbol{f}_e(\boldsymbol{L}_e\boldsymbol{V}\boldsymbol{q}) + \boldsymbol{K}_{r,\mathrm{lin,UDEIM}}\boldsymbol{q}, \tag{12.27}$$

$$\boldsymbol{K}_{r,\mathrm{UDEIM}} = \sum_{e\in\widetilde{\mathcal{E}}}\boldsymbol{H}_u\boldsymbol{P}_u^T\boldsymbol{L}_e^T\boldsymbol{K}_e(\boldsymbol{L}_e\boldsymbol{V}\boldsymbol{q})\boldsymbol{L}_e\boldsymbol{V} + \boldsymbol{K}_{r,\mathrm{lin,UDEIM}} \tag{12.28}$$

To obtain the unassembled collocation operator $\boldsymbol{P}_u$ using the unassembled force basis $\boldsymbol{U}_{f,u}$, the same collocation method for DEIM listed in Algorithm 4 is used. However, in the unassembled DEIM configuration, more collocation strategies are possible. In the next subsection, further strategies for collocation are discussed.

Since the assembled force quantities can be computed from the unassembled quantities but not vice versa, both the DEIM and UDEIM can be implemented in one routine operating on the unassembled domain. With the assembly operator $\boldsymbol{B}_A$ added at the relevant places in the formulas, the UDEIM can be easily transferred to the DEIM. Note, that the computation of the force basis $\boldsymbol{U}_f$ for DEIM has to be performed in the assembled domain. The computation of the SVD in the unassembled domain and the application of the assembly operator $\boldsymbol{B}_A$ onto $\boldsymbol{U}_{f,u}$ to obtain $\boldsymbol{U}_f$ is not allowed, since the extraction of the singular vectors of a matrix is a nonlinear operation.

An overview of the UDEIM is given in Figure 12.3. Since the UDEIM can be interpreted as a superset of the DEIM, the DEIM procedure can be inferred by substituting the unassembled quantities $\boldsymbol{f}_u$ and $\boldsymbol{K}_u$ by the assembled counterparts $\boldsymbol{f}$ and $\boldsymbol{K}$ and omitting the assembly operator $\boldsymbol{B}_A$.

### 12.2.2 Collocation Techniques

Algorithm 4 depicts the selection of the collocation points for a given force basis $\boldsymbol{U}_f$. In line 10 of Algorithm 4 the selected collocation can be processed, i.e., the selected set $\wp_{tmp}$ of dofs used for collocation can be extended to exploit more information of the sparse force evaluation, as suggested in [FCA13]. In the classical DEIM, the force of one dof is computed by evaluating all adjacent elements of the node to which the dof belongs to. Hence, it is reasonable to extend the set of selected dofs $\wp_{tmp}$ to all dofs associated to the node. In the UDEIM, however, the selected dofs are associated to unassembled elements making more collocation strategies plausible. They are listed together with the strategies for DEIM in Table 12.1.

The dof collocation strategy is the native DEIM strategy. In this method, $\wp_{tmp}$ is not manipulated so that it only contains the selected dof. Only in this case, the number of collocation points is equal to the number of force modes, i.e., $c = m$. For all other collocation strategies the number of collocation points is larger than the number of force modes leading to the overdetermined system (12.6). In the *Node* collocation strategy, all dofs associated of the selected node are added to $\wp_{tmp}$. In the *Element* collocation strategy, all dofs contained in the unassembled element are selected. With this method, all of the

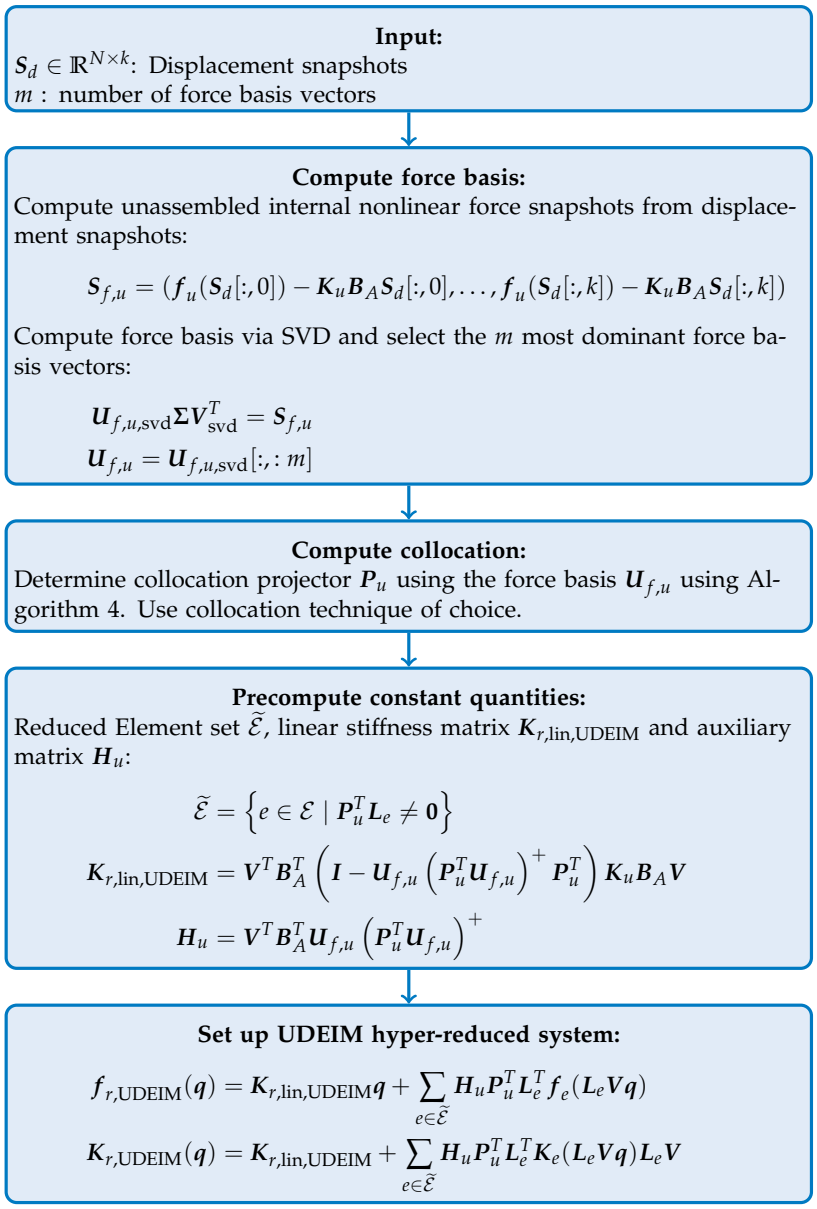


**Figure 12.3:** UDEIM algorithm for computing the nonlinear force approximation.

available information of the sparse internal forces computed in (12.27) is exploited. The *Component* collocation strategy is similar to the Element collocation. It determines the component of the selected node and adds the component of all nodes of the element to $\wp_{tmp}$.

### 12.2.3 A Failed Attempt to Symmetrize UDEIM

One main drawback of the DEIM and the UDEIM is that the symmetry of the problem is destroyed leading to the loss of many desirable properties, as discussed in Section 12.1. In the recent publication by Chaturantabut et al. [CBG16], a symmetric version of the DEIM is proposed, which preserves the Hamiltonian structure of mechanical systems and hence all the desirable properties like symmetry, stability and positive definiteness. The key idea of the symmetric DEIM is to apply the oblique projection operator $U_f(P^T U_f)^{-1} P^T$ not only on the dual variables, the forces, but also transposed on the primal variable, the displacements. The proposed formulation for the internal nonlinear force vector is:

**Table 12.1:** Possible collocation types for DEIM and UDEIM reduction

| Name | selection strategy for $\wp_{tmp}$ | DEIM | UDEIM |
|---|---|---|---|
| dof | only the selected dof | + | + |
| Node | all dofs of the selected node | + | + |
| Element | all dofs of the selected element | - | + |
| Component | all dofs belonging to the selected component of the selected element | - | + |

$$\boldsymbol{f}_r(\boldsymbol{q}) \approx \boldsymbol{V}^T\boldsymbol{U}_f \left(\boldsymbol{P}^T\boldsymbol{U}_f\right)^+ \boldsymbol{P}^T\boldsymbol{f}\left(\boldsymbol{P}\left(\left(\boldsymbol{P}^T\boldsymbol{U}_f\right)^+\right)^T \boldsymbol{U}_f^T\boldsymbol{V}\boldsymbol{q}\right). \tag{12.29}$$

This formulation is symmetric by definition, if the internal force vector $\boldsymbol{f}(\boldsymbol{u})$ is a symmetric function. In the example given in Chaturantabut et al. [CBG16], an application is given for a vector-valued nonlinear circuit system.

However, when applying this concept to the finite element framework, the forces are not evaluated in a vector-valued fashion as in finite differences, but in an element based framework, where the forces are computed on the element level. Hence it is important to reconstruct the full elemental displacement $\boldsymbol{u}_e$ of element $e$, even if only one component of the elemental forces $\boldsymbol{f}_e$ is used. As a consequence, the oblique projection operator $\boldsymbol{P}(\boldsymbol{P}^T\boldsymbol{U}_f)^{-T}\boldsymbol{U}_f^T$ should reconstruct the displacements of the selected elements $\widetilde{\mathcal{E}}$ particularly well.

The operator $\boldsymbol{P}(\boldsymbol{P}^T\boldsymbol{U}_f)^{-T}\boldsymbol{U}_f^T$ performs an oblique projection on the subspace spanned by $\boldsymbol{P}$, where the error of the projection is orthogonal to the subspace spanned by $\boldsymbol{U}_f$, as depicted in Figure 12.1. As a consequence, the results of the projection lie in the subspace spanned by $\boldsymbol{P}$. Since $\boldsymbol{P}$ is a Boolean collocation matrix, only the dofs selected in $\boldsymbol{P}$ contain results after the projection, while the other, non-selected dofs remain zero.

This is of particular concern, when the displacement of a finite element should be reconstructed. Since the displacements of all dofs of an element influence all other dofs, it makes only sense to reconstruct the full displacement of the selected element. Hence, only the UDEIM is reasonable, as only there a selected dof can be associated to a free, isolated element. Additionally, the Element collocation technique is necessary, as only this method has the ability to reconstruct the full displacement of the element.

In principle, the UDEIM with element collocation could lead to proper results. However, the experimental results are very poor. The computations are seldom stable and if they are, they do not follow the references in any way. The reason is, that the oblique projection distorts the elements severely. In Figure 12.4 a displacement field of the cantilever example is given in the displaced configuration, and the configuration after the oblique projection $\boldsymbol{P}(\boldsymbol{P}^T\boldsymbol{U}_f)^{-T}\boldsymbol{U}_f^T$ of the reduced displacement field. The DEIM selected elements highlighted in red are so severely distorted, that they have nothing in common with the original displacement field. As a consequence, the internal forces $\boldsymbol{f}_r$ are several orders of magnitude higher than the internal forces of the regular displaced configuration high as depicted in Figure 12.5.

Hence, the idea of symmetrizing DEIM in the finite element framework does not work in the manner described in Caturantabut et al. [CBG16], even with considerations reflecting the methodological differences. However, there exist hyper-reduction techniques which preserve the structure and are tailored to the finite element framework, which are discussed in the following chapter.

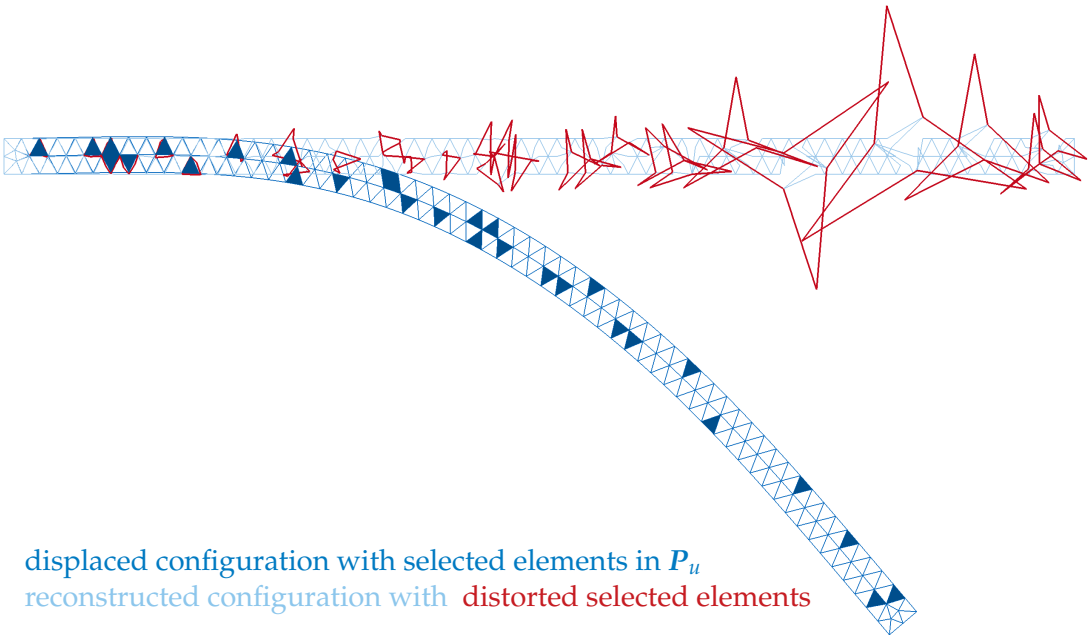


**Figure 12.4:** Example of the distortion of the mesh due to the symmetric UDEIM approach for the cantilever example at $t = 1.3\,\mathrm{s}$. The basis consists of 20 POD modes, the force basis $\boldsymbol{U}_{f,u}$ is of equal size. The deformed beam with the blue elements shows the deformed configuration; The highlight blue beam with the red elements depicts the same displacement field after the oblique projection.

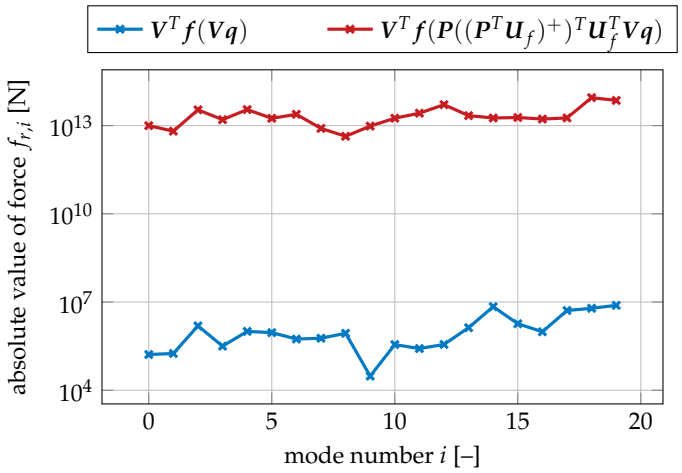


**Figure 12.5:** Nonlinear internal reduced forces of the symmetric UDEIM approach and the reduced model for a POD basis of $n = 20$. The forces correspond to the displacement fields in Figure 12.4. Since the elements are so drastically deformed, the nonlinear forces are several orders of magnitude higher than the real nonlinear forces.

## 12.3 Applications

To assess the properties of the (U)DEIM, the cantilever example used in the previous chapters is investigated. Since the (U)DEIM needs training sets stemming from a reference solution, a full, unreduced simulation is run to obtain the training snapshots which are gathered in the snapshot matrix $\boldsymbol{S}_d$. Then, a POD basis is computed using the first 10 POD modes. With the POD basis different (U)DEIM hyper-reduced models are built with different sizes of the force basis $\boldsymbol{U}_f$ and $\boldsymbol{U}_{f,u}$, respectively. For the numerical experiments, both the collocation strategies proposed in Subsection 12.2.2 and the size for the force basis $\boldsymbol{U}_f$ and $\boldsymbol{U}_{f,u}$ are varied and a time integration for every combination is run.

**Table 12.2:** Run times and errors for (U)DEIM reduction of the cantilever example with a reduced basis of 10 POD modes.

| Size $m$ of | DEIM-dof | | DEIM-Node | |
|---|---|---|---|---|
| force basis | $RE_{hr}[\%]$ | $t_w[s]$ | $RE_{hr}[\%]$ | $t_w[s]$ |
| 4 | - | - | - | - |
| 7 | - | - | - | - |
| 10 | - | - | - | - |
| 15 | - | - | - | - |
| 20 | - | - | 69.45 | 7.83 |
| 40 | 80.83 | 10.98 | 78.22 | 10.20 |

| Size $m$ of | UDEIM-dof | | UDEIM-Node | | UDEIM-Element | | UDEIM-Component | |
|---|---|---|---|---|---|---|---|---|
| force basis | $RE_{hr}[\%]$ | $t_w[s]$ | $RE_{hr}[\%]$ | $t_w[s]$ | $RE_{hr}[\%]$ | $t_w[s]$ | $RE_{hr}[\%]$ | $t_w[s]$ |
| 4 | - | - | - | - | - | - | - | - |
| 7 | - | - | - | - | - | - | - | - |
| 10 | - | - | - | - | 198.26 | 7.69 | - | - |
| 15 | - | - | - | - | - | - | - | - |
| 20 | - | - | 4.35 | 6.27 | 5.50 | 6.23 | 521.56 | 11.68 |
| 40 | 20.72 | 8.73 | 12.35 | 9.48 | 3.54 | 9.38 | 6.54 | 8.77 |

The wall time and the relative hyper-reduction error $RE_{hr}$ for the numerical experiments are depicted in Table 12.2. When no value is given in the table, the time integration scheme was not stable, i.e., the Newton-Raphson loop inside the time integration did not converge within 30 iterations. As Table 12.2 illustrates, stability is an issue with all DEIM and UDEIM variants with small force bases. Up to $m = 15$ no (U)DEIM variant is stable besides the UDEIM-Element. This variant, however, has a very poor accuracy with a relative hyper-reduction error $RE_{hr}$ being clearly greater than 100 %.

With an increasing size $m$ of the force basis $\boldsymbol{U}_f$ and $\boldsymbol{U}_{f,u}$, respectively, the (U)DEIM procedures become more stable, up to the point with $m = 40$, where every variant of the (U)DEIM exhibits stability. The accuracy of the hyper-reduction, though, is different for the investigated methods. While the DEIM hyper-reduced models show poor accuracy with $RE_{hr}$ values in the range of 70-80 %, the UDEIM exhibits better accuracy. The best method among all seems to be the UDEIM with Element collocation, the method

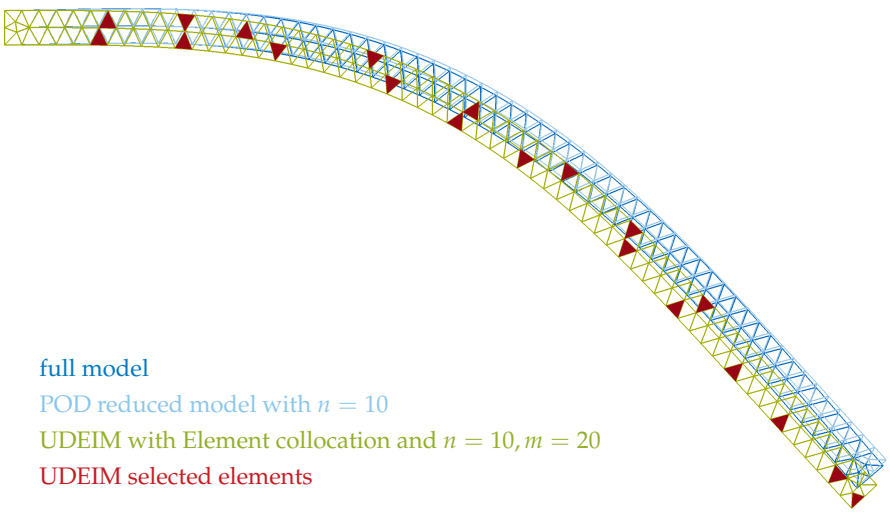


**Figure 12.6:** Displacement fields of the cantilever example at $t = 0.31\,\mathrm{s}$ for the full, reduced and UDEIM Element collocation reduced model with $m = 20$.

where the whole information of the selected elements is exploited. Figure 12.6 depicts the displaced mesh with the selected elements for the UDEIM with Element collocation and $m = 20$ force basis vectors at $t = 0.31\,\mathrm{s}$. With this collocation technique, 20 elements are selected for collocation which are scattered along the beam. The hyper-reduced cantilever depicted in green approximates the full and POD reduced model well.

The displacements of the tip of certain selected (U)DEIM reduced models are shown in Figure 12.7. The UDEIM with Node collocation and 10 force basis vectors becomes unstable at about $t = 0.31\,\mathrm{s}$. The DEIM with Node collocation and 20 force basis vectors does not follow the reference solution well. The UDEIM with Element collocation and 20 force basis vectors, however, shows a good match with the reference solution, which is in accordance with the low hyper-reduction relative error of $RE_{hr} = 5.50\,\%$.

With the (U)DEIM hyper-reduction the wall time for the computation decreases considerably. Whereas the full cantilever model requires $t_w = 97.01\,\mathrm{s}$ in wall time and the POD reduced model with 10 POD basis vectors requires $t_w = 36.50\,\mathrm{s}$, all (U)DEIM reduced models are below a wall time of 11.7 s. Especially the UDEIM with Element collocation and 20 basis vectors yields a good wall time of $t_w = 6.23\,\mathrm{s}$ with a good hyper-reduction error value.

The c-shape example depicted in Figure 6.15 is also investigated. Therefore, it is reduced using a POD basis with 10 POD modes and a time integration of the same variants of the previous example is run. In Table 12.3, the computation wall time and the accuracy are listed of the several variants. They confirm the observations of the cantilever example. For small force bases and hence small values for $m$, all (U)DEIM hyper-reduced models are unstable and fail to converge in the Newton-Raphson loop of the time integration scheme. For higher values of $m$ the UDEIM become stable with the UDEIM with Element collocation exhibiting the best results in terms of accuracy and wall time. This is in accordance with the previous experiment using the cantilever example. In the c-shape example, on the contrary, both the DEIM with dof collocation and the UDEIM with dof collocation never gain convergence for the given force basis sizes. As a consequence, the pure collocation with the number of force modes being equal to the number of collocation points, i.e., $m = c$, seems to have more issues with stability compared to a setup, where the number of collocation points is larger than the number of force modes. In this setup, where the collocation becomes a least square fit minimizing the residual $\widetilde{\boldsymbol{r}}$ in (12.6), the

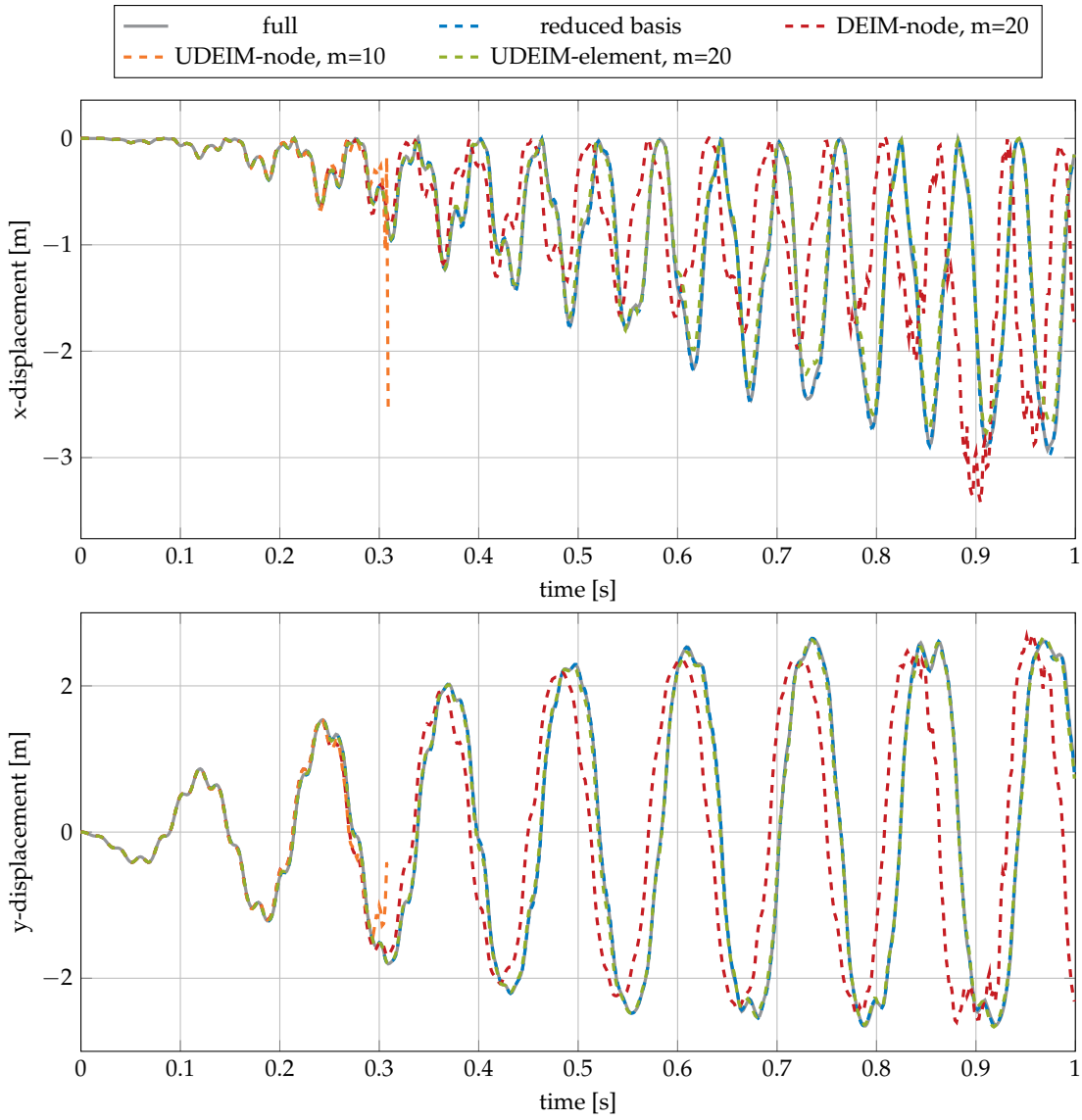


**Figure 12.7:** Displacement of the tip of the cantilever for different (U)DEIM hyper-reduction methods; $m$ is the size of the force basis. The UDEIM-node with $m = 10$ does not converge in the Newton-Raphson-iteration at approximately $t = 0.31\,\mathrm{s}$.

information of more dofs is used leading to more stable and more accurate results.

The speedup for the c-shape example are also impressive. The full model requires $t_w = 248.46\,\mathrm{s}$, the POD reduced model $t_w = 137.37\,\mathrm{s}$. With the UDEIM with Element collocation, the wall time with $t_w = 14.99\,\mathrm{s}$ yields a speedup factor of 16.6 versus the full model and 9.2 versus the POD reduced model. Thereby the hyper-reduction error with $RE_{hr} = 11.2$ is acceptable.

Both the cantilever example and the c-shape example exhibit large rotations of the elements, as depicted in Figure 6.7. Due to the large rotations, the restoring forces are highly nonlinear, as discussed in Section 2.5. However, since the (U)DEIM requires a split of the linear and the nonlinear part due to stability reasons as discussed in Subsection 12.1.2, the linear forces of the rotated elements are explicitly calculated. They are compensated by the nonlinear forces to yield the internal forces $\boldsymbol{f}$, see (12.9). This splitting of purely linear and purely nonlinear part, however, leads to a cumbersome force splitting which is not natural in the finite element framework of large rotations. It is exemplarily shown

**Table 12.3:** Run times and errors for (U)DEIM reduction of the c-shape example with a reduced basis of 10 POD modes.

| Size $m$ of | DEIM-dof | | DEIM-Node | |
|---|---|---|---|---|
| force basis | $RE_{hr}[\%]$ | $t_w[s]$ | $RE_{hr}[\%]$ | $t_w[s]$ |
| 4 | - | - | - | - |
| 7 | - | - | - | - |
| 10 | - | - | - | - |
| 15 | - | - | - | - |
| 20 | - | - | - | - |
| 40 | - | - | 46.77 | 17.08 |

| Size $m$ of | UDEIM-dof | | UDEIM-Node | | UDEIM-Element | | UDEIM-Component | |
|---|---|---|---|---|---|---|---|---|
| force basis | $RE_{hr}[\%]$ | $t_w[s]$ | $RE_{hr}[\%]$ | $t_w[s]$ | $RE_{hr}[\%]$ | $t_w[s]$ | $RE_{hr}[\%]$ | $t_w[s]$ |
| 4 | - | - | - | - | - | - | - | - |
| 7 | - | - | - | - | - | - | - | - |
| 10 | - | - | - | - | - | - | - | - |
| 15 | - | - | 68.05 | 9.07 | 23.39 | 9.07 | - | - |
| 20 | - | - | 88.36 | 11.03 | 35.21 | 10.9 | - | - |
| 40 | - | - | 11.8 | 15.22 | 11.2 | 14.99 | 17.82 | 15.5 |

with the cantilever example. Therefore, the cantilever example is reduced with a POD using 20 POD bases. The absolute values of the reduced internal forces $\boldsymbol{f}_r(\boldsymbol{q})$, the linearized internal forces $\boldsymbol{K}_r\boldsymbol{q}$ and the purely nonlinear forces $\boldsymbol{f}_{r,nl} = \boldsymbol{f}_r(\boldsymbol{q}) - \boldsymbol{K}_r\boldsymbol{q}$ are depicted in Figure 12.8. Every dot represents one component of the force vector for one time step of the time integration. Since all time steps are plotted in the graph, the order of magnitude of the internal, the linear and the purely nonlinear forces can be estimated.

The maximum values of the linear and purely nonlinear forces are at least one order of magnitude larger than the internal forces. This means, that the splitting of the internal forces in a linear and a nonlinear part leads to very high linear and nonlinear force contributions. They are approximately equal but of opposite sign, so that they result in clearly smaller internal forces when they are added. This is reasonable, since the linearization of large rotations leads to severely distorted elements, as illustrated in Figure 2.4 and Figure 2.6. Similarly, large rigid body rotations of an undistorted element lead to severe linear forces. They have to be compensated with the purely nonlinear forces to obtain the (nonlinear) internal forces of the element.

As a consequence, this splitting leads to high purely nonlinear forces $\boldsymbol{f}_{r,nl}$, where the values one or more magnitudes below determine the magnitude of the internal forces $\boldsymbol{f}_r$. As a consequence, the approximation of the purely nonlinear forces, as it is done in DEIM, has to be very accurate, since fluctuations several orders of magnitude below $\boldsymbol{f}_{r,nl}$ determine the leading values of $\boldsymbol{f}_r$.

This conceptual weakness of all DEIM and UDEIM variants are a downside of the method, since it is hard to estimate the nonlinear forces accurately. However, in the next chapter, a hyper-reduction method is given which avoids most of the issues associated with DEIM.

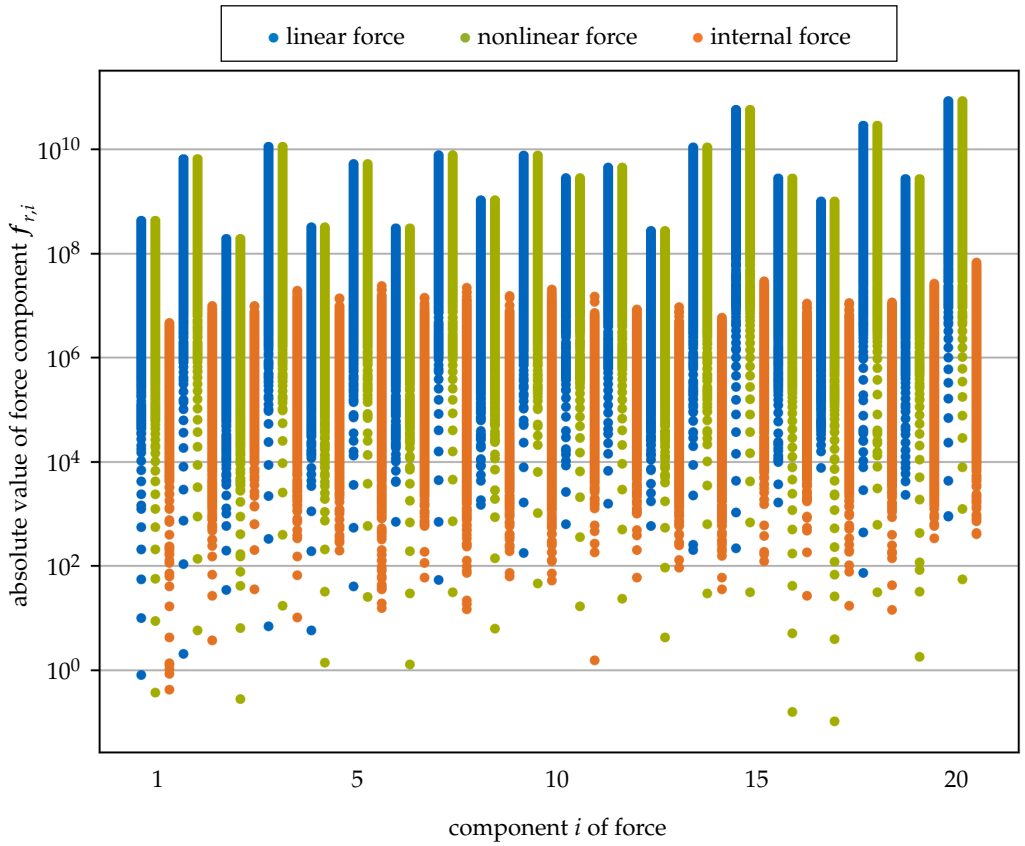


**Figure 12.8:** Absolute values of the reduced linear forces $\boldsymbol{K}_r\boldsymbol{q}$, reduced nonlinear forces $\boldsymbol{f}_{r,nl}(\boldsymbol{q})$ and reduced internal forces $\boldsymbol{f}_r(\boldsymbol{q})$ for the cantilever example using 20 POD modes. Every dot denotes the value of one component for one time step. The internal forces are clearly smaller than both the linear and the nonlinear forces.

# Chapter 13

# Energy Conserving Mesh Sampling and Weighting (ECSW)

The hyper-reduction methods considered in the previous two chapters have two different strategies. The polynomial hyper-reduction in Chapter 11 exploits the polynomial structure of the nonlinear finite element system and expands it using tensors. The computational speedup comes from the fact, that the internal forces can be computed faster using polynomial arrays than evaluating every element in the mesh, when the reduced basis is small enough. The DEIM hyper-reduction introduced in Chapter 12 approximates the internal forces of the high dimensional model by reducing the number of elements in the assembly routine. The computational speedup is due to the reduced mesh, i.e., the nonlinear internal forces are computed by evaluating only a few elements and not the full mesh of the high dimensional model. However, this method does not guarantee stability, since the symmetry is broken due to the approximation strategy based on collocation using an empirical force basis.

In this chapter, the ECSW hyper-reduction method is discussed. The speedup is also realized by reducing the mesh and hence reducing the number of elements to be called. However, this method does not approximate the internal forces in the high dimensional physical space and project them onto the reduced basis afterwards as the DEIM does, but computes the approximation of the reduced forces directly. Within this strategy, the desired properties like symmetry, stability and the existence of an underlying Lagrangian energy function are preserved. As a consequence, this method is among the most promising hyper-reduction methods in the finite element framework.

The ECSW method was first proposed by An et al. [AKJ08] in the context of computer graphics. It has been adapted to engineering applications by Farhat et al. [Far+14; FCA15], where the method is applied to structural dynamics problems using POD bases. Chapman et al. [Cha+17] investigate techniques to speed up the offline costs associated with the solution of the non-negative least square problem, which is the core operation of the ECSW method. Hernandez et al. [HCF17] propose a variation of the ECSW method, in which the nonlinearity is not selectively evaluated on the element level but on the Gauss point level.

Besides the ECSW, a structure preserving hyper-reduction method has been proposed by Carlberg [CTB15]. This method is based on the Gauss-Newton with Approximated Tensors (GNAT) method, see Carlberg et al. [CBF11], however with an adaptation to the Galerkin projection allowing for the preservation of the structure. Since it does not enjoy the simplicity of the ECSW in the finite element context, it is not further discussed in this thesis.

The ECSW needs training sets for the computation of the reduced mesh similar to the DEIM. When the reduced basis is computed using simulation-based methods like the POD, these training sets are commonly the snapshots of the high dimensional model.

However, when a simulation-free reduced basis is used, the training snapshots of the full model are not available. Alternative computations of training sets are proposed in this chapter, after the ECSW is outlined. Parts of this chapter are based on the publication [RR17].

## 13.1 Key Idea: Reduced Quadrature

The ECSW can be seen as a reduced cubature method which preserves the virtual work of the reduced forces for a set of training snapshots. The virtual work done by the reduced internal forces is given as

$$\delta W_{f,r} = \sum_{e \in \mathcal{E}} \delta \boldsymbol{q}^T \boldsymbol{V}^T \boldsymbol{L}_e^T \boldsymbol{f}_e(\boldsymbol{L}_e \boldsymbol{V} \boldsymbol{q}). \tag{13.1}$$

The ECSW is seeking to find an approximation $\delta \widetilde{W}_{f,r}$ for $\delta W_{f,r}$ with a sum over a smaller number of elements. In order to match the energy of the full system and to ensure positiveness, the virtual work of every element is weighted by a weighting factor $\xi_e \geq 0$, similar to a weighting factor of a quadrature method (cf. (2.28)). The approximated virtual work is then written as

$$\delta \widetilde{W}_{f,r} = \sum_{e \in \widetilde{\mathcal{E}}} \xi_e \delta \boldsymbol{q}^T \boldsymbol{V}^T \boldsymbol{L}_e^T \boldsymbol{f}_e(\boldsymbol{L}_e \boldsymbol{V} \boldsymbol{q}), \tag{13.2}$$

with the reduced set of elements $\widetilde{\mathcal{E}}$ containing all elements with a nonzero positive weighting factor $\xi_e$. The balance of virtual work $\delta W_{f,r}$ and approximated virtual work $\delta \widetilde{W}_{f,r}$ is enforced for a set of $m$ reduced training displacements $\boldsymbol{q}_{t,i}$. To cast the balance in a matrix-vector problem, one can collect the summands of (13.2) in the matrix $\mathbf{Y} \in \mathbb{R}^{n \cdot m \times |\mathcal{E}|}$ and the weights $\xi_e$ in $\boldsymbol{\xi} \in \mathbb{R}^{|\mathcal{E}|}$ with $|\mathcal{E}|$ being the number of elements in the element set $\mathcal{E}$. As the virtual displacements $\delta \boldsymbol{q}$ are arbitrary, the balance of approximated virtual work and the virtual work of the full system is written as:

$$\mathbf{Y} \boldsymbol{\xi} = \boldsymbol{b}, \tag{13.3}$$

$$\mathbf{Y} = \begin{pmatrix} \boldsymbol{V}^T \boldsymbol{L}_1^T \boldsymbol{f}_1(\boldsymbol{L}_1 \boldsymbol{V} \boldsymbol{q}_{t,1}) & \dots & \boldsymbol{V}^T \boldsymbol{L}_{|\mathcal{E}|}^T \boldsymbol{f}_{|\mathcal{E}|}(\boldsymbol{L}_{|\mathcal{E}|} \boldsymbol{V} \boldsymbol{q}_{t,1}) \\ \vdots & \ddots & \vdots \\ \boldsymbol{V}^T \boldsymbol{L}_1^T \boldsymbol{f}_1(\boldsymbol{L}_1 \boldsymbol{V} \boldsymbol{q}_{t,m}) & \dots & \boldsymbol{V}^T \boldsymbol{L}_{|\mathcal{E}|}^T \boldsymbol{f}_{|\mathcal{E}|}(\boldsymbol{L}_{|\mathcal{E}|} \boldsymbol{V} \boldsymbol{q}_{t,m}) \end{pmatrix}, \tag{13.4}$$

$$\boldsymbol{b} = \begin{pmatrix} \sum_{e \in \mathcal{E}} \boldsymbol{V}^T \boldsymbol{L}_e^T \boldsymbol{f}_e(\boldsymbol{L}_e \boldsymbol{V} \boldsymbol{q}_{t,1}) \\ \vdots \\ \sum_{e \in \mathcal{E}} \boldsymbol{V}^T \boldsymbol{L}_e^T \boldsymbol{f}_e(\boldsymbol{L}_e \boldsymbol{V} \boldsymbol{q}_{t,m}) \end{pmatrix}. \tag{13.5}$$

The equation above is fulfilled exactly for $\boldsymbol{\xi} = (1, \dots, 1)^T$; this translates (13.2) to (13.1) with $\widetilde{\mathcal{E}} = \mathcal{E}$ and means that every element is contained in the reduced set and weighted with one. In order to achieve a reduction, one tries to find a sparse weighting vector $\boldsymbol{\xi}^\star$ with as many zero entries as possible fulfilling (13.3) up to a tolerance $\tau$. Furthermore, all entries of $\boldsymbol{\xi}^\star$ have to be positive in order to guarantee the positive definiteness of the virtual work of the nonlinear forces. The problem to solve is stated as

$$\arg\min \#(i \mid \xi_i^\star \neq 0) \quad \text{subject to} \quad ||\mathbf{Y}\boldsymbol{\xi}^\star - \boldsymbol{b}||_2 \leq \tau ||\boldsymbol{b}||_2 \quad \text{and} \quad \boldsymbol{\xi}^\star \geq 0. \tag{13.6}$$

This optimization problem is NP-hard to solve, meaning that a solution cannot be sought in polynomial time. However, an approximate solution for $\boldsymbol{\xi}^\star$ can be computed

**Algorithm 5:** Sparse Non-Negative Least Square (sNNLS) Solver

**Input** : matrix $\mathbf{Y} \in \mathbb{R}^{n \cdot m \times |\mathcal{E}|}$, vector $\boldsymbol{b} \in \mathbb{R}^{n \cdot m}$, tolerance $\tau \in ]0, 1[$
**Output** : sparse vector $\boldsymbol{\xi} \in \mathbb{R}^{|\mathcal{E}|}$ with $||\mathbf{Y}\boldsymbol{\xi} - \boldsymbol{b}||_2 \leq \tau ||\boldsymbol{b}||_2$ and $\boldsymbol{\xi} \geq 0$

1 $\mathcal{I} := \{\}$ // Initialize empty active index set $\mathcal{I}$
2 $\boldsymbol{\xi} := \mathbf{0} \in \mathbb{R}^{|\mathcal{E}|}$ // Initialize empty solution vector $\xi$
3 $\boldsymbol{r} := \boldsymbol{b}$ // Initialize residual $\boldsymbol{r}$
4 **while** $||\boldsymbol{r}||_2 > \tau ||\boldsymbol{b}||_2$ // Outer iteration loop
5 **do**
6 $\quad \boldsymbol{\mu} := \mathbf{Y}^T \boldsymbol{r}$ // Compute column error measure $\mu$ of $\mathbf{Y}$
7 $\quad i := \text{index max}(\boldsymbol{\mu})$ // Pick index with maximum value of $\mu$
8 $\quad \mathcal{I} := \mathcal{I} \cup i$ // Update active index set $\mathcal{I}$
9 $\quad$ **while** *True* // Inner loop for enforcing positivity constraint
10 $\quad$ **do**
11 $\quad\quad \boldsymbol{\zeta} := \mathbf{0} \in \mathbb{R}^{|\mathcal{E}|}$ // Initialize trial solution $\zeta$ which should satisfy the positivity constraint
12 $\quad\quad \boldsymbol{\zeta}_{\mathcal{I}} := (\mathbf{Y}_{\mathcal{I}})^{+} \boldsymbol{b}$ // Solve least squares problem on index set $\mathcal{I}$
13 $\quad\quad$ **if** $\min(\boldsymbol{\zeta}) > 0$ // Check positivity constraint
14 $\quad\quad$ **then**
15 $\quad\quad\quad \boldsymbol{\xi} := \boldsymbol{\zeta}$ // Update solution vector $\xi$ with trial vector $\zeta$
16 $\quad\quad\quad$ **break** // exit inner loop
17 $\quad\quad$ **end**
18 $\quad\quad \mathcal{V} := \{i \in \mathcal{I} \mid \zeta_i < 0\}$ // Define index set $\mathcal{V}$ of constraint violation
19 $\quad\quad \alpha := \min\{\xi_i / (\xi_i - \zeta_i) \mid i \in \mathcal{V}\}$ // Compute the step width $\alpha$ necessary
// to set maximum constraint violation to zero.
20 $\quad\quad \boldsymbol{\xi} := \boldsymbol{\xi} + \alpha(\boldsymbol{\zeta} - \boldsymbol{\xi})$ // Correct constraint violation
21 $\quad\quad \mathcal{I} := \{i \mid \xi_i \neq 0\}$ // Update active index set
22 $\quad$ **end**
23 $\quad \boldsymbol{r} := \boldsymbol{b} - \mathbf{Y}_{\mathcal{I}} \boldsymbol{\xi}_{\mathcal{I}}$ // Update residual
24 **end**

with a *sparse Non-Negative Least Square* (sNNLS) solver listed in Algorithm 5. It seeks iteratively for a solution using a greedy method while enforcing the positivity constraint in every iteration. It can be interpreted as a greedy method which runs in the outer loop and a simplex-like algorithm in the inner loop to enforce the positiveness constraint, see Lawson and Hanson [LH95] for more in-depth information. In line 19, the matrix $\mathbf{Y}_{\mathcal{I}}$ is a sub-matrix of $\mathbf{Y}$ defined in (13.4). It is obtained by taking only the columns corresponding to the elements in the active set $\mathcal{I}$.

Figure 13.1 depicts the correction step of the positivity constraint. The vector $\boldsymbol{\xi}$ is conforming with the constraint since it lies in the first quadrant of the $x$-$y$ coordinate system, while the trial vector $\boldsymbol{\zeta}$ is violating the constraint, since the $x$-component is negative. The corrected vector $\boldsymbol{\xi}_{\text{corr}}$ is obtained with the correction factor $\alpha = \xi_i / (\xi_i - \zeta_i)$ and the correction vector $\boldsymbol{\zeta} - \boldsymbol{\xi}$. It is exactly the combination of $\boldsymbol{\xi}$ and the trial vector $\boldsymbol{\zeta}$, for which the positivity constraint is just fulfilled. Since the violated component, the $x$-component in this example, is zero in the corrected vector $\boldsymbol{\xi}_{\text{corr}}$, this component is dropped from the active element set $\mathcal{I}$. Note, that this update of the active element set, which happens in line 21 of Algorithm 5, has to take care for numerical round off errors. For a stable numerical implementation, the update of the index set is better accomplished by removing the index $i$ from the index set $\mathcal{I}$ which determines the minimum value $\alpha$ in line 19.

The most expensive part of the sNNLS algorithm is the solution of the least-square problem in line 12 in Algorithm 5. However, the index set $\mathcal{I}$ of active elements increases very often from one iteration to the other, since one element gets added to it and the

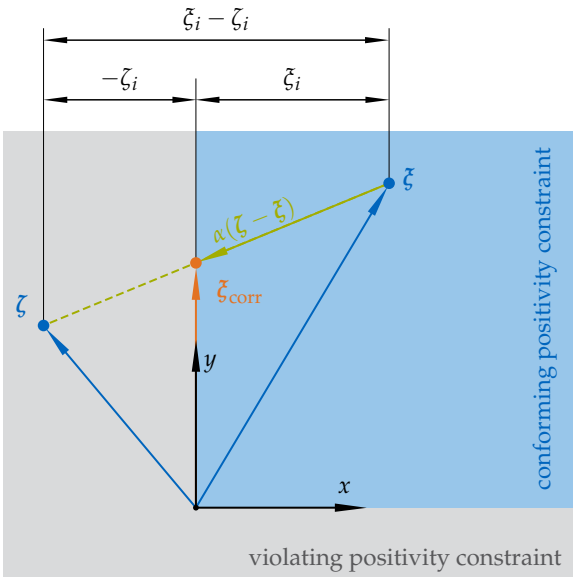


**Figure 13.1:** Illustration of the positivity constraint enforcement. When $\boldsymbol{\zeta}$ is violating the positivity constraint, the correction vector $\boldsymbol{\zeta} - \boldsymbol{\xi}$ is added to $\boldsymbol{\xi}$, such that $\boldsymbol{\xi}_{\text{corr}}$ is exactly zero in the violated coordinate direction $i$.

positivity condition in line 13 in Algorithm 5 is fulfilled. Hence, the information of the previous iteration step can be exploited to solve the least squares problem more efficiently, as proposed by Chapman et al. [Cha+17]. There, further algorithms for solving the problem (13.6) in an approximate fashion are discussed. In particular it is shown that the sNNLS solver equipped with the exploitation of the previous solutions in the least square problem is the best choice in terms of accuracy and computational time.

Once the sNNLS problem is solved and the weighting factors $\xi_e$ are obtained, the reduced element set $\widetilde{\mathcal{E}}$ is defined as

$$\widetilde{\mathcal{E}} = \{e \in \{1, \dots, |\mathcal{E}|\} \mid \xi_e > 0\}. \tag{13.7}$$

The hyper-reduced internal force vector and the tangential stiffness matrix are then given as

$$\boldsymbol{f}_{r,ECSW}(\boldsymbol{q}) = \sum_{e \in \widetilde{\mathcal{E}}} \xi_e \boldsymbol{V}^T \boldsymbol{L}_e^T \boldsymbol{f}_e(\boldsymbol{L}_e \boldsymbol{V} \boldsymbol{q}), \tag{13.8}$$

$$\boldsymbol{K}_{r,ECSW}(\boldsymbol{q}) = \sum_{e \in \widetilde{\mathcal{E}}} \xi_e \boldsymbol{V}^T \boldsymbol{L}_e^T \boldsymbol{K}_e(\boldsymbol{L}_e \boldsymbol{V} \boldsymbol{q}) \boldsymbol{L}_e \boldsymbol{V}. \tag{13.9}$$

In contrast to the (U)DEIM, the tangential stiffness matrix is symmetric. Hence, the ECSW does not require an artificial splitting of linear and nonlinear terms as it is common in the (U)DEIM. Therefore the problem with non-natural non-linear components as discussed in the previous chapter does not appear here. Even better, since the ECSW can be derived from the principle of virtual work, it enjoys a solid theoretical foundation and preserves desirable properties like passivity, stability, symmetry and variational consistency.

However, two questions remain: the choice of an adequate tolerance $\tau$ and the choice of appropriate training sets $\boldsymbol{q}_{t,i}$, for which the equality of the virtual work is enforced. Whilst for the first, a tolerance in the range of $0.0001 \leq \tau \leq 0.01$ is proposed in the literature, see Farhat et al. [FCA13] and Chapman et al. [Cha+17], the choice of training displacements remains a wider question.

## 13.2 Simulation-Based Training Sets

In the case when the reduction method is simulation-based, like the POD and its different flavors, displacement snapshots of the full simulation are available. Therefore it is straightforward to use all or a subset of the snapshots gathered from the training simulation as starting point for the training snapshots for the ECSW method. They have to be projected onto the reduction subspace, since the ECSW balances the virtual work of the reduced forces (13.1) and the hyper-reduced forces (13.2) which are both performing their work in the subspace spanned by $\boldsymbol{V}$ (cf. Farhat et al. [FCA13]). Hence, the training snapshots $\boldsymbol{q}_{t,i}$ can be obtained from the high dimensional displacement vectors $\boldsymbol{u}_{t,i}$ either by projecting them onto the subspace $\boldsymbol{V}$ using the $L^2$-norm as:

$$\boldsymbol{q}_{t,i} = (\boldsymbol{V}^T\boldsymbol{V})^{-1}\boldsymbol{V}^T\boldsymbol{u}_{t,i}, \tag{13.10}$$

or, using the $\boldsymbol{M}$-norm as:

$$\boldsymbol{q}_{t,i} = (\boldsymbol{V}^T\boldsymbol{M}\boldsymbol{V})^{-1}\boldsymbol{V}^T\boldsymbol{M}\boldsymbol{u}_{t,i}. \tag{13.11}$$

In the other case, when the reduced basis is built simulation-free, it is not appealing to pay the cost of a full training simulation. Two methods to obtain the training sets cheaper than using full training simulations are introduced in the subsequent section.

## 13.3 Simulation-Free Training Sets

When the reduced basis is built with a method which does not require a full simulation run and is hence simulation-free, the training snapshots should be obtained without the cost of a full training simulation. This is, however, not an easy task since the training sets should match the displacement fields of the system in the dynamical regime. If the training sets do not match these states, the balance of the virtual work between the hyper-reduced and the reduced-basis system is not enforced at the states of operation, yielding poor hyper-reduced models. Since the deformed states for a nonlinear, dynamical system are hard to guess, the goal of simulation-free or lean simulation approaches is to guess deformed states close to reality for the use as training sets as cheaply as possible.

### 13.3.1 Preliminaries and Previous Approaches

One approach proposed by Jain and Tiso [Jai15] suggests to use quadratically expanded linear displacement fields gathered from a simulation of the linearized system. However, this method works only for structures where the QM approach works as discussed in Chapter 7. The key idea of the QM-lifting approach is to linearize the equations of motion (3.1), reduce them with a reduced basis constructed of vibration modes and perform a linear time integration scheme. Since the linear equations of motion are transformed to the modal space, they are extremely cheap to solve. In order to account for the nonlinearity, the linear solution $\boldsymbol{\eta}$ in modal space is lifted on the quadratic manifold introduced in Section 7.2. The lifted full displacement field $\boldsymbol{u}_{\text{lifted}}$ serves then as training sets for the ECSW. It is expressed as

$$\boldsymbol{u}_{\text{lifted}} = \boldsymbol{V}\boldsymbol{\eta} + \boldsymbol{\Theta}\boldsymbol{\eta}\boldsymbol{\eta} \tag{13.12}$$

with the linear modal displacement field $\boldsymbol{\eta}$ obtained from the linear reduced time integration, the reduced basis $\boldsymbol{V}$ composed of vibration modes and the third order tensor $\boldsymbol{\Theta}$ describing the quadratic expansion of the displacement field with MDs or SMDs, as discussed in Chapter 7.

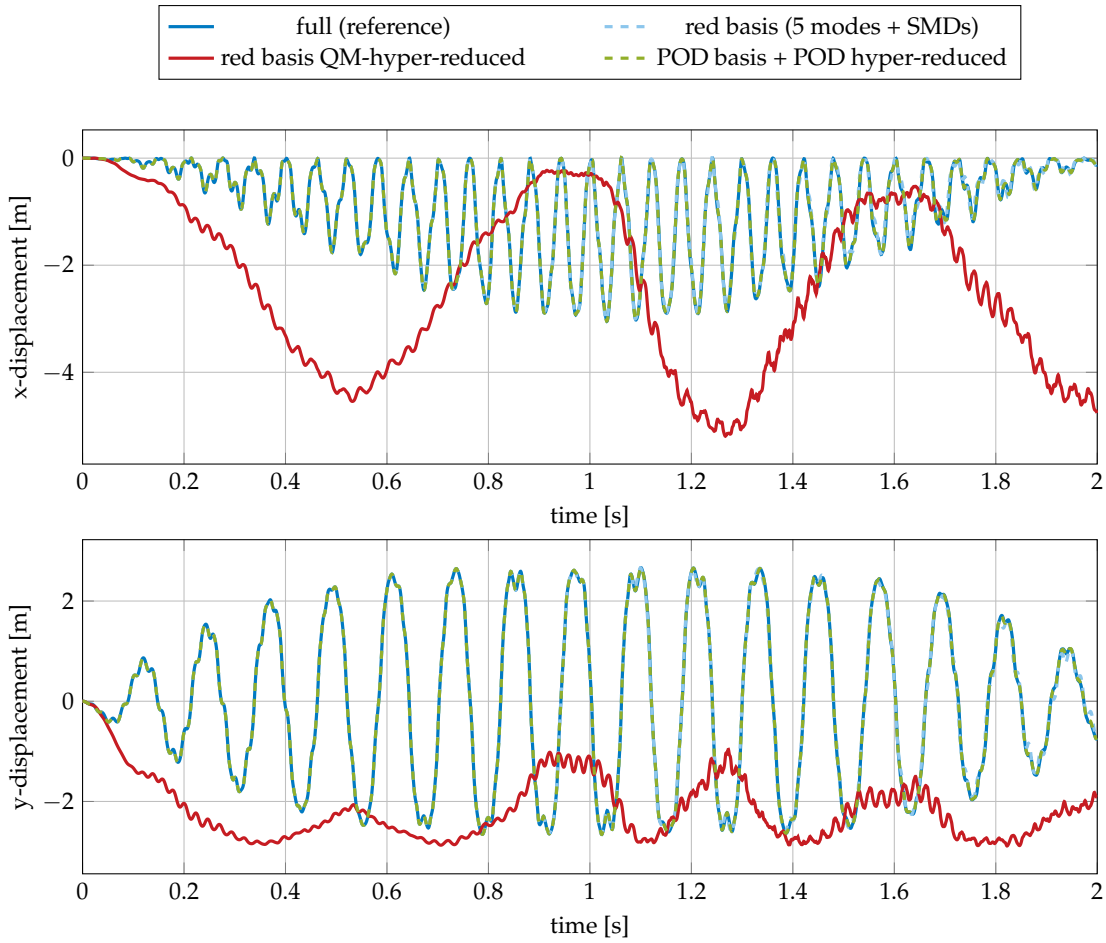


**Figure 13.2:** Tip displacements in the $x$ direction (top) and $y$ direction (bottom) of the cantilever for different reduction and hyper-reduction techniques.

To illustrate the limit of the QM-lifting, it is applied to the geometrically nonlinear cantilever example used in the previous chapters. As discussed in Section 6.4, the modes and the corresponding static derivatives form a suitable basis for the cantilever example. The result of the time integration of a reduced basis composed of five vibration modes plus the corresponding SMDs is depicted in Figure 13.2, where the $x$ and $y$ displacements of the tip are plotted over time. They show a good overlay indicating, that the simulation-free reduced basis suits the problem very well.

Next, the uplifting of the linear solution with modal derivatives is tested. Therefore, the linear training system is built using five vibration modes. Then a linear training simulation is run. The linear solution is lifted onto the quadratic manifold spanned by the modes and the corresponding SMDs according to (13.12). The QM lifted displacements are used as training set for the ECSW with tolerance $\tau = 0.001$. The results of the QM hyper-reduced system are depicted in Figure 13.2 and indicate that the displacements are far from the reference solution and do not capture the motion of the reference system at all.

Finally, a simulation-based ECSW hyper-reduction is tested. The beam model is reduced using a POD basis with 20 basis vectors and an ECSW with the displacement fields gathered in the full simulation. As expected, the results follow the reference very well (cf. Figure 13.2). A displacement snapshot of the deformed beam at $t = 1.0865\,\mathrm{s}$ is depicted in Figure 13.3. Once again, this figure shows that the displacements of the reduced system and of the simulation-based hyper-reduced system follow the reference solution

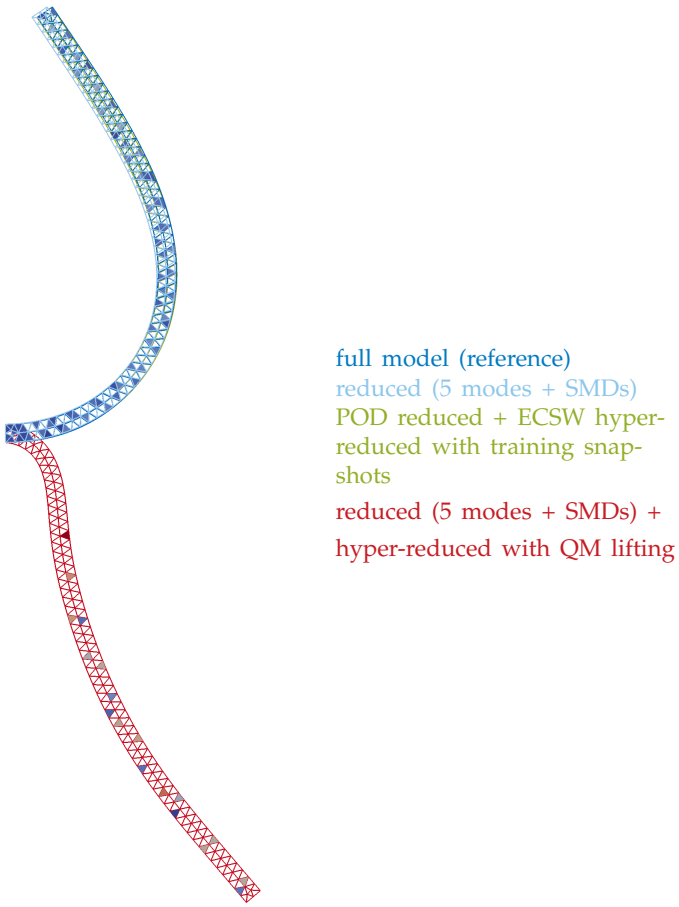


**Figure 13.3:** Displaced configurations of the cantilever at $t = 1.0865\,\mathrm{s}$. The color of the elements indicates the weights of the elements in the ECSW hyper-reduction.

pretty well. The hyper-reduction with the QM-lifting approach is clearly not suitable for this type of geometric nonlinearity. Despite the fact that the computation of training vectors is very cheap in the QM-lifting approach, the results are poor for systems which are not characterized by small, cable-effect driven nonlinearities as necessary for the success of QM approaches (cf. Chapter 7). In the next sections, the Nonlinear Stochastic Krylov Training Sets and the Nonlinear Stochastic Modal Training Sets are proposed which overcome this limitation.

### 13.3.2 Nonlinear Stochastic Krylov Training Sets (NSKTS)

As discussed in the previous section, the training sets used for ECSW should represent displacement fields of the reduced system as accurately as possible. The main issue in obtaining representative displacement fields is that linear superposition of displacement fields does not lead to physical results, as the system behaves nonlinearly.

In the governing equations of motion (3.1), the nonlinear internal forces are triggered by three types of forces: inertia forces which are linear with respect to the accelerations, damping forces which are linear with respect to the velocities and external forces applied to the system. These three types of forces have to be balanced by the (nonlinear) internal forces while adjusting the displacements, velocities and accelerations.

The key idea of the proposed method is to mimic the forces in the equations of motion which trigger the nonlinear internal forces and apply them as a pseudo external forces

to the nonlinear static system. The triggering forces are linear with respect to the displacements and their time derivatives, and can thus be superposed. They are assumed to live in a Krylov-subspace, which is built to represent the external forces $\boldsymbol{g}$ together with approximations of the inertia forces $\boldsymbol{M\ddot{u}}$. They are mixed using stochastic distributions in order to obtain multiple training sets of possible pseudo-dynamic force vectors. The damping forces are neglected in this method.

In the following, the construction of the Krylov force subspace is spelled out similar to the Krylov subspace for displacements in Section 4.4. It is based on the linearized system without damping, where $\boldsymbol{K} = \partial \boldsymbol{f} / \partial \boldsymbol{u}|_{\boldsymbol{u}=\boldsymbol{0}}$ is the linear stiffness matrix for a reference static equilibrium chosen here as $\boldsymbol{u} = \boldsymbol{0}$. The external forces $\boldsymbol{g}(t)$ are assumed to have a time-invariant input location expressed by $\bar{\boldsymbol{g}} \in \mathbb{R}^N$ and the time dependent amplitude $g(t) \in \mathbb{R}$, so that $\boldsymbol{g}$ can be represented by $\bar{\boldsymbol{g}}\, g(t)$. The linearized equations of motion are

$$\boldsymbol{M\ddot{u}} + \boldsymbol{Ku} = \bar{\boldsymbol{g}}\, g. \tag{13.13}$$

The inertia forces $\boldsymbol{M\ddot{u}}$ are unknown and thus represented by a first order force correction $\boldsymbol{h}_1$. The equation is then written as

$$\boldsymbol{h}_1 + \boldsymbol{Ku} = \bar{\boldsymbol{g}}\, g \tag{13.14}$$

and can be solved for the displacements $\boldsymbol{u}$:

$$\boldsymbol{u} = \boldsymbol{K}^{-1}(\bar{\boldsymbol{g}}\, g - \boldsymbol{h}_1). \tag{13.15}$$

If the displacements and the second time derivatives are inserted in (13.13), one obtains

$$\boldsymbol{MK}^{-1}(\bar{\boldsymbol{g}}\, \ddot{g} - \ddot{\boldsymbol{h}}_1) + \boldsymbol{KK}^{-1}(\bar{\boldsymbol{g}}\, g - \boldsymbol{h}_1) = \bar{\boldsymbol{g}}\, g. \tag{13.16}$$

The second derivative $\ddot{\boldsymbol{h}}_1$ of the correction is unknown and thus dropped from the equation in a first order approximation yielding

$$\boldsymbol{h}_1 = \boldsymbol{MK}^{-1}\bar{\boldsymbol{g}}\, \ddot{g}. \tag{13.17}$$

As the acceleration of the external forces $\bar{\boldsymbol{g}}\, \ddot{g}$ lives in the subspace spanned by $\bar{\boldsymbol{g}}$, $\boldsymbol{h}_1$ lives in the subspace

$$\boldsymbol{h}_1 \in \operatorname{span}(\boldsymbol{MK}^{-1}\bar{\boldsymbol{g}}). \tag{13.18}$$

This means that the direction of the unknown inertia forces $\boldsymbol{M\ddot{u}}$ is approximated in the first order by $\boldsymbol{h}_1$. The second order approximation can be obtained similarly, if the acceleration term $\boldsymbol{M\ddot{u}}$ in (13.13) is substituted by the sum of the already known $\boldsymbol{h}_1$ and the unknown second correction term $\boldsymbol{h}_2$:

$$\boldsymbol{h}_2 + \boldsymbol{h}_1 + \boldsymbol{Ku} = \bar{\boldsymbol{g}}\, g \quad \leftrightarrow \quad \boldsymbol{u} = \boldsymbol{K}^{-1}(\bar{\boldsymbol{g}}\, g - \boldsymbol{h}_1 - \boldsymbol{h}_2) \tag{13.19}$$

and inserted into the linearized equations of motion (13.13) yielding:

$$\boldsymbol{MK}^{-1}(\bar{\boldsymbol{g}}\, \ddot{g} - \ddot{\boldsymbol{h}}_1 - \ddot{\boldsymbol{h}}_2) + \boldsymbol{KK}^{-1}(\bar{\boldsymbol{g}}\, g - \boldsymbol{h}_1 - \boldsymbol{h}_2) = \bar{\boldsymbol{g}}\, g. \tag{13.20}$$

Once again, $\ddot{\boldsymbol{h}}_2$ is dropped as it is unknown, $\boldsymbol{MK}^{-1}\bar{\boldsymbol{g}}\, \ddot{g}$ and $-\boldsymbol{h}_1$ cancel out leading to the second order force expansion

$$\boldsymbol{h}_2 = \boldsymbol{MK}^{-1}(-\ddot{\boldsymbol{h}}_1). \tag{13.21}$$

As above, $\ddot{\boldsymbol{h}}_1$ lies in the subspace spanned by $\boldsymbol{h}_1$ and thus $\boldsymbol{h}_2$ lies in the subspace spanned by $\boldsymbol{M}\boldsymbol{K}^{-1}\boldsymbol{h}_1$. The Krylov procedure explained above can be conducted $p-1$ times leading to the Krylov force sequence

$$\boldsymbol{F}_{\text{kry,raw}} = \left(\bar{\boldsymbol{g}}, \boldsymbol{h}_1, \boldsymbol{h}_2, \dots, \boldsymbol{h}_{p-1}\right) = \left(\bar{\boldsymbol{g}}, \boldsymbol{M}\boldsymbol{K}^{-1}\bar{\boldsymbol{g}}, (\boldsymbol{M}\boldsymbol{K}^{-1})^2\bar{\boldsymbol{g}}, \dots, (\boldsymbol{M}\boldsymbol{K}^{-1})^{p-1}\bar{\boldsymbol{g}}\right), \tag{13.22}$$

where $p$ moments of the force subspace around frequency zero are matched. The vector $\bar{\boldsymbol{g}}$ is the low dimensional subspace in which the external forces $\boldsymbol{g}$ act. In the derivation above it was chosen to be a one dimensional subspace. However, it could have been equally derived for higher dimensional subspaces.

The Krylov force subspace is constructed similarly to a Krylov displacement subspace sequence for linear systems, which matches $p$ moments around frequency zero, as discussed in Section 4.4. However, in this case, the Krylov subspace is not a displacement subspace, but a force subspace representing the expansion of the acceleration forces. Likewise, it could have been obtained by multiplying the typical Krylov sequence for displacements $\boldsymbol{V}_{\text{kry}} = \left(\boldsymbol{K}^{-1}\bar{\boldsymbol{g}}, \dots, (\boldsymbol{K}^{-1}\boldsymbol{M})^{p-1}\boldsymbol{K}^{-1}\bar{\boldsymbol{g}}\right)$ with $\boldsymbol{K}$ to obtain the corresponding linear forces of the Krylov displacements.

Krylov sequences are usually poorly conditioned, as the vectors tend to become less linearly independent for higher order corrections (cf. [Str07] and [GR14]). In order to obtain a good basis, an orthogonalization is necessary, as classically performed in an Arnoldi procedure or a Lanczos procedure for symmetric systems respectively. When performing these orthogonalization iterations, a suitable norm has to be defined, for which the Krylov vectors are made orthogonal.

In the case of the Krylov force subspace, a norm allowing for physical interpretation is desirable. Forces applied to linear systems can be measured according to the work they do when applied. The work $W_g$ of the forces $\bar{\boldsymbol{g}}$ applied statically is defined as

$$W_g = \int \bar{\boldsymbol{g}}^T \mathrm{d}\boldsymbol{u} = \int \bar{\boldsymbol{g}}^T \boldsymbol{K}^{-1} \mathrm{d}\bar{\boldsymbol{g}} = \frac{1}{2}\bar{\boldsymbol{g}}^T \boldsymbol{K}^{-1}\bar{\boldsymbol{g}}, \tag{13.23}$$

as the linear displacements are $\boldsymbol{u} = \boldsymbol{K}^{-1}\bar{\boldsymbol{g}}$. The metric expressed in the static energy of force vector $\bar{\boldsymbol{g}}$ is the so-called impedance norm:

$$|\boldsymbol{x}|_{\boldsymbol{K}^{-1}} = \sqrt{\boldsymbol{x}^T\boldsymbol{K}^{-1}\boldsymbol{x}}. \tag{13.24}$$

This norm is chosen for the orthogonalization of the Krylov force sequence (13.22), such that the Krylov force basis is orthogonalized and normalized:

$$\boldsymbol{F}_{\text{kry}}^T \boldsymbol{K}^{-1} \boldsymbol{F}_{\text{kry}} = \boldsymbol{I}. \tag{13.25}$$

With this orthogonalization, every column vector in $\boldsymbol{F}_{\text{kry}} \in \mathbb{R}^{N\times p}$ represents a force vector that does the same work when applied to the static linear system. Or, from the other perspective, the elastic potential energy stored in the mechanical system is equal for every force vector applied.

In order to obtain force vectors mimicking the external and dynamic forces of the system, all column vectors of $\boldsymbol{F}_{\text{kry}}$ representing the subspace of these forces are amplified randomly. The amplification is cast in the $i$-th random vector $\boldsymbol{n}_i \in \mathbb{R}^p$ which carries Gaussian distributed random values:

$$\boldsymbol{n}_i(\mu,\sigma) = \left(\mathcal{N}_1(\mu,\sigma), \dots, \mathcal{N}_p(\mu,\sigma)\right)^T, \tag{13.26}$$

with a Gaussian distributed random number $\mathcal{N}(\mu,\sigma)$ for a given mean $\mu$ and standard deviation $\sigma$. A random force sample $\boldsymbol{f}_{\text{rand},i} \in \mathbb{R}^N$ is then expressed as

$$\boldsymbol{f}_{\text{rand},i} = \boldsymbol{F}_{\text{kry}} \boldsymbol{n}_i. \tag{13.27}$$

The training displacements are computed by solving the nonlinear static problem for the displacements $\boldsymbol{u}_i$ caused by the given random force vector. The force is applied in $k$ steps, for which the load is increased with $\frac{1}{k}$-th of $\boldsymbol{f}_{\text{rand},i}$:

$$\boldsymbol{f}(\boldsymbol{u}_{i,l}) = \frac{l}{k} \boldsymbol{f}_{\text{rand},i}, \quad \text{with} \quad l \in [1, \dots, k]. \tag{13.28}$$

The solution of the nonlinear displacements using a force increment technique leads to $k$ displacement training snapshots for one stochastic force vector $\boldsymbol{f}_{\text{rand},i}$. For $d$ random force vectors, $d \cdot k$ displacement training snapshots are generated. These training snapshots are then used as training sets to hyper-reduce a reduced-basis model by setting up (13.4) and solving (13.6) with an appropriate solution strategy.

As hyper-reduction works upon a reduced basis model for a given $\boldsymbol{V}$ (cf. (13.8)), two options exist for the computation of NSKTS:

1. compute the NSKTS in the physical, unreduced domain by solving full static problems and project the training snapshots $\boldsymbol{u}_t$ afterwards onto the reduced basis using (13.10), and

2. compute the NSKTS in the reduced basis domain of the desired hyper-reduced model and obtain the training snapshots $\boldsymbol{q}_t$ directly (cf. second block of Figure 13.4).

The first option involves the solution of the nonlinear static problem (13.28) in the high dimensional physical domain. As this is a costly procedure, it is desirable to circumvent this costs with the second option where the nonlinear static problem is solved in the domain of the reduced basis and the matrices to be factorized are of size $n$ instead of $N$. Furthermore, it has some additional advantages. First, in the author's experience, the number of iterations inside the Newton-Raphson loop of the nonlinear static solver is smaller than for option one. Second, the training snapshots are directly computed in the desired reduced form. Third, the reduced elemental forces are computed and can be stored for setting up the sNNLS problem (13.4). Hence, the second option is favored and the NSKTS are proposed to be computed in the reduced domain.

An overview of the NSKTS computation procedure is given in Figure 13.4. The amplitude of the input vector $\bar{\boldsymbol{g}}$ for the Krylov force subspace has no effect on the NSKTS, as $\bar{\boldsymbol{g}}$ is scaled to one in the impedance norm. Hence $\boldsymbol{g}$ is evaluated at a random amplitude to only obtain the direction of $\bar{\boldsymbol{g}}$ in the physical domain. Furthermore, the coefficients for the Gaussian distribution have to be specified. For the examples investigated by the author, a Gaussian distribution with mean $\mu = 0$ and with standard deviation $\sigma$ in the range of the impedance norm of the external forces is used. For practical use, the maximum absolute value of the impedance norm of the external force vector is computed and amplified with a correction factor $a$:

$$\sigma = a |\boldsymbol{g}_{\max}|_{\boldsymbol{K}^{-1}}. \tag{13.29}$$

As the external forces are time dependent, $\boldsymbol{g}_{\max}$ is the external force vector for a time, where $\boldsymbol{g}$ has its maximal $L^2$-norm. The correction factor $a$ is to be chosen such that the resulting static displacements $\boldsymbol{u}_i$ obtained by (13.28) are in the range of the expected motion of the system. In the author's practical experience, $a$ should be in the range $[0.1, 5]$, and toward the lower end, if the forces are applied for a short time or at a high frequency and rather high if they are applied at a low frequency or close to an eigenfrequency of the linearized system.

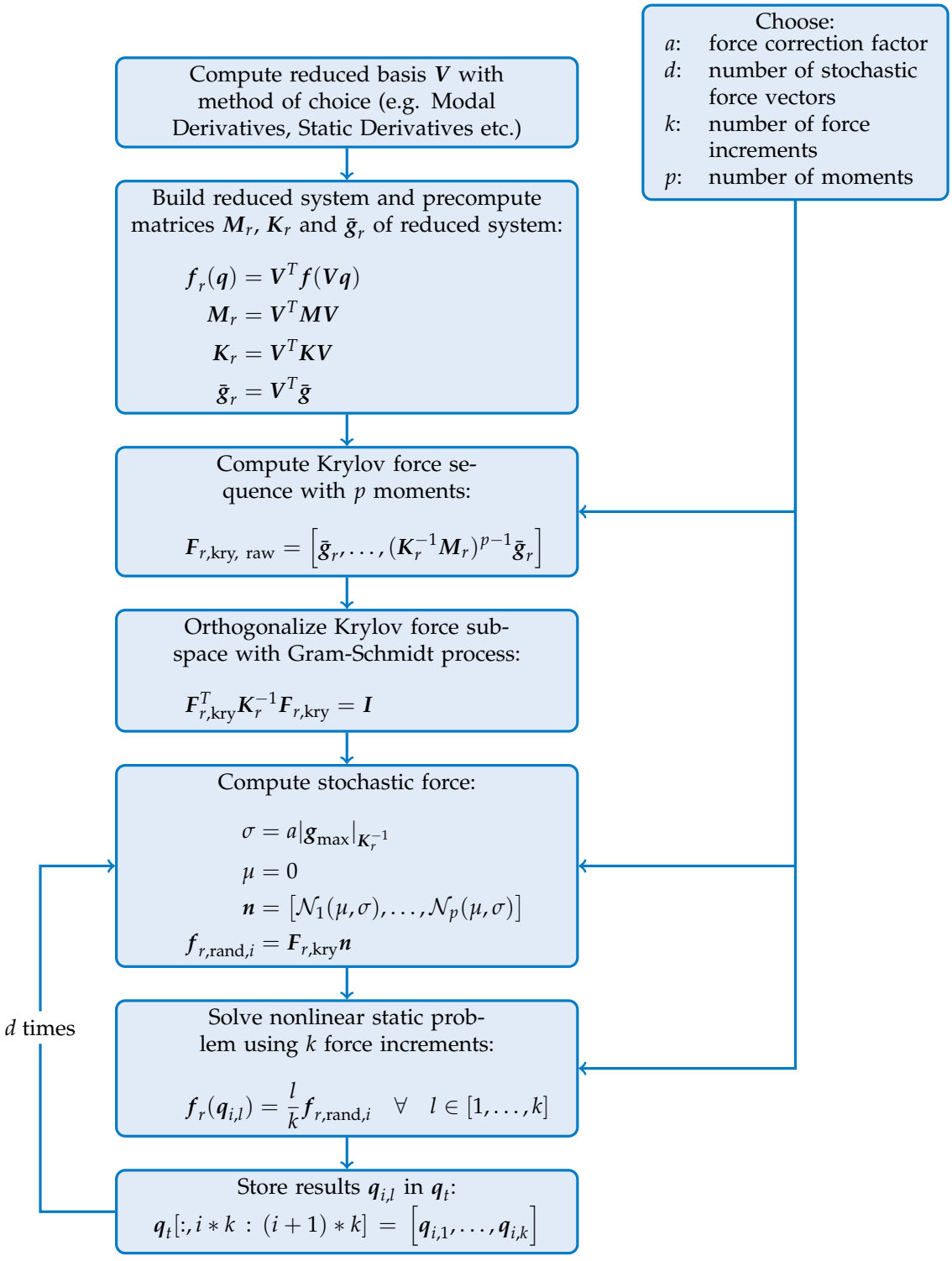


**Figure 13.4:** Computation of the Nonlinear Stochastic Krylov Training Sets (NSKTS)

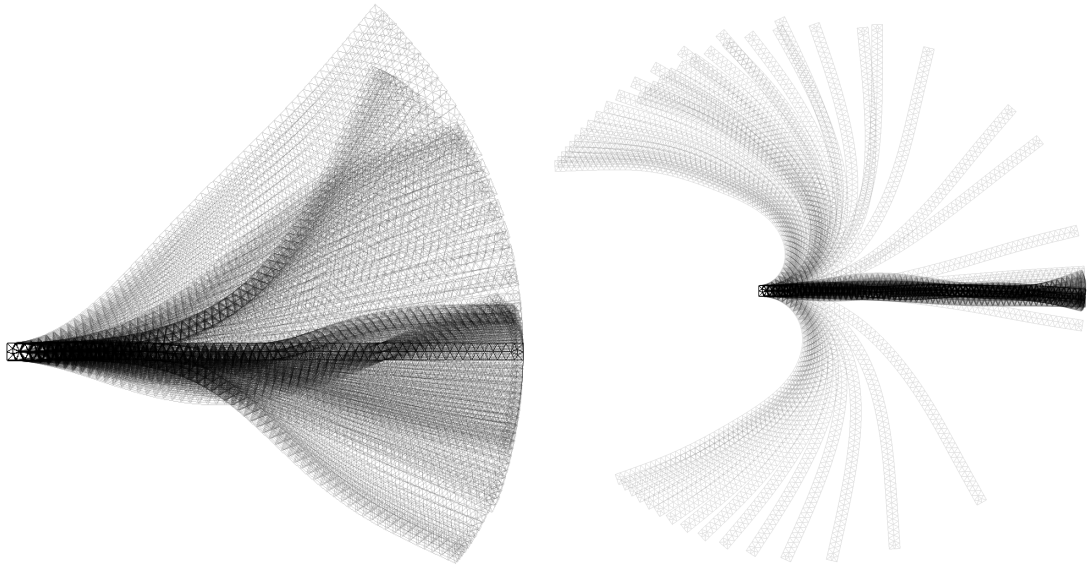

**Figure 13.5:** Overlay depiction of the NSKTS (left) and the NSMTS (right) for the cantilever example.

To illustrate the feasibility of the approach, the proposed method is applied to the cantilever example of the previous section, with which the QM lifting approach is investigated (cf. Subsection 13.3.1). The same reduced basis composed of 5 modes plus corresponding SMDs is used. The NSKTS are computed with the following parameters: the force correction factor $a = 3$, the number of stochastic force vectors $d = 8$, the number of force increments $k = 20$ and the number of moments $p = 4$. The left part of Figure 13.5 depicts the displacement fields of the NSKTS.

The NSKTS are used as training sets for the sNNLS solver of the ECSW with $\tau = 0.001$ leading to a hyper-reduced model depicted in Figure 13.6. The results of the time integration using modes + SMDs as reduced basis and NSKTS for the ECSW are given in Figure 13.7. The results indicate that the NSKTS are very good training sets for the ECSW reduction method. This is further substantiated in the Section 13.4, where a larger, less academic example is given.

### 13.3.3 Nonlinear Stochastic Modal Training Sets (NSMTS)

The NSKTS require the force input locations to obtain the Krylov force subspace, which is the basis for the pseudo dynamic forces. As mentioned in the previous section, the Krylov force subspace can be obtained from the Krylov displacement subspace as introduced in Section 4.4, which is subsequently multiplied with the stiffness matrix of the linearized system. Hence, the Krylov force subspace can be interpreted as the linear static force response to the Krylov displacement subspace. With this interpretation in mind, it is natural to extend the NSKTS technique to other reduction methods like the modal reduction, which does not require force input locations. This is the idea of the Nonlinear Stochastic Modal Training Sets (NSMTS).

The *raw* modal force subspace is defined as

$$\boldsymbol{F}_{r,\text{mod, raw}} = \left(\boldsymbol{K}_r\boldsymbol{\phi}_{r,1}, \ldots, \boldsymbol{K}_r\boldsymbol{\phi}_{r,p}\right) \tag{13.30}$$

with the dimension $p$ of the force subspace. The impedance normalized and orthogonalized modal force subspace $\boldsymbol{F}_{r,\text{mod}}$ could be obtained with a Gram-Schmidt like orthogonalization technique, such that the impedance orthonormality condition

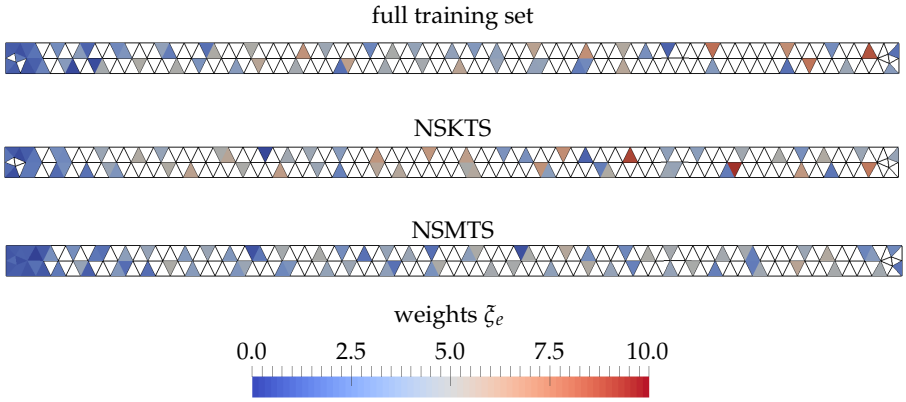


**Figure 13.6:** Weights of the simulation-based ECSW (top) using a basis of 20 POD modes and the lean-simulation ECSW using NSKTS (middle) and NSMTS (bottom) using a basis with 5 vibration modes plus SMDs. In the top mesh, 65 elements are chosen for the reduced element set, in the NSMTS mesh 82 elements, in the NSKTS mesh 62 elements.

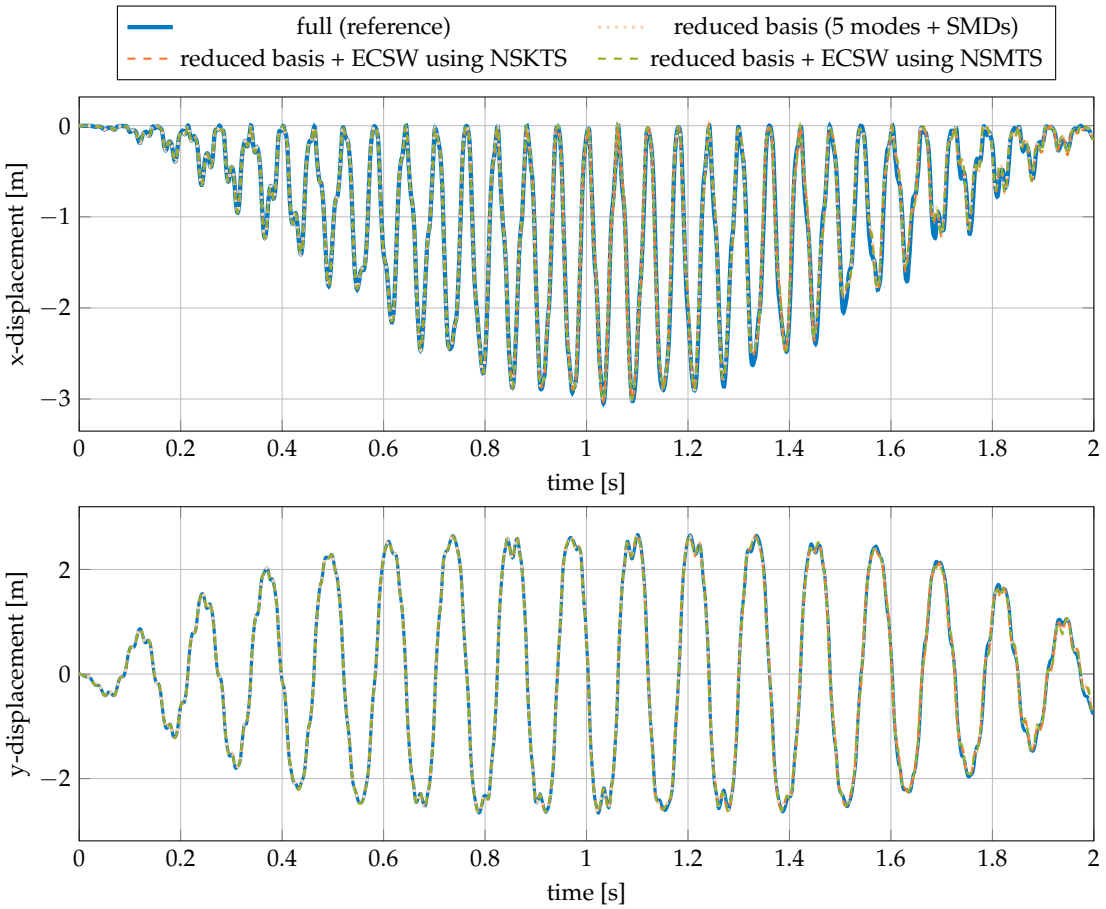


**Figure 13.7:** Tip displacement in the x direction (top) and y direction (bottom) of the cantilever for the hyper-reduced system trained with NSKTS. The reference line is almost overlaid by the reduced and hyper-reduced systems' lines.

$$\boldsymbol{F}_{r,\text{mod}}^T \boldsymbol{K}_r^{-1} \boldsymbol{F}_{r,\text{mod}} = \boldsymbol{I} \tag{13.31}$$

is fulfilled. Then, as in the NSKTS above, the force vectors forming the column vectors in the modal force basis $\boldsymbol{F}_{r,\text{mod}}$ perform the same work when applied to the linear system. However, since the modal force subspace is constructed with modes which are both $\boldsymbol{M}$ and $\boldsymbol{K}$ orthogonal, the raw modal force subspace $\boldsymbol{F}_{r,mod,\text{raw}}$ is impedance orthogonal but not normalized:

$$\boldsymbol{F}_{r,\text{mod, raw}}^T \boldsymbol{K}_r^{-1} \boldsymbol{F}_{r,\text{mod, raw}} = \boldsymbol{\Phi}_r^T \boldsymbol{K}_r \boldsymbol{K}_r^{-1} \boldsymbol{K}_r \boldsymbol{\Phi}_r = \boldsymbol{\Omega}_r^2. \tag{13.32}$$

The matrix $\boldsymbol{\Omega}_r = \text{diag}(\omega_{r,1}, \ldots, \omega_{r,p})$ is a diagonal matrix containing the eigenfrequencies corresponding to the eigenvectors in the raw modal force subspace. Consequently, the impedance norm of the modal force vector is exactly the eigenfrequency of the corresponding mode. The impedance-orthonormalized modal force subspace fulfilling (13.31) is then given as

$$\boldsymbol{F}_{r,\text{mod}} = \left( \tfrac{1}{\omega_{r,1}} \boldsymbol{K}_r \boldsymbol{\phi}_{r,1}, \ldots, \tfrac{1}{\omega_{r,p}} \boldsymbol{K}_r \boldsymbol{\phi}_{r,p} \right) \tag{13.33}$$

or, with the definition of the generalized eigenvalue problem (cf. (4.5)), equivalently as

$$\boldsymbol{F}_{r,\text{mod}} = \left( \omega_{r,1} \boldsymbol{M}_r \boldsymbol{\phi}_{r,1}, \ldots, \omega_{r,p} \boldsymbol{M}_r \boldsymbol{\phi}_{r,p} \right). \tag{13.34}$$

Hence, the Gram-Schmidt orthogonalization can be omitted by a direct weighting of the modes with the inverse of the eigenfrequency to obtain the already orthonormalized modal force subspace $\boldsymbol{F}_{r,\text{mod}}$.

The subsequent steps for the computation of the NSMTS are identical to the ones for the computation of the NSKTS and are depicted in Figure 13.8.

The feasibility of the NSMTS approach is demonstrated with the cantilever example, too. The NSMTS are computed with the number of modes $p = 6$, the force correction factor $a = 1$, the number of stochastic force vectors $p = 8$ and the number of force increments $k = 20$. All static displacement fields converged, so that a total of 160 displacement fields are used as training sets. In the right part of Figure 13.5, the displacement fields of the NSMTS are depicted. In contrary to the NSKTS shown in the left part of the figure, the NSMTS exhibit a stronger over-bending of the beam. This is reasonable, since the fifth mode is an in-plane mode and the corresponding modal forces act in the axial direction of the beam. Depending on the random value and especially the sign of it, the pseudo-dynamic forces may trigger strong nonlinearities as in the over-bending case shown in Figure 13.5.

Next, the reduced cantilever model is hyper-reduced using the NSMTS. The weights of the elements are depicted in Figure 13.6 and indicate, that the number of selected elements to achieve the desired tolerance $\tau = 0.001$ in the sNNLS solver is slightly higher than for the NSKTS and the full training sets. The tip displacement of the NSMTS hyper-reduced model is given in Figure 13.7. It shows clearly that the hyper-reduced model follows the reference very well.

In Table 13.1 the error measures and the computational wall times of the online costs are listed together with the number of selected elements. The computational speedup is associated with the number of elements selected in the active element set. While the NSKTS hyper-reduction needs 62 elements to achieve the desired accuracy in the sNNLS solver and is hence the fastest hyper-reduced model, the NSMTS trained model has 82 elements in the active element set and is hence the slowest hyper-reduced model.

The convergence of the sNNLS solver is depicted in Figure 13.9 for the different training sets. For the full simulation training set, all displacement snapshots from the full,

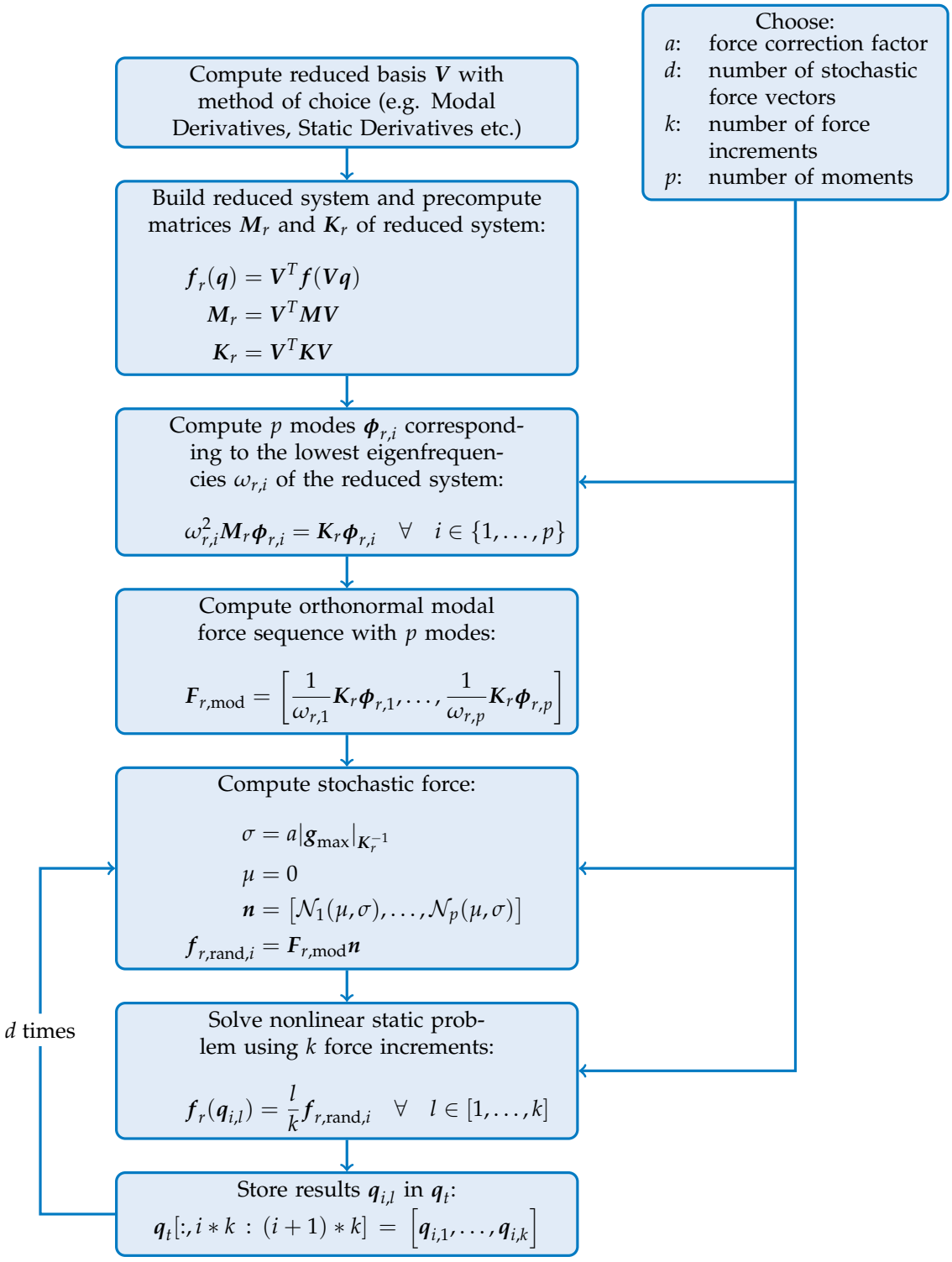


**Figure 13.8:** Computation of the Nonlinear Stochastic Modal Training Sets (NSMTS)

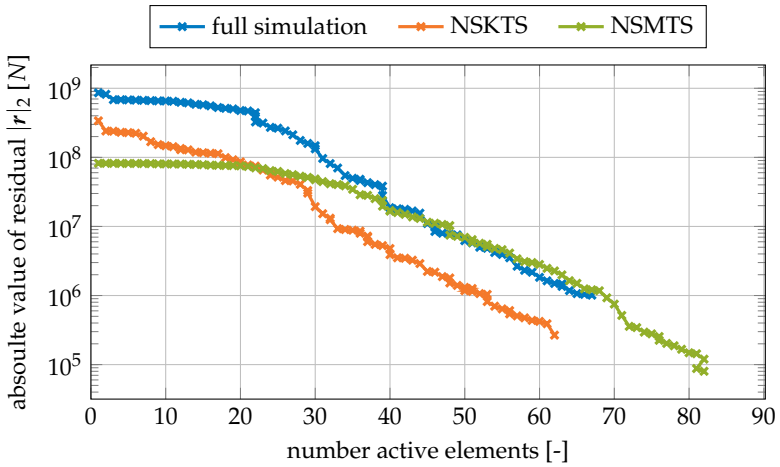


**Figure 13.9:** Residual of the sNNLS solver for the cantilever example. For the full simulation training sets, all displacement snapshots of the full, unreduced simulation are selected as training set.

unreduced time integration are used as training snapshots for the ECSW. Both the maximum residual and the minimum residual after the convergence of the sNNLS are different, since $\tau$ is a relative tolerance (cf. (13.6)). In general, the convergence is strictly monotonic, however with different rates depending on the training sets. Since elements are dropped from the active element set when violating the positivity constraint, the absolute value of the residual drops at certain points while the number of elements remains equal or even decreases.

**Table 13.1:** Accuracy and computational wall times for the ECSW hyper-reduced cantilever example.

| **Reduction Method** | **dofs** | **elements** | $RE_f$ **[%]** | $RE_{hr}$ **[%]** | $t_w$ **[s]** | **speedup [-]** |
|---|---|---|---|---|---|---|
| full | 1224 | 246 | — | — | 71.3 | — |
| POD | 20 | 246 | 0.12 | — | 38.6 | 1.85 |
| modes & SMDs | 20 | 246 | 1.42 | — | 38.5 | 1.85 |
| POD + ECSW (full simulation) | 20 | 67 | 0.2 | 0.12 | 15 | 4.75 |
| modes & SMDs + ECSW (NSKTS) | 20 | 62 | 0.84 | 0.75 | 14 | 5.09 |
| modes & SMDs + ECSW (NSMTS) | 20 | 82 | 1.3 | 0.16 | 18.6 | 3.83 |

## 13.4 Applications

In the previous section, the NSKTS and NSMTS training sets and a full training set stemming from a full, unreduced training simulation are applied to the cantilever example known from the previous chapters. Both, the NSKTS and the NSMTS exhibit excellent hyper-reduction results making them very attractive for real-life applications. To inves-

tigate the NSKTS and NSMTS more in detail, two further applications are given in this section. The first one is a nonlinear cantilever with hyperelastic material. The second is a more realistic rubber-boot example exhibiting a mesh size closer to real-life examples. To investigate if the method is suitable for real-life systems, further investigations regarding the robustness of the NSKTS and the NSMTS are performed.

### 13.4.1 Cantilever with Nonlinear Material

One main feature of the ECSW hyper-reduction and also the NSKTS or NSMTS is the ability to handle nonlinear, hyperelastic materials. The cantilever example of the previous section exhibits clear nonlinearities which are however triggered due to the large rotation of the elements. As a consequence, a material nonlinearity is barely triggered. The cantilever modeled with a Neo-Hookean material exhibits practically the same results as the same beam modeled with St. Venant-Kirchhoff material, if the Lamé constants of both material models are equal. As a consequence, the use of the St. Venant-Kirchhoff material seems to be a valid assumption for nonlinear materials in many cases of nonlinear structural dynamics.

In order to trigger the material nonlinearity in the cantilever, the height of the beam is increased about a factor of 3, so that the strain and stresses are larger at the top and bottom face of the cantilever. The mesh of the modified cantilever example is depicted in Figure 13.10. The nonlinear material is modeled with a Neo-Hookean material as introduced in Section 2.2. The properties of the mesh and the corresponding material values are listed in Table B.1 on page 247 in the Appendix.

Since the modified cantilever example is two dimensional, either a plane stress or a plane strain assumption has to be chosen. Since a plane stress assumption requires the solution of a nonlinear system of equations within the element formulation, the Neo-Hookean cantilever example exhibits a plane strain assumption, so that the stress can be computed in a closed form for every Gauss point.

First, the level of material nonlinearity is investigated. Therefore, the displacements of a full simulation of the modified cantilever modeled with Neo-Hookean material are compared with the same modified cantilever modeled with St. Venant-Kirchhoff material. In Figure 13.11, the tip displacement of the beam is depicted indicating that the nonlinearity of the Neo-Hookean material results in a different motion compared to the St. Venant-Kirchhoff material.

Next a reduced basis is computed. It is built using eight vibration modes and the 22 most important SMDs according to the frequency weighting scheme introduced in Subsection 6.2.4 leading to a reduced basis of 30 basis vectors. In Figure 13.11 the tip displacement of the reduced model is depicted indicating, that the reduced basis suits the material nonlinear problem also well.

Next, simulation-free ECSW reduced models are built using the NSKTS and NSMTS. The NSKTS are computed with the number of moments $p = 4$, the force correction factor $a = 5$, the number of stochastic force vectors $d = 8$ and the number of force increments $k = 20$. The NSMTS are computed with the number of force modes $p = 6$, the force correction factor $a = 2.5$, the number of stochastic force vectors $d = 8$ and $k = 20$ force increments. The sNNLS is run in both cases with $\tau = 0.001$. The error measures, the run times and the number of selected elements of the hyper-reduced models are listed in Table 13.2. They indicate, that the hyper-reduction using the NSKTS and the NSMTS leads to very low hyper-reduction errors $RE_{hr}$ clearly below 1 %. Since the reduced basis is not too accurate, the full error of the hyper-reduced models is in the range of the reduced basis model. For comparison, a POD reduced basis model as well as a POD ECSW hyper-reduced model trained with the full displacement sets are also listed in Table 13.2. They exhibit lower errors compared to the full reference simulation, since the

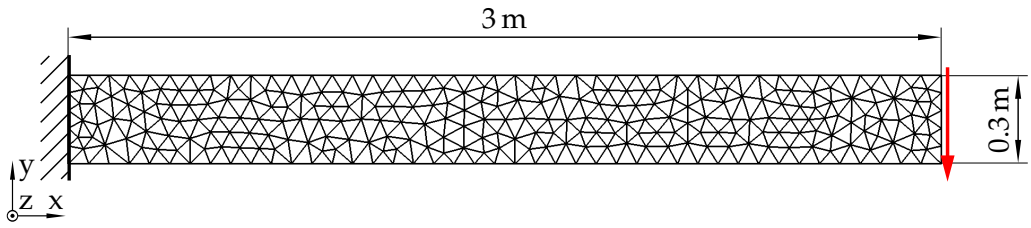


**Figure 13.10:** Mesh of the cantilever used for the Neo-Hookean material.

POD basis is more accurate than the basis of same size composed of the vibration modes and the SMDs. This fits the expectations, since the POD basis knows the results as it is trained by the full solution. Similarly, the ECSW is also very accurate since it knows the full solution as well. More interestingly, the NSKTS produce a hyper-reduction error in the same range of accuracy without knowing the full solution. Even better, the number of elements necessary for achieving this accuracy is lower compared to the full trained ECSW leading also to lower computational wall times and hence greater speedup factors. As a consequence, the NSKTS can be regarded as excellent training sets. In this case, the NSMTS do not exhibit the high accuracy of the NSKTS. This is presumably due to the fact, that the in-plane modal forces cause large over-bendings of the modified cantilever leading to unrealistic training sets. As a consequence, both the hyper-reduction error and the number of elements in the active elements set is higher.

**Table 13.2:** Accuracy and computational wall times for the ECSW hyper-reduced cantilever example with Neo-Hookean material.

| **Reduction Method** | **dofs** | **elements** | $RE_f$ **[%]** | $RE_{hr}$ **[%]** | $t_w$ **[s]** | **speedup [-]** |
|---|---|---|---|---|---|---|
| full | 2244 | 518 | — | — | 142.4 | — |
| POD | 30 | 518 | 0.011 | — | 77.9 | 1.83 |
| modes & SMDs | 30 | 518 | 3.361 | — | 77.6 | 1.83 |
| POD + ECSW (full training) | 30 | 209 | 0.088 | 0.091 | 38.9 | 3.66 |
| modes & SMDs + ECSW (NSKTS) | 30 | 172 | 3.416 | 0.066 | 33.9 | 4.20 |
| modes & SMDs + ECSW (NSMTS) | 30 | 223 | 3.722 | 0.376 | 43.1 | 3.30 |

### 13.4.2 Rubber Boot

Next, the less academic rubber boot example depicted in Figure 13.12 is investigated. The boot is 440 mm high and has a radius of 250 mm at its greatest extent. The boot is composed of two different materials: the top ring is made of steel and modeled as St. Venant-Kirchhoff material with Young's modulus $E = 210\,\text{GPa}$, Poisson's ratio $\nu = 0.3$ and density $\rho = 10,000\,\text{kg/m}^3$; the boot itself is made of polyethylene also using a St. Venant-Kirchhoff Material with Young's modulus $E = 200\,\text{MPa}$, Poisson's ratio $\nu = 0.3$ and density $\rho = 1000\,\text{kg/m}^3$. The boot is fixed on the bottom ring-shaped surface and has a traction force at the rubber top surface of $F = \begin{pmatrix}0 & 1 & 1\end{pmatrix}^T \sin(14 \cdot 2 \cdot \pi)\, 2 \cdot 10^5\,\text{N/m}^2$ acting in both the $y$ and $z$ directions. Rayleigh damping is applied so that the first two

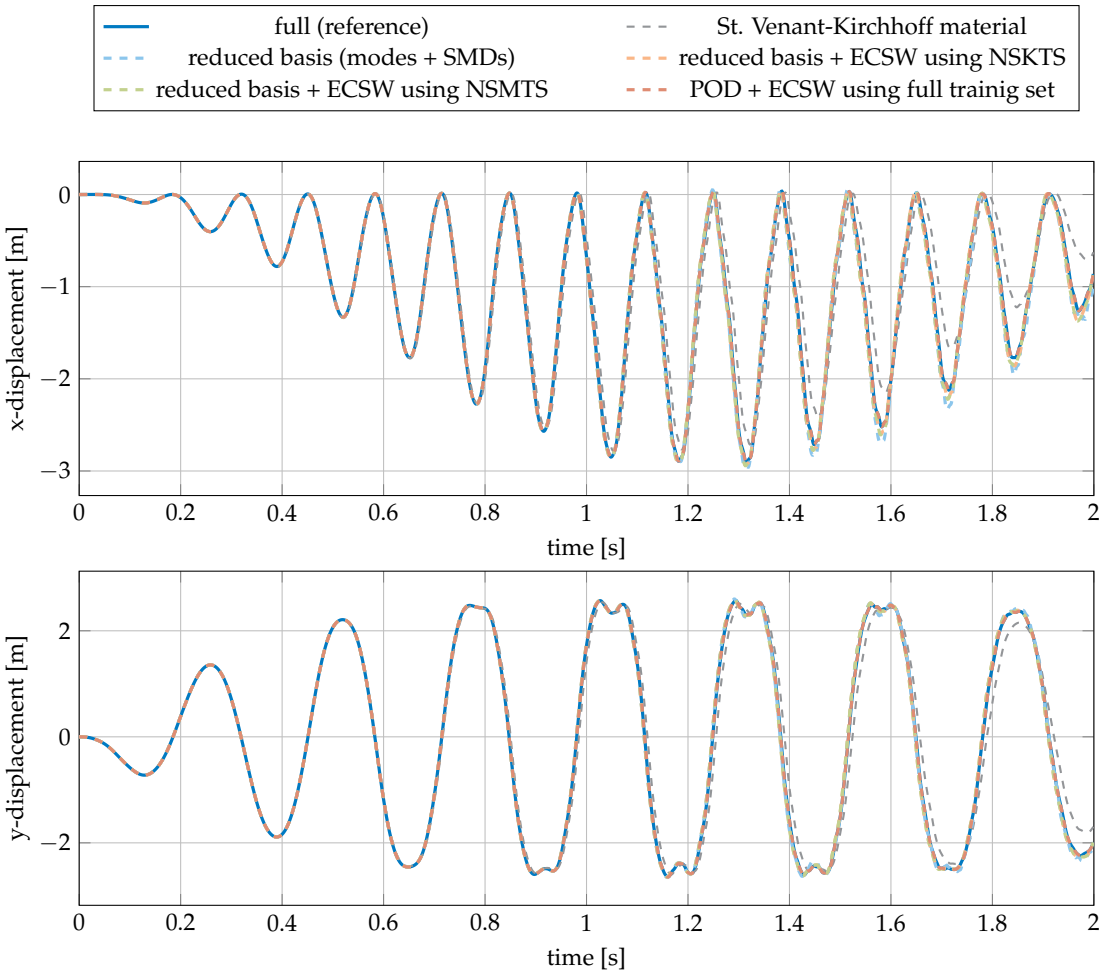


**Figure 13.11:** Tip displacement of the thick cantilever example depicted in Figure 13.10 with Neo-Hookean material.

distinct eigenfrequencies of 12.25 Hz and 18.14 Hz of the linearized system have a linear modal damping ratio of $\delta = 0.01$.

The boot is discretized with a total of 9630 Hexahedron elements with quadratic shape functions. Each element has 20 nodes, the constrained full model has 186,030 dofs in total.

First, a simulation-free reduction basis is computed. It is constructed using the first 25 Krylov subspace modes (cf. Section 4.4) using the force input location at the top ring as external forcing vector $\boldsymbol{g}$ and the corresponding SDs (cf. Subsection 6.2.2). The basis is deflated as described in Subsection 6.2.3 yielding a basis consisting of 346 basis vectors. At this point it is mentioned that the construction of the reduction basis could also have been performed with vibration modes instead of Krylov subspace vectors as linear basis starting points. However, the computation of MDs were not as straightforward as for SDs, as the system is symmetric and has identical eigenvalues. This has to be accounted for in case MDs are used for the simulation-free reduction basis construction.

Next, the NSKTS are computed. In total $d = 8$ stochastic force vectors are chosen with $k = 20$ force increments each. For the consideration of the dynamic effects, $p = 4$ moments are chosen; the external forces are corrected by a factor $a = 2$, so that the maximum external force vector used for the computation of the standard deviation is 2 times higher than $\boldsymbol{g}_{\text{max}}$. In the nonlinear static solver, the number of Newton-Raphson iterations is limited to 20 iterations. If convergence is not achieved within these iterations, the static solution procedure is stopped and all obtained training snapshots up to this

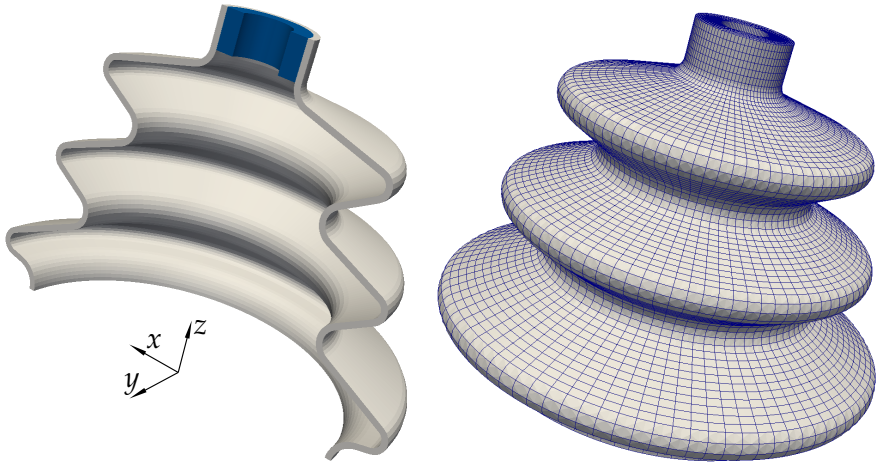


**Figure 13.12:** Cross section (left) and mesh (right) of the benchmark example rubber boot with the $x$-$y$-$z$ coordinate system.

point are used as training sets. In the example, one nonlinear static solution is terminated after 11 of 20 load increments, so that a total of 151 training snapshots are generated. Based on these training sets, the ECSW element selection routine is run using a standard sNNLS solver with $\tau = 0.01$. It selects 1267 elements out of 9630 elements using 1391 iterations. Both the obtained NSKTS and the selected elements of the ECSW scheme are depicted in Figure 13.13.

Also NSMTS are computed. There, similar to the NSKTS, $d = 8$ stochastic force vectors are used with $k = 20$ force increments each. Every stochastic force vector is composed of $p = 6$ modal forces, which are not artificially amplified since the external force correction factor $a = 1$. The number of iterations of the Newton-Raphson solver was also set to a maximum of 20 iterations; all of the eight force cases converged within these number of iterations so that in total 160 training snapshots are generated. Similar to the NSKTS, the sNNLS routine is run with $\tau = 0.01$ resulting in 1219 active elements using 1336 iterations. Both the NSMTS displacement shapes and the resulting selected elements are depicted in Figure 13.13.

For comparison, a simulation-based reduction is performed using POD and classical ECSW as proposed in [FCA13]. In general, the applied forces of the reduced model are not identical to the training simulation. To make the comparison more realistic, the training simulation is *slightly* modified by changing the excitation frequency from 14 Hz of the reference model to 21 Hz used for the training model. All other properties of the system remain similar, though. For the POD analysis, a smaller and a larger reduction base are chosen. The small POD basis consists of 100 POD vectors, the large POD basis has 346 POD vectors and thus the same size as the simulation-free basis.

The time integration of all systems is performed using an HHT-$\alpha$ scheme with numerical damping $\alpha = 0.1$. The time range goes from $t_0 = 0\,\mathrm{s}$ to $t_{\mathrm{end}} = 0.3\,\mathrm{s}$ with a time step size $\Delta t = 5 \cdot 10^{-4}\,\mathrm{s}$ yielding 600 time steps in total.

The time integration's resulting displacements are depicted in Figure 13.14. It shows the displacements in the $x$, $y$ and $z$ directions of a node at the tip of the boot. The solution of the linearized problem is also given to show the degree of nonlinearity of the problem.

As the figure shows, both reduced basis models, the POD basis with 100 basis vectors as well as the simulation-free reduced basis model lead to accurate reduced models that follow the full solution well. Consequently both the subspace spanned by the POD vectors as well as the Krylov subspace modes plus its SDs embrace the subspace of the solution. Also all hyper-reduced models, the model trained with the full solution as well as

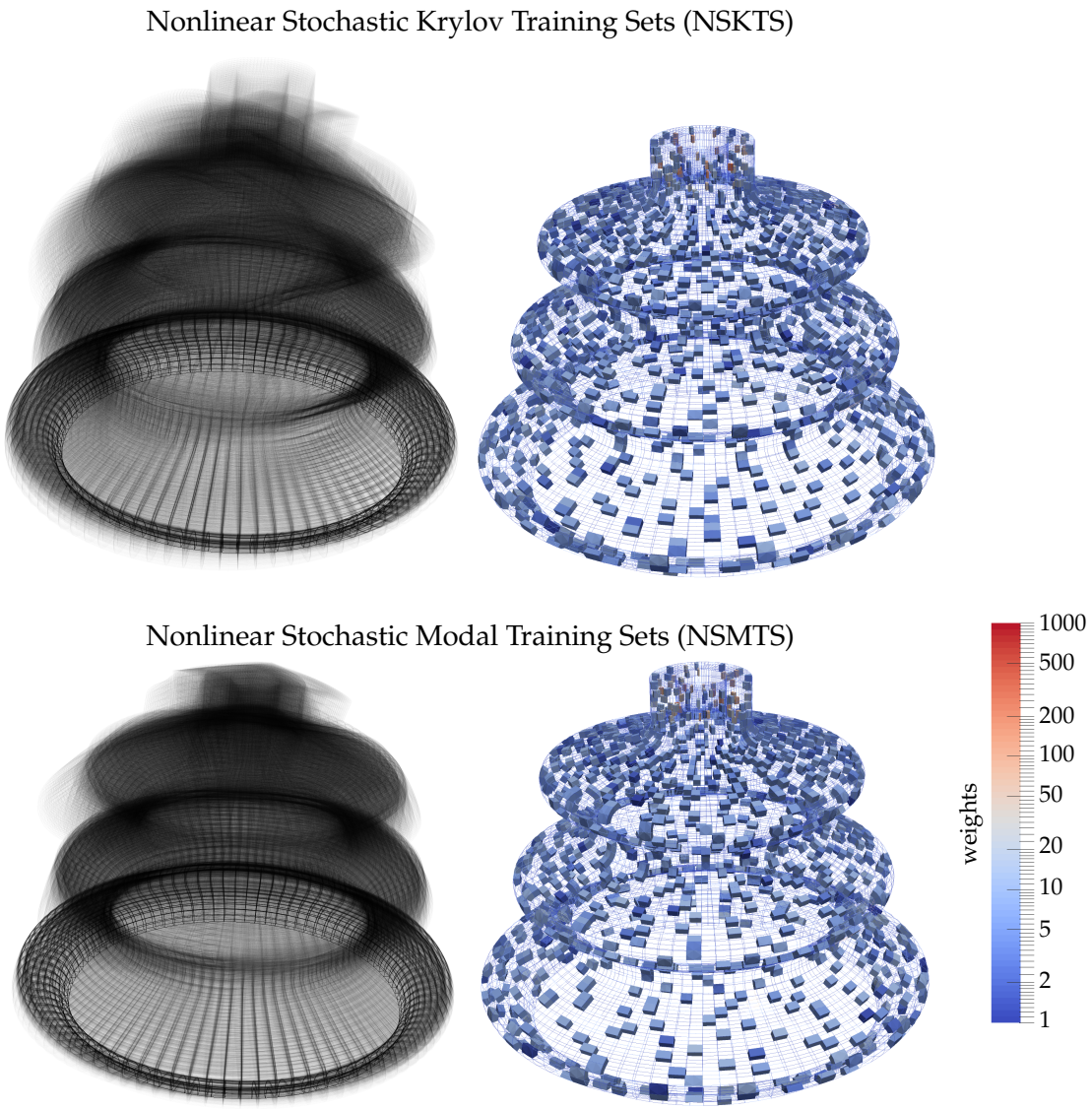


**Figure 13.13:** Overlay picture of the Nonlinear Stochastic Training Sets (left) and the corresponding reduced mesh (right) after the ECSW reduction. The top row shows the NSKTS, the bottom row the NSMTS. The highest element weights are in the steel ring for both cases.

the models trained with NSKTS and NSMTS, show a very good match with the reduced and full solutions.

An overview of the global error, time measures and number of dofs and selected elements is given in Table 13.3. The speedup factors indicate that the reduced basis models allow only moderate accelerations of the computation. Especially for a higher number of reduced dofs, the projection matrices $\boldsymbol{V}$ are very memory-intensive and slow down the computation considerably. The gain of the reduction of the dofs is almost lost, as the assembly routine and the basis projection are too expensive. Hence, hyper-reduction techniques are necessary to obtain real speedups. They are considerable especially for the small POD-hyper-reduced model. The combination of small reduced bases and a small element set shows a great achievement in time reduction. This speedup can not be realized with the larger bases, though the large POD basis yields a faster model than the simulation-free hyper-reduced models trained with NSKTS and NSMTS, as the former has fewer elements in the active element set.

A closer look at the selected elements of the hyper-reduced models trained with the full simulation shows that the selection of the elements is very problem-specific. They

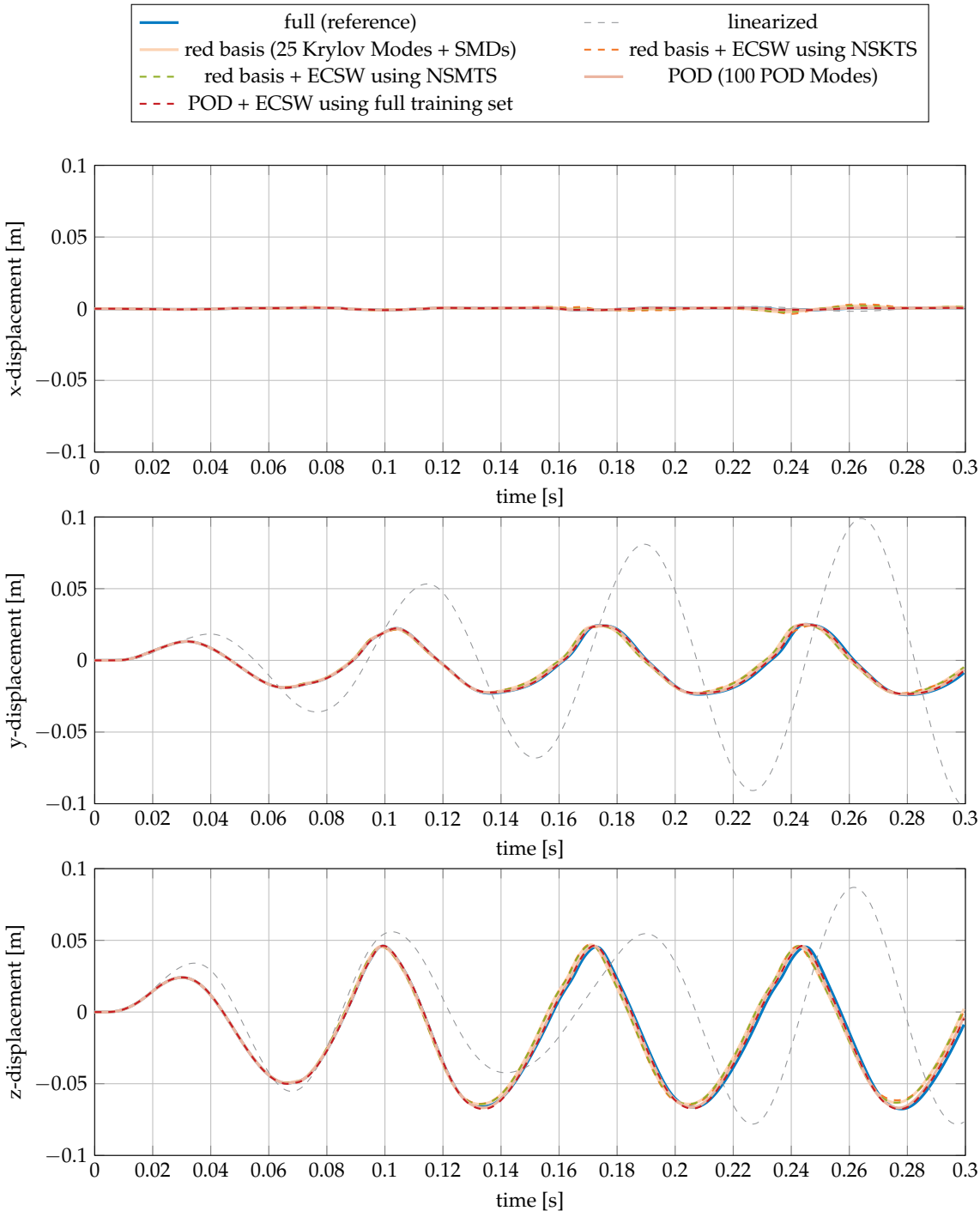


**Figure 13.14:** Displacements of a node on the top ring of the rubber boot for the full model, the reduced model using 25 Krylov subspace vectors plus corresponding SDs and the hyper-reduced models using the same basis and NSKTS or NSMTS for the ECSW training.

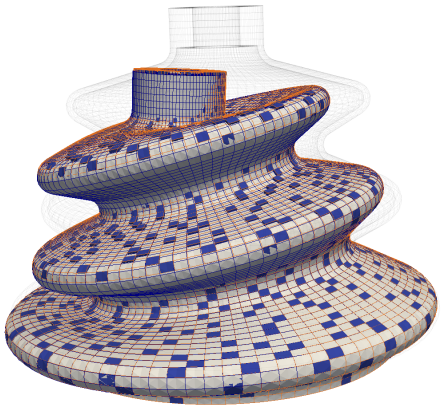

**Figure 13.15:** Deformed configuration of the rubber boot at $t = 0.206\,\mathrm{s}$. The grey solid body is the reference, the orange mesh depicts the mesh of the NSKTS hyper-reduced model with its selected elements. The light grey mesh gives the undeformed configuration.

are predominantly selected on the x-z plane, which can be clearly seen in Figure 13.16. The $x$-$z$ plane is the plane in which the external forces act. In contrast to that, the NSKTS and NSMTS-trained simulation-free models do not show any concentration of elements in a specific direction (cf. Figure 13.13). Instead in these models clearly more elements are selected which seem to be equally distributed. As more elements are selected, the speedup rates are not as high as in the POD hyper-reduced models trained with the full simulation.

However, the larger element set in the NSKTS-trained model results in a higher accuracy. The $RE_{hr}$ error, which gives the error of the hyper-reduced model with respect to the reduced-basis counterpart, is clearly smaller for both the NSKTS and NSMTS-trained model compared to both full simulation-trained POD models. For the task of building a hyper-reduced mesh for a given basis, the NSKTS and the NSMTS prove to be excellent training sets, but with a tendency to build more accurate and thus more expensive hyper-reduced models than the models trained with full simulation training sets. As several setting parameters have to be chosen to compute the NSKTS and the NSMTS, the robustness of the method is of great interest and hence investigated in the following subsection.

### 13.4.3 Robustness of the Method

To assess the robustness of the reduced models generated above, first the hyper-reduced models built with the nominal force are simulated with a higher force amplitude. The amplitude is increased by a factor of 3, while all other parameters are kept the same.

Similar to the previous case, the resulting displacements of the tip node are given in Figure 13.17 and the quantitative values are listed in Table 13.4. As the error of the hyper-reduction step is given with the $RE_{hr}$ value, the NSMTS-trained hyper-reduced model clearly has the lowest hyper-reduction error. Also the NSKTS-trained hyper-reduced model has a hyper-reduction error clearly below the full simulation trained POD hyper-reduced model. Again, this accuracy comes with the cost of a lower speedup factor compared to the full simulation-trained POD hyper-reduced models. In this example, however, the reduced basis of both the NSKTS and NSMTS hyper-reduced models does not suit the problem well, and hence the $RE_f$ is fairly high. But also the small POD system, which had a smaller global error than the NSKTS and NSMTS hyper-reduced model in the load case of Subsection 13.4.2, does not capture the relevant subspace of the problem and performs even worse. The great problem-specificity resulting in small and efficient

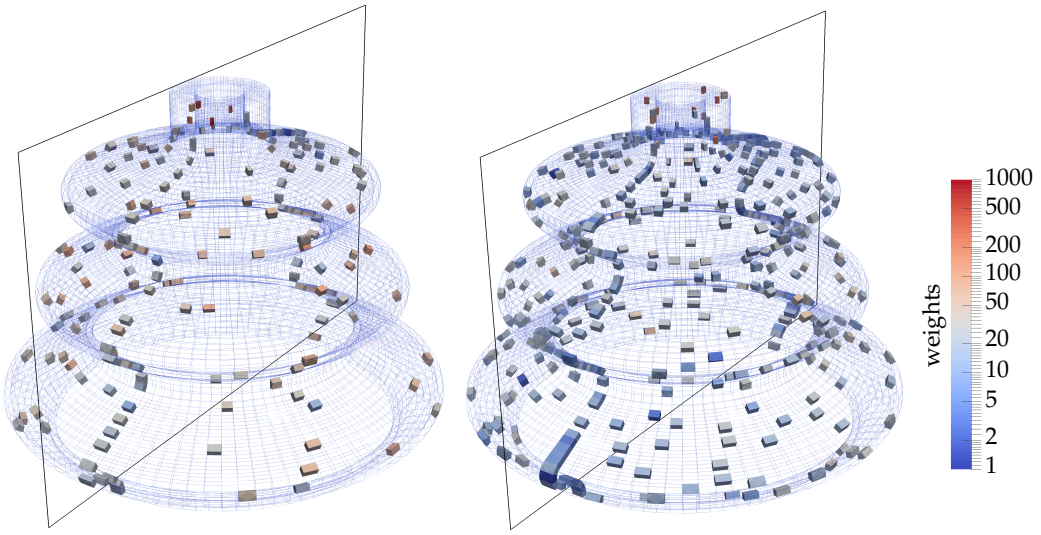


**Figure 13.16:** Weights of the POD-reduced rubber boot with 100 basis vectors (left) and 346 basis vectors (right). The plane depicts the x-z plane, which is the plane of the external forces; the selected elements are predominantly located on this plane.

**Table 13.3:** Overview of the different reduction scenarios for the rubber boot example.

| **Reduction method** | **dofs** | **elements** | $RE_{f/rb}$ **[%]** | $RE_{hr}$ **[%]** | $t_w$ **[s]** | **speedup [-]** |
|---|---|---|---|---|---|---|
| Full problem | 186,030 | 9630 | — | — | 23,293 | — |
| Reduced with 25 Krylov Modes + SDs | 346 | 9630 | 12.50 | — | 22,222 | 1.05 |
| Hyper-reduced with NSKTS | 346 | 1267 | 12.28 | 0.29 | 4,998 | 4.66 |
| Hyper-reduced with NSMTS | 346 | 1219 | 12.44 | 0.07 | 4,906 | 4.75 |
| POD-reduced | 100 | 9630 | 5.74 | — | 16,051 | 1.45 |
| POD-hyper-reduced | 100 | 183 | 4.33 | 2.10 | 757 | 30.77 |
| POD-reduced | 346 | 9630 | 0.50 | — | 22,317 | 1.04 |
| POD-hyper-reduced | 346 | 418 | 1.17 | 1.35 | 3,109 | 7.49 |

**Table 13.4:** Overview of the different reduction scenarios for the Rubber Boot example with increased forcing about factor 3.

| Reduction method | dofs | elements | $RE_{f/rb}$ [%] | $RE_{hr}$ [%] | $t_w$ [s] | speedup [-] |
|---|---|---|---|---|---|---|
| Full problem | 186,030 | 9630 | — | — | 24,947 | — |
| Reduced with 25 Krylov Modes + SDs | 346 | 9630 | 24.33 | — | 25,880 | 0.96 |
| Hyper-reduced with NSKTS | 346 | 1267 | 24.24 | 0.18 | 5,832 | 4.28 |
| Hyper-reduced with NSMTS | 346 | 1219 | 24.31 | 0.04 | 5,833 | 4.28 |
| POD-reduced | 100 | 9630 | 51.05 | — | 18,702 | 1.33 |
| POD-hyper-reduced | 100 | 183 | 51.01 | 0.92 | 879 | 28.38 |
| POD-reduced | 346 | 9630 | 9.04 | — | 26,281 | 0.95 |
| POD-hyper-reduced | 346 | 418 | 8.62 | 0.84 | 3,668 | 6.80 |

reduced models has its flip-side in the lack of robustness for different load cases or a change of boundary conditions. However, as the $RE_{hr}$ error is fairly small for all hyper-reduced models, the hyper-reduction step worked out well in all cases. It seems that for building good hyper-reduced models using ECSW, the main challenge is the construction of a suitable reduced basis.

In a second investigation, the NSKTS algorithm is inspected for the robustness of its setting parameters. Hence, the setting parameters are both decreased and increased, NSKTS are computed and hyper-reduced models with the same basis as above are trained with them. The four setting parameters are

- the force correction factor $a$ amplifying the external forces,
- the number of stochastic distributions $d$,
- the number of force increments $k$ for the nonlinear solver and
- the number of moments $p$ up to which the Krylov force sequence is constructed.

An overview of the NSKTS with varied setting parameters and the corresponding time integration results is given in Table 13.5. Despite the variations of all setting parameters, the resulting errors, both $RE_f$ and $RE_{hr}$ show very little fluctuation. Similarly, the number of selected elements, and with it the wall time and the speedup factor, remain in the same range. In the author's experience, the variations stay in the same range as the variations stemming from the random number generation in (13.26). As a consequence, NSKTS prove to be training sets which are very robust with respect to their setting parameters. In a practical user setting, they can be chosen intuitively without requiring any fine-tuning.

The last two rows in Table 13.5 show how the system behaves if the size of the basis is reduced or the tolerance is loosened. If the reduced basis is chosen to have only 100 dofs, the number of elements decreases by about a factor of $\frac{1}{2}$ leading to higher speedup rates,

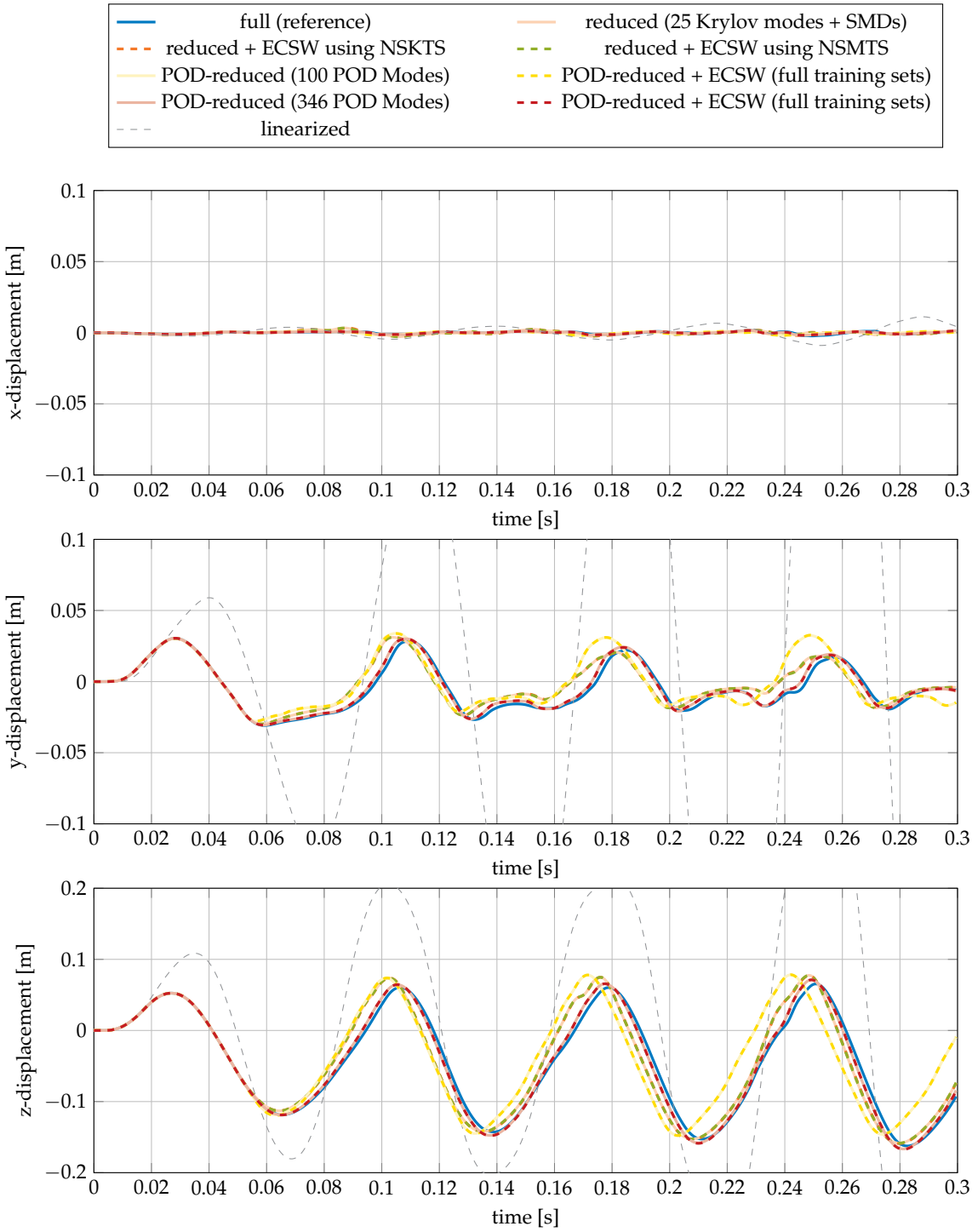


**Figure 13.17:** Displacements of a node on the top ring of the rubber boot with triple the loading.

**Table 13.5:** Overview of the variations of the setting parameters for the NSKTS and their results on the reduction accuracy. As a random number generation is involved in the NSKTS computation, the numbers show just the trend. $a$: force correction factor, $d$: number of stochastic force vectors, $k$: number of force increments, $p$: number of moments. The last two rows describe experiments, where first the basis is chosen to have only 100 basis vectors instead of 346 and lastly the tolerance $\tau$ of the sNNLS-solver was set to 0.1.

| **Name** | $a$ | $d$ | $k$ | $p$ | **elements** | $RE_f$ **[%]** | $RE_{hr}$ **[%]** | $t_w$ **[s]** | **speedup [-]** |
|---|---|---|---|---|---|---|---|---|---|
| regular | 2 | 8 | 20 | 4 | 1267 | 12.28 | 0.29 | 4,998 | 4.66 |
| $a \downarrow$ | 1 | 8 | 20 | 4 | 1068 | 12.24 | 0.37 | 4,555 | 5.11 |
| $a \uparrow$ | 4 | 8 | 20 | 4 | 1346 | 12.33 | 0.22 | 5,129 | 4.54 |
| $d \downarrow$ | 2 | 5 | 20 | 4 | 1106 | 12.21 | 0.41 | 4,586 | 5.08 |
| $d \uparrow$ | 2 | 12 | 20 | 4 | 1158 | 12.26 | 0.32 | 4,739 | 4.92 |
| $k \downarrow$ | 2 | 8 | 10 | 4 | 939 | 12.47 | 0.10 | 4,283 | 5.44 |
| $k \uparrow$ | 2 | 8 | 30 | 4 | 1168 | 12.16 | 0.45 | 4,797 | 4.86 |
| $p \downarrow$ | 2 | 8 | 20 | 2 | 1141 | 12.27 | 0.32 | 4,664 | 4.99 |
| $p \uparrow$ | 2 | 8 | 20 | 6 | 1251 | 12.45 | 0.07 | 4,943 | 4.71 |
| $\boldsymbol{V} \in \mathbb{R}^{N \times 100}$ | 2 | 8 | 20 | 4 | 537 | 54.65 | 1.53 | 1,362 | 17.11 |
| $\tau = 0.1$ | 2 | 8 | 20 | 4 | 833 | 7.55 | 11.11 | 4,086 | 5.70 |

of course with lower accuracy with respect to the full model. This fits into the picture of the POD hyper-reduced model, where the size of the active element set increases, when the size of the reduced basis is increased. A decrease of the tolerance $\tau$ leads, as expected, to a smaller element set and a larger $RE_{hr}$ error. Whether the decrease of the $RE_f$ error has a random or systematic pattern is a topic for future investigations.

The same robustness with respect to the setting parameters is achieved, when the NSMTS are used as training sets. In Table 13.6 the variations of the setting parameters of the NSMTS computation are listed. As with the NSKTS, the number of selected elements and hence the computational wall time fluctuates within the expected range, which exists, since the computation of NSMTS includes a random number generation. In the discussed rubber boot example, all variations of the NSMTS setting parameters result in very accurate hyper-reduced model, as every experiment results in a model with the $RE_{hr}$ value below 0.1 %. This is clearly below the average value of the hyper-reduced models trained with NSKTS.

The question, for which type of models the NSKTS are the better training sets and for which the NSMTS are better is a topic for future research. What can be stated is the fact, that both the NSMTS and the NSKTS are computed with routines which are very robust with respect to the setting parameters and can be performed without the necessity of a tedious parameter fine-tuning.

### 13.4.4 Offline Costs

The last question remaining is about the offline costs associated with the computation of the NSKTS, the NSMTS and the sNNLS procedure. For the computation of the NSKTS and the NSMTS, $d$ nonlinear independent problems with $k$ force increments have to be

**Table 13.6:** Overview of the variations of the setting parameters for the NSMTS and their results on the reduction accuracy. As a random number generation is involved in the NSMTS computation, the numbers show just the trend. $a$: force correction factor, $d$: number of stochastic force vectors, $k$: number of force increments, $p$: number of force modes.

| **Name** | $a$ | $d$ | $k$ | $p$ | **elements** | $RE_f$ **[%]** | $RE_{hr}$ **[%]** | $t_w$ **[s]** | **speedup [-]** |
|---|---|---|---|---|---|---|---|---|---|
| regular | 1 | 8 | 20 | 6 | 1219 | 12.44 | 0.072 | 4906 | 4.75 |
| $a \downarrow$ | 0.5 | 8 | 20 | 6 | 1106 | 12.43 | 0.087 | 4611 | 5.05 |
| $a \uparrow$ | 2 | 8 | 20 | 6 | 1308 | 12.47 | 0.032 | 5057 | 4.61 |
| $d \downarrow$ | 1 | 5 | 20 | 6 | 1146 | 12.45 | 0.092 | 4676 | 4.98 |
| $d \uparrow$ | 1 | 12 | 20 | 6 | 1229 | 12.46 | 0.057 | 4859 | 4.79 |
| $k \downarrow$ | 1 | 8 | 10 | 6 | 1070 | 12.47 | 0.072 | 4497 | 5.18 |
| $k \uparrow$ | 1 | 8 | 30 | 6 | 1240 | 12.45 | 0.093 | 4878 | 4.78 |
| $p \downarrow$ | 1 | 8 | 20 | 3 | 1026 | 12.52 | 0.079 | 4422 | 5.27 |
| $p \uparrow$ | 1 | 8 | 20 | 9 | 1253 | 12.43 | 0.089 | 4920 | 4.73 |

solved. As they can be parallelized with no effort, all NSKTS and NSMTS were computed in a parallel manner on all four cores available on the workstation. The computation of the NSKTS for $d = 8$ took 2811 s, and the solution of the sNNLS problem took 3218 s. For the NSMTS with $d = 8$, 3880 s were needed and the solution of the sNNLS problem took 3288 s. With the construction of the reduced basis requiring 554 s, the total computation effort of both offline and online is still considerably smaller than one single full training simulation required for the POD and the ECSW with full training set.

Hence, both the NSKTS and the NSMTS are very attractive not only if hyper-reduced models are to be obtained with little offline costs, but also for scenarios where the full simulation is too expensive, and an approximation of the full solution is to be obtained with less effort.

# Chapter 14

# Summary of Part II

The previous four chapters cover the hyper-reduction of geometrically nonlinear structural dynamics systems, which are summarized in the following. The starting point of hyper-reduction is the reduced basis method, which projects the equations of motion of the full, unreduced system onto a low-dimensional subspace, resulting in a reduced set of equations. The evaluation of the nonlinearities, however, is still associated with the dimension of the full, unreduced system leading to moderate speedup rates for nonlinear reduced systems. Hyper-reduction accelerates the computation of the nonlinear terms, which are the nonlinear internal forces and, as they are necessary for implicit time integration schemes, the derivatives forming the tangential stiffness matrix.

The first hyper-reduction method addressed in Chapter 11 speeds up the evaluation of the internal forces by treating it as a Taylor series. Thereby it is convenient, that the internal forces of a continuum modeled with the linear St. Venant-Kirchhoff material and discretized with geometrically nonlinear finite elements are third order polynomials. Consequently, the Taylor series converges with the cubic term resulting in an exact representation of the internal forces. The computational effort is reduced, since evaluating the polynomials is less computationally expensive for highly reduced systems than the standard finite element procedure involving an element-wise evaluation and a subsequent assembly. In the offline stage, the polynomial coefficients have to be determined, which can be arranged as multidimensional symmetric arrays of dimension two, three and four. A numerical differentiation scheme operating on tangential stiffness matrices is proposed for computing the polynomial coefficients. It turns out to be more efficient than established displacement-based identification methods using the internal forces, but less efficient than methods using the tangential stiffness matrix. The investigated methods are intrinsically simulation-free, since no training sets of the full, unreduced system are required.

One issue of the polynomial tensors hyper-reduction is the $\mathcal{O}(n^4)$ dependence of both the model size and the number of multiplications with respect to the dimension $n$ of the reduced system. While for small systems the method is unbeatably fast, the requirement of memory and computational time for large systems becomes quickly prohibitive. To address this issue, an efficient storage and multiplication scheme is proposed, which exploits the symmetry of the polynomial coefficient arrays. With this technique, remarkable speedups and perfect accuracies are reported for the examples investigated.

The Discrete Empirical Interpolation Method (DEIM) discussed in Chapter 12 takes a different approach and reduces the computational costs by evaluating only a subset of nodal forces. The omitted forces are accounted for with an empirically determined force basis and a collocation strategy. This approach can be interpreted as an oblique projection of the nodal forces onto the force basis which, however, destroys desirable properties like symmetry and stability. The strategy to alleviate this issue proposed in

the literature is a splitting of the internal forces into linear and nonlinear part. This is, however, cumbersome in the framework of geometrically nonlinear finite elements and leads to a very disadvantageous approximation of the internal forces.

A further issue of the DEIM is the operation on single nodal forces which are, however, computed on the element level in the finite element framework. The unassembled DEIM called UDEIM adapts the DEIM to the finite element framework allowing for reduced computational costs and more flexibility and modifications in the choice of collocation points. This approach is used to adapt an approach reported in the literature to symmetrize DEIM and transfer it to finite elements, however without success. The numerical experiments show that the symmetric application of the oblique projection distorts the elements so severely, that no reasonable force evaluation is possible in general. Furthermore, the numerical experiments reveal that both the DEIM and the UDEIM perform extremely poorly in geometrically nonlinear structural dynamics, especially when large rotations are present. The instability issues are accompanied by high hyper-reduction errors, even though training snapshots of the full solution are used in the experiments. Since the methods perform so poorly even when they 'know' the solution a priori, further possible modifications of the (U)DEIM to simulation-free techniques are not considered in this work.

The Energy Conserving Mesh Sampling and Weighting (ECSW) method discussed in Chapter 13 is based on the evaluation of a subset of elements, too. It does, however, not suffer from instability and does not require an artificial splitting of linear and nonlinear part making it a stable and accurate hyper-reduction technique suited for geometrically nonlinear structural dynamics systems. It carries the drawback, that it requires training sets which are commonly obtained from full, unreduced training simulations. To apply the method in a simulation-free context, lean, almost simulation-free training sets named Nonlinear Stochastic Krylov Training Sets (NSKTS) and Nonlinear Stochastic Modal Training Sets (NSMTS) are proposed. These training sets are obtained by solving a set of nonlinear, reduced, static problems, where the right hand side consists of stochastically weighted pseudo-dynamic forces. To compute these forces, a Krylov force basis or a modal force basis is built for the NSKTS or the NSMTS, respectively. It is orthonormalized using the impedance norm of the linearized system and then multiplied with a Gaussian distributed random vector resulting in one pseudo-dynamic force vector.

As reported by the extensive numerical experiments conducted, both the NSKTS and NSMTS yield very accurate hyper-reduced models with convenient speedup rates. The hyper-reduced models are very robust with respect to both parameter changes and different load cases. In contrast to simulation-based ECSW hyper-reduced models, the NSKTS and NSMTS ECSW hyper-reduced models tend to have more elements in the reduced element set. They are also more robust with respect to a change of the excitation forces, so that they are different from the ones used for generating the training set. Due to their success in both accuracy and speedup rates, the NSKTS and NSMTS open the ECSW to almost simulation-free scenarios.

# Part III

# Closure

# Chapter 15

# Real-Life Application

In this thesis, several methods for the reduction and hyper-reduction of geometrically nonlinear models are discussed. To illustrate the capability of these techniques, an industrial application is reduced and hyper-reduced with some of the addressed techniques. The model of this real-life application is a leaf spring of a truck which is mounted between the axle and the frame. It is a critical component for the simulation of the dynamic behavior of the structure, since all dynamic loads initially caused by the road are transferred through this leaf spring. To precisely answer engineering questions summarized under the term Noise-Vibration-Harshness (NVH) like comfort, vibration behavior, fatigue etc., accurate models of the leaf spring are critical for reasonable simulation results of the whole chassis, as reported in the literature by, e.g., Sugiyama et al. [Sug+06], Kong et al. [Kon+13] or Addepalli et al. [Add+15].

Commonly the overall chassis is simulated using an elastic multi-body system, see, e.g., Mousseau et al. [Mou+99]. Within this simulation framework, the sizes of full finite element models are prohibitive in terms of computational times, as reported by Wasfy and Noor [WN03] and Omar et al. [Oma+04]. Consequently, reduced models are necessary to fill the gap of the accuracy of finite element models, as they come from the design and the computational speed of multi-body systems, where the models due to the reduced number of dofs are commonly very efficient.

Hence, the full finite element model of the leaf spring is reduced and hyper-reduced to illustrate the capability of the proposed methods of this thesis. The embedding of the model within a multi-body framework goes beyond the scope of this work and is a topic for further research.

## 15.1 Introduction to the Leaf Spring Model

The leaf spring model is depicted in Figure 15.1 together with the $x$-$y$-$z$ coordinate system. The leaf spring consists of a top leaf and a bottom leaf, which are firmly clamped in the middle by a central fixture. This central fixture consists of layers of sheet metal in between the leaf springs which are pressed together by a strap not contained in the model. The strap fixing the central fixture is mounted on the axle, which is connected with the wheels contacting the road. On both ends, the top leaf is curled forming an eye. The eye pointing towards the driving direction, which is defined in negative $x$-direction, is referred to as the front eye, the eye pointing against the driving direction is the rear eye. The front eye is fixed with the frame with a joint allowing for rotations about the $y$-axis. The rear eye is connected with the frame with a pendulum support allowing for both a rotation about the $y$-axis and a translation in the $x$-direction. On the front side and on the rear side, the top and bottom leaf are connected with a rubber pad. Both pads are firmly clamped on the top side with the top leaf and have a sliding contact on the bottom,

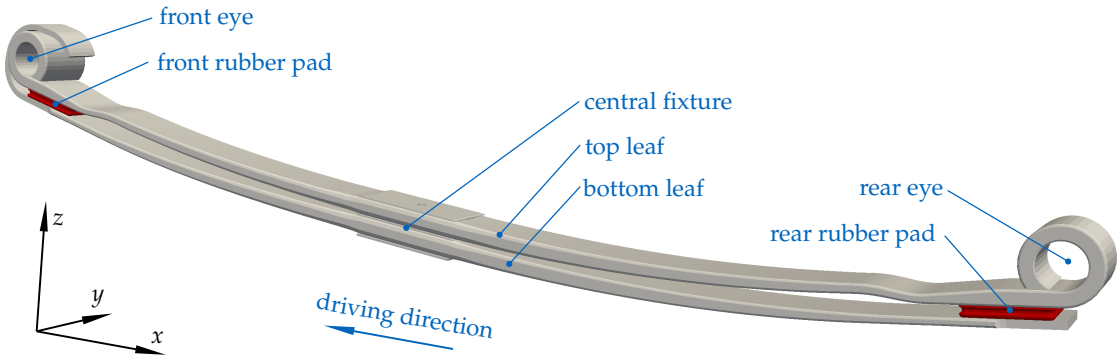


**Figure 15.1:** Leaf spring model

where they touch the bottom leaf.

All steel parts of the leaf spring, i.e., the top and bottom leaf and the sheet metal layers in the central fixture are modeled with linear Hexahedron elements. The steel is modeled with a St. Venant-Kirchhoff material with Young's modulus $E = 210\,\text{GPa}$, Poisson's ratio $\nu = 0.3$ and a density $\rho = 7.85 \cdot 10^3\,\text{kg/m}^3$. The rubber part of the leaf spring is also modeled with a St. Venant-Kirchhoff material with Young's modulus $E = 1\,\text{GPa}$, Poisson's ratio $\nu = 0.4$ and a density $\rho = 1.0 \cdot 10^3\,\text{kg/m}^3$. The mesh of the rubber parts is built using linear Tetrahedron elements. In total, the leaf spring model consists of 53240 Hexahedron elements and 32522 Tetrahedron elements yielding a total of 85762 elements. The Dirichlet boundary conditions are applied as depicted in Figure 15.2. The joint in the front eye is modeled with a fixation of all displacement dofs of a row of nodes inside the eye. The joint in the rear eye is modeled with a fixation of the $y$ and $z$-dofs of a row of the eye allowing for a rotation about the $y$-axis and a displacement in the $x$-direction.

The mesh of the rubber pads and the leafs is non-conforming. Hence, the rubber pads are firmly fixed with the top leaf with a nodal collocation technique, where the nodes of the top surface of the rubber pads are treated as slave nodes. The corresponding nodes of the top leaf are the master nodes. The contact of the pads with the bottom leaf is modeled as a linear sliding contact with the bottom leaf nodes being the master nodes and the rubber pad bottom surface nodes being the slave nodes. The constraint is however a linear constraint which is not updated, i.e., the relative motion of the nodes is admissible with respect to the reference configuration and not with the actual deformed configuration.

The loads for the model stem from a pseudo-realistic driving maneuver from an elastic multibody simulation, where the internal forces and moments between the axle and the leaf spring are tracked. They are given in $x$, $y$ and $z$ direction and form the excitation of the leaf spring. To apply these forces and moments on the finite element system, force distributions are computed which apply the unit forces and moments on the top face of the central fixture. They are depicted in Figure 15.3. The time evolution of the internal forces and moments between axle and leaf spring stemming from the multibody simulation are depicted in Figure 15.4.

To measure the displacements and rotations of the finite element system at the interface to the axle, the displacements are taken from the so-called master node depicted in Figure 15.5. It is located almost in the middle on the top face of the central fixture. To obtain the rotations of the central fixture, the displacements of the four nodes $P_1$, $P_2$, $P_3$ and $P_4$ depicted in Figure 15.5 are evaluated. The rotations $\alpha$ about the $x$-axis, $\beta$ about the $y$-axis and $\gamma$ about the $z$ axis are computed as

$$\alpha = \frac{u_z(P_2) - u_z(P_4)}{l_2}, \quad \beta = \frac{u_z(P_3) - u_z(P_1)}{l_1}, \quad \gamma = \frac{u_x(P_4) - u_x(P_2)}{l_2}, \tag{15.1}$$

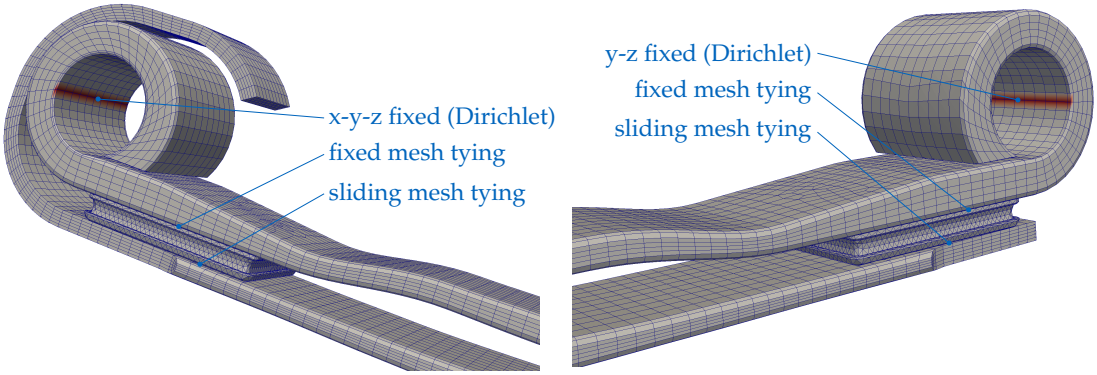


**Figure 15.2:** Dirichlet constraints in the front eye (left) and on the rear eye (right). The red colored nodes of the front eye are fixed in all directions, the colored nodes of the rear eye are fixed in the y and z direction.

with $u_x(P)$ and $u_z(P)$ being the x or z displacement component of point $P$, respectively. The length values $l_1$ and $l_2$ are the difference of the x-component of $P_1$ and $P_3$ and the y-component of $P_2$ and $P_4$ in the undeformed configuration.

The full model is equipped with proportional Rayleigh damping. The constants $\alpha$ and $\beta$ are adjusted such, that the first two eigenfrequencies around the undeformed configuration of $f_1 = 13.3\,\text{Hz}$ and $f_2 = 41.7\,\text{Hz}$ have a damping ratio of $\zeta = 0.005$ resulting in $\alpha = 0.633$ and $\beta = 2.89 \cdot 10^{-5}$. The subsequent undamped eigenfrequencies are at $f_3 = 43.2\,\text{Hz}$, $f_4 = 72.2\,\text{Hz}$ and $f_5 = 100.0\,\text{Hz}$.

The full model is integrated using a generalized-$\alpha$ time integration scheme with $\rho_\infty = 0.8$ and a time step width $\Delta t = 3.333 \cdot 10^{-3}\,\text{s}$. The simulation time goes from $t_0 = 0$ to $t_{end} = 5\,\text{s}$ resulting in a total of 1,500 time steps. The wall time for the simulation is $t_w = 40,022\,\text{s}$ and hence with more than eleven hours extremely long. For comparison, a linearized model is computed. It is simulated with the same time integration settings and takes a wall time of $t_w = 288\,\text{s}$. Even though the computational time is below five minutes and hence very attractive, the results are clearly different to the nonlinear simulation, as depicted in Figure 15.6. The linearized simulation exhibits oscillations which are not visible in the nonlinear simulation. This is presumably due to the force excitation in the $x$-direction, which excites the first mode at 13.3 Hz. Since in linear models the undeformed configuration is the reference configuration in which the external forces are applied to, the force in $x$-direction has a lever enabling the excitation of the first eigenmode. In the nonlinear simulation, however, the force is applied onto the deformed configuration reflecting the change of the point of force application. Since the lever of the forces in $x$-direction is almost zero, no excitation of modes is present in the nonlinear simulation.

Besides the oscillation, the mean displacements in both $x$ and $z$-direction are overestimated in the linear simulation. Consequently it is necessary to reflect the nonlinearity of the leaf spring in order to achieve accurate results.

## 15.2 Simulation-Free Basis

Next a simulation-free basis is built. Therefore the first 40 vibration modes are computed. To compute the static derivatives, the step width for the finite difference scheme has to be chosen appropriately (cf. Section 6.3). Hence, the symmetry error of the SMDs is computed. To accelerate the computation, not the full basis of 40 modes is used for the symmetry check but a pseudo basis of five modes composed of mode number 1, 5, 10, 15 and 20 is used for the check. The symmetry error defined in (6.22) is plotted over the step

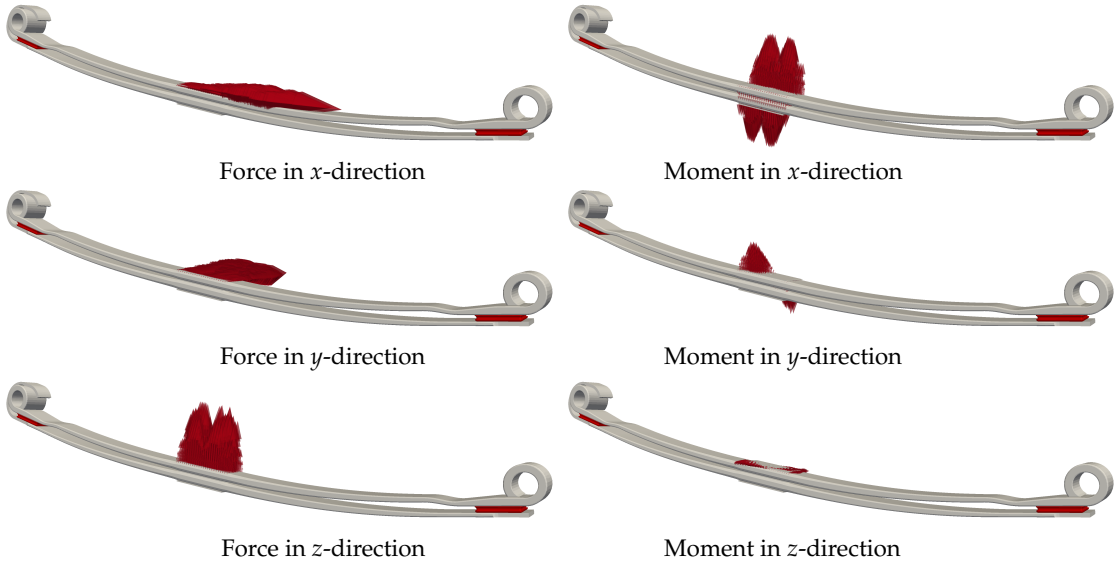


**Figure 15.3:** Unit force distributions at the central sheet package to apply the loads of the multi body simulation.

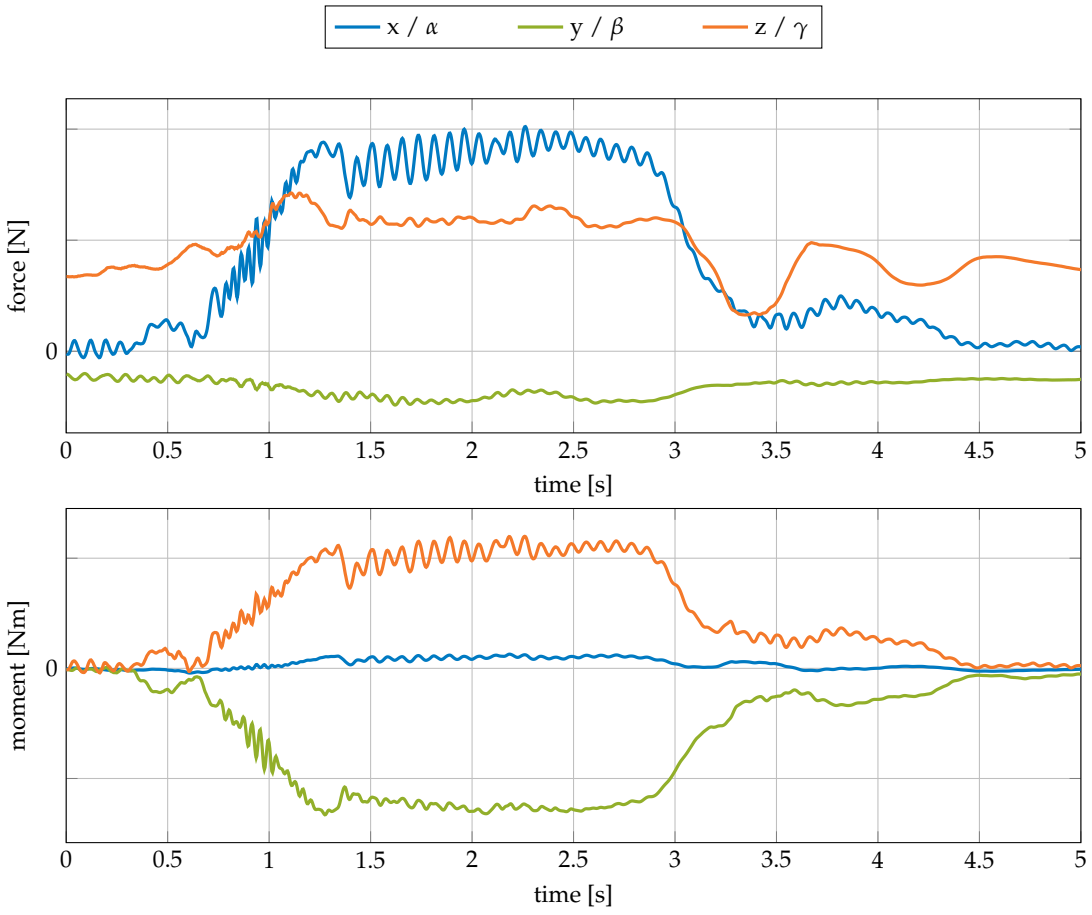


**Figure 15.4:** External forces and moments from a virtual road setting.

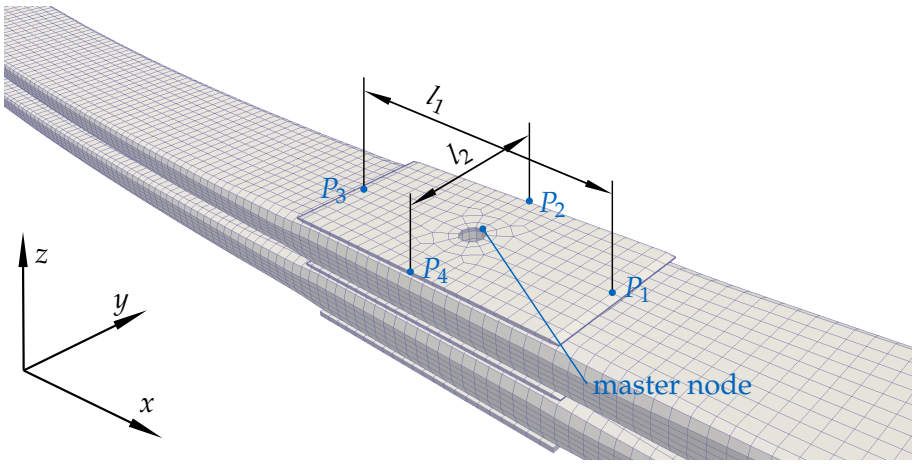


**Figure 15.5:** Leaf spring model: Displacements and angles

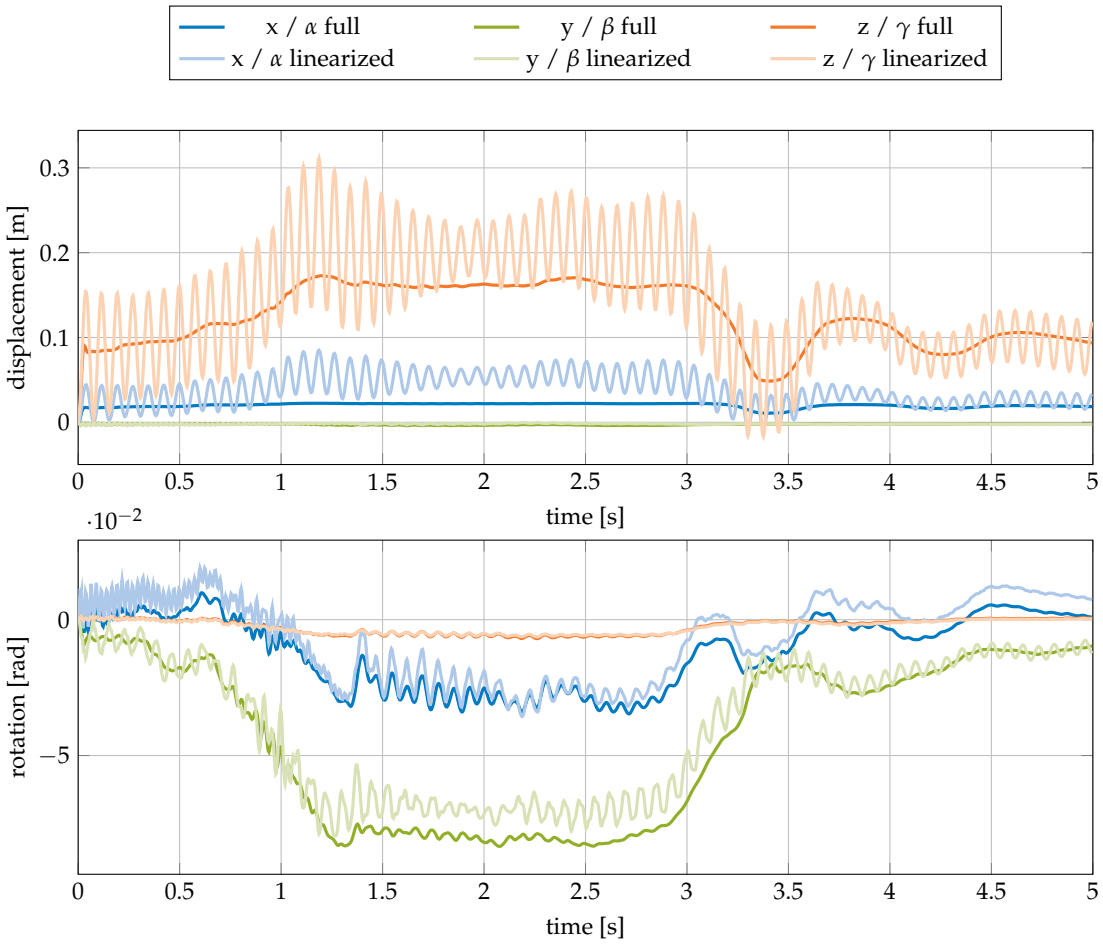


**Figure 15.6:** Displacements of a linear and a nonlinear time integration.

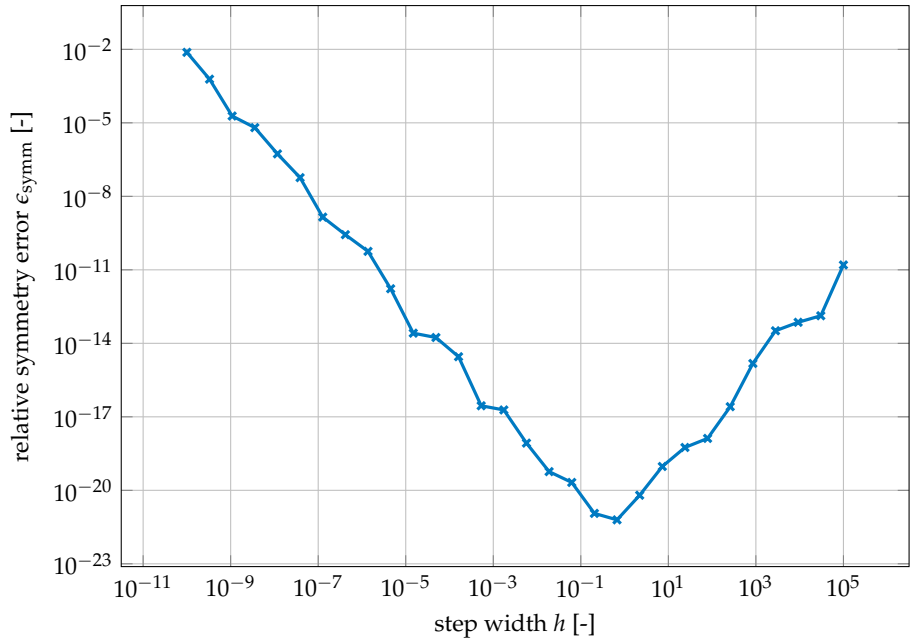


**Figure 15.7:** Relative symmetry error of the central finite difference scheme. To accelerate the computation, the error is computed with a basis composed of four vibration modes.

width $h$ of the central finite difference scheme in Figure 15.7. It indicates that, as in the academic examples before, the optimal step width is in the range of $h = 1$. With this step width the full static derivative tensor $\boldsymbol{\Theta}$ is built for the 40 vibration modes, as discussed in Subsection 6.2.2. In Figure 15.8, the first three vibration modes and the corresponding static derivatives are depicted. They show a similar behavior as for the academic cantilever plate example depicted in Figure 6.2, especially for mode $\boldsymbol{\phi}_1$ and mode $\boldsymbol{\phi}_3$ which are the first and second bending mode. The corresponding static derivatives exhibit a displacement field describing a contraction of the leaf spring in longitudinal direction, which is necessary for the reproduction of geometrically nonlinear motions. The second mode $\boldsymbol{\phi}_2$ is a bending mode in the $y$-direction. In this mode, the lower leaf slips off the rubber pads in the $y$-direction, since only a normal contact is enforced there.

For an efficient reduction, it is desirable to keep the reduced basis compact. Hence, the frequency weighting selection criterion discussed in Subsection 6.2.4 is chosen, since it proved to be both simple and effective in Section 6.4. Additionally to the 40 vibration modes, 60 static derivatives are chosen according to this selection resulting in a reduced basis $\boldsymbol{V}$ of 100 basis vectors. This basis is orthogonalized and deflated according to Subsection 6.2.3, which does for this example, however, not change the number of basis vectors.

Next a reduced simulation is run with the same time integration scheme as the full system in the previous section. The displacements and rotations of the central fixture are depicted in Figure 15.9 indicating a good fit with the reference solution especially for the displacements. The rotation $\alpha$ about the $x$ axis is not captured well. However, this rotation is not of great interest, since most of the relevant dynamics of the leaf spring happens in the $x$-$z$-plane. The underestimation of the rotations about the $x$-direction and also slightly about the $y$-direction of the reduced basis model is due to the additional constraints, which are applied to the reduced system, as discussed in Chapter 3.

The wall time for the reduced computation is 24127 s. Even though this is a speedup of a factor of 1.66 compared to the full model, the time of over six and a half hours is still very unappealing for dynamic simulations. Consequently, hyper-reduction is necessary

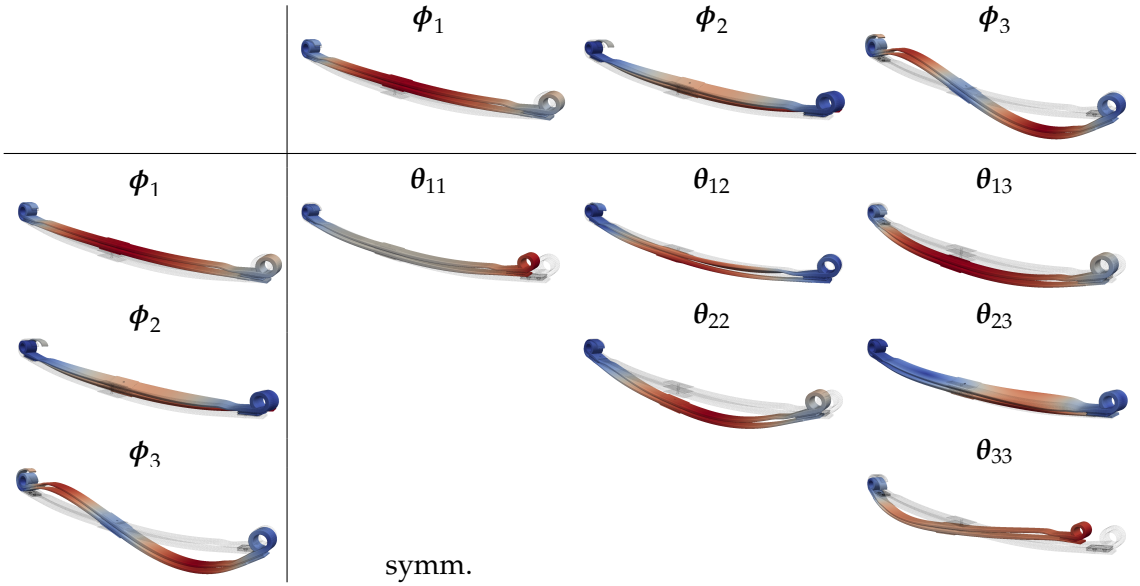


**Figure 15.8:** First three vibration modes and the corresponding static derivatives of the leaf spring model.

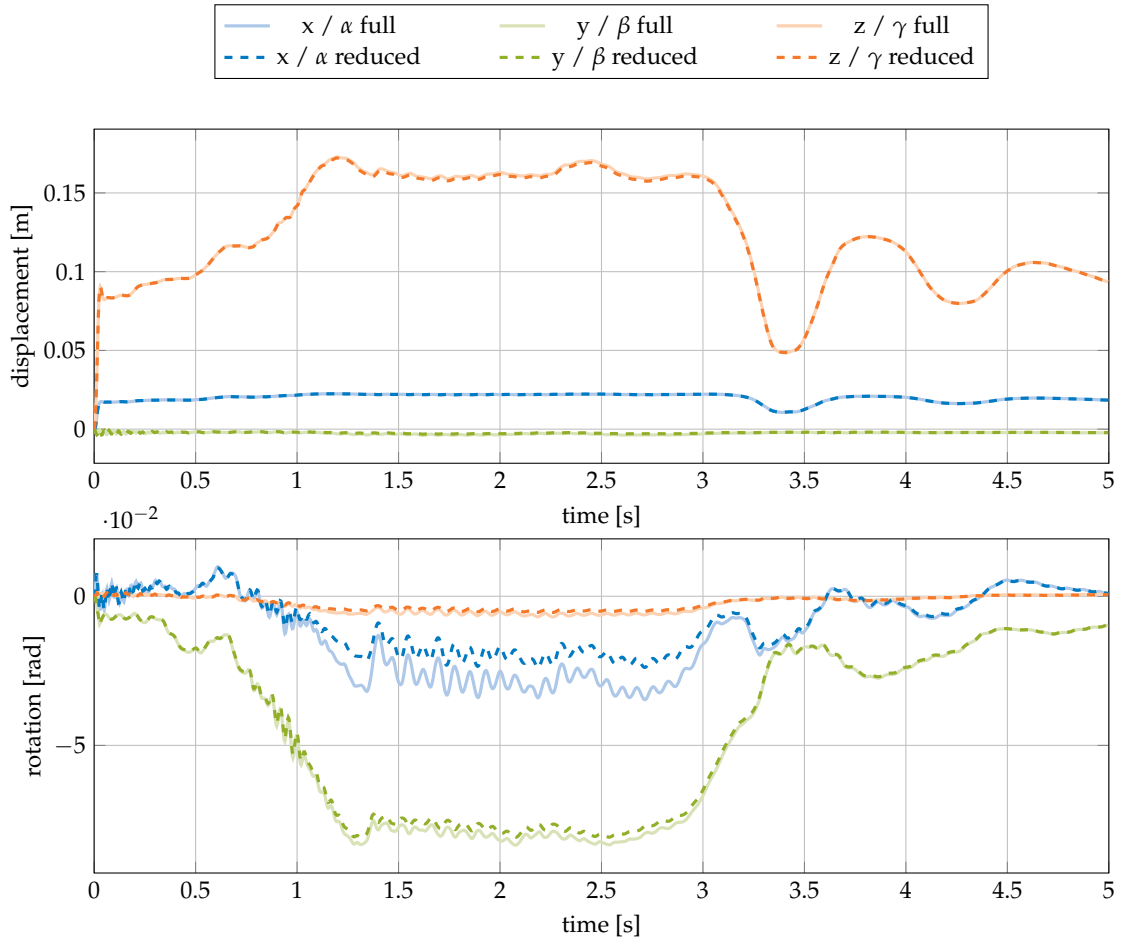


**Figure 15.9:** Displacements of the time integration with the reduced model.

to increase the speedup further.

## 15.3 Hyper-Reduction

The reduced basis reduction of the previous section allowed for a reduction of the computational time in a moderate range of 1.66. The reason is, as discussed in Chapter 10, that the reduced basis projection accelerates the solution of the system of linear equations, but not the computation of the reduced nonlinear forces and the tangential stiffness matrix, which are still associated with the size of the high dimensional mesh.

To achieve even greater speedup factors, hyper-reduction methods are necessary. Since the reduced basis for the problem is constructed in a simulation-free manner, lean and efficient training sets proposed in this thesis, i.e., NSKTS and NSMTS, are used for the ECSW hyper-reduction of the leaf spring model. Furthermore, the polynomial reduction is applied to the model as well.

### 15.3.1 Nonlinear Stochastic Krylov Training Sets (NSKTS)

To compute the NSKTS, the Krylov force subspace for all six force input directions depicted in Figure 15.3 is constructed with $p = 4$ moments each. The impedance norm for the maximum external force is computed independently for all six components. Therefore, the maximum value of each force component (cf. Figure 15.4) is chosen. Then the NSKTS are computed with the force correction factor $a = 1$, the number of stochastic force vectors $d = 8$ and the number of force increments $k = 20$. The computation of the NSKTS is performed in parallel on four cores taking 1525 s in wall time, i.e., less than 26 minutes. Every load step converged within 30 iterations leading to 160 training sets depicted in Figure 15.12. Then the ECSW is run with the NSKTS. The construction of the matrices $\mathbf{Y}$ and $\boldsymbol{b}$ and the sNNLS solver take in total 2840 s for $\tau = 0.01$, i.e., about 47 minutes. The resulting reduced mesh consists of 816 elements, which are depicted in Figure 15.11. The wall time for the time integration of the hyper-reduced system is 1047 s, i.e. less than 18 minutes. Compared to the full solution, this is a speedup factor of more than 38 allowing for the use of the reduced model within a full chassis simulation.

The accuracy of the hyper-reduced model using NSKTS is very high, which is reflected in the low hyper-reduction error of $RE_{hr} = 0.13\,\%$. The displacements and rotations of the central fixture of the hyper-reduced simulation practically overlay the displacements of the full, unreduced simulation, as depicted in Figure 15.10.

### 15.3.2 Nonlinear Stochastic Modal Training Sets (NSMTS)

Also the NSMTS are used as training sets to hyper-reduce the reduced basis model of Section 15.2. The NSMTS are computed with $p = 6$ modal force bases, $d = 8$ stochastic force vectors and $k = 20$ force increments for the nonlinear solution. The standard deviation $\sigma$ of the random variable was chosen as the sum of the impedance norm of the respective maximum value of all six independent force input directions resulting of a force correction factor $a = 1$.

The NSMTS are computed on four processors with a wall time of $t_w = 1728$ s, i.e. about 29 minutes. All static solutions converged, so that a total of 160 training sets, which are depicted in Figure 15.13, are available for the sNNLS solver. The ECSW reduction for $\tau = 0.01$ takes a total wall time of $t_w = 5465$ s, i.e. about 91 minutes, so clearly longer than for the NSKTS. The resulting mesh contains of 1133 elements and is depicted in Figure 15.13.

Next the time integration of the NSMTS hyper-reduced model is performed. The wall time is $t_w = 1177$ s, i.e., less than 20 minutes, resulting in a speedup factor of 34 compared

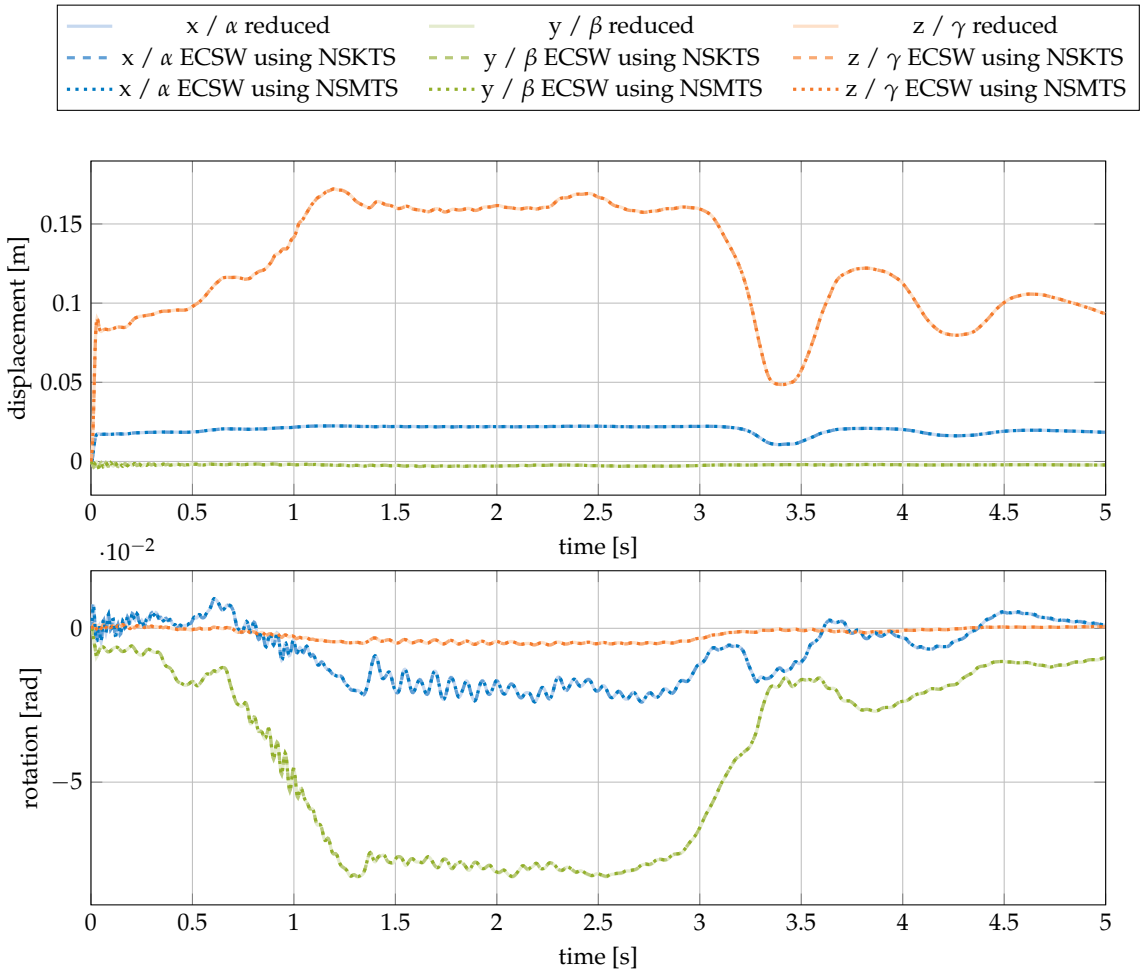


**Figure 15.10:** Displacements of the time integration with the reduced model and different hyper-reduced models.

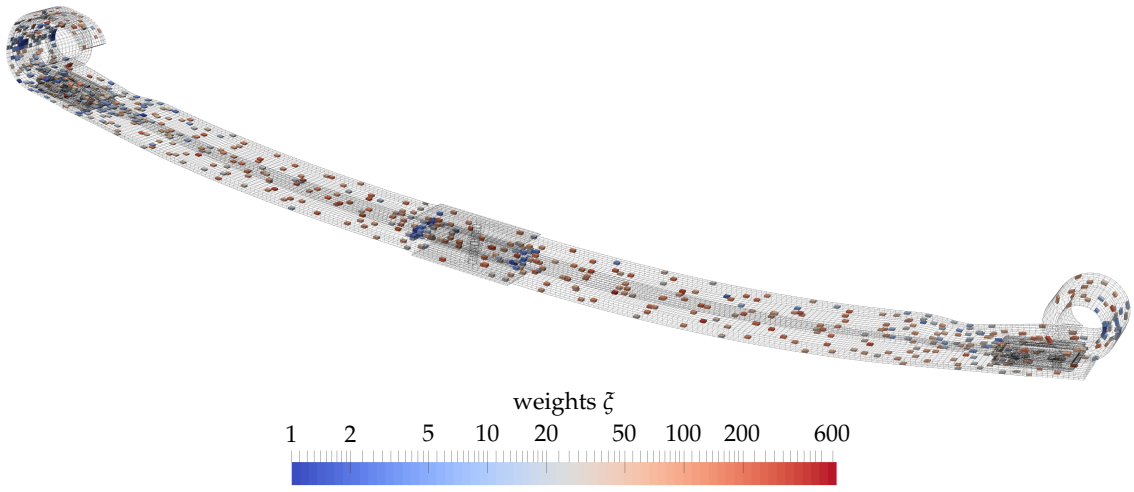


**Figure 15.11:** Hyper-reduced mesh of the leaf spring using NSKTS as training snapshots.

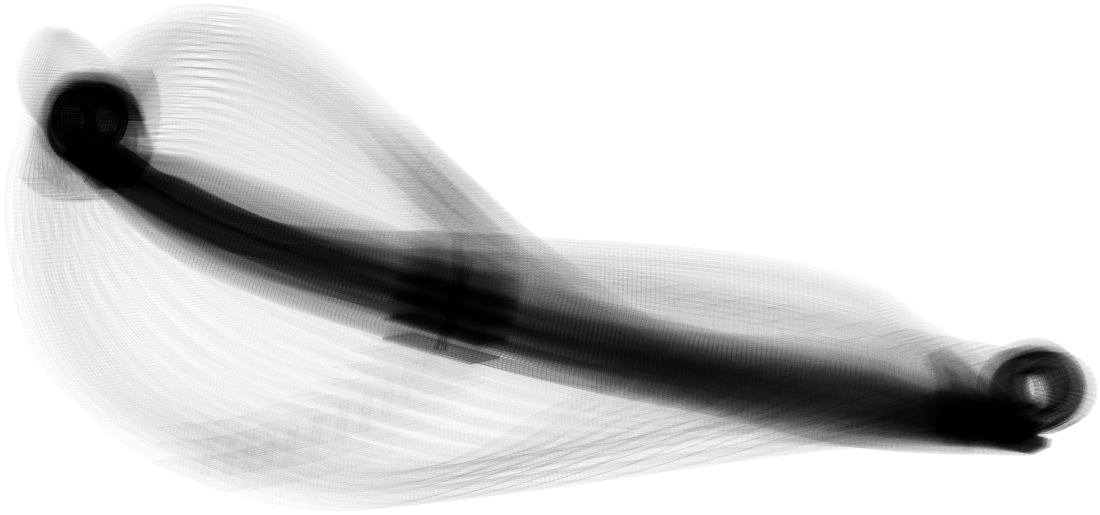

**Figure 15.12:** NSKTS training snapshots

to the full, unreduced simulation. Consequently, also the NSMTS are very good hyper-reduction training sets resulting in a model which is in the same range of speedup as the NSKTS hyper-reduced models. The accuracy of the NSMTS hyper-reduced model is also illustrated in Figure 15.10, where the displacements and angles of the NSMTS hyper-reduced model practically overlay the reduced basis model. The accuracy is reflected also with the tight error measure of $RE_{hr} = 0.08\,\%$.

### 15.3.3 Polynomial Tensors

The last hyper-reduction technique investigated is the polynomial tensor reduction discussed in Chapter 11. Since the material used for the leaf spring is a St. Venant-Kirchhoff material, this hyper-reduction strategy should represent the reduced basis model accurately.

First, the polynomial tensors $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ are computed. Since the different computations strategies result in equal higher-order arrays besides tiny numerical errors, the identification using the tangential stiffness matrix is chosen. It is associated with the lowest off-line costs (cf. Figure 11.3). For the simulation-free basis of $n = 100$ basis vectors, the identification requires 5150 evaluations of the tangential stiffness matrix. It takes a wall time of 28,856 s, slightly more than eight hours. The higher dimensional arrays $\boldsymbol{K}^{(2)}$ and $\boldsymbol{K}^{(3)}$ are stored in the efficient storage scheme reducing the memory requirement from 8 MB for $\boldsymbol{K}^{(2)}$ and 800 MB for $\boldsymbol{K}^{(3)}$ to 1.4 MB and 35.4 MB, respectively.

Next, the time integration is performed using the efficient multiplication scheme implemented in Numba, see Subsection 11.5.2. The wall time for the computation is 113 s and hence with less than two minutes clearly below all other computational times. The speedup factor is enormous with over 354 in comparison to the full, unreduced simulation. The polynomially hyper-reduced system is even faster than the linearized simulation of the full system, since the single factorization of the iteration matrix and the forward and backward substitution in every time step consume more computational effort than the computation of the tangential stiffness matrix based on the polynomial algorithm discussed in Section 11.4.

Also the accuracy of the polynomial tensors hyper-reduction approach is excellent. The hyper-reduction error is $RE_{hr} = 5.3 \cdot 10^{-7} = 5.3 \cdot 10^{-5}\,\%$, and hence in the range of the relative tolerance of the time integration scheme. As a consequence, the polyno-

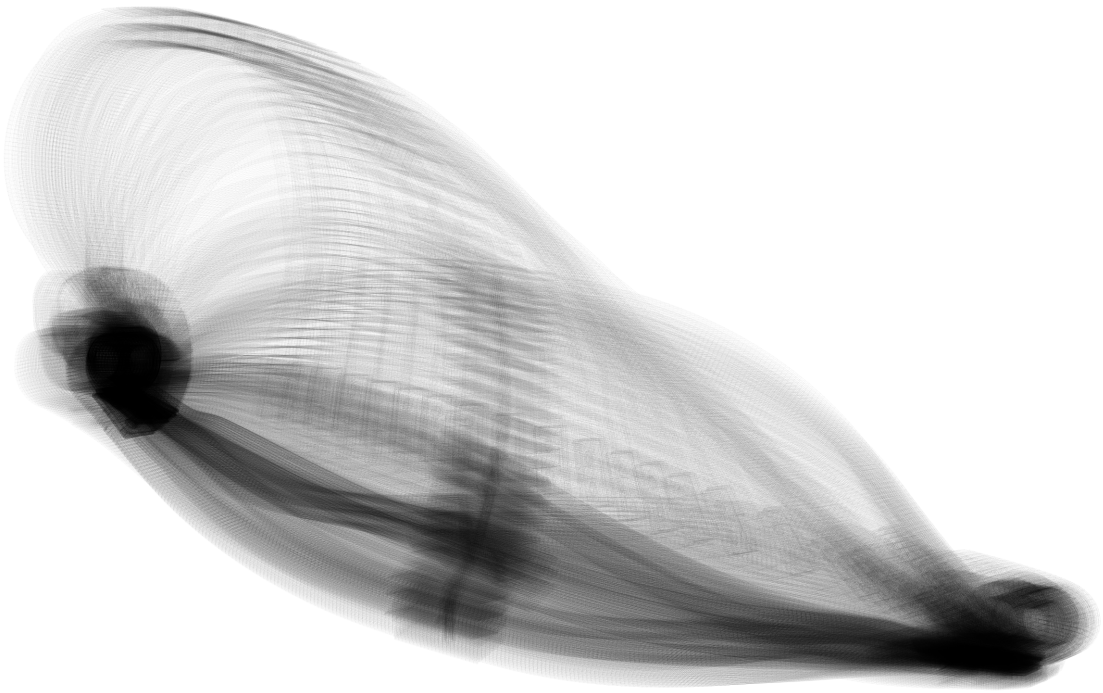

**Figure 15.13:** NSMTS training snapshots

mial hyper-reduction scheme is extremely efficient for the online stage making it a very attractive hyper-reduction technique for geometrically nonlinear systems for St. Venant-Kirchhoff material systems, where the reduced basis is reasonably small.

## 15.4 Discussion

The real life example introduced in Section 15.1 with initially 216,499 dofs has been reduced using a simulation-free basis composed of vibration modes and SMDs yielding a reduced basis of order $n = 100$. In Table 15.1 an overview of the experiments performed in this chapter is given. The low reduction error $RE_f$ of the reduced model indicates, that the reduced basis using modes and static derivatives works well for the given loads and settings, since the static derivatives capture the nonlinearities well.

Then, three different hyper-reduction techniques are applied. The training sets proposed in this thesis, the NSKTS and NSMTS, prove to be very good for the hyper-reduction using the ECSW. The speedup rates in the range of 35 show, that the ECSW in combination with NSKTS and NSMTS leads to both, accurate and fast hyper-reduced models.

The polynomial hyper-reduction, though, yields speedup factors in the range of one order of magnitude higher than the NSKTS or NSMTS trained ECSW models. This speedup rates, or, from the other perspective, the reduction of the online costs from more than six hours to under two minutes make the simulation of the leaf spring in the context of multibody simulations a feasible scenario.

The two drawbacks of the polynomial tensor reduction are the limitation to St. Venant-Kirchhoff materials and the high requirement for memory and offline computational time when the reduced system is fairly large. Thereby, the memory requirement scales with $\mathcal{O}(n^4)$ with the size $n$ of the reduced model. While the basis size of $n = 100$ leads to excellent results in this example, the size of $n = 346$ like in the rubber-boot example leads to a full $\boldsymbol{K}^{(3)}$ array requiring 114.7 GB in memory. Even with the reduced storage con-

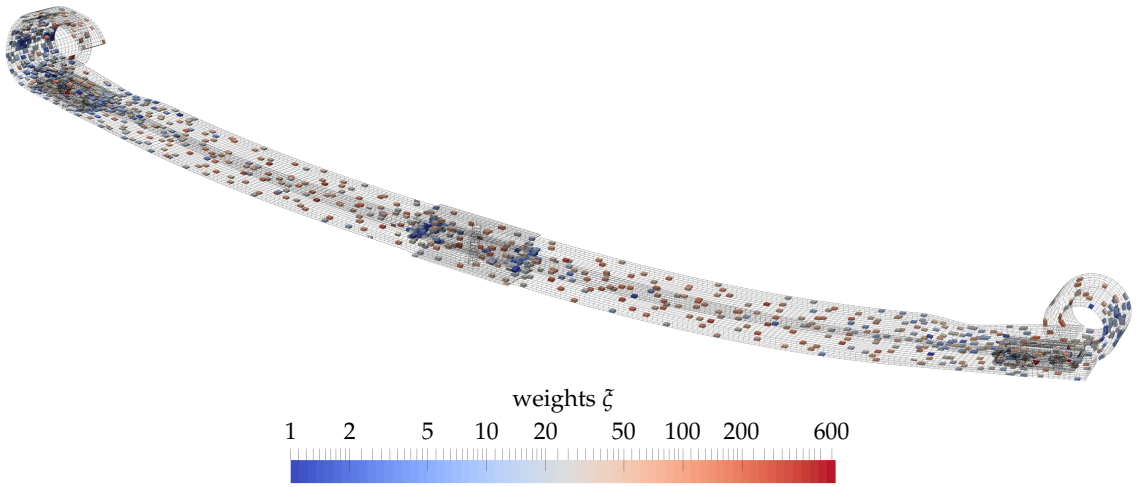


**Figure 15.14:** Hyper-reduced mesh of the leaf spring using NSMTS as training snapshots.

cept of Subsection 11.4.1, the size is still 4.9 GB. For even larger bases, e.g., $n = 1000$, the size of 335.3 GB for the efficient storage of $\boldsymbol{K}^{(3)}$ is prohibitive for simulations on a regular workstation.

**Table 15.1:** Errors and speedup rates of the different reduction techniques for the leaf spring example.

| **Reduction Method** | **dofs** | **elements** | $RE_f$ **[%]** | $RE_{hr}$ **[%]** | $t_w$ **[s]** | **speedup [-]** |
|---|---|---|---|---|---|---|
| full | 216499 | 85762 | — | — | 40022 | — |
| reduced basis (modes + SMDs) | 100 | 85762 | 0.68 | — | 24127 | 1.66 |
| Modes & SMDs + ECSW using NSKTS | 100 | 816 | 0.78 | 0.13 | 1047 | 38.23 |
| Modes & SMDs + ECSW using NSMTS | 100 | 1133 | 0.74 | 0.08 | 1177 | 34.00 |
| Modes & SMDs + Poly3 | 100 | — | 0.68 | 0.00 | 113 | 354.18 |
| linearized | 216499 | 85762 | 45.14 | 45.56 | 288 | 138.97 |

# Chapter 16

# Closure

In this thesis, simulation-free model order reduction methods are investigated in the context of geometrically nonlinear structural dynamics. This chapter discusses the main results of the thesis and emerging research topics for future work. Since the thesis is divided in three parts, the rather detailed summary of the first two parts is given at its respective last chapter: Part I is summarized in Chapter 9 on page 107, Part II in Chapter 14 on page 179. The summary and a discussion of the real-life example addressed in Part III of the thesis is given at the end of the previous chapter in Section 15.4. The main conclusions resulting from the work described in this thesis are given in the following.

## 16.1 Conclusions and Discussion

**Linear reduction methods cannot be naively applied to nonlinear systems.** When geometrically nonlinear systems are considered, the computation of a reduced basis is not a trivial task. For linear systems various methods exist which rely on system intrinsic properties like decoupling of states, approximation of the transfer function or balancing of observability and controllability. In nonlinear systems, however, these properties do not exist or are extremely expensive to compute. Hence, the linear reduction methods, which are addressed in Chapter 4, cannot be applied to nonlinear systems. When they are applied to the linearized version of the nonlinear system, the resulting reduced basis is not suited for the nonlinear system, since it does not capture the nonlinear behavior and leads to severe locking, as illustrated in Section 6.1. As a consequence, the common approach to reduce nonlinear systems is a detour over a statistical investigation of training sets obtained in training simulations of the full, unreduced system, as discussed in Chapter 5. A further class of approaches used in nonlinear structural dynamics is based on perturbations of reduced bases, which are addressed in Chapter 6.

**Modal derivatives and static derivatives are excellent simulation-free augmentation vectors for linear reduction techniques to reduce geometrically nonlinear structural dynamics systems.** The reduced bases obtained with linear reduction methods applied to a linearized system are not suited to the original nonlinear system. However, the linear basis can be extended by modal derivatives (MDs) and static derivatives (SDs) which both capture the nonlinear behavior and make the augmented basis applicable to the nonlinear system. While the modal derivatives are based on the perturbation of vibration modes and are hence limited to bases composed of those, the static derivatives are applicable to all types of linear bases. In the literature, MDs and SDs are sometimes used synonymously, but they are different vectors as shown in the numerical experiments conducted in Chapter 6. The experiments also reveal that both MDs and SDs are well suited to reduce geometrically nonlinear systems. One drawback is, though, that their

number grows quadratically with respect to the number of modes gathered in the initial linear basis. To address this issue, heuristic selection strategies are available to select a small number of MDs or SDs forming smaller bases.

**Quadratic manifold approach allows to reduce a system to a minimum set of generalized coordinates, however it is limited in its applicability.** A conceptually different method for reduction is the quadratic manifold approach, where the transformation between physical coordinates and reduced generalized coordinates is a quadratic function and hence nonlinear. The full system is projected onto the tangent subspace of the nonlinear transformation resulting in more complicated reduced equations of motion. The quadratic manifold approach allows to reduce a geometrically nonlinear system to a minimum set of generalized coordinates. In the method proposed, the quadratic transformation is composed of a linear part, which is formed by a linear basis, and a quadratic part, which is chosen to be either MDs or SDs. The quadratic manifold approach is closely related to nonlinear static condensation when applied to beams and shells with von Karman kinematics. While the quadratic manifold approach leads to accurate results in these cases, it suffers from locking when applied to general structures discretized with solid elements. As revealed by the numerical experiments, they are then only applicable to structures with a membrane-dominated nonlinearity exhibiting small rotations.

**Subspace angles allow for comparison of reduced bases. Principal vectors can show the differences of reduction bases making them a useful tool for analyses.** With subspace angles and the corresponding principal vectors, the overlapping and distant spaces of two reduced bases can be identified independently of the realization. The principal vectors correspond to displacement fields which can be visualized for an intuitive investigation of the differences. In Section 8.3, this technique is used to descriptively show that the linear bases do not capture the contraction of a mesh necessary to represent large rotational deformations. To assess the suitability of a reduced basis to a certain problem, a comparison to a POD basis is often expedient, since this basis can be considered optimal for a certain system under given excitation. When multiple bases need to be assessed, it is in general cheaper to compute the POD basis and compare the basis against this optimal basis using subspace angles instead of running full simulations of the reduced system with a subsequent investigation of the reduction error.

**Hyper-reduction is necessary to exploit the full potential of projective model order reduction for nonlinear systems.** While the reduced basis method decreases the number of dofs of a finite element system, it does not reduce the cost for the computation of the nonlinear terms, which are evaluated on the finite element level. As a consequence, the computation of the nonlinear internal force and the tangential stiffness matrix is the computational bottleneck of a nonlinear reduced basis system limiting clearly the speedup rate of the reduced basis system. To alleviate the computational burden associated with the evaluation of the nonlinearity, hyper-reduction is necessary. It approximates the nonlinear internal force leading to clearly reduced computational costs allowing for clearly larger speedup rates, as illustrated in the numerical examples given in Part II.

**Polynomial tensors hyper-reduction is the method to go for reduced systems with small number of dofs and St. Venant-Kirchhoff material.** Among the hyper-reduction methods investigated within this thesis, the polynomial tensors hyper-reduction has clearly the lowest online costs for systems with a low reduced order $n$. This is due to the fact that the computational cost of the evaluation of the polynomial tensors scales with $\mathcal{O}(n^4)$,

which makes this method the clear winner for small $n$. This hyper-reduction method expresses the nonlinear internal forces as third order polynomials, which is an exact representation for St. Venant-Kirchhoff materials. Hence, the hyper-reduction error is in the range of the tolerances used for the computation, e.g., the numerical tolerances set in the time integration scheme. A further advantage of the method is that the reduced system is independent of the finite element formulation, since the three symmetric arrays $\boldsymbol{K}^{(1)} \in \mathbb{R}^{n\times n}$, $\boldsymbol{K}^{(2)} \in \mathbb{R}^{n\times n\times n}$ and $\boldsymbol{K}^{(3)} \in \mathbb{R}^{n\times n\times n\times n}$ completely describe the nonlinear internal forces of the reduced system. This makes the method very attractive for scenarios like model predictive control, in which a nonlinear system needs to be exported to a controller software framework, for instance. The drawbacks of this method, however, are the limited applicability to only St. Venant-Kirchhoff materials and the fourth order growth of the memory requirement and the computational effort with respect to the order $n$ of the reduced system. The efficient storage and multiplication scheme proposed in Section 11.4 allows to keep both memory and computational effort to a minimum but does not revoke this general issue.

**Discrete Empirical Interpolation Method (DEIM) has many drawbacks in the context of geometrically nonlinear structural dynamics.** The DEIM turns out to be a second-class hyper-reduction method for geometrically nonlinear structural dynamics when compared to the other investigated methods. This is mostly due to the fact that the approximation of the internal nonlinear force using a force basis and a collocation strategy destroys the symmetry of the system and with it desirable properties like stability. Furthermore, the DEIM requires a cumbersome splitting of the internal force in a linear and a nonlinear part, which is very unfavorable in the context of large rotations of elements present in geometrically nonlinear systems. While the unassembled DEIM (UDEIM) as a special adaptation to the finite element framework allows for more flexibility in the application of collocation strategies, it does not repeal the main issues. As shown in the numerical experiments, both the DEIM and the UDEIM suffer of instability and poor accuracy making them an inferior choice as hyper-reduction method for geometrically nonlinear structural dynamics.

**Energy Conserving Mesh Sampling and Weighting (ECSW) is a very promising method. The use of NSKTS and NSMTS allows to use this method in a simulation-free context.** The ECSW is a hyper-reduction method similar to the DEIM which, however, does not carry the drawbacks of it. It allows for the hyper-reduction of geometrically nonlinear structural dynamics systems with different hyperelastic materials and leads to accurate and stable hyper-reduced systems. However, training snapshots are required for the computation of the active element set, which is the reduced set of elements responsible for the reduced computational costs. To use the ECSW in a simulation-free setup, the so-called Nonlinear Stochastic Krylov Training Sets (NSKTS) and Nonlinear Stochastic Modal Training Sets (NSMTS) are proposed in this thesis. They allow for an accurate hyper-reduction in an almost simulation-free way, since the training snapshots are obtained by solving a set of nonlinear problems with pseudo-dynamic right hand sides. As demonstrated in Section 13.3, both the NSKTS and NSMTS are very robust in terms of changes of the parameters which are needed for the computation as well as changes of the excitation forces of the reduced system. While the polynomial tensors hyper-reduction distinguishes itself with the extremely low online costs for small reduced systems, the ECSW using NSKTS and NSMTS shows clearly lower offline costs especially for larger reduced systems making them an attractive hyper-reduction method.

**The proposed simulation-free reduced basis and hyper-reduction methods are applicable in an industrial context.** As proved by the industrial example of a leaf spring of a truck discussed in Chapter 15, both simulation-free reduced basis methods and (almost) simulation-free hyper-reduction methods are applicable to drastically reduce the computational times in geometrically nonlinear structural dynamics. Thereby, speedup rates of over two orders of magnitude for the online computation are possible. The polynomial tensors hyper-reduction achieves the best online speedups while the ECSW using NSKTS and NSMTS are clearly slower. However, this trend is flipped in the offline costs, where the computation of the polynomial tensors is clearly more computationally demanding than the computation of the NSKTS or the NSMTS and the ECSW hyper-reduction. The methods proposed and discussed form a foundation for future applications in geometrically nonlinear structural dynamics and further research in this field.

## 16.2 Future Directions of Research

This thesis shows that geometrically nonlinear dynamical systems can successfully be reduced by the appropriate methods in a simulation-free fashion. However, the successful simulation of a system is usually not the end point of a development cycle but the starting point. In many cases, parameter studies, optimizations, *what if?*-scenarios and more in-depth investigations are executed, in which numerical simulations should be quick and affordable making model order reduction an attractive option. However, many aspects which are required in these studies are not addressed within this thesis. They form exciting questions and possible directions of future research:

- For optimization and parameter studies, the systems are treated as parametric requiring parametric model order reduction techniques. Parametric reduced basis systems have been studied in the literature in the field of simulation-based reduced basis methods, see e.g., Amsallem et. al. [Ams+09; Ams10; AF11], where reduced bases are computed for certain points in the parameter space. However, it is yet unclear, if parametric, geometrically nonlinear structural dynamics systems can be reduced more efficiently by exploiting intrinsic physical properties. Furthermore, the hyper-reduction of parametric systems is an open topic in both simulation-based and simulation-free setups.

- Within this thesis, solely systems with linear Dirichlet boundary conditions are addressed. Many real-life applications, however, are characterized by nonlinear constraints such as joints, rigid body elements or contacts. One way to express these nonlinear constraints systematically is the use of Lagrange multipliers, where the nonlinear constraint equations are considered explicitly. It has not been clarified yet, to what extent the concepts of simulation-free reduced bases and hyper-reduction are applicable to systems with nonlinear constraints. For possible adaptations, novel methods are required to handle these systems.

- The systems covered in this thesis are only characterized by viscous damping. It does, however, not recognize the geometric nonlinearity since it is not objective and results in damping forces for pure rigid body motions (cf. Section 2.1). A more accurate damping approximation can be expected from viscoelastic material models like the Kelvin-Voigt model, where the stress originated from viscous damping is proportional to the rate of deformation of the nonlinear strain measure. These models, however, result in equations of motion with a different mathematical structure compared to the ones discussed here, which stem from hyperelastic materials. Hence, further research is necessary to adapt the simulation-free methods discussed and developed here to the field of geometric nonlinearity with viscoelasticity.

- In several industrial applications, nonlinear structural dynamics problems are coupled with other physical areas such as thermal effects, plasticity, fatigue or fluid structure interaction. Fundamental research is necessary to answer the questions, how these coupled systems can be reduced in a simulation-free manner.

- Untapped potential lies in the transformation and adaptation of the discussed simulation-free reduced basis methods and the hyper-reduction methods to substructure frameworks. This would allow to assemble large nonlinear systems using small, reduced substructures with the potential to higher speedup rates, particularly for large systems with complex dynamics requiring a high number of basis vectors. Notably the polynomial tensors hyper-reduction technique could unfold its potential, when the system to be reduced is divided in small parts of which each part requires only a small number of basis vectors. Since the computational costs of this method are associated with $\mathcal{O}(n^4)$ with respect to the dimension $n$ of the reduced system, or in this context with the substructure, impressive online cost reductions even for large systems are feasible in principle.

- The polynomial tensors hyper-reduction suffers of the curse of dimensionality making it not attractive for reduced systems with large $n$. In principle, a low rank approximation of the higher order tensors could allow for further speedup factors making the polynomial tensors hyper-reduction accessible for higher reduction orders $n$. However, approaches for low rank approximations for higher order symmetric arrays as well as identification techniques still need to be developed.

# Bibliography


[AP03] P. G. A. Cizmas and A. Palacios. "Proper orthogonal decomposition of turbine rotor-stator interaction". In: *Journal of propulsion and power* 19.2 (2003), pp. 268–281.

[AJ02] R. G. Aarts and J. B. Jonker. "Dynamic simulation of planar flexible link manipulators using adaptive modal integration". In: *Multibody System Dynamics* 7.1 (2002), pp. 31–50.

[ACM99] K. Abdel-Motaglay, R. Chen, and C. Mei. "Nonlinear flutter of composite panels under yawed supersonic flow using finite elements". In: *AIAA journal* 37.9 (1999), pp. 1025–1032.

[Add+15] K. C. Addepalli, N. Remisoski, A. Sleath, and S. Liu. "A Multibody Dynamics Approach to Leaf Spring Simulation for Upfront Analyses". In: *SAE Technical Paper*. SAE International, June 2015.

[Akg94] M. A. Akgun. "New family of modal methods for calculating eigenvector derivatives". In: *AIAA Journal* 32.2 (1994), pp. 379–386.

[All03] R. J. Allemang. "The modal assurance criterion–twenty years of use and abuse". In: *Sound and vibration* 37.8 (2003), pp. 14–23.

[ASB78] B. Almroth, P. Stern, and F. Brogan. "Automatic choice of global shape functions in structural analysis". In: *Aiaa Journal* 16.5 (1978), pp. 525–528.

[Ams10] D. Amsallem. "Interpolation on manifolds of CFD-based fluid and finite element-based structural reduced-order models for on-line aeroelastic predictions". PhD thesis. Stanford University, 2010.

[Ams+09] D. Amsallem, J. Cortial, K. Carlberg, and C. Farhat. "A method for interpolating on manifolds structural dynamics reduced-order models". In: *International journal for numerical methods in engineering* 80.9 (2009), pp. 1241–1258.

[AF11] D. Amsallem and C. Farhat. "An online method for interpolating linear parametric reduced-order models". In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2169–2198.

[AF08] D. Amsallem and C. Farhat. "Interpolation method for adapting reduced-order models and application to aeroelasticity". In: *AIAA Journal* 46.7 (2008), pp. 1803–1813.

[AKJ08] S. S. An, T. Kim, and D. L. James. "Optimizing cubature for efficient integration of subspace deformations". In: *ACM Transactions on Graphics (TOG)* 27.5 (2008), No. 165.

[AHS14] H. Antil, M. Heinkenschloss, and D. C. Sorensen. "Application of the discrete empirical interpolation method to reduced order modeling of nonlinear and parametric systems". In: *Reduced order methods for modeling and computational reduction*. Springer, 2014, pp. 101–136.

[ASG06] A. Antoulas, D. Sorensen, and S. Gugercin. "A survey of model reduction methods for large-scale systems". In: *Contemporary mathematics* 280 (2006), pp. 193–220.

[Ant09] A. C. Antoulas. *Approximation of large-scale dynamical systems*. Vol. 6. Society for Industrial and Applied Mathematics, 2009.

[Ant05] A. C. Antoulas. *Approximation of large-scale dynamical systems*. Vol. 6. Siam, 2005.

[Ast04] P. Astrid. *Reduction of process simulation models: a proper orthogonal decomposition approach*. Technische Universiteit Eindhoven PhD-thesis, 2004.

[Ast+08] P. Astrid, S. Weiland, K. Willcox, and T. Backx. "Missing point estimation in models described by proper orthogonal decomposition". In: *IEEE Transactions on Automatic Control* 53.10 (2008), pp. 2237–2251.

[AV01] M. Azeez and A. Vakakis. "Proper orthogonal decomposition (POD) of a class of vibroimpact oscillations". In: *Journal of Sound and Vibration* 240.5 (2001), pp. 859–889.

[AV99] M. F. A. Azeez and A. F. Vakakis. "Numerical and experimental analysis of a continuous overhung rotor undergoing vibro-impacts". In: *International journal of non-linear mechanics* 34.3 (1999), pp. 415–435.

[Bai+00] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst. *Templates for the solution of algebraic eigenvalue problems: a practical guide*. SIAM, 2000.

[BS05] Z. Bai and Y. Su. "Dimension reduction of large-scale second-order dynamical systems via a second-order Arnoldi method". In: *SIAM Journal on Scientific Computing* 26.5 (2005), pp. 1692–1709.

[BJ05] J. Barbič and D. L. James. "Real-Time subspace integration for St. Venant-Kirchhoff deformable models". In: *ACM Transactions on Graphics (TOG)*. Vol. 24. 3. ACM. 2005, pp. 982–990.

[BV10] T. K. Barlas and G. Van Kuik. "Review of state of the art in smart rotor control research for wind turbines". In: *Progress in Aerospace Sciences* 46.1 (2010), pp. 1–27.

[Bar+04] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. "An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations". In: *Comptes Rendus Mathematique* 339.9 (2004), pp. 667–672.

[Bat06] K.-J. Bathe. *Finite element procedures*. Klaus-Jurgen Bathe, 2006.

[BG93] O. Bauchau and D. Guernsey. "On the choice of appropriate bases for nonlinear dynamic modal analysis". In: *Journal of the American Helicopter Society* 38.4 (1993), pp. 28–36.

[BG16] C. Beattie and S. Gugercin. "Model Reduction by Rational Interpolation". In: *Model Reduction and Approximation: Theory and Algorithms* (2016).

[BLM00] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons, 2000.

[BGW15] P. Benner, S. Gugercin, and K. Willcox. "A Survey of Projection-based Model Reduction Methods for Parametric Dynamical Systems". In: *SIAM Review* 57.4 (2015), pp. 483–531.

[BCB05] M. Bergmann, L. Cordier, and J.-P. Brancher. "Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model". In: *Physics of fluids* 17.9 (2005), pp. 97–101.

[Ber92] G. Berkooz. "Observations on the proper orthogonal decomposition". In: *Studies in Turbulence*. Springer, 1992, pp. 229–247.

[BHL93] G. Berkooz, P. Holmes, and J. L. Lumley. "The proper orthogonal decomposition in the analysis of turbulent flows". In: *Annual review of fluid mechanics* 25.1 (1993), pp. 539–575.

[Bes+13] B. Besselink, U. Tabak, A. Lutowska, N. van de Wouw, H. Nijmeijer, D. J. Rixen, M. Hochstenbach, and W. Schilders. "A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control". In: *Journal of Sound and Vibration* 332.19 (2013), pp. 4403–4422.

[BT05] T. Betcke and L. N. Trefethen. "Reviving the method of particular solutions". In: *SIAM review* 47.3 (2005), pp. 469–491.

[BG73] Å. Björck and G. H. Golub. "Numerical methods for computing angles between linear subspaces". In: *Mathematics of computation* 27.123 (1973), pp. 579–594.

[Blo+15] B. Blockmans, T. Tamarozzi, F. Naets, and W. Desmet. "A nonlinear parametric model reduction method for efficient gear contact simulations". In: *International Journal for Numerical Methods in Engineering* 102.5 (2015), pp. 1162–1191.

[BW97] J. Bonet and R. D. Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.

[Bou+14] L. Boucinha, A. Ammar, A. Gravouil, and A. Nouy. "Ideal minimal residual-based proper generalized decomposition for non-symmetric multi-field models – Application to transient elastodynamics in space-time domain". In: *Computer Methods in Applied Mechanics and Engineering* 273 (2014), pp. 56–76.

[BGA13] L. Boucinha, A. Gravouil, and A. Ammar. "Space–time proper generalized decompositions for the resolution of transient elastodynamic models". In: *Computer Methods in Applied Mechanics and Engineering* 255 (2013), pp. 67–88.

[BMS16] B. Brands, J. Mergheim, and P. Steinmann. "Reduced-order modelling for linear heat conduction with parametrised moving heat sources". In: *GAMM-Mitteilungen* 39.2 (2016), pp. 170–188.

[Bro15] R. W. Brockett. *Finite dimensional linear systems*. SIAM, 2015.

[BDG07] O. Brüls, P. Duysinx, and J.-C. Golinval. "The global modal parameterization for non-linear model-order reduction in flexible multibody dynamics". In: *International journal for numerical methods in engineering* 69.5 (2007), pp. 948–977.

[BDW03] T. Bui-Thanh, M. Damodaran, and K. Willcox. "Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics". In: *AIAA paper* 4213 (2003).

[CW09] L. Cai and R. E. White. "Reduction of model order based on proper orthogonal decomposition for lithium-ion battery simulations". In: *Journal of The Electrochemical Society* 156.3 (2009), A154–A161.

[Cap+02] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popović. "Interactive skeleton-driven dynamic deformations". In: *ACM Transactions on Graphics (TOG)*. Vol. 21. 3. ACM. 2002, pp. 586–593.

[CSM14] E. Capiez-Lernout, C. Soize, and M.-P. Mignolet. "Post-buckling nonlinear static and dynamical analyses of uncertain cylindrical shells and experimental validation". In: *Computer Methods in Applied Mechanics and Engineering* 271 (2014), pp. 210–230.

[CBF11] K. Carlberg, C. Bou-Mosleh, and C. Farhat. "Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations". In: *International Journal for Numerical Methods in Engineering* 86.2 (2011), pp. 155–181.

[CTB15] K. Carlberg, R. Tuminaro, and P. Boggs. "Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics". In: *SIAM Journal on Scientific Computing* 37.2 (2015), B153–B184.

[Cha+06] Y. Chahlaoui, D. Lemonnier, A. Vandendorpe, and P. V. Dooren. "Second-order balanced truncation". In: *Linear Algebra and its Applications* 415.2 (2006). Special Issue on Order Reduction of Large-Scale Systems, pp. 373–384.

[CE91] C. Chang and J. J. Engblom. "Nonlinear dynamical response of impulsively loaded structures-A reduced basis approach". In: *AIAA Journal* 29.4 (1991), pp. 613–618.

[Cha+17] T. Chapman, P. Avery, P. Collins, and C. Farhat. "Accelerated mesh sampling for the hyper reduction of nonlinear computational models". In: *International Journal for Numerical Methods in Engineering* 109.12 (2017), pp. 1623–1654.

[CBG16] S. Chaturantabut, C. Beattie, and S. Gugercin. "Structure-preserving model reduction for nonlinear port-Hamiltonian systems". In: *SIAM Journal on Scientific Computing* 38.5 (2016), B837–B865.

[CS10] S. Chaturantabut and D. C. Sorensen. "Nonlinear model reduction via discrete empirical interpolation". In: *SIAM Journal on Scientific Computing* 32.5 (2010), pp. 2737–2764.

[CZ06] D. Chelidze and W. Zhou. "Smooth orthogonal decomposition-based vibration mode identification". In: *Journal of Sound and Vibration* 292.3 (2006), pp. 461–473.

[Che+04] J. Chen, S.-M. Kang, J. Zou, C. Liu, and J. E. Schutt-Ainé. "Reduced-order modeling of weakly nonlinear MEMS devices with Taylor-series expansion and Arnoldi approach". In: *Journal of Microelectromechanical systems* 13.3 (2004), pp. 441–451.

[CH93] J. Chung and G. Hulbert. "A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-$\alpha$ method". In: *Journal of applied mechanics* 60.2 (1993), pp. 371–375.

[CN01] J. A. Clarke and R. Namburu. *The eXtensible Data Model and Format: A High Performance Data Hub for Connecting Parallel Codes and Tools*. Tech. rep. DTIC Document, 2001.

[CDM15] A. Corigliano, M. Dossi, and S. Mariani. "Model Order Reduction and domain decomposition strategies for the solution of the dynamic elastic–plastic structural problem". In: *Computer Methods in Applied Mechanics and Engineering* 290 (2015), pp. 127–155.

[CCI14] A. Cosimo, A. Cardona, and S. Idelsohn. "Improving the k-compressibility of hyper reduced order models with moving sources: applications to welding and phase change problems". In: *Computer Methods in Applied Mechanics and Engineering* 274 (2014), pp. 237–263.

[Cra81] R. R. Craig. *Structural dynamics: an introduction to computer methods*. John Wiley & Sons Inc, 1981.

[CB68] R. R. Craig Jr and M. C. Bampton. "Coupling of substructures for dynamic analyses." In: *AIAA Journal* 6.7 (1968), pp. 1313–1319.

[CK06] R. R. Craig and A. J. Kurdila. *Fundamentals of structural dynamics*. John Wiley & Sons, 2006.

[C+12] M. A. Crisfield, J. J. Remmers, C. V. Verhoosel, et al. *Nonlinear finite element analysis of solids and structures*. John Wiley & Sons, 2012.

[CM69] E. Cuthill and J. McKee. "Reducing the bandwidth of sparse symmetric matrices". In: *Proceedings of the 1969 24th national conference*. ACM. 1969, pp. 157–172.

[DS84] J. Davis and R. Skelton. "Another balanced controller reduction algorithm". In: *Systems & Control Letters* 4.2 (1984), pp. 79–83.

[DG03] P. De Boe and J.-C. Golinval. “Principal component analysis of a piezosensor array for damage localization”. In: *Structural health monitoring* 2.2 (2003), pp. 137–144.

[DV00] H. De Boer and F. Van Keulen. “Refined semi-analytical design sensitivities”. In: *International journal of solids and structures* 37.46 (2000), pp. 6961–6980.

[DM01] I. S. Dhillon and D. S. Modha. “Concept decompositions for large sparse text data using clustering”. In: *Machine learning* 42.1 (2001), pp. 143–175.

[DP92] J. Dickens and K. Pool. “Modal truncation vectors and periodic time domain analysis applied to a cyclic symmetry structure”. In: *Computers & structures* 45.4 (1992), pp. 685–696.

[AM02] A. Al-Dmour and K. Mohammad. “Active control of flexible structures using principal component analysis in the time domain”. In: *Journal of sound and vibration* 253.3 (2002), pp. 545–569.

[Dod] T. Dodt. *Introducing the 787*. Presentation at the International Society of Air Safety Investigators. URL: `http://www.isasi.org/Documents/library/technical-papers/2011/Introducing-787.pdf`, visited on 04/28/2017.

[DG16] Z. Drmac and S. Gugercin. “A new selection operator for the discrete empirical interpolation method—Improved a priori error bound and extensions”. In: *SIAM Journal on Scientific Computing* 38.2 (2016), A631–A648.

[Epu03] B. Epureanu. “A parametric analysis of reduced order models of viscous flows in turbomachinery”. In: *Journal of fluids and structures* 17.7 (2003), pp. 971–982.

[ES95] R. Everson and L. Sirovich. “Karhunen–Loeve procedure for gappy data”. In: *Journal of the Optical Society of America A* 12.8 (1995), pp. 1657–1664.

[Ewi84] D. J. Ewins. *Modal testing: theory and practice*. Vol. 15. Chichester: Research studies press, 1984.

[Far+14] C. Farhat, P. Avery, T. Chapman, and J. Cortial. “Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency”. In: *International Journal for Numerical Methods in Engineering* 98 (2014), pp. 625–662.

[FCA13] C. Farhat, T. Chapman, and P. Avery. “ECSW: An energy-based structure-preserving method for the hyper reduction of nonlinear finite element reduced-order models”. In: *International Journal for Numerical Methods in Engineering* 00 (2013), pp. 1–33.

[FCA15] C. Farhat, T. Chapman, and P. Avery. “Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models”. In: *International Journal for Numerical Methods in Engineering* 102.5 (2015), pp. 1077–1110.

[FF08a] U. Farooq and B. Feeny. "Smooth orthogonal decomposition for modal analysis of randomly excited systems". In: *Journal of Sound and Vibration* 316.1 (2008), pp. 137–146.

[FF08b] B. Feeny and U. Farooq. "A nonsymmetric state-variable decomposition for modal analysis". In: *Journal of Sound and Vibration* 310.4 (2008), pp. 792–800.

[FK98] B. Feeny and R. Kappagantu. "On the physical interpretation of proper orthogonal modes in vibrations". In: *Journal of Sound and Vibration* 211.4 (1998), pp. 607–616.

[FL03] B. Feeny and Y. Liang. "Interpreting proper orthogonal modes of randomly excited vibration systems". In: *Journal of Sound and Vibration* 265.5 (2003), pp. 953–966.

[FE10] J. Fehr and P. Eberhard. "Error-controlled model reduction in flexible multibody dynamics". In: *Journal of Computational and Nonlinear Dynamics* 5.3 (2010), pp. 031005–031005-8.

[Fol+11] M. Folk, G. Heber, Q. Koziol, E. Pourmal, and D. Robinson. "An overview of the HDF5 technology suite and its applications". In: *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*. ACM. 2011, pp. 36–47.

[FI99] M. I. Friswell and D. J. Inman. "Sensor validation for smart structures". In: *Journal of intelligent material systems and structures* 10.12 (1999), pp. 973–982.

[FT01] Y. C. Fung and P. Tong. *Classical and computational solid mechanics*. Vol. 1. World Scientific Publishing Co Inc, 2001.

[Gaw97] W. Gawronski. "Actuator and sensor placement for structural testing and control". In: *Journal of Sound and Vibration* 208.1 (1997), pp. 101–109.

[Gaw06] W. Gawronski. *Balanced control of flexible structures*. Vol. 211. Lecture Notes in Control and Information Sciences. Springer, 2006.

[Gaw04] W. K. Gawronski. *Dynamics and control of structures: A modal approach*. Springer Science & Business Media, 2004.

[GL96] W. Gawronski and K. Lim. "Balanced actuator and sensor placement for flexible structures". In: *International Journal of Control* 65.1 (1996), pp. 131–145.

[GR14] M. Géradin and D. J. Rixen. *Mechanical vibrations: theory and application to structural dynamics*. John Wiley & Sons, 2014.

[GR09] C. Geuzaine and J.-F. Remacle. "Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities". In: *International journal for numerical methods in engineering* 79.11 (2009), pp. 1309–1331.

[GAM09] S. Ghoman, M. S. Azzouz, and C. Mei. "Time domain method for nonlinear flutter of curved panels under yawed supersonic flow at elevated temperature". In: *Proceedings of the 50th Structures, Structural Dynamics and Materials Conference*. Palm Springs, California, April 2009, pp. 2009–2598.

[GMW81] P. E. Gill, W. Murray, and M. H. Wright. *Practical optimization*. Academic press, 1981.

[GV12] G. H. Golub and C. F. Van Loan. *Matrix computations*. Vol. 3. JHU Press, 2012.

[GSD08] P. Gonçalves, F. Silva, and Z. Del Prado. "Low-dimensional models for the nonlinear vibration analysis of cylindrical shells based on a perturbation procedure and proper orthogonal decomposition". In: *Journal of Sound and Vibration* 315.3 (2008), pp. 641–663.

[GW08] A. Griewank and A. Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.

[Gri97] E. J. Grimme. "Krylov projection methods for model reduction". PhD thesis. University of Illinois at Urbana-Champaign Urbana-Champaign, IL, 1997.

[GRR16] F. M. Gruber, J. B. Rutzmoser, and D. J. Rixen. "Generalized Craig-Bampton Method Using Robin Boundary Conditions". In: *Topics in Modal Analysis & Testing, Volume 10*. Springer, 2016, pp. 111–115.

[GKA16] L. C. Guerin, R. J. Kuether, and M. S. Allen. "Considerations for Indirect Parameter Estimation in Nonlinear Reduced Order Models". In: *Nonlinear Dynamics, Volume 1*. Springer, 2016, pp. 327–342.

[GA04] S. Gugercin and A. C. Antoulas. "A survey of model reduction by balanced truncation and some new results". In: *International Journal of Control* 77.8 (2004), pp. 748–766.

[GAB08] S. Gugercin, A. C. Antoulas, and C. Beattie. "$\mathcal{H}_2$ model reduction for large-scale linear dynamical systems". In: *SIAM journal on matrix analysis and applications* 30.2 (2008), pp. 609–638.

[Guo11] X. Guo. "Energy-Weighted Modes Selection in Reduced-Order Nonlinear Simulations". In: *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. Ed. by AIAA. AIAA 2011-2063. AIAA. April 2011.

[GM03] X. Guo and C. Mei. "Using aeroelastic modes for nonlinear panel flutter at arbitrary supersonic yawed angle". In: *AIAA journal* 41.2 (2003), pp. 272–279.

[Guy65] R. J. Guyan. "Reduction of stiffness and mass matrices". In: *AIAA Journal* 3.2 (1965), pp. 380–380.

[HTD00] K. C. Hall, J. P. Thomas, and E. H. Dowell. "Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows". In: *AIAA Journal* 38.10 (2000), pp. 1853–1862.

[HD01] F. M. Hemez and S. W. Doebling. "Review and assessment of model updating for non-linear, transient dynamics". In: *Mechanical Systems and Signal Processing* 15.1 (2001), pp. 45–74.

[HCF17] J. A. Hernandez, M. A. Caicedo, and A. Ferrer. "Dimensional hyper-reduction of nonlinear finite element models via empirical cubature".

In: *Computer methods in applied mechanics and engineering* 313 (2017), pp. 687–722.

[HKS01] Hibbitt, Karlsson, and Sorensen. *ABAQUS/standard User's Manual*. Vol. 1. 2001.

[HHT77] H. M. Hilber, T. J. Hughes, and R. L. Taylor. "Improved numerical dissipation for time integration algorithms in structural dynamics". In: *Earthquake Engineering & Structural Dynamics* 5.3 (1977), pp. 283–292.

[Hil+11] K. Hildebrandt, C. Schulz, C. V. Tycowicz, and K. Polthier. "Interactive surface modeling using modal analysis". In: *ACM Transactions on Graphics (TOG)* 30.5 (2011), p. 119.

[HG08] J. J. Hollkamp and R. W. Gordon. "Reduced-order models for nonlinear response prediction: Implicit condensation and expansion". In: *Journal of Sound and Vibration* 318.4 (2008), pp. 1139–1153.

[HGS05] J. J. Hollkamp, R. W. Gordon, and S. M. Spottswood. "Nonlinear modal models for sonic fatigue response prediction: a comparison of methods". In: *Journal of Sound and Vibration* 284.3 (2005), pp. 1145–1163.

[Hol12] P. Holmes. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012.

[Hol00] G. A. Holzapfel. *Nonlinear solid mechanics: A continuum approach for engineering*. Vol. 24. Wiley Chichester, 2000.

[IC85a] S. R. Idelsohn and A. Cardona. "A load-dependent basis for reduced nonlinear structural dynamics". In: *Computers & Structures* 20.1 (1985), pp. 203–210.

[IC85b] S. R. Idelsohn and A. Cardona. "A reduction method for nonlinear structural dynamic analysis". In: *Computer Methods in Applied Mechanics and Engineering* 49.3 (1985), pp. 253–279.

[Iro65] B. Irons. "Structural eigenvalue problems - elimination of unwanted variables". In: *AIAA Journal* 3.5 (1965), pp. 961–962.

[JE92] B. Jacob and N. Ebecken. "Adaptive reduced integration method for nonlinear structural dynamic analysis". In: *Computers & structures* 45.2 (1992), pp. 333–347.

[Jai15] S. Jain. "Model Order Reduction for Non-Linear Structural Dynamics". MA thesis. Delft University of Technology, 2015.

[Jai+17] S. Jain, P. Tiso, J. B. Rutzmoser, and D. J. Rixen. "A quadratic manifold for model order reduction of nonlinear structural dynamics". In: *Computers & Structures* 188 (2017), pp. 80–94.

[Jon84] E. A. Jonckheere. "Principal component analysis of flexible systems—Open-loop case". In: *Mathematical Theory of Networks and Systems*. Springer. 1984, pp. 494–512.

[KF99] R. Kappagantu and B. Feeny. “An "optimal" modal reduction of a system with frictional excitation”. In: *Journal of Sound and vibration* 224.5 (1999), pp. 863–877.

[Kar46] K. Karhunen. “Über lineare Methoden in der Wahrscheinlichkeitsrechnung”. In: *Annals of Academic Science Fennicae, Series A1 Mathematics and Physics* 37 (1946), pp. 3–79.

[Ker+04] G. Kerschen, P. De Boe, J.-C. Golinval, and K. Worden. “Sensor validation using principal component analysis”. In: *Smart Materials and Structures* 14.1 (2004), pp. 36–42.

[KG02] G. Kerschen and J.-C. Golinval. “Physical interpretation of the proper orthogonal modes using the singular value decomposition”. In: *Journal of Sound and Vibration* 249.5 (2002), pp. 849–865.

[Ker+05] G. Kerschen, J.-C. Golinval, A. F. Vakakis, and L. A. Bergman. “The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview”. In: *Nonlinear Dynamics* 41.1-3 (2005), pp. 147–169.

[KAS07] M. Khalil, S. Adhikari, and A. Sarkar. “Linear system identification using proper orthogonal decomposition”. In: *Mechanical Systems and Signal Processing* 21.8 (2007), pp. 3123–3145.

[Kim+13] K. Kim, A. G. Radu, X. Wang, and M. P. Mignolet. “Nonlinear reduced order modeling of isotropic and functionally graded plates”. In: *International Journal of Non-Linear Mechanics* 49 (2013), pp. 100–110.

[KS09] N.-H. Kim and B. V. Sankar. *Introduction to finite element analysis and design*. Wiley, 2009.

[Kim98] T. Kim. “Frequency-domain Karhunen-Loeve method and its application to linear dynamic systems”. In: *AIAA Journal* 36.11 (1998), pp. 2117–2123.

[KA02] A. V. Knyazev and M. E. Argentati. “Principal angles between subspaces in an A-based scalar product: algorithms and perturbation estimates”. In: *SIAM Journal on Scientific Computing* 23.6 (2002), pp. 2008–2040.

[Kon+13] Y. Kong, M. Z. Omar, L. Chua, and S. Abdullah. “Explicit nonlinear finite element geometric analysis of parabolic leaf springs under various loads”. In: *The Scientific World Journal* 2013 (2013).

[Kos43] D. D. Kosambi. “Statistics in function space”. In: *Journal of Indian Mathematical Society* 7 (1943), pp. 76–88.

[KLM01] P. Krysl, S. Lall, and J. Marsden. “Dimensional model reduction in non-linear finite element dynamics of solids and structures”. In: *International Journal for numerical methods in engineering* 51.4 (2001), pp. 479–504.

[KA16] R. J. Kuether and M. S. Allen. “Validation of nonlinear reduced order models with time integration targeted at nonlinear normal modes”. In: *Nonlinear Dynamics, Volume 1*. Springer, 2016, pp. 363–375.

[KA14] R. J. Kuether and M. S. Allen. "Craig-Bampton substructuring for geometrically nonlinear subcomponents". In: *Dynamics of Coupled Structures, Volume 1*. Springer, 2014, pp. 167–178.

[KAH15] R. J. Kuether, M. S. Allen, and J. J. Hollkamp. "Modal substructuring of geometrically nonlinear finite element models with interface reduction". In: *AIAA Journal* 54.2 (2015), pp. 1–12.

[KBA14] R. J. Kuether, M. R. Brake, and M. S. Allen. "Evaluating convergence of reduced order models using nonlinear normal modes". In: *Model Validation and Uncertainty Quantification, Volume 3*. Springer, 2014, pp. 287–300.

[LPS15] S. K. Lam, A. Pitrou, and S. Seibert. "Numba: A llvm-based python jit compiler". In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. ACM. 2015, pp. 1–6.

[LD06] F. Lanata and A. Del Grosso. "Damage detection and localization for continuous static monitoring of structures using a proper orthogonal decomposition of signals". In: *Smart Materials and Structures* 15.6 (2006), pp. 1811–1829.

[LHB04] T. J. Larsen, A. M. Hansen, and T. Buhl. "Aeroelastic effects of large blade deflections for wind turbines". In: *Special topic conference: The science of making torque from wind*. 2004, pp. 238–246.

[LH95] C. L. Lawson and R. J. Hanson. *Solving least squares problems*. Vol. 15. SIAM, 1995.

[LE06] M. Lehner and P. Eberhard. "On the use of moment-matching to build reduced order models in flexible multibody dynamics". In: *Multibody System Dynamics* 16.2 (2006), pp. 191–211.

[LKG03] V. Lenaerts, G. Kerschen, and J.-C. Golinval. "Identification of a continuous structure with a geometrical non-linearity. Part II: Proper orthogonal decomposition". In: *Journal of Sound and vibration* 262.4 (2003), pp. 907–919.

[Lia+02a] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu. "Proper orthogonal decomposition and its applications—Part I: Theory". In: *Journal of Sound and vibration* 252.3 (2002), pp. 527–544.

[Lia+02b] Y. Liang, W. Lin, H. Lee, S. Lim, K. Lee, and H. Sun. "Proper orthogonal decomposition and its applications–part II: Model reduction for MEMS dynamical analysis". In: *Journal of Sound and Vibration* 256.3 (2002), pp. 515–532.

[LF07] T. Lieu and C. Farhat. "Adaptation of aeroelastic reduced-order models and application to an F-16 configuration". In: *AIAA journal* 45.6 (2007), pp. 1244–1257.

[LL04] T. Lieu and M. Lesoinne. "Parameter adaptation of reduced order models for three-dimensional flutter analysis". In: *42nd AIAA Aerospace Sciences Meeting and Exhibit*. 2004, p. 888.

[Loè48] M. Loève. "Fonctions Aléatoires du Second Ordre". In: *Processus stochastiques et mouvement Brownien* (1948).

[Lor94] B. J. W. S. Lord Rayleigh. *Theory of Sound, Vol. 1*. 2nd. London and New York (first edition in 1877): Macmillan and Co., 1894.

[LTO13] F. A. Lülf, D.-M. Tran, and R. Ohayon. "Reduced bases for nonlinear structural dynamic systems: A comparative study". In: *Journal of Sound and Vibration* 332.15 (2013), pp. 3897–3921.

[Lum07] J. L. Lumley. *Stochastic tools in turbulence*. Courier Corporation, 2007.

[Lut10] M. Lutz. *Programming Python: Powerful Object-Oriented Programming*. O'Reilly, 2010.

[Mac71] R. H. MacNeal. "A hybrid method of component mode synthesis". In: *Computers & Structures* 1.4 (1971), pp. 581–601.

[McC72] C. W. McCormick. *The NASTRAN User's Manual (Level 15)*. 1972.

[McE+01] M. McEwan, J. Wright, J. Cooper, and A. Leung. "A finite element/modal technique for nonlinear plate and stiffened panel response prediction". In: *Proceedings of the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit Technical Papers*. 2001, pp. 3061–3070.

[McK12] W. McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. O'Reilly, 2012.

[Mig+13] M. P. Mignolet, A. Przekop, S. A. Rizzi, and S. M. Spottswood. "A review of indirect/non-intrusive reduced order modeling of nonlinear geometric structures". In: *Journal of Sound and Vibration* 332.10 (2013), pp. 2437–2460.

[MA13] D. Millán and M. Arroyo. "Nonlinear manifold learning for model reduction in finite elastodynamics". In: *Computer Methods in Applied Mechanics and Engineering* 261 (2013), pp. 118–131.

[Moo81] B. Moore. "Principal component analysis in linear systems: Controllability, observability, and model reduction". In: *IEEE transactions on automatic control* 26.1 (1981), pp. 17–32.

[Moo65] G. E. Moore. "Cramming more components onto integrated circuits". In: *Electronics* 38.8 (1965).

[Mou+99] C. Mousseau, T. Laursen, M. Lidberg, and R. Taylor. "Vehicle dynamics simulations with coupled multibody and finite element models". In: *Finite Elements in Analysis and Design* 31.4 (1999), pp. 295–315.

[Muj+11] L. Mujica, J. Rodellar, A. Fernandez, and A. Güemes. "Q-statistic and T2-statistic PCA-based measures for damage assessment in structures". In: *Structural Health Monitoring* 10.5 (2011), pp. 539–553.

[MR76] C. Mullis and R. Roberts. "Synthesis of minimum roundoff noise fixed point digital filters". In: *IEEE Transactions on Circuits and Systems* 23.9 (1976), pp. 551–562.

[MR03] A. A. Muravyov and S. A. Rizzi. "Determination of nonlinear stiffness with application to random vibration of geometrically nonlinear structures". In: *Computers & Structures* 81.15 (2003), pp. 1513–1523.

[Nae+11] F. Naets, G. Heirman, D. Vandepitte, and W. Desmet. "Inertial force term approximations for the use of global modal parameterization for planar mechanisms". In: *International journal for numerical methods in engineering* 85.4 (2011), pp. 518–536.

[Nae+12] F. Naets, T. Tamarozzi, G. H. Heirman, and W. Desmet. "Real-time flexible multibody simulation with Global Modal Parameterization". In: *Multibody System Dynamics* 27.3 (2012), pp. 267–284.

[Nas78] M. Nash. "Nonlinear structural dynamics by finite element model synthesis". PhD thesis. Imperial College London (University of London), 1978.

[Nel76] R. B. Nelson. "Simplified calculation of eigenvector derivatives". In: *AIAA Journal* 14.9 (1976), pp. 1201–1205.

[New59] N. M. Newmark. "A method of computation for structural dynamics". In: *Journal of the engineering mechanics division* 85.3 (1959), pp. 67–94.

[NPP08] N. Nguyen, A. Patera, and J. Peraire. "A 'best points' interpolation method for efficient approximation of parametrized functions". In: *International journal for numerical methods in engineering* 73.4 (2008), pp. 521–543.

[NP08] N. Nguyen and J. Peraire. "An efficient reduced-order modeling approach for non-linear parametrized partial differential equations". In: *International Journal for Numerical Methods in Engineering* 76.1 (2008), pp. 27–55.

[Noo82] A. K. Noor. "On making large nonlinear problems small". In: *Computer methods in applied mechanics and engineering* 34.1 (1982), pp. 955–985.

[NAP93] A. K. Noor, C. Andersen, and J. M. Peters. "Reduced basis technique for nonlinear vibration analysis of composite panels". In: *Computer methods in applied mechanics and engineering* 103.1-2 (1993), pp. 175–186.

[NP80] A. K. Noor and J. M. Peters. "Reduced basis technique for nonlinear analysis of structures". In: *AIAA Journal* 18.4 (1980), pp. 455–462.

[Now+13] C. Nowakowski, P. Kürschner, P. Eberhard, and P. Benner. "Model reduction of an elastic crankshaft for elastic multibody simulations". In: *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 93.4 (2013), pp. 198–216.

[Obu54] A. Obukhov. "Statistical description of continuous fields". In: *Transactions of the Geophysical International Academy Nauk USSR* 24.24 (1954), pp. 3–42.

[OAR89] J. O'Callahan, P. Avitabile, and R. Riemer. "System equivalent reduction expansion process (SEREP)". In: *Proceedings of the 7th international modal analysis conference*. Vol. 1. Union College Schnectady, NY. 1989, pp. 29–37.

[Ogd97] R. W. Ogden. *Non-linear elastic deformations*. Courier Corporation, 1997.

[Oli12] T. Oliphant. “Numba python bytecode to LLVM translator”. In: *Proceedings of the Python for Scientific Computing Conference (SciPy)*. 2012.

[Oma+04] M. A. Omar, A. A. Shabana, A. Mikkola, W.-Y. Loh, and R. Basch. “Multibody system modeling of leaf springs”. In: *Modal Analysis* 10.11 (2004), pp. 1601–1638.

[Pan14] H. K. Panzer. “Model order reduction by krylov subspace methods with global error bounds and automatic choice of parameters”. PhD thesis. Technische Universität München, 2014.

[Peh+14] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz. “Localized discrete empirical interpolation method”. In: *SIAM Journal on Scientific Computing* 36.1 (2014), A168–A192.

[PWM14] R. Perez, X. Wang, and M. P. Mignolet. “Nonintrusive Structural Dynamic Reduced Order Modeling for Large Deformations: Enhancements for Complex Structures”. In: *Journal of Computational and Nonlinear Dynamics* 9.3 (2014), pp. 031008-1–12.

[Phl+14] G. Phlipot, X. Wang, M. P. Mignolet, L. Demasi, and R. Cavallaro. “Nonintrusive reduced order modeling for the nonlinear geometric response of some joined wings”. In: *Proceedings of the AIAA Science and Technology Forum and Exposition (SciTech2014), National Harbor, MD*. 2014.

[Pol09] E. Polizzi. “Density-matrix-based algorithm for solving eigenvalue problems”. In: *Physical Review B* 79.11 (2009), p. 115112.

[Prz+04] A. Przekop, M. S. Azzouz, X. Guo, C. Mei, and L. Azrar. “Finite element multiple-mode approach to nonlinear free vibrations of shallow shells”. In: *AIAA Journal* 42.11 (2004), pp. 2373–2381.

[PGR12] A. Przekop, X. Guo, and S. A. Rizzi. “Alternative modal basis selection procedures for reduced-order nonlinear random response simulation”. In: *Journal of sound and vibration* 331.17 (2012), pp. 4005–4024.

[Pug53] V. S. Pugachev. “The general theory of correlation of random functions”. In: *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya* 17.5 (1953), pp. 401–420.

[QSS10] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*. Vol. 37. Springer Science & Business Media, 2010.

[RR14a] A. Radermacher and S. Reese. “Model reduction in elastoplasticity: proper orthogonal decomposition combined with adaptive sub-structuring”. In: *Computational Mechanics* 54.3 (2014), pp. 677–687.

[RR16] A. Radermacher and S. Reese. “POD-based model reduction with empirical interpolation applied to nonlinear elasticity”. In: *International Journal for Numerical Methods in Engineering* 107.6 (2016), pp. 477–495.

[Rav00] S. S. Ravindran. “A reduced-order approach for optimal control of fluids using proper orthogonal decomposition”. In: *International journal for numerical methods in fluids* 34.5 (2000), pp. 425–448.

[Red04] J. N. Reddy. *Nonlinear finite element analysis*. New York: Oxford University Press, 2004.

[RS08] T. Reis and T. Stykel. "Balanced truncation model reduction of second-order systems". In: *Mathematical and Computer Modelling of Dynamical Systems* 14.5 (2008), pp. 391–406.

[RSB13] M. Rezaee, V. Shaterian-Alghalandis, and A. Banan-Nojavani. "Development of the smooth orthogonal decomposition method to derive the modal parameters of vehicle suspension system". In: *Journal of Sound and Vibration* 332.7 (2013), pp. 1829–1842.

[RL93] J. Ricles and P. Leger. "Use of load-dependent vectors for dynamic analysis of large space structures". In: *International Journal for Numerical Methods in Biomedical Engineering* 9.11 (1993), pp. 897–908.

[Rit09] W. Ritz. "Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik". In: *Journal für die reine und angewandte Mathematik* 135 (1909), pp. 1–61.

[Rix04] D. Rixen. "A Dual Craig-Bampton method for dynamic substructuring". In: *Journal of Computational and Applied Mathematics* 168.1-2 (2004), pp. 383–391.

[Rix01] D. Rixen. "Generalized mode acceleration methods and modal truncation augmentation". In: *Structures, Structural Dynamics and Material Conference and Exhibit, 42st AIAA/ASME/ASCE/AHS/ASC. AIAA*. Vol. 1300. 2001.

[Rub75] S. Rubin. "Improved component-mode representation for structural dynamic analysis". In: *AIAA Journal* 13.8 (1975), pp. 995–1006.

[RGR15] J. B. Rutzmoser, F. M. Gruber, and D. J. Rixen. "A Comparison on Model Order Reduction Techniques for Geometrically Nonlinear Systems Based on a Modal Derivative Approach Using Subspace Angles". In: *Proceedings of the 11th International Conference on Engineering Vibration*. Ljubljana, Slovenia, September 2015.

[RR17] J. B. Rutzmoser and D. J. Rixen. "A lean and efficient snapshot generation technique for the Hyper-Reduction of nonlinear structural dynamics". In: *Computer Methods in Applied Mechanics and Engineering* 325 (2017), pp. 330–349.

[RR14b] J. B. Rutzmoser and D. J. Rixen. "Model Order Reduction for Geometric Nonlinear Structures with Variable State-Dependent Basis". In: *Dynamics of Coupled Stuctures*. Ed. by M. Allen, R. Mayes, and D. Rixen. Vol. 1. Conference Proceedings of the Society for Experimental Mechanics Series. Proceedings of the 32nd IMAC, A Conference and Exposition on Structural Dynamics. 2014, pp. 455–462.

[RRT14] J. B. Rutzmoser, D. J. Rixen, and P. Tiso. "Model Order Reduction Using an Adaptive Basis for Geometrically Nonlinear Structural Dynamics". In: *Proceedings of the Conference on noise and vibration engineering (ISMA)*. Leuven, 2014.

[Rut+17] J. B. Rutzmoser, D. J. Rixen, P. Tiso, and S. Jain. "Generalization of quadratic manifolds for reduced order modeling of nonlinear structural dynamics". In: *Computers & Structures* 192 (2017), pp. 196–209.

[Ryc05] D. Ryckelynck. "A priori hyperreduction method: an adaptive approach". In: *Journal of Computational Physics* 202.1 (2005), pp. 346–366.

[SL06] B. Salimbahrami and B. Lohmann. "Order reduction of large scale second-order systems using Krylov subspace methods". In: *Linear Algebra and its Applications* 415.2-3 (2006), pp. 385–405.

[Sch+10] O. Schenk, K. Gärtner, G. Karypis, S. Röllin, and M. Hagemann. "PARDISO solver project". In: *URL: http://www. pardiso-project. org* (2010).

[Sch+01] O. Schenk, K. Gärtner, W. Fichtner, and A. Stricker. "PARDISO: a high-performance serial and parallel sparse linear solver in semiconductor device simulation". In: *Future Generation Computer Systems* 18.1 (2001), pp. 69–78.

[Sey93] A. P. Seyranian. "Sensitivity analysis of multiple eigenvalues". In: *Journal of Structural Mechanics* 21.2 (1993), pp. 261–284.

[SM96] Y. Shi and C. Mei. "A finite element time domain modal formulation for large amplitude free vibrations of beams and plates". In: *Journal of Sound and Vibration* 193.2 (1996), pp. 453–464.

[SH06] J. C. Simo and T. J. Hughes. *Computational inelasticity*. Vol. 7. Springer Science & Business Media, 2006.

[Sir87] L. Sirovich. "Turbulence and the dynamics of coherent structures. I. Coherent structures". In: *Quarterly of applied mathematics* 45.3 (1987), pp. 561–571.

[SdS95] P. Slaats, J. de Jongh, and A. Sauren. "Model reduction tools for nonlinear structural dynamics". In: *Computers & structures* 54.6 (1995), pp. 1155–1171.

[ST01] A. Steindl and H. Troger. "Methods for dimension reduction and their application in nonlinear dynamics". In: *International Journal of Solids and Structures* 38.10 (2001), pp. 2131–2147.

[Str07] G. Strang. *Computational science and engineering*. Wellesley-Cambridge Press Wellesley, 2007.

[SF73] G. Strang and G. J. Fix. *An analysis of the finite element method*. Vol. 212. Prentice-Hall Englewood Cliffs, NJ, 1973.

[Sug+06] H. Sugiyama, A. A. Shabana, M. A. Omar, and W.-Y. Loh. "Development of nonlinear elastic leaf spring model for multibody vehicle systems". In: *Computer methods in applied mechanics and engineering* 195.50 (2006), pp. 6925–6941.

[THD14] T. Tamarozzi, G. H. Heirman, and W. Desmet. "An on-line time dependent parametric model order reduction scheme with focus on dynamic stress recovery". In: *Computer Methods in Applied Mechanics and Engineering* 268 (2014), pp. 336–358.

[TDH03] J. P. Thomas, E. H. Dowell, and K. C. Hall. "Three-dimensional transonic aeroelasticity using proper orthogonal decomposition-based reduced-order models". In: *Journal of Aircraft* 40.3 (2003), pp. 544–551.

[Tis11] P. Tiso. "Optimal second order reduction basis selection for nonlinear transient analysis". In: *Modal Analysis Topics, Volume 3*. Springer, 2011, pp. 27–39.

[TJA11] P. Tiso, E. Jansen, and M. Abdalla. "Reduction method for finite element nonlinear dynamic analysis of shells". In: *AIAA Journal* 49.10 (2011), pp. 2295–2304.

[TR13] P. Tiso and D. J. Rixen. "Discrete empirical interpolation method for finite element structural dynamics". In: *Topics in Nonlinear Dynamics, Volume 1*. Springer, 2013, pp. 203–212.

[TR11] P. Tiso and D. J. Rixen. "Reduction methods for MEMS nonlinear dynamic analysis". In: *Nonlinear Modeling and Applications, Volume 2*. Springer, 2011, pp. 53–65.

[TM94] D. A. Tortorelli and P. Michaleris. "Design sensitivity analysis: overview and review". In: *Inverse problems in Engineering* 1.1 (1994), pp. 71–105.

[TWS05] M. A. Trindade, C. Wolter, and R. Sampaio. "Karhunen–Loeve decomposition of coupled axial/bending vibrations of beams subject to impacts". In: *Journal of sound and vibration* 279.3 (2005), pp. 1015–1036.

[Tyc+13] C. von Tycowicz, C. Schulz, H.-P. Seidel, and K. Hildebrandt. "An efficient construction of reduced deformable objects". In: *ACM Transactions on Graphics (TOG)* 32.6 (2013), p. 213.

[VD98] F. Van Keulen and H. De Boer. "Rigorous improvement of semi-analytical design sensitivities by exact differentiation of rigid body motions". In: *International journal for numerical methods in engineering* 42.1 (1998), pp. 71–91.

[VHK05] F. Van Keulen, R. Haftka, and N. Kim. "Review of options for structural design sensitivity analysis. Part 1: Linear systems". In: *Computer methods in applied mechanics and engineering* 194.30 (2005), pp. 3213–3243.

[VD93] P. Van Overschee and B. De Moor. "Subspace algorithms for the stochastic identification problem". In: *Automatica* 29.3 (1993), pp. 649–660.

[Vana] J. Vanderplaas. *Numba vs. Cython: Take 2*. URL: `https://jakevdp.github.io/blog/2013/06/15/numba-vs-cython-take-2`, visited on 04/21/2017.

[Vanb] J. Vanderplaas. *Optimizing Python in the Real World: NumPy, Numba, and the NUFFT*. URL: `https://jakevdp.github.io/blog/2015/02/24/optimizing-python-with-numpy-and-numba`, visited on 04/21/2017.

[Wan+09] X. Wang, M. Mignolet, T. Eason, and S. Spottswood. "Nonlinear reduced order modeling of curved beams: a comparison of methods". In: *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*

*17th AIAA/ASME/AHS Adaptive Structures Conference 11th AIAA*. 2009-2433. 2009.

[WN03] T. M. Wasfy and A. K. Noor. "Computational strategies for flexible multibody systems". In: *Applied Mechanics Reviews* 56.6 (2003), pp. 553–614.

[WWS14] O. Weeger, U. Wever, and B. Simeon. "Nonlinear frequency response analysis of structural vibrations". In: *Computational Mechanics* 54 (2014), pp. 1477–1495.

[WWS16] O. Weeger, U. Wever, and B. Simeon. "On the use of modal derivatives for nonlinear model order reduction". In: *International Journal for Numerical Methods in Engineering* 108.13 (2016), pp. 1579–1602.

[WT14] F. Wenneker and P. Tiso. "A substructuring method for geometrically nonlinear structures". In: *Dynamics of Coupled Structures, Volume 1*. Springer, 2014, pp. 157–165.

[Wil06] K. Willcox. "Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition". In: *Computers & fluids* 35.2 (2006), pp. 208–226.

[WP02] K. Willcox and J. Peraire. "Balanced model reduction via the proper orthogonal decomposition". In: *AIAA Journal* 40.11 (2002), pp. 2323–2330.

[WP14] W. Witteveen and F. Pichler. "Efficient Model Order Reduction for the Dynamics of Nonlinear Multilayer Sheet Structures with Trial Vector Derivatives". In: *Shock and Vibration* 2014 (2014).

[Wol14] T. Wolf. "$\mathcal{H}_2$ pseudo-optimal model order reduction". PhD thesis. Technische Universität München, 2014.

[Wri08] P. Wriggers. *Nonlinear finite element methods*. Springer Science & Business Media, 2008.

[WT16a] L. Wu and P. Tiso. "A Geometrically Nonlinear Craig-Bampton Method Using Quadratic Projection". In: *IMAC-XXXIV: International Modal Analysis Conference, Orlando, FL*. Society for Experimental Mechanics. Bethel, CT, January 2016.

[WT16b] L. Wu and P. Tiso. "Nonlinear model order reduction for flexible multibody dynamics: a modal derivatives approach". In: *Multibody System Dynamics* 36.4 (2016), pp. 405–425.

[XX15] D. Xie and M. Xu. "A comparison of numerical and semi-analytical proper orthogonal decomposition methods for a fluttering plate". In: *Nonlinear Dynamics* 79.3 (2015), pp. 1971–1989.

[XXD14] D. Xie, M. Xu, and E. H. Dowell. "Proper orthogonal decomposition reduced-order model for nonlinear aeroelastic oscillations". In: *AIAA Journal* 52.2 (2014), pp. 229–241.

[You11] M. I. Younis. *MEMS linear and nonlinear statics and dynamics*. Vol. 20. Springer Science & Business Media, 2011.

[ZJ07] L. ZhangPing and X. JinWu. "Novel modal method for efficient calculation of complex eigenvector derivatives". In: *AIAA Journal* 45.6 (2007), pp. 1406–1414.

[ZT13] O. C. Zienkiewicz and R. L. Taylor. *The finite element method for solid and structural mechanics*. Vol. 7. Elsevier Butterworth-Heinemann, 2013.

[ZTZ13] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method: Its basis & fundamentals*. Vol. 7. Elsevier Butterworth-Heinemann, 2013.

# List of Figures

# List of Tables

# Nomenclature

## Operators

$\dot{(\cdot)}$ . . . . Time derivative
$\ddot{(\cdot)}$ . . . . Second time derivative
$(\cdot)^T$ . . Transpose
$(\cdot)^{-1}$ . Inverse
$(\cdot)^+$ . . Moore-Penrose pseudo inverse
$|(\cdot)|$ . . Absolute value of vector / cardinality of a set

## Subscripts

$(\cdot)_{\text{approx}}$ . . . approximated
$(\cdot)_b$ . . . . . bending
$(\cdot)_e$ . . . . . element
$(\cdot)_{hr}$ . . . . . hyper-reduced
$(\cdot)_{int}$ . . . . . internal
$(\cdot)_{kin}$ . . . . . kinetic
$(\cdot)_{\text{kry}}$ . . . . . krylov
$(\cdot)_{\text{lin}}$ . . . . . linear
$(\cdot)_m$ . . . . . membrane
$(\cdot)_{nl}$ . . . . . nonlinear
$(\cdot)_r$ . . . . . reduced
$(\cdot)_{\text{raw}}$ . . . . . raw
$(\cdot)_{ref}$ . . . . . reference
$(\cdot)_{ss}$ . . . . . state-space
$(\cdot)_{\text{svd}}$ . . . . . singular value decomposition

## Greek Symbols

$\alpha$ . . . . . Time integration constant
$\beta$ . . . . . Time integration constant
$\gamma$ . . . . . Time integration constant
$\gamma_i$ . . . . Subspace angle
$\eta$ . . . . . Modal amplitude
$\lambda$ . . . . . Lamé constant
$\lambda_i$ . . . . $i$-th eigenvalue
$\mu$ . . . . . Lamé constant
$\nu$ . . . . . Poisson ratio
$\rho_\infty$ . . . . Spectral radius
$\Pi$ . . . . Potential of elastic system
$\sigma$ . . . . . Singular value
$\tau$ . . . . . Tolerance of sNNLS
$\xi_e$ . . . . Weight of element $e$
$\omega$ . . . . . Eigenfrequency
$\omega_0$ . . . . Shift frequency
$\Omega_0$ . . . Reference configuration domain
$\Omega_t$ . . . . Deformed configuration domain

## Boldface Greek Symbols

$\boldsymbol{\Gamma}$ . . . . . Nonlinear displacement function
$\boldsymbol{\theta}$ . . . . . Modal derivative
$\boldsymbol{\theta}_s$ . . . . Static derivative
$\boldsymbol{\Theta}$ . . . . Second order tensor of quadratic mapping
$\boldsymbol{\phi}$ . . . . . Vibration mode
$\boldsymbol{\Phi}$ . . . . Matrix of vibration modes
$\boldsymbol{\sigma}$ . . . . . Cauchy stress tensor
$\boldsymbol{\Sigma}$ . . . . . Singular value matrix
$\boldsymbol{\xi}$ . . . . . Vector of element weights
$\mathbf{Y}$ . . . . . Matrix of elemental force contributions
$\boldsymbol{\zeta}$ . . . . . Trial vector for element weights

## Latin Symbols

$E$ . . . . . Young's modulus
$g$ . . . . . External force amplitude
$h$ . . . . . Step width
$J$ . . . . . Determinant of deformation gradient
$m$ . . . . Number of force modes in force basis
$n$ . . . . . Number of dofs of the reduced system
$N$ . . . . Number of dofs of the unreduced system
$N_e$ . . . . Number of dofs of element $e$
$p$ . . . . . Parameter
$rhs$ . . . Right hand side
$RE$ . . . Relative error
$RE_f$ . . Relative full error
$RE_{hr}$ . . Relative hyper-reduction error
$RE_M$ . Relative error in $M$-norm
$RE_{rb}$ . . Relative reduced basis error
$s$ . . . . . Laplace transform of time $t$
$t$ . . . . . Time
$t_w$ . . . . Wall time
$W$ . . . . Energy density function

## Boldface Latin Symbols

$\boldsymbol{A}_{ss}$ . . . State-space system matrix
$\boldsymbol{b}$ . . . . . Vector of reduced forces of training snapshots
$\boldsymbol{B}_{ss}$ . . . State-space input matrix
$\boldsymbol{B}_A$ . . . Assembly matrix
$\boldsymbol{c}$ . . . . . Force amplitudes
$\boldsymbol{C}$ . . . . . Damping matrix
$\boldsymbol{C}_{SE}$ . . . Tangent modulus
$\boldsymbol{C}_{ss}$ . . . State-space output matrix
$\boldsymbol{D}$ . . . . Finite difference matrix
$\boldsymbol{e}$ . . . . . Unit vector
$\boldsymbol{E}$ . . . . . Green-Lagrange strain tensor
$\boldsymbol{f}$ . . . . . Internal force vector
$\boldsymbol{f}_u$ . . . . Unassembled internal force vector
$\boldsymbol{F}$ . . . . . Deformation gradient
$\boldsymbol{g}$ . . . . . External force vector
$\bar{\boldsymbol{g}}$ . . . . . External force shape vector
$\hat{\boldsymbol{g}}$ . . . . . External force amplitude vector
$\boldsymbol{G}$ . . . . Matrix of force input locations
$\boldsymbol{H}$ . . . . Transfer function
$\boldsymbol{I}$ . . . . . Identity matrix
$\boldsymbol{K}$ . . . . . Stiffness matrix
$\boldsymbol{k}^{(2)}$ . . . Vector of quadratic stiffness coefficients
$\boldsymbol{k}^{(3)}$ . . . Vector of cubic stiffness coefficients
$\boldsymbol{K}^{(1)}$ . . Reduced linear stiffness tensor
$\boldsymbol{K}^{(2)}$ . . Reduced quadratic stiffness tensor
$\boldsymbol{K}^{(3)}$ . . Reduced cubic stiffness tensor
$\boldsymbol{K}_u$ . . . Unassembled stiffness matrix
$\boldsymbol{L}$ . . . . . Cholesky factorization matrix
$\boldsymbol{L}_e$ . . . . Boolean localization matrix for element $e$
$\boldsymbol{M}$ . . . . Mass matrix
$\boldsymbol{n}$ . . . . . Normal vector
$\boldsymbol{N}$ . . . . Vector of shape functions
$\boldsymbol{P}$ . . . . . Boolean collocation matrix
$\boldsymbol{P}_\Gamma$ . . . . Tangent projector of transformation $\boldsymbol{\Gamma}$
$\boldsymbol{q}$ . . . . . Reduced generalized coordinates
$\boldsymbol{r}$ . . . . . Residual
$\boldsymbol{R}$ . . . . . Rotation tensor
$\boldsymbol{S}$ . . . . . Second Piola-Kirchhoff stress tensor
$\boldsymbol{S}_d$ . . . . Displacement snapshot matrix
$\boldsymbol{S}_v$ . . . . Velocity snapshot matrix
$\boldsymbol{t}$ . . . . . Traction vector
$\boldsymbol{T}_i$ . . . . $i$-th moment
$\boldsymbol{u}$ . . . . . Nodal displacement vector
$\boldsymbol{u}_e$ . . . . Nodal displacement vector of element $e$
$\bar{\boldsymbol{u}}$ . . . . . Continuous displacement field
$\hat{\boldsymbol{u}}_e$ . . . . Nodal displacement matrix of element $e$
$\boldsymbol{u}_{\text{svd}}$ . . Left singular vector
$\boldsymbol{U}$ . . . . Stretch tensor
$\boldsymbol{U}_f$ . . . Force basis
$\boldsymbol{U}_{\text{svd}}$ . . Matrix of left singular vectors
$\boldsymbol{v}$ . . . . . Vector of reduced basis $\boldsymbol{V}$
$\boldsymbol{V}$ . . . . . Reduced basis
$\boldsymbol{V}_{ss}$ . . . Right state-space projection matrix
$\boldsymbol{V}_{\text{svd}}$ . . Matrix of right singular vectors
$\boldsymbol{w}$ . . . . Transversal displacements
$\boldsymbol{W}$ . . . . Weighting matrix
$\boldsymbol{W}_{ss}$ . . Left state-space projection matrix
$\boldsymbol{x}$ . . . . . State-space vector
$\bar{\boldsymbol{x}}$ . . . . . Continuous coordinates in deformed configuration
$\bar{\boldsymbol{X}}$ . . . . Continuous coordinates in reference configuration
$\boldsymbol{y}_{ss}$ . . . . State-space output
$\boldsymbol{z}$ . . . . . Nonlinear reduced generalized coordinates

## Calligraphic Symbols

$\mathcal{T}$ . . . . Set of time steps
$\mathcal{E}$ . . . . . Element set
$\widetilde{\mathcal{E}}$ . . . . . Reduced element set
$\mathcal{I}$ . . . . . Set of indices
$\mathcal{N}$ . . . . Gaussian distributed random number
$\wp$ . . . . . Set of collocation dofs

## Abbreviations

DEIM . . . . Discrete Empirical Interpolation Method
dof . . . . . degree of freedom
ECSW . . . Energy Conserving Mesh Sampling and Weighting
EIM . . . . . Empirical Interpolation Method
FW . . . . . Frequency Weighting
GB . . . . . Giga Byte
GNAT . . . Gauss-Newton with Approximated Tensors
HHT-$\alpha$ . . . Hilber Hughes Taylor-$\alpha$
IC . . . . . Implicit Condensation
ICE . . . . . Implicit Condensation and Expansion
IRKA . . . . Iterative Rational Krylov Algorithm
KrySD . . . Krylov Static Derivative
LB . . . . . Linear Basis
LLVM . . . Low-Level Virtual Machine
MAC . . . . Modal Assurance Criterion
MB . . . . . Mega Byte
MD . . . . . Modal Derivative
MKL . . . . . Math Kernel Library
MMI . . . . . Maximum Modal Interaction
MTA . . . . . Modal Truncation Augmentation
MVW . . . . Modal Virtual Work
NP . . . . . Non-Polynomial
NSKTS . . Nonlinear Stochastic Krylov Training Sets
NSMTS . . Nonlinear Stochastic Modal Training Sets
NVH . . . . Noise Vibration Harshness
ODE . . . . . Ordinary Differential Equation
PCA . . . . . Principal Component Analysis
PDE . . . . . Partial Differential Equation
PGD . . . . . Proper Generalized Decomposition
POD . . . . . Proper Orthogonal Decomposition
QM . . . . . Quadratic Manifold
RAM . . . . Random Access Memory
RE . . . . . Relative Error
SD . . . . . Static Derivative
sNNLS . . sparse Non-Negative Least Square
SMD . . . . . Static Modal Derivative
SMVW . . Symmetric Modal Virtual Work
SOD . . . . . Smooth Orthogonal Decomposition
SOM . . . . . Smooth Orthogonal Mode
SVD . . . . . Singular Value Decomposition
UDEIM . . Unassembled Discrete Empirical Interpolation Method

# Appendices

# Appendix A

# AMFE: The Finite Element Toolbox for Python with Simplicity in Mind

All numerical experiments in this thesis are performed with the finite element framework AMFE, which has been developed in the context of this thesis. In comparison to many other finite element frameworks, the goal of AMFE is to provide a tool for the rapid prototyping of algorithms in an interpreted environment. The goals determining the design are given in the following list:

- Comprehensiveness: The code should be able to handle regular finite element problems as they are common in industry with a large element library.
- Access to all internal routines: In nonlinear model order reduction, especially in hyper-reduction, internal routines of the computational process have to be accessed and modified.
- Interactivity: The data types appearing in a simulation should be available in an interactive interface allowing for quick investigations, analyses and debugging.
- Modularity and Extensibility: The code should be organized such, that the dependency of different routines is set to a minimum. The interactions of the routines should be clearly specified.
- Speed: The code should have execution times similar to commercial packages in order to make runtime comparisons reasonable.
- Simplicity: The code should exhibit a clear structure and be simple to read and understand.

The AMFE toolbox is developed in a way to achieve the best possible balance of the often contradictory requirements. It is mostly written in Python with some of the time-critical parts implemented in Fortran in order to achieve comparable speeds to commercial software packages.

## A.1 Design of the Code

A finite element simulation is typically split in three parts: pre-processing, solving and post-processing. The AMFE code is a pure solver only capable of the intermediate solution step. Hence, it relies on preprocessing tools for the mesh generation and post-processing tools for the analysis of the results.

In Figure A.1, the general workflow of a simulation using AMFE is depicted. In the preprocessing step, the mesh of the spatial domain is generated. A mesh is defined by

its nodes and the elements composed of nodes. Furthermore, areas for the application of boundary conditions are defined in this preprocessing step, too. Typically, the domain is also subdivided in different parts, for instance when different materials are present in the body. These information are stored in a mesh file in a specified manner and form the output of the preprocessing step.

In many finite element programs, the mesh information and the instruction for the solver are cast together in the so-called input file (cf. [HKS01; McC72]). In AMFE, the design decision was made against that concept. The concept of AMFE is that of a toolbox embedded in the scripting language Python, so that the scripted instructions are specified in the same ecosystem as the finite element core functionality is. Consequently, the mesh file which can be created with *gmsh* [GR09], *Abaqus* [HKS01] or *Nastran* [McC72] contains solely the geometry. All other specifications like the material properties, the type and parameters of Dirichlet and Neumann boundary conditions and the instructions for the solution procedure are all given in the Python script using the functionality of the AMFE module.

The output of the computation using the AMFE toolbox can be stored for graphical postprocessing according to the *xdmf standard* [CN01]. This standard proposes a separation of so-called *heavy data* and *light data*. The heavy data are the mesh information and the field variables like displacements, stresses or strains. They are stored using the hdf5 file specification [Fol+11], which allows for storing the heavy data in a hierarchical manner. The light data, i.e., the complementary metadata of the heavy data like element types, mesh groups etc., are stored in an xdmf-file using the xml-standard. Consequently, the resulting data can be post-processed both in tools with an xdmf-interface like the open-source ParaView as well as further analysis environments like Python by using the `.hdf5` file.

As pointed out above, the AMFE framework is a Python library solely addressing the finite element computations. It is organized in a modular fashion, as depicted in Figure A.2. The main class used for setting up the finite element problem in the python script file is the `MechanicalSystem` class, which covers the whole finite element system containing the Mesh with its element lists and the material, the Assembly class and the Boundary class.

The MechanicalSystem serves as the interface class interacting with routines of the Solver module or other routines part of the Tools or the ModelReduction module like modal analysis, Krylov subspace reduction, NSKTS or NSMTS computation or the computation of (S)MDs. Consequently, the finite element routines are hidden behind this class, which can be modified to represent other systems. Hence, the ReducedSystem is a sub-class of MechanicalSystem realized with the object-oriented core principle *inheritance* [Lut10]. The HyperReducedSystem is then a sub-class of the ReducedSystem, as depicted in Figure A.2 realizing the accelerated evaluation of the nonlinearity.

In the following, the different modules of the AMFE toolbox shown in Figure A.2 are briefly explained:

**Material** The material class represents the constitutive law of any hyperelastic material. It provides the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$ and the tangent modulus $\boldsymbol{C}^{SE}$ for 2D and 3D problems for a given Green-Lagrange strain $\boldsymbol{E}$. The Material class is commonly a member of the Element class, so that the Element performs a callback on the constitutive law to compute the stress at the Gauss points. Since the evaluation of a nonlinear material is performed extremely often in a nonlinear finite element procedure, the routines are implemented in Fortran for high execution speed.

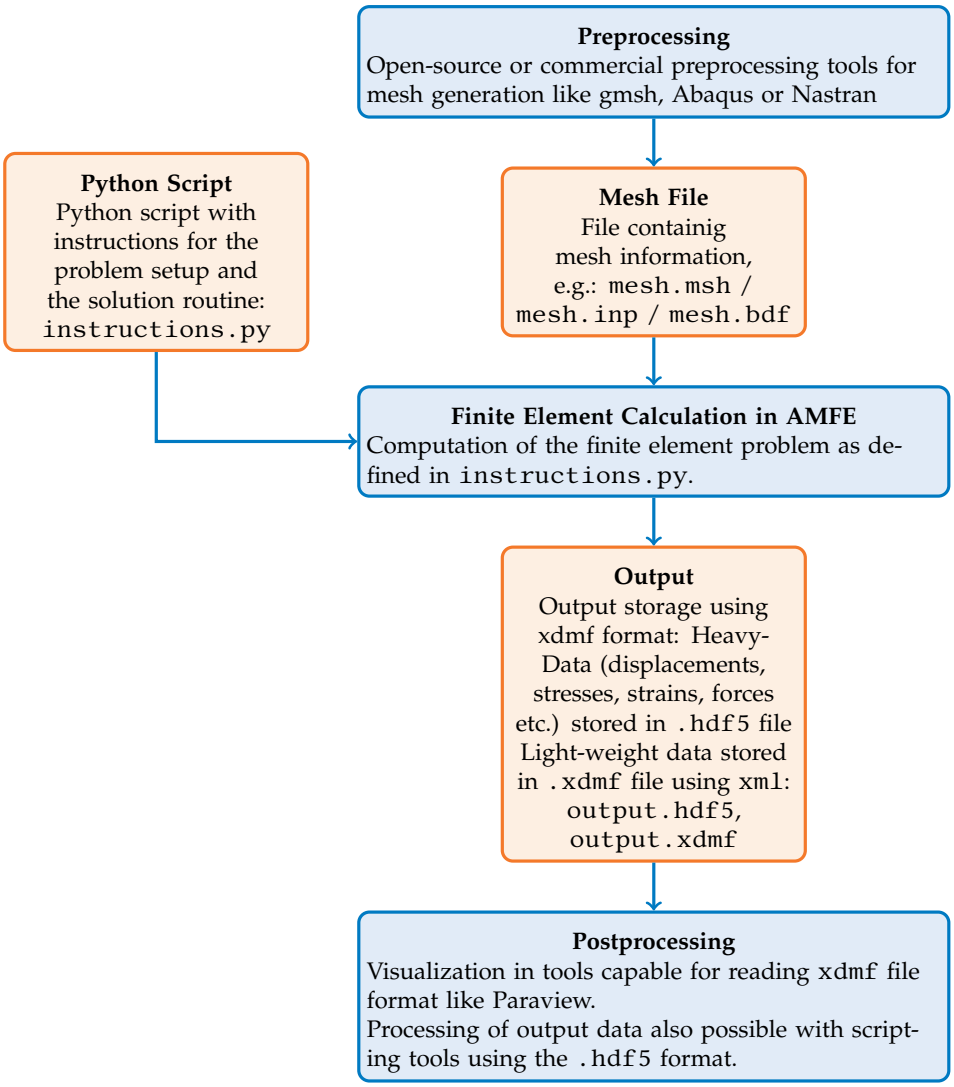


**Figure A.1:** Structure of the general workflow with AMFE.

**Element** The Element class represents a finite element of the system. It is a super-class providing the interface for the element internal forces $\boldsymbol{f}_e(\boldsymbol{u}_e)$, the tangential stiffness matrix $\boldsymbol{K}_e(\boldsymbol{u}_e)$ and the mass matrix $\boldsymbol{M}_e$. The different element types like Tri3, Tri6, Quad4, Quad8, Tet4 and Tet10 or Hex8 and Hex20 are realized as sub-class of the super-class Element providing a standardized interface. For the evaluation of the constitutive law, every Element class has a shallow copy of the Material class. Since the elemental quantities are evaluated very often in the finite element procedure, the methods for computing $\boldsymbol{f}_e$, $\boldsymbol{K}_e$ and $\boldsymbol{M}_e$ are implemented in Fortran for high execution speed.

In AMFE, the handling of Neumann boundary conditions, i.e., external loads, is realized using so-called *skin elements*. These elements, which are also realized as sub-classes of the Element super-class, describe the boundary of the domain and provide the traction or pressure forces. They are assembled in the same routine as the internal forces to the global external force vector using a method provided in the Assembly class. With this concept it is also possible to realized more sophisticated external forces like contact forces.

**Mesh** The Mesh class handles the import and export of meshes. Thereby it has different import methods for the different file types like .msh from gmsh, .inp from Abaqus or

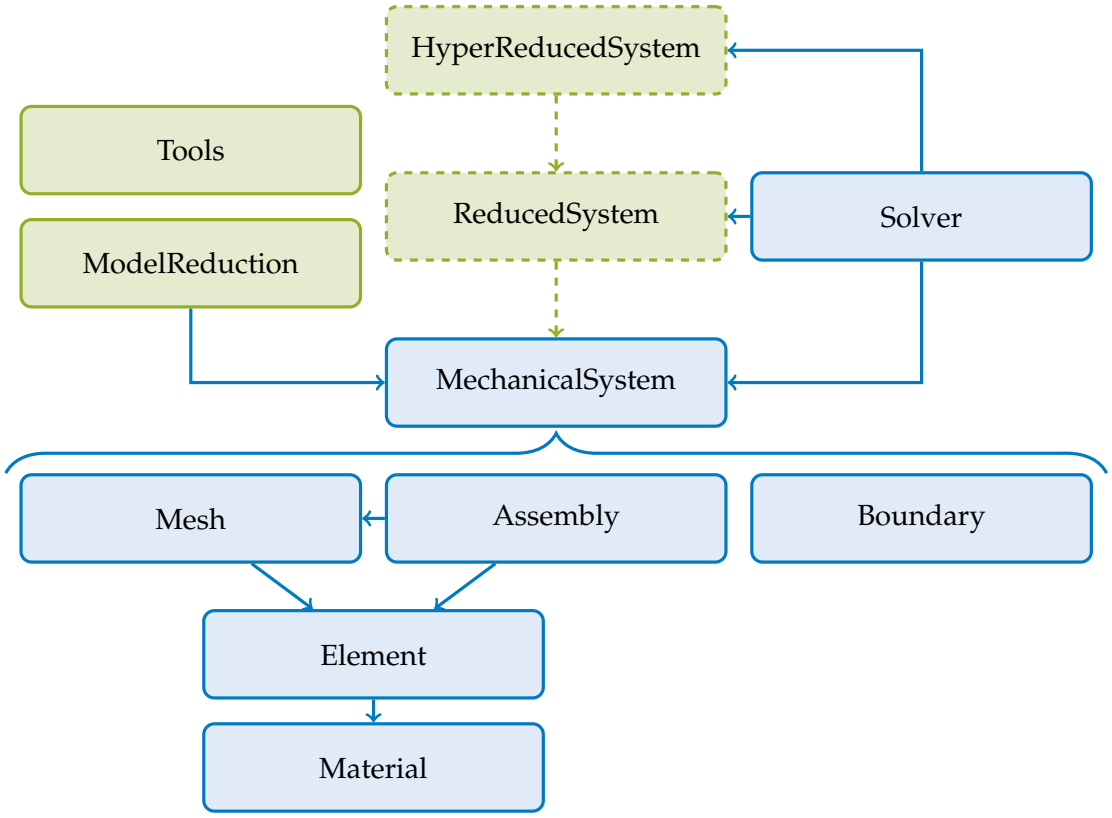


**Figure A.2:** Structure of the AMFE toolbox.

.bdf from Nastran. It also handles the export to the .hdf5 files for the heavy data and the .xdmf files for the light data. It has furthermore methods for deflating a mesh, i.e., removing nodes which are not associated with an element as well as methods for selecting the elements and dofs associated to different mesh domains. For the database-like operations for selecting different mesh properties, it operates on a Pandas Dataframe object (cf. [McK12]) representing the mesh, allowing for flexibly selecting elements belonging to different physical groups or geometric domains. The Mesh class contains a list of the nodal coordinates and a list of all element nodes. The list of all elements is realized as a list of shallow element copies, i.e, pointers to a set of unique elements. For assembly, this list is provided to the Assembly class where the desired method of every element of this list is called.

**Assembly** The assembly class organizes the assembly of the elemental contributions to the global vectors and matrices. Thereby, it has efficient methods implemented in Fortran to perform the assembly to the global, sparse matrices. Furthermore, this class has a method for preallocation, i.e., for the determination of the sparsity pattern of the system matrices. Additionally, this class has methods for the assembly of reduced systems and hyper-reduced systems.

**Boundary** The boundary class organizes the application of linear Dirichlet boundary conditions. This is performed with a sparse matrix $\boldsymbol{B}$, which maps the dofs of the constrained system $\boldsymbol{u}_{constr}$ to the dofs of the unconstrained system $\boldsymbol{u}_{unconstr}$ as

$$\boldsymbol{u}_{unconstr} = \boldsymbol{B}\boldsymbol{u}_{constr}. \tag{A.1}$$

The unconstrained domain is the domain, in which the quantities are assembled. This

has the advantage, that the assembly is independent of the constraints of the system simplifying these routines. The assembled mass and stiffness matrices in the unconstrained domain can be transformed to the constrained domain by

$$\boldsymbol{M}_{constr} = \boldsymbol{B}^T \boldsymbol{M}_{unconstr} \boldsymbol{B} \quad \boldsymbol{K}_{constr} = \boldsymbol{B}^T \boldsymbol{K}_{unconstr} \boldsymbol{B}. \tag{A.2}$$

Similarly, the internal force vector is constrained by

$$\boldsymbol{f}_{constr} = \boldsymbol{B}^T \boldsymbol{f}_{unconstr}. \tag{A.3}$$

The generation of a constrained vector from the unconstrained displacement vector, for instance, is performed with

$$\boldsymbol{u}_{constr} = (\boldsymbol{B}^T \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{u}_{unconstr}. \tag{A.4}$$

When the matrix $\boldsymbol{B}$ is orthogonal, as it is the case when the boundary conditions involve only the fixation of certain dofs, the term $(\boldsymbol{B}^T \boldsymbol{B})^{-1} = \boldsymbol{I}$ and can hence be omitted in (A.4).

The realization of the boundary conditions via a projection matrix $\boldsymbol{B}$ comes with the drawback, that the application of constraints involves the multiplication of sparse matrices and hence computational costs. In the numerical profiling experiments, the impact on the computational time was so low that this procedure was chosen instead of the commonly used elimination of row and column entries. The elimination method has the drawback that many operations like the computation of eigenvalues and eigenvectors involve special handling of the blocked dofs. For AMFE, the design decision was made that the dofs provided by the MechanicalSystem class are equal to the dofs of the underlying mechanical system.

**MechanicalSystem** The MechanicalSystem class is the abstraction class representing the full finite element model. It organizes the interaction of the mesh class, the Assembly class and the Boundary class, so that these objects are hidden behind the MechanicalSystem class. As illustrated in Listing A.1, the common setup operations are performed directly with methods of the MechanicalSystem class not requiring method calls of the mesh class contained in the MechanicalSystem.

The MechanicalSystem class offers the interface methods for static or dynamic solution schemes like providing the internal or external forces, the tangential stiffness matrix, the mass matrix and the storing of displacement fields. Furthermore, it offers the interface for building up the finite element system like loading a mesh, applying Dirichlet boundary conditions or applying Neumann Boundary conditions. Moreover, it enables the export of the mesh with the saved displacement fields and possibly other field variables to ParaView.

As pointed out in the previous paragraph, the MechanicalSystem provides the finite element system in a minimal coordinate style, i.e., all dofs in the internal force vector for instance represent a real dof of the underlying mechanical system. Consequently, the MechanicalSystem class is not bound to finite element Systems, but could also represent a multibody system or a constrained system, where the additional dofs represent Lagrange multipliers.

**Solver** The solver module contains functions for the efficient solution of problems by interacting with a MechanicalSystem object. It has a binding to the Intel MKL solver Pardiso [Sch+10] allowing for the efficient solution of sparse systems. It provides functions for linear and nonlinear static solutions as well as linear and nonlinear time integration schemes.

**Tools** The tools module provides functions necessary for software development like unit test routines and functions necessary for scientific investigations like hdf5 file input and output, error computation or numerical methods like the sNNLS solver. It is not associated with a particular class.

**ModelReduction** The ModelReduction module is a collection of functions providing linear and nonlinear reduced basis methods. Examples are routines for generating a modal basis, a Krylov subspace, a Craig-Bampton basis, modal derivatives, static derivatives or POD modes. The module also provides routines to compute training sets like the NSKTS or the NSMTS. Furthermore, it has functions to convert a MechanicalSystem to a ReducedSystem or a HyperReducedSystem.

**ReducedSystem** The ReducedSystem class is an inherited sub-class of the MechanicalSystem class. It provides the same interface like the super-class, however returns the reduced variables of a reduced basis system in contrary to the full system. It can be built using a function from the ModelReduction module as shown in Listing A.2.

**HyperReducedSystem** The class HyperReducedSystem is a sub-class of ReducedSystem. It has the same interface as both super-classes. In contrary to the ReducedSystem it provides the reduced nonlinear forces bases on the selected hyper-reduction scheme.

To demonstrate the workflow of a finite element simulation in AMFE, Listing A.1 depicts an example of a 2D finite element setup realized with AMFE. After the loading of the `amfe` module, the finite element system is set up by interacting with the MechanicalSystem object `my_system`. Different computations can be executed in an interactive fashion like the solution of a linear static problem, a nonlinear static problem or the computation of the eigenmodes of the system. Note that the displacement fields are saved in the `my_system` object, so that the export to ParaView exports the results of the previous computation.

Listing A.2 shows a more advanced setup, where a hyper-reduced model is built. After the definition of a mechanical problem in line 5 - 11, a simulation-free basis is computed using vibration modes and static derivatives (lines 13-17). This basis is then used for the computation of NSKTS, which are used as training sets for the ECSW hyper-reduction. Finally, a time integration is performed and the results of the time integration are exported.

In Listing A.2, the concept of AMFE is illustrated. The computations not directly associated with finite element systems can be performed using the numerical toolbox `numpy` of Python like the definition of a time function for the excitation in line 11 or the generation of time steps in line 33. Since the input is interpreted and does not require compilation, workflows like the ones in Listing A.2 or Listing A.2 can be developed interactively, which can be a huge benefit in practice.

**Listing A.1:** Example of a Python input file solving a planar finite element problem using AMFE. The mesh is generated with gmsh.

```
1 import amfe
2
3 my_material = amfe.KirchhoffMaterial(E=210E9, nu=0.3, rho=1E4,
4                                      plane_stress=True)
5 my_system = amfe.MechanicalSystem()
6 # load a mesh and assign the material to domain number 7
7 my_system.load_mesh_from_gmsh('mesh_file.msh', 7, my_material)
8 # fix x and y coordinate of domain number 8
9 my_system.apply_dirichlet_boundaries(8, 'xy')
```

```
# apply a load on domain number 9
my_system.apply_neumann_boundaries(key=9, val=1E8, direct=(0,-1),
                                   time_func=lambda t: t)

# Perform a linear static solution
u_lin = amfe.solve_linear_displacement(my_system)
# Export displacement to ParaView
my_system.export_paraview('displacement_linear')

# Perform a nonlinear static solution
u_nonlin = amfe.solve_nonlinear_displacement(my_system, no_of_load_steps=50)
# Export displacement steps to ParaView
my_system.export_paraview('displacement_nonlinear')

# Perform a modal analysis
omega, V = amfe.vibration_modes(my_system, save=True)
# Export mode shapes to ParaView
my_system.export_paraview('mode_shapes')
```

**Listing A.2:** Example of a Python input file performing a lean hyper-reduction using a reduced basis composed of vibration modes and SMDs and NSKTS for the hyper-reduction.

```
import amfe
# Load the numerical library numpy
import numpy as np

# define a mechanical Problem
my_material = amfe.KirchhoffMaterial(E=210E9, nu=0.3, rho=1E4)
my_system = amfe.MechanicalSystem()
my_system.load_mesh_from_gmsh('mesh_file.msh', 7, my_material)
my_system.apply_dirichlet_boundaries(8, 'xy')
my_system.apply_neumann_boundaries(key=9, val=1E8, direct=(0,-1),
                                   time_func=lambda t: np.sin(35*np.pi*t))

# compute a simulation-free basis using modes + SDs
omega, V = amfe.vibration_modes(my_system, n=10)
Theta = amfe.static_correction_theta(V, K_func=my_system.K, h=1.0,
                                     finite_diff='central')
Q = amfe.linear_qm_basis(V, Theta, tol=1E-8, symm=True)

# compute NSKTS using a reduced system
my_reduced_system = amfe.reduce_mechanical_system(my_system, Q)
nskts = amfe.compute_nskts(my_reduced_system, no_of_moments=4,
                           no_of_static_cases=8, load_factor=2,
                           no_of_force_increments=20)
# build hyper-reduced system
my_hyper_system = amfe.hyper_reduce_mechanical_system(my_system, Q)
# perform mesh reduction using ECSW
my_hyper_system.reduce_mesh(nskts, tau=0.001)

# perform a time integration from 0 to 2s with dt=0.001
q0 = np.zeros(shape=Q.shape[1])
dq0 = np.zeros_like(q0)
dt = 0.001
T = np.arange(0, 2, dt)
amfe.integrate_nonlinear_gen_alpha(my_system, q0, dq0, T, dt, rho_inf=0.8)
my_system.export_paraview('time_stepping_output')
```

## A.2 Nonlinear Finite Element Formulation

The finite element formulation, as pointed out in Chapter 2, is about the computation of the matrix and vector quantities of the finite element. They are the internal forces $\boldsymbol{f}_e(\boldsymbol{u}_e)$, the tangential stiffness matrix $\boldsymbol{K}_e$ equal to the Jacobian $\partial \boldsymbol{f}_e / \partial \boldsymbol{u}_e$ of the nonlinear forces and the mass matrix $\boldsymbol{M}_e$. The tangential stiffness matrix is needed for linear and nonlinear static solution techniques, for linear dynamical systems and for nonlinear implicit time integration schemes. In the following, the Total Lagrangian nonlinear finite element procedure is spelled out and explained on the two dimensional four node quadrilateral element. Since this element is a two dimensional element, the number of dimensions is $ndim = 2$. The number of nodes is $n_n = 4$. The formulation presented here is also the formulation used in the AMFE code framework developed by the author of this thesis.

Starting with the kinematics, the approximation of all field variables is expressed in terms of the nodal coordinates and the shape functions, see also (2.21) to (2.23):

$$\bar{\boldsymbol{X}}(\boldsymbol{\xi}) = \sum_{i=1}^{n_e} N_i(\boldsymbol{\xi}) \boldsymbol{X}_{e,i} = \hat{\boldsymbol{X}}_e^T \boldsymbol{N}(\boldsymbol{\xi}), \tag{A.5}$$

$$\bar{\boldsymbol{x}}(\boldsymbol{\xi}) = \sum_{i=1}^{n_e} N_i(\boldsymbol{\xi}) \boldsymbol{x}_{e,i} = \hat{\boldsymbol{x}}_e^T \boldsymbol{N}(\boldsymbol{\xi}), \tag{A.6}$$

$$\bar{\boldsymbol{u}}(\boldsymbol{\xi}) = \sum_{i=1}^{n_e} N_i(\boldsymbol{\xi}) \boldsymbol{u}_{e,i} = \hat{\boldsymbol{u}}_e^T \boldsymbol{N}(\boldsymbol{\xi}). \tag{A.7}$$

The nodal quantities $\hat{\boldsymbol{X}}_e \in \mathbb{R}^{n_n \times ndim}$, $\hat{\boldsymbol{x}}_e \in \mathbb{R}^{n_n \times ndim}$ and $\hat{\boldsymbol{u}}_e \in \mathbb{R}^{n_n \times ndim}$ denoted with a hat are quantities arranged in matrix form, where the rows stand for the nodes and the columns form the $x$-$y$ and possibly $z$-components. Nodal quantities without a hat are arranged as a full column vector, as shown in (A.19). The continuous quantities are denoted with a bar, where also nodal quantities exist, i.e., $\bar{\boldsymbol{X}}_e$, $\bar{\boldsymbol{x}}_e$ and $\bar{\boldsymbol{u}}_e$. For a four node Quadrilateral element, for instance, the nodal coordinates $\hat{\boldsymbol{X}}_e \in \mathbb{R}^{4\times 3}$ and displacements $\hat{\boldsymbol{u}}_e \in \mathbb{R}^{4\times 3}$ are given as

$$\hat{\boldsymbol{X}}_e = \begin{pmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ X_3 & Y_3 & Z_3 \\ X_4 & Y_4 & Z_4 \end{pmatrix} \quad \hat{\boldsymbol{u}}_e = \begin{pmatrix} u_{x1} & u_{y1} & u_{z1} \\ u_{x2} & u_{y2} & u_{z2} \\ u_{x3} & u_{y3} & u_{z3} \\ u_{x4} & u_{y4} & u_{z4} \end{pmatrix}. \tag{A.8}$$

The shape functions $\boldsymbol{N} \in \mathbb{R}^{n_n}$ for the four node quadrilateral element are given as

$$\boldsymbol{N}(\boldsymbol{\xi}) = \begin{pmatrix} N_1(\xi,\eta) \\ N_2(\xi,\eta) \\ N_3(\xi,\eta) \\ N_4(\xi,\eta) \end{pmatrix} = \begin{pmatrix} \frac{1}{4}(1-\xi)(1-\eta) \\ \frac{1}{4}(1+\xi)(1-\eta) \\ \frac{1}{4}(1+\xi)(1+\eta) \\ \frac{1}{4}(1-\xi)(1+\eta) \end{pmatrix}. \tag{A.9}$$

The spatial derivatives of the field variables are performed on the shape functions. Thereby, the derivatives are always computed with respect to the element coordinate system $\boldsymbol{\xi}$ allowing for an analytical computation. The derivatives with respect to the reference coordinates $\bar{\boldsymbol{X}}$, as typical in the Total Lagrangian framework, are carried out using a detour over the element Jacobian $\boldsymbol{J} = \partial \bar{\boldsymbol{X}} / \partial \boldsymbol{\xi} \in \mathbb{R}^{ndim \times ndim}$, which is computed with the derivative of the shape functions $\boldsymbol{N}$ with respect to the element coordinates $\boldsymbol{\xi}$:

$$\boldsymbol{J} = \partial \bar{\boldsymbol{X}} / \partial \boldsymbol{\xi} = \hat{\boldsymbol{X}}_e^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}}. \tag{A.10}$$

For the Quad4 element, for instance, the derivative of the shape functions $\boldsymbol{N}$ with respect to the element coordinates $\boldsymbol{\xi} = (\xi, \eta)^T$ is written as

$$\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} = \begin{pmatrix} \frac{1}{4}(\eta - 1) & \frac{1}{4}(\xi - 1) \\ \frac{1}{4}(-\eta + 1) & \frac{1}{4}(-\xi - 1) \\ \frac{1}{4}(\eta + 1) & \frac{1}{4}(\xi + 1) \\ \frac{1}{4}(-\eta - 1) & \frac{1}{4}(-\xi + 1) \end{pmatrix}. \tag{A.11}$$

The spatial derivatives are then obtained using the inverse of the element Jacobian. The deformation gradient $\boldsymbol{F} \in \mathbb{R}^{ndim \times ndim}$ for instance, is written as

$$\boldsymbol{F} = \frac{\partial \bar{\boldsymbol{x}}}{\partial \bar{\boldsymbol{X}}} = \boldsymbol{I} + \frac{\partial \bar{\boldsymbol{u}}}{\partial \bar{\boldsymbol{X}}} = \boldsymbol{I} + \frac{\partial \bar{\boldsymbol{u}}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \bar{\boldsymbol{X}}} = \boldsymbol{I} + \hat{\boldsymbol{u}}_e^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} \left( \hat{\boldsymbol{X}}_e \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} \right)^{-1}. \tag{A.12}$$

The spatial derivative with respect to the reference configuration can be concatenated yielding the spatial derivative operator $\widetilde{\boldsymbol{B}}_0 \in \mathbb{R}^{n_n \times ndim}$:

$$\frac{\partial \bar{\boldsymbol{u}}}{\partial \bar{\boldsymbol{X}}} = \hat{\boldsymbol{u}}_e^T \widetilde{\boldsymbol{B}}_0 \quad \text{with} \quad \widetilde{\boldsymbol{B}}_0 = \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} \left( \hat{\boldsymbol{X}}_e \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\xi}} \right)^{-1}. \tag{A.13}$$

The Green-Lagrange strain tensor $\boldsymbol{E} \in \mathbb{R}^{ndim \times ndim}$ is computed using the auxiliary matrix $\boldsymbol{H} = \partial \bar{\boldsymbol{u}} / \partial \bar{\boldsymbol{X}}$. In comparison to the direct computation with the deformation gradient, the following procedure is numerically more stable [BLM00]:

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{H} + \boldsymbol{H}^T + \boldsymbol{H}^T \boldsymbol{H}) \quad \text{with} \quad \boldsymbol{H} = \hat{\boldsymbol{u}}_e^T \widetilde{\boldsymbol{B}}_0. \tag{A.14}$$

The next step is the evaluation of the internal virtual work (2.27). There, the inner product of the two tensors $\boldsymbol{S}$ and $\boldsymbol{E}$ has to be computed. Hence, the so-called *Voigt notation* has been established to describe the equivalent operations with one dimensional vectors. The Voigt representation $\boldsymbol{S}^v$ of $\boldsymbol{S}$ and $\boldsymbol{E}^v$ of $\boldsymbol{E}$ is written as

$$\boldsymbol{S}^v = (S_{11}, S_{22}, S_{33}, S_{23}, S_{13}, S_{12})^T \quad \text{with} \quad \boldsymbol{S} = \begin{pmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{pmatrix} \tag{A.15}$$

$$\boldsymbol{E}^v = (E_{11}, E_{22}, E_{33}, 2E_{23}, 2E_{13}, 2E_{12})^T \quad \text{with} \quad \boldsymbol{E} = \begin{pmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{pmatrix} \tag{A.16}$$

for three dimensional problems and as

$$\boldsymbol{S}^v = (S_{11}, S_{22}, S_{12})^T \quad \text{with} \quad \boldsymbol{S} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \tag{A.17}$$

$$\boldsymbol{E}^v = (E_{11}, E_{22}, 2E_{12})^T \quad \text{with} \quad \boldsymbol{E} = \begin{pmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{pmatrix} \tag{A.18}$$

for two dimensional problems. Also the nodal values are written as vectors, e.g.,

$$\boldsymbol{u}_e = (u_{x1}, u_{y1}, u_{z1}, u_{x2}, u_{y2}, u_{z2}, u_{x3}, u_{y3}, u_{z3}, u_{x4}, u_{y4}, u_{z4})^T \tag{A.19}$$

representing $\hat{\boldsymbol{u}}_e$ in Voigt notation. The nodal quantities without a hat are given in Voigt notation forming a column vector of dimension $ndim \cdot n_n$. The off-diagonal entries in the strain measure are multiplied with a factor of 2 in order to keep the internal virtual work of the element $e$ simple. To compute the virtual work, only an inner vector product needs to be evaluated:

$$\delta W_{int,e} = \int_{\Omega_{0,e}} \boldsymbol{S} : \delta \boldsymbol{E} \, \mathrm{d}\Omega_{0,e} = \int_{\Omega_{0,e}} (\delta \boldsymbol{E}^v)^T \boldsymbol{S}^v \, \mathrm{d}\Omega_{0,e}. \tag{A.20}$$

For the application of the principle of virtual work, the variation of the Green-Lagrange tensor $\delta \boldsymbol{E}$ is necessary. According to the definition (2.13), recalled here for clarity:

$$\delta \boldsymbol{E} = \frac{1}{2}(\delta \boldsymbol{F}^T \boldsymbol{F} + \boldsymbol{F}^T \delta \boldsymbol{F}), \tag{A.21}$$

the variation of the Voigt representation $\delta \boldsymbol{E}^v$ can be expressed in terms of the variations of the displacement vector $\delta \boldsymbol{u}$:

$$\delta \boldsymbol{E}^v = \boldsymbol{B}_0 \delta \boldsymbol{u}_e. \tag{A.22}$$

The tangent matrix $\boldsymbol{B}_0 \in \mathbb{R}^{ndim(ndim+1)/2 \times ndim \cdot n_n}$ is dependent on the deformation and hence a linear function of the nodal displacements. It is composed of both entries of the deformation gradient and of the spatial derivative matrix $\widetilde{\boldsymbol{B}}_0$. The first three columns are given as

$$\boldsymbol{B}_0[:, :3] = \begin{pmatrix} F_{11}B_{11} & F_{21}B_{11} & F_{31}B_{11} \\ F_{12}B_{12} & F_{22}B_{12} & F_{32}B_{12} \\ F_{13}B_{13} & F_{23}B_{13} & F_{33}B_{13} \\ F_{12}B_{13} + F_{13}B_{12} & F_{22}B_{13} + F_{23}B_{12} & F_{32}B_{13} + F_{33}B_{12} \\ F_{11}B_{13} + F_{13}B_{11} & F_{21}B_{13} + F_{23}B_{11} & F_{31}B_{13} + F_{33}B_{11} \\ F_{11}B_{12} + F_{12}B_{11} & F_{21}B_{12} + F_{22}B_{11} & F_{31}B_{12} + F_{32}B_{11} \end{pmatrix} \tag{A.23}$$

with

$$\boldsymbol{F} = \begin{pmatrix} F_{11} & F_{21} & F_{31} \\ F_{12} & F_{22} & F_{32} \\ F_{13} & F_{23} & F_{33} \end{pmatrix} \quad \text{and} \quad \widetilde{\boldsymbol{B}}_0 = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ \vdots & \vdots & \vdots \end{pmatrix}. \tag{A.24}$$

Having defined the tangent operator $\delta \boldsymbol{E}^v$, the virtual work $\delta W_{int,e}$ yields the nonlinear internal forces $\boldsymbol{f}_e(\boldsymbol{u}_e)$:

$$\delta W_e = \int_{\Omega_{0,e}} (\delta \boldsymbol{E}^v)^T \boldsymbol{S}^v \, \mathrm{d}\Omega_{0,e} = \delta \boldsymbol{u}_e^T \int_{\Omega_{0,e}} \boldsymbol{B}_0^T \boldsymbol{S}^v \, \mathrm{d}\Omega_{0,e} = \delta \boldsymbol{u}_e^T \boldsymbol{f}_e(\boldsymbol{u}_e) \tag{A.25}$$

as

$$\boldsymbol{f}_e(\boldsymbol{u}_e) = \int_{\Omega_{0,e}} \boldsymbol{B}_0^T \boldsymbol{S}^v \, \mathrm{d}\Omega_{0,e}, \tag{A.26}$$

since the variations of the nodal displacements $\delta \boldsymbol{u}_e$ are arbitrary. The evaluation of the integral in the equation above is performed using Gauss integration (cf. (2.28)).

Next the virtual work of the inertia forces is evaluated. Recalling (2.25), $\delta W_{kin,e}$ is written as

$$\delta W_{kin,e} = \int_{\Omega_{0,e}} \delta \bar{\boldsymbol{u}}^T \rho \ddot{\bar{\boldsymbol{u}}} \, \mathrm{d}\Omega_{0,e}. \tag{A.27}$$

To receive the mass matrix in terms of the displacements $\boldsymbol{u}_e$ in Voigt notation, the matrix of shape functions $\boldsymbol{N}$ need to be given in Voigt notation as well, so that the continuous displacement field is a function of the nodal displacements given in Voigt notation $\bar{\boldsymbol{u}} = \boldsymbol{N}^v \boldsymbol{u}_e$:

$$\boldsymbol{N}^v = \begin{pmatrix} N_1(\boldsymbol{\xi}) & 0 & 0 & N_2(\boldsymbol{\xi}) & 0 & 0 & \dots \\ 0 & N_1(\boldsymbol{\xi}) & 0 & 0 & N_2(\boldsymbol{\xi}) & 0 & \dots \\ 0 & 0 & N_1(\boldsymbol{\xi}) & 0 & 0 & N_2(\boldsymbol{\xi}) & \dots \end{pmatrix} \tag{A.28}$$

Then the virtual work of the inertia forces $\delta W_{kin,e}$ is rewritten as

$$\delta W_{kin,e} = \int_{\Omega_{0,e}} \delta \boldsymbol{u}_e^T (\boldsymbol{N}^v)^T \rho \boldsymbol{N}^v \ddot{\boldsymbol{u}}_e \, \mathrm{d}\Omega_{0,e} = \delta \boldsymbol{u}_e^T \int_{\Omega_{0,e}} \rho (\boldsymbol{N}^v)^T \boldsymbol{N}^v \, \mathrm{d}\Omega_{0,e} \, \ddot{\boldsymbol{u}}_e = \delta \boldsymbol{u}_e^T \boldsymbol{M}_e \ddot{\boldsymbol{u}}_e \tag{A.29}$$

yielding the mass matrix $\boldsymbol{M}_e$ of element $e$ as

$$\boldsymbol{M}_e = \int_{\Omega_{0,e}} \rho (\boldsymbol{N}^v)^T \boldsymbol{N}^v \, \mathrm{d}\Omega_{0,e}. \tag{A.30}$$

Similar to the internal nonlinear forces, the integrals are evaluated with a Gauss quadrature scheme.

Finally the tangential stiffness matrix $\boldsymbol{K}_e$ is derived. It is the Jacobian of the internal forces (A.26) with respect to the nodal displacements. Since in the expression (A.26) both members in the integrand are dependent on the nodal displacements, the stiffness matrix is split into two parts, the geometric stiffness matrix $\boldsymbol{K}_{geo,e}$ and the material stiffness matrix $\boldsymbol{K}_{mat,e}$:

$$\frac{\partial \boldsymbol{f}_e}{\partial \boldsymbol{u}_e} = \boldsymbol{K}_e = \underbrace{\int_{\Omega_{0,e}} \frac{\partial \boldsymbol{B}_0^T}{\partial \boldsymbol{u}_e} \boldsymbol{S}^v \, \mathrm{d}\Omega_{0,e}}_{\boldsymbol{K}_{geo,e}} + \underbrace{\int_{\Omega_{0,e}} \boldsymbol{B}_0^T \frac{\partial \boldsymbol{S}^v}{\partial \boldsymbol{u}_e} \, \mathrm{d}\Omega_{0,e}}_{K_{mat,e}}. \tag{A.31}$$

The geometric stiffness matrix $\boldsymbol{K}_{geo,e}$ is different from zero only for deformed states and is thus omitted in linear finite elements. The material stiffness matrix $\boldsymbol{K}_{mat,e}$ is equal to the linear stiffness matrix for a zero displacement. It is simple to derive using the chain rule:

$$\boldsymbol{K}_{mat,e} = \int_{\Omega_{0,e}} \boldsymbol{B}_0^T \frac{\partial \boldsymbol{S}^v}{\partial \boldsymbol{u}_e} \, \mathrm{d}\Omega_{0,e} = \int_{\Omega_{0,e}} \boldsymbol{B}_0^T \frac{\partial \boldsymbol{S}^v}{\partial \boldsymbol{E}^v} \frac{\partial \boldsymbol{E}^v}{\partial \boldsymbol{u}_e} \, \mathrm{d}\Omega_{0,e} = \int_{\Omega_{0,e}} \boldsymbol{B}_0^T \boldsymbol{C}^{SE} \boldsymbol{B}_0 \, \mathrm{d}\Omega_{0,e}. \tag{A.32}$$

The matrix $\boldsymbol{C}^{SE} = \partial \boldsymbol{S}^v / \partial \boldsymbol{E}^v$ is the so-called *tangent modulus* describing the tangential map between the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$ and the Green-Lagrange strain tensor $\boldsymbol{E}$. This tangent modulus is provided by the constitutive law as discussed in Section 2.2.

The geometric stiffness matrix is more complicated to derive. The internal virtual work can be expressed by another work-conjugate pair, the deformation gradient $\boldsymbol{F}$ and the first Piola–Kirchhoff stress tensor $\boldsymbol{P} = \boldsymbol{F}\boldsymbol{S}$ yielding

$$\delta W_{int,e} = \delta \hat{\boldsymbol{u}}_e : \hat{\boldsymbol{f}}_e = \int_{\Omega_{0,e}} \delta \boldsymbol{F} : \boldsymbol{P} \, \mathrm{d}\Omega_{0,e} = \int_{\Omega_{0,e}} \delta F_{ij} P_{ij} \, \mathrm{d}\Omega_{0,e}. \tag{A.33}$$

Since the variation of the deformation gradient is $\delta \boldsymbol{F} = \delta \hat{\boldsymbol{u}}_e^T \widetilde{\boldsymbol{B}}_0$, in index notation $\delta F_{ij} = \delta \hat{u}_{e,ki} \widetilde{B}_{0,kj}$, the internal forces $\hat{\boldsymbol{f}}_e$ given in matrix notation are

$$\delta W_{int,e} = \delta \hat{\boldsymbol{u}}_e : \hat{\boldsymbol{f}}_e = \delta \hat{u}_{e,ki} \hat{f}_{e,ki} \tag{A.34}$$

$$= \int_{\Omega_{0,e}} \delta F_{ij} P_{ij} \, \mathrm{d}\Omega_{0,e} = \delta \hat{u}_{e,ki} \int \widetilde{B}_{0,kj} P_{ij} \, \mathrm{d}\Omega_{0,e} \tag{A.35}$$

$$= \delta \hat{\boldsymbol{u}}_e : \int_{\Omega_{0,e}} \widetilde{\boldsymbol{B}}_0 \boldsymbol{P}^T \, \mathrm{d}\Omega_{0,e}. \tag{A.36}$$

Hence the nonlinear forces $\hat{\boldsymbol{f}}_e$ in matrix notation are given as

$$\hat{\boldsymbol{f}}_e = \int_{\Omega_{0,e}} \widetilde{\boldsymbol{B}}_0 \boldsymbol{P}^T \, \mathrm{d}\Omega_{0,e} = \int_{\Omega_{0,e}} \widetilde{\boldsymbol{B}}_0 \boldsymbol{S} \boldsymbol{F}^T \, \mathrm{d}\Omega_{0,e}. \tag{A.37}$$

The tangential stiffness matrix can be obtained by taking the time derivative of the internal forces, since Jacobians of matrices are difficult to book-keep:

$$\dot{\hat{\boldsymbol{f}}}_e = \underbrace{\int_{\Omega_{0,e}} \widetilde{\boldsymbol{B}}_0 \dot{\boldsymbol{S}} \boldsymbol{F}^T \, \mathrm{d}\Omega_{0,e}}_{\dot{\hat{\boldsymbol{f}}}_{mat,e}} + \underbrace{\int_{\Omega_{0,e}} \widetilde{\boldsymbol{B}}_0 \boldsymbol{S} \dot{\boldsymbol{F}}^T \, \mathrm{d}\Omega_{0,e}}_{\dot{\hat{\boldsymbol{f}}}_{geo,e}}. \tag{A.38}$$

The first summand represents the change of the forces due to the material change induced by $\dot{\boldsymbol{S}}$. The second summand represents the change of the forces due to the change of the geometry, represented by the change rate of the deformation gradient $\dot{\boldsymbol{F}}$. The second change rate leads to the geometric stiffness matrix $\boldsymbol{K}_{geo,e}$, since the rate of the deformation gradient $\dot{\boldsymbol{F}}$ can be expressed in terms of the nodal velocities to $\dot{\boldsymbol{F}} = \dot{\hat{\boldsymbol{u}}}_e^T \widetilde{\boldsymbol{B}}_0$:

$$\dot{\hat{\boldsymbol{f}}}_{geo,e} = \int_{\Omega_{0,e}} \widetilde{\boldsymbol{B}}_0 \boldsymbol{S} \dot{\boldsymbol{F}}^T \, \mathrm{d}\Omega_{0,e} = \int_{\Omega_{0,e}} \widetilde{\boldsymbol{B}}_0 \boldsymbol{S} \widetilde{\boldsymbol{B}}_0^T \, \mathrm{d}\Omega_{0,e} \; \dot{\hat{\boldsymbol{u}}}_e = \hat{\boldsymbol{K}}_{geo,e} \dot{\hat{\boldsymbol{u}}}_e, \tag{A.39}$$

leading to the geometric stiffness matrix $\hat{\boldsymbol{K}}_{geo}$ for one coordinate direction:

$$\hat{\boldsymbol{K}}_{geo,e} = \int \widetilde{\boldsymbol{B}}_0 \boldsymbol{S} \widetilde{\boldsymbol{B}}_0^T \, \mathrm{d}\Omega_{0,e}. \tag{A.40}$$

The matrix $\hat{\boldsymbol{K}}_{geo,e}$ can be expanded to the geometric stiffness matrix $\boldsymbol{K}_{geo,e}$ for Voigt notation by applying the Kronecker product with the identity matrix $\boldsymbol{I} \in \mathbb{R}^{ndim \times ndim}$:

$$\boldsymbol{K}_{geo,e} = \hat{\boldsymbol{K}}_{geo,e} \otimes \boldsymbol{I}. \tag{A.41}$$

The tangential stiffness $\boldsymbol{K}_e$ of element $e$ is then given as the sum of both, the material and the geometric stiffness:

$$\boldsymbol{K}_e = \boldsymbol{K}_{mat,e} + \boldsymbol{K}_{geo,e}. \tag{A.42}$$

The procedure of the finite element method to compute the elemental matrices $\boldsymbol{K}_e$, $\boldsymbol{M}_e$ and the internal force vector $\boldsymbol{f}_e$ is given in Algorithm 6. For more in-depth information the reader is referred to [BLM00].

**Algorithm 6:** Computation of the nonlinear elemental quantities

**Input :** Nodal positions $\boldsymbol{X}_e \in \mathbb{R}^{ndim \cdot n_n}$, nodal displacements $\boldsymbol{u}_e \in \mathbb{R}^{ndim \cdot n_n}$
**Output :** Nonlinear internal force $\boldsymbol{f}_e \in \mathbb{R}^{ndim \cdot n_n}$, Tangential stiffness matrix $\boldsymbol{K}_e \in \mathbb{R}^{ndim \cdot n_n \times ndim \cdot n_n}$, Mass matrix $\boldsymbol{M}_e \in \mathbb{R}^{ndim \cdot n_n \times ndim \cdot n_n}$

Allocate output variables to zero: $\boldsymbol{f}_e = 0 \in \mathbb{R}^{ndim \cdot n_n}$, $\boldsymbol{K}_e = 0 \in \mathbb{R}^{ndim \cdot n_n \times ndim \cdot n_n}$, $\boldsymbol{M}_e = 0 \in \mathbb{R}^{ndim \cdot n_n \times ndim \cdot n_n}$
Allocate $\hat{\boldsymbol{K}}_{geo,e} = 0 \in \mathbb{R}^{n_n \times n_n}$, $\boldsymbol{K}_{mat,e} = 0 \in \mathbb{R}^{ndim \cdot n_n \times ndim \cdot n_n}$
**for** $\boldsymbol{\xi}_i \in$ *Gauss points and* $w_i \in$ *weights* // Loop over all Gauss points
**do**
  Compute element Jacobian $\boldsymbol{J} = \partial \hat{\boldsymbol{X}}_e / \partial \boldsymbol{\xi}_i$ according to (A.10)
  Compute spatial derivation matrix $\widetilde{\boldsymbol{B}}_e(\boldsymbol{\xi}_i)$ according to (A.13)
  Compute auxiliary matrix $\boldsymbol{H}$ and Green-Lagrange strain $\boldsymbol{E}$ according to (A.14)
  Compute deformation gradient $\boldsymbol{F} = \boldsymbol{I} + \boldsymbol{H}$
  Compute second Piola-Kirchhoff stress tensor $\boldsymbol{S}(\boldsymbol{E})$ and tangent modulus $\boldsymbol{C}^{SE}$ according to constitutive law
  Compute tangent strain matrix $\boldsymbol{B}_0$ according to (A.23)
  Add Gauss point contribution to elemental internal force $\boldsymbol{f}_e$ (cf. (A.26)): $\boldsymbol{f}_e = \boldsymbol{f}_e + \boldsymbol{B}_0^T \boldsymbol{S}^v w_i \det(\boldsymbol{J})$
  Add Gauss point contribution of material stiffness to matrix $\boldsymbol{K}_{mat,e}$ according to (A.32): $\boldsymbol{K}_{mat,e} = \boldsymbol{K}_{mat,e} + \boldsymbol{B}_0^T \boldsymbol{C}^{SE} \boldsymbol{B}_0 w_i \det(\boldsymbol{J})$
  Add Gauss point contribution to geometric stiffness matrix $\hat{\boldsymbol{K}}_{geo,e}$ according to (A.40) and (A.41): $\hat{\boldsymbol{K}}_{geo,e} = \hat{\boldsymbol{K}}_{geo,e} + \widetilde{\boldsymbol{B}}_0 \boldsymbol{S} \widetilde{\boldsymbol{B}}_0^T w_i \det(\boldsymbol{J})$
  Add Gauss point contribution of mass matrix to $\boldsymbol{M}_e$ according to (A.30): $\boldsymbol{M}_e = \boldsymbol{M}_e + \boldsymbol{N}^v (\boldsymbol{N}^v)^T \rho_0 w_i \det(\boldsymbol{J})$
**end**
Compute tangential stiffness matrix $\boldsymbol{K}_e = \boldsymbol{K}_{mat,e} + \hat{\boldsymbol{K}}_{geo,e} \otimes \boldsymbol{I}$ with $\boldsymbol{I} \in \mathbb{R}^{ndim \times ndim}$

# Appendix B

# Parameters of Numerical Experiments

**Table B.1:** Parameters of the examples cantilever, c-shape and cantilever with Neo-Hookean material used in several chapters.

| **Property** | **cantilever** | **c-shape** | **cantilever Neo-Hookean** |
|---|---|---|---|
| **Mesh** | | | |
| no of elements | 246 | 612 | 571 |
| no of dofs | 1224 | 2796 | 2244 |
| element type | Tri6 | Tri6 | Tri6 |
| **Material** | | | |
| Stress configuration | plane stress | plane stress | plane strain |
| Young's modulus $E$ | 210 GPa | 210 GPa | 0.9 GPa |
| Poisson ratio $\nu$ | 0.3 | 0.3 | 0.4 |
| Density $\rho$ | $10000\,\mathrm{kg/m^3}$ | $10000\,\mathrm{kg/m^3}$ | $1000\,\mathrm{kg/m^3}$ |
| **Time integration** | | | |
| time step size $\Delta t$ | $5 \cdot 10^{-4}\,\mathrm{s}$ | $5 \cdot 10^{-4}\,\mathrm{s}$ | $1 \cdot 10^{-3}\,\mathrm{s}$ |
| $t_{end}$ | 1 s | 2 s | 2 s |
| $\rho_\infty$ | 0.8 | 0.8 | 0.4 |
| **Excitation** | | | |
| x-direction | | | |
| excitation | — | $\sin(3 \cdot 2\pi t)$ | — |
| magnitude | — | $1 \cdot 10^7\,\mathrm{N/m}$ | — |
| y-direction | | | |
| excitation | $\sin(50 \cdot 2\pi t) + \sin(8 \cdot 2\pi t)$ | $\sin(5 \cdot 2\pi t)$ | $\sin(4 \cdot 2\pi t) + \sin(25 \cdot 2\pi t)$ |
| magnitude | $5 \cdot 10^6\,\mathrm{N/m}$ | $1 \cdot 10^7\,\mathrm{N/m}$ | $2 \cdot 10^5\,\mathrm{N/m}$ |