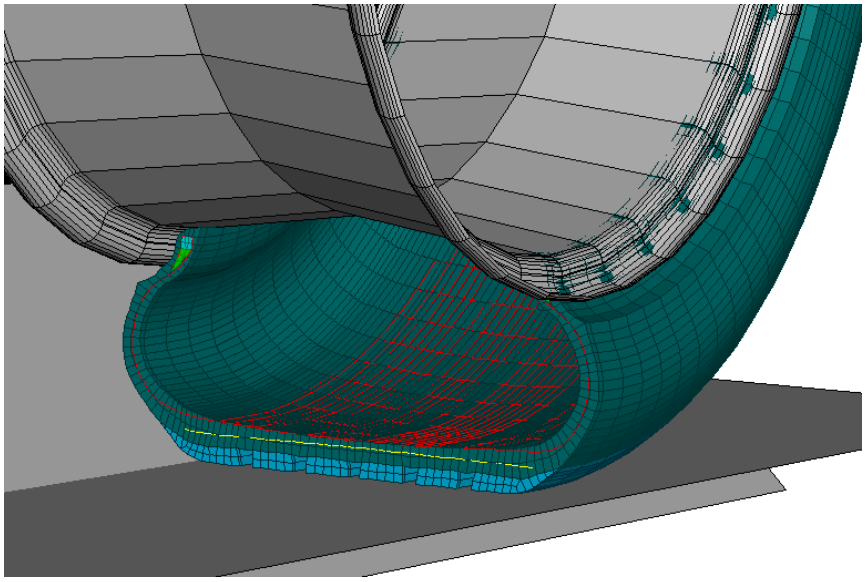
Model Reduction of Nonlinear Problems in Structural Mechanics: Towards a Finite Element Tyre Model for Multibody Simulation

Sabrina Herkt



Vom Fachbereich Mathematik
der Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften (Doctor rerum naturalium, Dr. rer. nat.)
genehmigte Dissertation

Gutachter:
Prof. Dr. René Pinnau, Universität Kaiserslautern
Prof. Dr. Michael Hinze, Universität Hamburg

Datum der Disputation:
19. Dezember 2008

D386

dedicated to the memory of my father

Acknowledgement

I would like to thank all those who - in professional or personal sense - contributed to the completion of this work.

This thesis was generated in the department MDF at Fraunhofer ITWM in Kaiserslautern and funded by the Fraunhofer Gesellschaft.

First of all, I would like to thank my supervisor Prof. René Pinnau. The discussions with him were very valuable in a professional sense, and helped me to structure my work and plans. Moreover, our meetings were always highly encouraging and made me regain confidence when I was in doubt with my work.

I would also like to thank my second referee Prof. Michael Hinze. I had the chance to meet him at the University of Kaiserslautern for a few weeks during his sabbatical. The time spent at the university helped me a lot to retrieve my affection to the mathematical analysis, and I enjoyed the profound discussions with him and René Pinnau.

Furthermore, I would like to thank Dr. Klaus Dreßler, who launched the context of my thesis and gave me the possibility to work in his group. The research topics covered in his department and the contacts to industry were a useful source of inspiration for this thesis.

I am also grateful to my colleagues from department MDF who contributed to a great working atmosphere. They gave me input in discussions and presentations and always showed interest in my work. Besides, they were often available for social activities on weekends and evenings, which made the framework of my PhD time very enjoyable.

Finally I would like to give my special thanks to my family and friends for their continuous support throughout the years.

Contents

Chapter 1

Motivation and Background

1.1 Problem Description

In the framework of automotive manufacturing, computational simulation is already a crucial part of the design process. Quality assurance and competitive capability require technical knowledge as well as intensive testing. To reduce the cost and effort needed for measurements, numerical computations accompany nearly every step of development and optimisation.

For most mechanical computations in this context, it is not sufficient to observe only the behaviour of single components. It is also important to understand the interaction of a vehicle's elements in order to estimate the effect of changes in construction. Every element that undergoes constructional modifications - such as geometry or material - changes the entire system and its dynamical properties. Due to the modified deformation and inertia properties, the stress resultants in the connection to adjacent parts are affected. This effect is carried on through the system and can again influence forces on the altered component. Therefore, automotive simulation has to rely on description and computation of the whole vehicle.

To assess the dynamical properties of vehicles, the analyses basically address issues of *handling*, *comfort*, *durability* or combinations thereof.
The term handling describes the behaviour of the vehicle on a smooth road or one that contains only light unevennesses with low frequencies. Here, lateral and longitudinal effects are dominant, like in cornering or braking and accelerating.
If the vertical transmittance behaviour is investigated, for example the vibrations passed on to the driver when riding over a bumpy road, the problem is generally called comfort analysis.
In the third possibility of application - the durability - the stress history that single elements of the vehicle undergo in a certain dynamical load case is examined. This data provides information on the possible damage of the element and allows a prediction of its overall lifespan.

All three analysis cases share the need to model a full vehicle with all its components. In addition to that, it is necessary to describe the process in its entire time span. For example, test tracks usually contain sections with strongly differing characteristics. Due to the highly nonlinear behaviour of the system, no extrapolation of the results can be made from a smooth part of the track to a rough one. Therefore, the whole load history needs to be accounted for.

In the cases described here, experiment durations of several minutes are in a usual range.

1.1.1 Tyres in Vehicle Simulation

In full vehicle simulation, tyres play a crucial role. In general, the tyre is the only element that is in contact with the road. Therefore, it transmits all road excitations to the system by transforming them into forces and moments on the spindle. Although most tyres share the same build-up concerning their structural composition (see Figure 1.1), there are large differences in their dynamical behaviour.

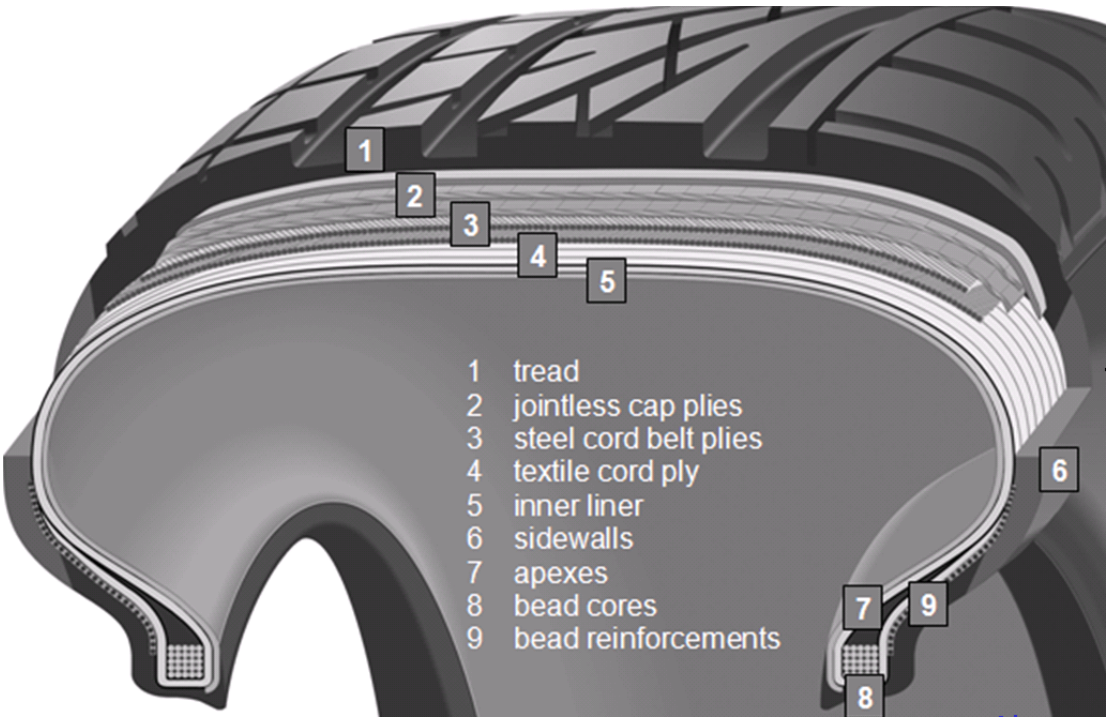


Figure 1.1: Exemplary build-up of an automotive tyre [19]

The largest portion of the tyre volume consists of rubber material. Usually, different rubber mixtures are used for the outer area (especially the tread), the inner area and the filler material. The detailed material composition differs between the tyre manufacturing brands and is a well protected secret. Changes in tread material strongly affect the road friction

and the tread wear [34]. Variations of sidewall or belt material stiffness influence handling and comfort properties.

The geometry of the tyre depends on the area of application. Of course, tyres for aircrafts, trucks, motorcycles and tractors strongly vary in size and shape. Figure 1.1 shows an exemplary composition of a radial automotive tyre. However, most of its elements can be found in other tyre types as well.

The area that is attached to the rim, the bead, consists of several steel cords. It can be regarded as a massive steel core which provides stability and prohibits the tyre's displacement off the rim.

Automotive and aircraft tyres additionally contain steel, textile or synthetic plies reinforcing the sidewall and the inner area of the tread (see Figure 1.1). These are called carcass and belt, respectively. In modern automotive tyres, the carcass is built up radially, i.e. the fibres run orthogonal to the direction of travel. The belt consists of several cord layers at an angle of $0 - 60°$ to the circumferential direction. Successive layers are arranged in opposite direction to achieve a balanced and symmetric stress distribution.

The complex interaction of the tyre's elements and materials determines its performance and lifespan.

For the modelling of tyres in vehicle simulation, it is of particular importance to capture and accurately describe their dynamical transmittance properties. Highly nonlinear effects can be observed in both lateral and longitudinal forces and moments with respect to slip quantities. This nonlinearity arises from stick/slip effects inherent in dynamical friction ([34], [21]). Vertical resultants show nonlinearities due to road contact. In addition, dependencies on velocity, inner pressure and sometimes heat distribution and tread wear need to be accounted for ([17]).

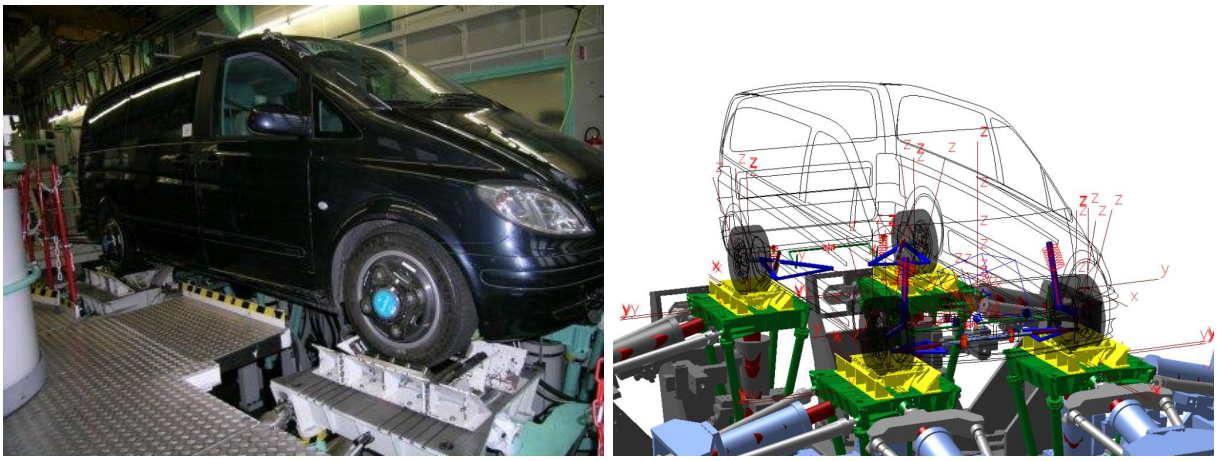
1.2 General Modelling Techniques

In this section, two popular approaches in vehicle simulation are presented: the Multibody Simulation incorporating mainly rigid elements and the Finite Element Method used to describe deformable bodies.

1.2.1 Multibody Simulation

In the areas of application mentioned above, *Multibody Simulation (MBS)* has grown in importance. Figure 1.2 shows a test rig for comfort and durability experiments and the corresponding Multibody Simulation model created with SimPack.

The underlying simulation concept is based on the assumption that all parts of the model can be described as *rigid bodies* characterised by their mass and inertia properties. This assumption is both necessary and legitimate in the context of vehicle simulation. Due to



(a) Test rig (b) Simulation model

Figure 1.2: Vehicle on a test rig and simulation model in SimPack (see [47])

the long simulation timespan, it is important to keep the number of degrees of freedom as small as possible. The assumption of rigidity is valid as long as the forces caused by deformation of the bodies are negligible compared to its inertia effects. This prerequisite applies to regular use of vehicles, where most elements undergo large rigid body motions and accelerations but only very small elastic deformations.

This set-up can be described by a set of ordinary differential equations (ODE)

$$M\left(q\right)\ddot{q} = f\left(t, q, \dot{q}\right) \tag{1.1}$$

where the vector $q \in \mathbb{R}^N$ contains the displacements and rotations of the bodies with a total of N degrees of freedom. In three-dimensional simulations, each body has 6 possibilities of rigid body motion. Dots denote derivatives with respect to time. The symmetric positive definite matrix $M \in \mathbb{R}^{N\times N}$ is called the mass matrix of the system. The vector $f \in \mathbb{R}^N$ describes the external forces and the ones arising in *force elements*, like springs or dampers. Both M and f depend on the current configuration q.

Furthermore, bodies can be attached to each other by fixed connections or *joints* which lock some of their degrees of freedom. Interactions of this type can generally not be described by force elements as the relative motions need to be set to zero exactly and not only to a small value. Restrictions of this type are called *constraints* and induce algebraic side conditions into the equations of motion:

$$M\left(q\right)\ddot{q} = f\left(t, q, \dot{q}\right) - G(q)^T\lambda \tag{1.2a}$$

$$g(q) = 0 \tag{1.2b}$$

where $g(q) \in \mathbb{R}^m$ is the vector of constraints and $G(q)$ denotes its Jacobian $G(q) = \nabla g \in \mathbb{R}^{m\times N}$. The vector $\lambda \in \mathbb{R}^m$ contains the Lagrange multipliers. Such system

of equations consisting of ODEs with algebraic constraints is called a set of *differential algebraic equations (DAEs)*.

In some simulations, the constraints can be eliminated by insertion into the ODE. This is generally possible if the model can be described by an open tree structure. If closed loops are contained in the model, this description can only be done locally ([41]). For a general treatment - like in commercial MBS software - the DAE system needs to be solved respecting its peculiar theory and numerics ([2], [16], [8]).

1.2.2 Finite Element Description of Elastic Bodies

As a very general approach, the *Finite Element Method (FEM)* is well established in computational mechanics for the static and dynamical simulation of flexible bodies. It is based on a discretisation of the continuum mechanical equilibrium equations in their weak form. For completion, only a very short derivation of the discretised equations is given here. For a detailed description of the method, see, for example, [51], [52] or [50].

Equations of Motion for an Elastic Body

Figure 1.3 shows a flexible body in its reference configuration and in its deformed state.

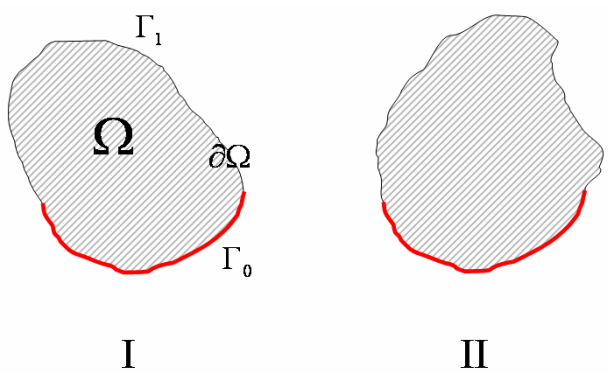


Figure 1.3: Flexible body in undeformed (I) and deformed (II) state

In its undeformed state, the body is described by a domain $\Omega \subset \mathbb{R}^3$ and its boundary $\partial\Omega$. This boundary is divided into two disjoint parts, Γ_0 and Γ_1, on which a given displacement or surface traction is prescribed, respectively.

The governing equation describes the *force equilibrium* for a continuum:

$$\rho\ddot{u}(x,t) = div\sigma\left(u(x,t)\right) + \beta(x,t) \quad \forall x \in \Omega,\ t \in I \tag{1.3}$$

where $u(x,t)$ denotes the displacement in space and time, σ the stress distribution and β the external forces. Dots denote derivatives with respect to time and ρ is the density of

the material. Here, $\beta, u : \Omega \times I \to \mathbb{R}^3$ and $\sigma : \Omega' \to \mathbb{R}^{3\times 3}$ are smooth functions in space Ω and on the time interval I.

The equation can be derived directly from the momentum conservation principle, by energy minimisation, by the principle of virtual work or virtual displacements or by solution of a Lagrange/Hamilton formalism ([41], [16]).

In addition to the equilibrium equation, the displacement u must fulfil given boundary conditions

$$u(x,t) = u_0(x,t) \quad on \quad \Gamma_0 \subset \partial\Omega, \ \forall t \in I \tag{1.4a}$$

$$\sigma(u) \cdot n = \tau(x,t) \quad on \quad \Gamma_1 \subset \partial\Omega, \ \forall t \in I \tag{1.4b}$$

where n denotes the outer normal vector of $\partial\Omega$ and τ is the given surface stress on Γ_1.

The crucial aspect in the preceding equilibrium equation is the dependency of the stresses σ on the deformation u. The equation of *kinematics* describes how local changes in length induce strains ε in the body:

$$\varepsilon = \frac{1}{2}\left(\nabla u + \nabla u^T + \nabla u^T \nabla u\right) \tag{1.5}$$

In the case of *linear elasticity*, where strains and deformations are assumed to be small, the equation is written as:

$$\varepsilon = \frac{1}{2}\left(\nabla u + \nabla u^T\right) \tag{1.6}$$

The system of equations is completed by the definition of a *material law*. This specifies the connection between the strains ε and the stresses σ. The equation can describe effects of plasticity, hyperelasticity, viscoelasticity, as well as non-isotropic material, creep or damage. Many more material characteristics can be modelled depending on the intended load case.

Due to the diversity of material effects, no general description of the equation can be made. The simplest example is an isotropic linear elastic material, described by Hooke's law:

$$\underline{\sigma} = C \cdot \underline{\varepsilon} \tag{1.7}$$

where

$$\underline{\sigma} = (\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{23}, \sigma_{13})^T \tag{1.8a}$$

$$\underline{\varepsilon} = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \varepsilon_{12}, \varepsilon_{23}, \varepsilon_{13})^T \tag{1.8b}$$

are re-arrangements of the symmetric matrices σ and ε. The constitutive matrix C contains the material constants *Young's modulus* E and *Poisson number* ν.

For the numerical treatment of the equation system, the equilibrium equation is transformed into its weak form. For the linear elastic case, a short description of the relevant functional spaces is given.

Let H denote the Sobolev space $H^m(\Omega)$ containing all elements $u \in L^2(\Omega)$ that possess weak derivatives $D^\alpha u \in L^2(\Omega)$ for all multiindices $\alpha \in \mathbb{N}^m,\ \|\alpha\| \leq m$.

Equipped with the scalar product

$$(u,v)_H := \sum_{\|\alpha\| \leq m} \langle D^\alpha u, D^\alpha v \rangle_{L^2(\Omega)} \tag{1.9}$$

H is a Hilbert space.

V is a linear subspace of H, $V \subset H$, which contains the elements with vanishing trace at the boundary:

$$V = \{v \in H^m(\Omega),\ tr(v) = 0 \quad on \quad \Gamma_0\} \tag{1.10}$$

Then the weak equation of linear elasticity reads:

Find $u \in V$ such that, for almost every $t \in I$,

$$\langle \ddot{u}, v \rangle + a\,(u,v) = \langle \beta, v \rangle \quad \forall v \in V \tag{1.11}$$

with the linear functional $\beta \in H'$, and the continuous symmetric coercive bilinear form $a(\cdot,\cdot)$, defined by

$$a(u,v) := \int_\Omega \sigma(u) : \varepsilon(v) dx \tag{1.12}$$

where : denotes the scalar product of two tensors, $\sigma : \varepsilon = trace(\sigma^T \varepsilon)$

Finite Elements and Galerkin Method

Let V^n be a linear subspace of V with $dim V^n = n$ and a basis $\{\Phi_1, \ldots, \Phi_n\}$. Then an element u_n of V^n can be expressed by

$$u_n(x,t) = \sum_{i=1}^{n} \Phi_i(x) q_i(t) \tag{1.13}$$

Equivalently defining test functions $v_j \in V^n$ by simply using $v_j = \Phi_j(x)$, the weak equilibrium equation is transformed to

$$\sum_{i=1}^{n} \ddot{q}_i \langle \Phi_i, \Phi_j \rangle + \sum_{i=1}^{n} q_i a\,(\Phi_i, \Phi_j) = \langle \beta, \Phi_j(x) \rangle \quad \text{for} \quad j = 1, \ldots, n \tag{1.14}$$

which can be written as

$$\begin{aligned} M\ddot{q} + Kq &= f \\ \text{with} \quad M &= \langle \Phi_i, \Phi_j \rangle \\ K &= a\,(\Phi_i, \Phi_j) \end{aligned} \tag{1.15}$$

Here, M and K are called the mass and stiffness matrix, respectively, and f is the vector of external forces.

As all materials inhere some sort of dissipation, a respective damping term is included in the equations. Depending on the material effects, this can become arbitrarily complex. A frequently used damping description uses a proportional damping matrix $C = \alpha M + \beta K$ yielding a dynamical system

$$M\ddot{q} + C\dot{q} + Kq = f \tag{1.16}$$

This approach is called Rayleigh damping.

In the popular case of linear Finite Elements, the set Ω is divided into cuboid or trapezoidal elements. Their corners define nodes $x_i, i = 1, \ldots, n$ and the shape functions Φ_i are constructed to fulfil

$$\Phi_i(x_j) = \delta_{ij} \tag{1.17}$$

These local functions yield sparse mass and stiffness matrices which are strongly banded. In this case, the participation factors $q_i(t)$ correspond to the nodal displacements $u(t, x_i)$. Therefore, the unknown variables are often denoted by the displacement vector $u(t)$.

Using a suitable time integration scheme (e.g. Newmark's method), the semi-discretised equation 1.15 can be solved numerically over time.

Tyres in Nonlinear Structural Mechanics

A tyre modelled by the Finite Element Method must incorporate all three basic sources of nonlinearities in mechanics:

- *geometrical nonlinearities* that arise in large deformation problems
- *material nonlinearities* which must be taken into account when hypo- or hyperelastic materials undergo large or even only moderate strains, and
- the nonlinear behaviour of *contact conditions*.

In general mechanical problems with large strains or nonlinear material laws, the Finite Element equilibrium equation 1.15 contains a nonlinear inner forces term $R(q)$:

$$M\ddot{q} + R(q) = f \tag{1.18}$$

Here, the stiffness matrix is defined as the derivative of R, $K(q) = \frac{\partial R}{\partial q}$.

However, this approach induces a severe problem of simulation time. A Finite Element tyre model which properly describes all relevant aspects can have several ten thousands degrees of freedom. Compared to the low dimension of a full vehicle multibody system, this yields an exponential rise in computational effort.

Some contexts of Multibody Simulation even require real-time applicability, for example hardware-in-the-loop or human-in-the-loop problems. Therefore, the tyre description needs to be kept simple and computationally inexpensive.

The full automotive test rig run mentioned above takes approximatively 180 seconds. The computation of a single Finite Element tyre passing a cleat within 0.5 seconds keeps 4 processors occupied for 15 hours. This comparison shows that it is clearly impossible to include four Finite Element tyres into a vehicle simulation without simplification.

Fortunately, a detailed description of the tyre structure and the exact stress-strain distribution is not needed for automotive simulation. Questions of localising stress peaks or material failure generally lie in the field of interest of tyre manufacturers.
For the assessment of vehicle dynamics, only the tyre's dynamic effects count. It is therefore sufficient to properly describe the transfer behaviour with respect to vertical, lateral and longitudinal excitations.

1.3 Tyre Models in Multibody Simulation

To handle the problem of an accurate but fast tyre description, a large number of different approaches can be found in commercial MBS codes and research projects.

Among the most frequently used models range Pacejka's *magic formula tyre*, Gipser's *FTire* and *RmodK* by Oertel and Fandre.

Figure 1.4 shows an overview of different commercially available tyre models arranged in order of complexity, which is defined here as the number of degrees of freedom. In this example, higher complexity generally implies a wider range of application.

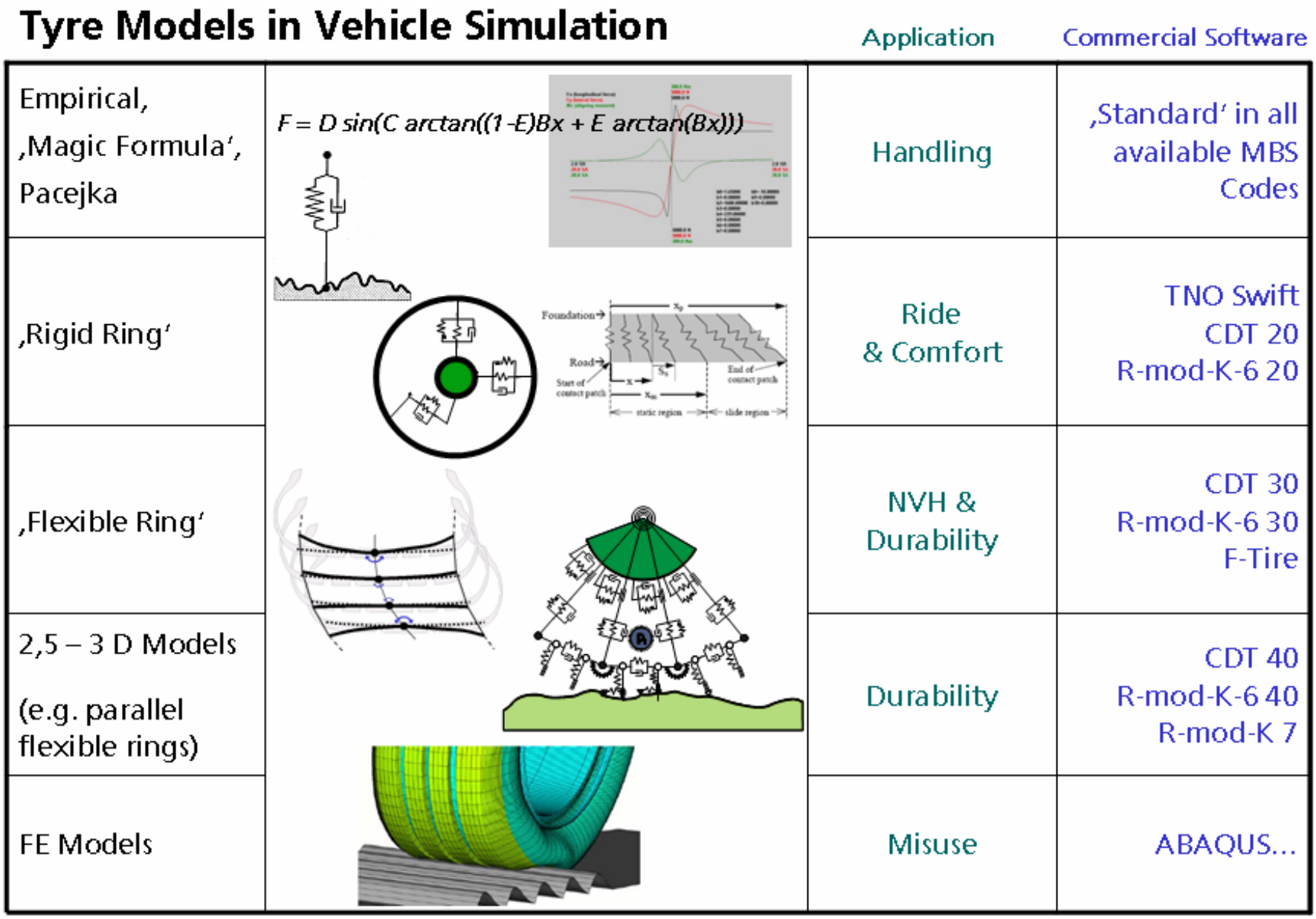


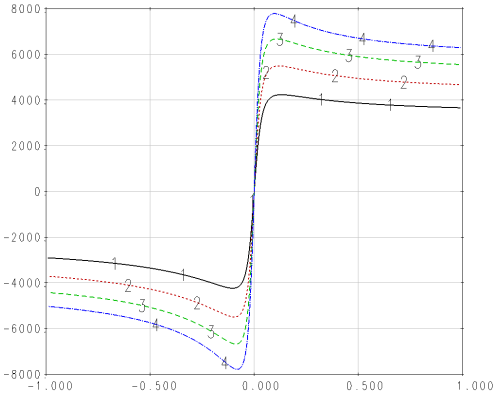
Figure 1.4: Overview of popular commercially available tyre models

1.3.1 Magic Formula Tyre

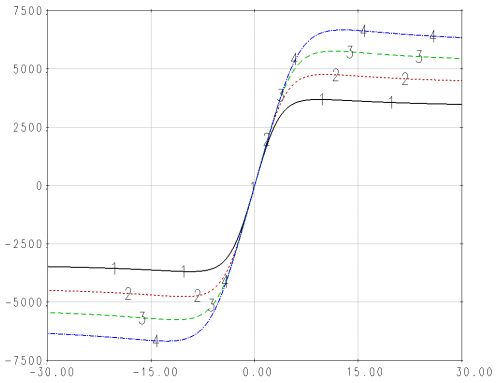
In short, Pacejka's model is based on the assumption that all effects in the tyre can be modelled with a single equation called the magic formula [40]:

$$y(x) = D \sin\left[C \arctan\left(Bx - E\left\{Bx - \arctan Bx\right\}\right)\right] \tag{1.19}$$

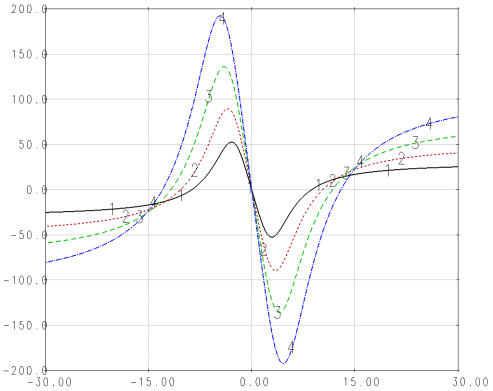
where x describes a slip quantity (longitudinal or lateral) and y the respective force. The parameters B, C, D and E are determined from least squares fits to measurement curves. Figure 1.5 shows exemplary characteristics for lateral and longitudinal force as well as aligning torque represented by the magic formula model.



(a) Longitudinal force w.r.t. longitudinal slip



(b) Lateral force w.r.t. side slip angle



(c) Aligning moment w.r.t. side slip angle

Figure 1.5: Forces and moment for an automotive tyre represented by a magic formula model (created with *SimPack*).
Numbering denotes different vertical load cases: (1)$F_z = 4000N$, (2)$F_z = 5333N$, (3)$F_z = 6667N$, (4)$F_z = 8000N$

Longitudinal slip occurs when the tyre is rotating faster or more slowly than the current vehicle velocity would imply, i.e. in the event of accelerating or braking. Consequently, a

lateral slip angle occurs in cornering manoeuvres, when the velocity direction of the road contact area does not coincide with the one of the rim. The aligning torque is defined as the moment around the vertical axis. It is calculated from the lateral and longitudinal forces using a weighting factor that results from the cosine form of the magic formula 1.19.

The vertical dynamics are modelled by a linear spring-damper element with a single-point contact formulation (see Figure 1.6(a)). This simplification is clearly invalid for road unevennesses with high frequencies or cleats. Without modifications, the model is mainly designed for handling problems and generally not suitable for comfort analyses. However, due to its simple description, the magic formula tyre is one of the few models usable in real-time applications.

For a better treatment of handling and light comfort problems, several extensions of the model exist. For example, the tyre model *SWIFT* is based on a magic formula formulation with a more elaborate road contact description to handle high frequency obstacles. The contact area is determined by two elliptical cams moving with the tyre's ground contact point (Figure 1.6(b)). From their trajectory, the effective road profile is computed. With this approach, the enveloping properties of the tyre are accounted for. For further details, see [40].

1.3.2 FTire

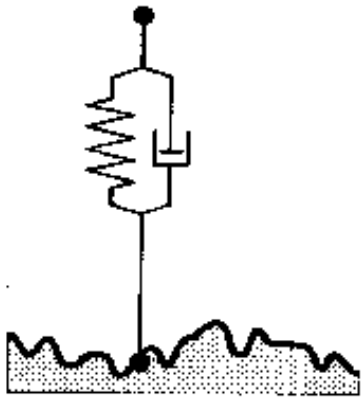
The model *FTire* [19] is based on the approach that the belt is decomposed into deformable beam-like elements (Figure 1.6(c)). Its parametrisation requires some geometrical information on rim, belt and tread, as well as stiffnesses and friction data.

Different parts of the package FTire Tools help to parametrise the tyre based on available inputs. The tyre's geometry can serve for a first estimation of the parameters. If additional measurement data is known, the FTire parameters can be adapted further to better match these results. The data may include forces from a cleat test, eigenfrequencies or static stiffness measurements.

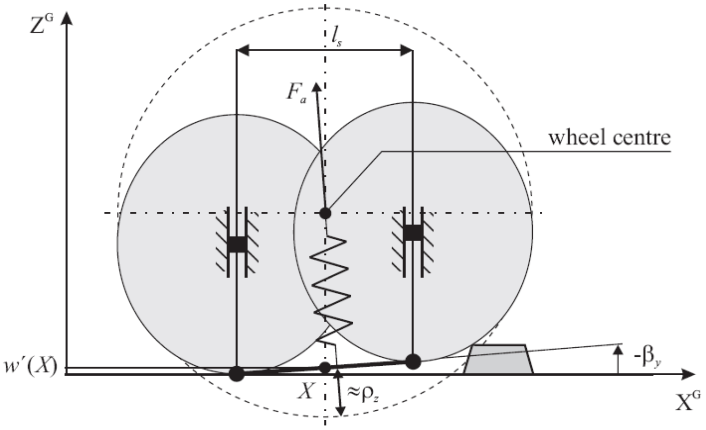
1.3.3 RmodK and CDTire

Oertel and Fandre's *RmodK.6* [1] also sets the basis for *CDTire* [2] in the context of LMS's software Virtual Lab. The model is actually a group consisting of three models for different ranges of application. The simplest of the three, RmodK20, resp. CDTire 20, consists of a rigid ring which is connected to the rim by three spring and damper elements for in-plane forces and moments. As it cannot represent deformations of the belt, it is merely used for obstacles of rather large wavelengths. For a more accurate description of the tread

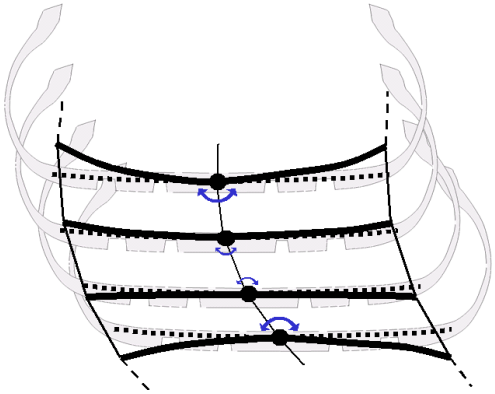
[1] Reifenmodell für Komfortuntersuchungen - tyre model for comfort analysis
[2] comfort and durability tire



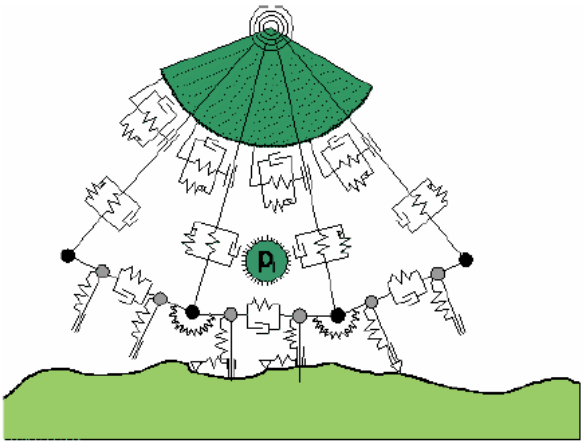
(a) Magic Formula Tyre (from [40])



(b) MF-SWIFT (from [40])



(c) FTire (from [19])



(d) RmodK 6.30 (from [9])

Figure 1.6: Schematic structure of different commercial tyre models

behaviour, the contact patch is discretised by an element grid with a prescribed pressure distribution.

RmodK30, resp. CDTire30, is composed of a given number of mass points connected by springs which represent the flexible belt (Figure 1.6(d)). Each mass point is connected to the rim by a radial spring-damper element with additional springs working against shearing and bending of the sidewalls and the belt. The circular arc between two adjacent mass points is equipped with a number of sensor points to track the road profile and induce friction with the ground. The model is suitable for in-plane excitations with deformations of the belt and problems with moderate lateral effects [9].

Finally, RmodK31, resp. CDTire40, can be viewed as a number of RmodK30 tyres shifted in direction of the rotational axis with interconnections in between to model deformations

of the belt in axis direction. With this approach, out-of-plane excitations like running along a lateral slope, can be described.

The latest version of RmodK, v7.0, is developed from a Finite Element background partly relying on beam theory.

1.3.4 Tyre Model Parametrisation

In all MBS tyre models the problem of parametrisation is an essential issue. The magic formula tyre parameters must be adapted to handling measurements on specially designed tyre test rigs, like on a rotating drum or on conveyor belts. Amongst others, an RmodK tyre uses the eigenfrequencies of the system as input and therefore requires eigenmode measurements.

In both cases the parameter sets are not transferable from one tyre type to another. The complex influence of slight changes in geometry or material composition generally does not allow calculation of the changed parameter set based on the original one. This implies that it takes separate measurements even for only slightly different tyres.

Another disadvantage of most existing models is the limited range of application due the strong connection to passenger cars. The sizes of these tyres only vary a few centimetres in width and radius. Furthermore, the approximation of rotational symmetry holds, which is a basic assumption of all MBS tyre models mentioned above. For the simulation of trucks or agricultural vehicles, the tyre effects are diverse which possibly leads to the necessity of adapted parametrisation routines ([23]). For example, the tread pattern of tractor tyres can not be regarded as rotationally symmetric, and the position of the lugs needs to be taken into account. In addition to that, the parametrisation procedure for both truck and agricultural tyres is problematic, as their sizes and load cases lead to high requirements on the test rig.

For parametrisation of a Finite Element tyre model, a proper geometry and material description of the individual components is necessary. This can be a labour-intensive task, as the nonlinear material properties need to be sufficiently described by a suitable material law and its parameters be determined accurately. However, the test environments needed for static material tests are less expensive and more reliable than the tyre test rigs, especially for truck and agricultural tyres. Simple measurements that are cheaper to achieve - like eigenfrequencies - can be used to further adapt the Finite Element parameters to match the real tyre.

1.4 Aim of this Thesis

As shown in the preceding sections, the classical MBS tyre models show good properties for full vehicle simulations concerning computational effort and accuracy. However, the

parametrisation routines can become problematic, as often expensive measurements are required.

Additionally, the range of applicability of spring-damper models is strongly restricted. In non-standard setups like test rig or misuse applications, or for larger tyres with high loads and considerable influence of the profile, all established MBS tyre models sooner or later reach their limit of applicability.

In contrary to that, a Finite Element tyre description provides high flexibility in modelling and is easier to parametrise. Yet, the large number of degrees of freedom prohibits the straight-forward inclusion of an FEM tyre into an MBS vehicle model.

The aim of this thesis is to use the advantages of Finite Element description, i.e. the easier access to parameters and the modelling possibilities, but to bring it into a dimension that is handleable in a Multibody context.

With this intention, techniques of model reduction are used. Emanating from a Finite Element description for flexible bodies, reduction methods capture the dominant effects of the full Finite Element model and reproduce them with a lower dimensional model. Some of the nonlinear properties of the Finite Element tyre shall be handled.

Chapter 2 presents some popular linear and nonlinear model reduction methods based on projection of the original equations. In Chapter 3, an error analysis of the POD method is performed. Chapter 4 deals with the nonlinearities inherent in the Finite Element model and describes a lookup table approach to handle the nonlinear effects in a reduced model. In Chapter 5, the nonlinear model reduction techniques are applied to a Finite Element example set up in the commercial code Abaqus.

Chapter 2

Model Reduction Methods

Simulation of industrial problems like flow or heat transfer often requires the solution of large linear or nonlinear systems consisting of several ten thousands degrees of freedom. Problems of such high dimensions can be handled by using powerful computers with large storage capabilities. Additionally, in some applications, these simulations need to be repeated several times with slightly different input, like in general controller design problems or in the durability simulation of wind turbines ([33]). Often even real-time applicability is required, like in multibody dynamics with hardware-in-the-loop or human-in-the-loop systems.

In the context of tyre modelling in Multibody Simulations, a Finite Element model (see appendix A) is able to describe the dynamical effects of a tyre without the disadvantages of the established spring-damper models. However, the large number of degrees of freedom needed for a full continuum description makes it difficult to combine the benefits of the Finite Element Method with the system representation in a Multibody Simulation framework.

In all these cases, simulation time becomes an important issue.

Over the years, various methods of model reduction for both linear and nonlinear systems have been developed. These methods allow the construction of low-dimensional reduced models conserving the essential properties and features of the large model.

In the following sections, some popular reduction methods for linear and nonlinear systems are presented. All methods described here are based on subspace projection of the original system and mainly differ in the construction of the corresponding subspace definition.

2.1 Popular Model Reduction Methods for Linear Systems

In this section, three commonly used projection methods for model reduction are shortly described: The Krylov method, the method of balanced truncation and the family of methods based on eigenmode analysis. Whereas the first two approaches originate from a first order application field, the latter was developed for second order structural mechanics problems. It constitutes the state of the art in MBS software for including flexible bodies into a rigid body system.

2.1.1 Krylov and Moment Matching Methods

Originally, the method of Krylov subspace projection is designed for first order problems. It is well established in linear system theory where the governing equations are expressed in state-space form ([44])

$$E\dot{x}(t) = Ax(t) + Bu(t) \tag{2.1a}$$

$$y(t) = Cx(t) + Du(t) \tag{2.1b}$$

where $x(t) \in \mathbb{R}^n$ denotes the state of the system, $u(t) \in \mathbb{R}^m$ the input vector, and $y(t) \in \mathbb{R}^p$ the output. $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$ are called the system matrices.

The method of moment matching ensures that the first j moments of the transfer function of the reduced system match with the ones from the full system, where the moments are defined by

$$T_j = C(A^{-1}E)^j A^{-1}B \tag{2.2}$$

Krylov subspaces implicitly fulfil the moment matching demand without the need to compute the moments of the system ([30]).

The Krylov subspace for the matrix X and the matrix of starting vectors Y of order $j \in \mathbb{N}$ is defined as:

$$\mathcal{K}_j(X, Y) = colspan\{Y, XY, X^2Y, \ldots, X^{j-1}Y\} \tag{2.3}$$

For model reduction via Krylov subspaces, often a two-sided projection method is proposed:

$$Z^T EV\dot{x}_r(t) = Z^T AVx_r(t) + Z^T Bu(t) \tag{2.4a}$$

$$y(t) = CVx_r(t) + Du(t) \tag{2.4b}$$

where the columns of the matrices Z and V arise from an orthogonalisation of two different Krylov subspaces, associated with the input and the output behaviour of the system, respectively.

Choosing, for example, Z such that its columns span the subspace

$$\mathcal{K}_{j_1}(A^{-1}E, A^{-1}B)$$

and V with the columns of

$$\mathcal{K}_{j_2}(A^{-T}E^T, A^{-T}C^T),$$

a matching of the first $j_1 + j_2$ moments is achieved (see [20], [30]).

For second order systems, there are two possibilities: First, a second order system of the form

$$M\ddot{q} + C\dot{q} + Kq = f \tag{2.5}$$

can be transformed into state space form by setting:

$$\begin{bmatrix} I_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & M \end{bmatrix} \begin{pmatrix} \dot{q} \\ v \end{pmatrix} = \begin{bmatrix} 0_{N\times N} & I_{N\times N} \\ -K & -C \end{bmatrix} \begin{pmatrix} q \\ v \end{pmatrix} + \begin{bmatrix} 0_{N\times N} \\ I_{N\times N} \end{bmatrix} f \tag{2.6a}$$

$$y = [I_{N\times N} \quad 0_{N\times N}] \begin{pmatrix} q \\ v \end{pmatrix} + [0_{N\times N}]\, f \tag{2.6b}$$

where $v := \dot{q}$. Here, $M, C, K \in \mathbb{R}^{N\times N}$ denote mass, damping and stiffness matrix and $q \in \mathbb{R}^N$ is the vector of nodal coordinates, as in the Finite Element context (1.16).

With this approach the Krylov projection method can be performed analogously to the first order case. However, in this case, the structure preserving property of the method is not assured.

A second way is to define second order Krylov subspaces and to apply them directly to the second order equation:

$$Z^T M V \ddot{z}_r + Z^T C V \dot{z}_r + Z^T K V z_r = Z^T G u \tag{2.7a}$$

$$y = L V z_r \tag{2.7b}$$

with the second order input and output Krylov subspaces defining the column space of V and Z respectively:

$$\mathcal{K}_{j_1}(-K^{-1}C, -K^{-1}M, -K^{-1}G)$$
$$\mathcal{K}_{j_2}(-K^{-T}C^T, -K^{-T}M^T, -K^{-T}L^T)$$

with $G \in \mathbb{R}^{N\times m}$ and $L \in \mathbb{R}^{p\times N}$.

Lohmann and Salimbahrami ([31]) provide the definition of second order Krylov subspaces as follows:

$$\begin{aligned} \mathcal{K}_j(A_1, A_2, G_1) &= colspan\{P_0, P_1, \ldots, P_{j-1}\} \\ \text{using} \quad P_0 &= G_1 \\ P_1 &= A_1 P_0 \\ P_i &= A_1 P_{i-1} + A_2 P_{i-2}, i = 2, 3, \ldots \end{aligned}$$

When using this direct approach, the second order structure of the system is preserved. Furthermore, the reduction requires less computational effort than the treatment in state space. See [31] for details.

2.1.2 Balanced Truncation

The method of balanced truncation is designed for first order problems given in state space form:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.8a}$$

$$y(t) = Cx(t) + Du(t) \tag{2.8b}$$

The idea is to transform the variables to a basis which makes it easy to identify the states which require much control energy but yield little observing energy ([30], [43]).

Again, this is achieved by a two-sided projection approach:

$$Z^T V \dot{x}_r(t) = Z^T A V x_r(t) + Z^T B u(t) \tag{2.9a}$$

$$y(t) = CV x_r(t) + Du(t) \tag{2.9b}$$

where the matrices Z and V are computed from the *controllability* and *observability* Gramians W_c and W_o by singular value decomposition.

The underlying matrices W_o and W_c are the unique solutions of the Lyapunov equations

$$W_o A + A^T W_o + C^T C = 0 \tag{2.10a}$$

$$\text{or} \quad A W_c + W_c A^T + B B^T = 0 \tag{2.10b}$$

respectively.

Note that the computation of the projection matrices is restricted to models of first order. As for the Krylov subspace method, a second order system can be transformed into state-space form to apply the balanced truncation method in the regular manner. Again, approaches for a straight-forward second order treatment are successfully established in [10], [32].

2.1.3 Methods Based on Eigenmode Analysis

Most multibody programmes provide the possibility to include FEM-discretised flexible components into a rigid body structure. Frequently, they use an approach based on the *Craig-Bampton* method ([12]).

Emanating from the homogeneous linear equilibrium equation for a flexible body in FEM

$$M\ddot{u} + C\dot{u} + Ku = 0 \tag{2.11}$$

a linear eigenmode analysis of the system is performed. Note that the matrices M, C, K can either incorporate the boundary conditions or describe the free structure.

First, using a Fourier series approach, the deformation is decomposed into shape functions in space and participation factors in time:

$$u(x,t) = \sum_{k=1}^{N} p_k(t) \cdot \Phi_k(x) = \sum_{k=1}^{N} e^{i\omega_k t} \cdot \Phi_k(x) \tag{2.12}$$

Inserting this approach into the equilibrium equation, we get

$$\sum_{k=1}^{N} -\omega_k^2 M\Phi_k(x) + i\omega_k C\Phi_k(x) + K\Phi_k(x) = 0, \quad k = 1, \ldots, N \tag{2.13}$$

For a structure without damping, the search for the factors ω_k yields a generalised eigenvalue problem for the mass and stiffness matrices:

$$\left(K - \omega_k^2 M\right)\Phi_k = 0, \quad k = 1, \ldots, N \tag{2.14}$$

Thus, the resulting quantities ω_k and Φ_k are called the eigenfrequencies and the eigenmodes of the system.

Despite of the derivation of the modes, the projection is also used for systems with moderate damping.

By multiplying the equilibrium equation with Φ^T, (2.11) is transformed to

$$\underbrace{\Phi^T M\Phi}_{=:\widetilde{M}}\ddot{p} + \underbrace{\Phi^T C\Phi}_{=:\widetilde{C}}\dot{p} + \underbrace{\Phi^T K\Phi}_{=:\widetilde{K}}p = 0 \tag{2.15}$$

where Φ denotes the matrix with columns $\Phi_k, k = 1 \ldots N$. The matrices $\widetilde{M}, \widetilde{C}, \widetilde{K}$ are called the modal mass, damping and stiffness matrices, respectively.

The eigenmodes of the system form a basis of the deformation space. Therefore, a transformation into the full eigenmode basis does not yield a reduction of computational effort compared to a full Finite Element Analysis. The reduction achieved with this method consists in omitting non-relevant eigenmodes. These are determined due to symmetry or by their corresponding frequencies.
The process done in practice relies on some previous system knowledge. The external excitation of the flexible structure can be estimated from experience or from a previous MBS run done with rigid components. Very high frequencies in the flexible body are unlikely to be excited, as they require more energy. Thus, usually all eigenfrequencies that significantly exceed the excitation frequency are dropped.

The Fourier approach can also be inserted directly into the weak equation of the continuous flexible component 1.11. In practice, however, a preceding Finite Element Analysis is necessary to provide the mass and stiffness matrices for the MBS software.

In addition to the linear eigenmodes of the homogeneous system, eigenmode bases contain modes $\underline{\Phi}$ that account for the boundary conditions. They are computed by applying a *static unit load* at attachment points of the structure, i.e., in positions and directions where the flexible body is connected to other parts of the MBS system.

$$K\underline{\Phi}_j = f_j, \quad j = 1, \ldots, L \tag{2.16}$$

where L is the number of connected degrees of freedom, f_j is a unit force vector which is 1 in the position of the degree of freedom j and 0 everywhere else. The resulting modes $\underline{\Phi}_j$ are called *attachment modes*.

If a *static unit displacement* is applied to one attachment point on the structure, the corresponding deformation is called *constraint mode*.

Depending on the computation of the eigenmodes from 2.11, the varying boundary conditions need to be taken into account. Due to the fact that - in MBS - the boundaries are able to move and yet carry loads, both free or fixed boundary assumptions in 2.11 are technically wrong.

Therefore, the Craig-Bampton method and related approaches employ fixed-interface normal modes and constraint modes. Examples for using free-interface normal modes together with attachment modes are MacNeal's method and Rubin's method. For an overview of different methods and the corresponding mode sets, see [12].

In the method used in the software *SimPack*, an extended version of the static attachment modes is included in the set. The *frequency response modes* are dynamic deformation states excited by a unit load with a given frequency.

$$\left(K - \Omega_j^2 M\right) \underline{\Phi}_j = f_j, \quad j = 1, \ldots, L \tag{2.17}$$

with a chosen excitation frequency Ω_j. For more details, see [15].

The methods based on linear eigenmode analysis are designed for rather stiff members, where the linearisation approach is justified. Therefore, the applicability on nonlinear FEM models like tyres is strongly restricted.

2.2 The Method of Proper Orthogonal Decomposition

Whereas the beforementioned reduction methods are only suitable for linear problems, the method of *proper orthogonal decomposition (POD)* can also be applied to nonlinear systems. Its flexibility in application is based on analysing a given data set to provide the reduced model.

In the form introduced by Karhunen, the method of POD provides a subspace to approximate large data sets. The idea to use the method for model reduction originates from fluid dynamic applications including turbulence and coherent structures [7].

Transport problems still serve as the basis for a large part of the POD utilisation. Its application include glass melting processes ([5]) and heat transfer for semiconductor wafers ([35]). The method has also proved useful for certain problems in control theory ([22]) and in circuit simulations ([46]).

Willcox et al. ([49], [48], [6], [3]) presented several ideas to extend the POD method by combination with balanced truncation and for the use with incomplete data sets.

In the field of structural mechanics, Meyer ([33]) applied the POD method to geometrically nonlinear beams for the modelling of wind turbines with stochastic wind excitations. Furthermore, he used a dual-weighted residual method to find the appropriate subspace which optimises a functional. The structure preserving property of POD was examined by Krysl, Lall and Marsden ([25], [28]) with application to a docking device and by Sorensen and Antoulas ([43]) with an extension of the Gramian computation in the frequency domain.

To justify the method mathematically, Kunisch and Volkwein ([26], [27]) proved error bounds for POD-Galerkin approximations of parabolic equations. Among other things, they introduced a new POD approach incorporating derivative approximations and proved its superiority over the classical method both theoretically and numerically. In [22], Hinze and Volkwein extended this analysis to optimal control problems using pod-surrogate models. Rathinam and Petzold ([37]) performed a sensitivity analysis and applied the method to nonlinear problems.

2.2.1 Construction of a Subspace for Data Reduction

The method of POD can be regarded as an approach to approximate a given data set with a low dimensional subspace.

Let V be a Hilbert space with $dim V = N$ and let $\mathcal{Y} = \{y_1, \dots, y_m\}$ be a data set $\subset V$ with $rank(\mathcal{Y}) = d$. Furthermore, let $V^l = span\{\varphi_1 \dots \varphi_l\}$ be an l-dimensional subspace, $l \leq d, N$, with orthonormal basis $\{\varphi_i\}_{i=1,\dots,l}$. Then the projection error of the data set onto the subspace is given by

$$PE(\mathcal{Y}, V^l) = \frac{1}{m} \sum_{k=1}^{m} \left\| y_k - \sum_{j=1}^{l} \langle y_k, \varphi_j \rangle_X \cdot \varphi_j \right\|_X^2 \tag{2.18}$$

where $\langle \cdot, \cdot \rangle_X$ denotes a scalar product in V.
In the following, the projection of y_k onto the subspace V^l is denoted by

$$P^l(y_k) = \sum_{j=1}^{l} \langle y_k, \varphi_j \rangle_X \cdot \varphi_j \tag{2.19}$$

The crucial idea behind the POD method is the construction of the subspace based on data.

In brief, we search for the basis $\{\varphi_i\}_{i=1,\dots,l}$ of given dimension l which minimises the projection error (2.18). Then, the resulting subspace V^l can be seen as the best approximation to the set $\mathcal{Y}$ in a least squares sense.

Mathematically, this yields the following constrained optimisation problem:

$$\begin{aligned} &\min J(\varphi_1,\dots,\varphi_l) && \text{over} \quad \varphi_i \in V \\ \text{with} \quad &J(\varphi_1,\dots,\varphi_l) &&= PE(\mathcal{Y}, span\{\varphi_1,\dots,\varphi_l\}) \\ \text{subject to} \quad &\langle \varphi_i, \varphi_j \rangle &&= \delta_{ij} \end{aligned}$$

Setting up the *Lagrange* functional and the respective *Karush-Kuhn-Tucker* equations for the system, we get ([45]):

$$\begin{aligned} L(\varphi_1,\dots,\varphi_l,\lambda_{11},\dots,\lambda_{ll}) \quad &= \quad J(\varphi_1,\dots,\varphi_l) + \sum_{i,j=1}^{l} \lambda_{ij} \left(\langle \varphi_i, \varphi_j \rangle - \delta_{ij} \right) \\ \frac{\partial L}{\partial \varphi_i} = 0 \quad &\Longleftrightarrow \quad \sum_{j=1}^{m} y_j \langle y_j, \varphi_i \rangle = \lambda_{ii} \varphi_i \\ &\qquad\qquad \text{and} \quad \lambda_{ij} = 0 \quad \text{for} \quad i \neq j \\ \frac{\partial L}{\partial \lambda_{ij}} = 0 \quad &\Longleftrightarrow \quad \langle \varphi_i, \varphi_j \rangle = \delta_{ij} \end{aligned}$$

Setting $\lambda_i = \lambda_{ii}$, we identify the vectors $\varphi_1,\dots,\varphi_l$ as the solution of the following *eigenvalue problem*:

$$\sum_{j=1}^{m} y_j \langle y_j, \varphi_i \rangle = \lambda_i \varphi_i \quad \text{for} \quad i = 1,\dots,l. \tag{2.20}$$

This problem possesses $d = rank(\mathcal{Y})$ solutions $\lambda_1,\dots,\lambda_d$.

Combining 2.20 and 2.18, we find [45]:

$$PE(\mathcal{Y}, \{\varphi_1,\dots,\varphi_l\}) = \frac{1}{m} \sum_{k=1}^{m} \left\| y_k - \sum_{j=1}^{l} \langle y_k, \varphi_j \rangle_X \cdot \varphi_j \right\|_X^2 = \sum_{i=l+1}^{d} \lambda_i \tag{2.21}$$

where $d < m$ denotes the dimension of the set $\mathcal{Y}$, i.e., $d = rank(\mathcal{Y})$.

Remark 1 *Equation 2.21 states that the projection error of the set $\mathcal{Y}$ onto the subspace $V^l = span\{\varphi_1 \dots \varphi_l\}$ can be expressed by the sum of eigenvalues λ_i corresponding to the eigenvectors $\varphi_i, i = l+1,\dots,d$, that are not included in the basis of V^l.*

Including the eigenvectors φ of the l largest eigenvalues into the basis, we obtain the subspace V^l with the smallest projection error of all possible l-dimensional subspaces in V for the set $\mathcal{Y}$ in a least squares sense. This choice of basis vectors is called the Karhunen-Loève *basis [39].*

2.2.2 The Snapshot POD Method for Model Reduction

If subspaces created by the POD method serve as the foundation for a Galerkin projection, the system can be used for the reduction of large models. In the following the procedure for the *snapshot POD method* ([42]) is described.

Let $y(t) \in V$ be defined as the solution of a dynamical system

$$\frac{\partial y(t)}{\partial t} = f\left(y(t), t\right), \quad t \in [0, T] \tag{2.22}$$

and $y_i = y(t_i)$ be snapshots of the solution at time instances $t_i, i = 1, \ldots, m$, $t_i \in [0, T]$. If the precision is sufficient for the corresponding application, these snapshots can also be taken from measurements or from computations with many degrees of freedom (e.g. large FEM models).

Defining the *snapshot matrix*

$$Y = [y_1, \ldots, y_m] \in V^m$$

the eigenvalue problem 2.20 to solve for the POD basis vectors $\varphi_1, \ldots, \varphi_l \in V$ can be written as

$$YY^*\varphi_i = \lambda_i \varphi_i \quad \text{for} \quad i = 1, \ldots, l \tag{2.23}$$

where Y^* denotes the transpose of Y.

Clearly, we get the same results from

$$Y^*Y v_i = \lambda_i v_i \quad \text{for} \quad i = 1, \ldots, l \quad \text{with} \quad \varphi_i = Y^* v_i. \tag{2.24}$$

Depending on the dimension of Y, we solve 2.23 if $N << m$ or 2.24 if $m << N$, where $N = dim V$.

The matrix $C = Y^*Y$ with $C_{ij} = \langle y_i, y_j \rangle$ is referred to as the *correlation matrix* of the snapshot set.

Using 2.24, each eigenvector v_k of the correlation matrix defines a basis vector φ_k of the POD subspace. Depending on the number of basis vectors used for the subspace $V^l = span\{\varphi_1, \ldots, \varphi_l\}$, the projection error for

$$P^l y := \sum_{j=1}^{l} \langle y, \varphi_j \rangle_X \cdot \varphi_j \tag{2.25}$$

is defined by:

$$\frac{1}{m} \sum_{k=1}^{m} \left\| y_k - \sum_{j=1}^{l} \langle y_k, \varphi_j \rangle_X \cdot \varphi_j \right\|_X^2 = \sum_{j=l+1}^{d} \lambda_j \tag{2.26}$$

where $d < n$ shall denote the dimension of the snapshot set Y and $l < d$ the number of POD basis vectors used for the projection.

Applying the subspace projection to 2.22, we get the reduced surrogate model

$$\frac{\partial \alpha(t)}{\partial t} = \Phi f\left(\Phi^* \alpha(t), t\right), \quad t \in [0, T]. \tag{2.27}$$

where Φ denotes the POD projection matrix with columns $\varphi_k, k = 1 \dots l$.

For linear second order equations like 1.16, we get (similar to 2.15)

$$\underbrace{\Phi^T M \Phi}_{=:\widetilde{M}} \ddot{p} + \underbrace{\Phi^T C \Phi}_{=:\widetilde{C}} \dot{p} + \underbrace{\Phi^T K \Phi}_{=:\widetilde{K}} p = \Phi^T \beta \tag{2.28}$$

Note that the POD-reduced matrices $\widetilde{M}, \widetilde{C}, \widetilde{K}$ are, in general, fully populated. Thus, the POD method transforms a large sparse system 1.16 into a small dense system.

Remark 2 *Note that the POD-reduced system is not an approximation of the original system itself, but of the system and its external excitation. Due to the subspace construction which is based on the snapshot data set, the reduction scheme depends on the previously computed setup. In the strict sense, only the computed solutions can be properly represented by the reduced system.*
In addition to the overall computation setup, the position of snapshots within the time-span is an important issue. Especially at time instances when the dynamics of the system are changing rapidly, the sampling rate for snapshots shall be increased.

States and phenomena not represented by the snapshot set can not be represented by the reduced system as well.

The procedure of snapshot computation depends on the model's later purpose of use and therefore requires thorough considerations.

Chapter 3

Convergence Analysis of POD for Linear Second Order Equations

This chapter investigates the POD discretisation method for linear second order evolution equations. We present error estimates for two different choices of snapshot sets, one consisting of solution snapshots only and one consisting of solution snapshots and their derivatives up to second order. We show that the results of [26] and [27] for parabolic equations can be extended to a second order PDE and that the derivative snapshot POD method behaves better than the classical method for small time steps. Numerical comparison of the different approaches are presented underlining the theoretical results.

3.1 POD for linear second order evolution equations

The linear wave equation is a simple example for a partial differential equation of second order. In this section, we want to outline the mathematical framework required to handle such problems. Furthermore, we describe the discretisation by Newmark's method and the POD scheme.

3.1.1 Problem Description

Let V and H be real, separable Hilbert spaces and $a : V \times V \to \mathbb{R}$ be a continuous, coercive and symmetric bilinear form.

Following [14] and [26] we require

$$V \hookrightarrow H = H' \hookrightarrow V'$$

where V' denotes the dual of V. Each embedding is assumed to be dense and continuous.

There exist constants $\alpha, \beta, \kappa \geq 0$ such that

$$\|\phi\|_H^2 \leq \alpha \|\phi\|_V^2, \tag{3.1}$$

$$\|a(\phi, \psi)\| \leq \beta \|\phi\|_V \|\psi\|_V, \tag{3.2}$$

$$\kappa \|\phi\|_V^2 \leq a(\phi, \phi), \tag{3.3}$$

for all $\phi, \psi \in V$.

As a simple example for a second order evolution equation, we chose the damped linear wave equation expressed in weak formulation:

$$\langle \ddot{x}(t), \phi \rangle_H + D\langle \dot{x}(t), \phi \rangle_H + a(x(t), \phi) = \langle f(t), \phi \rangle_H \tag{3.4a}$$

$$\text{for all } \phi \in V \text{ and } t \in [0, T],$$

$$\langle x(0), \psi \rangle = \langle x_0, \psi \rangle_H \quad \text{for all } \psi \in H, \tag{3.4b}$$

$$\langle \dot{x}(0), \psi \rangle = \langle \dot{x}_0, \psi \rangle_H \quad \text{for all } \psi \in H, \tag{3.4c}$$

where $f(t) \in H$ is a given external force and $x(t) \in V$ denotes the sought deformation over time $t \in [0, T]$. Note that a damping term is incorporated which corresponds to a Rayleigh-type damping matrix C which is D times the mass matrix.

Lemma 1 *For $f \in L^2((0, T); H)$ and $x_0, \dot{x}_0 \in H$, problem 3.4 admits a unique weak solution $x \in V$.*

For the proof of existence and uniqueness of the solution the reader is referred to [13], XVII B, §1 and p417ff.

3.1.2 POD-Newmark scheme

For the time discretisation of (3.4) we divide the time interval $[0, T]$ into m subintervals of equal size $\Delta t = T/m$ and use Newmark's structure mechanics time integration scheme [14], i.e. we seek a sequence $(X_k) \subset V$, $k = 0, \ldots, m$, satisfying the following equations at each time level $t_k = k \cdot \Delta t$:

$$\langle \partial\partial X_k, \phi \rangle_H + D\langle \partial X_k, \phi \rangle_H + a(X_k, \phi) = \langle f(t_k), \phi \rangle_H \tag{3.5a}$$

$$\text{for all } \phi \in V \text{ and } k = 1, \ldots m,$$

$$\langle X_0, \psi \rangle = \langle x_0, \psi \rangle_H \quad \text{for all } \psi \in V, \tag{3.5b}$$

$$\langle \partial X_0, \psi \rangle = \langle \partial x_0, \psi \rangle_H \quad \text{for all } \psi \in V. \tag{3.5c}$$

Here we use the derivative approximations

$$\partial X_{k+1} = \frac{2}{\Delta t} X_{k+1} - \frac{2}{\Delta t} X_k - \partial X_k, \tag{3.6a}$$

$$\partial\partial X_{k+1} = \frac{4}{\Delta t^2} X_{k+1} - \frac{4}{\Delta t^2} X_k - \frac{4}{\Delta t} \partial X_k - \partial\partial X_k, \tag{3.6b}$$

for $k = 1, \ldots, m$.

Remark 3 *The case $k = 0$ is covered by the initial conditions x_0 and ∂x_0 for deformation X_0 and velocity ∂X_0, which yield the acceleration $\partial\partial X_0$ by solution of the equilibrium equation.*

Setting $V^l = span\{\varphi_1, \ldots, \varphi_l\} \subset V$, the POD-Newmark scheme for the wave equation consists in finding a sequence $\{X_k\}_{k=0,\ldots,m} \subset V^l$ which satisfies

$$\begin{aligned} \langle \partial\partial X_k, \phi\rangle_H + D\langle \partial X_k, \phi\rangle_H + a\,(X_k, \phi) &= \langle f(t_k), \phi\rangle_H && \text{(3.7a)}\\ &\quad \text{for all } \phi \in V^l \text{ and } k = 1, \ldots, m, \\ \langle X_0, \psi\rangle &= \langle x_0, \psi\rangle_H \quad \text{for all } \psi \in V^l, && \text{(3.7b)}\\ \langle \partial X_0, \psi\rangle &= \langle \partial x_0, \psi\rangle_H \quad \text{for all } \psi \in V^l. && \text{(3.7c)} \end{aligned}$$

Lemma 2 *Under the above assumptions there exists a unique solution $X_k \in V^l$ to problem (3.7) for each time level k (c.f. [13].)*

In this chapter we consider POD subspaces built from two different snapshot sets: set I consisting of deformation snapshots $\{x(t_k)\}$ at all time instances, and set II consisting of deformations and derivative approximations $\{x(t_k), \partial x(t_k), \partial\partial x(t_k)\}$. These sets yield the snapshot matrices Y_I and Y_{II} defined by:

$$\begin{aligned} Y_I &= [x(t_0), \ldots, x(t_m)] \quad \text{and} && \text{(3.8)}\\ Y_{II} &= [\underbrace{x(t_0), \ldots, x(t_m)}_{m+1}, \underbrace{\partial x(t_1), \ldots, \partial x(t_m)}_{m}, \underbrace{\partial\partial x(t_1), \ldots, \partial\partial x(t_{m-1})}_{m-1}] && \text{(3.9)} \end{aligned}$$

Note that the derivative approximations $\partial x(t_k)$ and $\partial\partial x(t_k)$ are elements of the space V. Furthermore, their inclusion does not change the dimension of the snapshot set, since they can be expressed as linear combinations of the deformation snapshots

$$\begin{aligned} \partial X_{k+1} + \partial X_k &= \frac{2}{\Delta t}\,(X_{k+1} - X_k)\,, \\ \partial\partial X_{k+1} + 2\partial\partial X_k + \partial\partial X_{k-1} &= \frac{4}{\Delta t^2}\,(X_{k+1} - 2X_k + X_{k-1})\,. \end{aligned}$$

We write $Y_{I,II} = [y_0, \ldots, y_d]$ with either $d = m$ or $d = 3m - 1$. In both cases, we follow the regular POD recipe (see [45]) by constructing the correlation matrix C from scalar products of the snapshots y_i

$$\begin{aligned} C_{ij} &= \langle y_i, y_j\rangle_X, \quad i, j = 1, \ldots, d, \\ \text{solving the eigenvalue problem} \quad Cv^k &= \lambda_k v^k, \\ \text{and defining the POD basis vectors by} \quad \varphi_k &= Y \cdot v^k, \end{aligned}$$

with $X = V$ or $X = H$.

3.2 Error Estimates

The error of the POD-Newmark scheme is defined as the difference between the numerical solution X of (3.7) and the analytical solution x of (3.4). Our goal consists in proving a bound for the H-Norm of the solution difference.

Theorem 1 *Let $x(t)$ be the smooth solution of (3.4) and $X_k, k = 1, \dots, m$, be the solution of (3.7) on each time level t_k. Let the POD subspace V^l be constructed from snapshot set Y_I or Y_{II},respectively. Then there exist constants C_I and C_{II} depending on T, D, $x^{(3)}$ and $x^{(4)}$, but not on Δt, m or l, such that it holds that:*

$$\begin{aligned}
&\frac{1}{m}\sum_{k=1}^{m}\left\|X^k - x(t_k)\right\|_H^2 \leq \\
&\leq\ C_I\left(\left\|X^0 - P^l x(t_0)\right\|_H^2 + \left\|X^1 - P^l x(t_1)\right\|_H^2 + \Delta t\left\|\partial X^0 - P^l\dot{x}(t_0)\right\|_H^2\right. \\
&\qquad \left. + \Delta t\left\|\partial X^1 - P^l\dot{x}(t_1)\right\|_H^2 + \Delta t^4 + \left(\frac{1}{\Delta t^4} + \frac{1}{\Delta t} + 1\right)\sum_{j=l+1}^{d}\lambda_{Ij}\right) \qquad (3.10)
\end{aligned}$$

for snapshots constructed via Y_I and

$$\begin{aligned}
&\frac{1}{m}\sum_{k=1}^{m}\left\|X^k - x(t_k)\right\|_H^2 \leq \\
&\leq\ C_{II}\left(\left\|X^0 - P^l x(t_0)\right\|_H^2 + \left\|X^1 - P^l x(t_1)\right\|_H^2 + \Delta t\left\|\partial X^0 - P^l\dot{x}(t_0)\right\|_H^2\right. \\
&\qquad \left. + \Delta t\left\|\partial X^1 - P^l\dot{x}(t_1)\right\|_H^2 + \Delta t^4 + \sum_{j=l+1}^{d}\lambda_{IIj}\right) \qquad (3.11)
\end{aligned}$$

for snapshots constructed via Y_{II} and

The estimates are constructed in a similar way as the ones given in [26] and [27].

In analogy to [26], Lemma 2, we state:

Lemma 3 *For all $x \in V$ it holds*

$$\begin{aligned}
\|x\|_H &\leq \sqrt{\|M\|_2 \cdot \|K^{-1}\|_2}\cdot\|x\|_V \text{ for all } x \in V^l, \qquad (3.12)\\
\|x\|_V &\leq \sqrt{\|K\|_2 \cdot \|M^{-1}\|_2}\cdot\|x\|_H \text{ for all } x \in V^l, \qquad (3.13)\\
\text{with} &\quad M_{ij} = \langle\Phi_i, \Phi_j\rangle_H, \quad K_{ij} = \langle\Phi_i, \Phi_j\rangle_V,
\end{aligned}$$

where $\|\cdot\|_2$ denotes the spectral norm for symmetric matrices. M and K are called the system's mass and stiffness matrix respectively.

Remark 4 *These inequalities allow us to set up an error estimate in the H-norm and also to control the error in the V-norm as long as $\|M\|_2$, $\|M^{-1}\|_2$, $\|K\|_2$ and $\|K^{-1}\|_2$ are finite. Hence, we restrict ourselves to the H-norm in the following.*

Let X^k be the solution of the POD system (3.7) for the time instances $t_k = k \cdot \Delta t, k = 0, \ldots, m$, and $x(t_k)$ be the corresponding solution of the original system (3.5). In order to estimate

$$\frac{1}{m} \sum_{k=1}^{m} \left\| X^k - x(t_k) \right\|_H^2 \tag{3.14}$$

we decompose the local error into a projection part ρ and a part ϑ arising from the numerical discretisation procedure:

$$X^k - x(t_k) = \underbrace{X^k - P^l x(t_k)}_{\vartheta_k} + \underbrace{P^l x(t_k) - x(t_k)}_{\rho_k}, \tag{3.15}$$

which yields

$$\frac{1}{m} \sum_{k=1}^{m} \left\| X^k - x(t_k) \right\|_H^2 \leq \frac{2}{m} \sum_{k=1}^{m} \|\vartheta_k\|_H^2 + \frac{2}{m} \sum_{k=1}^{m} \|\rho_k\|_H^2 . \tag{3.16}$$

For an estimate of $\|\rho_k\|_H^2$ we use the error bound (2.26). Case I is constructed in the "classical" way and simply yields the POD projection error ([45])

$$\frac{1}{m+1} \sum_{k=0}^{m} \left\| x_k - \sum_{j=1}^{l} \langle x_k, \phi_j \rangle_X \cdot \phi_j \right\|_X^2 = \sum_{j=l+1}^{d} \lambda_{Ij}. \tag{3.17}$$

Here, x_k denotes the snapshot $x(t_k)$.

For later use we derive

$$\begin{aligned}
&\sum_{k=1}^{m-1} \left\| \partial \left(x_{k+1} + 2x_k + x_{k-1} \right) - \sum_{j=1}^{l} \langle \partial \left(x_{k+1} + 2x_k + x_{k-1} \right), \phi_j \rangle_X \cdot \phi_j \right\|_X^2 = \\
&= \sum_{k=1}^{m-1} \frac{1}{\Delta t^4} \left\| x_{k+1} - 2x_k + x_{k-1} - P^l x_{k+1} + 2P^l x_k - P^l x_{k-1} \right\|_X^2 \\
&\leq \frac{4}{\Delta t^2} \sum_{k=1}^{m-1} 2 \Big(\left\| x_{k+1} - P^l x_{k+1} \right\|_X^2 + \left\| x_{k-1} - P^l x_{k-1} \right\|_X^2 \Big) \\
&\leq \frac{16}{\Delta t^2} \sum_{k=0}^{m} \left\| x_k - P^l x_k \right\|_X^2 \\
&\leq \frac{16}{\Delta t^2} (m+1) \sum_{j=l+1}^{d} \lambda_{Ij}
\end{aligned} \tag{3.18}$$

and

$$\begin{aligned}
&\sum_{k=1}^{m-1} \left\| \partial\partial \left(x_{k+1} + 2x_k + x_{k-1}\right) - \sum_{j=1}^{l} \langle \partial\partial \left(x_{k+1} + 2x_k + x_{k-1}\right), \phi_j \rangle_X \cdot \phi_j \right\|_X^2 = \\
&= \sum_{k=1}^{m-1} \frac{16}{\Delta t^4} \left\| x_{k+1} - 2x_k + x_{k-1} - P^l x_{k+1} + 2P^l x_k - P^l x_{k-1} \right\|_X^2 \\
&\leq \frac{16}{\Delta t^4} \sum_{k=1}^{m-1} 4 \Big(\left\| x_{k+1} - P^l x_{k+1} \right\|_X^2 + \left\| 2x_k - 2P^l x_k \right\|_X^2 + \left\| x_{k-1} - P^l x_{k-1} \right\|_X^2 \Big) \\
&\leq \frac{384}{\Delta t^4} \sum_{k=0}^{m} \left\| x_k - P^l x_k \right\|_X^2 \\
&\leq \frac{384}{\Delta t^4} (m+1) \sum_{j=l+1}^{d} \lambda_{Ij}. \qquad (3.19)
\end{aligned}$$

For the second case, the POD error is analogously defined for the sum over all snapshots (solution x and derivatives ∂x and $\partial\partial x$):

$$\begin{aligned}
&\frac{1}{3m} \sum_{k=0}^{m} \left\| x_k - \sum_{j=1}^{l} \langle x_k, \phi_j \rangle_X \cdot \phi_j \right\|_X^2 + \\
&+ \frac{1}{3m} \sum_{k=1}^{m} \left\| \partial x_k - \sum_{j=1}^{l} \langle \partial x_k, \phi_j \rangle_X \cdot \phi_j \right\|_X^2 + \\
&+ \frac{1}{3m} \sum_{k=1}^{m-1} \left\| \partial\partial x_k - \sum_{j=1}^{l} \langle \partial\partial x_k, \phi_j \rangle_X \cdot \phi_j \right\|_X^2 = \sum_{j=l+1}^{d} \lambda_{IIj}, \qquad (3.20)
\end{aligned}$$

which yields

$$\begin{aligned}
\frac{1}{m} \sum_{k=1}^{m} \|\rho_k\|_X^2 = \frac{1}{m} \sum_{k=1}^{m} \left\| x_k - \sum_{j=1}^{l} \langle x_k, \phi_j \rangle_X \cdot \phi_j \right\|_X^2 &\leq 3 \cdot \sum_{j=l+1}^{d} \lambda_{IIj}, \\
\frac{1}{m} \sum_{k=1}^{m} \left\| \partial x_k - \sum_{j=1}^{l} \langle \partial x_k, \phi_j \rangle_X \cdot \phi_j \right\|_X^2 &\leq 3 \cdot \sum_{j=l+1}^{d} \lambda_{IIj}, \\
\frac{1}{m} \sum_{k=1}^{m-1} \left\| \partial\partial x_k - \sum_{j=1}^{l} \langle \partial\partial x_k, \phi_j \rangle_X \cdot \phi_j \right\|_X^2 &\leq 3 \cdot \sum_{j=l+1}^{d} \lambda_{IIj}
\end{aligned}$$

and

$$\sum_{k=1}^{m-1} \left\| \partial \left(x_{k+1} + 2x_k + x_{k-1} \right) - \sum_{j=1}^{l} \langle \partial \left(x_{k+1} + 2x_k + x_{k-1} \right), \phi_j \rangle_X \cdot \phi_j \right\|_X^2$$

$$\leq \; 24 \sum_{k=1}^{m-1} \left\| \partial x_k - \sum_{j=1}^{l} \langle \partial x_k, \phi_j \rangle_X \cdot \phi_j \right\|_X^2$$

$$\leq \; 72m \cdot \sum_{j=l+1}^{d} \lambda_{IIj}, \tag{3.21}$$

and an analogous estimate for the second derivatives

$$\sum_{k=1}^{m-1} \left\| \partial\partial \left(x_{k+1} + 2x_k + x_{k-1} \right) - \sum_{j=1}^{l} \langle \partial\partial \left(x_{k+1} + 2x_k + x_{k-1} \right), \phi_j \rangle_X \cdot \phi_j \right\|_X^2$$

$$\leq \; 24 \sum_{k=1}^{m-1} \left\| \partial\partial x_k - \sum_{j=1}^{l} \langle \partial\partial x_k, \phi_j \rangle_X \cdot \phi_j \right\|_X^2$$

$$\leq \; 72m \cdot \sum_{j=l+1}^{d} \lambda_{IIj}. \tag{3.22}$$

Remark 5 *Note that the eigenvalues λ_{Ij} and λ_{IIj} are not identical. The weighting of snapshots is changed by inclusion of the derivative approximations, which leads to different choices of basis vectors for the subspaces V_I^l and V_{II}^l. In both cases, the snapshot correlation matrix C is generally not invertible, so the sum of the eigenvalues remains finite.*

For an estimate of $\|\vartheta_k\|_H^2 = \left\|X^k - P^l x(t_k)\right\|_H^2$ we set up the following equation:

$$\begin{aligned}
&\langle \partial\partial\vartheta_k, \psi \rangle_H + D\langle \partial\vartheta_k, \psi \rangle_H + a\left(\vartheta_k, \psi\right) = \\
&= \; \langle \partial\partial X_k, \psi \rangle_H - \langle \partial\partial P^l x(t_k), \psi \rangle_H + D \cdot \langle \partial X_k, \psi \rangle_H - D \cdot \langle \partial P^l x(t_k), \psi \rangle_H + \\
&\quad + a\left(X_k, \psi\right) - a\left(P^l x(t_k), \psi\right) \\
&= \; \langle f(t_k), \psi \rangle_H - a\left(P^l x(t_k), \psi\right) - \langle \partial\partial P^l x(t_k), \psi \rangle_H - D\langle \partial P^l x(t_k), \psi \rangle_H \\
&= \; \langle f(t_k), \psi \rangle_H - a\left(x(t_k), \psi\right) - \langle \partial\partial P^l x(t_k), \psi \rangle_H - D\langle \partial P^l x(t_k), \psi \rangle_H \\
&= \; \langle \ddot{x}(t_k), \psi \rangle_H + D\langle \dot{x}(t_k), \psi \rangle_H - \langle \partial\partial P^l x(t_k), \psi \rangle_H \\
&= \; \langle \left(\ddot{x}(t_k) - \partial\partial P^l x(t_k)\right) + D\left(\dot{x}(t_k) - \partial P^l x(t_k)\right), \psi \rangle_H \\
&=: \; \langle v_k, \psi \rangle_H,
\end{aligned} \tag{3.23}$$

which holds for all $\psi \in V^l$.

I.e., that the sequence ϑ_k can be regarded as the solution of a linear, damped wave equation with the "force term" v_k. In analogy to the centered scheme described in [14], the Newmark scheme for this equation can be written as:

$$\frac{1}{\Delta t^2}\langle \vartheta_{k+1} - 2\vartheta_k + \vartheta_{k-1}, \psi\rangle_H + \frac{2D}{\Delta t}\langle \vartheta_{k+1} - \vartheta_{k-1}, \psi\rangle_H + \\ + \frac{1}{4}a\left(\vartheta_{k+1} + 2\vartheta_k + \vartheta_{k-1}, \psi\right) = \frac{1}{4}\langle v_{k+1} + 2v_k + v_{k-1}, \psi\rangle_H$$

For notational convenience we define $\gamma_k = v_{k+1}+2v_k+v_{k-1}$. Choosing $\psi = \vartheta_{k+1}-\vartheta_{k-1} \in V^l$ as a test function in (3.23) we get

$$\underbrace{\frac{1}{\Delta t^2}\langle \vartheta_{k+1} - 2\vartheta_k + \vartheta_{k-1}, \vartheta_{k+1} - \vartheta_{k-1}\rangle_H}_{(i)} + \underbrace{\frac{2D}{\Delta t}\left\|\vartheta_{k+1} - \vartheta_{k-1}\right\|_H^2}_{=:d^k \le 0} + \\ + \underbrace{\frac{1}{4}a\left(\vartheta_{k+1} + 2\vartheta_k + \vartheta_{k-1}, \vartheta_{k+1} - \vartheta_{k-1}\right)}_{(ii)} = \underbrace{\frac{1}{4}\langle \gamma_k, \vartheta_{k+1} - \vartheta_{k-1}\rangle_H}_{=:W^k}$$

with term (i):

$$\begin{aligned}
&\frac{1}{\Delta t^2}\langle \vartheta_{k+1} - 2\vartheta_k + \vartheta_{k-1}, \vartheta_{k+1} - \vartheta_{k-1}\rangle_H = \\
&= \frac{1}{\Delta t^2}\langle (\vartheta_{k+1} - \vartheta_k) - (\vartheta_k - \vartheta_{k-1}), (\vartheta_{k+1} - \vartheta_k) + (\vartheta_k - \vartheta_{k-1})\rangle_H = \\
&= \frac{1}{\Delta t^2}\left(\|\vartheta_{k+1} - \vartheta_k\|_H^2 - \|\vartheta_k - \vartheta_{k-1}\|_H^2\right)
\end{aligned}$$

and term (ii):

$$\begin{aligned}
&\frac{1}{4}a\left(\vartheta_{k+1} + 2\vartheta_k + \vartheta_{k-1}, \vartheta_{k+1} - \vartheta_{k-1}\right) = \\
&= \frac{1}{4}a\left((\vartheta_{k+1} + \vartheta_k) + (\vartheta_k + \vartheta_{k-1}), (\vartheta_{k+1} + \vartheta_k) - (\vartheta_k + \vartheta_{k-1})\right) \\
&= \frac{1}{4}\left[a\left(\vartheta_{k+1} + \vartheta_k, \vartheta_{k+1} + \vartheta_k\right) - a\left(\vartheta_k + \vartheta_{k-1}, \vartheta_k + \vartheta_{k-1}\right)\right].
\end{aligned}$$

This yields

$$\begin{aligned}
E^{k+1} + d^k &= E^k + W^k \\
\text{with} \quad E^{k+1} &:= \left\|\frac{\vartheta_{k+1} - \vartheta_k}{\Delta t}\right\|_H^2 + \frac{1}{4}a\left(\vartheta_{k+1} + \vartheta_k, \vartheta_{k+1} + \vartheta_k\right).
\end{aligned}$$

Due to the coercivity of the bilinear form a it holds that

$$\left\|\frac{\vartheta_{k+1} - \vartheta_k}{\Delta t}\right\|_H^2 \le E^{k+1} \tag{3.24}$$

and

$$\begin{aligned}
E^{k+1} + d^k &= E^1 + \sum_{i=1}^{k} W^i = E^1 + \frac{1}{4}\sum_{i=1}^{k}\langle \gamma_i, \vartheta_{i+1} - \vartheta_{i-1}\rangle_H \\
&= E^1 + \frac{1}{4}\sum_{i=1}^{k}\langle \gamma_i, (\vartheta_{i+1} - \vartheta_i) + (\vartheta_i - \vartheta_{i-1})\rangle_H \\
&= E^1 + \frac{1}{4}\sum_{i=1}^{k}\langle \gamma_i, \vartheta_{i+1} - \vartheta_i\rangle_H + \frac{1}{4}\sum_{i=1}^{k}\langle \gamma_i, \vartheta_i - \vartheta_{i-1}\rangle_H \\
&= E^1 + \frac{1}{4}\left(\sum_{i=1}^{k-1}\langle \gamma_i, \vartheta_{i+1} - \vartheta_i\rangle_H + \langle \gamma_k, \vartheta_{k+1} - \vartheta_k\rangle_H\right) + \\
&\quad + \frac{1}{4}\left(\sum_{p=1}^{k-1}\langle \gamma_{p+1}, \vartheta_{p+1} - \vartheta_p\rangle_H + \langle \gamma_1, \vartheta_1 - \vartheta_0\rangle_H\right) \\
&= E^1 + \frac{1}{4}\langle \gamma_k, \vartheta_{k+1} - \vartheta_k\rangle_H + \frac{1}{4}\langle \gamma_1, \vartheta_1 - \vartheta_0\rangle_H + \\
&\quad + \frac{1}{4}\sum_{i=1}^{k-1}\langle \gamma_{i+1} + \gamma_i, \vartheta_{i+1} - \vartheta_i\rangle_H.
\end{aligned}$$

Using Young's inequality we get

$$\begin{aligned}
\left\|\frac{\vartheta_{k+1} - \vartheta_k}{\Delta t}\right\|_H^2 &\leq E^1 + \frac{\Delta t}{32}\|\gamma_1\|_H^2 + \frac{\Delta t}{2}\left\|\frac{\vartheta_1 - \vartheta_0}{\Delta t}\right\|_H^2 + \frac{\Delta t}{32}\|\gamma_k\|_H^2 + \frac{\Delta t}{2}\left\|\frac{\vartheta_{k+1} - \vartheta_k}{\Delta t}\right\|_H^2 + \\
&\quad + \sum_{i=1}^{k-1}\frac{\Delta t}{32}\|\gamma_{i+1} + \gamma_i\|_H^2 + \sum_{i=1}^{k-1}\frac{\Delta t}{2}\left\|\frac{\vartheta_{i+1} - \vartheta_i}{\Delta t}\right\|_H^2 \\
&\leq E^1 + \frac{\Delta t}{32}\|\gamma_1\|_H^2 + \frac{\Delta t}{2}\left\|\frac{\vartheta_1 - \vartheta_0}{\Delta t}\right\|_H^2 + \frac{\Delta t}{32}\|\gamma_k\|_H^2 + \frac{1}{2}\left\|\frac{\vartheta_{k+1} - \vartheta_k}{\Delta t}\right\|_H^2 + \\
&\quad + \sum_{i=1}^{k-1}\frac{\Delta t}{32}\|\gamma_{i+1} + \gamma_i\|_H^2 + \sum_{i=1}^{k-1}\frac{\Delta t}{2}\left\|\frac{\vartheta_{i+1} - \vartheta_i}{\Delta t}\right\|_H^2
\end{aligned}$$

This yields for $\Delta t \leq 1$

$$\begin{aligned}
\left\|\frac{\vartheta_{k+1} - \vartheta_k}{\Delta t}\right\|_H^2 &\leq 2\cdot E^1 + \frac{\Delta t}{16}\|\gamma_1\|_H^2 + \Delta t\left\|\frac{\vartheta_1 - \vartheta_0}{\Delta t}\right\|_H^2 + \frac{\Delta t}{16}\|\gamma_k\|_H^2 + \\
&\quad + \sum_{i=1}^{k-1}\frac{\Delta t}{16}\|\gamma_{i+1} + \gamma_i\|_H^2 + \sum_{i=1}^{k-1}\Delta t\left\|\frac{\vartheta_{i+1} - \vartheta_i}{\Delta t}\right\|_H^2 \\
&\leq 2\cdot E^1 + \Delta t\left\|\frac{\vartheta_1 - \vartheta_0}{\Delta t}\right\|_H^2 + \sum_{i=1}^{k}\frac{\Delta t}{4}\|\gamma_i\|_H^2 + \sum_{i=1}^{k-1}\Delta t\left\|\frac{\vartheta_{i+1} - \vartheta_i}{\Delta t}\right\|_H^2
\end{aligned}$$

We use the discrete Gronwall lemma [4]:

Theorem 2 *Discrete Gronwall's theorem*
Let a_k, b_k, c_k and x_k $(k = 1, 2, ...)$ be non-negative real numbers and let $r \geq 1$ be a real number. If

$$x_k \leq a_k + b_k \cdot \left[\sum_{i=1}^{k-1} c_i x_i \right]^{1/r} \qquad (k = 1, 2, ...) \tag{3.25}$$

then

$$\sum_{i=1}^{k} c_i x_i^r \leq \left[1 - (1 - p_k)^{1/r} \right]^{-r} \cdot \sum_{i=1}^{k} a_i^r c_i p_i \quad (k = 1, 2, ...) \tag{3.26}$$

where

$$p_i = \prod_{j=1}^{i} (1 + b_j^r c_j)^{-1} \quad (j = 1, 2, ...) \tag{3.27}$$

In our case, we set:

$$\begin{aligned}
x_k &= \left\| \frac{\vartheta_{k+1} - \vartheta_k}{\Delta t} \right\|_H^2 \\
a_k &= 2 \cdot E^1 + \left\| \frac{\vartheta_1 - \vartheta_0}{\Delta t} \right\|_H^2 + \sum_{i=1}^{k} \frac{\Delta t^2}{4} \left\| \gamma_i \right\|_H^2 \\
b_k &\equiv 1 \\
c_i &\equiv \Delta t \\
r &\equiv 1 \\
p_i &= \prod_{j=1}^{i} (1 + \Delta t)^{-1} = \left(1 + \frac{1}{\Delta t} \right)^{-i}
\end{aligned}$$

and using

$$(1 + \Delta t)^{k-i} \leq \left(1 + \frac{T}{m} \right)^{m-i} \leq \left(1 + \frac{T}{m} \right)^{m} \leq e^T$$

it yields:

$$\begin{aligned}
&\sum_{i=1}^{k} \Delta t \left\| \frac{\vartheta_{i+1} - \vartheta_i}{\Delta t} \right\|_H^2 \leq \\
&\leq \; (1+\Delta t)^k \sum_{i=1}^{k} (1+\Delta t)^{-i} \left(2 \cdot E^1 + \Delta t \left\| \frac{\vartheta_1 - \vartheta_0}{\Delta t} \right\|_H^2 + \sum_{j=1}^{i-1} \frac{\Delta t}{4} \left\| \gamma_j \right\|_H^2 \right) \\
&\leq \; e^T \cdot \left(2 \cdot E^1 + \Delta t \left\| \frac{\vartheta_1 - \vartheta_0}{\Delta t} \right\|_H^2 \right) + \sum_{i=1}^{k} \frac{\Delta t}{4} \left\| \gamma_i \right\|_H^2 + \sum_{i=2}^{k} \sum_{j=1}^{i-1} \frac{\Delta t}{4} \left\| \gamma_j \right\|_H^2 \\
&\leq \; e^T \cdot \left(2 \cdot E^1 + \Delta t \left\| \frac{\vartheta_1 - \vartheta_0}{\Delta t} \right\|_H^2 \right) + \sum_{i=1}^{k} \frac{\Delta t}{4} \left\| \gamma_i \right\|_H^2 + \frac{\Delta t}{4} \sum_{i=2}^{k} (k-i) \left\| \gamma_i \right\|_H^2 \\
&\leq \; e^T \cdot \left(2 \cdot E^1 + \Delta t \left\| \frac{\vartheta_1 - \vartheta_0}{\Delta t} \right\|_H^2 + \frac{\Delta t}{4} \sum_{i=1}^{k} \left\| \gamma_i \right\|_H^2 \right).
\end{aligned}$$

Therefore,

$$\Delta t \sum_{i=1}^{m} \left\| \frac{\vartheta_{i+1} - \vartheta_i}{\Delta t} \right\|_H^2 \leq e^T \cdot \left(2 \cdot E^1 + \Delta t \left\| \frac{\vartheta_1 - \vartheta_0}{\Delta t} \right\|_H^2 + \frac{\Delta t}{4} \sum_{i=1}^{m} \left\| \gamma_i \right\|_H^2 \right),$$

which depends only on the initial conditions ϑ_0 and $\dot{\vartheta}_0$ and on the sequence (γ_k). Further, we have

$$\begin{aligned}
\left\| \vartheta_{k+1} \right\|_H^2 &\leq \left\| \vartheta_1 + \sum_{i=1}^{k-1} \vartheta_{i+1} - \vartheta_i \right\|_H^2 \\
&\leq 2 \left\| \vartheta_1 \right\|_H^2 + 2 \left\| \sum_{i=1}^{k} \vartheta_{i+1} - \vartheta_i \right\|_H^2 \\
&\leq 2 \left\| \vartheta_1 \right\|_H^2 + 2k \sum_{i=1}^{k} \left\| \vartheta_{i+1} - \vartheta_i \right\|_H^2 \\
&\leq 2 \left\| \vartheta_1 \right\|_H^2 + 2m \sum_{i=1}^{m} \left\| \vartheta_{i+1} - \vartheta_i \right\|_H^2 \\
&\leq 2 \left\| \vartheta_1 \right\|_H^2 + 2m\Delta t \sum_{i=1}^{m} \left\| \frac{\vartheta_{i+1} - \vartheta_i}{\Delta t} \right\|_H^2 \\
&\leq 2 \left\| \vartheta_1 \right\|_H^2 + 2Te^T \left(2 \cdot E^1 + \Delta t \left\| \frac{\vartheta_1 - \vartheta_0}{\Delta t} \right\|_H^2 + \frac{\Delta t}{4} \sum_{i=1}^{m} \left\| \gamma_i \right\|_H^2 \right)
\end{aligned}$$

which yields the averaged sum

$$\frac{1}{m}\sum_{k=0}^{m}\|\vartheta_k\|_H^2 \;\leq\; 2\|\vartheta_1\|_H^2 + 2Te^T\left(2\cdot E^1 + \Delta t\left\|\frac{\vartheta_1-\vartheta_0}{\Delta t}\right\|_H^2 + \frac{\Delta t}{4}\sum_{i=1}^{m}\|\gamma_i\|_H^2\right).$$

In the following we construct a bound for the right hand side terms which are dominated by the sum over $\|\gamma_k\|_H^2$. Again, this sequence $\|\gamma_k\|_H^2 = \|v_{k+1}+2v_k+v_{k-1}\|_H^2$ is separated into two terms, a "projection" and a "discretisation" part:

$$\begin{aligned} v_k &= \ddot{x}(t_k) - \partial\partial P^l x(t_k) + D\left(\dot{x}(t_k) - \partial P^l x(t_k)\right) \\ &= \underbrace{\ddot{x}(t_k) - \partial\partial x(t_k)}_{=:w_k} + \underbrace{\partial\partial x(t_k) - \partial\partial P^l x(t_k)}_{=:z_k} + \\ &\quad + D\left(\underbrace{\dot{x}(t_k) - \partial x(t_k)}_{=:\widetilde{w}_k} + \underbrace{\partial x(t_k) - \partial P^l x(t_k)}_{=:\widetilde{z}_k}\right), \end{aligned}$$

yielding finally

$$\begin{aligned} \|\gamma_k\|_H^2 \;\leq\; & 4\|w_{k+1}+2w_k+w_{k-1}\|_H^2 + 4\|z_{k+1}+2z_k+z_{k-1}\|_H^2 + \\ & + 4\|\widetilde{w}_{k+1}+2\widetilde{w}_k+\widetilde{w}_{k-1}\|_H^2 + 4\|\widetilde{z}_{k+1}+2\widetilde{z}_k+\widetilde{z}_{k-1}\|_H^2. \end{aligned}$$

For w_k it holds that

$$\begin{aligned} & w_{k+1}+2w_k+w_{k-1} \\ &= \ddot{x}(t_{k+1}) + 2\ddot{x}(t_k) + \ddot{x}(t_{k-1}) - (\partial\partial x(t_{k+1}) + 2\partial\partial x(t_k) + \partial\partial x(t_{k-1})) \\ &= \ddot{x}(t_{k+1}) + 2\ddot{x}(t_k) + \ddot{x}(t_{k-1}) - \frac{4}{\Delta t^2}\left(x(t_{k+1}) - 2x(t_k) + x(t_{k-1})\right) \end{aligned}$$

and Taylor expansion yields

$$\begin{aligned} \ddot{x}(t_{k+1}) &= \ddot{x}(t_k) + (t_{k+1}-t_k)\cdot\dddot{x}(t_k) + \int_{t_k}^{t_{k+1}}(t_{k+1}-s)\cdot\ddddot{x}(s)ds \\ \ddot{x}(t_{k-1}) &= \ddot{x}(t_k) + (t_{k-1}-t_k)\cdot\dddot{x}(t_k) + \int_{t_k}^{t_{k-1}}(t_{k-1}-s)\cdot\ddddot{x}(s)ds \\ &= \ddot{x}(t_k) - (t_k-t_{k-1})\cdot\dddot{x}(t_k) + \int_{t_{k-1}}^{t_k}(s-t_{k-1})\cdot\ddddot{x}(s)ds \end{aligned}$$

and

$$\begin{aligned}
x(t_{k+1}) &= x(t_k) + (t_{k+1} - t_k) \cdot \dot{x}(t_k) + \frac{(t_{k+1} - t_k)^2}{2}\ddot{x}(t_k) + \\
&\quad + \frac{(t_{k+1} - t_k)^3}{6}\dddot{x}(t_k) + \int_{t_k}^{t_{k+1}} \frac{(t_{k+1} - s)^3}{6}\ddddot{x}(s)ds \\
x(t_{k-1}) &= x(t_k) + (t_{k-1} - t_k) \cdot \dot{x}(t_k) + \frac{(t_{k-1} - t_k)^2}{2}\ddot{x}(t_k) + \\
&\quad + \frac{(t_{k-1} - t_k)^3}{6}\dddot{x}(t_k) + \int_{t_k}^{t_{k-1}} \frac{(t_{k-1} - s)^3}{6}\ddddot{x}(s)ds \\
&= x(t_k) - (t_k - t_{k-1}) \cdot \dot{x}(t_k) + \frac{(t_k - t_{k-1})^2}{2}\ddot{x}(t_k) - \\
&\quad - \frac{(t_k - t_{k-1})^3}{6}\dddot{x}(t_k) + \int_{t_{k-1}}^{t_k} \frac{(s - t_{k-1})^3}{6}\ddddot{x}(s)ds
\end{aligned}$$

Summing up, considering the constant step size $\Delta t = t_{k+1} - t_k = t_k - t_{k-1}$:

$$\begin{aligned}
&\ddot{x}(t_{k+1}) + 2\ddot{x}(t_k) + \ddot{x}(t_{k-1}) \\
&= \ddot{x}(t_k) + \Delta t \cdot \dddot{x}(t_k) + \int_{t_k}^{t_{k+1}} (t_{k+1} - s) \cdot \ddddot{x}(s)ds + 2\ddot{x}(t_k) + \\
&\quad + \ddot{x}(t_k) - \Delta t \cdot \dddot{x}(t_k) + \int_{t_{k-1}}^{t_k} (s - t_{k-1}) \cdot \ddddot{x}(s)ds \\
&= 4\ddot{x}(t_k) + \int_{t_k}^{t_{k+1}} (t_{k+1} - s) \cdot \ddddot{x}(s)ds + \int_{t_{k-1}}^{t_k} (s - t_{k-1}) \cdot \ddddot{x}(s)ds \\
&x(t_{k+1}) - 2x(t_k) + x(t_{k-1}) \\
&= x(t_k) + \Delta t \cdot \dot{x}(t_k) + \frac{\Delta t^2}{2}\ddot{x}(t_k) + \frac{\Delta t^3}{6}\dddot{x}(t_k) + \int_{t_k}^{t_{k+1}} \frac{(t_{k+1} - s)^3}{6}\ddddot{x}(s)ds - 2x(t_k) \\
&\quad + x(t_k) - \Delta t \cdot \dot{x}(t_k) + \frac{\Delta t^2}{2}\ddot{x}(t_k) - \frac{\Delta t^3}{6}\dddot{x}(t_k) + \int_{t_{k-1}}^{t_k} \frac{(s - t_{k-1})^3}{6}\ddddot{x}(s)ds \\
&= \Delta t^2 \ddot{x}(t_k) + \int_{t_k}^{t_{k+1}} \frac{(t_{k+1} - s)^3}{6}\ddddot{x}(s)ds + \int_{t_{k-1}}^{t_k} \frac{(s - t_{k-1})^3}{6}\ddddot{x}(s)ds
\end{aligned}$$

leads to:

$$\begin{aligned} w_{k+1} + 2w_k + w_{k-1} &= \int_{t_k}^{t_{k+1}} (t_{k+1} - s) \cdot \dddot{\ddot{x}}(s) ds + \int_{t_{k-1}}^{t_k} (s - t_{k-1}) \cdot \dddot{\ddot{x}}(s) ds - \\ &\quad - \int_{t_k}^{t_{k+1}} \frac{4(t_{k+1} - s)^3}{6\Delta t^2} \dddot{\ddot{x}}(s) ds - \int_{t_{k-1}}^{t_k} \frac{4(s - t_{k-1})^3}{6\Delta t^2} \dddot{\ddot{x}}(s) ds \\ &= \int_{t_k}^{t_{k+1}} \dddot{\ddot{x}}(s) \cdot \left(t_{k+1} - s - \frac{2(t_{k+1} - s)^3}{3\Delta t^2} \right) ds \\ &\quad + \int_{t_{k-1}}^{t_k} \dddot{\ddot{x}}(s) \cdot \left(s - t_{k-1} - \frac{2(s - t_{k-1})^3}{3\Delta t^2} \right) ds \end{aligned}$$

From this representation an estimate for $\|w_{k+1} + 2w_k + w_{k-1}\|_H^2$ follows:

$$\begin{aligned} &\|w_{k+1} + 2w_k + w_{k-1}\|_H^2 \\ &\leq 2 \left(\int_{t_k}^{t_{k+1}} \dddot{\ddot{x}}(s) \cdot \left(t_{k+1} - s - \frac{2(t_{k+1} - s)^3}{3\Delta t^2} \right) ds \right)^2 + \\ &\quad + 2 \left(\int_{t_{k-1}}^{t_k} \dddot{\ddot{x}}(s) \cdot \left(s - t_{k-1} - \frac{2(s - t_{k-1})^3}{3\Delta t^2} \right) ds \right)^2 \\ &\leq 2 \int_{t_k}^{t_{k+1}} \dddot{\ddot{x}}(s)^2 ds \cdot \int_{t_k}^{t_{k+1}} \left(t_{k+1} - s - \frac{2(t_{k+1} - s)^3}{3\Delta t^2} \right)^2 ds + \\ &\quad + 2 \int_{t_{k-1}}^{t_k} \dddot{\ddot{x}}(s)^2 ds \cdot \int_{t_{k-1}}^{t_k} \left(s - t_{k-1} - \frac{2(s - t_{k-1})^3}{3\Delta t^2} \right)^2 ds \\ &= 2 \int_{t_k}^{t_{k+1}} \dddot{\ddot{x}}(s)^2 ds \cdot \int_{t_k}^{t_{k+1}} (t_{k+1} - s)^2 - \frac{4(t_{k+1} - s)^4}{3\Delta t^2} + \frac{4(t_{k+1} - s)^6}{9\Delta t^4} ds + \\ &\quad + 2 \int_{t_{k-1}}^{t_k} \dddot{\ddot{x}}(s)^2 ds \cdot \int_{t_{k-1}}^{t_k} (s - t_{k-1})^2 - \frac{4(s - t_{k-1})^4}{3\Delta t^2} + \frac{4(s - t_{k-1})^6}{9\Delta t^4} ds \\ &= 2 \int_{t_k}^{t_{k+1}} \dddot{\ddot{x}}(s)^2 ds \cdot \left[-\frac{(t_{k+1} - s)^3}{3} + \frac{4(t_{k+1} - s)^5}{15\Delta t^2} - \frac{4(t_{k+1} - s)^7}{63\Delta t^4} \right]_{t_k}^{t_{k+1}} + \\ &\quad + 2 \int_{t_{k-1}}^{t_k} \dddot{\ddot{x}}(s)^2 ds \cdot \left[\frac{(s - t_{k-1})^3}{3} - \frac{4(s - t_{k-1})^5}{15\Delta t^2} + \frac{4(s - t_{k-1})^7}{63\Delta t^4} \right]_{t_{k-1}}^{t_k} \\ &= 2 \int_{t_k}^{t_{k+1}} \dddot{\ddot{x}}(s)^2 ds \cdot \left(\frac{\Delta t^3}{3} - \frac{4\Delta t^5}{15\Delta t^2} + \frac{4\Delta t^7}{63\Delta t^4} \right) + 2 \int_{t_{k-1}}^{t_k} \dddot{\ddot{x}}(s)^2 ds \cdot \left(\frac{\Delta t^3}{3} - \frac{4\Delta t^5}{15\Delta t^2} + \frac{4\Delta t^7}{63\Delta t^4} \right) \\ &= \frac{82\Delta t^3}{315} \int_{t_{k-1}}^{t_{k+1}} \dddot{\ddot{x}}(s)^2 ds \end{aligned}$$

which leads to:

$$\begin{aligned}\sum_{k=1}^{m-1} \|w_{k+1} + 2w_k + w_{k-1}\|_H^2 &\leq \frac{82\Delta t^3}{315} \sum_{k=1}^{m-1} \int_{t_{k-1}}^{t_{k+1}} \dddot{x}(s)^2 ds \\ &\leq \frac{164\Delta t^3}{315} \int_0^T \dddot{x}(s)^2 ds\end{aligned}$$

Accordingly, we find for $\widetilde{w}$:

$$\begin{aligned}&\widetilde{w}_{k+1} + 2\widetilde{w}_k + \widetilde{w}_{k-1} \\ &= \dot{x}(t_{k+1}) + 2\dot{x}(t_k) + \dot{x}(t_{k-1}) - \frac{2}{\Delta t}\left(x(t_{k+1}) - x(t_{k-1})\right)\end{aligned}$$

by Taylor expansion:

$$\begin{aligned}\dot{x}(t_{k+1}) &= \dot{x}(t_k) + \Delta t \cdot \ddot{x}(t_k) + \int_{t_k}^{t_{k+1}} (t_{k+1} - s) \cdot \dddot{x}(s) ds \\ \dot{x}(t_{k-1}) &= \dot{x}(t_k) - \Delta t \cdot \ddot{x}(t_k) - \int_{t_k}^{t_{k-1}} (s - t_{k-1}) \cdot \dddot{x}(s) ds\end{aligned}$$

yielding:

$$\begin{aligned}&\dot{x}(t_{k+1}) + 2\dot{x}(t_k) + \dot{x}(t_{k-1}) - \frac{2}{\Delta t}\left(x(t_{k+1}) - x(t_{k-1})\right) = \\ &= \int_{t_k}^{t_{k+1}} \dddot{x}(s) \cdot \left(s - t_{k+1} + \frac{2(t_{k+1} - s)^2}{2\Delta t}\right) ds - \\ &\quad - \int_{t_{k-1}}^{t_k} \dddot{x}(s) \cdot \left(s - t_{k-1} - \frac{2(s - t_{k-1})^2}{2\Delta t}\right) ds\end{aligned}$$

and

$$\sum_{k=1}^{m-1} \|\widetilde{w}_{k+1} + 2\widetilde{w}_k + \widetilde{w}_{k-1}\|_H^2 \leq c \cdot \Delta t^3 \int_0^T \dddot{x}(s)^2 ds$$

The estimates for $z_k = \partial\partial x(t_k) - \partial\partial P^l x(t_k)$ and $\widetilde{z}_k = \partial x(t_k) - \partial P^l x(t_k)$ depend on the choice of the POD subspace, see equations (3.18), (3.19), resp. (3.21) and (3.22) :

$$\begin{aligned}
\text{case I:} \qquad & \sum_{k=1}^{m-1} \|z_{k+1} + 2z_k + z_{k-1}\|_X^2 \;\leq\; \frac{24}{\Delta t^4}(m+1) \sum_{j=l+1}^{d} \lambda_{Ij} \\
\text{and} \qquad & \sum_{k=1}^{m-1} \|\widetilde{z}_{k+1} + 2\widetilde{z}_k + \widetilde{z}_{k-1}\|_X^2 \;\leq\; \frac{16}{\Delta t^2}(m+1) \sum_{j=l+1}^{d} \lambda_{Ij}, \\
\text{case II:} \qquad & \sum_{k=1}^{m-1} \|z_{k+1} + 2z_k + z_{k-1}\|_X^2 \;\leq\; 72m \cdot \sum_{j=l+1}^{d} \lambda_{IIj} \\
\text{and} \qquad & \sum_{k=1}^{m-1} \|\widetilde{z}_{k+1} + 2\widetilde{z}_k + \widetilde{z}_{k-1}\|_X^2 \;\leq\; 72m \cdot \sum_{j=l+1}^{d} \lambda_{IIj}.
\end{aligned}$$

Combining the estimates for z_k , $\widetilde{z}_k$, w_k and $\widetilde{w}_k$ we have

$$\begin{aligned}
\|\gamma_k\|_H^2 &= \|v_{k+1} + 2v_k + v_{k-1}\|_H^2 \\
&\leq 4\,\|w_{k+1} + 2w_k + w_{k-1}\|_H^2 + 4\,\|z_{k+1} + 2z_k + z_{k-1}\|_H^2 + \\
&\quad + 4D\,\|\widetilde{w}_{k+1} + 2\widetilde{w}_k + \widetilde{w}_{k-1}\|_H^2 + 4D\,\|\widetilde{z}_{k+1} + 2\widetilde{z}_k + \widetilde{z}_{k-1}\|_H^2
\end{aligned}$$

and get for case I:

$$\begin{aligned}
\sum_{i=1}^{m} \|\gamma_i\|_H^2 \;\leq\; & \frac{328\Delta t^3}{315} \int_0^T \dddot{x}(s)^2 ds + \frac{48}{\Delta t^4}(m+1) \sum_{j=l+1}^{d} \lambda_{Ij} + \\
& + c\Delta t^3 \int_0^T \ddot{x}(s)^2 ds + \frac{16}{\Delta t^2}(m+1) \sum_{j=l+1}^{d} \lambda_{Ij}
\end{aligned}$$

and for case II:

$$\begin{aligned}
\sum_{i=1}^{m} \|\gamma_i\|_H^2 \;\leq\; & \frac{328\Delta t^3}{315} \int_0^T \dddot{x}(s)^2 ds + 144m \cdot \sum_{j=l+1}^{d} \lambda_{IIj} + \\
& + c\Delta t^3 \int_0^T \ddot{x}(s)^2 ds + 144m \cdot \sum_{j=l+1}^{d} \lambda_{IIj}
\end{aligned}$$

In conclusion, we have:

$$\begin{aligned}\frac{1}{m}\sum_{k=1}^{m}\left\|X^k - x(t_k)\right\|_H^2 &\leq \frac{2}{m}\sum_{k=1}^{m}\|\vartheta_k\|_H^2 + \frac{2}{m}\sum_{k=1}^{m}\|\rho_k\|_H^2 \\ \text{with}\quad \frac{1}{m}\sum_{k=0}^{m}\|\vartheta_k\|_H^2 &\leq 2\|\vartheta_1\|_H^2 + 2Te^T\left(2\cdot E^1 + \Delta t\left\|\frac{\vartheta_1-\vartheta_0}{\Delta t}\right\|_H^2 + \frac{\Delta t}{4}\sum_{i=1}^{m}\|\gamma_i\|_H^2\right)\end{aligned}$$

and the error estimate for case I can be written as:

$$\begin{aligned}&\frac{1}{m}\sum_{k=1}^{m}\left\|X^k - x(t_k)\right\|_H^2 \leq \\ &\leq\ 4\sum_{j=l+1}^{d}\lambda_{Ij} + 4\|\vartheta_1\|_H^2 + \\ &\quad + 4Te^T\left(2\cdot E^1 + \Delta t\left\|\frac{\vartheta_1-\vartheta_0}{\Delta t}\right\|_H^2 + \frac{82\Delta t^4}{315}\int_0^T \dddot{\ddot{x}}(s)^2 ds + \right. \\ &\quad \left. + cD\Delta t^4\int_0^T \ddot{\ddot{x}}(s)^2 ds + \left(\frac{12}{\Delta t^3} + \frac{4D}{\Delta t}\right)(m+1)\sum_{j=l+1}^{d}\lambda_{Ij}\right)\end{aligned}$$

The term E^1 contains the expression $(\vartheta_1 - \vartheta_0)/\Delta t$ which can be regarded as an extended initial condition for the velocities $(\partial\vartheta_1 + \partial\vartheta_0)/2$ due to Newmark's scheme (3.6a).

Therefore:

$$\begin{aligned}&\frac{1}{m}\sum_{k=1}^{m}\left\|X^k - x(t_k)\right\|_H^2 \leq \\ &\leq\ C_I\left(\left\|X^0 - P^l x(t_0)\right\|_H^2 + \left\|X^1 - P^l x(t_1)\right\|_H^2 + \Delta t\left\|\partial X^0 - P^l\dot{x}(t_0)\right\|_H^2\right. \\ &\quad \left. + \Delta t\left\|\partial X^1 - P^l\dot{x}(t_1)\right\|_H^2 + \Delta t^4 + \left(\frac{1}{\Delta t^4} + \frac{1}{\Delta t} + 1\right)\sum_{j=l+1}^{d}\lambda_{Ij}\right)\end{aligned}$$

with C_I being a constant that depends on T, D, $\ddot{x}$ and $\dddot{\ddot{x}}$, but not on Δt or m.

Case II yields the following estimate:

$$\begin{aligned}
&\frac{1}{m}\sum_{k=1}^{m}\left\|X^k - x(t_k)\right\|_H^2 \leq \\
&\leq\ 6\sum_{j=l+1}^{d}\lambda_{IIj} + 4\left\|\vartheta_1\right\|_H^2 + \\
&\quad + 4Te^T\left(2\cdot E^1 + \Delta t\left\|\frac{\vartheta_1-\vartheta_0}{\Delta t}\right\|_H^2 + \frac{82\Delta t^4}{315}\int_0^T \dddot{x}(s)^2 ds + \right. \\
&\quad \left. + 36T(1+D)\cdot\sum_{j=l+1}^{d}\lambda_{IIj} + c\Delta t^4\int_0^T \ddot{x}(s)^2 ds\right)
\end{aligned}$$

which can similarly be interpreted as

$$\begin{aligned}
&\frac{1}{m}\sum_{k=1}^{m}\left\|X^k - x(t_k)\right\|_H^2 \leq \\
&\leq\ C_{II}\left(\left\|X^0 - P^l x(t_0)\right\|_H^2 + \left\|X^1 - P^l x(t_1)\right\|_H^2 + \Delta t\left\|\partial X^0 - P^l\dot{x}(t_0)\right\|_H^2\right. \\
&\quad \left. + \Delta t\left\|\partial X^1 - P^l\dot{x}(t_1)\right\|_H^2 + \Delta t^4 + \sum_{j=l+1}^{d}\lambda_{IIj}\right),
\end{aligned} \tag{3.28}$$

where C_{II} depends only on T, D, $\ddot{x}$ and $\dddot{x}$, but not on Δt or m.

In both cases, we find terms that are independent of the time step Δt. Both cases also contain terms that depend on Δt in the numerator. These terms vanish as Δt goes to zero.

In case I, which only uses the deformation snapshots, the error estimate additionally contains a term that carries Δt in the denominator. For this choice of POD subspace no valid error bound can be set any more as the term tends to infinity with $\Delta t \to 0$.

This means that convergence cannot be assured formally, if a snapshot set consisting of deformations only is used. If velocities and accelerations are added into the set, convergence results from (3.28).

3.3 Numerical Results

For a numerical comparison of the different POD techniques discussed above, a simple test model was set up in Matlab. The example shows a one-dimensional linear wave equation on the interval $\Omega = (0, L)$ with homogeneous Dirichlet boundary conditions, which can be regarded as a vibrating string fixed at both ends.

Mathematically, our model problem is described by the following initial-boundary value problem:

$$\begin{aligned} \mu \cdot \ddot{x}(s,t) - S \cdot x''(s,t) &= f(s,t) \quad \text{in } (0,L) \times (0,T), && (3.29a)\\ x(s,0) &= x_0 \quad \text{in } (0,L), && (3.29b)\\ \dot{x}(s,0) &= \dot{x}_0 \quad \text{in } (0,L), && (3.29c)\\ x(s,t) &= 0 \quad \text{on } \partial\Omega = \{0,L\} \quad \text{for all } t \in (0,T). && (3.29d) \end{aligned}$$

We chose $L = 1$, $S = 1$, $\mu = 1$ and $T = 2$.

The initial deformation x_0 is set to be a weighted sum of sinus shapes:

$$x_0 = x(t_0) = \sum_{i=1}^{n} a_i \cdot sin\left(i \cdot \pi \frac{s}{L}\right),$$

with weights $a_i \in \mathbb{R}$.

Furthermore, we set the external force $f(t)$ to zero, yielding the analytical solution and its derivatives:

$$\begin{aligned} x(s,t) &= \sum_{i=1}^{n} a_i \cdot sin\left(i\pi \frac{s}{L}\right) \cdot cos\left(i\pi \frac{c}{L} t\right), && (3.30a)\\ \dot{x}(s,t) &= \sum_{i=1}^{n} -a_i \cdot sin\left(i\pi \frac{s}{L}\right) \cdot sin\left(i\pi \frac{c}{L} t\right) \cdot i\pi \frac{c}{L}, && (3.30b)\\ \ddot{x}(s,t) &= \sum_{i=1}^{n} -a_i \cdot sin\left(i\pi \frac{s}{L}\right) \cdot cos\left(i\pi \frac{c}{L} t\right) \cdot i^2\pi^2, \frac{c^2}{L^2} && (3.30c)\\ \text{with} \quad c &= \sqrt{\frac{S}{\mu}}. \end{aligned}$$

The POD method was realised using snapshots at $m + 1$ uniformly distributed points in time. To observe the error behaviour with decreasing time step, we investigate three different step sizes dividing the interval into $m = 400$, 2000 and 20000 sub intervals. With this setup, we use Newmark's method for the time integration. The spatial discretisation is done by a linear Finite Element approach consisting of 500 elements.

For the case of only including the displacements into the POD set, the snapshots are simply $\{x(t_k)\}_{k=0,\ldots,m}$. In the second case, where deformations, velocities and accelerations were taken into account, the snapshot set was built using the resulting quantities from the analytical solution: $x(t_0), \ldots, x(t_m), \dot{x}(t_0), \ldots, \dot{x}(t_m), \ddot{x}(t_0), \ldots, \ddot{x}(t_m)$.

Remark 6 *Note that when using unweighted snapshots, the eigenvalues of the derivative set differ from the ones of the deformation set by a factor of* 10^8 *(Figure 3.1). This*

observation originates from the fact that the velocities and accelerations are about 2*, respectively* 3*, orders of magnitude larger than the deformations. This difference leads not only to large eigenvalues but also to an overrating of the derivatives in the correlation matrix. For this reason, the derivative snapshots were divided by the respective maximum over space and time.*

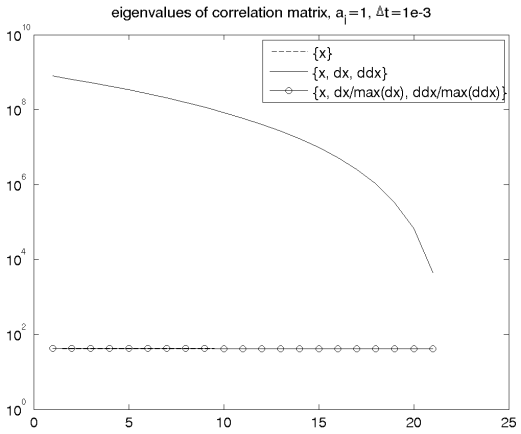


Figure 3.1: Decay of eigenvalues of the snapshot correlation matrix, for $a_i = 1$

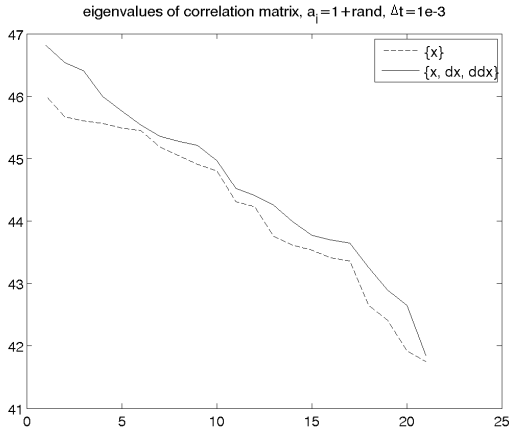


Figure 3.2: Decay of eigenvalues of the snapshot correlation matrix, for $a_i = 1+$ *rand*

Furthermore, to investigate the influence of the correlation matrix eigenvalues, we compare two different initial conditions: one consisting of uniformly weighted sinus shapes ($a_i = 1$), and one additionally containing small random numbers ($a_i = 1+rand$, $max(rand) = 0.05$). The former yields a nearly constant distribution of eigenvalues up to the dimension of a, whereas the eigenvalues for the latter set decay linearly (Figures 3.1 and 3.2). Note that for problems including damping, the eigenvalues usually decay exponentially.

Figures 3.3 and 3.4 compare the norms of the relative global errors for the case $a_i = 1$. In this setup, the classical snapshot POD method shows no improvement with decreasing time step size whereas the one which uses derivative snapshots performs significantly better.

In the case of a random distribution of sinus weights, both methods show a diminishing error for smaller time steps (Figures 3.5 and 3.6). One possible reason is the influence of the eigenvalue decay on the error norm which dominates the error in this case.

Remark 7 *Note that in all cases mentioned above the absolute values of the error norms are still high (*$> 10\%$*). The dimension of the model corresponds to the dimension of* a *(here:* $dim(a) = 21$*). As soon as a larger number of POD vectors is used, the error drops instantly. This behaviour is also seen in the eigenvalue distribution, yielding* $\lambda_i = 0$ *for* $i > dim(a)$*.*

Therefore, a setup with such weighting of modes actually forces the user to work with all occurring eigenvalues as every neglected basis vector still has a considerable influence on

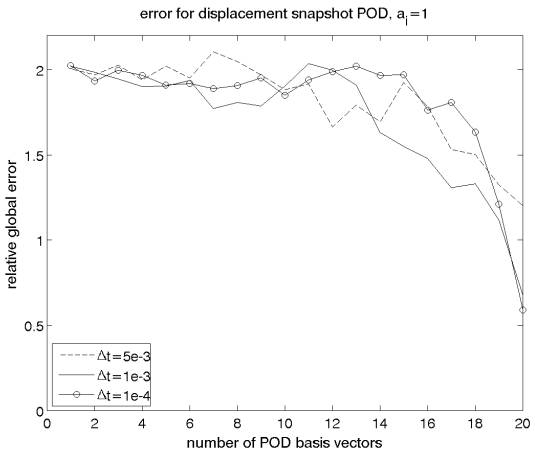


Figure 3.3: Error norms for deformation snapshot set, $a_i = 1$

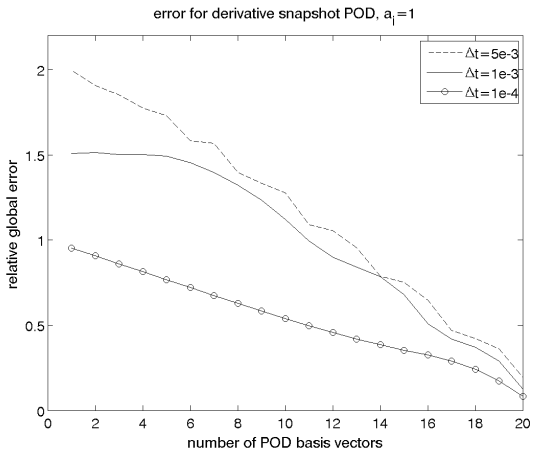


Figure 3.4: Error norms for derivative snapshot set, $a_i = 1$

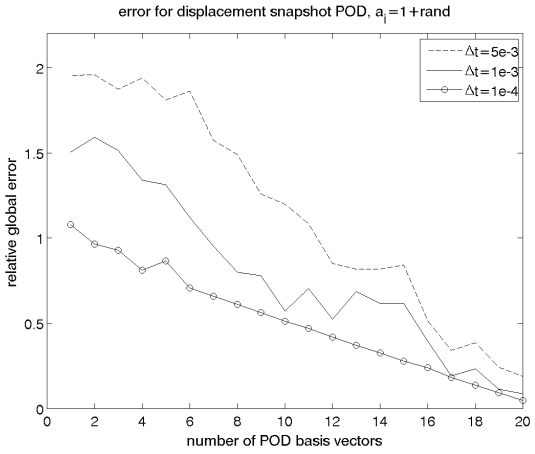


Figure 3.5: Error norms for deformation snapshot set, $a_i = 1 + rand$

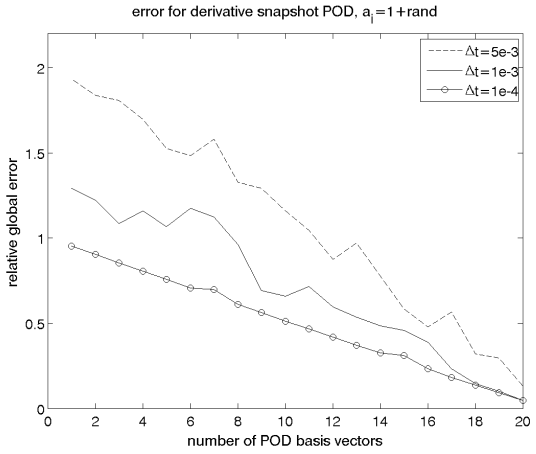


Figure 3.6: Error norms for derivative snapshot set, $a_i = 1 + rand$

the solution. In this case, dimension reduction is risky and the example shall only be seen as a constructed model to demonstrate the error behaviour.

As a second example, we use a non-smooth initial condition u_0 (Figure 3.7) on the same setup as above and compare the POD methods with the classical eigenmode method frequently used for linear systems. Furthermore, we set the damping factor $d = 10$.

In the case of high damping, we get an exponential decay of eigenvalues (Figure 3.8). A fast eigenvalue decay leads to a small error in subspace approximation of the snapshot set (see (2.26)). This yields a better condition for the POD method than the example above.

Figure 3.9 shows a comparison of the relative global errors of both POD and the eigenmode methods. The errors are computed in the H-norm $\frac{1}{m}\sum_{k=1}^{m}\left\|X^k - x(t_k)\right\|_H^2$. In this case,

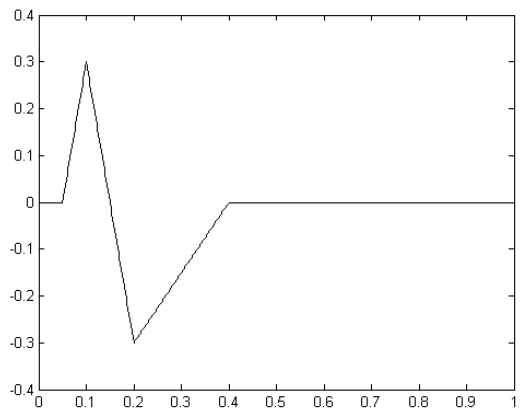


Figure 3.7: Initial condition for the linear wave equation

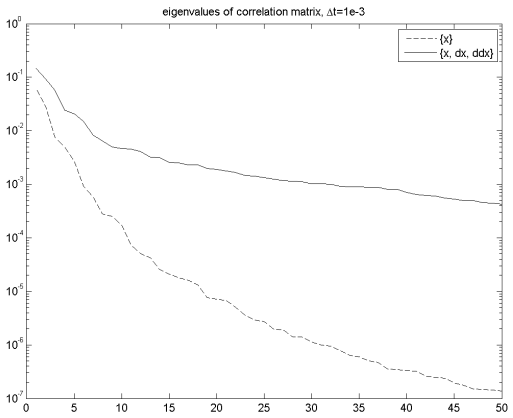


Figure 3.8: Decay of eigenvalues for the POD snapshot sets, $\Delta t = 10^{-3}$

the derivative POD method performs slightly worse than the classical one. The errors for the eigenmode method range between the ones of both POD methods.

If we measure the error in the V-norm, we find that both POD methods perform better than the eigenmodes (Figure 3.10).

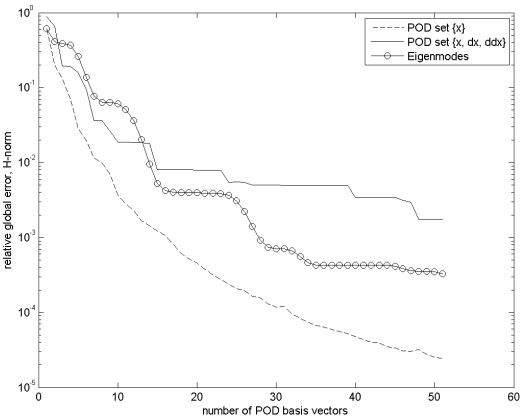


Figure 3.9: Error norms for POD and eigenmode analysis, damping $d = 10$, H-norm, $\Delta t = 10^{-3}$

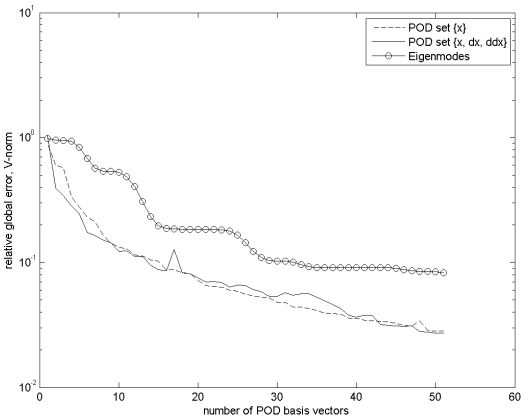


Figure 3.10: Error norms for POD and eigenmode analysis, damping $d = 10$, V-norm, $\Delta t = 10^{-3}$

In the case of zero damping, the snapshot correlation eigenvalues decay slowlier than in the damped case. Therefore, the advantage of the POD subspace with respect to the eigenmode subspace decreases. Figures 3.11 and 3.11 show that the error behaves similarly for the eigenmode and the POD subspace methods.

In conclusion, the POD method for the linear wave equation was studied comparing two

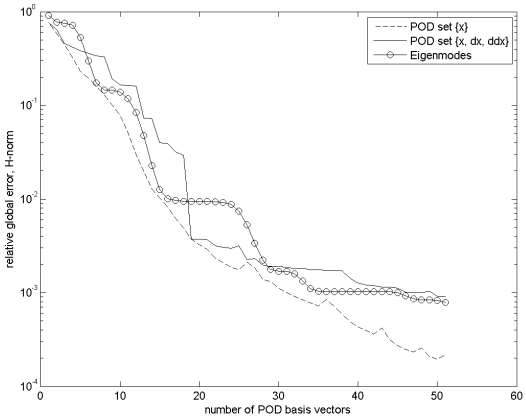


Figure 3.11: Error norms for POD and eigenmode analysis, undamped system, H-norm, $\Delta t = 10^{-3}$

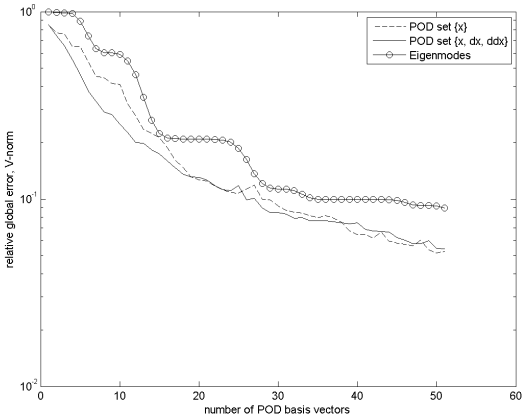


Figure 3.12: Error norms for POD and eigenmode analysis, undamped system, V-norm, $\Delta t = 10^{-3}$

different choices of snapshot sets. Set I consisted of deformation snapshots, and set II additionally contained velocities and accelerations.

The error estimate is closely related to the work of Kunisch and Volkwein ([26]). It proved no convergence guarantee for deformations only, but for inclusion of derivatives it yielded an error bound diminishing for small time steps.

The numerical results showed a better behaviour for the derivative snapshot method, as long as the sum of the left-over eigenvalues is significant. Still, the influence of eigenvalue decay on the error shall be further investigated. Furthermore, for damped systems, the POD subspaces produce smaller errors than the subspace of eigenmodes. In the undamped case, both subspaces perform equally well.

Chapter 4

Model Reduction of Nonlinear Problems

This chapter addresses model reduction by POD for nonlinear problems. In the approach introduced here, the nonlinear term and its derivative are stored during the computation run of the full model. The information serves as a lookup table for the reduced models and gives an approximation of the nonlinearities. In this manner, the reduced model is decoupled from the full model. Furthermore, the method of POD yields a set of basis vectors regardless of the nonlinearity.

4.1 Considerations for the Nonlinear Case

Like all model reduction methods that are based on subspace projection, POD reduces the effort required to solve the linearised system of equations in each iteration step. For linear models, the reduced model is set up in the first step and can be used unchanged throughout the whole computation:

$$M\ddot{u} + C\dot{u} + Ku = f_{ext} \tag{4.1}$$

$$P^T \cdot M \cdot (\ddot{P\alpha}) + P^T \cdot C \cdot (\dot{P\alpha}) + P^T \cdot K \cdot (P\alpha) = P^T \cdot f_{ext} \tag{4.2}$$

$$\widehat{M}\ddot{\alpha} + \widehat{C}\dot{\alpha} + \widehat{K}\alpha = \widehat{f_{ext}} \tag{4.3}$$

where $\widehat{M}, \widehat{C}, \widehat{K} \in \mathbb{R}^{m \times m}$, $M, C, K \in \mathbb{R}^{N \times N}$ and $m << N$.

In nonlinear problems, the model equations depend on the current state u:

$$M\ddot{u} + C\dot{u} + R(u) = f_{ext} \tag{4.4}$$

$$P^T \cdot M \cdot (\ddot{P\alpha}) + P^T \cdot C \cdot (\dot{P\alpha}) + P^T \cdot R(P\alpha) = P^T \cdot f_{ext} \tag{4.5}$$

The effort to set up the nonlinear term $R(P\alpha)$ requires the transformation of the current state α to the full dimensional variable $u = P\alpha$. In general industrial problems - except in

rare cases - the nonlinearity is not explicitly known or can not be computed in the reduced variable. Obviously, model reduction by POD is only helpful and sensible when most of the computational cost lies in the solution of the linearised problem and not in the composition of the equation system.

In the problems described in this work, this is not generally the case. The use of commercial FEM codes is necessary for a reasonable description of a rolling tyre with all its physical effects and complex interaction of the involved parts. Due to the black-box character of these codes, communication with commercial programs requires a cumbersome read- and write-procedure whenever the current equation system is needed. Some FEM programs only allow exchange of information by text files. This leads to computational costs which are far beyond any feasible time scale. Furthermore, the problem of licences can become a serious issue, as the solver of the FEM program is used in each time step to set up the equations.

For full exploitation of the capabilities of the POD method, the computation of the reduced problem needs to be decoupled from the commercial tool. In general, profound knowledge of the full system is needed in order to formulate the nonlinear equations with respect to the reduced variables only. However, black box tools do not allow such approaches as only few details of the underlying equations can be gained from commercial programs.

In the following sections, an approach is presented which uses the value of the nonlinear term and the derivative matrices of a full model to build up the decoupled reduced system.

4.2 State of the Art

The most popular way to deal with nonlinearities in model reduction is the *trajectory piecewise linear (TPWL)* method analysed in [38]. Here, a first run of the full problem is performed with a training input. Along the resulting solution trajectory, linearisations of the model equations at chosen states x_i are stored. Using a suitable subspace, the projected linearisations are used to construct a linear model

Emanating from the full first order problem

$$\frac{d}{dt}g(x) = f(x) + B(x)u \tag{4.6a}$$

$$y = C^T x \tag{4.6b}$$

linearisations at $x_i, i = 0 \dots (s-1)$ are computed:

$$\frac{d}{dt}\left(g(x_i) + G_i(x - x_i)\right) = f(x_i) + A_i(x - x_i) + B_i u \tag{4.7}$$

where $G = \frac{\partial g}{\partial x}$, $A = \frac{\partial f}{\partial x}$ and G_i, A_i, B_i are the respective evaluations at x_i.

For the construction of the reduced space in first order problems, Rewieński proposes the use of an extended Krylov subspace. This subspace incorporates the Krylov spaces of all linearised models. The basis vectors with the highest correlation are then chosen for projection.

This can be combined with the snapshot creation step needed for the POD method.

At the beginning of each time step, a linear system is set up as a weighted sum of the stored linearisations:

$$\frac{d}{dt}\left[\sum_{i=0}^{s-1} w_i(z)\left(g(z_i)+G_i(z-z_i)\right)\right] = \sum_{i=0}^{s-1} w_i(z)\left(f(z_i)+A_i(z-z_i)+B_i u\right) \tag{4.8}$$

in the reduced space using $z_i = V^T x_i$ with weights $w_i \in \mathbb{R}$.

The computation of the weights depends on the distance of the current state z to the states of the linearisations z_i:

$$\begin{aligned} d_i &= \|z - z_i\|, \quad i = 0 \ldots s-1 \\ m &= \min_i d_i \\ \widehat{w}_i &= e^{-\beta d_i/m} \\ w_i &= \frac{\widehat{w}_i}{\sum_i \widehat{w}_i} \end{aligned}$$

where Rewieński ([38]) suggests to set $\beta = 25$. Once set up, a linear equation is solved in each time interval.

For the construction of a Krylov subspace for second order problems, see [31].

In a strongly different approach introduced by Ali, Damodaran and Willcox in [3], the full model arises from a black-box application

$$\dot{U} = X(U) \tag{4.9}$$

with an unknown spatial operator X. The crucial idea behind their method is to model the reduced system by a quadratic equation

$$\dot{U} = A + B \cdot U + U^T \cdot (C \cdot U) \tag{4.10}$$

where A, B and C are a vector, a matrix and a tensor of appropriate dimension.

Using the classical POD approximation to $U(x,t)$

$$U_N(x,t_k) = \sum_{i=1}^{N} \alpha_i(t_k)\Phi_i(x) \tag{4.11}$$

with the POD basis $\{\Phi_i\}_{i=1,\dots,N}$, the reduced model can be written as

$$\frac{d\alpha_i}{dt} = a_i + \sum_{j=1}^{N} b_{ij}\alpha_j + \sum_{k=1}^{N}\sum_{j=1}^{N} c_{ijk}\alpha_j\alpha_k \tag{4.12}$$

with a_i, b_{ij} and c_{ijk} arising from projection of the unknown quantities A, B and C onto the POD subspace.

As the true model is unknown, the coefficients of a, b and c need to be determinded from the POD snapshot set. Applying a Finite Difference Method to the snapshot participation factors $\alpha_i(t_k), k = 1, \dots, M$, an approximation of $d\alpha_i/dt$ is computed. This data set is used to identify the coefficients a_i, b_{ij} and c_{ijk} in the reduced model 4.12 by a least squares method.

In [3], also a linear approach is proposed, setting $c_{ijk} = 0 \ \forall \ i, j, k$. The method performs well for linear examples tested in that publication.

4.3 Lookup Table Approach

The approach of using a lookup table starts with the same steps as the TPWL method. First, linearisations of the system are stored during a first run of the full problem. These vectors and matrices are projected onto a low-dimensional subspace and used as a lookup table whenever the nonlinear term or its derivative is needed in the further computation. In each time step, a quasi-Newton routine is performed to solve the nonlinear equation.

4.3.1 Construction of the Lookup Table

We assume we have a nonlinear differential equation of second order

$$F(u, \dot{u}, \ddot{u}, t) = 0 \tag{4.13}$$

where F is continuous and differentiable with respect to u, $\dot{u}$ and $\ddot{u}$. Furthermore, it may depend nonlinearly on u and u', but linearly on $\dot{u}$ and $\ddot{u}$.

Applying a Finite Element discretisation in space we define the semi-discretised equations. We search for a vector $u(t) \in \mathbb{R}^N$ satisfying

$$\begin{aligned} M(u)\ddot{u} + C(u)\dot{u} + R(u) &= f_{ext}(t) \\ \text{where } M(u) &:= \frac{\partial F}{\partial \ddot{u}} \\ C(u) &:= \frac{\partial F}{\partial \dot{u}} \end{aligned} \tag{4.14}$$

where the matrices $M, C \in \mathbb{R}^{N\times N}$ denote the mass and damping matrix, respectively. All three matrices may depend on the current deformation $u = u(t)$.

Assuming the matrices M, C, the nonlinearity $R(u)$ and its linearisation $K(u)$ can be extracted from the Finite Element software in each time step, the following procedure is set up:

(1) on each time level t_i, store the current state u_i, the nonlinear deformation expression $R(u_i)$ and its derivative $K(u_i) = \frac{\partial R}{\partial u}\Big|_{u_i}$

(2) (if coupled to a POD reduction scheme) project the data onto the given POD subspace

(3) if necessary, select few relevant states which define the look-up table

(4) in each iteration of the reduced model, construct approximation of the nonlinear term and its linearisations from the given data stored in the table

Note that steps $(1) - (3)$ are treated offline.

The computation steps of the reduced problem mainly consist of the solution of the projected equations and the treatment of the lookup table. Depending on the deformation variety and the nonlinearity of the system, the table can become rather large. Therefore, a simple search algorithm is needed.
Furthermore, during application of the reduced model, states u_i may occur which are not included in the table. The lookup algorithm should account for variations of external loads and therefore variations of the deformations within a reasonable range.

This implies that lookup strategies should allow for interpolation and extrapolation of the entries and still remain stable.

4.3.2 The Reduced Surrogate Model

Setting up a surrogate model, we have

$$\widehat{M}\ddot{u} + \widehat{C}\dot{u} + \widehat{R}(u) = \widehat{f}_{ext}(t) \tag{4.15}$$

where $\widehat{M}, \widehat{C}$ and $\widehat{f}$ can be identical to the corresponding quantities in the original model or suitable subspace approximations. In the computational treatment of model 4.15, the nonlinear part is approximated using the following lookup scheme.

Given the current state u_τ, the approximation of $R(u_\tau)$ and $K(u_\tau)$ is computed by data linearisation in the lookup table. Let the lookup data consist of states $u_i, i = 1, \ldots, m$, their corresponding deformation terms $R(u_i)$ and linearisations $K(u_i) = \frac{\partial R}{\partial u}\Big|_{u_i}$. In each

iteration, we determine the approximated model by Taylor expansion in the neighbourhood of a given state u_i:

$$\begin{aligned} R(u_\tau) &= R(u_i) + K(u_i)(u_\tau - u_i) + \\ &\quad + \frac{1}{2}(u_\tau - u_i)^T \frac{\partial K}{\partial u}\Big|_{u_i}(u_\tau - u_i) + \mathbf{O}\left(\|u_\tau - u_i\|^3\right) \qquad (4.16) \\ K(u_\tau) &= K(u_i) + (u_\tau - u_i)^T \frac{\partial K}{\partial u}\Big|_{u_i} + \mathbf{O}\left(\|u_\tau - u_i\|^2\right) \qquad (4.17) \end{aligned}$$

The state u_i is chosen as the nearest state to the current deformation u_τ measured in the L_2 norm. Note that the *Hessian tensor* $H(u_i) = \frac{\partial K}{\partial u}\big|_{u_i}$ is not extracted from the full model.

Emanating from the full lookup table with small time steps, it is possible to make use of the additional information of the model progress in time. If $t_{i+1} - t_i$ and therefore $u(t_{i+1}) - u(t_i)$ is small enough, the development of $K(u(t_i))$ to $K(u(t_{i+1}))$ yields an approximation of the derivative of $K(u(t_i))$ in the direction of $u(t_{i+1}) - u(t_i)$.

$$\begin{aligned} \text{From} \quad K(u_{i+1}) &\approx K(u_i) + (u_{i+1} - u_i)^T H(u_i) \\ \text{we get} \quad (u_{i+1} - u_i)^T H(u_i) &\approx K(u_{i+1}) - K(u_i) \\ \text{and} \quad (\widetilde{u}_\tau - u_i)^T H(u_i) &\approx d \cdot [K(u_{i+1}) - K(u_i)] \qquad (4.18) \end{aligned}$$

where $\widetilde{u}_\tau$ denotes the projection of u_τ onto the direction $u(t_{i+1}) - u(t_i)$ and

$$d := \frac{\langle u_\tau - u_i, u_{i+1} - u_i \rangle}{\|u_{i+1} - u_i\|^2}. \qquad (4.19)$$

If the factor d is negative or larger than 1, the projection of u_τ does not lie on the line between u_k and u_{k+1}. In the case of $d < 0$, the backward difference $u_i - u_{i-1}$ is chosen for projection.

In the following, the term *Lookup 1* denotes the lookup method using Taylor expansion up to first order. When the nonlinear term and its linearisation are required, they are approximated by

$$\begin{aligned} R(u_\tau) &\approx R(u_i) + K(u_i)(u_\tau - u_i) \qquad (4.20) \\ K(u_\tau) &\approx K(u_i). \qquad (4.21) \end{aligned}$$

The *Lookup 2* method uses Taylor series up to second order, i.e.,

$$\begin{aligned} R(u_\tau) &\approx R(u_i) + K(u_i)(u_\tau - u_i) + \frac{1}{2}(u_\tau - u_i)^T \frac{\partial K}{\partial u}\Big|_{u_i}(u_\tau - u_i) \qquad (4.22) \\ K(u_\tau) &\approx K(u_i) + (u_\tau - u_i)^T \frac{\partial K}{\partial u}\Big|_{u_i} \qquad (4.23) \end{aligned}$$

with approximation of the Hessian 4.18.

4.4 Example: Nonlinear Vibrating String

The one-dimensional linear wave equation describing a vibrating string without external excitation is given by

$$\mu \frac{\partial^2 u}{\partial t^2} - S_0 \frac{\partial^2 u}{\partial x^2} = 0 \tag{4.24}$$

where μ denotes the mass per unit length and S_0 is the initial tension.

For large displacements, the geometrical nonlinearities need to be taken into account. As a simple nonlinear factor we introduce ([18])

$$S = S_0 + \frac{\pi^2 EA}{4l^2} u^2(x) \tag{4.25}$$

which describes the change in tension S due to the elastic properties EA which consists of the *Young's modulus* E multiplied with the cross sectional area A.

This expression does not capture the full nonlinearity inherent in an arbitrary vibrating string, but yields a simple nonlinear equation in structural mechanics.

We set up the semi-discretised equation

$$M\ddot{u} + C\dot{u} + R(u) = f_{ext}(t) \tag{4.26}$$

where

$$\begin{aligned} C &:= \alpha M \\ R(u) &:= S_0 \widetilde{K} u + \frac{\pi^2 EA}{4l^2} u^2 \widetilde{K} u \\ K(u) &:= S_0 \widetilde{K} + \frac{\pi^2 EA}{4l^2} \left(u^2 \widetilde{K} + 2u \cdot \widetilde{K} u \right) \end{aligned}$$

with the linear stiffness matrix $\widetilde{K}_{ij} := \langle \Phi_x^i, \Phi_x^j \rangle$ computed from the derivative of the shape functions Φ.

Note that Rayleigh damping is used with a mass dependent part only.

Using Newmark's time integration scheme, we solve for $u_i = u(x, t_i)$ and store the solution, the nonlinear deformation part $R(u_i)$ and its linearisation $K(u_i)$ at each time instance t_i. These data define the entries of the lookup table. Furthermore, we store the mass and damping matrix at the beginning of the time integration. As the mass and damping terms are assumed to be linear, no lookup table interpolation is needed for this part.

In the next step, the POD eigenvalue problem is solved to create a reduced basis and a projection matrix for the computed snapshot set. The mass, damping and stiffness matrices as well as the nonlinear force term and the snapshot entries are projected onto the POD

subspace:

$$\begin{aligned} \widehat{S} &= P^T S P \quad \text{where} \quad S = M, C, K(u_i) \\ \widehat{R}_i &= P^T R(u_i) \\ \alpha_i &= P^T u_i \end{aligned}$$

where $\widehat{C} = P^T(\alpha M)P = \alpha\widehat{M}$.

The computational steps only use the lookup table $\left(\alpha_i, \widehat{R}_i, \widehat{K}_i\right)$, the mass matrix $\widehat{M}$, the damping factor α and the projection matrix P for transformation of the external forces and evaluation of the results.

The reduced basis in the following results consists of the usual POD basis over L_2. This basis is also included in the full Krylov subspace ([31]) proposed for the TPWL method ([38]). To better compare the lookup table and TPWL methods, the same POD projection basis is used for both approaches.

Remark 8 *In general, the subspace basis is determined before storing the lookup system. To reduce the amount of data, only the projected matrices are stored for the reduced model. Furthermore, if not all states are incorporated into the lookup table, it can make sense to run the full system two times. In the first step, the deformation snapshots are taken and the POD basis is computed. The matrices for the lookup table are stored in a second run of the full system at time instances of interest.*

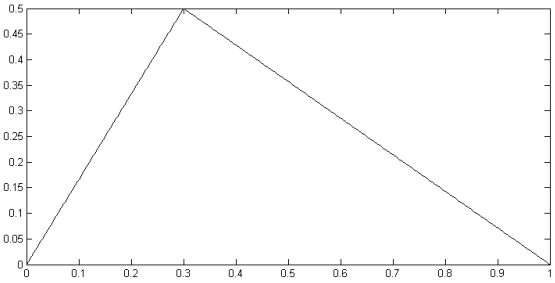


Figure 4.1: Initial condition u_0

In the computations performed here, no external excitation is applied. The string starts from a triangular initial deformation u_0 with a maximum defelxion of 0.5 (see Figure 4.1) and oscillates with damping $\alpha = 2$ until $t_{end} = 0.5s$ in 500 time steps. This case defines the snapshots for the lookup table as well as the POD subspace. In the following, lookup tables not using all states are built from states with equal distance over time. The geometry and material data for the string are $L = 1$, $S = 3.4$, $EA = 6$ and $\mu = 0.11$. Due to the high initial deformation, the nonlinear part has a large effect on the solution.

method	number of dof	size of table	CPU time	relative error
full system	301	n.a.	115.47s	n.a.
TPWL	20	101	1.78s	0.3%
Lookup 1	20	101	0.58s	0.2%
Lookup 2	20	101	1.17s	0.01%
TPWL	10	101	1.53s	0.08%
Lookup 1	10	101	0.53s	0.08%
Lookup 2	10	101	1.02s	0.06%
POD, full nonlinearity	10	n.a.	7.27s	6.5%
POD, full nonlinearity	20	n.a.	10.92s	3.0%
TPWL without POD	n.a.	101	194.23s	19.9%
Lookup 1 without POD	n.a.	101	7.59s	9.0%
Lookup 2 without POD	n.a.	101	41.92s	3.0%

Table 4.1: CPU time and comparison of relative errors in L_2 norm

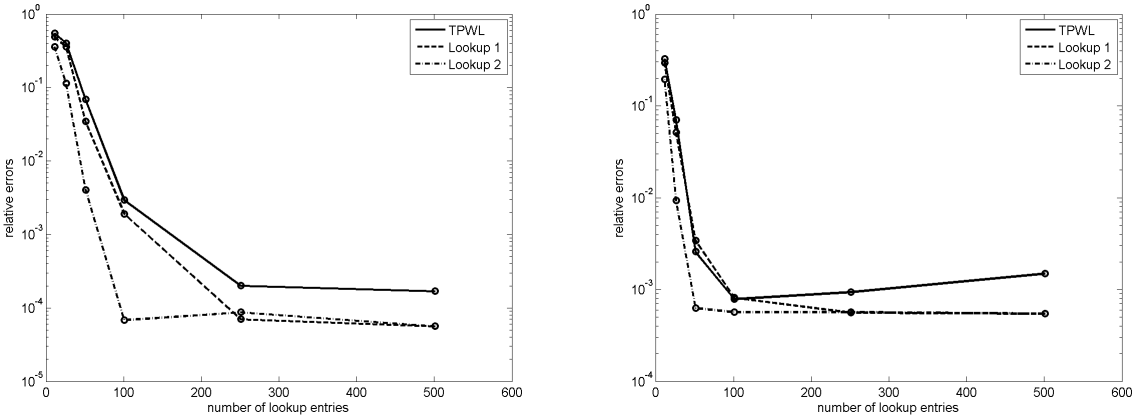
Table 4.1 compares CPU times and error norms for the different approaches. It shows that the lookup methods perform slightly faster than the TPWL method for all examined setups. This is due to the fact that the TPWL method has to incorporate all states of the lookup table into the linear equation. The lookup method with second order approximations is slower than the one with first order only, and the errors are in a comparable range.

Figure 4.2 shows that the error of the lookup table methods is generally smaller than the TPWL method, especially when many states are chosen in the table.

The errors of the lookup methods without POD reduction are shown in Figure 4.3(a). In these computations, the full system is computed using lookup and TPWL strategies for the treatment of the nonlinearities. In addition, the case of using POD subspace reduction without lookup methods is computed. Figure 4.3(b) illustrates the error of the POD reduced system using $1-40$ POD basis vectors with full evaluation of the nonlinear terms.

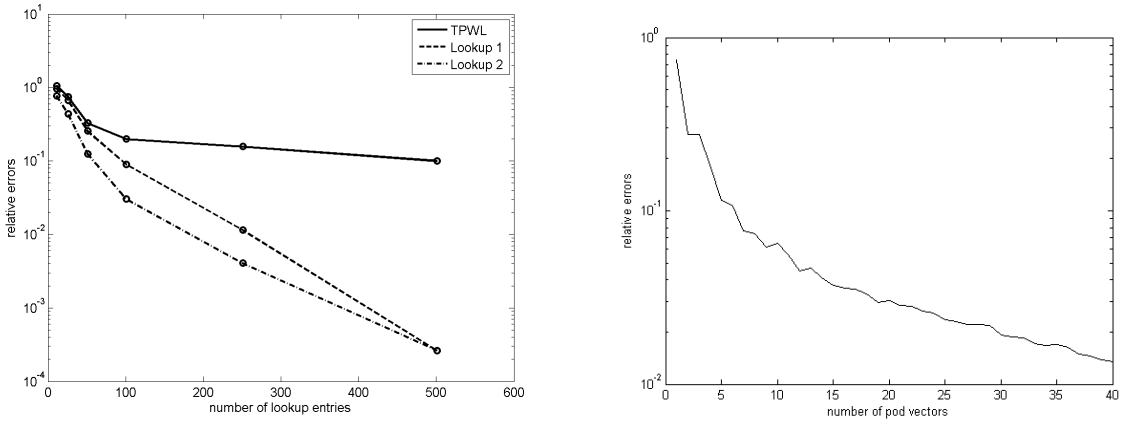
Comparing these errors to the ones of the POD/Lookup or POD/TPWL reduced systems, it can be seen that the errors of POD and Lookup methods partly even out. The deformations of the full system over time show that both the solutions produced by the POD reduction and the ones using only lookup strategies lag behind the full solution. When including all lookup states into the table, both Lookup 1 and Lookup 2 methods reproduce the full solution very well, whereas the solution of the TPWL method again moves slowlier in time yielding a larger error (Figure 4.3(a)).

A possible explanation is that the POD method supports the lookup and TPWL strategies in finding the right entries, as the subspace basis already contains some information on the dynamics of the system. Then again, the stored lookup solutions force the POD/lookup methods to use the projected correct terms instead of assembling the nonlinear terms from



(a) 20 POD vectors (b) 10 POD vectors

Figure 4.2: Relative global errors using TPWL, Lookup 1 and Lookup 2 methods with 10 or 20 POD basis vectors, respectively



(a) Lookup without POD (b) POD without lookup

Figure 4.3: Relative global errors using TPWL and Lookup 1 methods without POD reduction and errors using POD reduction with full evaluation of the nonlinearities

the projected solution as in the full POD case. However, these effects require further investigations.

Note that the system includes a strong nonlinearity. The entries of the stiffness matrix vary up to 100%, which implicates the necessity of Newton iterations within one time step. Using the full code with only one iteration per time step produces a relative error of 1.6%. Allowing a second iteration per time step yields exactly the same solution as the reference model for the current setup. This explains the better performance of the lookup methods compared to TPWL, especially in the case of using no POD projection.

Chapter 5

Model Reduction for a Finite Element Model in Abaqus

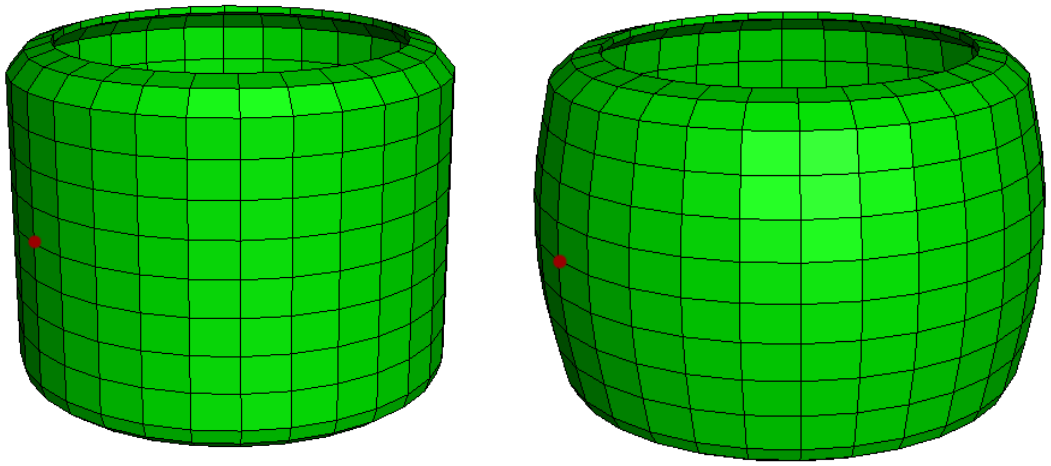
For application of the model reduction procedure introduced in the preceding chapters, it is necessary to include all effects inherent in the Finite Element formulation for a rolling tyre. In particular, this includes an appropriate description of the contact behaviour. Currently, this topic is still a problem in research and no efficient methods exist. It was not addressed within the framework of this thesis. Therefore, the following sections deal with a Finite Element model in Abaqus including geometrical nonlinearities but no contact mechanics.

5.1 Model Description of a Finite Element Airspring

As a moderately large industrially motivated example an airspring model was chosen (Figure 5.1). The Finite Element mesh is composed of 450 linear 8-node continuum elements yielding 960 nodes and therefore 2880 degrees of freedom. The entire spring cushion consisted of a nonlinear *Neo-Hooke* rubber material with the parameters $C_{10} = 2.9 \cdot 10^5$ and $D_1 = 3.5 \cdot 10^{-7}$ (see [1], [50], [52]) and a density of $\rho = 1.1 \cdot 10^3 kg/m^3$. Furthermore, mass proportional Rayleigh damping ($\alpha = 100$) was included in the material.

The computation consists of two steps. In the first static step, an inner pressure of $4kPa$ is applied to the inner surface. Notice that this type of load induces difficulties in the geometrically nonlinear case, as pressure forces have to be perpendicular to the inner surface. This leads to deformation dependent external forces $f_{ext}(u, t)$ which require additional iteration routines within the computations of the reduced model. For simplification, inner pressure was only applied to six element rows in the middle of the spring cushion (see Figure 5.2), as these are assumed to not undergo excessive rotation.

The second step consists of the dynamic computations. In this step a vertical load varying in magnitude and sign is applied to the upper element ring (Figures 5.2, 5.3). The dashed line denotes the run used for the setup of the reduced models, whereas the continuous line



(a) Undeformed configuration

(b) Deformed configuration after static step (inner pressure)

Figure 5.1: Abaqus model geometry of an airspring, output node marked

denotes the computation used for comparison of the model performance. Like in the static step, the nodes of the lower boundary are fixed in all three directions. The nodes of the upper boundary are allowed to move in vertical direction only (see Figure 5.2).

The computation is accomplished by an implicit transient analysis in Abaqus as described in [1]:

$$M\ddot{u}_i + C\dot{u}_i + I\left(u_i\right) - P_i + L_j = 0 \tag{5.1}$$

with constant mass and damping matrices M and $C = \alpha \cdot M$. The vector I denotes the inner forces in the element, P_i the external excitation at time t_i and L_j the Lagrangian forces induced by boundary conditions at node j.

In general nonlinear computations, the inner forces term $I\left(u_i\right)$ as well as its linearisation, the stiffness matrix $K\left(u_i\right) = \frac{\partial I}{\partial u}|_{u_i}$, depend on the current deformation u. The damping term may also include several nonlinear relations, depending on the underlying material law. For the airspring model, the simplest form of Rayleigh damping was used.

Unless otherwise stated, the following results refer to the solution at the output reference node number 718 (Figure 5.1).

5.2 Communication between Abaqus and Matlab

Abaqus output possibilities yield the deformation vector U, the inner forces vector *NFORC* and the stiffness matrix at each computed time instance. The deformations and forces are written in ascii text format to the Abaqus *dat* file and are read in by a Matlab routine.

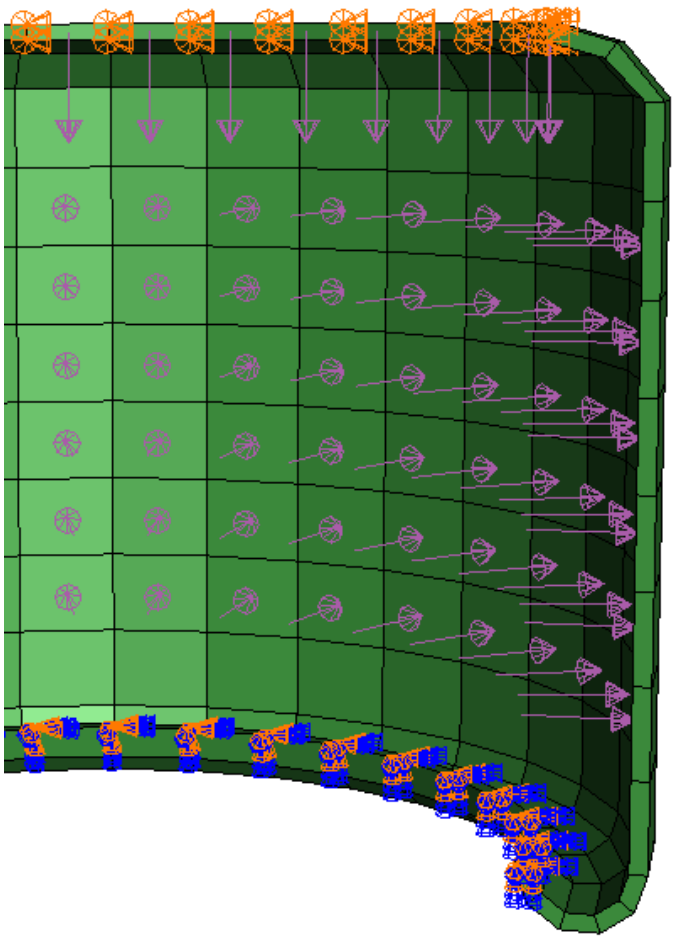


Figure 5.2: Position and direction of inner pressure and boundary conditions

The stiffness matrices $K_i, i = 1 \dots m$ are written to a separate *mtx* text file and are available in element matrix form. The corresponding assembly process is carried out in Matlab in combination with the reading procedure.

Furthermore, the constant mass matrix M is constructed and written in the first iteration of the dynamical step. It is read and assembled analogously to the stiffness matrices. The damping matrix is computed using the Rayleigh coefficient defined in the input ($C = \alpha M$).

The nodal coordinates are read from the Abaqus input file (*inp*).

For external excitations, only distributed loads can be written to an *mtx* file. For a better automatisation of the computation process in this example, the external force is applied to the upper ring by a surface traction load. Note that - in general - point loads are easier to use as they don't have to be read out of the Abaqus output. However, in the case of geometrical nonlinearities, the global direction of external loads might change with the structure's deformation. This case requires special considerations.

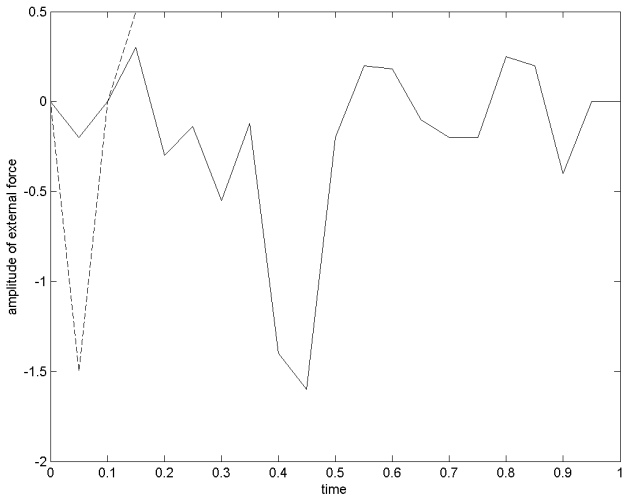


Figure 5.3: Amplitude of the external load over time: the dashed line denotes the training input and the continuous line denotes the input for the reference computation

5.3 Results of the Reduced Model

In a first run with the full model, the lookup table is generated. The airspring is objected to a large tensile force followed by a large compressive force to cover the relevant range of deformation. The step is finished at time $t = 0.15$ after a rather generous number of 150 equally sized increments ($\Delta t = 10^{-3}s$).

For the lookup table, it is not necessary to include all snapshots, as the deformation states are likely to repeat themselves. If possible, the deformation range should be checked and double states be dismissed. For the airspring, this task is relatively simple as the deformation of the nodes in the upper ring can only move in one direction. Yet, the displacement of the nodes in the inner part of the airspring can be different depending on the preceding load history due to dynamic effects.

In this example, the lookup table consists of the states $55-151$ of the dynamic computation step, as this range covers the transition from the largest tension to the largest compression deformation of the upper ring (see Figure 5.3). The computation of the POD basis is done with all deformation snapshots $u(t_i)$ using the L_2 norm. The reduced mass and damping matrices $\widetilde{M}$, $\widetilde{C} = \alpha\widetilde{M}$ and the lookup table $\{p_i, \widetilde{R}_i, \widetilde{K}_i\}_{i=1,\ldots,97}$ are constructed by projection of the respective quantities onto the POD subspace.

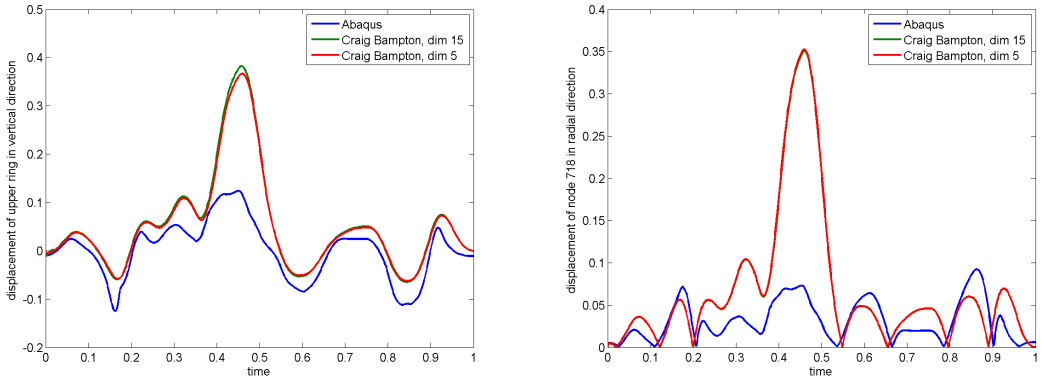
This setup defines the input for the reduced computations.

The static step is not reproduced by the POD model. From the full model, the final state of the inflation step is projected onto the POD subspace to provide the initial conditions $p_0, \dot{p}_0$ for the dynamic computations with the reduced one.

In the following result presentation, five different setups are compared: the full solution

computed with Abaqus, the POD reduced lookup table approach using Jacobians (*Lookup 1*) or Jacobians and Hessian approximations (*Lookup 2*), the POD reduced TPWL approach and a Craig-Bampton method. For the latter one, the model was linearised at the beginning of the dynamic step, i.e., after inflation of the airspring. The Abaqus simulation serves as the reference solution for the reduced models. For the reduced bases, POD and Craig-Bampton, 15 and 5 basis vectors were chosen.

To study the extrapolation abilities of the reduced approaches, the external excitation is adjusted in order to provoke larger deformations than the lookup table includes.



(a) Vertical displacement

(b) Radial displacement

Figure 5.4: Vertical displacement of upper ring and radial displacement of node 718, computed with Abaqus and Craig-Bampton method, using 5 and 15 basis vectors

Figure 5.4 shows the vertical displacement of the upper ring and the radial displacement of the reference node 718 for the Craig Bampton method. For both bases using 15 or 5 eigenmodes, the deformation is entirely different from the full solution. Obviously, the linear approach is not justified in this model setup.

In the case of using 15 basis vectors, the POD methods capture the nonlinear behaviour in a large portion of the time span. However, both the vertical and radial displacements show that all examined methods fail to reproduce the deformations correctly, when the lookup table range is exceeded (Figures 5.5, 5.6). This effect is seen in the periods around $t = 0.18s$, $t = 0.6s$ and $t = 0.85s$.

The solution for the Lookup 2 approach are not plotted for this setup, as the computation experienced numerical instabilities from $t = 0.7s$ on. Except from the three cases mentioned above, the solutions for the TPWL and the Lookup 1 approach produce only small errors in a comparable range.

Using a basis of 5 POD vectors, both Lookup methods and the TPWL method produce similar errors, both for the vertical and the radial component of the displacement (see Figures 5.7 and 5.8). In this case, the Lookup 2 method did not show any instabilities. All methods produce curves that are less smooth than in the case of 15 basis vectors.

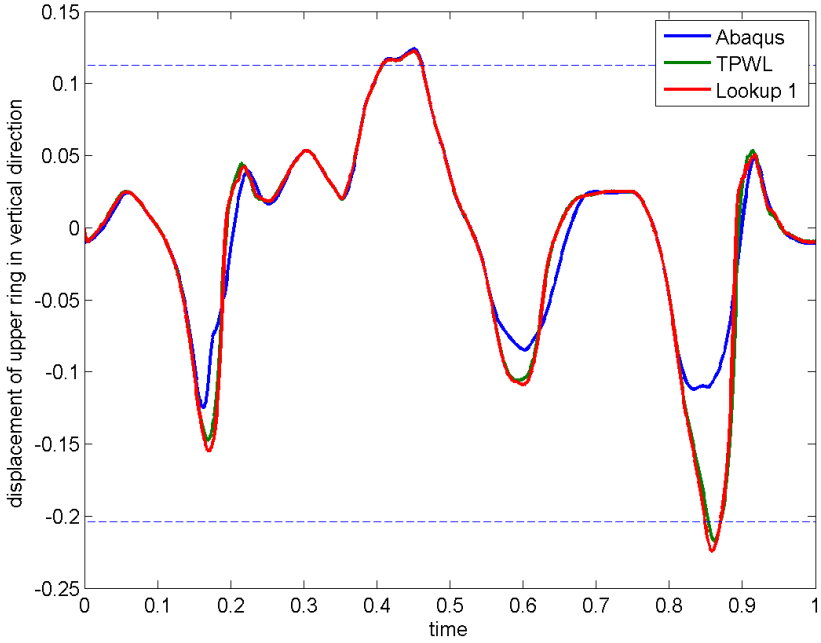


Figure 5.5: Vertical displacement at upper ring, computed with Abaqus, TPWL and Lookup 1, using 15 POD basis vectors
The horizontal dashed lines border the range covered by the lookup table

In other computations, where the deformations remained within the lookup table range, the results of the reduced models were nearly identical to the full solution for all three computed POD and lookup setups.

The following section takes a closer look at the deformation of the full system at time $t = 0.18s$, where the lookup methods do not reproduce the displacements correctly. Around this time period, the airspring undergoes a large compression followed by a rapid change of force direction. Due to inertia effects, the central belt rows of the airspring bulge downwards and develop a deformation that is not included in the training setup (see Figure 5.9). Therefore, neither the POD basis nor the lookup table contain information on this state.

To deal with this shortcoming, an additional training input is set up. The new load case covers a time span of $0.2s$ in 200 time steps and contains a short period of tensile loading after compression. With this, the structure deforms similarly as the full solution at time $t = 0.18s$ of the dynamic step. A new reduced model is defined using 201 snapshots and a lookup table of 147 entries $\{p_i, \widetilde{R}_i, \widetilde{K}_i\}_{i=1,\dots,147}$ for the states $55 - 201$.

Figures 5.10 compare the vertical and radial displacements of the full solution with the reduced one for both the original and the additional training input. The original lookup table consists of 97 entries and the one with the additional input of 147 states. In both

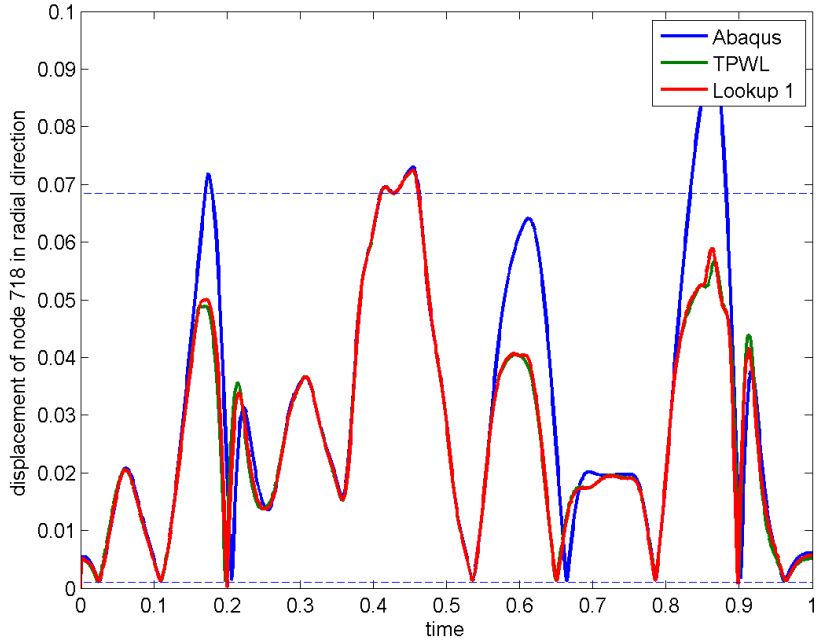


Figure 5.6: Radial displacement at node 718, computed with Abaqus, TPWL and Lookup 1, using 15 POD basis vectors
The horizontal dashed lines border the range covered by the lookup table

cases, the Lookup 1 method was used with a POD dimension of 15. Both figures show a significant improvement not only at position $t = 0.18s$, but also around the critical regions $t = 0.6$ and $t = 0.85$. The listing of errors in Table 5.1 confirms this observation.

Table 5.2 shows a comparison of CPU time for the different approaches using 5 and 15 POD basis vectors. Again, the Lookup 1 method performs slightly faster than the TPWL method. However, all tested reduction methods run at least two orders of magnitude faster in Matlab than the original model in Abaqus.

Despite the computational savings obtained by the reduced methods, the CPU time for the model setup needs to be taken into account (see Table 5.3). The effort basically consists of the communication with Abaqus and the computation of the POD basis. Note that these computations can be done offline and need to be undertaken only once for the model setup and not for each computation. Due to the slow communication with Abaqus via text files, the read steps require most effort - especially the reading and assembling procedure for the full stiffness matrices. This cost can be reduced if an efficient positioning of the lookup table states within the time interval is performed. Then only the relevant matrices need to be output from Abaqus and imported in Matlab. Furthermore, if the POD basis is determined in advance from a preceding run, the element matrices can be read in and projected onto the POD subspace within the same step. Assembly of the matrices is done by adding the projected element matrices.

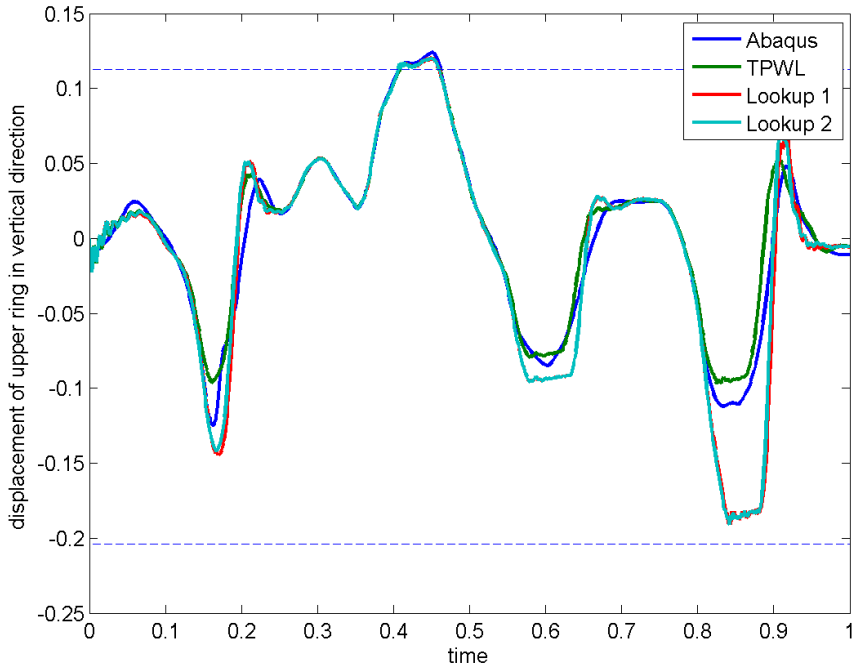


Figure 5.7: Vertical displacement at upper ring, computed with Abaqus, TPWL, Lookup 1 and Lookup 2, using 5 POD basis vectors
The horizontal dashed lines border the range covered by the lookup table

Method	Number of DOF	Number of lookup entries	relative error, L_2 norm	relative error, L_∞ norm
TPWL	15	97	25.1%	25.6%
Lookup 1	15	97	22.1%	25.7%
TPWL	5	97	57.9%	35.1%
Lookup 1	5	97	52.0%	39.9%
TPWL	15	147	7.7%	14.9%
Lookup 1	15	147	4.7%	13.3%
TPWL	5	147	44.0%	34.0%
Lookup 1	5	147	40.4%	33.4%

Table 5.1: Comparison of errors for the Abaqus airspring
The reduced models are based on 151 snapshots and 97 Lookup table entries or 201 snapshots and 147 Lookup table entries, respectively

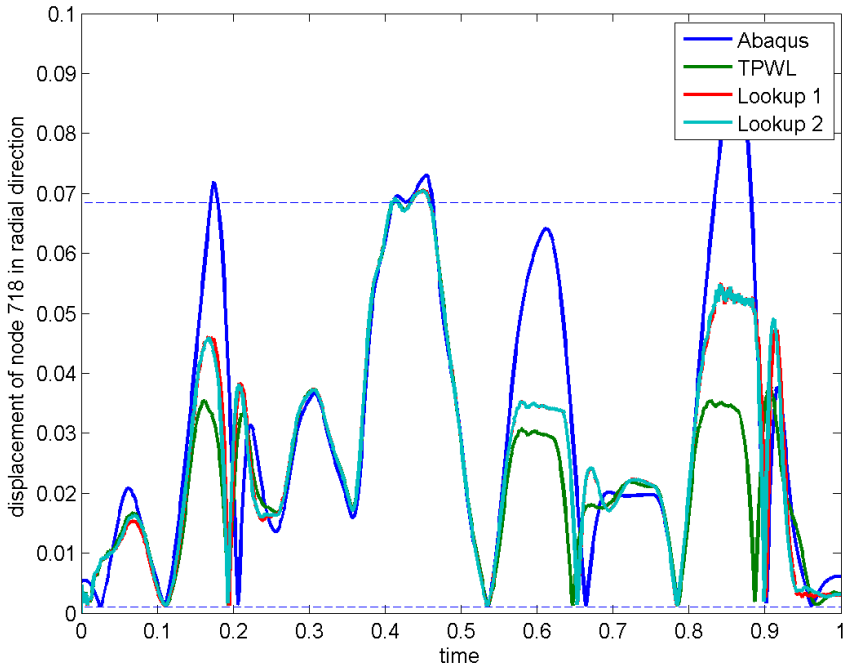


Figure 5.8: Radial displacement at node 718, computed with Abaqus, TPWL, Lookup 1 and Lookup 2, using 5 POD basis vectors
The horizontal dashed lines border the range covered by the lookup table

Method	Number of DOF	CPU Time	Number of DOF	CPU Time
Abaqus	2880	810.4s	2880	810.4s
TPWL	15	3.8s	5	2.9s
Lookup 1	15	1.7s	5	1.3s
Lookup 2	15	4.0s	5	2.4s
Craig-Bampton	15	0.6s	5	0.3s

Table 5.2: Comparison of CPU time for the Abaqus airspring
The reduced models are based on 151 snapshots and 97 Lookup table entries

In conclusion, this section shows that the POD method combined with a lookup table provides a way to significantly reduce CPU time in moderately dimensioned Abaqus computations. Whereas the classical Craig-Bampton method performs slightly faster than the POD methods, it is unable to reproduce any nonlinear effects. In this example, the method yielded completely wrong results.

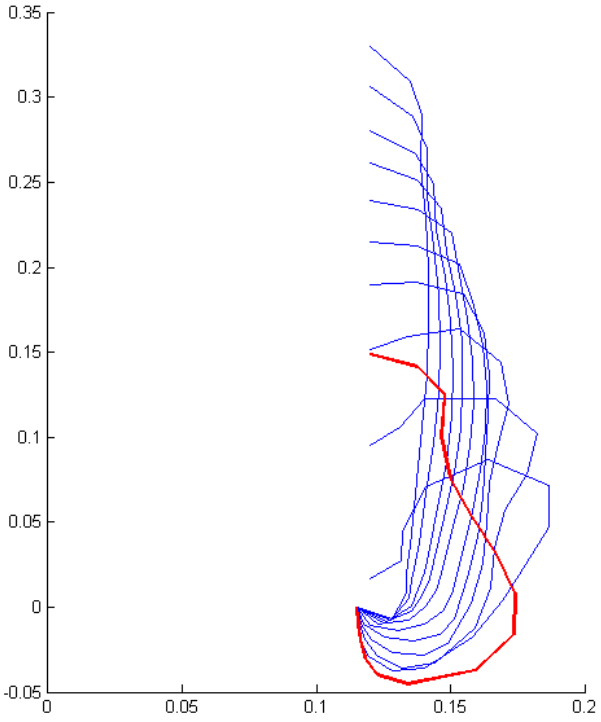
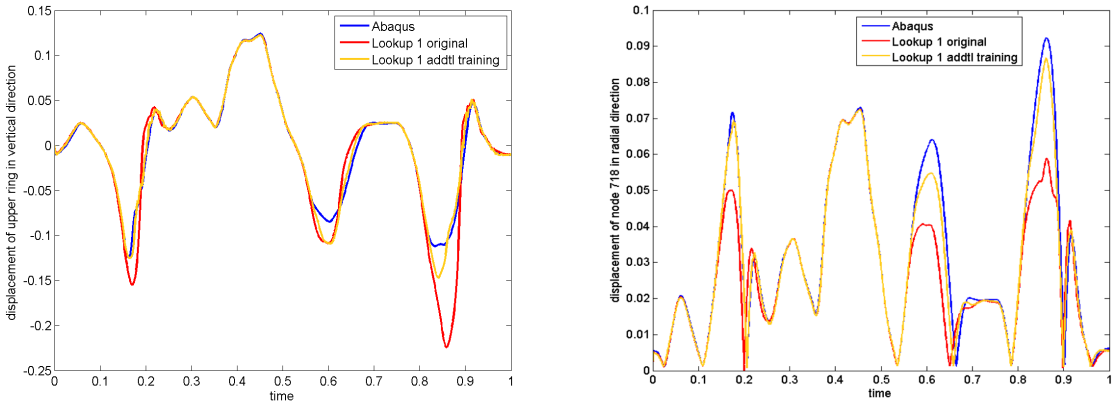


Figure 5.9: Training deformation states $u_i, i = 61, 71, \ldots, 151$ (blue lines) and deformation state of the full solution at time $t = 0.18s$ (red line) in a vertical line along the outer surface of the airspring



(a) Vertical displacement

(b) Radial displacement

Figure 5.10: Radial displacement at upper ring and vertical displacement at node 718, computed with Abaqus, Lookup 1 with original input and Lookup 1 with additional input, using 15 basis vectors

Procedure	CPU Time
Read and assemble stiffness matrices	1011.8s
Read and assemble mass matrix	3.6s
Read deformations	36.4s
Read inner forces	34.5s
Compute POD basis	4.7s

Table 5.3: CPU time for the Abaqus communication and offline computations

The errors of both the TPWL method and the Lookup 1 method remain in a comparable range. Most deformation states can be reproduced by the reduced models, using 15 as well as 5 POD basis vectors. The deviation from the original model increases dramatically when deformations arise that are not included in the snapshot range. Therefore, in all cases, the training excitation must be chosen carefully. Using the additional training input, both methods show very good conformance with the original model in the period of investigation.

The Lookup 1 method performs faster than the TPWL method, which makes the former the method of choice, especially when a large number of lookup states is used.

Using the Lookup 2 method is not recommended in the present configuration due to occurring instabilities.

Conclusion

In the context of Multibody Simulation of full vehicles, the tyres play an important role. Besides the ability to correctly reproduce the highly nonlinear behaviour of a real tyre, an MBS tyre model must fulfil certain performance requirements. Due to the long simulation times typical in vehicle simulations, a tyre model has to work with few degrees of freedom.

Classical tyre models - often based on spring-damper constructions - solve the problem of fast but accurate tyre description. Their disadvantage is the parametrisation which often requires costly measurements. Parametrisation of truck or agricultural tyres constitutes an immense problem, as the test rigs need to be constructed to fit to the larger sizes and to bear high loads.

In contrary to the conventional MBS tyre models, the Finite Element tyre model requires relatively few parameters (geometry and material), which can be further adapted by simple measurements. The large number of degrees of freedom makes it impossible to include the tyre model into an MBS computation without modifications.

The approach proposed in this thesis aims to describe the tyre with a Finite Element approach combined with nonlinear model reduction methods. Unlike most model reduction methods - as the frequently used Craig-Bampton approach - the method of Proper Orthogonal Decomposition (POD) offers a projection basis suitable for nonlinear models. In linear model reduction problems of structural mechanics, the connection between eigenmodes and POD modes for different setups may be a topic for future work.

For the linear wave equation, the POD method was studied comparing two different choices of snapshot sets. Set I consisted of deformation snapshots, and set II additionally contained velocities and accelerations. An error analysis proved no convergence guarantee for deformations only. For inclusion of derivatives it yielded an error bound diminishing for small time steps. The numerical results showed a better behaviour for the derivative snapshot method, as long as the sum of the left-over eigenvalues is significant. The influence of eigenvalue decay on the error shall be further examined. With the POD subspace methods it was also possible to better reproduce the motion than with the eigenmode subspaces.

The reduction of nonlinear problems introduces additional difficulties. The projection-based method of POD reduces the effort needed for solution of the model equations, but not for the function evaluations required for the equation setup. Especially in the case

of collaboration with black-box software, the effort for equation setup exceeds the savings obtained from reduction of the degrees of freedom.

To decouple the reduced surrogate system from the full model, a lookup table approach is presented. It makes use of the preceding computation step with the full model necessary to set up the POD basis. The nonlinear term of inner forces and the stiffness matrix are output and stored in a lookup table for the reduced system. Two approaches are proposed for the treatment of the lookup table, one using only first order approximations of the nonlinear terms (Lookup 1) and one approximating the Hessian tensor (Lookup 2) by using information from the dynamic training step.

Numerical examples include a nonlinear string in Matlab and an airspring computed in Abaqus. Both examples show that effort reductions of two orders of magnitude are possible within a reasonable error tolerance. The lookup approaches perform faster than the Trajectory Piecewise Linear (TPWL) method and produce comparable errors. In the first example of a nonlinear string, the Lookup 2 method clearly yields the best results. The computation of the Lookup 2 reduced airspring induced numerical instabilities for certain lookup and POD combinations. This implies that higher order methods have some potential but can not be recommended in the form proposed here.

Furthermore, the Abaqus example shows the influence of training excitation on the quality of the reduced model. The reduced solution strongly deviates from the full one when dynamical effects occur that were not included in the training. An adapted training input accounting for the missing states yields considerable improvements.

Overall, the numerical examples show that the Lookup 1 method combined with a POD subspace projection constitutes a good method to massively reduce the computational effort of large nonlinear structures. It is faster and more stable than the higher order method Lookup 2 and faster than the TPWL method, especially when many states are included in the lookup table.

Still, the process of communication between black-box software and Matlab requires a lot of time and might be accelerated using a smart choice of states for the lookup table. Moreover, the treatment of the lookup table can be improved concerning stability for higher order approaches.

Further work in the field of model reduction for tyre modelling may focus on the representation of the *contact condition*. In general, contact description is a complex topic in model reduction, which was not addressed within this thesis. For the computation of tyres, a simple contact description based on extended lookup tables might be sufficient but requires thorough investigations.

Appendix A

The Finite Element Tyre Model in Abaqus

The Finite Element software Abaqus offers several possibilities to model a rolling tyre. This section aims to describe one approach yielding an FE model and some computations relevant for applications in Multibody Simulation.

A.1 Parts and Materials

Figure A.1.1 shows the cut through a Finite Element tyre on an angular ground profile. The colours mark different materials used in the model.

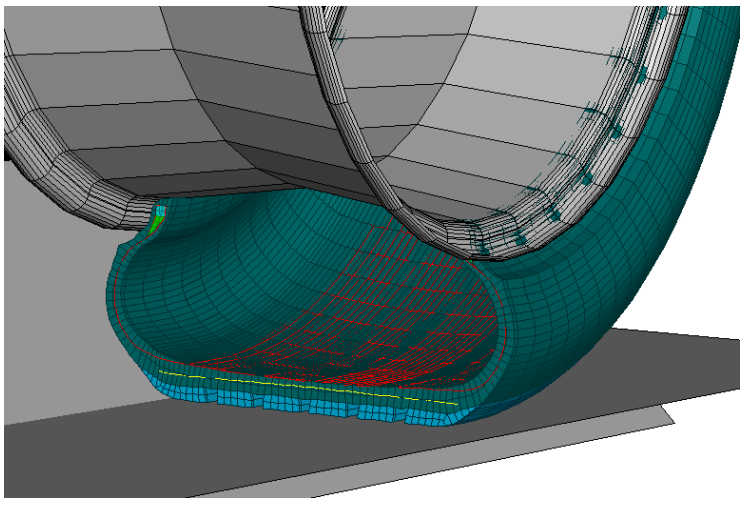


Figure A.1.1: Abaqus model of an automotive tyre

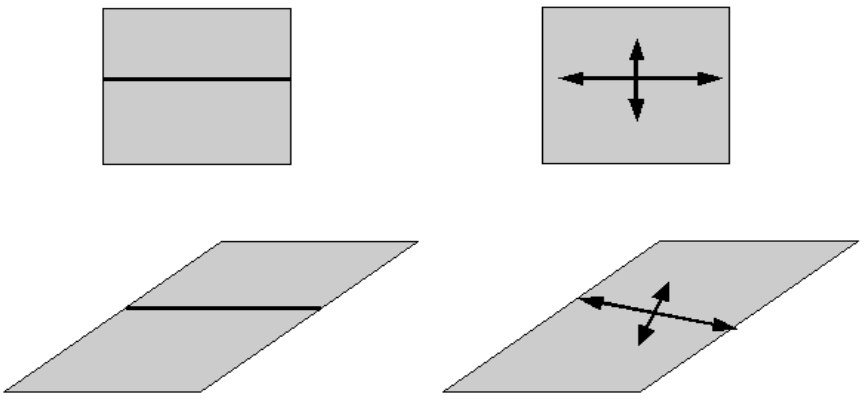
The bead consists of steel, which - under usual tyre strains - remains within the isotropic linear-elastic range of the material. It can therefore be described by its Young's modulus

E and Poisson number μ. The rubber parts of the tyre are divided into three regions: tread, outer and filler rubber. In general, rubber is a highly nonlinear material with a hyperelastic characteristic. Here it is modelled with *Neo-Hooke* or *Mooney-Rivlin* material models ([1]).

For the computation of misuse events, the rim must be modelled with a deformable material with the possibility of plastification or even failure. In general comfort and handling analyses, the rim does not undergo noticeable deformations. Therefore it can either be treated as a rigid body contacting the tyre bead or be left off entirely and replaced by boundary conditions for the rim contacting area. In the latter case, the dynamic analyses include a mass element attached to the tyre centre.

Carcass and belt consist of several plies of cords embedded in the rubber. They are either made of steel or of synthetic fibres. Whereas the carcass is usually built up radially (radial tyre), the belt plies lie in direction of travel or at an angle of $0 - 60°$ to the midline. Succeeding plies lie at an opposite angle to each other for a balanced stress distribution.

Due to the rather small strains, carcass and belt can be modelled with linear elastic material. Abaqus provides the element type *rebar*, which includes the cord ply into the rubber continuum element. The reinforcing fibres are not treated as individual components but their stresses and strains are included in the formulation of inner forces and stiffness matrix of the surrounding continuum element. Note that - in contrary to an orthotropic material formulation - the rebar elements retain their position and direction when the element is sheared A.1.2.



(a) Rebar element (b) Orthotropic material

Figure A.1.2: Continuum elements under shear deformation: Direction of rebar and orthotropic axes

A.2 Procedures

A.2.1 Model Setup

The practical procedure for setting up a Finite Element tyre model is as follows: If the tyre geometry exists as a CAD-file, it can be imported directly into the Finite Element software. In the case of possessing the tyre itself, it can be cut open by water streams to obtain a clean cut through the cross section. This slice is photographed and the image properly processed to pass the geometry to Abaqus. Figure A.2.1 shows an image of the cut through a truck tyre and the respective meshed 2D cut model in Abaqus.

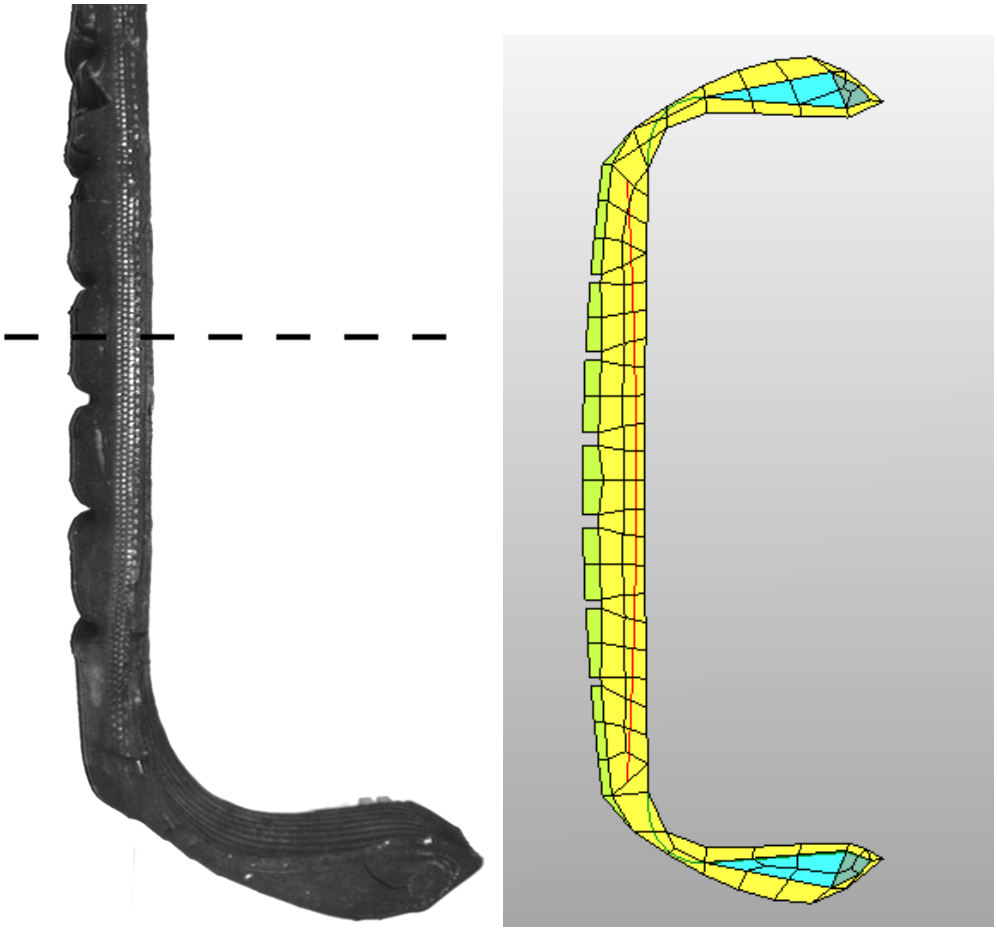


Figure A.2.1: Photograph of the cut through a truck tyre and respective FEM mesh

If the parameters of the material models are unknown, a sample of the rubber can be cut out of the tyre and determined by tensile tests. Another possibility is to adapt the material properties until measurement results (e.g. eigenfrequencies) are reached.

Once the materials are defined in their respective geometric areas, the tyre cross section is meshed in Abaqus. If the tread is rotationally symmetric, the section is revolved by 360°

to define the full $3D$ model. For non-symmetric tyre profiles like in agricultural machines, the tread needs to be modelled separately on a section of the tyre and later be revolved to the full circumference.

A.2.2 Computed Load Cases

Inflation

The first step to be computed is the inflation process. For problems that involve acoustic computations, the air inside the tyre needs to be modelled separately, as the waves spreading through the inner volume are a crucial factor. If the focus of the computation is on the mechanical properties, it is generally sufficient to apply the inner pressure as a constant distributed load on the inner surface of the tyre.

For rotationally symmetric tyres, the inner pressure can be applied to the two-dimensional cut model of the structure. This procedure is computed in a rather short time as the number of degrees of freedom is very small. The resulting stresses and strains in the elements can be transferred to the full three-dimensional model in the revolving step. Abaqus applies some small corrections on the results in the full model, but the effort for the inflating procedure is significantly reduced.

Ground Contact

After inflation, the tyre is pressed onto the contacting surface. The surface can either be rigid or deformable and can represent part of a road or a test rig (e.g. a rotating drum). The contact pressure serves as a possibility to check agreement with footprint measurements.

The contact condition can be formulated as *Lagrange* or *penalty* contact. In short, penalty formulations allow the contacting surfaces to penetrate each other and the contacting force is calculated by the penetrated volume and a given contact stiffness. If this contact stiffness is set very high, numerical instabilities are likely to occur. Methods using Lagrange multipliers strictly enforce the contact constraint, but introduce additional degrees of freedom into the system. For details on contact modelling in Finite Element computations, see, for example, [29]. By default, Abaqus uses an augmented Lagrange method that switches between a penalty and a Lagrange formulation depending on the applied contact stiffness and its accompanying numerical properties ([1]).

In general, the contacting step is computed using a static formulation. Therefore, the contact condition in the tread is formulated with static properties only.

Eigenmodes

Eigenfrequencies and eigenmodes can be computed on the free or on the loaded tyre in contact with the ground. Depending on the application, the system is linearised at the

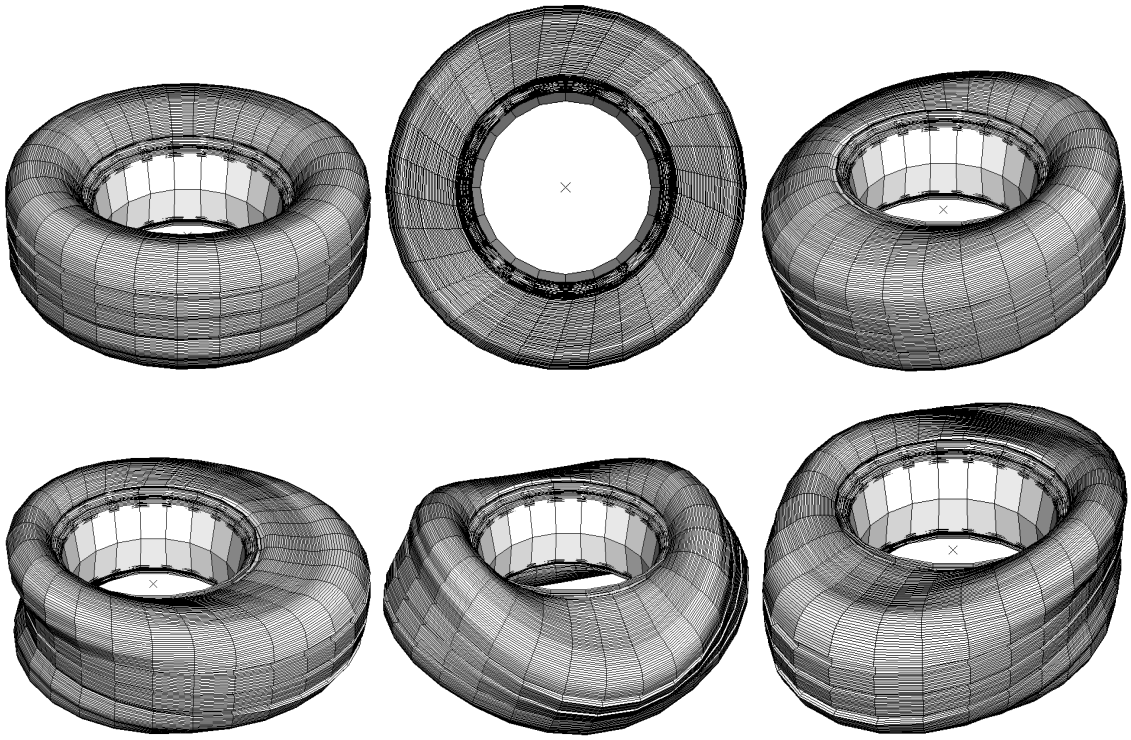


Figure A.2.2: Finite Element result of an eigenmode analysis

respective set-up. Figure A.2.2 shows six eigenmodes of a free Finite Element tyre.

Steady State Handling Computations

For the study of handling behaviour, a steady-state step is computed where inertia effects are ignored. The tyre is rolling on the ground with a constant rotational velocity and a constant forward velocity of the rim. Depending on their relation, braking and accelerating as well as cornering can be described. The underlying theory is based on a mixed Euler-Lagrange approach where the deformation part of the energy is accounted for in global coordinates whereas the velocity effects are represented in a steadily rotating coordinate system. It can be regarded as a formulation where the material flows through the tyre's mesh [1].

The assessment of side effects is done by applying a small rotation angle to the rim midsurface around the vertical axis. Furthermore, the influence of camber angle (angle between rim midsurface and vertical axis) on the side forces can be studied.

Transient Analyses

Inertia effects become important when comfort or misuse analyses are carried out. These types of simulation require a fully transient approach.

However, the steady-state computations can serve as initial conditions for a transient run. If the main focus of the computation does not lie on representation of the acceleration process, the tyre can be input from a previous handling step. This ensures that initial deformations, velocities and forces are in equilibrium at the beginning of the cleat test without the need to compute a long acceleration process.

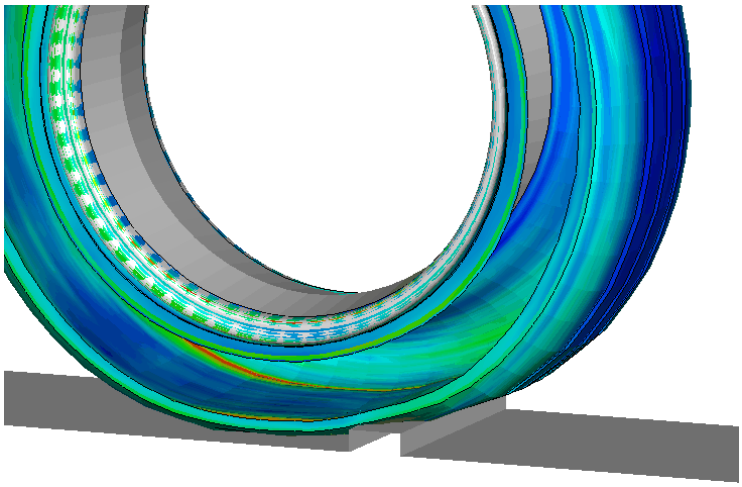


Figure A.2.3: Finite element tyre rolling over a cleat, computed with Abaqus

Depending on the manoeuvre, implicit or explicit algorithms can be used. Due to the higher efficiency in parallel computing and the large deformations, often the explicit formulation is preferred. Note that, in Abaqus, the transfer between the Standard and Explicit solver package requires some caution and experience, as not all model properties defined in Standard are recognised in the same way by the Explicit solver ([1]).

Figure A.2.3 shows the simulation of a Finite Element tyre rolling over a cleat.

A.2.3 Computations for Parametrisation of MBS Tyres

The procedures relevant for Multibody Simulations include the static contact computations, eigenmode analysis, handling and cleat test computations.

Static contact computations can be used to determine stiffness parameters for existing MBS tyre models, like vertical or lateral tread stiffness in tension, compression or bending. Among others, these parameters are necessary for the input of *RmodK30*.

Furthermore, the results can also serve as a reference for replacement formulations to compensate shortcomings of the classical tyre models. An example can be seen in [47]. In

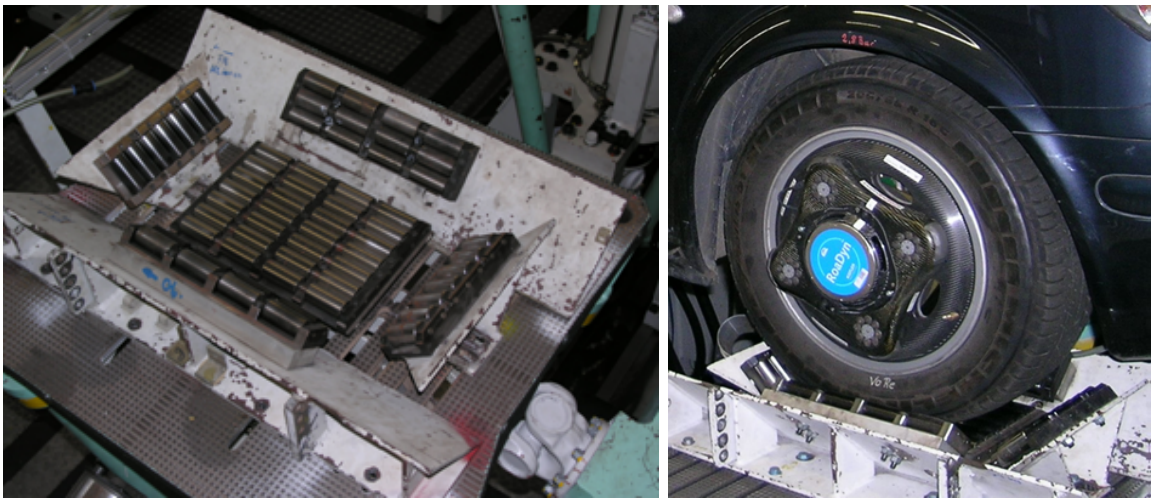
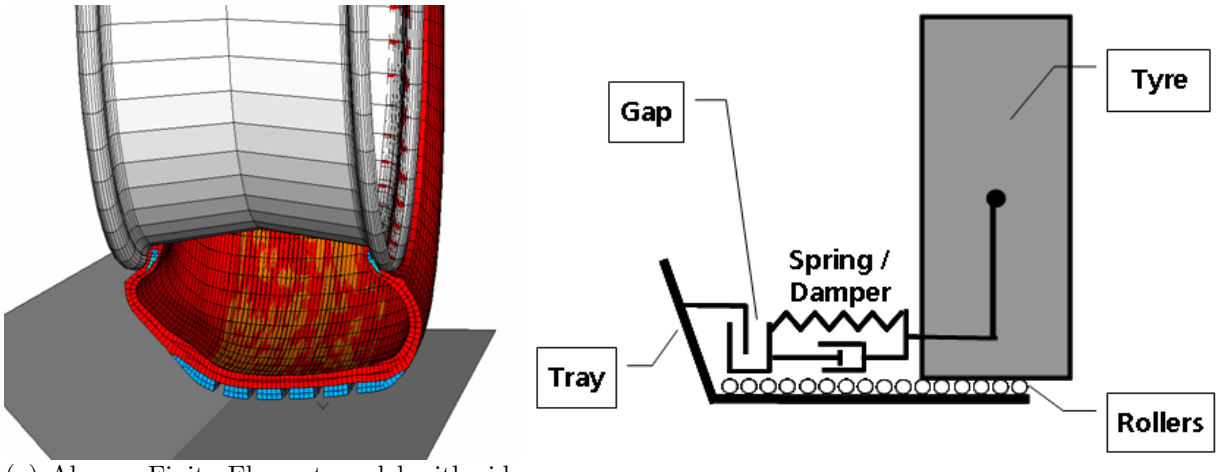


Figure A.2.4: Tray for tyre contact on a vehicle test rig (see [47])

order to model a vehicle on a test rig (see Figure 1.2), an appropriate description of the tyres' excitation from the test rig is necessary. Here, the lateral forces are dominated by impacts from the inclined plates on the sides of the trays (see Figure A.2.4). Such excitation can not be represented by a tyre model like *RmodK30*. Therefore, the two-dimensional tyre model was additionally equipped with lateral one-sided springs between rim and tray to reproduce the side plate impacts as illustrated in Figure A.2.5. The stiffness parameters of the spring were determined after a respective Finite Element computation.



(a) Abaqus Finite Element model with side impact

(b) Replacement formulation using a spring model

Figure A.2.5: Enhancement of a tyre model using a lateral contact spring (see [47])

The computation of eigenmodes and the handling analysis can be used directly as input for, e.g., *RmodK* or *Magic Formula* tyre models. Eigenmode measurements require rather low

effort, even for large tyres. Thus, the parameters of the Finite Element tyre can be further adapted referring to these test. The model validated in this manner can be employed for the simulation of more expensive or complicated measurements.

Cleat test computations can be used to optimise the overall parameters of MBS tyre models to match the Finite Element solution.

A.3 Newmark's Time Integration Scheme

In the following section, *Newmark's method* for the case of $\beta = 0.5$ and $\gamma = 0.25$ is shortly described. By this choice of parameters, the scheme is precise and unconditionally stable [14]. It is the default method for linear and nonlinear dynamic computations used in Abaqus Standard ([1]). For a more general description of the method with different choices of parameters, see [14].

Newmark's time integration scheme is customised for equation systems arising in structural mechanics:

$$M\ddot{u} + C\dot{u} + Ku = f_{ext} \tag{A.3.1}$$

where M, C and K are the mass, damping and stiffness matrices arising from a semi-discretised continuous system or from a mass-spring system with damping.

A crucial assumption of the method is that the acceleration $a = \ddot{u}$ remains constant throughout the step:

$$a(\tau) = \frac{1}{2}\left(a_n + a_{n+1}\right), \quad \tau \in [0, \Delta t] \tag{A.3.2}$$

where $a_n = a(t_n)$.

Integration over time yields a linear path for the velocities $v = \dot{u}$ and a quadratic one for the displacements u within the time interval $[t_n, t_{n+1}]$:

$$v(\tau) = \frac{\tau}{2}\left(a_n + a_{n+1}\right) + v_n \tag{A.3.3}$$

$$u(\tau) = \frac{\tau^2}{4}\left(a_n + a_{n+1}\right) + \tau v_n + u_n \tag{A.3.4}$$

which can be evaluated at the end of the time step and written in terms of the displacement u_{n+1}:

$$v_{n+1} = \frac{2}{\Delta t}\left(u_{n+1} - u_n\right) - v_n \tag{A.3.5}$$

$$a_{n+1} = \frac{4}{\Delta t^2}\left(u_{n+1} - u_n\right) - \frac{4}{\Delta t} v_n - a_n \tag{A.3.6}$$

Setting these formulae into the equilibrium equation A.3.1 at time t_{n+1}, a linear system of equations can be solved directly for the unknown displacement u_{n+1}:

$$\left[\frac{4}{\Delta t^2}M + \frac{2}{\Delta t}C + K\right] \cdot u_{n+1} = \\ f_{ext} + M \cdot \left(\frac{4}{\Delta t^2}u_n + \frac{4}{\Delta t}v_n + a_n\right) + \left(\frac{2}{\Delta t}u_n + v_n\right) \tag{A.3.7}$$

Bibliography

[1] Abaqus Theory Manual, Version 6.6

[2] M. Arnold: Zur Theorie und zur numerischen Lösung von Anfangswertproblemen für differentiell-algebraische Systeme von höherem Index, Reihe 20, Nr. 264., VDI Verlag Düsseldorf, 1998

[3] S. Ali, M. Damodaran, K.E. Willcox: Approximate Low Dimensional Models based on Proper Orthogonal Decomposition for Black-Box Applications, Paper Presented at the Singapore-MIT Alliance Symposium, Singapore, 2007

[4] H. Alzer: Discrete analogues of a Gronwall-type inequality, Acta Mathematica Hungarica 72 (3), pp. 209-213, 1996

[5] P. Astrid: Reduction of Process Simulation Models: a Proper Orthogonal Decomposition Approach, PhD Thesis, Technische Universiteit Eindhoven, Netherlands, 2004

[6] P. Astrid, S. Weiland, K. Willcox, T. Backx: Missing Point Estimation in Models Described by Proper Orthogonal Decomposition, 43rd IEEE Conference on Decision and Control, Paradise Island, Bahamas, 2004

[7] G. Berkooz, P. Holmes, J. L. Lumley: Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge Monogr. Mech., Cambridge University Press, Cambridge, UK, 1996

[8] K. E. Brenan, S. L. Campbell, L. R. Petzold: Numerical Solution of Initial-Value Problems in Differential Algebtraic Equations, Elsevier, North Holland, New York, N.Y., 1989

[9] CD-Tire User's Manual, LMS Virtual Lab

[10] Y. Chahlaoui, D. Lemonnier, A. Vandendorpe, P. Van Dooren: Second Order Balanced Truncation, Linear Algebra and its Applications, 415, 2004

[11] Continental AG: Reifengrundlagen PKW, 2008/09, www.conti-online.com

[12] R. R. Craig, Jr.: Coupling of Substructures for Dynamic Analyses: an Overview, AIAA-2000-1573, 2000

[13] R. Dautray, J.-L. Lions: Mathematical Analysis and Numerical Methods for Science and Technology, Volume 5, Evolution Problems I, Springer Verlag, 1992

[14] R. Dautray, J.-L. Lions: Mathematical Analysis and Numerical Methods for Science and Technology, Volume 6, Evolution Problems II, Springer Verlag, 1993

[15] S. Dietz: Vibration and Fatigue Analysis of Vehicle Systems Using Component Modes, Dissertation TU Berlin, Reihe 12, Nr. 401, VDI Verlag, 1999

[16] E. Eich-Soellner, C. Führer: Numerical Methods in Multibody Dynamics, Teubner, 1998

[17] P. Février, G. Fandard: A New Thermal and Mechanical Tire Model for Handling Simulation, Reifen - Fahrwerk - Fahrbahn, VDI-Berichte Nr. 2014, 2007

[18] H. Fleischer: Nichtlineare Schwingungen, Höhere Technische Mechanik II/Schwingungslehre, UniBw München

[19] F. Gipser: Ftire, a Physically Based, Application-Oriented Tire Model for Use with Detailed MBS and FEM Suspension Models, www.ftire.com, 2006

[20] E.J. Grimme: Krylov Projection Methods for Model Reduction, PhD Thesis, University of Illinois at Urbana-Champaign, 1997

[21] G. Heinrich, M. Klüppel: Rubber Friction and Tire Traction, Reifen - Fahrwerk - Fahrbahn, VDI-Berichte Nr. 2014, 2007

[22] M. Hinze, S. Volkwein: Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition, Report No. 2/2005, Graz University of Technology and University of Graz, 2005

[23] C. v. Holst: Vergleich von Reifenmodellen zur Simulation der Fahrdynamik von Traktoren, Reihe 14, Nr. 102, VDI Verlag, Düsseldorf, 2001

[24] M. Kleiber: Parameter Sensitivity in Nonlinear Mechanics, Wiley, 1997

[25] P. Krysl, S. Lall, J.E. Marsden: Dimensional Model Reduction in Non-linear Finite Element Dynamics of Solids and Structures, Int. J. Numer. Meth. Engng, 51, 479-504, 2000

[26] K. Kunisch, S. Volkwein: Galerkin proper orthogonal decomposition methods for parabolic problems, Numer. Math. 90: 117-148, 2001

[27] K. Kunisch, S. Volkwein: Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM J. Numer. Anal., Vol. 40, No.2, 492-515, 2002

[28] S. Lall, P. Krysl, J.E. Marsden: Structure-Preserving Model Reduction of Mechanical Systems, Physica D: Nonlinear Phenomena, Volume 184, Issues 1-4, 304-318, 2003

[29] T.A. Laursen: Computational Contact and Impact Mechanics, Springer, 2002

[30] M. Lehner: Modellreduktion in elastischen Mehrkörpersystemen, PhD Thesis, University of Stuttgart, Germany, 2007

[31] B. Lohmann, B. Salimbahrami: Reduction of Second Order Systems using Second Order Krylov Subspaces, IFAC World Congress, Prag, Czech Rep., 2005

[32] D.G. Meyer, S. Srinivasan: Balancing and Model Reduction for Second Order Form Linear Systems, IEEE Transactions on Automatic Control, pp. 1632-1645, November, 1996

[33] M. Meyer: Reduktionsmethoden zur Simulation des aeroelastischen Verhaltens von Windkraftanlagen, PhD thesis, University of Braunschweig, Germany, 2002

[34] Michelin: Der Reifen - Haftung / Komfort / Rollwiderstand und Kraftstoffersparnis, Société de Technologie Michelin, Clermont-Ferrand, France, 2005

[35] A. Newman, P.S. Krishnaprasad: Nonlinear model reduction for RTCVD, in: Proceedings of the 32nd Conference in Information Sciences and Systems, Princeton, NJ, 1998

[36] R. Pinnau, A. Schulze: Model Reduction Techniques for Frequency Averaging in Radiative Heat Transfer, to appear in JCP, 2008

[37] M. Rathinam, L, Petzold: A New Look at Proper Orthogonal Decomposition, SIAM J. Numer. Anal., Vol. 41, No. 5, pp. 1893-1925, 2003

[38] M. Rewieński: A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems, PhD thesis, Massachusetts Institute of Technology, 2003

[39] W.H.A. Schilders, H. van der Vorst: Model Order Reduction: Theory, Research Aspects and Applications, Springer, 2008

[40] A.J.C. Schmeitz: A Semi-Empirical Three-Dimensional Model of the Pneumatic Tyre Rolling over Arbitrarily Uneven Road Surfaces, PhD thesis, Technical University of Delft, 2004

[41] B. Simeon: Numerische Simulation gekoppelter Systeme von partiellen und differential-algebraischen Gleichungen in der Mehrkörperdynamik, Reihe 20, Nr. 325, VDI Verlag, Düsseldorf, 2000

[42] L. Sirovich: Turbulence and the dynamics of coherent structures I—III, Quart. Appl. Math., 45(3):561–590, 1987

[43] D.C. Sorensen, A.C. Antoulas: Gramians of Structured Systems and an Error Bound for Structure-preserving Model Reduction. In P. Benner, V. Mehrmann, and D. Sorensen, editors, Dimension Reduction of Large-Scale Systems, volume 45 of Lecture Notes in Computational Science and Engineering, pages 117-130, Springer-Verlag, Berlin, Heidelberg, 2005

[44] A. Vandendorpe: Model Reduction of Linear Systems, an Interpolation Point of View, PhD Thesis, Catholic University of Louvain, Belgium, 2004

[45] S. Volkwein: Proper orthogonal decomposition and singular value decomposition, SFB-Preprint No. 153, 1999

[46] T. Voß: Model reduction for nonlinear differential algebraic equations, M.Sc. Thesis, University of Wuppertal, Germany, 2005

[47] N. Weigel, S. Weihe, G. Bitsch, K. Dreßler: Einsatz von Simulationswerkzeugen zur Auslegug und Optimierung von Prüfkonzepten, DVM-Report 133, 33. Tagung des DVM-Arbeitskreises Betriebsfestigkeit, S. 163-176, Steyr, 2006

[48] K. Willcox: Unsteady Flow Sensing and Estimation via the Gappy Proper Orthogonal Decomposition, Proceedings of the 5th SMA Symposium, 2004

[49] K. Willcox, J. Peraire: Balanced Model Reduction via the Proper Orthogonal Decomposition, AIAA Journal Vol. 40, No. 11, 2002

[50] P. Wriggers: Nichtlineare Finite-Element-Methoden, Springer, 2001

[51] O.C. Zienkiewicz, R.L. Taylor: The Finite Element Method, Volume 1: The Basis, Butterworth-Heinemann, 2000

[52] O.C. Zienkiewicz, R.L. Taylor: The Finite Element Method, Volume 2: Solid Mechanics, Butterworth-Heinemann, 2000

Curriculum Vitae

Sabrina Herkt

December 19, 1980	born in Erlangen

Education

1986 - 1990	Primary school in Rosenheim
1990 - 1999	Ignaz-Günther-Gymnasium (grammar school) in Rosenheim
10/1999 - 04/2005	studies in Technomathematics at the Technical University of Munich (TUM)
12/2001 - 05/2002	semester abroad at the National University of Singapore (NUS)
04/2005	Diploma in Technomathematics
05/2005 - 12/2008	PhD studies in Mathematics at the University of Kaiserslautern
05/2005 - 09/2008	PhD scholarship at the department *Mathematical Methods in Dynamics and Durability* at the Fraunhofer Institute for Industrial Mathematics (ITWM) Kaiserslautern