Efficient time integration and nonlinear model reduction for incompressible hyperelastic materials

Urs Becker

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

Tag der Disputation: 30. November 2012

Gutachter:
Prof. Dr. Bernd Simeon
Prof. Dr. Claus Führer

D386

to the Cookie Monster

Acknowledgments

I want to thank the people of the Fraunhofer Institute for Industrial Mathematics ITWM, those who worked with me in the department for “Mathematical Methdods in Dynamics and Durability” and especially my thanks go to Sabrina Herkt, Clarisse Weischedel, Oliver Weinhold, Rheinhard Priber, and Martin Obermayr.

This thesis would not have been possible without Dr. Klaus Dressler, who envisioned the topic and setting, and my supervisors Prof. Dr. Bernd Simeon and Prof. Dr. Tobias Damm. I want to thank them for their support, my special thank goes to Prof. Dr. Bernd Simeon for his strong focus, the helpful discussions and the motivations he gave me throughout this thesis.

This work was partly funded by the German Federal Ministry of Education and Research, within its applied mathematical funding program. The associate project is called SNiMoRed, which is a German abbreviation and stands for “Multidisziplinäre Simulation, nichtlineare Modellreduktion und proaktive Regelung in der Fahrzeugdynamik”. Additional it was supported by Fraunhofer Institute for Industrial Mathematics and the “Fraunhofer-Gesselschaft zur Förderung der angewandten Forschung e.V.”. My thanks also go to the “Vibracoustic GmbH&Co. KG” for providing a detailed finite element model of a rubber bushing.

Contents

Chapter One

Motivation

> *Make no little plans. They have no magic to stir men's blood and probably themselves will not be realized. Make big plans; aim high in hope and work, remembering that a noble, logical diagram once recorded will never die, but long after we are gone will be a living thing, asserting itself with ever-growing insistency. Remember that our sons and grandsons are going to do things that would stagger us. Let your watchword be order and your beacon beauty.*
>
> Think big.
>
> *Daniel Burnham*

Let us consider this work in a bigger context to demonstrate our motivation. A classic mechanical system consists of many connected parts, think of a train, a car, a bicycle or some other type of machine. The parts of the machine are, e.g., for a car its chassis, outer frame, wheels, transmission, the brake and so on, all connected into one big system, the car. The dynamics of such a system are the movements of all parts while using the machine, e.g., driving a car along some road. An approach to simulate the dynamics is to assume that all parts inside the machine are rigid and thus non deformable and somehow connected through joints. So the wheels of the car are connected to the chassis such that they can rotate along their axis but they are also connected to the gearbox to accelerate them for driving. The chassis can rotate the anchor point of the wheels for steering and is thus connected to the steering system, the steering system is connected to the steering wheel, and so on.

However, we notice that not every element can be considered rigid, there is an important exception, the so called force elements. A force element represents a special kind of connection, to attain a specific relative displacement of two parts, connected by a force element, a force is needed. For example consider the suspension of the car, you need some force to attain a deformation.

One of the simplest kind of force elements are linear springs, a change in length requires some force, doubling the force doubles the change in length, but force elements can easily become more difficult. In the following we will get more specific and take a look into the front axle of a car (Figure 1.1). In particular, we see several steel parts connected by rubber bushings. One of these bushings can be seen in Figure 1.2, this is also a force element but it has a nonlinear force-displacement relation, acts in multiple directions and has a changing topology dependent on how large the displacement of the inner ring is. Until this point the described approach is called *multibody simulation*.

To accurately model the behavior of such a rubber bushing a totally different approach has to be used. Assuming that for a small block of rubber we know how force is transferred into deformation, we call the small block an element. This local knowledge can then be used to discretize the whole part into such small elements

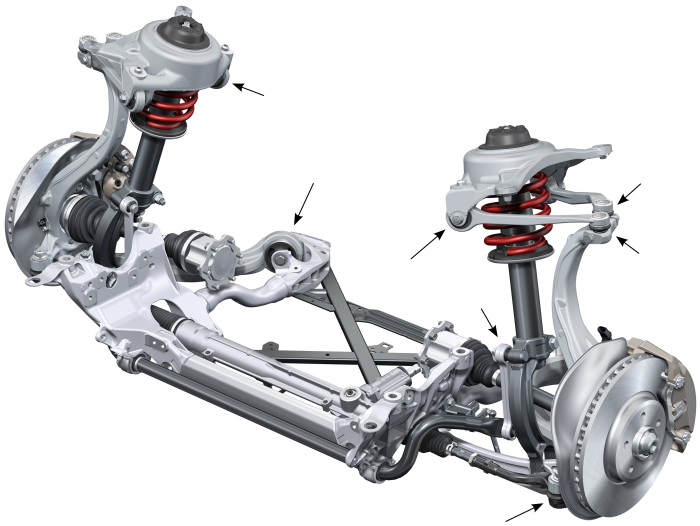


Figure 1.1: Front-axle of a car [SHKH10] with several rubber bushing elements marked by arrows

to estimate the whole force-displacement relation. The approach is known as the *finite element method*.

In the example of a rubber bushing the finite element model has to incorporate, on the one side, that rubber is very flexible and thus very large deformations can occur and, on the other that side, that rubber is incompressible in its volume. Note that the rubber-bushing is only one example of a force element, others could be the wheels for driving on the road, or we could be considering larger steel parts also as deformable, and so on. We see that each force-element needs a special handling in its modeling. A quite systematical approach is the use of the finite

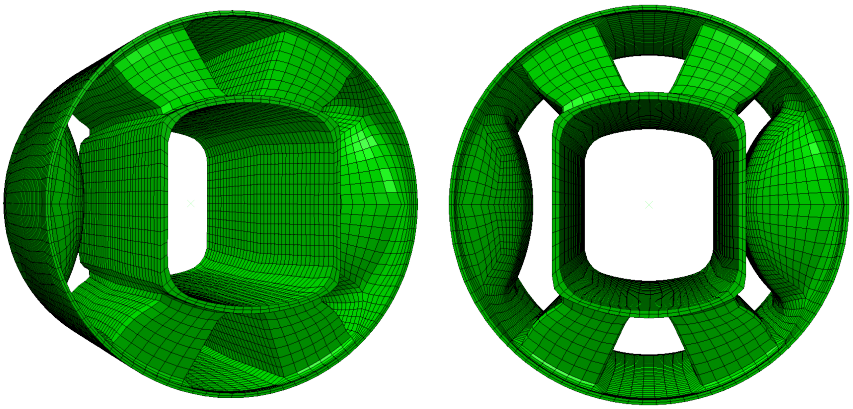


Figure 1.2: Detailed finite element model of one rubber bushing used in a front axle

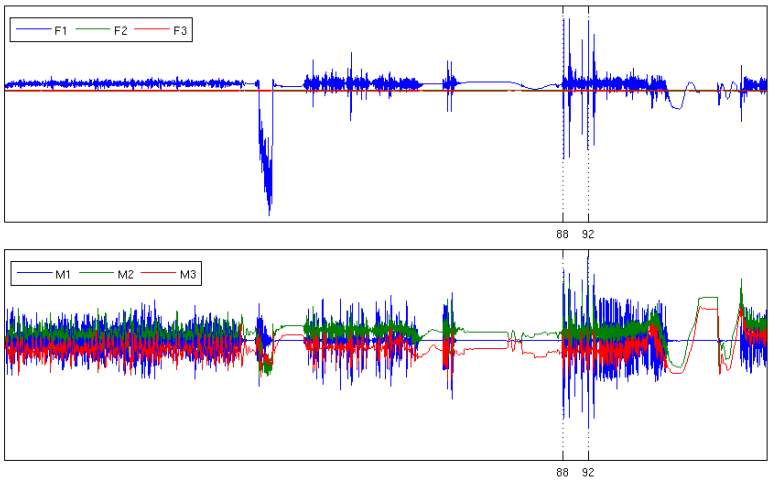


Figure 1.3: 2 minute - 6D time trajectory for forces and moments

element method.

By trying to include a rubber bushing modeled by finite elements into a multibody system we face multiple problems. First of all the finite element model uses many elements to estimate the part's geometry in its deformed and undeformed configuration. Thus we face a problem dimension which has more then 10^5 degrees of freedom. Secondly because of the nonlinear material characteristic and the large deformations occurring, an approach by using linearizations and eigenmodes of the part is not possible.

Further the multibody system lives in an other time scale than the finite element model, because of the difference in degrees of freedom. A multibody model can be used to simulate the dynamics of a system for several minutes, whereas a finite element model can only be used for the simulation of several seconds. In Figure 1.3 we show a 2 minute excitation which is calculated for the rubber bushing. A simulation of such an excitation would cost several weeks of computation time using the finite element model.

To be able to connect these two worlds, we need to find a way to simulate a long time-excitation by the finite element model of the rubber bushing. We think, this shall be somehow possible since the excitations observed are not all unique throughout the simulation. So that they may be reused, and also we think that the occurring deformations of a rubber bushing are not in need of 10^5 degrees of freedom. There are deformations which are more reasonable than others (e.g., consider one note which is pulled out of the part while all others stay in place, this is a deformation which is rather unlikely). This knowledge shall be utilized for a reduced model.

Overview

In this thesis we are going to concentrate on the case of a rubber bushing and discuss what can be done to reach the goal of simulating long time excitations within a finite element model.

The work consists of two parts. In the first part we want to discuss the modeling of incompressible hyperelastic (rubber like) materials to see where the computational costs arise and how a more efficient integration is possible. The second part is about nonlinear model reduction which can be utilized to reduce the large dimension of a finite element model and to further reduce the simulation time.

Starting with Part I, Chapter 2, gives a short introduction to continuum mechanics. We describe the linear and nonlinear modeling of a structural dynamical problem and introduce the hydrostatic pressure which is necessary to model incompressible materials. We discretize and linearize the system to bring it into a form which allows for a numerical treatment of the problem.

In Chapter 3 we start our mathematical investigation by introducing the notion of singularly perturbed systems. We will give some examples of perturbed systems, as they naturally arise whenever a constraint is exchanged by a very stiff spring and give some of their properties. Afterwards we show how an incompressible hyperelastic structural dynamical problem can be interpreted as a singular singularly perturbed system.

Chapter 4 handles the numerical treatment of the introduced singular singularly perturbed systems. In the case of Runge-Kutta methods an order reduction is observed, by using so called linear implicit or Rosenbrock methods we will see an advantage. We introduce these methods and give conditions which have to be fulfilled for convergence in the case of singular singularly perturbed systems by examination of an appropriate test equation.

An efficient implementation of the introduced Rosenbrock methods is given in Chapter 5. We pay special attention to achieve a good performance for large second order systems. The attained performance is afterwards compared to a common method for structural dynamical problems. Notice that by efficiency we don't head for a real-time application but for relatively fast simulations.

Increasing the simulation time of our full finite element model can only be a small step towards the inclusion into a multibody system. Since so far we have not handled the huge dimensions, those bring us to Part II, which is a nonlinear model reduction. For the obtained reduced model an inclusion into some multibody system shall be possible, while recovering feasible simulation times.

We describe how to do a model reduction of a nonlinear system in Chapter 6. There we motivate the use of the proper orthogonal decomposition and show how the method can be applied to structural dynamical systems. For a further reduction of simulation time we describe the use of an additional lookup method.

In Chapter 7 we will show some of the simulation results which can be attained using the described model reduction. In the end we apply the method to a large finite element model and show that the methods were able to reduce the simulation time drastically.

Part I

Efficient time integration

Chapter Two

Structural mechanics and mixed formulation

In this chapter we want to introduce the basic notations and formulations of structural mechanics. The cases considered are those of linear elastic and nonlinear hyperelastic materials. We show how one can obtain a mixed formulation by separating deformation and inner/ hydrostatic pressure. We are going to give the discretized equations which are obtained by transforming the mixed mechanical system into the abstract notation of bilinear forms. The reader may compare these efforts to the saddle point problems obtained in [BF91b] for the linear quasi-static case. As references on the general modeling of nonlinear structural mechanics and hyperelastic materials we give [Ogd97, Hol07, MH94, Wri08] and [YBBK12].

For the mechanical formulation we start with some notations, an undeformed body is described as an open subset $\mathscr{B} \subset \mathbb{R}^3$. Its deformations are described by the mapping $\phi : (\mathscr{B}, [t_0, t_1]) \to \mathbb{R}^3$. In general we denote a point $X \in \mathscr{B}$ by $X = (X_1, X_2, X_3)$ while a point in the deformed configuration is denoted by $x \in \phi(\mathscr{B}, t)$ with $x = (x_1, x_2, x_3)$. The spatial derivative of ϕ in coordinates of $\mathscr{B}$ is the deformation gradient

$$F = \left(\frac{\partial \phi^i}{\partial X_j} \right)_{i,j}.$$

Deformations can also be described by the map of *relative* deformations

$$u(X,t) = \phi(X,t) - X.$$

The velocity is defined as the time derivative of ϕ,

$$V(X,t) = \dot{\phi}(X,t),$$

for a point $x = \phi(X,t)$ we write $V(X,t) = v(x,t)$. Also we introduce the deformation tensors of Green-Lagrange E and Cauchy-Green C by

$$E = \frac{1}{2}(F^T F - \mathrm{Id}), \qquad C = F^T F. \tag{2.1}$$

In our model we let the body $\mathscr{B}$ be subject to some internal and external forces, as illustrated in Figure 2.1. For all inner points we define a load $l(x,t)$ normalized per unit mass, and respectively $L(X,t)$ for $X \in \mathscr{B}$. The boundary is divided into two subsets. For $\Gamma_0 \subset \partial\mathscr{B}$ we have a prescribed relative deformation $u(X,t)|_{\Gamma_0} = u_0(X,t)$, for the complementary boundary $\Gamma_1 = \partial\mathscr{B} \setminus \Gamma_0$ we apply a force $T(X,t)$ also normalized per unit area (and respectively $\tau(x,t)$ for $x \in \phi(B)$). Additional to the prescribed loads, the inner forces due to deformations, as can be seen for an arbitrary cut through the body, are denoted by $\mathbf{t}(x,t,n)$ (normalized per unit area of the current configuration), where n is a vector pointing along the

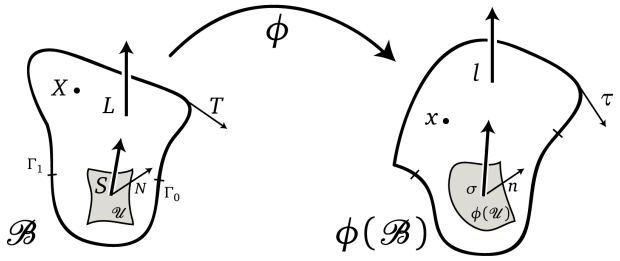


Figure 2.1: Illustration of a body and the used notation

normal direction of the cut. By Cauchy's Theorem we know that in equilibrium positions the inner force vector $\mathbf{t}(x,t,n)$ is linear in n. Thus, we introduce the Cauchy stress tensor $\sigma(x,t)$

$$\mathbf{t}(x,t,n) = \sigma(x,t)n.$$

The conservation law we want to obey is the so called *balance of momentum*. In the current configuration for $\mathscr{U} \subset \mathscr{B}$, it is stated as

$$\frac{d}{dt}\int_{\phi(\mathscr{U},t)} \rho v dx = \int_{\partial\phi(\mathscr{U},t)} \mathbf{t}(x,t,n)da + \int_{\phi(\mathscr{U},t)} \rho l dx \tag{2.2}$$

$$\Leftrightarrow \qquad \frac{d}{dt}\int_{\phi(\mathscr{U},t)} \rho v dx = \int_{\phi(\mathscr{U},t)} \operatorname{div}\sigma(x,t)dx + \int_{\phi(\mathscr{U},t)} \rho l dx. \tag{2.3}$$

For convenience we transform these relations such that they are defined on the reference configuration $\mathscr{B}$. To transform $\mathbf{t}(x,t,n)$ such that it measures force relative to the undeformed area dA instead of the deformed area da, we find the relation

$$n\,da = JF^{-T}\,dA \qquad \text{with} \qquad J = \det F$$

which in hand gives the transformation of σ

$$\sigma n\,da = J\sigma F^{-T}N\,dA,$$

and defines the first Piola Kirchhoff tensor $P = J\sigma F^{-T}$. By the second Piola Kirchhoff Tensor $S = F^{-1}P$ "the base point" is also transformed back to the undeformed configuration. So the balance of momentum with internal force vector $L(X,t)$ in terms of the reference configuration is

$$\frac{d}{dt}\int_{\mathscr{U}} \rho_{ref} V dX = \int_{\partial\mathscr{U}} PN dA + \int_{\mathscr{U}} \rho_{ref} L dX. \tag{2.4}$$

For an elastic material the Piola Kirchhoff tensor P can be written at every point only depending on the current deformation gradient F

$$P = \hat{P}(X,F).$$

Thus we are going to formulate a stored energy potential W and relate it to $\hat{P}$ via

$$\hat{P} = \frac{\partial W}{\partial F}.$$

In this way deformations generate energy into the potential W. How much energy is produced will depend on the used material law. Notice the relations between the second Piola Kirchhoff, the Green-Lagrange and Cauchy-Green tensor

$$S = \frac{\partial W}{\partial E}, \qquad S = 2\frac{\partial W}{\partial C}. \tag{2.5}$$

2.1 Linear case

In the linear case only small deformations are considered, we can first simplify the Green-Lagrange strain tensor E (2.1) by neglecting the mixed term $\nabla u \nabla u^T$ to

$$\epsilon(u) = \frac{1}{2}(\nabla u + \nabla u^T),$$

and formulate a strain energy function in terms of ϵ

$$W = \frac{\Lambda}{2}(\operatorname{tr}\epsilon)^2 + 2\mu\epsilon : \epsilon. \tag{2.6}$$

This corresponds to Hook's law using the Lamé parameters Λ and μ, where for two tensors A, B

$$A : B = \operatorname{tr}(A^T B).$$

Remark 2.1.1. *The Lamé parameters are related to Young's modulus E and Poisson's ratio ν via*

$$\Lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)}. \tag{2.7}$$

Motivated by the fact that $\operatorname{tr}\epsilon(u) = \operatorname{div} u$, we decompose the energy into a volumetric and isochoric (constant-volume) part by

$$W_{vol} = \frac{\Lambda}{2}(\operatorname{tr}\epsilon)^2, \qquad W_{iso} = 2\mu\epsilon : \epsilon, \tag{2.8}$$

$$W = W_{vol} + W_{iso}. \tag{2.9}$$

Formulation (2.8) is inappropriate for the case of incompressible materials, because for $\nu \to \frac{1}{2}$ we have $\Lambda \to \infty$. To overcome this, we introduce a mixed formulation by adding the hydrostatic pressure $p = \frac{\partial W_{vol}}{\partial(\operatorname{tr} e)} = \Lambda \operatorname{tr}\epsilon$. We obtain

$$W = \frac{p}{2}\operatorname{tr}\epsilon + 2\mu\epsilon : \epsilon. \tag{2.10}$$

Additionally, we now have to fulfill the constraint

$$\frac{p}{\Lambda} = \operatorname{div} u. \tag{2.11}$$

We see the advantage that in the incompressible case (2.11) remains feasible and reduces to

$$0 = \operatorname{div} u.$$

The pressure p is acting as a Lagrange multiplier.

The second Piola Kirchhoff stress for (2.6) using (2.5) is

$$S = \frac{\partial W}{\partial \epsilon} = \Lambda \operatorname{tr} \epsilon + 2\mu\epsilon \tag{2.12}$$

and for the mixed formulation (2.10)

$$S = p\,\mathrm{Id} + 2\mu\epsilon. \tag{2.13}$$

Using the kinetic and potential energies

$$\Pi_{kin} = \int_{\mathscr{B}} \rho_{ref} \frac{1}{2} \dot{V}^T V \, dX, \tag{2.14}$$

$$\Pi_{pot} = \frac{1}{2} \int_{\mathscr{B}} S : \epsilon(u) dX - \int_{\mathscr{B}} u^T \rho_{ref} L dX - \int_{\partial \mathscr{B}} u^T T dA, \tag{2.15}$$

and the energy conservation law

$$\int_{t_0}^{t_1} (\Pi_{kin} - \Pi_{pot}) dt \longrightarrow \text{stationary}, \tag{2.16}$$

we derive the weak formulation after introducing the appropriate function spaces.

Definition 2.1.2 (Sobolev spaces)**.** *For $\Omega \subset \mathbb{R}^n$ we introduce the Lebesgue space of square integrable functions*

$$L^2(\Omega) = \left\{ v \middle| \int_\Omega |v|^2 < \infty \right\},$$

and the Sobolev spaces

$$H^m(\Omega) = \left\{ v \middle| D^\alpha v \in L^2(\Omega), \quad \forall\, |\alpha| \le m \right\}, \qquad |\alpha| = \sum \alpha_i,$$

$$D^\alpha v = \frac{\partial^{|\alpha|} v}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}}$$

with their corresponding norms for $k \in L^2(\Omega)$, $u \in H^m(\Omega)$, we have

$$\|k\|_{L^2}^2 = \int_\Omega \|k\|^2 dx \qquad \textit{and} \qquad \|u\|_{H^m} = \sum_{k \le m} \sum_{|\alpha| = k} \|D^\alpha u\|_{L^2}^2.$$

We will skip the strong form of the equation and directly write down Problem (2.16) in its weak formulation with the boundary conditions

$$u(X,t) = u_0(X,t) \qquad \text{on } \Gamma_0 \tag{2.17}$$

$$P(X,t)N = T(X,t) \qquad \text{on } \Gamma_1. \tag{2.18}$$

Let V be the space of test functions

$$V = \left\{ v \middle| v \in (H^1(\mathscr{B}))^3, \quad v|_{\Gamma_0} = 0 \right\}.$$

Then the weak form is: Find

$$u \in U = \left\{ u \middle| u \in (H^1(\mathscr{B}))^3 \text{ and } u(\cdot,t)|_{\Gamma_0} = u_0(\cdot,t) \right\}$$

such that for all $v \in V$ it holds that

$$\int_{\mathscr{B}} \rho_{ref} v^T \ddot{u}\, dX + \int_{\mathscr{B}} S : \nabla v dX = \int_{\mathscr{B}} v^T \rho_{ref} L dX + \int_{\partial \mathscr{B}} v^T T dA.$$

By inserting the material law (2.12) we have

$$\int_{\mathscr{B}} S : \nabla v = \mu \int_{\mathscr{B}} \epsilon(u) : \epsilon(v) dX + \Lambda \int_{\mathscr{B}} \operatorname{div}(u) \operatorname{div}(v) dX.$$

While the weak form in the mixed setup (2.13) is: Find $u \in U$, $p \in K = L^2(\mathscr{B})$ fulfilling

$$\begin{aligned} \int_{\mathscr{B}} \rho_{ref} v^T \ddot{u}\, dX + \mu \int_{\mathscr{B}} \epsilon(u) : \epsilon(v) dX + \lambda \int_{\mathscr{B}} p \operatorname{div}(v) dX \\ = \int_{\mathscr{B}} v^T \rho_{ref} L dX + \int_{\Gamma} v^T T dA \end{aligned} \quad (2.19)$$

$$\int_{\mathscr{B}} \operatorname{div}(u) k dX = \frac{1}{\Lambda} \int_{\mathscr{B}} p k dX,$$

for all $k \in K$ and $v \in V$.

Using the inner-product of the Sobolov space $(H^1(\mathscr{B}))^3$

$$\langle u, v \rangle = \int_{\mathscr{B}} v^T u\, dX,$$

the bilinear forms

$$\begin{aligned} a(u,v) &= \mu \int_{\mathscr{B}} \epsilon(u) : \epsilon(v)\, dX, \\ b(v,p) &= \int_{\mathscr{B}} p \operatorname{div}(v)\, dX, \\ c(p,k) &= \int_{\mathscr{B}} k^T p\, dX, \end{aligned}$$

and

$$l = \int_{\mathscr{B}} v^T L dX + \int_{\Gamma} v^T T dA,$$

we rewrite Problem (2.19) in the abstract form for all $v \in V$

$$\begin{aligned} \langle \rho_{ref} \ddot{u}, v \rangle + a(u,v) + b(v,p) &= \langle l, v \rangle \\ b(u,k) - \frac{1}{\Lambda} c(p,k) &= 0. \end{aligned} \quad (2.20)$$

For the discretization we choose the finite dimensional function spaces $V_h \subset V$, $K_h \subset K$. Here we have to be careful in the choice of ansatz-functions to get a well defined system. A detailed discussion on the used spaces will follow in Chapter 3.2. For now we only assume that the elements of the used spaces can be represented by ansatz-functions $\phi_i \in V_h$ (for relative displacements) and $\psi_i \in K_h$ (for pressure variables) such that

$$u_h(x,t) = \sum_{i=1}^{n_q} \phi_i(x) q_i(t), \qquad p_h(x,t) = \sum_{j=1}^{n_\lambda} \psi_j(x) \lambda_j(t). \tag{2.21}$$

The discretized equations are

$$\begin{aligned} M\ddot{q} + Aq + B^T\lambda &= f(t) \\ Bq - \frac{1}{\Lambda} M_\lambda \lambda &= 0 \end{aligned} \tag{2.22}$$

with the mass matrices

$$M = \left(\int_{\mathscr{B}} \rho_{ref} \phi_i^T \phi_j \, dX \right)_{i,j}, \qquad M_\lambda = \left(\int_{\mathscr{B}} \psi_i \psi_j \, dX \right)_{i,j}, \tag{2.23}$$

the stiffness matrix

$$A = \left(\mu \int_{\mathscr{B}} \epsilon(\phi_i) : \epsilon(\phi_j) \, dX \right)_{i,j},$$

and the constraint matrix

$$B^T = \left(\int_{\mathscr{B}} \psi_j^T \operatorname{div} \phi_i \, dX \right)_{i,j}. \tag{2.24}$$

2.2 Hyperelastic case

While considering large deformations, which typically occur in rubber components, due to their much lower stiffness compared to the surrounding parts, the assumption of a linear stress-strain relation is not satisfied. We replace the linear elastic material law by a so called hyperelastic one, while preserving the isotropicity. Thus, we are entering the nonlinear setting. Structurally we will follow the same path as in the linear case by first deriving the corresponding energy strain function W. To do this, we have to state a few considerations about the used parametrization of W.

Since we assume a hyperelastic material to be isotropic, W shall not depend on the orientation of the deformation gradient F, and also it has to be independent of the material orientation so that if we decompose $F = Q_1 R Q_2$ with Q_1, Q_2 orthonormal it holds that

$$W(F) = W(R).$$

Thus we are going to use the invariants of the right Cauchy-Green tensor C (2.1) for the parametrization of W, since they are also invariant to a change of orientation.

Definition 2.2.1. *The invariants of a matrix $A \in \mathbb{R}^{3\times 3}$ are the coefficients of the characteristic polynomial*

$$\chi_A(\lambda) = \lambda^3 + I_1(A)\lambda^2 + I_2(A)\lambda + I_3(A).$$

Lemma 2.2.2. *For a symmetric matrix $A \in \mathbb{R}^{3\times 3}$ the invariants are*

$$I_1(A) = \mathrm{tr}A, \qquad I_2(A) = \det A \mathrm{tr}A^{-1}, \qquad I_3(A) = \det A,$$

if $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of A then

$$I_1(A) = \lambda_1 + \lambda_2 + \lambda_3, \qquad I_2(A) = \lambda_1^{-1} + \lambda_2^{-1} + \lambda_3^{-1} \qquad I_3(A) = \lambda_1\lambda_2\lambda_3.$$

Like in the linear case we go for a split of volumetric and isochoric deformations. As a change in volume is characterized by $J = \det F \neq 1$, we do a multiplicative decomposition of

$$F = J^{\frac{1}{3}}\bar{F}$$

into volume change J and isochoric part $\bar{F}$. This also defines the isochoric part of the Cauchy-Green tensor to be

$$\bar{C} = \bar{F}^T\bar{F} = J^{-\frac{2}{3}}C.$$

The separation is used for modeling the difference in the behavior of bulk and shear deformations

$$W = W_{vol}(J) + W_{iso}(\bar{C}). \tag{2.25}$$

W_{vol} tends also to be helpful in giving a continuous way of switching between incompressible and compressible formulations by increasing bulk modulus, as we will see.

Remark 2.2.3. *The material invariants of $\bar{C}$ can be calculated from those of C by*

$$I_1(\bar{C}) = I_1(C)J^{-\frac{2}{3}}, \qquad I_2(\bar{C}) = I_2(C)J^{-\frac{4}{3}}.$$

Definition 2.2.4. *For $c_{ij} \in \mathbb{R}$ we define the isochoric part of the strain energy function of polynomial type as*

$$W_{iso} = \text{𝒲}_{poly} = \sum_{i,j\geq 0} c_{ij}(I_1(\bar{C}) - 3)^i(I_2(\bar{C}) - 3)^j. \tag{2.26}$$

Example 2.2.5. *Two popular examples of polynomial type strain energy functions are the* Neo-Hook *material law, which consists only of $c_{10} \neq 0$ and thus has*

$$W_{iso} = \text{𝒲}_{neo} = c_{10}(I_1(\bar{C}) - 3), \tag{2.27}$$

and the Mooney-Rivlin *type material laws where additionally $c_{01} \neq 0$,*

$$W_{iso} = \text{𝒲}_{mooney} = c_{10}(I_1(\bar{C}) - 3) + c_{01}(I_2(\bar{C}) - 3). \tag{2.28}$$

Remark 2.2.6. *Another choice, not directly using the invariants, was proposed by Ogden [Ogd72]. He uses the principle stretches, i.e., the eigenvalues of* $\bar{C} = \{\lambda_1, \lambda_2, \lambda_3\}$*, and parameters* $\alpha_i, \mu_i \in \mathbb{R}$*,* $i = 1 \dots N$*, for an energy-strain function*

$$W_{iso} = \mathscr{W}_{ogden} = \sum_{i=1}^{N} 2\frac{\mu_i}{\alpha_i^2}(\lambda_1^{\alpha_i} + \lambda_2^{\alpha_i} + \lambda_3^{\alpha_i} - 3).$$

Physically this form is easier to interpret, since a principal stretch is the change of length in a corresponding principal direction, and $\lambda_k^{\alpha_i}$ *are arbitrary powers of these length changes.*

For the volumetric dependencies $W_{vol}(J)$ we choose a potential around the volume change J

$$W_{vol}(J) = \frac{1}{2}\kappa(J-1)^2 \qquad \text{with} \qquad \kappa = \Lambda + \frac{2\mu}{3}. \tag{2.29}$$

The material parameter κ is called *bulk modulus* and can be expressed in terms of the Lamé parameters Λ, μ known from the linear theory (Remark 2.1.1).

Remark 2.2.7. *Other choices of a potentials around a volume change are also possible, indeed in the literature one finds different suggestions for the volumetric dependency, e.g., [SM92]*

$$W_{vol}(J) = \kappa\frac{1}{4}(J^2 - 1 - 2\ln J),$$

or [Ogd72] suggests for $\beta > 0$

$$W_{vol}(J) = \kappa\beta^{-2}(\beta \ln J + J^{-\beta} - 1).$$

To gain the ability of handling volumetric deformations separately, we introduce again a mixed formulation using an extra pressure variable p via (2.29)

$$p = \frac{\partial W_{vol}}{\partial J} = \kappa(J-1), \tag{2.30}$$

and find it in the volumetric strain energy

$$W_{vol} = \frac{\kappa}{2}(J-1)^2 = \kappa(J-1)^2 - \frac{\kappa}{2}(J-1)^2 = p(J-1) - \frac{p^2}{2\kappa}. \tag{2.31}$$

Remark 2.2.8. *Since the linearization of the determinant around the identity is equal to the trace*

$$\det(\mathrm{Id} + hX) = 1 + h\,\mathrm{tr}\,X + O(h^2),$$

we have that the linearization of (2.30)

$$\begin{aligned}\frac{\partial}{\partial X}\kappa(J(X)-1) = \frac{\partial}{\partial X}\kappa(\det(F(X))-1)) &= \frac{\partial}{\partial X}\kappa(\det(\mathrm{Id}+\nabla u)-1)\\ &= \kappa\,\mathrm{tr}\,\nabla u = \kappa\,\mathrm{div}\,u\end{aligned}$$

corresponds to the linear case.

The second Piola Kirchhoff tensor is also split using (2.25) into an isochor and volumetric part

$$S = \frac{\partial W}{\partial E} = S_{iso} + S_{vol}$$

with

$$S_{iso} = \frac{\partial W_{iso}}{\partial E} \qquad \text{and} \qquad S_{vol} = \frac{\partial W_{vol}}{\partial E},$$

where we have the p variable in

$$S_{vol} = \kappa J(J-1)C^{-1} = JpC^{-1}. \tag{2.32}$$

In order to give the weak formulation, we reuse the kinetic energy of the linear case (2.14) and add the potential energy

$$\Pi_{pot} = \frac{1}{2}\int_{\mathscr{B}} W dX + \int_{\mathscr{B}} u^T L dX - \int_{\partial\mathscr{B}} u^T T dA,$$

into (2.16). Again we spare out the details of this calculation. The weak form of (2.16) together with the boundary conditions (2.17), (2.18) is: Find $u \in U$ such that

$$\int_{\mathscr{B}} \rho_{ref} v^T \ddot{u} dX + \int_{\mathscr{B}} S : \frac{1}{2}(\nabla v^T F(u) + F(u)^T \nabla v)\, dX \\ - \int_{\partial\mathscr{B}} v^T T\, dA - \int_{\mathscr{B}} \rho_{ref} v^T L\, dX = 0 \tag{2.33}$$

holds for all $v \in V$, where $F(u) = \mathrm{Id} + \frac{\partial u}{\partial X}$ is the relative deformation gradient. Inserting the mixed form, the weak formulation is

$$\int_{\mathscr{B}} \rho_{ref} v^T \ddot{u} dX + \int_{\mathscr{B}} (S_{iso} + S_{vol}) : \frac{1}{2}(\nabla v^T F(u) + F(u)^T \nabla v)\, dX \\ - \int_{\partial\mathscr{B}} T \cdot v\, dA - \int_{\mathscr{B}} \rho_{ref} L \cdot v\, dX = 0 \tag{2.34}$$

$$\int_{\mathscr{B}} (J(u) - 1) k\, dX = \frac{1}{\kappa}\int_{\mathscr{B}} pk\, dX, \tag{2.35}$$

to be satisfied additionally for all $k \in L^2(\mathscr{B})$.

We calculate the differential of $\int_{\mathscr{B}} J(u) - 1\, dX$ around u at u_0 applied to δu

$$\left.\frac{d\int_{\mathscr{B}} J(u) - 1\, dX}{du}\right|_{u_0} (\delta u) = \\ \int_{\mathscr{B}} JC^{-1} : \frac{1}{2}(\nabla(\delta u)^T F(u_0) + F(u_0)^T \nabla(\delta u))\, dX, \tag{2.36}$$

and find the same relation in the S_{vol} part of (2.34) while inserting (2.32)

$$\int_{\mathscr{B}} JpC^{-1} : \frac{1}{2}(\nabla v^T F(u) + F(u)^T \nabla v)\, dX. \tag{2.37}$$

For a semi-discretization we choose (2.34) using again the ansatz (2.21) in each of the three space dimensions ordered from top to bottom as $\begin{pmatrix} q^1 \\ q^2 \\ q^3 \end{pmatrix}$

$$M\ddot{q} = f_a(q,\lambda) + f_b(t) \tag{2.38}$$

$$\frac{1}{\kappa} M_\lambda \lambda = j(q) \tag{2.39}$$

with M, M_λ from (2.23). For the other terms we have

$$(f_a(q,\lambda))_{j=k+(i-1)n_q=1\ldots 3n_q}$$

$$= \int_{\mathscr{B}} (S_{iso}(q) + S_{vol}(q,\lambda)) : \frac{1}{2}\left(\frac{\partial \phi}{\partial X}^T F(q) + F(q)^T \frac{\partial \phi}{\partial X}\right) dX \tag{2.40}$$

$$= \sum_{k,l=1}^{3} \int_{\mathscr{B}} \frac{1}{2}((S_{iso}(q))_{k,l} + (S_{vol}(q,\lambda))_{k,l}) \left(\frac{\partial \phi_k}{\partial X_k} F_{il}(q) + F_{ik}(q) \frac{\partial \phi_k}{\partial X_l}\right) dX,$$

$$(f_b(t))_{j=k+(i-1)n_q=1\ldots 3n_q} = \int_{\Gamma_1} T_i \phi_k dA + \int_{\mathscr{B}} \rho_{ref} L_i \phi_k dX$$

with space dimension $i = 1\ldots 3$ and $k = 1\ldots n_q$, i.e., $f_a(q,\lambda) \in \mathbb{R}^{3n_q}$ and

$$(j(q))_{i=1\ldots n_p} = \int_{\mathscr{B}} (J(q) - 1)\psi_i dX. \tag{2.41}$$

The linearization of (2.41), i.e., the gradient of the constraint is the projection of (2.36) onto λ. Differentiating the j-th row of (2.41) gives for $i = 1\ldots 3n_q$, $m = 1\ldots 3$.

$$\begin{aligned} B_j &= \int_{\mathscr{B}} \psi_j JC^{-1} : \frac{1}{2}(\nabla(\delta u)^T F(u_0) + F(u_0)^T \nabla(\delta u))\, dX \\ &= \sum_{k,l=1}^{3} \int_{\mathscr{B}} \frac{1}{2} \psi_j JC^{-1}_{kl} \left(\frac{\partial \phi_i}{\partial X_k} F_{ml} + F_{mk} \frac{\partial \phi_i}{\partial X_l}\right) dX, \end{aligned} \tag{2.42}$$

which is again similar to the S_{vol} term in (2.40). And we have the local structure

$$\begin{aligned} M\ddot{q} &= f(t,q) - B^T(q)\lambda \\ \frac{1}{\kappa} M_\lambda \lambda &= j(q). \end{aligned} \tag{2.43}$$

For linearization (see also [YBBK12]) let us introduce the elasticity tensor $\mathbb{C} = \frac{\partial S}{\partial E}$ and its splitting into $\mathbb{C} = \mathbb{C}_{iso} + \mathbb{C}_{vol}$,

$$\mathbb{C}_{iso} = \frac{\partial S_{iso}}{\partial E}, \qquad \mathbb{C}_{vol} = \frac{\partial S_{vol}}{\partial E} = \mathbb{C}^{\flat}_{vol} + \mathbb{C}^{\natural}_{vol}, \tag{2.44}$$

and

$$\mathbb{C}^{\flat}_{vol} = p\frac{\partial (JC^{-1})}{\partial E}, \qquad \mathbb{C}^{\natural}_{vol} = JC^{-1}\frac{\partial p}{\partial E}. \tag{2.45}$$

The linearization of the material part $\int S : \frac{1}{2}(\nabla v^T F(u) + F(u)\nabla v)$ of (2.34) around u at the point u_0 evaluated at δu is

$$\frac{d\int S : \frac{1}{2}(\nabla v^T F(u) + F(u)\nabla v)}{du}|_{\delta u_0}(\delta u) =$$

$$\begin{aligned}
&\int_{\mathscr{B}} \frac{1}{2}\left(\nabla v^T F(u_0) + F(u_0)^T \nabla v\right) : (\mathbb{C}_{iso} + \mathbb{C}_{vol}) \\
&\qquad : \frac{1}{2}\left(\nabla(\delta u)^T F(u_0) + F(u_0)^T \nabla(\delta u)\right) dX \\
&\qquad + \int_{\mathscr{B}} (S_{vol} + S_{iso}) : (\nabla(\delta u)^T \nabla v)\, dX
\end{aligned}$$

$$\begin{aligned}
= &\int_{\mathscr{B}} \frac{1}{2}\left(\nabla v^T F(u_0) + F(u_0)^T \nabla v\right) : (\mathbb{C}_{iso} + \mathbb{C}^{\flat}_{vol}) \\
&\qquad : \frac{1}{2}\left(\nabla(\delta u)^T F(u_0) + F(u_0)^T \nabla(\delta u)\right) dX \\
&+ \int_{\mathscr{B}} \frac{1}{2}\left(\nabla v^T F(u_0) + F(u_0)^T \nabla v\right) : (JC^{-1} dp)\, dX \\
&\qquad + \int_{\mathscr{B}} (S_{vol} + S_{iso}) : (\nabla(\delta u)^T \nabla v)\, dX
\end{aligned} \tag{2.46}$$

with (2.44), (2.45), and

$$dp = \frac{1}{2}\kappa JC^{-1} : \left(\nabla(\delta u)^T F(u_0) + F(u_0)^T \nabla(\delta u)\right).$$

The bilinear forms from the linearized weak formulation (2.46) at the point (u_0, p_0) for $u = u_0 + \delta u$ and $p = p_0 + \delta p$ are

$$\begin{aligned}
a(\delta u, v) = &\int_{\mathscr{B}} \frac{1}{2}\left(\nabla v^T F(u_0) + F(u_0)^T \nabla v\right) : (\mathbb{C}_{iso}|_{u_0} + \mathbb{C}^{\flat}_{vol}|_{u_0,p_0}) \\
&\qquad : \frac{1}{2}\left(\nabla(\delta u)^T F(u_0) + F(u_0)^T \nabla(\delta u)\right) dX
\end{aligned}$$

and

$$b(v,\delta p)=\int_{\mathscr{B}}\frac{1}{2}\left(\nabla v^T F(u_0)+F(u_0)\nabla v\right):JC^{-1}\delta p\,dX.$$

So the local abstract formulation around some working point u_0, p_0 ends up to be the same as for the linear case

$$\langle\rho_{ref}\,\delta\ddot{u},v\rangle+a(\delta u,v)+b(v,\delta p)=\langle l,v\rangle \tag{2.47}$$

$$b(\delta u,k)-\frac{1}{\kappa}c(\delta p,k)=0. \tag{2.48}$$

Ending up with a mixed formulation of the system of structural dynamics including the nonlinear effects of large deformations and hyperelastic materials. In the next chapters we discuss the general properties of systems in the form of (2.38) and show how they can be solved. This leads us to a discussion of singularly perturbed systems.

Chapter Three

Singularly perturbed systems

To study the effects of introducing the mixed formulation onto the structure and solution of the structural dynamical problem, we are going to give the notation of differential index and singularly perturbed systems. This will allow us to interpret the mixed formulation of nearly incompressible material in terms of singularly perturbed systems.

3.1 Introduction and some properties

We start with a brief preface on differential algebraic equations.

Definition 3.1.1. *The* differentiation index *of a system*

$$\begin{aligned} \dot{x} &= f(x,z) \\ 0 &= g(x,z) \end{aligned}$$

is the smallest number of time derivatives k of the constraint $\frac{d}{dt}g(x,z)$ *such that we obtain an explicit differential equation in* $\dot{z}$*. In this way we obtain the* underlying ordinary differential equation*.*

Example 3.1.2. *The system*

$$\begin{aligned} \dot{x} &= f(x,z) \\ 0 &= g(x,z) \end{aligned} \tag{3.1}$$

is of index 1 *iff* $g_z(x,z)$ *is non singular. This can be seen by differentiating* $g(x,z)$ *once with respect to the time t*

$$\frac{d}{dt} \Rightarrow \qquad 0 = g_x(x,z)\dot{x} + g_z(x,z)\dot{z}.$$

For a non singular $g_z(x,z)$ *we have*

$$\dot{z} = -g_z^{-1} g_x f(x,z),$$

and obtain the underlying ordinary differential equation

$$\begin{aligned} \dot{x} &= f(x,z) \\ \dot{z} &= -g_z^{-1} g_x f(x,z). \end{aligned}$$

A constraint defines a manifold

$$\mathcal{M} = \{(x,z) : g(x,z) = 0\},$$

which contains all solutions of the problem.

Definition 3.1.3. *The* tangent-space *of a point $x \in \mathcal{M} = \{x : g(x) = 0\}$ is defined by*

$$T_x\mathcal{M} = \left\{(x, v) : g_x(x)v = 0\right\}.$$

Example 3.1.4. *The general constrained mechanical system*

$$\begin{aligned} \dot{q} &= v \\ M\dot{v} &= f(q, v) - G^T(q)\lambda \\ 0 &= g(q) \end{aligned} \tag{3.2}$$

with $G = g_q$ is of index 3 *if $GM^{-1}G^T$ is non singular.*

Proof. Differentiating (3.2) twice gives

$$0 = g_q\dot{q} = g_q v, \tag{3.3}$$

$$0 = g_{qq}(v, v) + g_q\dot{v} = g_{qq}(v, v) + g_q M^{-1} f(q, v) - g_q M^{-1} G^T \lambda. \tag{3.4}$$

Arising from the last term of (3.4) we obtain an explicit dependence on $\dot{\lambda}$ by differentiating once more. □

Remark 3.1.5. *Due to the structure and definition of $\mathcal{M}$, the solutions to the mechanical system are still contained in the manifold. They are further constrained onto a hidden sub-manifold because of the additional constraints arising during the differentiation of the equation, as seen in* (3.3).

Definition 3.1.6 (Singularly perturbed system)**.** *The system*

$$\begin{aligned} \dot{x} &= f(x, z) \\ \epsilon\dot{z} &= g(x, z) \end{aligned}$$

is called singularly perturbed *if g_z is non singular.*

Observe the influence of ϵ to the differential index of the systems. For $\epsilon \to 0$ the system is of index 1, while for all cases $\epsilon > 0$ we have index 0. Singularly perturbed systems are also discussed in [TVS85, REOM88, Joh05]. Our formulation is especially interesting in the convergence analysis of numerical integration methods for differential algebraic equations [HW96].

A singular perturbation in the case of mechanical systems brings us to a higher index and thus to different problems. We are going to discuss the influence of the structural change by introducing a stiff spring via ϵ^2 to a mechanical system. As a prototype of a mechanical system, we take a look at second order differential equations of the form

$$\begin{aligned} M\ddot{q} &= f(q, \dot{q}) - G(q)^T\lambda \\ \epsilon^2 M_\lambda \lambda &= g(q) \end{aligned} \tag{3.5}$$

with $G(q) = g_q$ of full row rank, $y \in \mathbb{R}^n$, $\lambda \in \mathbb{R}^m$, $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^m$, and $0 < \epsilon \ll 1$. Looking at the index of system (3.5), we see that for $\epsilon > 0$ one differentiation is needed to obtain

$$\dot{\lambda} = \frac{1}{\epsilon^2} M_\lambda^{-1} G(q)\dot{q},$$

but for $\epsilon^2 \to 0$ the system has the form of Example 3.1.4 and is of index 3.

Definition 3.1.7 (Singular singularly perturbed system)**.** *The second order system with positive symmetric matrix M*

$$\begin{aligned} M\ddot{q} &= f(q,\dot{q}) - G^T\lambda \\ \epsilon^2\lambda &= g(q) \end{aligned} \tag{3.6}$$

is called singular singularly perturbed *if* $g_q M^{-1} g_q^T$ *is non singular.*

While solving mechanical systems, singular singularly perturbed systems are likely to arise, for example when the constraints of a mechanical system are replaced by stiff springs.

Example 3.1.8 (Stiff spring-pendulum)**.** *The dynamics of a pendulum with unit mass, unit length and gravity g are described by*

$$\begin{aligned} \ddot{q}_1 &= -2q_1\lambda \\ \ddot{q}_2 &= -2q_2\lambda - g \\ 0 &= q_1^2 + q_2^2 - 1. \end{aligned} \tag{3.7}$$

The constraint is responsible for holding the pendulums length at exactly 1. Replacing this constraint by a stiff spring such that a change in the length of the pendulum $\sqrt{q_1^2+q_2^2}$ *away from* 1 *is penalized by a strong force* $\frac{1}{\epsilon^2}$*, and bounding this force by considering only the relative change of length we have the constraint*

$$\epsilon^2\lambda = \frac{\sqrt{q_1^2+q_2^2}-1}{\sqrt{q_1^2+q_2^2}}.$$

This gives us a singular singularly perturbed index 1 system.

Here λ *can also be expressed in terms of known quantities and thus inserted directly into* (3.7) *by*

$$\begin{aligned} \ddot{q}_1 &= -\frac{1}{\epsilon^2} 2q_1 \frac{\sqrt{q_1^2+q_2^2}-1}{\sqrt{q_1^2+q_2^2}} \\ \ddot{q}_2 &= -\frac{1}{\epsilon^2} 2q_2 \frac{\sqrt{q_1^2+q_2^2}-1}{\sqrt{q_1^2+q_2^2}} - g. \end{aligned} \tag{3.8}$$

By (3.8) *we have formulated the system without a constraint, in index 0 form.*

Remark 3.1.9. *Looking at a singular singularly perturbed system as in Definition 3.1.7, for* $\epsilon^2 > 0$ *one can obtain an index 0 formulation by solving the second relation for* λ *and inserting into the first one*

$$M\ddot{q} = f(q,\dot{q}) - \frac{1}{\epsilon^2} G^T g(q). \tag{3.9}$$

Theorem 3.1.10 (Smooth Motion)**.** *For system* (3.5) *we have that for every* $(q_0, \dot{q}_0)$ *satisfying*

$$g(q_0) = 0, \qquad\qquad G(q_0)\dot{q}_0 = 0, \tag{3.10}$$

there exists a pair $(q_\epsilon, \dot{q}_\epsilon)$, unique up to $O(\epsilon^{2N})$ for arbitrary N, with $q_0 - q_\epsilon$, $\dot{q}_0 - \dot{q}_\epsilon$ of magnitude $O(\epsilon^2)$ and situated in the M-orthogonal complement of

$$\mathcal{M} = \{q : g(q) = 0\}$$

such that the solution with initial values $(q_\epsilon, \dot{q}_\epsilon)$ is smooth and of the form

$$\begin{aligned} q(t) &= q_0(t) + \epsilon^2 q_1(t) + \cdots + \epsilon^{2N} q_N(t) + O(\epsilon^{2N+2}), \\ \dot{q}(t) &= \dot{q}_0(t) + \epsilon^2 \dot{q}_1(t) + \cdots + \epsilon^{2N} \dot{q}_N(t) + O(\epsilon^{2N+2}). \end{aligned} \tag{3.11}$$

In this expression, the functions $q_k(t), \dot{q}_k(t)$, and the domain $[0, T]$ are independent of ϵ for $k = 0 \ldots N$.

Proof. Following the lines of the proof given by [Lub93] in the index 0 case (see also Remark 3.1.11), we do an analysis of the index 1 formulation.

First we construct a truncated expansion by comparison of the ϵ coefficients of q_k in the solution of (3.5) where we also write λ in an expanded form

$$\lambda = \lambda_0 + \epsilon^2 \lambda_1 + \cdots + \epsilon^{2N} \lambda_N.$$

The ϵ^{-2} term only appears in the constraint and vanishes iff

$$g(q_0) = 0. \tag{3.12}$$

For ϵ^0 we find

$$M\ddot{q}_0 = f(q_0, \dot{q}_0) - G^T(q_0)\lambda_0 \tag{3.13}$$

$$M_\lambda \lambda_0 = G(q_0) q_1. \tag{3.14}$$

Here q_0, λ_0 can be determined by solving the index 3 system (3.13) together with the position constraint (3.12). Because of the full rank of G, the system has got a unique solution for all initial values $(q_0(0), \dot{q}_0(0))$ in the tangent bundle $T\mathcal{M}$. Going on to the ϵ^2 coefficient, we have

$$M\ddot{q}_1 = f_q(q_0, \dot{q}_0) q_1 + f_{\dot{q}}(q_0, \dot{q}_0)\dot{q}_1 - G^T(q_0)\lambda_1 \tag{3.15}$$

$$M_\lambda \lambda_1 = G(q_0) q_2 + \frac{1}{2} H(q_0)(q_1, q_1) \tag{3.16}$$

so that we consider for known values of $q_0, \dot{q}_0$ equation (3.15) together with (3.14) again as an index 3 system with unique solution for $q_1, \dot{q}_1$. The initial values $q_1(0), \dot{q}_1(0)$ are determined uniquely by the condition that both $q_1, \dot{q}_1$ are in the range of $M^{-1}G^T(q_0)$. We can proceed in this way and construct more elements in the sequence of $q_k, \dot{q}_k$ by introducing more index 3 systems, up to an arbitrary k.

It remains to show that every solution with starting values close to the constructed epsilon expansion of q remains in an $O(\epsilon^{2N})$ neighborhood. So let $\xi, \dot{\xi}$ be

$$\xi = q_0 + \epsilon^2 q_1 + \cdots + \epsilon^{2N} q_N, \tag{3.17}$$

$$\dot{\xi} = q_0 + \epsilon^2 \dot{q}_1 + \cdots + \epsilon^{2N} \dot{q}_N. \tag{3.18}$$

The defect while inserting (3.17) into (3.5) due to construction is

$$\begin{aligned} M\ddot{\xi} &= f(\xi, \dot{\xi}) - G^T \lambda + O(\epsilon^{2N+2}) \\ \lambda &= \epsilon^{-2} g(\xi) + O(\epsilon^{2N}). \end{aligned} \tag{3.19}$$

We will show that every solution $q, \dot{q}$ of the system (3.5) with starting values

$$q(0) - \xi(0) = O(\epsilon^{2N+1}), \qquad \dot{q}(0) - \dot{\xi}(0) = O(\epsilon^{2N})$$

remains near $\xi, \dot{\xi}$, i.e.,

$$q(t) - \xi(t) = O(\epsilon^{2N}), \qquad \dot{q}(t) - \dot{\xi}(t) = O(\epsilon^{2N}) \tag{3.20}$$

uniformly for t on bounded intervals.

For the following we assume that the system is written in coordinates such that $g(q) = \begin{bmatrix} 0 & \mathrm{Id} \end{bmatrix} q$ and with $M = \mathrm{Id}$. We consider the difference of $\delta q = q - \xi$ and add the constraint via $\delta\lambda$ using (3.19)

$$\begin{aligned} \delta\ddot{q} &= O(\delta q) + O(\delta\dot{q}) - \begin{bmatrix} 0 \\ Id \end{bmatrix} \delta\lambda + O(\epsilon^{2N+2}) \\ \delta\lambda &:= \epsilon^{-2} \begin{bmatrix} 0 & \mathrm{Id} \end{bmatrix} \delta q + O(\epsilon^{2N}) \end{aligned} \tag{3.21}$$

which is, due to the linearization of f, valid if δq is at least $O(\epsilon^2)$. Next we insert $\delta\lambda$ explicitly into (3.21)

$$\delta\ddot{q} = O(\delta q) + O(\delta\dot{q}) - \epsilon^{-2} \begin{bmatrix} 0 & 0 \\ 0 & \mathrm{Id} \end{bmatrix} \delta q + \begin{bmatrix} 0 & 0 \\ 0 & \mathrm{Id} \end{bmatrix} O(\epsilon^{2N}) + O(\epsilon^{2N+2}). \tag{3.22}$$

If we write down the differential equation (3.22) separated into $\delta q = \begin{pmatrix} \delta u & \delta v \end{pmatrix}^T$ for δv being those components which contain ϵ^{-2}, then we have that

$$\begin{aligned} \delta\ddot{u} &= O(\|\delta u\| + \|\delta\dot{u}\| + \|\delta v\| + \|\delta\dot{v}\|) + O(\epsilon^{2N+2}), \\ \delta\ddot{v} &= -\epsilon^{-2}\,\mathrm{Id}\,\delta v + O(\|\delta u\| + \|\delta\dot{u}\| + \|\delta v\| + \|\delta\dot{v}\|) + O(\epsilon^{2N}). \end{aligned} \tag{3.23}$$

Rewriting δv to first order form by $\delta w = \begin{pmatrix} \delta v & \epsilon^{-1}\delta\dot{v} \end{pmatrix}^T$ yields

$$\delta\dot{w} = \epsilon^{-1} \begin{bmatrix} 0 & \mathrm{Id} \\ -\mathrm{Id} & 0 \end{bmatrix} \delta w + O(\|\delta w\| + \epsilon\|\delta u\| + \epsilon\|\delta\dot{u}\|) + O(\epsilon^{2N+1}).$$

Calculating the energy estimate

$$\begin{aligned} \|\delta w\| \cdot \frac{d}{dt}\|\delta w\| = \frac{1}{2}\frac{d}{dt}\|\delta w\|^2 &= \delta w^T \delta\dot{w} \\ &= O(\|\delta w\| \cdot (\|\delta w\| + \epsilon\|\delta u\| + \epsilon\|\delta\dot{u}\| + O(\epsilon^{2N+1}))) \end{aligned}$$

and using Gronwall's Lemma with $\delta w(0) = O(\epsilon^{2N+1})$ gives

$$\|\delta w(t)\| \le C\epsilon \max_{0\le\tau\le t} (\|\delta u(\tau)\| + \|\delta\dot{u}(\tau)\|) + O(\epsilon^{2N+1}).$$

Finally reinserting into (3.23) gives the result $\delta u = O(\epsilon^{2N})$, $\delta\dot{u} = O(\epsilon^{2N})$.

To show the smoothness of the constructed solution q of (3.5) fulfilling (3.20), we investigate $\delta\ddot{q} = \ddot{q} - \ddot{\xi}$ in (3.22). By inserting (3.20) into (3.22), we see that

$$\delta\ddot{q} = O(\epsilon^{2N-2}). \tag{3.24}$$

Differentiating (3.24) once more and using again (3.20), we have that also

$$\delta q^{(3)} = O(\epsilon^{2N-2}).$$

By further differentiation of (3.22) we see that

$$\delta q^{(4)} = O(\epsilon^{2N-4}), \qquad \delta q^{(5)} = O(\epsilon^{2N-4}),$$
$$\delta q^{(6)} = O(\epsilon^{2N-6}), \qquad \delta q^{(7)} = O(\epsilon^{2N-4}).$$

And thus the constructed solution is smooth.

□

Remark 3.1.11. *A similar result to Theorem 3.1.10 can be obtained for the more general index 0 system*

$$M(q)\ddot{q} = f(q,\dot{q}) - \frac{1}{\epsilon^2}\nabla U(q),$$

with a potential $U : \mathbb{R}^n \to \mathbb{R}$, $(\nabla U)^T = \frac{\partial U}{\partial q}$, $f : \mathbb{R}^{2n} \to \mathbb{R}^n$, *and* $M : \mathbb{R}^n \to \mathbb{R}^{n\times n}$. *However, we have some additional assumptions on the potential U and on the configuration-dependent mass matrix :*

(a) $M(q)$ *is symmetric and positive definite for all* $q \in \mathbb{R}^n$.

(b) The potential U attains a local minimum on a d-dimensional manifold $\mathcal{U}$*, i.e., for some region* $D \subset \mathbb{R}^n$

$$\mathcal{U} = \left\{ u \in D : U(u) = \min_{q\in D} U(q) \right\} = \{u \in D : \nabla U(u) = 0\}.$$

(c) U is in a neighborhood of $\mathcal{U}$*, strongly convex along directions non-tangential to* $\mathcal{U}$*, i.e., there exists* $\alpha > 0$ *such that for* $u \in \mathcal{U}$ *it holds that*

$$v^T \nabla^2 U(u) v \geq \alpha \dot{v}^T M(u) v$$

for all v in the $M(u)$*-orthogonal complement to the tangent space* $T_u\mathcal{U}$.

The proof relies on a local transformation into the index 1 case, which leads to the same arguments as in the previous proof. It can be seen in [Lub93].

Theorem 3.1.10 shows how index 1 and index 3 formulations are connected. In the index 3, i.e., $\epsilon = 0$, case all solutions to system (3.5) have to be inside the tangent bundle $T\mathcal{M}$, the system is thus already of the form of (3.12), (3.13). For $\epsilon > 0$ we see how the solutions starting from the tangent bundle are influenced. The crucial aspect are the initial values, for initial values outside the smooth motion additional oscillations occur. The theorem tells us how solutions starting in a $O(\epsilon^2)$ radius around the tangent bundle behave with respect to ϵ, they stay close to the corresponding solution of the index 3 system. This justifies the exchange of constraints by strong penalizing forces, in favor of a lower indexed system, since the dependence of the solution on the constraint is small for small values of ϵ.

Example 3.1.12 (Second order Prothero-Robinson)**.** *An example of a singular singularly perturbed system in the form of* (3.5) *is the extension of the first order linear Prothero-Robinson equation [PR74] to a, still linear, second order form. The initial value problem for a two times continuous differentiable function* ϕ *is*

$$\ddot{q} = \ddot{\phi} - \lambda \qquad q(t_0) = \phi(t_0) + \gamma$$
$$\epsilon^2\lambda = q - \phi \qquad \dot{q}(t_0) = \phi(t_0) + \eta$$

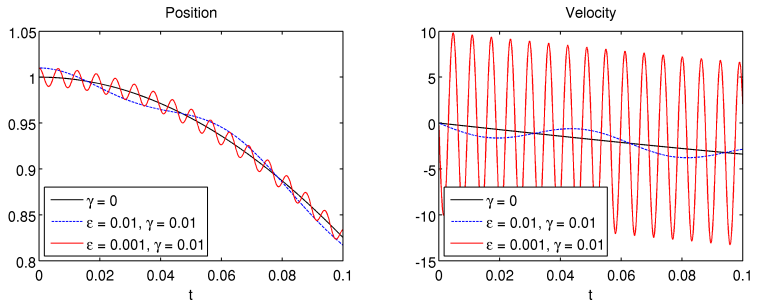


Figure 3.1: Solution of Prothero-Robinson equation for different values of ϵ and perturbed initial value, $\gamma = .01$ plotted in position q and velocity $\dot{q}$ for $\phi(t) = \cos(6t)$

The idea of this equation is that we try to enforce the motion of ϕ onto q via a stiff spring $\epsilon^2\lambda$. The analytic solution to this problem is

$$q(t) = \phi(t) + \gamma \cos\left(\frac{1}{\epsilon}(t - t_0)\right) + \eta\epsilon \sin\left(\frac{1}{\epsilon}(t - t_0)\right). \tag{3.25}$$

For the equation all higher order terms in the ϵ-expansion (3.11) *of the smooth solutions, i.e., $\eta = \gamma = 0$ are zero. This is due to the linearity of the constraint.*

Nevertheless, we see that solutions starting with $\eta, \gamma \in O(\epsilon^2)$ oscillate with a the high frequency of $\frac{1}{2\pi\epsilon}$ but only an amplitude of $O(\epsilon^2)$ around the smooth solution $\phi(t)$. The behavior is illustrated in Figure 3.1. We see how a perturbation in the initial value provokes oscillations around the smooth solution: by decreasing the value of ϵ, the frequency of the oscillations increases. In the velocity component we see also an increase of the amplitude away from $O(\epsilon^2)$ while $\gamma > \epsilon^2$.

Looking at the Prothero-Robinson example one can understand that the problem arises while tackling the perturbed systems by numerical methods. Due to the error of a time integrator, a perturbation away from the smooth motion is introduced and thus oscillations are provoked. This will be a topic in the following chapter on linear implicit methods.

3.2 In the context of mixed formulations

We want to bring the mixed systems derived in Chapter 2 into the formulation of perturbed systems. So we are going to show that the hyperelastic mixed formulation is a singular singularly perturbed system.

Looking at the linear case by discretizing the bilinear forms, we obtained a system of equations of the form

$$\begin{aligned} M\ddot{q} + Aq + B^T\lambda &= f(t) \\ \frac{1}{\Lambda}M_\lambda\lambda &= Bq. \end{aligned} \tag{3.26}$$

In (2.23) we already saw that the mass matrices M, M_λ are both symmetric and positive definite. For the incompressible system to be of index 3 we need to show

that the constraint matrix $B \in \mathbb{R}^{n_\lambda \times n_q}$ is of full rank. A useful criterion for this is the min-max characterization of B which is in a similar context also used in [Sim06].

Lemma 3.2.1. *A matrix $B \in \mathbb{R}^{n_\lambda \times n_q}$ with $n_q > n_\lambda$ is of full rank iff*

$$\min_\lambda \max_\nu \frac{\lambda^T B \nu}{\|\lambda\|_2 \|\nu\|_2} = \sigma_{min} > 0.$$

Proof. Let U, V be orthogonal matrices such that

$$U^T B V = \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_{n_\lambda} & \end{bmatrix} \in \mathbb{R}^{n_\lambda \times n_q},$$

with ordered singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n_\lambda} \geq 0$. For a full rank of B the smallest singular value has to be greater than zero, $\sigma_{n_\lambda} > 0$. This implies for all $\lambda \neq 0$

$$\frac{\lambda^T B B^T \lambda}{\lambda^T \lambda} \geq \sigma_{n_\lambda}^2.$$

For e_i being the i-th unit vector and $\lambda = U e_{n_\lambda}$ this inequality is sharp, we have

$$\sigma_{n_\lambda}^2 = \min_\lambda \frac{\lambda^t B B^T \lambda}{\lambda^T \lambda} \qquad \Leftrightarrow \qquad \sigma_{n_\lambda} = \min_\lambda \frac{\|B^T \lambda\|_2}{\|\lambda\|_2}.$$

Since the operator norm is defined as

$$\|B^T \lambda\|_2 = \max_\nu \frac{\|\nu^T B^T \lambda\|_2}{\|\nu\|_2} = \max_\nu \frac{\lambda^T B \nu}{\|\nu\|_2},$$

we derive the min-max characterization

$$\min_\lambda \max_\nu \frac{\lambda^T B \nu}{\|\lambda\|_2 \|\nu\|_2} = \sigma_{n_\lambda} > 0,$$

which is again equivalent to the full rank criterion. □

Remark 3.2.2. *The criterion derived on the constraint matrix B is equivalent to the inf-sup condition on finite elements [BF91b]*

$$\inf_{k \in K} \sup_{v \in V} \frac{b(v,k)}{\|v\|_V \|k\|_K} > 0.$$

In our case $B = (B_{ij})_{i=1\ldots n_\lambda, j=1\ldots n_q}$ with (see also (2.24))

$$B_{ij} = \int_{\mathscr{B}} \psi_i(x)^T \operatorname{div} \phi_j(x) dx \tag{3.27}$$

the criterion derived in Lemma 3.2.1 boils down to the used finite elements (ϕ_j, ψ_i) being divergence free.

Definition 3.2.3. *For a domain D let $P_n(D)$ be the space of all polynomials up to degree n in each variable.*

Lemma 3.2.4. *Let V be a space of appropriate test functions, then for a partition τ of $\mathscr{B}$ into quadtrilaterals and n arbitrary the finite dimensional spaces*

$$V_h = \left\{ v_h \in V \,\middle|\, v_h|_{\tau_i} \in P_n \text{ for all } \tau_i \in \tau \right\},$$
$$K_h = \left\{ k_h \in H^1(\mathscr{B}) \,\middle|\, k_h|_{\tau_i} \in P_{n-1} \text{ for all } \tau_i \in \tau \right\}$$

fulfill the inf-sup condition for (3.27).

Proof. [BF91a] □

Remark 3.2.5. *For $n = 2$ we obtain the well known Taylor-Hood element [HT73] which provides a quadratic approximation of position/ velocity and a linear approximation of pressure variables.*

Knowing this and again that M_λ is non-singular, we see that system (3.26) is of index 1 for $\Lambda > 0$ and thus is a singular singularly perturbed system in the form of (3.5).

Hyperelastic case with large deformations

Similarly to Chapter 2 we want to discuss the structure of the system in case of large deformations and hyperelastic materials. As we already saw in (2.47), the linearized nonlinear system

$$\langle \rho_{ref} \ddot{u}, v \rangle + a(u, v) + b(v, p) = \langle l, v \rangle$$
$$b(u, k) - \frac{1}{\kappa} c(p, k) = 0.$$

matches the structure of the linear one (2.20).

For justifying a full rank of the discretized form of $b(v, p)$, Lemma 3.2.1 remains valid. Thus, we also have to fulfill the inf-sup condition

$$\inf_{p \in K} \sup_{v \in V} \frac{b(v, p)}{\|v\|_V \|p\|_K} > 0.$$

The only difference to the linear case is inside of b, because now we have for some fixed linearization point u_0 (see also (2.48)) that

$$b(v, p) = \int_{\mathscr{B}} \frac{1}{2} (\nabla v^T F(u_0) + F(u_0)^T \nabla v) : JC^{-1} p dX.$$

By inserting the Cauchy-Green tensor $C = F^T F$ (2.1) and expanding the contraction, we have

$$b(v, p) = \int_{\mathscr{B}} \frac{1}{2} \operatorname{tr} \left((\nabla v^T F(u_0) + F(u_0)^T \nabla v) J F(u_0)^{-1} F(u_0)^{-T} \right) p dX$$

using some properties of the trace operator

$$\Big(\text{e.g.,} \qquad \operatorname{tr}(A^T B) = \operatorname{tr}(B^T A) \qquad \text{and} \qquad \operatorname{tr}(A + B) = \operatorname{tr}(A) + \operatorname{tr}(B) \Big)$$

and the regularity of F at u_0

$$\begin{aligned} b(v,p) &= \int_{\mathscr{B}} \frac{1}{2} J \Big(\operatorname{tr}(\nabla v^T F(u_0) F(u_0)^{-1} F(u_0)^{-T}) \\ &\qquad + \operatorname{tr}(F(u_0)^T \nabla v F(u_0)^{-1} F(u_0)^{-T}) \Big) p \, dX \\ &= \int_{\mathscr{B}} \frac{1}{2} J \left(\operatorname{tr}(\nabla v^T F(u_0)^{-T}) + \operatorname{tr}(F(u_0)^{-1} \nabla v) \right) p \, dX \\ &= \int_{\mathscr{B}} J \operatorname{tr}(F(u_0)^{-1} \nabla v) p dX \\ &= \int_{\mathscr{B}} J \operatorname{div}(F(u_0)^{-1} v) p \, dX. \end{aligned}$$

We end up at a divergence relation. Observe further for $v \in V = H_0^1(\mathscr{B})^3$ also $F(u_0)^{-1} v \in V$ since $\nabla(F(u_0)^{-1} v) = F(u_0)^{-1} \nabla v$. From this we see that the elements chosen in the linear case are still sufficient for fulfillment of the inf-sup condition and thus the full rank of the discretized bilinear form.

Remark 3.2.6. *Our concern here was only the inf-sup condition on b to obtain a singular singularly perturbed system. Nevertheless, for the problem to have a solution the usual elipticity condition on $a(u,v)$ still has to be satisfied. For large deformations this can also lead to difficulties as pointed out in [PB97].*

We see that the considered mixed formulations of Chapter 2, linear (2.22) and hyperelastic case (2.47) (2.48), are indeed singular singularly perturbed systems. In the next chapter we want to discuss how the numerical solution of the systems is affected by this property.

Chapter Four

Linear implicit methods

In this chapter we present the numerical methods used by us for efficiently simulating the presented stiff mechanical systems. Generally, the class of integration methods can be separated into explicit and implicit ones. Explicit methods can easily handle systems with many degrees of freedom because the solution of linear systems is avoided. They do this by restricting the step-size dependent on the smallest element and the density distribution inside the body. However, additional constraints can lead to unstable systems.

On the other hand there are implicit methods which rely on the solution of nonlinear systems up to some accuracy, which is usually done by different linearizations and Newton's method. The linear implicit methods considered here are a mixture of both types. In every time-step one linear system has to be solved, while a Newton iteration is avoided. This is a benefit which helps to save computation time. Also these methods give us an advantage for constrained systems and allow for larger time-steps than the explicit methods.

Implicit Runge-Kutta methods

Let us first do a excursion into implicit Runge-Kutta methods. We will recall their definition and structure and show their behavior for singular singularly perturbed systems.

Definition 4.0.7. *For the autonomous system*

$$\dot{x} = f(x)$$

we call the recursion formula

$$k_i = f\left(x_n + h\sum_{j=1}^{s} a_{ij}k_j\right), \qquad i = 1\dots s \tag{4.1}$$

$$x_{n+1} = x_n + h\sum_{i=1}^{s} b_i k_i$$

for initial value x_0 and step-size h a s-stage Runge-Kutta method with method-dependent coefficients $A = (a_{ij})_{ij} \in \mathbb{R}^{s\times s}$ and $b = (b_i)_i \in \mathbb{R}^s$.

Depending on the matrix A, these methods are either explicit (if all $a_{ij} = 0$ for $j \geq i$) or implicit. In the implicit case in every time-step the nonlinear system (4.1) has to be solved, herein the size of the system to be solved depends on the count of "future" k_j used in the construction of k_i, which is related to those $a_{ij} \neq 0$ for $j \geq i$.

Remark 4.0.8. *We can extend Runge-Kutta methods to implicit differential equations of the form*

$$F(x, \dot{x}) = 0$$

by solving

$$0 = F(X_{n,i}, \dot{X}_{n,i}), \tag{4.2}$$
$$X_{n,i} = x_n + h \sum_{j=1}^{s} a_{ij} \dot{X}_{n,j}$$

for $\dot{X}_{n,i}$, $i = 1 \dots s$ *and setting*

$$x_{n+1} = x_n + h \sum_{j=1}^{s} a_{ij} \dot{X}_{n,i}.$$

In the same manner the method can be extended to higher order systems.

Definition 4.0.9. *An autonomous iteration scheme* $x_{n+1} = f(x_n)$ *is called* stable *if* x_n*, for* $n \to \infty$ *is bounded.*

Definition 4.0.10 (A-stability)**.** *A numerical integration method is called* A-stable *if it is stable when applied to the linear test equation* $\dot{x} = \lambda x$ *for all* $\lambda < 0$*.*

Definition 4.0.11 (Convergence and Order)**.** *An integration method is said to be of* order p *if its local error satisfies*

$$x(t_0 + h) - x_1 = O(h^{p+1}),$$

where $x(t)$ *is the exact solution and* x_1 *the result of one time-step with starting value* $x_0 = x(t_0)$ *and step-size* h*. Further the method is said to be* convergent of order p *if its global error satisfies*

$$x(t_0 + nh) - x_n = O(h^p)$$

for x_n *being the n-th successive steps of the iteration method using the initial value* $x_0 = x(t_0)$*.*

Definition 4.0.12 (Stage order)**.** *A Runge-Kutta method has* stage order $r \geq 1$ *iff for all* $l = 1 \dots r$ *the recursive formula*

$$\sum_{j=1}^{s} a_{ij} c_j^{l-1} = \frac{c_i^l}{k}$$

is fulfilled, where $c_i = \sum_{j=1}^{s} a_{ij}$*.*

Theorem 4.0.13. *An A-stable Runge-Kutta method with stage order* $r \geq 1$ *applied to index 3 systems of the form of the general constrained mechanical system (Example 3.1.4) is convergent of order* r*, i.e.,*

$$x(t_n) - x_n = O(h^r).$$

Proof. [Lub93] □

Remark 4.0.14. *Collocation methods [HW96] satisfying*

$$0 < c_1 < \cdots < c_s = 1$$

with order of convergence $p > r$ for ordinary differential equations are also convergent of order p for index 3 systems [Jay93].

Example 4.0.15 (Radau5)**.** *A prominent example of implicit Runge-Kutta methods is implemented in the* `RADAU5` *code developed in [HW96]. It utilizes a 3 stage method consisting of*

$$b = \begin{pmatrix} \frac{16-\sqrt{6}}{36} \\ \frac{16+\sqrt{6}}{36} \\ \frac{1}{9} \end{pmatrix}, \quad c = \begin{pmatrix} \frac{4-\sqrt{6}}{10} \\ \frac{4+\sqrt{6}}{10} \\ 1 \end{pmatrix}, \quad A = \begin{bmatrix} \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{bmatrix}.$$

The method is convergent of order 5 *and has stage order* 3*. Since it fulfills also that $0 < c_1 < c_2 < c_3 = 1$,* `RADAU5` *is convergent of order* 5 *in the q component, even for the index 3 case and of order 3 in $v = \dot{q}$.*

Coming back to the perturbed system for the solution of (3.1) via a Runge-Kutta method, one can show:

Theorem 4.0.16. *Given an A-stable Runge-Kutta method with stage order r and starting values $q_0^\epsilon, \dot{q}_0^\epsilon$ inside M^ϵ, for $0 < \epsilon \leq h \leq h_0$ there exists a unique Runge-Kutta solution $q_n = \begin{pmatrix} q_n^\epsilon & \dot{q}_n^\epsilon \end{pmatrix}^T$ of* (3.5) *whose error satisfies*

$$\begin{aligned} q_n^\epsilon - q^\epsilon(t_n) &= q_n^0 - q^0(t_n) + O(\epsilon^2 h^{r-2}), \\ \dot{q}_n^\epsilon - \dot{q}^\epsilon(t_n) &= \dot{q}_n^0 - \dot{q}^0(t_n) + O(\epsilon^2 h^{r-2}) \end{aligned}$$

uniformly for all $0 \leq t_n \leq T$. Here q_n^0, q^0 denote Runge-Kutta and exact solution of the corresponding index 3 system with starting values satisfying the conditions of Theorem 3.1.10.

Proof. See [Lub93]. □

Considering that the system to be solved is of index 1, the estimated convergence order for the Runge-Kutta method reduces at least down to the index 3 case. Also the advantage of a collocation methods is lost.

Example 4.0.17. *In the case of* `RADAU5` *as seen in Example 4.0.15, we obtain by Theorem 4.0.16 for a singular singularly perturbed system with $\epsilon < h$,*

$$\begin{aligned} q_n^\epsilon - q^\epsilon(t_n) &= O(h^5) + O(\epsilon^2 h), \\ \dot{q}_n^\epsilon - \dot{q}^\epsilon(t_n) &= O(h^3) + O(\epsilon^2 h). \end{aligned}$$

For large time-steps h, i.e. $h > \epsilon$, the order reduces from $O(h^5)$ in the q component to $O(h^3)$. On the other hand, for $h \ll \epsilon$ we see the classical asymptotic behavior

$$\begin{aligned} q_n^\epsilon - q^\epsilon(t_n) &= O(h^5), \\ \dot{q}_n^\epsilon - \dot{q}^\epsilon(t_n) &= O(h^3). \end{aligned}$$

This effect can also be verified numerically as seen in [Sim98].

Additionally to the order reduction, convergence issues while solving the nonlinear system (4.2) via a Newton iteration are observed. Especially, if one considers the index 0 formulation of the singular singularly perturbed system (3.9), the perturbation leads to a step-size restriction dependent on ϵ, in order to obtain a convergent Newton iteration [Lub93].

4.1 Rosenbrock methods

For the construction of the so called Rosenbrock methods [Ros63] we first start by the Runge-Kutta discretization of an autonomous system

$$\dot{x} = f(x).$$

For a given initial value x_0 the solution via a Runge-Kutta method is obtained by

$$\begin{aligned} x_1 &= x_0 + \sum_{j=1}^{s} b_j k_j, \\ k_i &= hf\left(x_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_j + \sum_{j=i}^{s} \alpha_{ij} k_j\right), \qquad i = 1 \dots s. \end{aligned}$$

The essential idea now is to use a linearization of $J = f_x$ and estimate k_i via

$$\begin{aligned} k_i &= hf(v_i) + hJ(v_i) \sum_{j=i}^{s} \alpha_{ij} k_j, \\ v_i &= x_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_j. \end{aligned}$$

This can be even more simplified by replacing $J(v_i) \approx J(x_0)$ and considering only methods with $\alpha_{ij} = 0$ for $j > i$. The full implicit formulation of these methods are so called diagonally implicit Runge-Kutta methods (DIRK). In the case of all $\alpha_{ii} = \alpha_{jj}$ (for all i, j) they are known as singly diagonally implicit Runge-Kutta methdods (SDIRK).

Definition 4.1.1. *For the constrained system with nonsingular mass matrix M*

$$M\dot{x} = f(x,z) \tag{4.3}$$
$$0 = g(x,z) \tag{4.4}$$

we call the approximation to the solution by

$$x_{n+1} = x_n + \sum_{j=1}^{s} b_i k_i, \qquad z_{n+1} = z_n + \sum_{j=1}^{s} b_i l_i, \tag{4.5}$$

$$\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} k_i \\ l_i \end{pmatrix} = h \begin{pmatrix} f(v_i, w_i) \\ g(v_i, w_i) \end{pmatrix} + h \begin{bmatrix} f_x & f_z \\ g_x & g_z \end{bmatrix}_{|(x_n, z_n)} \sum_{j=1}^{i} \gamma_{ij} \begin{pmatrix} k_j \\ l_j \end{pmatrix}, \tag{4.6}$$

$$v_i = x_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j, \qquad w_i = z_n + \sum_{j=1}^{i-1} \alpha_{ij} l_j \tag{4.7}$$

for $i = 1 \dots s$, and coefficients $\alpha_{ij}, \gamma_{ij}, b_i$, a Rosenbrock method.

Remark 4.1.2. *a) As intended, the direct need of solving a nonlinear system is avoided.*

b) In the special case of all $\gamma_{ii} = \gamma$ only one decomposition of the iteration matrix

$$\underbrace{\left(\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} - h\gamma \begin{bmatrix} f_x & f_z \\ g_x & g_z \end{bmatrix}_{|(x_n,z_n)}\right)}_{\text{iteration-matrix}} \begin{pmatrix} k_i \\ l_i \end{pmatrix}$$
$$= h \begin{pmatrix} f(v_i, w_i) \\ g(v_i, w_i) \end{pmatrix} + h \begin{bmatrix} f_x & f_z \\ g_x & g_z \end{bmatrix}_{|(x_n,z_n)} \sum_{j=1}^{i-1} \gamma_{ij} \begin{pmatrix} k_j \\ l_j \end{pmatrix}$$

is needed. This is useful for obtaining fast implicit simulations.

c) Although in this chapter, for the sake of simplicity, we focus on equations of first order. For solving mechanical problems like those of Chapter 2, an extensions to the second order case will be useful. By considering the special structure of second order systems the cost while solving one step can be additionally reduced, this will be discussed in chapter 5.

Example 4.1.3 (Linear implicit Euler)**.** *The linear implicit Euler method can be interpreted as a first example of a Rosenbrock method. For the autonomous system it reads*

$$x_1 = x_0 + k_1,$$
$$k_1 = hf(x_0) + hf_x(x_0)k_1,$$

i.e., that the coefficients are $s = b_1 = \gamma_{11} = 1$. The method is convergent of order 1.

Lemma 4.1.4. *The stability function of a Rosenbrock method is*

$$R(h\lambda) = 1 + h\lambda b^T (\mathrm{Id} - h\lambda B)^{-1} \mathbb{1} \tag{4.8}$$

with $B = \left(\alpha_{ij} + \gamma_{ij}\right)_{ij}$ and $\mathbb{1} = \begin{pmatrix} 1 & \dots & 1 \end{pmatrix}^T$. The stability function evaluated at $h\lambda = \infty$ is

$$\rho_\infty = |R(\infty)| = |1 - b^T B^{-1} \mathbb{1}|,$$

the method is called L-stable *if $\rho_\infty = 0$.*

Proof. The stability function of the method is directly computed by inserting the test equation $\dot{x} = \lambda x$

$$x_1 = x_0 + \sum_{i=1}^{s} b_i k_i,$$
$$k_i = h\lambda \left(x_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_j \right) + h\lambda \sum_{j=1}^{i} \gamma_{ij} k_j$$
$$= h\lambda \left(x_0 + \sum_{j=1}^{i} (\alpha_{ij} + \gamma_{ij}) k_j \right).$$

So by inserting B we have with (4.8) that $x_1 = R(h\lambda)x_0$. The result for ρ_∞ follows by taking the limit $h\lambda \to \infty$. □

Classic convergence

For an ordinary differential equation we derive the conditions on the coefficients of the method, which are sufficient for its convergence. In the following we will assume, without loss of generality, that $M = \mathrm{Id}$. The given conditions are also derived in [HW96, Roc88], we will repeat those results for classic and index 1 convergence.

To obtain a convergent method of order p, i.e.,

$$x(t_0 + h) - x_1 = O(h^{p+1}),$$

we differentiate numerical and exact solutions. For this we write (4.5) in tensor notation for

$$k_j = (k_j^J)_{J=1\dots n}, \quad f(v_i) = (f^J(v_i))_{J=1\dots n}, \quad v_i = (v_i^J)_{J=1\dots n}, \quad x_i = (x_i^J)_{J=1\dots n},$$

we have

$$k_j^J = hf^J(v_j) + h\sum_K f_K^J(x_0)\sum_k \gamma_{jk} k_k^K, \tag{4.9}$$

$$v_i^J = x_0^J + \sum_j \alpha_{ij} k_j^J,$$

$$x_1^J = x_0^J + \sum_j b_j k_j^J. \tag{4.10}$$

Here we use the notation $f_K^J = \frac{\partial f^J}{\partial x^K}$, subsequent labels will stand for additional differentiation like $f_{KL}^J = \frac{\partial^2 f^J}{\partial x^K \partial x^L}$.

The q-th differential of the stage k_j^J (4.9) is

$$(k_j^J)^{(q)}|_{h=0} = q(f^J(v_j))^{(q-1)}|_{h=0} + q\sum_K f_K^J(x_0)\sum_k \gamma_{ik}(k_k^K)^{(q-1)}|_{h=0}. \tag{4.11}$$

In the following we will omit the subscript $h = 0$ and always evaluate at this point. For evaluating $(f^J(v_j))^{(q)}$ we use the chain rule

$$(f^J(v_j))^{(1)} = \sum_K f_K^J(v_j)\cdot(v_j^K)^{(1)},$$

$$(f^J(v_j))^{(2)} = \sum_{K,L} f_{KL}^J(v_j)\cdot(v_j^K)^{(1)}(v_j^L)^{(1)} + \sum_K f_K^J(v_j)\cdot(v_j^K)^{(2)}.$$

Inserted into (4.11) we get

$$(k_j)^{(1)} = f^J,$$

$$(k_j)^{(2)} = 2\sum_K f_K^J f^K \sum_k \alpha_{jk} + 2\sum_K f_K^J f^K \sum_k \gamma_{jk}$$

$$= 2\sum_K f_K^J f^K \sum_k (\alpha_{jk} + \gamma_{jk}),$$

$$(k_j)^{(3)} = 3\sum f_{KL}^J f^K f^L \sum_{k,l} \alpha_{jk}\alpha_{jl} + 6\sum_{K,L} f_K^J f_L^K f^L \sum_{k,l} (\alpha_{jk} + \gamma_{jk})(\alpha_{kl} + \gamma_{kl}).$$

order	condition	
1	$\sum_i b_i$	$= 1$
2	$\sum_{i,k} b_i \beta'_{ik}$	$= \frac{1}{2} - \gamma$
3	$\sum_{i,k,l} b_i \alpha_{ik} \alpha_{il}$	$= \frac{1}{3}$
3	$\sum_{i,k,l} b_i \beta'_{ik} \beta'_{kl}$	$= \frac{1}{6} - \gamma + \gamma^2$
4	$\sum_{i,k,l,m} b_i \alpha_{ik} \alpha_{il} \alpha_{im}$	$= \frac{1}{4}$
4	$\sum_{i,k,l,m} b_i \alpha_{ik} \beta'_{kl} \alpha_{jm}$	$= \frac{1}{8} - \frac{\gamma}{3}$
4	$\sum_{i,k,l,m} b_i \beta'_{ik} \alpha_{kl} \alpha_{km}$	$= \frac{1}{12} - \frac{\gamma}{3}$
4	$\sum_{i,k,l,m} b_i \beta'_{ik} \beta'_{kl} \beta'_{lm}$	$= \frac{1}{24} - \frac{\gamma}{2} + \frac{3}{2}\gamma^2 - \gamma^3$

Table 4.1: Order conditions up to order 4

By further differentiating and inserting into (4.10) we get the resulting differential

$$(x_1^J)^{(q)} = \sum_j b_j (k_j^J)^{(q)}|_{h=0}. \tag{4.12}$$

While differentiating the true solution yields

$$\begin{aligned}
(x^J)^{(1)} &= f^J(x), \\
(x^J)^{(2)} &= \sum_K f_K^J(x) \cdot (x^K)^{(1)} = \sum_K f_K^J(x) f^K(x), \\
(x^J)^{(3)} &= \sum_{K,L} f_{KL}^J(x) f^K(x) f^L(x) + \sum_{K,L} f_K^J(x) f_L^K(x) f^L(x).
\end{aligned} \tag{4.13}$$

Comparing the coefficient of (4.13) with those of the exact solution (4.12), we arrive at the following conditions for a method of order 3

$$\begin{gathered}
\sum b_j = 1, \\
\sum b_j (\alpha_{jk} + \gamma_{jk}) = \frac{1}{2}, \\
\sum b_j \alpha_{jk} \alpha_{jl} = \frac{1}{3}, \qquad \sum b_j (\alpha_{jk} + \gamma_{jk})(\alpha_{kl} + \gamma_{kl}) = \frac{1}{6}.
\end{gathered}$$

For higher order methods the arising terms become bigger and harder to write down. The procedure of obtaining the coefficients can be nicely formalized by using labeled trees which than give a comprehensive view on the different conditions that a method has to fulfill. Since we are not heading to higher order, we will not introduce these. For convenience we wrote down the order conditions up to order 4 in Table 4.1, setting $\gamma = \gamma_{ii} = \gamma_{jj}$ and using the abbreviations

$$\beta_{ij} = \alpha_{ij} + \gamma_{ij}, \qquad \beta'_{ij} = \begin{cases} \beta_{ij} \\ 0 \text{ for } i = j \end{cases}.$$

Index 1 convergence

In the index 1 case we will use the same idea of comparing coefficients in the Taylor expansion. This time we start by calculating the derivatives of the exact solution. First of all, we have to remove the constraint. By differentiation of (4.4) we obtain the equation

$$z^{(1)} = (-g_z^{-1})g_x f.$$

Together with (4.3) we can now calculate further differentials of the exact solution using the differential of the inverse mapping and the chain rule

$$\begin{aligned} x^{(2)} &= f_x x^{(1)} + f_z z^{(1)} = f_x f + f_z(-g_z^{-1})g_x f, \\ z^{(2)} &= (-g_z^{-1})(g_{zx}((-g_z^{-1})g_x f, f) + g_{zz}((-g_z^{-1})g_x f, (-g_z^{-1})g_x f)) \\ &\quad + (-g_z^{-1})(g_{xx}(f,f) + g_{xz}(f,(-g_z^{-1}g_x f)) + (-g_z^{-1})g_x(f_x f + f_z(-g_z^{-1})g_x f). \end{aligned} \tag{4.14}$$

By a Taylor expansion of the numerical solution (4.5) we now have for the differential of the stages (everything evaluated at $h = 0$)

$$k_i^{(q)} = q(f(v_i, w_i))^{(q-1)} + (f_x)_0 q \sum_{j=1}^{i} \gamma_{ij} k_j^{(q-1)} + (f_z)_0 q \sum_{j=1}^{i} \gamma_{ij} l_j^{(q-1)},$$

and the differential of the constraint row after dividing by h is

$$0 = (g(v_i, w_i))^{(q)} + (g_x)_0 \sum_{j=1}^{i} \gamma_{ij} k_j^{(q)} + (g_z)_0 \sum_{j=1}^{i} \gamma_{ij} l_j^{(q)}. \tag{4.15}$$

For the differentials $(f(v_i, w_i))^{(q-1)}$ and $(g(v_i, w_i))^{(q)}$ we may use Faà di Bruno's formula or directly calculate using the chain rule

$$\begin{aligned} (g(v_i, w_i))^{(1)} &= g_x v_i^{(1)} + g_z w_i^{(1)}, \\ (g(v_i, w_i))^{(2)} &= g_{xx}(v_i^{(1)}, v_i^{(1)}) + g_x(v_i^{(2)}) + g_{xz}(v_i^{(1)}, w_i^{(1)}) \\ &\quad + g_{zx}(w_i^{(1)}, v_i^{(1)}) + g_z(w_i^{(2)}) + g_{zz}(w_i^{(1)}, w_i^{(1)}). \end{aligned} \tag{4.16}$$

Lemma 4.1.5. *The q-th derivative of $f(v, w)$ can be represented as*

$$f(v,w)^{(q)} = \sum_{(m,n)\in Q} \frac{\partial^{m+n} f(v,w)}{\partial x^m \partial z^n} \left(v^{(\mu_1)}, \cdots, v^{(\mu_m)}, w^{(\nu_1)}, \cdots, w^{(\nu_n)} \right)$$

for some set Q and coefficients μ_i, ν_i fulfilling additionally $\sum \mu_i + \sum \nu_j = q$.

Proof. Use Faà di Brunos's formula or see [HNW08]. □

Inserting

$$v_i^{(q)} = \sum_{j=1}^{i-1} \alpha_{ij} k_j^{(q)}, \qquad w_i^{(q)} = \sum_{j=1}^{i-1} \alpha_{ij} l_j^{(q)}$$

order	condition	
2	$\sum b_j \omega_{jk} \alpha_{kl} \alpha_{km}$	$= 1$
3	$\sum b_j \omega_{jk} \alpha_{kl} \alpha_{km} \alpha_{kn}$	$= 1$
3	$\sum b_j \omega_{jk} \alpha_{kl} \alpha_{km} \beta'_{mn}$	$= \frac{1}{2} - \gamma$
3	$\sum b_j \omega_{jk} \alpha_{kl} \alpha_{km} \omega_{mn} \alpha_{np} \alpha_{nq}$	$= 1$

Table 4.2: Order conditions of the z component for index 1 Rosenbrock methods up to order 3

into $(g(v_i, w_i))^{(q)}$ using Lemma 4.1.5 and (4.16) we have that (4.15) is

$$0 = \sum_{\substack{(m,n)\in Q \\ \text{s.t. } m+n\geq 2}} \frac{\partial^{m+n} g(x_0, z_0)}{\partial x^m \partial z^n} \left(\sum_{j=1}^{i-1} \alpha_{ij} k_j^{(\mu_1)}, \cdots, \sum_{j=1}^{i-1} \alpha_{ij} l_j^{(\nu_1)}, \cdots \right) + (g_y)_0 \sum_{j=1}^{i} \beta_{ij} k_j^{(q)} + (g_z)_0 \sum_{j=1}^{i} \beta_{ij} l_j^{(q)} \quad (4.17)$$

with $\beta_{ij} = \alpha_{ij} + \gamma_{ij}$. Now (4.17) can be solved for $l_j^{(q)}$ using $(\omega_{ij})_{ij} = ((\beta_{ij})_{ij})^{-1}$,

$$l_i^{(q)} = (-g_z)_0^{-1} \sum_{j=1}^{i} \omega_{ij} \sum_{\substack{(m,n)\in Q \\ \text{s.t. } m+n\geq 2}} \frac{\partial^{m+n} g(x_0, z_0)}{\partial x^m \partial z^n} \left(\sum_{j=1}^{i-1} \alpha_{ij} k_j^{(\mu_1)}, \cdots, \sum_{j=1}^{i-1} \alpha_{ij} l_j^{(\nu_1)}, \cdots \right) + ((-g_z^{-1}) g_y)_0 k_i^{(q)} \quad (4.18)$$

and also for $k_i^{(q)}$

$$k_i^{(q)} = q \sum_{\substack{(m,n)\in Q \\ \text{s.t. } m+n\geq 2}} \frac{\partial^{m+n} f(x_0, z_0)}{\partial x^m \partial z^n} \left(\sum_{j=1}^{i-1} \alpha_{ij} k_j^{(\mu_1)}, \cdots, \sum_{j=1}^{i-1} \alpha_{ij} l_j^{(\nu_1)}, \cdots \right) + q(f_y)_0 \sum_{j=1}^{i-1} \beta_{ij} k_j^{(q-1)} + q(f_z)_0 \sum_{j=1}^{i} \beta_{ij} l_j^{(q-1)}. \quad (4.19)$$

By inserting (4.18) and (4.19) for $q \geq 1$ into

$$x_1^{(q)} = \sum_{j=1}^{s} b_j k_j^{(q)} \qquad \text{and} \qquad z_1^{(q)} = \sum_{j=1}^{s} b_j l_j^{(q)},$$

respectively, we finally have an expansion of the numerical solution. Comparing the coefficients with those of (4.14) we obtain the conditions on the order. For the z component these are listed up to order four in Table 4.2. Starting with order four also additional conditions for the x component arise, see Table 4.3.

Theorem 4.1.6 (Convergence). *For the index 1 system with consistent initial values* (x_0, z_0), *stability function* $|R(\infty)| < 1$ *and local errors*

$$x_1 - x(t_0 + h) = O(h^{p+1}), \qquad z_1 - z(t_0 + h) = O(h^p),$$

order	condition
4	$\sum b_j \alpha_{jk} \alpha_{jl} \omega_{lm} \alpha_{mn} \alpha_{mp} = 1$

Table 4.3: Additional order condition for the x component in the index 1 case up to order 4

a Rosenbrock method is convergent of order p, i.e.,

$$x_n - x(t_n) = O(h^p), \qquad z_n - z(t_n) = O(h^p)$$

for $t_n - t_0 = nh \leq C$.

Proof. [HW96] □

Singularly perturbed systems

In the search of additional conditions, which could arise for singular singularly perturbed systems, we consider the Prothero-Robinson type equation (Example 3.1.12) as a test problem. Similar approaches have also be used for first order [Sch89] and index 0 formulations [Sim98]. The crucial question is if we observe an order-reduction, and if we can avoid it by a clever choice of coefficients.

Remark 4.1.7. *The Prothero-Robinson equation is also considered for the construction of B-convergent Runge-Kutta methods [PR74].*

We are going to apply a general Rosenbrock method to the linear test-equation. After this, we will look at the local error to find sources for a possible order reduction. Since the Prothero-Robinson equation

$$\begin{aligned} \begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} q \\ \dot{q} \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \lambda + \begin{pmatrix} 0 \\ \ddot{\phi}(t) \end{pmatrix} \\ \epsilon^2 \lambda &= q - \phi(t) \end{aligned} \tag{4.20}$$

is not autonomous, we have to use an adequate form of the Rosenbrock method. However, the autonomization of the non-autonomous equation will be introduced in Section 5.1. So we apply (5.6) to system (4.20) using

$$J = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & -\epsilon^2 \end{bmatrix}, \qquad J_t = \begin{pmatrix} 0 \\ \phi^{(3)}(t_0) \\ -\dot{\phi}(t_0) \end{pmatrix}, \qquad \bar{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.21}$$

(4.20) reads

$$\begin{pmatrix} \dot{q} \\ \ddot{q} \\ 0 \end{pmatrix} = f(q, \dot{q}, z, t) = J \begin{pmatrix} q \\ \dot{q} \\ z \end{pmatrix} + \begin{pmatrix} 0 \\ \ddot{\phi}(t) \\ -\phi(t) \end{pmatrix}$$

and the non-autonomous Rosenbrock method is

$$\begin{pmatrix} q_{1,q} \\ q_{1,v} \\ z_1 \end{pmatrix} = \begin{pmatrix} q_{0,q} \\ q_{0,v} \\ z_0 \end{pmatrix} + \sum_{j=1}^{s} b_j \begin{pmatrix} k_{j,q} \\ k_{j,v} \\ l_j \end{pmatrix}, \tag{4.22}$$

$$\begin{aligned} \bar{M} \begin{pmatrix} k_{i,q} \\ k_{i,v} \\ l_i \end{pmatrix} &= hf \left(\begin{pmatrix} q_0 + \sum_{j=1}^{i-1} \alpha_{ij} \begin{pmatrix} k_{j,q} \\ k_{j,v} \\ l_j \end{pmatrix}, t_0 + \alpha_i h \end{pmatrix} + h^2 \gamma_i J_t + hJ \sum_{j=1}^{i} \gamma_{ij} \begin{pmatrix} k_{j,q} \\ k_{j,v} \\ l_j \end{pmatrix} \right. \\ &= hJ \left(\begin{pmatrix} q_0 + \sum_{j=1}^{i-1} \alpha_{ij} \begin{pmatrix} k_{j,q} \\ k_{j,v} \\ l_j \end{pmatrix} \end{pmatrix} + h\phi_i + h^2 \gamma_i J_t + hJ \sum_{j=1}^{i} \gamma_{ij} \begin{pmatrix} k_{j,q} \\ k_{j,v} \\ l_j \end{pmatrix} \right. \\ &= hJ \left(\begin{pmatrix} q_0 + \sum_{j=1}^{i} \beta_{ij} \begin{pmatrix} k_{j,q} \\ k_{j,v} \\ l_j \end{pmatrix} \end{pmatrix} + h\phi_i + h^2 \gamma_i J_t, \right. \end{aligned} \tag{4.23}$$

where

$$\phi_i = \begin{pmatrix} 0 \\ \ddot{\phi}(t_0 + \alpha_i h) \\ -\phi(t_0 + \alpha_i h) \end{pmatrix}.$$

The last row of (4.23) reads

$$h\epsilon^2 \sum_{j=1}^{i} \beta_{ij} lj = h\tilde{q}_{0,3} + h \sum_{j=1}^{i} \beta_{ij} k_{j,q} - h\phi(t_0 + \alpha_i h) - h^2 \gamma_i \dot{\phi}(t_0). \tag{4.24}$$

By inserting (4.24) into the second row of (4.23) we can eliminate the constraint from the system as long as $\epsilon > 0$

$$\begin{aligned} k_{i,v} &= h\tilde{q}_{0,2.1} - h \sum_{j=1}^{i} \beta_{ij} l_j + h\ddot{\phi}(t_0 + \alpha_i h) + h^2 \gamma_i \phi^{(3)}(t_0) \\ &= h\tilde{q}_{0,2.2} - \frac{h}{\epsilon^2} \sum_{j=1}^{i} \beta_{ij} k_{j,q} + \frac{h}{\epsilon^2} \phi(t_0 + \alpha_i h) + \frac{h^2}{\epsilon^2} \gamma_i \dot{\phi}(t_0) \\ &\quad + h\ddot{\phi}(t_0 + \alpha_i h) + h^2 \gamma_i \phi^{(3)}(t_0). \end{aligned}$$

Collecting all terms, we write (4.23) for $\epsilon > 0$ as

$$\begin{aligned} \begin{pmatrix} k_{i,q} \\ k_{i,v} \end{pmatrix} &= h \begin{bmatrix} 0 & 1 \\ -\epsilon^{-2} & 0 \end{bmatrix} \left(q_0 + \sum_{j=1}^{i} \beta_{ij} \begin{pmatrix} k_{j,q} \\ k_{j,v} \end{pmatrix} - \begin{pmatrix} \phi(t_0 + \alpha_i h) \\ \dot{\phi}(t_0 + \alpha_i h) \end{pmatrix} \right) \\ &+ h \begin{pmatrix} \dot{\phi}(t_0 + \alpha_i h) \\ \ddot{\phi}(t_0 + \alpha_i h) \end{pmatrix} + h^2 \gamma_i \left(- \begin{bmatrix} 0 & 1 \\ -\epsilon^{-2} & 0 \end{bmatrix} \begin{pmatrix} \dot{\phi}(t_0) \\ \ddot{\phi}(t_0) \end{pmatrix} + \begin{pmatrix} \ddot{\phi}(t_0) \\ \phi^{(3)}(t_0) \end{pmatrix} \right). \end{aligned} \tag{4.25}$$

Solving equation (4.25) is equivalent to solving (4.23).

Lemma 4.1.8. *The stability function corresponding to the Rosenbrock method applied to the Prothero-Robinson equation after eliminating the constraint* (4.25) *is*

$$R(hJ) = \mathrm{Id}_2 + (b^T \otimes \mathrm{Id}_2)(\mathrm{Id}_{2s} - B \otimes hJ)^{-1}(\mathbb{1} \otimes hJ).$$

By reordering the terms, such that the blocks of $R(hJ)$ correspond to the components of $x = \begin{pmatrix} q & \dot{q} \end{pmatrix}^T$, we have that

$$R(hJ) = \begin{bmatrix} 1 - h^2\epsilon^{-2} b^T S B & h b^T S \\ -h\epsilon^{-2} b^T S & 1 - h^2\epsilon^{-2} b^T S B \end{bmatrix}$$

for $S = (\mathrm{Id} + h^2\epsilon^{-2}B^2)^{-1}$ and

$$J = \begin{bmatrix} 0 & 1 \\ -\epsilon^{-2} & 0 \end{bmatrix}. \tag{4.26}$$

Proof. By the Rosenbrock method we have that

$$x_1 = x_0 + (b^T \otimes \mathrm{Id}_2)K_s = x_0 + (\mathrm{Id}_2 \otimes b^T)K_g \tag{4.27}$$

for different orderings of the components

$$K_s = \begin{bmatrix} k_{1,q} & k_{1,v} & \cdots & k_{s,q} & k_{s,v} \end{bmatrix}^T,$$
$$K_g = \begin{bmatrix} k_{1,q} & \cdots & k_{s,q} & k_{1,v} & \cdots & k_{s,v} \end{bmatrix}^T.$$

Using (4.26) we see from (4.25) that

$$\begin{aligned} & K_s = (\mathrm{Id}_s \otimes hJ)(\mathbb{1} \otimes x_0 + (B \otimes \mathrm{Id}_2)K_s) + r_s \\ \Leftrightarrow \quad & (\mathrm{Id}_{2s} - B \otimes hJ)K_s = (\mathrm{Id}_s \otimes hJ)(\mathbb{1} \otimes x_0) + r_s, \end{aligned}$$

where r is the term not containing x_0 or K_s, while for the version ordered by components

$$\begin{aligned} & K_g = (hJ \otimes \mathrm{Id}_s)\left(x_0 \otimes \mathbb{1} + (\mathrm{Id}_2 \otimes B)K_g\right) + r_g \\ \Leftrightarrow \quad & (\mathrm{Id}_{2s} - hJ \otimes B)K_g = (hJ \otimes \mathrm{Id}_s)(x_0 \otimes \mathbb{1}) + r_g. \end{aligned}$$

Inserting K_s into (4.27) gives

$$R(hJ) = \mathrm{Id}_2 + (b^T \otimes \mathrm{Id}_2)(\mathrm{Id}_{2s} - B \otimes hJ)^{-1}(hJ \otimes \mathbb{1}).$$

By inserting K_g we evaluate the ordered version. There we have

$$(\mathrm{Id}_2 \otimes b^T)(\mathrm{Id}_{2s} - hJ \otimes B)^{-1} = \begin{bmatrix} b^T S & h b^T S B \\ -h\epsilon^{-2} b^T S B & b^T S \end{bmatrix}$$

and finally

$$R(hJ) = \begin{bmatrix} 1 - h^2\epsilon^{-2} b^T S B & h b^T S \\ -h\epsilon^{-2} b^T S & 1 - h^2\epsilon^{-2} b^T S B \end{bmatrix}.$$

□

Using Lemma 4.1.8 we obtain the local error

$$\delta_1 = \phi(t_0) - \phi(t_0+h) + (b^T \otimes \mathrm{Id}_2)(\mathrm{Id}_{2s} - B \otimes hJ)^{-1} \left(\mathbb{1} \otimes hJ x_0 + \begin{pmatrix} r_q \\ r_v \end{pmatrix} \right),$$

where

$$\begin{aligned} r_q &= 0, \\ r_v &= h\epsilon^{-2}(\phi(t_0+\alpha_i h) + h\gamma_i \dot{\phi}(t_0)) + h\ddot{\phi}(t_0+\alpha_i h) + h^2\gamma_i \phi^{(3)}(t_0). \end{aligned}$$

Looking closer into δ_1 there are two components

$$\delta_{1,q} = \phi(t_0) - \phi(t_0+h) + b^T S(-h^2\epsilon^{-2}B\mathbb{1}\phi(t_0) + \mathbb{1}h\dot{\phi}(t_0) + hBr_v), \tag{4.28}$$

$$\delta_{1,v} = \dot{\phi}(t_0) - \dot{\phi}(t_0+h) + b^T S(-\mathbb{1}h\epsilon^{-2}\phi(t_0) - h^2\epsilon^{-2}B\mathbb{1}\dot{\phi}(t_0) + r_v). \tag{4.29}$$

Inserting r_q, r_v we can expand the local error for a consistent method (i.e., with $\sum_{i=1}^{s} b_i = 1$) into

$$\delta_{1,q} = \sum_{i\geq 2} \frac{h^i}{i!}\xi_i \phi^{(i)}(t_0), \qquad \delta_{1,v} = \sum_{i\geq 2} \frac{h^{i-1}}{(i-1)!}\zeta_i \phi^{(i)}(t_0). \tag{4.30}$$

The first terms of ξ_i, ζ_i are

$$\begin{aligned} \xi_2 &= -1 + b^T SB(h^2\epsilon^{-2}(\alpha_i^2) + 2\mathbb{1}), \\ \zeta_2 &= -1 + b^T S(h^2\epsilon^{-2}(\alpha_i^2)/2 + \mathbb{1}), \\ \xi_3 &= -1 + b^T SB(h^2\epsilon^{-2}(\alpha_i^3) + 6(\alpha_i + \gamma_i)), \\ \zeta_3 &= -1 + b^T S(h^2\epsilon^{-2}(\alpha_i^3)/3 + 2(\alpha_i + \gamma_i)), \\ \xi_4 &= -1 + b^T SB(h^2\epsilon^{-2}(\alpha_i^4) + 12(\alpha_i^2)), \\ \zeta_4 &= -1 + b^T S(h^2\epsilon^{-2}(\alpha_i^4)/4 + 3(\alpha_i^2)). \end{aligned} \tag{4.31}$$

Examining ξ_2 further for how it behaves in the limit $\epsilon \to 0$

$$\begin{aligned} \lim_{\epsilon\to 0} \xi_2 &= -1 + \lim_{\epsilon\to 0} b^T(1 - h^2\epsilon^{-2}B^2)^{-1}B(h^2\epsilon^{-2}(\alpha_i^2) + 2\mathbb{1}) \\ &= -1 + b^T B^{-1}(\alpha_i^2)_{i=1\ldots s} \end{aligned} \tag{4.32}$$

we see that the order of the method can exceed 1 iff (4.32) vanishes. This means the order drops in the stiff case and thereby motivates us to introduce stiffly accurate methods.

Definition 4.1.9 ([HLR89b]). *A s-stage Rosenbrock method is called* stiffly accurate *iff*

$$\alpha_{si} + \gamma_{si} = b_i, \qquad \sum_{i=1}^{s-1} \alpha_{si} = 1.$$

Remark 4.1.10. *Advantages of stiffly accurate methods are also observed for Runge-Kutta methods B-convergence [PR74].*

Lemma 4.1.11. *Stiffly accurate Rosenbrock methods are L-stable*

Proof. Inserting Definition 4.1.9 into the stability function Lemma 4.1.4 for a stiffly accurate method, the last row of B equals b^T and

$$b^T B^{-1} = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}$$

thus the formula for ρ_∞ gives

$$\rho_\infty = 1 - b^T B^{-1} \mathbb{1} = 0.$$

□

Lemma 4.1.12. *For stiffly accurate Rosenbrock methods it holds that*

$$b^T B^{-1} (\alpha_i)^k = 1 \qquad \text{for all} \qquad k \in \mathbb{N}.$$

Proof. The result follows, since for stiffly accurate methods it holds that

$$\alpha_s = \sum_{i=1}^{s} \alpha_{s,i} = 1 \qquad \text{and} \qquad b^T B^{-1} = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}.$$

□

So, for stiffly accurate methods $\lim_{\epsilon \to 0} \xi_2 = 0$ but on the other side for $\delta_{1,v}$ (4.29) the limit

$$\lim_{\epsilon \to 0} \zeta_2 = -1 + \frac{1}{2} b^T B^{-2} (\alpha_i^2)_{i=1..s}$$

does not vanish. So we obtain a local error of

$$\delta_{1,q} = O(\epsilon^2) + O(\epsilon^2 h), \qquad \delta_{1,v} = O(h) + O(\epsilon^2). \tag{4.33}$$

But is this enough for avoiding an order reduction? To get further insight we recover the previously eliminated z component of the solution by back-substitution of the obtained solution of (4.25) into (4.24), and evaluating $z_1 = z_0 + \sum_{j=1}^{s} b_j l_j$ in (4.22). From (4.24) we have

$$h\epsilon^2 B l = h\phi(t_0) + hBk_q - h\phi(t_0 + \alpha_i h) - h^2 \gamma_i \dot{\phi}(t_0). \tag{4.34}$$

Using again the stability function, we obtain k_q like in $\delta_{1,q}$

$$k_q = S(-h^2 \epsilon^{-2} B\phi(t_0) + \mathbb{1} h\dot{\phi}(t_0) + hBr_v). \tag{4.35}$$

By inserting (4.35) into (4.34)

$$\begin{aligned} h\epsilon^2 B l = h\phi(t_0) + hBS(-h^2 \epsilon^{-2} B\phi(t_0) + \mathbb{1} h\dot{\phi}(t_0) + hBr_v) \\ - h\phi(t_0 + \alpha_i h) - h^2 \gamma_i \dot{\phi}(t_0), \end{aligned}$$

we have the stage vector l to be

$$\begin{aligned} l = \epsilon^{-2} B^{-1} \phi(t_0) + \epsilon^{-2} S(-h^2 \epsilon^{-2} B\phi(t_0) + \mathbb{1} h\dot{\phi}(t_0) + hBr_v) \\ - \epsilon^{-2} B^{-1} \phi(t_0 + \alpha_i h) - h\gamma_i B^{-1} \dot{\phi}(t_0). \end{aligned} \tag{4.36}$$

Theorem 4.1.13. *Let ξ_i be the expansion coefficients* (4.30) *for the singular singularly perturbed system* (4.20) *and the error in z be denoted by*

$$\delta_{1,z} = \sum_{i\geq 2} \frac{h^i}{i!} \tau_i \phi^{(i)}(t_0),$$

further let $b^T B^{-1}(\alpha_i^n) = 1$, then it holds that

$$\tau_i = \frac{1}{\epsilon^2} \xi_i.$$

Proof. By using (4.36) we construct the solution $z_1 = z_0 + b^T l$. Since the exact solution is $\lambda \equiv 0$, we have $z_1 = b^T l$ and thus $\delta_{1,z} = z_1$.

We want to show that $\delta_{1,z} = \epsilon^{-2}\delta_{1,x}$ and thus both expressions share the same expansion in ξ, respectively τ. For this we look at the term $b^T B^{-1}\phi(t_0 + \alpha_i h)$, expanding it into a Taylor series and using $b^T B^{-1}(\alpha_i^n) = 1$ gives

$$\begin{aligned} b^T B^{-1}\phi(t_0 + \alpha_i h) &= b^T B^{-1} \sum_{i=1}^{\infty} \frac{\alpha_i^i h^i}{i!} \phi^{(i)}(t_0) \\ &= \sum_{i=1}^{\infty} \frac{h^i}{i!} \phi^{(i)}(t_0) \\ &= \phi(t_0 + h) \end{aligned}$$

such that

$$\begin{aligned} \delta_{1,z} = \frac{1}{\epsilon^2}(b^T B^{-1} \mathbb{1} \phi(t_0) - \phi(t_0 + h) + \\ b^T S(-h^2\epsilon^{-2} B\phi(t_0) + \mathbb{1} h\dot{\phi}(t_0) + hBr_v) - hb^T \gamma_i B^{-1}\dot{\phi}(t_0)). \end{aligned}$$

The result follows by comparing the terms which account to ξ_i from (4.28). □

Using Theorem 4.1.13 and observing τ_2, the local error in $\delta_{1,z}$ drops to $O(1)$.

Example 4.1.14. *One effect observed using methods, which have $\delta_{1,z} = O(1)$, is seen in this example for the linear implicit Euler method (seen in Example 4.1.3). By applying it to system* (4.20) *together with* (4.21) *we have to solve in each step*

$$\begin{aligned} (\bar{M} - hJ)k &= hf(x_0, t_0) + h^2 J_{t_0}, \\ x_1 &= x_0 + k. \end{aligned}$$

By choosing $\phi(t) = \cos(at)$ we have $\dot{\phi}(t) = -a\sin(at)$, $\ddot{\phi}(t) = -a^2 cos(at)$ and for a consistent initial value $x_0 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T$ evaluating f and J_t given as

$$f(x_0, t_0) = \begin{pmatrix} 0 \\ -a^2 \\ 0 \end{pmatrix}, \qquad J_{t_0} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

The solution after one step is

$$k = \begin{pmatrix} -a^2 h \left(h - \frac{h^3}{\epsilon^2+h^2}\right) \\ -a^2 h \left(1 - \frac{h^2}{\epsilon^2+h^2}\right) \\ -a^2 h^2 \frac{1}{\epsilon^2+h^2} \end{pmatrix}.$$

From the exact solution (3.25) *we know that λ is zero, i.e., in the numerical solution the z component has to be small, while for $\epsilon = 0$ the component is only small if $a^2 \ll 1$. In the case of $\epsilon > 0$ the behavior can also be compensated by h. But notice that h must at least satisfy $h^2 > \epsilon^2$.*

Doing a few steps of the implicit Euler method for $a = 6$ and $\epsilon^2 = 10^{-6}$, we see in Figure 4.1 that for $h^2 < \epsilon^2$ the solution makes a jump in the first iteration, and after this converges to a wrong solution. Only for $h^2 > \epsilon^2$ we stay close to the analytic solution.

The same behavior can be observed for the stiffly accurate method, e.g., RO2*, which will be presented in Example 4.2.3, see Figure 4.2, as here as well $\delta_{1,z} = O(1)$.*

This brings us to a different approach for the construction of methods which are fulfilling $\xi_i \equiv 0$ for arbitrary h and ϵ, these were first suggested and constructed in [Sch89].

Definition 4.1.15. *For $i \geq 2$ the further condition on the method's coefficients such that ξ_i, ζ_i* (4.30) *are zero independent of h and ϵ are called* Scholz conditions*. We say that the Scholz condition is fulfilled up to order k if it is fulfilled for all $2 \leq i \leq k$.*

Remark 4.1.16. *The Scholz conditions are independent from stiffly accuracy (as in Definition 4.1.9).*

The consequence to the local error of a Rosenbrock method for the Prothero-Robinson example are evident, since by construction of a method which fulfills the Scholz condition up to order k all terms ξ_i, ζ_i for $i \leq k$ vanish. So, for $k = 2$ we have using (4.31)

$$\delta_{1,q} = O(\epsilon^2 h) + O(\epsilon^2 h^2), \qquad \delta_{1,v} = O(h^2) + O(\epsilon^2),$$

which is already advantageous in comparison to (4.33).

In the next section we want to numerically compare different methods which do either fulfill the Scholz conditions or are stiffly accurate.

4.2 Overview of methods

For the singular singularly perturbed systems we have now the choice between several methods. We consider only methods which fulfill the additional conditions on index 1 systems and which are at least A-stable. Further, as argued in the previous subsection, the methods should either be stiffly accurate (and thus L-stable) or fulfill some of the Scholz conditions.

For efficiency we consider only methods with $\gamma = \gamma_i = \cdots = \gamma_s$ such that only one matrix decomposition is needed, even if we don't head for a very high order with many stages.

Remark 4.2.1. *The number of function calls has not to be equal to the number of stages, because of equal coefficients $\alpha_{ij} = \alpha_{kj}$ for some i, k, and all j in* (4.7).

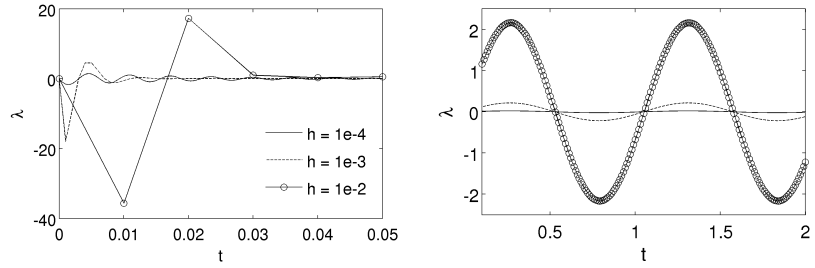


Figure 4.1: Numerical Solution of the z-component with linear implicit Euler method

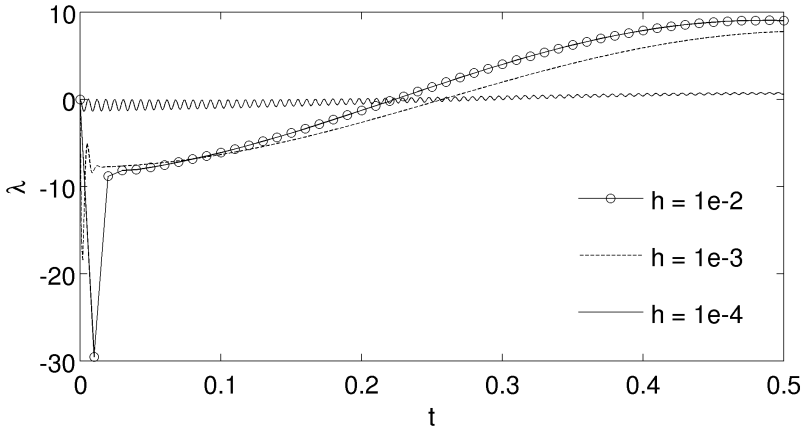


Figure 4.2: Solution by R02

The costs per step will be more dominated by the number of function evaluations then by the number of stages. Thus we have to minimize the use of function evaluations, which for Rosenbrock methods are only needed for a fixed number of times in one time-step since no iteration is done.

Remark 4.2.2. *The calculated Jacobi matrices can be used only for one step of the method. This can be circumvented by using so called W-methods [SW79]. They guaranty stability of the method independent of the used Jacobi matrix.*

For a stiffly accurate method, we give a small example how a Rosenbrock methods generally and a stiffly accurate methods in particular can be constructed.

Example 4.2.3 (R02)**.** *To obtain a second order stiffly accurate method with* 2 *stages we have to fulfill the order conditions of Table 4.1*

$$b_1 + b_2 = 1,$$
$$b_2(\alpha_{21} + \gamma_{21}) = \frac{1}{2} - \gamma.$$

For a stiffly accurate method we also require $b_2 = \gamma$, $\alpha_{21} + \gamma_{21} = b_1$ *and* $\alpha_{21} = 1$. *Inserting we obtain for* γ

$$\gamma^2 - 2\gamma + \frac{1}{2} = 0$$

and thus have the choices $\gamma_1 = 1 + \frac{\sqrt{2}}{2}$ *or* $\gamma_2 = 1 - \frac{\sqrt{2}}{2}$. *All other coefficients are already fixed with respect to* γ

$$\alpha_{ij} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \qquad \gamma_{ij} = \begin{bmatrix} \gamma & 0 \\ -\gamma & \gamma \end{bmatrix}, \qquad b = \begin{bmatrix} 1-\gamma & \gamma \end{bmatrix},$$
$$\omega_{ij} = \begin{bmatrix} \gamma^{-1} & 0 \\ (\gamma-1)\gamma^{-2} & \gamma^{-1} \end{bmatrix}.$$

Nevertheless the method already fulfills the first condition in Table 4.2. Since only $\alpha_{21} \neq 0$ *and the matrix* γ_{ij} *is triangular, we have* $\omega_{22} = \gamma^{-1}$ *and thus* $b_2\omega_{22} = 1$. *However notice that this condition is not needed for the convergence. As we saw in Theorem 4.1.6, we obtain a method convergent of order 2.*

For an error estimate we can construct an embedded method by choosing different coefficients in place of b_i *such that the method converges up to one order less. So, for an embedded method of order 1 only one condition is given*

$$\hat{b}_1 + \hat{b}_2 = 1.$$

We choose it to minimize $|R(\infty)|$,

$$R(\infty) = 1 - \hat{b}_1\gamma^{-1} - \hat{b}_2(2\gamma^{-1} - \gamma^{-2})$$

and thus solve the system for $R(\infty) = 0$ *by* $\hat{b} = \left(1 - \frac{\gamma}{1-\gamma} \quad \frac{\gamma}{1-\gamma}\right)$ *by inserting* $\gamma = \gamma_1$ *we obtain* $\hat{b} = \left(2+\sqrt{2} \quad -1-\sqrt{2}\right)$. *In the further we refer to the Rosenbrock method using this set of coefficients as* `RO2`.

How a general embedded method can be used for step-size selection is shown in the next chapter.

For a list of different stiffly accurate methods with increasing order see Table 4.4.

				embedded method			
Name	order	stages	f-calls	order	ρ_∞	s.a.	
`RO2`	2	2	2	1	0	-	Example 4.2.3
`ROWDA3`	3	3	2	2	.96	-	[Roc88]
`ROS3PL`	3	4	3	2	.25	-	[LT08]
`ROWDAIND2`	3	4	3	2	0	-	[Roc88]
`RODAS3`	3	4	3	2	0	✓	[SVB+97]
`RODAS4`	4	6	6	3	0	✓	[HW96]
`RODAS4P`	4	6	6	3	0	✓	[Ste95]

Table 4.4: Overview of stiffly accurate methods for index 1 systems

Methods which are constructed such that they do not only satisfy the order conditions but also several Scholz conditions are found in Table 4.5. Special attention is to be drawn to the order 4 method RODAS4P, which is a variant of RODAS4 [Ste95], since it fulfills the Scholz conditions up to order 3. Also we want to pick out the ROS3P method constructed in [LV01], which is designated for a minimal number of function evaluations and stages. The main compromise which has to be made while choosing ROS3P is its lack of L-stability. To achieve L-stability one additional step and one additional function evaluation is necessary, see for ROS3PL.

Name	order	stages	f-calls	s.a.	$\xi_i \equiv 0$	$\zeta_i \equiv 0$	ρ_∞	
RO2P	2	2	2	-	$i < 3$	$i < 3$	1.0 (!)	[Sch89]
ROS3P	3	3	2	-	$i < 3$	$i < 3$	.73	[LV01]
ROS3PL	3	4	3	✓	$i < 3$	$i < 3$	0	[LT08]
RODAS4P	4	6	6	✓	$i < 4$	$i < 4$	0	[Ste95]

Table 4.5: Overview of methods designed to fulfill additional Scholz conditions

Stability functions for methods of different number of stages are plotted in Figure 4.3. The more stages a method has the broader is the plateau of exactly transferred frequencies. For being interested in the smooth motion more damping at high frequencies seems to be advantageous as examined in [Stu04].

Remark 4.2.4. *Rosenbrock methods can also be directly constructed for higher-index systems. This was done up to the index* 3 *case, namely by* ROWDAIND2 *for index 2 systems, constructed in [HLR89a] (listed also in Table 4.4) and by a partitioned* 8*-stage fourth order method for index* 3 *systems, developed in [Wen98, Wen97].*

Numerical examples

For the given methods we would like to come back to the Prothero-Robinson Example (4.20) and have a look at the achieved convergence order of the different methods. For this purpose we apply the methods to Example 4.1.14 for $a = 6$ and

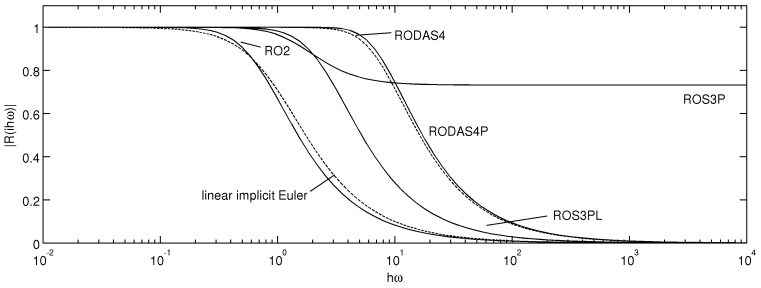


Figure 4.3: Stability function lemma 4.1.4 of some Rosenbrock methods evaluated at the imaginary axis

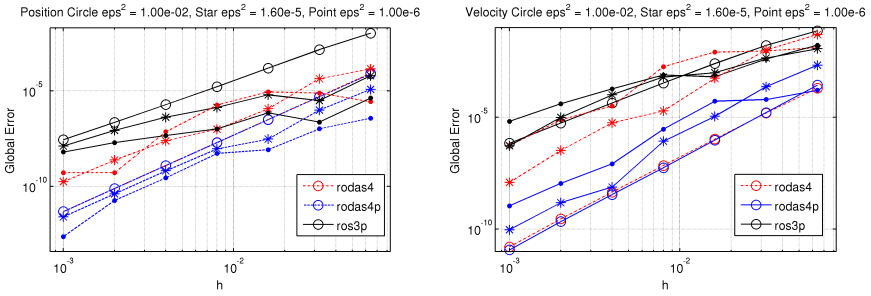


Figure 4.4: Convergence plot of index 1 Prothero-Robinson equation in position and velocity components comparison of RODAS4, RODAS4P and ROS3PL for different ϵ

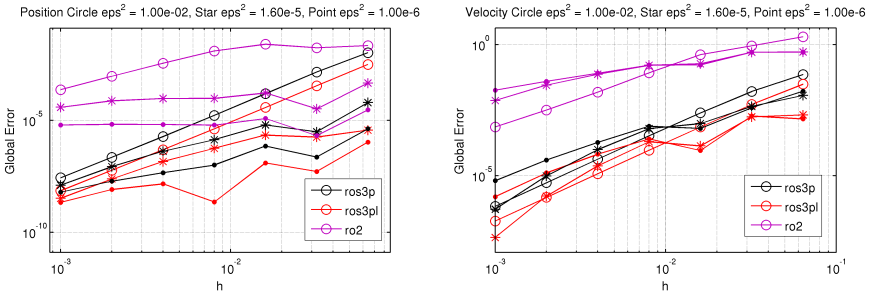


Figure 4.5: Convergence plot of index 1 Prothero-Robinson equation in position and velocity components comparison of ROS3P and stiffly accurate methods ROS3PL, RO2 for varying ϵ

$t_{end} = 2.2$. The plots are given in Figure 4.4 and Figure 4.5. For all methods we observe its classical order in the case of $\epsilon^2 = 0.01$. Decreasing ϵ^2 to 10^{-6}, for RODAS4 we see a loss in its order of positions as well as velocities, the method behaves in the same way as the much cheaper to compute ROS3P. By fulfilling that $\xi_2 \equiv \xi_3 \equiv \zeta_2 \equiv \zeta_3 \equiv 0$ RODAS4P shows no order reduction in position and velocity. For RO2 we see the order dropping to $O(1)$. A comparison between ROS3P and ROS3PL doesn't show an advantage of the L-stable method in our example.

Remembering our intention, we are looking for methods with a high efficiency. Thus, we want a minimum number of stages and possibly even less function calls in addition to good properties when considering a singular singularly perturbed system. A high order isn't necessary for this application, so the most economic methods by observing the linear example is ROS3P. The method has got only one additional stage and no additional function call compared to RO2 but brings the advantage in fulfilling some of the Scholz conditions. However, we have to compromise and accept that we loose L-stability.

Remark 4.2.5. *The plots for the z component of Figure 4.4 and Figure 4.5 are not shown because of their exact same behavior compared to those of the x component, except that they are scaled by their corresponding factor of ϵ^{-2}.*

4.3 Applied to perturbed nonlinear systems

As a last step, we want to observe the behavior of Rosenbrock methods for singularly perturbed nonlinear systems. We would like to have a look at the iteration matrix and how it behaves in the limiting case $\epsilon \to 0$

For the singular singularly perturbed system

$$\begin{aligned} \dot{q} &= v \\ \dot{v} &= f(q, v, \lambda) \\ 0 &= g(q) + \epsilon^2 \lambda \end{aligned} \quad , \qquad f_v = 0,$$

we are investigating the iteration Matrix $S_t = (\bar{M} - \hat{J})$ for Jacobi matrix J and mass matrix $\bar{M}$ of the Rosenbrock method with $\hat{h} = h\gamma$

$$S_t = \begin{bmatrix} \mathrm{Id} & -\hat{h}\,\mathrm{Id} & 0 \\ -\hat{h} f_q & \mathrm{Id} & -\hat{h} f_\lambda \\ -\hat{h} g_q & 0 & -\epsilon^2 \hat{h} \end{bmatrix}.$$

Using the Schur complement $K = A - \epsilon^{-2} B_1 B_2$ for the matrix S_t structured like

$$S_t = \begin{bmatrix} A & B_1 \\ B_2 & \mathrm{Id}\,\epsilon^{-2} \end{bmatrix},$$

we find its inverse

$$S_t^{-1} = \begin{bmatrix} \hat{h}^3 \left(\hat{h}^2 f_q + 1\right) f_\lambda K^{-1} g_q + \hat{h}^2 f_q + 1 & \hat{h}^4 f_\lambda K^{-1} g_q + \hat{h} & \hat{h}^2 f_\lambda K^{-1} \\ \hat{h}^2 \left(\hat{h}^2 f_q + 1\right) f_\lambda K^{-1} g_q + \hat{h} f_q & \hat{h}^3 f_\lambda K^{-1} g_q + 1 & \hat{h} f_\lambda K^{-1} \\ \hat{h} \left(\hat{h}^2 f_q + 1\right) K^{-1} g_q & \hat{h}^2 K^{-1} g_q & K^{-1} \end{bmatrix}.$$

Observing K and S_t^{-1}, we see that as long K is of full rank the iteration matrix stays regular even for $\epsilon \to 0$, as long as A is of full-rank.

Example of a stiff spring pendulum

The results for the linear Prothero-Robinson equation suggest an advantage for the specifically constructed methods. To see how the methods compare in a nonlinear setup we want to discuss the stiff spring pendulum also in the singular singularly perturbed index 1 formulation as seen in Example 3.1.8. The stiff spring in the pendulum can already be seen as a one dimensional incompressibility constraint, since it penalizes a change in the length of the pendulum.

First we take a look at the Jacobi matrix for the vector $x = \left(q_1, q_2, \dot{q}_1, \dot{q}_2\right)^T$

$$J_{x,\lambda} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\lambda & 0 & 0 & 0 & -q_1 \\ 0 & -\lambda & 0 & 0 & -q_2 \\ -Aq_1 & -Aq_2 & 0 & 0 & \epsilon^2 \end{bmatrix}, \qquad \bar{M} = \begin{bmatrix} \mathrm{Id}_4 & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.37}$$

$$\begin{bmatrix} -\frac{A\hat{h}^3q_1^2\left(-\hat{h}^2\lambda+1\right)}{K}-\hat{h}^2\lambda+1 & -\frac{A\hat{h}^3q_1q_2\left(-\hat{h}^2\lambda+1\right)}{K} & -\frac{A\hat{h}^4q_1^2}{K}+\hat{h} & -\frac{A\hat{h}^4q_1q_2}{K} & -\frac{h^2q_1}{K} \\ -\frac{A\hat{h}^3q_1q_2\left(-\hat{h}^2\lambda+1\right)}{K} & -\frac{A\hat{h}^3q_2^2\left(-\hat{h}^2\lambda+1\right)}{K}-\hat{h}^2\lambda+1 & -\frac{A\hat{h}^4q_1q_2}{K} & -\frac{A\hat{h}^4q_2^2}{K}+\hat{h} & -\frac{h^2q_2}{K} \\ -\frac{A\hat{h}^2q_1^2\left(-\hat{h}^2\lambda+1\right)}{K}-\hat{h}\lambda & -\frac{A\hat{h}^2q_1q_2\left(-\hat{h}^2\lambda+1\right)}{K} & -\frac{A\hat{h}^3q_1^2}{K}+1 & -\frac{A\hat{h}^3q_1q_2}{K} & -\frac{hq_1}{K} \\ -\frac{A\hat{h}^2q_1q_2\left(-\hat{h}^2\lambda+1\right)}{K} & -\frac{A\hat{h}^2q_2^2\left(-\hat{h}^2\lambda+1\right)}{K}-\hat{h}\lambda & -\frac{A\hat{h}^3q_1q_2}{K} & -\frac{A\hat{h}^3q_2^2}{K}+1 & -\frac{hq_2}{K} \\ \frac{A\hat{h}q_1\left(-\hat{h}^2\lambda+1\right)}{K} & \frac{A\hat{h}q_2\left(-\hat{h}^2\lambda+1\right)}{K} & \frac{A\hat{h}^2q_1}{K} & \frac{A\hat{h}^2q_2}{K} & \frac{1}{K} \end{bmatrix}$$

Figure 4.6: Inverse iteration matrix of stiff spring pendulum (4.38)

with

$$A=\frac{1}{q_1^2+q_2^2}-\frac{\sqrt{q_1^2+q_2^2}-1}{\left(q_1^2+q_2^2\right)^{\frac{3}{2}}}.$$

We can already give the iteration matrix, since only h and γ dependent on the actual choice of coefficients. So, for $\hat{h}=\gamma h$ we have

$$(\bar{M}-\hat{h}J)=\begin{bmatrix} 1 & 0 & -\hat{h} & 0 & 0 \\ 0 & 1 & 0 & -\hat{h} & 0 \\ \hat{h}\lambda & 0 & 1 & 0 & \hat{h}q_1 \\ 0 & \hat{h}\lambda & 0 & 1 & \hat{h}q_2 \\ A\hat{h}q_1 & A\hat{h}q_2 & 0 & 0 & -\epsilon^2\hat{h} \end{bmatrix}. \tag{4.38}$$

The Schur complement of matrix (4.38) is

$$K=-\frac{A\hat{h}^3(q_1^2+q_2^2)}{\hat{h}^2\lambda+1}-\epsilon^2h \tag{4.39}$$

and can be used to compute the inverse of (4.38), which is shown in Figure 4.6. As noted before, we observe no problems in the iteration matrix for $\epsilon\to 0$, since of

$$\lim_{\epsilon\to 0}K=-\frac{A\hat{h}^3(q_1^2+q_2^2)}{\hat{h}^2\lambda+1}$$

the inverse K^{-1} is not influenced by a small value of ϵ. We have that $K^{-1}=O(h^{-3})$.

Order behavior

For a numerical analysis of the global error we computed a reference solution using `RADAU5` and appropriate tolerances 10^{-12}. We evaluate the solution at $t_{end}=4$ which corresponds to two periods of the pendulum while setting `g = 13.7503716`. Afterwards we applied the different Rosenbrock methods to the problem and evaluated the difference in the q_1 and the $\dot{q}_1$ components.

In Figure 4.7 we compare `RODAS4` and its variant `RODAS4P` for a range of different ϵ^2. We observe the classical order for both methods. Doing the same

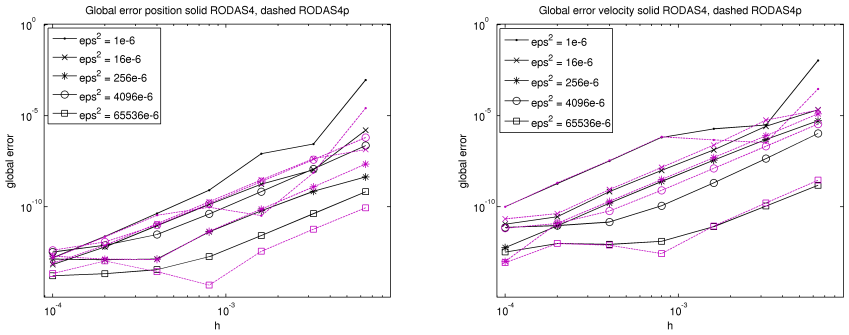


Figure 4.7: Global error behavior for the stiff spring pendulum dependent on ϵ and h for RODAS4 and RODAS4P in position and velocity component

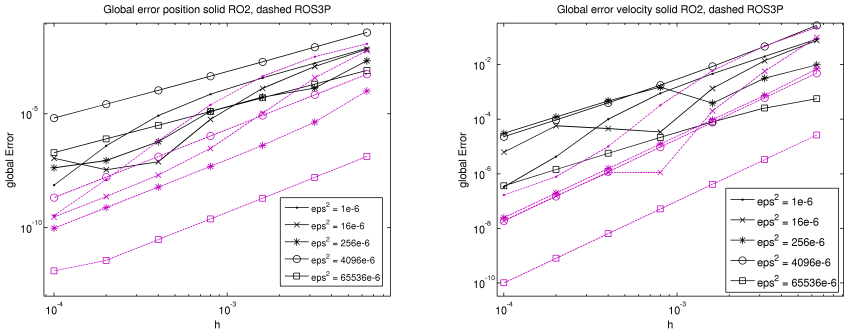


Figure 4.8: Global error behavior for the stiff spring pendulum dependent on ϵ and h for RO2 and ROS3P in position and velocity component

experiment for a comparison of the RO2 method and ROS3P (Figure 4.8) again we can see some influence of ϵ but can't really identify an order reduction. Also the direct comparison of $\epsilon^2 = 10^{-6}$ in Figure 4.9 shows the classical error behavior for the considered methods.

Only in the limiting case $\epsilon = 0$ (Figure 4.10) we observe a drop of order down to 2 of all methods but RODAS4P which is the only considered method fulfilling the additional conditions of Scholz for ξ_3, ζ_3, and more astonishing for ROWSPP4 (see Table 4.4) which fails to fulfill any of the Scholz conditions.

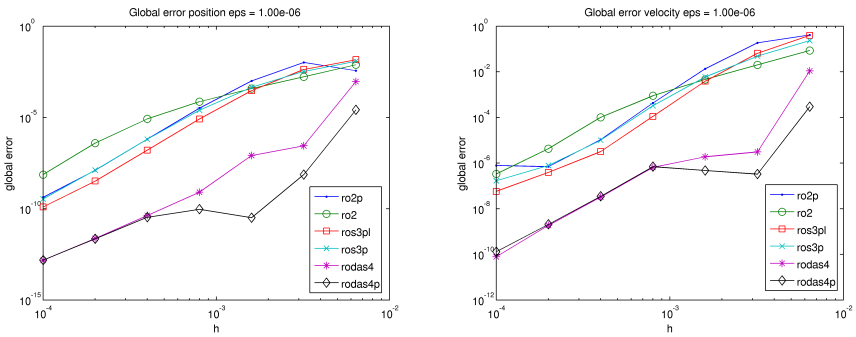


Figure 4.9: Comparison of global error behavior $\epsilon^2 = 10^{-6}$ for different Rosenbrock methods

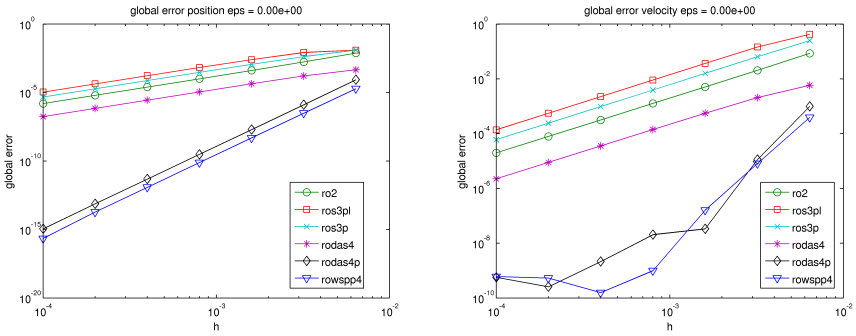


Figure 4.10: Order reduction in the stiff limit, $\epsilon = 0$

Chapter Five

Performance

5.1 Implementation

We have seen the construction and convergence properties of Rosenbrock methods in Chapter 4. Our interest now is to show how they can be efficiently implemented in the case of non-autonomous second order systems.

We start by the first order formulation for stiff systems

$$\begin{pmatrix} x_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} x_0 \\ z_0 \end{pmatrix} \sum_{i=1}^{s} b_i \begin{pmatrix} k_i \\ l_i \end{pmatrix}, \tag{5.1}$$

$$\underbrace{\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}}_{\tilde{M}} \begin{pmatrix} k_i \\ 0 \end{pmatrix} = h \begin{pmatrix} f(v_i, w_i) \\ g(v_i, w_i) \end{pmatrix} + h \begin{bmatrix} f_x & f_z \\ g_x & g_z \end{bmatrix} \sum_{j=1}^{i} \gamma_{ij} \begin{pmatrix} k_j \\ l_j \end{pmatrix}, \tag{5.2}$$

$$\begin{pmatrix} v_i \\ w_i \end{pmatrix} = \begin{pmatrix} x_0 \\ z_0 \end{pmatrix} + \sum_{j=1}^{i-1} \alpha_{ij} \begin{pmatrix} k_j \\ l_j \end{pmatrix}. \tag{5.3}$$

An extension to non-autonomous equations is done by adding an artificial variable t with time derivative and mass 1 to the equation. Considering the unconstrained system, adding the time leads to the augmented system

$$\begin{bmatrix} 1 & 0 \\ 0 & M \end{bmatrix} \begin{pmatrix} 1 \\ \dot{x} \end{pmatrix} = \begin{pmatrix} 1 \\ f(t, x) \end{pmatrix}, \tag{5.4}$$

which can be solved explicitly for t in (5.2) since the Jacobian of (5.4) is

$$J = \begin{bmatrix} 0 & 0 \\ f_t & f_x \end{bmatrix}.$$

The relation for the stage k_i, split into $\begin{pmatrix} k_{i,t} & k_{i,x} \end{pmatrix}^T$, in (5.2) is

$$\begin{bmatrix} 1 & 0 \\ 0 & M \end{bmatrix} \begin{pmatrix} k_{i,t} \\ k_{i,x} \end{pmatrix} = h \begin{pmatrix} 1 \\ f(t_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_{i,t}, v_i) \end{pmatrix} + hJ \sum_{j=1}^{i} \gamma_{ij} \begin{pmatrix} k_{i,t} \\ k_{i,x} \end{pmatrix},$$

it solves to $k_{i,t} = 1$ for all i. We insert the solution for $k_{i,t}$ into $k_{i,x}$ so that we arrive at a method for non-autonomous systems. Writing again k_i for $k_{i,x}$ we have

$$Mk_i = hf(t + \alpha_i h, v_i) + hf_x|_{(t_0,x_0)} \sum_{j=1}^{i} \gamma_{ij} k_j + \gamma_i h^2 f_t|_{(t_0,x_0)}, \tag{5.5}$$

$$\alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}, \qquad \gamma_i = \sum_{j=1}^{i} \gamma_{ij}.$$

Equally in the index 1 case, we have

$$\bar{M}\begin{pmatrix} k_i \\ 0 \end{pmatrix} = h\begin{pmatrix} f(t_i, v_i, w_i) \\ g(t_i, v_i, w_i) \end{pmatrix} + h\begin{bmatrix} f_x & f_z \\ g_x & g_z \end{bmatrix} \sum_{j=1}^{i} \gamma_{ij} \begin{pmatrix} k_j \\ l_j \end{pmatrix} + h^2 \gamma_i \begin{pmatrix} f_t \\ g_t \end{pmatrix}, \tag{5.6}$$

$$t_i = t_0 + \alpha_i h \tag{5.7}$$

in addition to (5.1) and (5.3).

Remark 5.1.1. *Observe that we explicitly need to evaluate the time-derivative of the non-autonomous function in* (5.5) *and also of the constraint in* (5.6)*. This is in addition to the spatial derivatives which are needed by most methods, thus some extra effort is necessary to obtain it.*

Structural savings by implementation

The computational efficiency of a method can be tuned by some model parameters, e.g., the number of stages or the choice of the α_{ij} coefficients (see Remark 4.2.1). But besides these optimization's we also have a look at what can be efficiently implemented for a general choice of Rosenbrock methods. What is the complexity of the separated solution steps?

In every step of the method we have to calculate the solution of

$$\left(M - h\gamma_{ii} f_x\right) k_i = hf(v_i) + hf_x \sum_{j=1}^{i-1} \gamma_{ij} k_j. \tag{5.8}$$

Thus the most expensive parts beside the general evaluation of f and f_x (which is out of our scope) are:

(a) Solution of the linear system $(M - h\gamma_{ii} f_x)$.

(b) Matrix multiplication between f_x and $\sum \gamma_{ij} k_j$.

For (a) we have to calculate one LU decomposition per time-step.

Remark 5.1.2. *The advantage of only one LU decomposition per step for full system matrices can be preserved in the spare matrix case by the use of sparse LU factorization as implemented in packages such as* `MUMPS` *[ADKL01, AGLP06] or* `UMFPACK` *[Dav06, Dav04]. By using* `UMFPACK` *in* `MATLAB` *we already saw savings in the second evaluation of the decomposition for iteration matrices obtained from problems like those described in Chapter 2 compared to a direct solution method. For the numerical results in this thesis we utilized this sparse LU decomposition by* `MATLAB` *and* `UMFPACK`*.*

For (b) the matrix multiplication can be avoided [KPB85] by the use of

$$u_i = \sum_{j=1}^{i-1} \gamma_{ij} k_j + \gamma_{ii} k_i$$

instead of k_i. We may recover k_i by using the matrix $\Gamma = (\gamma_{ij})_{ij}$

$$k_i = \frac{1}{\gamma_{ii}} u_i - \sum_{j=1}^{i-1} \underbrace{c_{ij} u_j}_{=\gamma_{ij} k_j}, \qquad C = \begin{bmatrix} \gamma_{11}^{-1} & & \\ & \ddots & \\ & & \gamma_{ss}^{-1} \end{bmatrix} - \Gamma^{-1}$$

inserting k_i into (5.8) and dividing by h we rewrite the method as

$$\left(\frac{1}{h\gamma_{ii}}M - f_x\right) u_i = f\left(x_0 + \sum_{j=1}^{i-1} a_{ij}u_j\right) + M\sum_{j=1}^{i-1}\frac{c_{ij}}{h}u_j, \tag{5.9}$$

$$x_1 = x_0 + \sum_{j=1}^{s} m_j u_j, \tag{5.10}$$

$$a_{ij} = \alpha_{ij}\Gamma, \qquad \left(m_1 \dots m_s\right) = \left(b_1 \dots b_s\right)\Gamma^{-1}.$$

Equivalently for the non-autonomous system the simplification using u_i can be added after explicitly solving for t, i.e., in place of (5.5) we have

$$\left(\frac{1}{h\gamma_{ii}}M - f_x\right) u_i = f\left(t_i, x_0 + \sum_{j=1}^{i-1} a_{ij}u_j\right) + \frac{1}{h}M\sum_{j=1}^{i-1} c_{ij}u_j + \gamma_i h f_t(t_0, x_0), \tag{5.11}$$

together with (5.7).

In the constrained case we do the same, but have to extend the vector u_i such that it contains also the l_i variables,

$$u_i = \sum_{j=1}^{i-1}\gamma_{ij}\begin{pmatrix}k_j\\ l_j\end{pmatrix} + \gamma_{ii}\begin{pmatrix}k_i\\ l_i\end{pmatrix}$$

so that in the constrained non-autonomous setting we end up with

$$\left(\frac{1}{h\gamma_{ii}}\bar{M} - f_x\right) u_i = f\left(t_i, x_0 + \sum_{j=1}^{i-1} a_{ij}u_j\right) + \frac{1}{h}\bar{M}\sum_{j=1}^{i-1} c_{ij}u_j + \gamma_i h f_t(t_0, x_0). \tag{5.12}$$

Optimization for systems of second order

Considering the second order system, we bring it to first order form. The special structure of the first order form allows to reduce the size of the iteration matrix. Let us consider the first-order system

$$\begin{aligned}\dot{q} &= v\\ M\dot{v} &= f(q, v).\end{aligned} \tag{5.13}$$

Be aware that we slightly changed the notation of the mass matrix M. As we have unit mass for the velocity components $\dot{q}$, M now acts only on the $\dot{v}$ part. Inserting (5.13) into the Rosenbrock method (5.10) with (5.11) yields

$$\begin{bmatrix}\frac{1}{h\gamma_{ii}}\mathrm{Id} & -\mathrm{Id}\\ -f_q & \frac{1}{h\gamma_{ii}}M - f_v\end{bmatrix}\begin{pmatrix}u_{i,q}\\ u_{i,v}\end{pmatrix} = \begin{pmatrix}a_{i,v}\\ f(a_{i,q}, a_{i,v})\end{pmatrix} + \frac{1}{h}\begin{bmatrix}\mathrm{Id} & 0\\ 0 & M\end{bmatrix}\sum_{j=1}^{i-1} c_{ij}\begin{pmatrix}u_{j,q}\\ u_{j,v}\end{pmatrix}, \tag{5.14}$$

$$a_{i,*} = *_0 + \sum_{j=1}^{i-1} a_{ij}u_{j,*}.$$

Writing down the first row in (5.14) explicitly

$$\frac{1}{h\gamma_{ii}}u_{i,q} - u_{i,v} = a_{i,v} + \frac{1}{h}\sum_{j=1}^{i-1} c_{ij}u_{j,q}, \tag{5.15}$$

solving for $u_{i,v}$, and inserting into the second row gives

$$\left(-f_q - \frac{1}{h\gamma_{ii}}f_v + \frac{1}{h^2\gamma_{ii}^2}M\right)u_{i,q} = f(a_{i,q}, a_{i,v}) + \frac{1}{h}M\sum_{j=1}^{i-1} c_{ij}u_{j,v} + \left[\tfrac{1}{h\gamma_{ii}}M - f_v\right]\left(a_{i,v} + \frac{1}{h}\sum_{j=1}^{i-1} c_{ij}u_{j,q}\right). \tag{5.16}$$

$u_{i,v}$ is again determined by back substitution of $u_{i,q}$ into (5.15). In this way the size of the iteration matrix is reduced by a factor of 2.

In the constrained case we consider

$$\begin{aligned} \dot{q} &= v \\ M\dot{v} &= f(q, v, z) \\ 0 &= g(q, v, z), \end{aligned} \tag{5.17}$$

using the optimized Rosenbrock method for constrained systems (5.12), we have to solve in every step

$$\begin{bmatrix} \gamma' \mathrm{Id} & -\mathrm{Id} & 0 \\ -f_q & \gamma' M - f_v & -f_z \\ -g_q & -g_v & -g_z \end{bmatrix}\begin{pmatrix} u_{i,q} \\ u_{i,v} \\ u_{i,z} \end{pmatrix} = \begin{pmatrix} a_{i,v} \\ f(a_{i,q}, a_{i,v}, a_{i,z}) \\ g(a_{i,q}, a_{i,v}, a_{i,z}) \end{pmatrix} + \frac{1}{h}\begin{bmatrix} \mathrm{Id} & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix}\sum_{j=1}^{i-1} c_{ij}\begin{pmatrix} u_{j,q} \\ u_{j,v} \\ u_{j,z} \end{pmatrix}, \tag{5.18}$$

where we used the abbreviation $\gamma' = \frac{1}{h\gamma}$. Inserting (5.15) into the second row of (5.18) gives

$$-g_q u_{i,q} - \gamma' g_v u_{i,q} - g_z u_{i,z} = g(a_{i,q}, a_{i,v}, a_{i,z}) + g_v\left(a_{i,v} + \frac{1}{h}\sum_{j=1}^{i-1} c_{ij}u_{i,q}\right)$$

and the second order representation

$$\begin{bmatrix} \gamma'^2 M - \gamma' f_v - f_q & -f_z \\ -g_q - \gamma' g_v & -g_z \end{bmatrix}\begin{pmatrix} u_{i,q} \\ u_{i,z} \end{pmatrix} = \begin{pmatrix} f(a_{i,q}, a_{i,v}, a_{i,z}) \\ g(a_{i,q}, a_{i,v}, a_{i,z}) \end{pmatrix} + \begin{bmatrix} \frac{1}{h}M & 0 \\ 0 & 0 \end{bmatrix}\sum_{j=1}^{i-1} c_{ij}\begin{pmatrix} u_{j,v} \\ u_{j,z} \end{pmatrix} + \begin{bmatrix} \gamma M - f_v & 0 \\ 0 & -g_v \end{bmatrix}\left(\begin{pmatrix} a_{i,v} \\ a_{i,v} \end{pmatrix} + \frac{1}{h}\sum_{j=1}^{i-1} c_{ij}\begin{pmatrix} u_{i,q} \\ u_{i,q} \end{pmatrix}\right). \tag{5.19}$$

Observe that the second row in the last term of (5.19) cancels in the case of $g_v = 0$. We finally get

$$\begin{bmatrix} \gamma'^2 M - \gamma' f_v - f_q & -f_z \\ -g_q & -g_z \end{bmatrix} \begin{pmatrix} u_{i,q} \\ u_{i,z} \end{pmatrix} = \begin{pmatrix} f(a_{i,q}, a_{i,v}, a_{i,z}) \\ g(a_{i,q}, a_{i,v}, a_{i,z}) \end{pmatrix} + \begin{pmatrix} \frac{1}{h} M \sum_{j=1}^{i-1} c_{ij} u_{j,v} \\ 0 \end{pmatrix} + \begin{pmatrix} (\gamma M - f_v)(a_{i,v} + \frac{1}{h} \sum_{j=1}^{i-1} c_{ij} u_{i,q}) \\ 0 \end{pmatrix}. \quad (5.20)$$

Also in the constrained case the size of the iteration matrix is reduced by the number of velocity components.

Remark 5.1.3. *To account for a bad scaling of the iteration matrix in* (5.19) *between the constraint block compared to the upper left block, we may add a preconditioning by a factor of γ'^2 following the idea of [BDT08]. This is done by pre- and post-multiplication of the iteration matrix S by*

$$D_L = \begin{bmatrix} \gamma'^2 & 0 \\ 0 & \mathrm{Id} \end{bmatrix}, \qquad D_R = \begin{bmatrix} \mathrm{Id} & 0 \\ 0 & \gamma'^{-2} \end{bmatrix},$$

i.e., we replace the problem of finding x s.t. $Sx = b$ by the preconditioned version

$$D_L S D_R \tilde{x} = D_L b$$
$$D_R \tilde{x} = x.$$

Instead of pre- and post-multiplication the factors can also be directly incorporated into the matrices of the system.

Error estimate and step-size control

For most Rosenbrock methods it is possible to construct an embedded method, i.e., a method of one order less while sharing all coefficients expect those of b_i, $i = 1 \dots s$ in (5.1). We denote these different coefficients by $\hat{b}_i$, and calculate another approximation to the solution by

$$\hat{x}_{n+1} = x_n + \sum_{i=1}^{s} \hat{b}_i k_i, \qquad \hat{z}_{n+1} = z_n + \sum_{i=1}^{s} \hat{b}_i l_i,$$

reusing the already calculated stages k_i, l_i from (5.2).

The embedded method is then used to estimate the local error at time-step x_n by $e_n = x_n - \hat{x}_n$. To evaluate the error into one scalar, we specify vectors of absolute and relative tolerances, a_{tol} and r_{tol}, for all individual components, so that we use the estimate of each component of $x \in \mathbb{R}^{n_x}$ and obtain the scalar

$$\tilde{e}_n = \left\| \left(\frac{e_{n,1}}{a_{tol,1} + r_{tol,1} \max(|x_{n-1,1}|, |x_{n,1}|)} \quad \cdots \quad \frac{e_{n,n_x}}{a_{tol,n_x} + r_{tol,n_x} \max(|x_{n-1,n_x}|, |x_{n,n_x}|)} \right)^T \right\|_2.$$

The value of $\tilde{e}_n$ is also used to determine if a step was successfully. For $\tilde{e}_n > 1$ the step is rejected and a reattempt is started using a smaller step-size h.

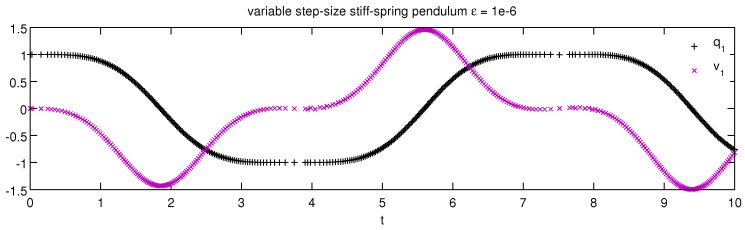


Figure 5.1: Result of ROS3P applied to the stiff spring pendulum using variable step-sizes (first component and velocity $v = \dot{q}$ shown)

For very stiff system close to index 3, as in the case of a singular singularly perturbed system, the error estimates suffer from the structure of the system and tend to be of one order less in the velocity components $v = \dot{q}$ and of two orders less in Lagrange multiplier components z. So that, the estimate is to high and the suggested step-size will become unnecessarily small. To circumvent this in the calculation of e_n, we consider quantities $\frac{v}{h}$ and $\frac{\lambda}{h^2}$, scaled by the step-size h. The effect and the scaling approach is also proposed in [HLR89a].

Because of its one step nature a step-size control can be easily employed for a Rosenbrock method. Considering two time-steps of a Rosenbrock method, the step-size of the second step may be chosen arbitrarily. We used the error estimation for the selection, based on the choice done in RODAS4 and proposed by [GLS88] we select the step-size for the next step h_{n+1} by

$$\text{fac} = \frac{h_{n-1}}{h_n}\sqrt[p]{\frac{\tilde{e}_n^2}{\tilde{e}_{n-1}}}, \qquad \overline{\text{fac}} = \min(6, \max(0.2, fac)), \qquad h_{n+1} = \frac{h_n}{\overline{\text{fac}}},$$

with p equal to the order of the used method.

Example 5.1.4. *We apply the* ROS3P *method using the described step-size selection scheme to the stiff spring pendulum with* $\epsilon = 10^{-6}$ *as seen in Example 3.1.8 (for* $g = 1.0$*). Setting the tolerances* $a_{tol} = r_{tol} = 1 \cdot 10^{-4}$ *for all components and use the described scaling of velocity and constraint components, we obtain the result for* $t_{end} = 10[s]$ *by* 744 *function calls and* 372 *evaluations of the Jacobi matrix* $\begin{bmatrix} f_q & f_z \\ g_q & g_z \end{bmatrix}$ *while 3 steps had to be rejected. The result is shown in Figure 5.1.*

5.2 Generalized alpha

For a comparison with classic integration methods we introduce a recent variant of Newmark's method. Newmark's method [New59] is very popular for the computation of structural dynamics [Wri08] since it is directly formulated in second order form and easy to implement. The variant presented here is an improved version with selectable numerical damping called generalized-α. It was introduced in [CH93] and is also valid for constrained mechanical systems of index 3 as shown in [AB07].

We begin by considering the unconstrained mechanical system

$$M\ddot{q} = f(q, \dot{q}, t).$$

Newmark's integration formula [New59] for generalized coordinates q_n, accelerations $a_n = \ddot{q}_n$ and time-step h is given by

$$\begin{aligned} q_{n+1} &= q_n + h\dot{q}_n + h^2\left(\frac{1}{2} - \beta\right)a_n + h^2\beta a_{n+1}, \\ \dot{q}_{n+1} &= \dot{q}_n + h(1-\gamma)a_n + h\gamma a_{n+1}, \end{aligned} \tag{5.21}$$

where β and γ are the methods parameters. Their optimal choice is

$$\gamma = \frac{1}{2} \qquad \text{and} \qquad \beta = \frac{1}{4},$$

which gives an A-stable method with $\rho_\infty = 1$ convergent of second-order. One can introduce some numerical damping such that ρ_∞ becomes less than 1 by adding $\alpha > 0$ and choosing

$$\gamma = \frac{1}{2} + \alpha, \qquad \beta = \frac{1}{4}\left(\gamma + \frac{1}{2}\right)^2,$$

but this leads also to Newmark methods which are only convergent of first-order.

To obtain the generalized-α method, we follow the way proposed in [AB07] and add a recurrence relation into the Newmark scheme, by redefining the vector a_n as acceleration like variables to

$$(1-\alpha_m)a_{n+1} + \alpha_m a_n = (1-\alpha_f)\ddot{q}_{n+1} + \alpha_f \ddot{q}_n \tag{5.22}$$

with initial value $a_0 = \ddot{q}_0$.

Remark 5.2.1. *The recurrence relation* (5.22) *enforces the equilibrium at every time-step by adding the vector of accelerations a_n which satisfy the property of the true accelerations at time t_{n+1}*

$$Ma_{n+1} = f(q_{n+1}, \dot{q}_{n+1}, t_{n+1}) = f_{n+1}.$$

Together with the idea of averaging the time instants

$$(1-\alpha_m)M\ddot{q}_{n+1} + \alpha_m M\ddot{q}_n = (1-\alpha_f)f_{n+1} + \alpha_f f_n,$$

we arrive at the generalized-α method.

Remark 5.2.2. *An additional advantage of* (5.22) *is that also the accelerations are computed with second order accuracy.*

By (5.22) we gain two additional parameters α_f, α_m, which can be chosen for suitable accuracy and stability properties. In the fixed step-size case the algorithm is convergent of second order provided that

$$\gamma = \frac{1}{2} + \alpha_f - \alpha_m.$$

Observe that for $\alpha_f = \alpha_m = 0$ we obtain Newmark's method (5.21). Optimal values for α_m, α_f, and β are proposed in [CH93]. It is possible to relate the parameters to the spectral radius $\rho_\infty = |R(\infty)|$ of the algorithm. For an A-stable

algorithm it has to be chosen between $\rho_\infty \in [0,1]$, where $\rho_\infty = 0$ means L-stability and $\rho_\infty = 1$ disables the numerical damping. The optimal choices for second-order accuracy are

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \qquad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1}, \qquad \beta = \frac{1}{2}\left(\gamma + \frac{1}{2}\right)^2.$$

With iteration matrix

$$S_t = M\beta' + C_t\gamma' + K_t,$$

Jacobi matrices

$$K_t = \frac{\partial(-f)}{\partial q}, \qquad C_t = \frac{\partial(-f)}{\partial \dot{q}},$$

and constants

$$\beta' = \frac{\partial \ddot{q}_{n+1}}{\partial q_{n+1}} = \frac{1-\alpha_m}{h^2\beta(1-\alpha_f)},$$
$$\gamma' = \frac{\partial \dot{q}_{n+1}}{\partial q_{n+1}} = \frac{\gamma}{h\beta},$$

the method can be implemented as in Algorithm 1.

Algorithm 1 one step of generalized-α

$q_{n+1} = q_n + h\dot{q}_n + h^2(.5-\beta)a$
$\dot{q}_{n+1} = \dot{q}_n + h(1-\gamma)a$
$a = 1/(1-\alpha_m)(\alpha_f \ddot{q}_n - \alpha_m a)$
$q_{n+1} = q_{n+1} + h^2\beta a$
$\dot{q}_{n+1} = \dot{q}_{n+1} + h\gamma a$
$\ddot{q}_{n+1} = 0$
for $i = 1$ to i_{max} **do**
  compute residuum r^q
  if $r^q <$ tol **then**
    break
  end if
  $\delta q = -S_t^{-1} r^q$
  $q_{n+1} = q_{n+1} + \delta q$
  $\dot{q}_{n+1} = \dot{q}_{n+1} + \gamma'\delta q$
  $\ddot{q}_{n+1} = \ddot{q}_{n+1} + \beta'\delta q$
end for
$a = a + (1-\alpha_f)/(1-\alpha_m)\ddot{q}_{n+1}$

Remark 5.2.3. *One interesting property of the method is that a two-step formulation can be obtained by doing two time-steps and eliminating the auxiliary variable a_n. If we further define $g = M^{-1}f$, the generalized-α method reads*

$$\sum_{k=0}^{2} a_k q_{n+k-1} + h\sum_{k=0}^{1} u_k \dot{q}_{n+k-1} = h^2 \sum_{k=0}^{2} b_k g_{n+k-1},$$
$$\sum_{k=0}^{2} a_k \dot{q}_{n+k-1} = h\sum_{k=0}^{2} c_k g_{n+k-1},$$

with

$$\begin{aligned}
a_0 &= -\alpha_m, & a_1 &= -1 + 2\alpha_m, & a_2 &= 1 - \alpha_m,\\
u_0 &= -\alpha_m, & u_1 &= -1 + \alpha_m, & &\\
b_0 &= \alpha_f(1/2 - \beta), & b_1 &= (1-\alpha_f)/(1/2-\beta) + \alpha_f\beta, & b_2 &= (1-\alpha_f)\beta,\\
c_0 &= \alpha_f(1-\gamma), & c_1 &= (1-\alpha_f)(1-\gamma) + \alpha_f\gamma, & c_2 &= (1-\alpha_f)\gamma.
\end{aligned}$$

Remark 5.2.4. *Attention has to be paid in the case of variable step-size, because of the hidden multi-step structure, the step-size can't be changed independently of the method's parameters. If one wants to preserve second order accuracy the choice of γ has to be adapted. Let $h = t_{n+1} - t_n$ and let s be the factor such that $t_n - t_{n-1} = \frac{h}{s}$. Then we need to set γ_{n+1} for the next iteration to obey*

$$\frac{1-\alpha_m-\gamma_{n+1}}{1-\alpha_m-\gamma_n} = s\frac{(1-\alpha_f)\gamma_{n+1} - (1-\alpha_m)/2}{\alpha_f(1-\gamma_n) - \alpha_m/2}$$

to keep second order accuracy, as shown by [AB08].

Constrained cases

In the index 1 and index 3 case ($\epsilon = 0$) we will solve system

$$\begin{aligned}
M\ddot{q} &= f(q,\dot{q},t) - G^T\lambda\\
\epsilon^2\lambda &= g(q,t)
\end{aligned}$$

with $G = g_q$ as the matrix of constrained gradients. Thus the iteration matrix changes to

$$S_t = \begin{bmatrix} M\beta' + C_t\gamma' + K_t & G^T \\ G & \epsilon^2\,\mathrm{Id} \end{bmatrix},$$

where the sub-matrices K_t and C_t are now given by

$$\begin{aligned}
K_t &= \frac{\partial(M\ddot{q} - f + G^T\lambda)}{\partial q},\\
C_t &= -\frac{\partial f}{\partial \dot{q}}.
\end{aligned}$$

Also in the algorithm we introduce the extra degrees of freedom λ as seen in Algorithm 2.

This method is convergent of second order in the coordinates $q, \dot{q}$, and also in Lagrange multipliers λ, as shown in [AB07], even for systems of index 3.

Remark 5.2.5 (Error estimate)**.** *Due to the structure of Newmark-like iterations, there cannot exist an embedded method. Therefore it is more difficult to obtain an error estimate. The most pragmatic ways of obtaining a measure for the error are either controlling the number of Newton iterations, or evaluating the residual forces at intermediate time-steps as done by* `HALFTOL` *inside the finite element software package* `ABAQUS` *[Das11].*

Algorithm 2 generalized-α including lambda

$q_{n+1} = q_n + h\dot{q}_n + h^2(.5-\beta)a$
$\dot{q}_{n+1} = \dot{q}_n + h(1-\gamma)a$
$a = 1/(1-\alpha_m)(\alpha_f \ddot{q}_n - \alpha_m a)$
$q_{n+1} = q_{n+1} + h^2\beta a$
$\dot{q}_{n+1} = \dot{q}_{n+1} + h\gamma a$
$\ddot{q}_{n+1} = 0$
$\lambda_{n+1} = 0$
for $i = 1$ to i_{max} **do**
  compute residuum r^q and r^λ
  if $\sqrt{\|r^q\|^2 + \|r^\lambda\|^2} <$ tol **then**
    break
  end if
  $\begin{bmatrix} \delta q \\ \delta\lambda \end{bmatrix} = -S_t^{-1} \begin{bmatrix} r^q \\ r^\lambda \end{bmatrix}$
  $q_{n+1} = q_{n+1} + \delta q$
  $\dot{q}_{n+1} = \dot{q}_{n+1} + \gamma'\delta q$
  $\ddot{q}_{n+1} = \ddot{q}_{n+1} + \beta'\delta q$
  $\lambda_{n+1} = \lambda_{n+1} + \delta\lambda$
end for
$a = a + (1-\alpha_f)/(1-\alpha_m)\ddot{q}_{n+1}$

5.3 Comparison of numerical results

For a fair comparison we implemented the generalized-α method and the Rosenbrock method `ROS3P` in `MATLAB` and apply them to a structural mechanical problem involving hyperelastic materials modeled and discretized by `COMSOL`. The considered problem consists of $3,552$ degrees of freedom and is similar to the one that will be described in Section 7.2, here we choose a one dimensional excitation by a displacement of the inner ring

$$u(t) = \left(\frac{1}{1+e^{(-10t+8)}} - \frac{1}{1+e^8} \right) 10\sin(2\pi \cdot 3t)[mm],$$

depicted in Figure 5.3. The simulated time horizon consists of two seconds $t_0 = 0$, $t_{end} = 2$.

For the generalized-α method we obtain the results doing two simulations with different bulk-modulus $\kappa_1 = 30[MPa]$ and $\kappa_2 = 300,000[MPa]$, which also means Poisson ratio $\nu_1 = 0.48$ and $\nu_2 = 0.499998$. Using a fixed step-size of $h = 3 \cdot 10^{-3}$ and calculate one Jacobi matrix per Newton step, we end up with the simulation result in Table 5.1. The simulation result at $t = 1.707$ is plotted in Figure 5.4. A detailed comparison of which step requires how many Newton-steps and thus function evaluations is found in Figure 5.2. We see for increasing κ the number of Newton iterations increases in those parts of the excitation with high absolute value.

For the considered Rosenbrock methods the number of Jacobians needed is equal to the number of time-steps. Further the number of function calls per time-step is fixed. Choosing `ROS3P` ends up in two function calls per step (see Table 4.5), and one additional function call to evaluate the time-derivative $(f_t, g_t)^T$ for

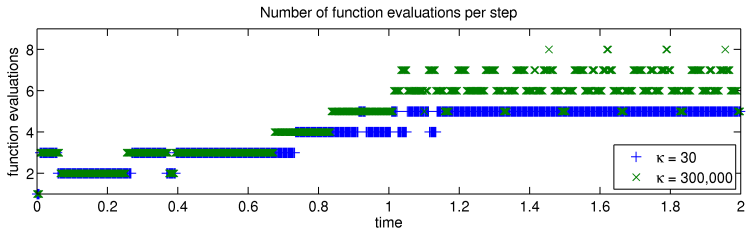


Figure 5.2: Influence of increasing bulk-modulus κ on the number of function evaluations using the generalized-α method

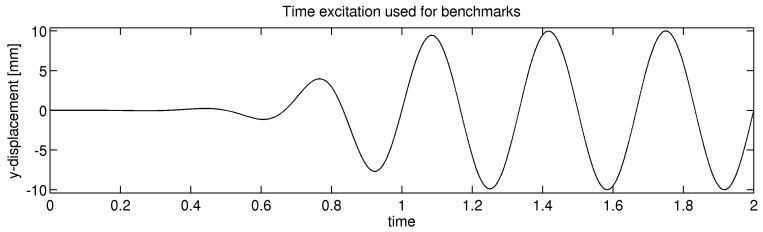


Figure 5.3: Used time-excitation for comparison

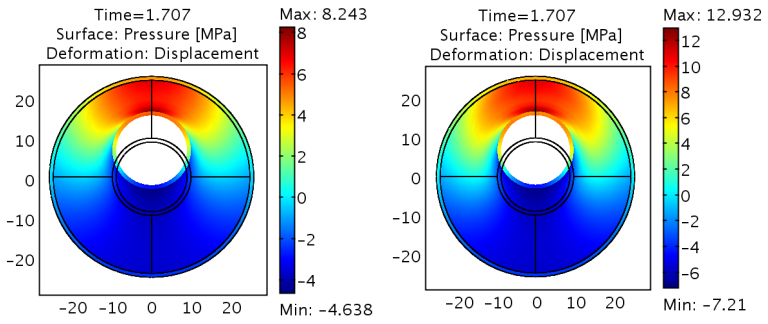


Figure 5.4: Comparison of pressure distribution and magnitude left $\kappa_1 = 30$, right $\kappa_2 = 300,000$

$\kappa[MPa]$	#Jacobi matrices	#Newton-steps	#function calls	CPU-time [s]
30	664	1,993	2,660	768
300,000	664	2,519	3,186	828

Table 5.1: Effort needed by generalized-α

$\kappa[MPa]$	#Jacobi matrices	#function calls	CPU-time [s]
30	667	2001	550
300,000	667	2001	550

Table 5.2: Simulation using optimized version ROS3P

component	absolute difference	relative difference
displacement	11.07	0.49%
velocity	320.80	0.52%
pressure	11.07	0.66%

Table 5.3: Relative and absolute differences between generalized α and ROS3P. Given in the 2-norm of the matrix of differences of all time-steps.

the non-autonomous system (5.6). This evaluation was done using a finite difference approximation of the derivative for $\tau \ll h$

$$f_t = \frac{f(t_0+\tau, x_0) - f(t_0, x_0)}{\tau}.$$

The CPU-time of the version of ROS3P optimized for second order systems is only $550[s]$ when κ_1 and κ_2 for the $2[s]$ excitation signal u (see also Table 5.2). Comparing the result of the Rosenbrock method to the one obtained by generalized-α in Table 5.3, we see a relative difference of 0.5%, but notice that ROS3P is a method convergent of order 3 whereas generalized-α is only of order 2.

Using the embedded error-estimator for a step-size selection we can further reduce the simulation costs while choosing $a_{tol} = r_{tol} = 1 \cdot 10^{-3}$ and a scaling the velocity components by h and pressure components by h^2. The selected step-sizes are depicted in Figure 5.5. The method needed 472 evaluations of the Jacobi matrix and made 1.416 function calls while 7 steps were rejected. All in all, we got the result after $440[s]$.

5.4 Conclusion

In the first part of this thesis we saw how incompressible materials can be included into a structural dynamical problem. Especially for the nonlinear case, using a

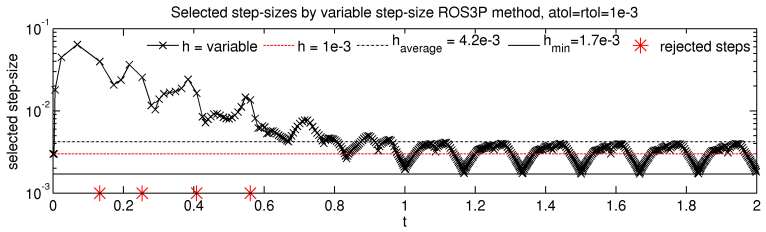


Figure 5.5: Selected step-size by a variable step-size scheme using ROS3P for a nearly incompressible, hyperelastic model in mixed formulation

hyperelastic material formulation, we showed that this is a singular singularly perturbed system.

We discussed some of the numerical problems while solving perturbed systems and motivated the use of Rosenbrock methods. These fit perfectly into our use-case, since we are more interested in fast simulation than in perfect accuracy in every time-step. We conducted our analysis using the index 1 form Rosenbrock methods and showed the advantages of those methods that additionally fulfill the Scholz conditions at a singular singularly perturbed system using a Prothero-Robinson-like test equation. Requiring only stiffly accuracy turned out to be insufficient.

After this, we gave an implementation of the Rosenbrock methods in a way such that they are competitive to methods usually used for structural mechanics. In the nearly incompressible singular singularly perturbed case the Rosenbrock methods were even able to outperform the generalized-α method. In general we experienced a good behavior while applying the methods to structural dynamical problems.

An appropriate starting point for a further analysis, is to considering Rosenbrock methods for the nonlinear case of singularly perturbed systems, as we observed some good results and also saw that the global convergences seems to be only weakly affected by the perturbation (at least in the case of a stiff spring pendulum).

Part II

Nonlinear model reduction

Chapter Six

Nonlinear model reduction technique

6.1 Introduction

In this part we aim at a further improvement of calculation time by using model reduction techniques. In contrary to Part I, where we have not altered the system equations, we now allow for changes to reduce the total number of equations which need to be solved. Our interest is to first give an overview of the existing methods, then we will focus on POD as the method of our choice for nonlinear systems. We show briefly how these methods work for structural dynamical problems and give the idea of some lookup methods which are necessary to decrease the computational costs.

There are different approaches to model reduction. One way, the so called black box modeling or system identification approach, is to construct a new parametric system of equations

$$\begin{aligned} \dot{x}_P &= f_P(x_P, u, p) \\ y_P &= g_p(x_P, u, p), \end{aligned}$$

and fit the system's parameters p to some system trajectories $\phi(x, u, t)$. The simplest form of this process is known as the description of a system only by its characteristic curves. More advanced examples can be found in [Lju87] and some recent ones used for mechanical systems in [BVB+11, SDR11]. Another systematic approach which combines different forms of system knowledge and measurements of the full system is called Grey-box modeling. A recent work about this topic is found at [Hau08].

In contrary to the methods above all the methods considered in this thesis are *projection based*, i.e., we always start with a full model description which already contains all effects that shall be considered. The challenge is to reduce the state dimension and computation time of the system simulation while retaining the system's dynamics. This we do by projecting the whole state space of the system onto a smaller subspace. Redundant state information can in this way be mainly compressed into a smaller uncorrelated representation.

Definition 6.1.1. *A linear map $T : V \to V$ is called projection iff $T \circ T = T$.*

Lemma 6.1.2. *Let T be a projection, let further* $\operatorname{Im} T = V$ *be spanned by*

$$V_k = \begin{bmatrix} v_1 & \cdots & v_k \end{bmatrix}$$

and $(\operatorname{Ker} T)^{\perp} = W$ *be spanned by*

$$W_k = \begin{bmatrix} w_1 & \cdots & w_k. \end{bmatrix}$$

Then we can represent T by

$$T = V_k W_k^T.$$

If V_k is orthogonal we can choose $W_k = V_k$.

Definition 6.1.3 (Galerkin Projection)**.** *Let Σ be a general system*

$$\Sigma : \dot{x} = f(x),$$

and $T = VV^T$ an orthogonal projection. Then we call $\hat{\Sigma}$,

$$\hat{\Sigma} : \dot{\tilde{x}} = Vf(V^T\tilde{x}),$$

the Galerkin projection *of* Σ*.*

There are different techniques for obtaining an appropriate projection basis for model reduction. For linear systems we will give a really short overview how the most popular methods work. For further insight we want to refer to [Ant05, BMS05]. Consider a linear system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}, \tag{6.1}$$

$A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{s\times n}$. One of the most common methods is called *Krylov moment matching*, which makes use of the so-called transfer function $H(s)$ of (6.1) defined as

$$\begin{aligned} H(s) = C(s\,\mathrm{Id} - A)^{-1}B &= \frac{1}{s}C\left(\sum_{l=0}^{\infty}\frac{1}{s^l}A^l\right)B \\ &= \sum_{l=1}^{\infty}\frac{1}{s^l}CA^{l-1}B. \end{aligned}$$

The reduced system is chosen such that its transfer function coincides in the first k terms $CA^{l-1}B$, $l = 1\dots k$ with the full system. The necessary projection is obtained by orthonormalizing the Krylov matrix $\mathscr{K}_n(A,B) = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$, which can be efficiently computed even for very big systems.

Another popular method for linear systems is *balanced truncation* [Moo81]. For an asymptotically stable linear system we define the reachability and observability Gramians as

$$P = \int_0^\infty e^{At}BB^T e^{A^T t}\,dt, \qquad Q = \int_0^\infty e^{A^T t}C^T C e^{At}\,dt. \tag{6.2}$$

This definition is such that $x_1^T P^{-1} x_1$ describes the minimal energy needed to steer a system from zero state to a given state x_1. Notice that P is non singular if the system is controllable, while $x_1^T Q x_1$ is the output energy produced by a state x_1 with input $u \equiv 0$, which will be finite for asymptotically stable systems. We can interpret a large value of $x_1^T P^{-1} x_1$ for some x_1 as a state which is difficult to attain. On the other side, if $x_1^T Q x_1$ has a small value, we are in a state which is only merely recognized in the systems output.

Theorem 6.1.4. *A linear asymptotically stable and controllable system* (6.1) *has always a balanced representation*

$$P = Q = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix}, \qquad \sigma_1 \geq \ldots \geq \sigma_n.$$

Where we call σ_i Hankel singular-values, they are the singular values of the matrix PQ.

This result can be nicely interpreted, small Hankel singular-values correspond to those states of the balanced system which are hard to observe and since of the inverse relation in P^{-1} also hard to attain. The model is reduced by considering only the first k states of the transformed system. One of the main advantages of this method is that we also get an error bound:

Theorem 6.1.5 (Error bound of balanced truncation)**.** *Let $H(s)$ be the transfer function of an asymptotically stable and controllable system, further let $\hat{H}_k(s)$ be the transfer function of the reduced system obtained by projecting onto the basis corresponding to the first k Hankel singular values. Then*

$$\|H(s) - \hat{H}_k(s)\|_{\mathscr{H}_\infty} = \max_{\omega \in \mathbb{R}} \|H(i\omega) - \hat{H}_k(i\omega)\|_2 \leq 2(\sigma_{k+1} + \cdots + \sigma_n).$$

Proof (Theorem 6.1.4, Theorem 6.1.5). [Ant05] □

Remark 6.1.6. *Balanced truncation is what can be achieved by the later described POD method applied to linear systems [Moo81].*

Since our main interest is in structural dynamics, we have a short look at what is done for second order linear systems like

$$\begin{aligned} M\ddot{q} &= Kq + D\dot{q} + Bu \\ y &= C_1 q + C_2 \dot{q} + Du. \end{aligned} \tag{6.3}$$

The most popular approach, by far, is modal reduction. For a modal reduction we consider (6.3) with $D = 0$, $u = 0$. Then we represent the system by a number of its generalized eigenmodes $V = \begin{bmatrix} v_i & \cdots & v_j \end{bmatrix}$, for which holds that

$$Kv_i = \lambda M v_i.$$

As there are as many eigenmodes as space dimensions, the decision which v_i shall contribute to the projection V is always difficult. Rather new works therefore suggest also an approach by balanced truncation for the second order system (6.3) [RS08, FE11, BS11, NKEB12]. As in the first order case, Krylov subspace methods can also be used as, for example, presented in [SL06].

Starting from the linear case there are extensions like methods for bilinear systems [BB11] [BD10] and discrete-time control systems [BBD10] appearing which use a balancing transformation. In general, for the nonlinear case it is increasingly difficult to apply the discussed methods. Nevertheless, the application of balanced truncation may still be possible as shown by [FS10, FS05], but the calculation of the corresponding Gramians (the nonlinear versions of (6.2)) is very difficult and currently only feasible for very small systems.

For us it seems that the only established and feasible method for general nonlinear systems remaining is the proper orthogonal decomposition. We are going to describe it in the following.

6.2 Singular value decomposition

As the main ingredient to proper orthogonal decomposition, we discuss the singular value decomposition (SVD). The SVD is very important tool in matrix analysis, it has its roots in works of Beltrami [Bel73] (see also [Ste93]) and was brought to attention of the numerical community by [GvL96]. We will repeat the theorem, give a short proof and also state some of its properties.

Theorem 6.2.1. *For all $A \in \mathbb{R}^{m\times n}$ there are orthogonal matrices $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ such that*

$$U^T A V = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{with} \qquad \Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_r \end{bmatrix},$$

where $r = \operatorname{rank}(A)$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$.

Proof. **Case 1:** $m = n$ and $\det(A) \neq 0$.

Then AA^T is symmetric and positive definite, thus there exists an orthogonal matrix U such that

$$AA^T = U\Sigma^2 U^T.$$

So we can write

$$A = U\Sigma \underbrace{(\Sigma U^T A^{-T})}_{V^T}.$$

From

$$\begin{aligned} VV^T &= A^{-1}U\Sigma\Sigma U^T A^{-T} \\ &= A^{-1}AA^T A^{-T} = \mathrm{Id}, \end{aligned}$$

we see that V indeed is orthogonal.

Case 2: For a general matrix A.

Let $Y = \begin{bmatrix} Y_1 & Y_2 \end{bmatrix} \in \mathbb{R}^{n\times n}$ and $W = \begin{bmatrix} W_1 & W_2 \end{bmatrix} \in \mathbb{R}^{m\times m}$ be orthogonal matrices such that

$$\operatorname{Im} Y_2 = \operatorname{Ker} A \qquad \text{and} \qquad \operatorname{Im} W_2 = \operatorname{Ker} A^T.$$

Thus

$$\begin{aligned} W^T A Y &= \begin{bmatrix} W_1^T \\ W_2^T \end{bmatrix} A \begin{bmatrix} Y_1 & Y_2 \end{bmatrix} \\ &= \begin{bmatrix} A_k & 0 \\ 0 & 0 \end{bmatrix} \overset{\text{(case 1)}}{=} \begin{bmatrix} U^T \Sigma V & 0 \\ 0 & 0 \end{bmatrix}, \end{aligned}$$

and hence

$$\begin{bmatrix} U & 0 \\ 0 & \mathrm{Id} \end{bmatrix} W^T A Y \begin{bmatrix} V^T & 0 \\ 0 & \mathrm{Id} \end{bmatrix} = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}.$$

□

Remark 6.2.2. *The singular values are the square roots of the non-zero eigenvalues of $AA^T > 0$ or A^TA*

$$\begin{aligned}\{\sigma_1^2 \dots \sigma_r^2\} &= \sigma(AA^T) \setminus \{0\} \\ &= \sigma(A^TA) \setminus \{0\}.\end{aligned}$$

Theorem 6.2.3 (Schmidt-Eckhart-Young-Mirsky)**.** *Let $A \in \mathbb{R}^{n\times m}$, it holds for the minimal* rank k *approximation of A in 2-norm that*

$$\min_{X \text{ s.t. } \operatorname{rank} X = k} \|A - X\|_2 = \sigma_{k+1}(A).$$

A minimizing solution $\hat{X}$ is obtained by the first k terms of the singular value decomposition of $A = U\Sigma V$, i.e.,

$$\hat{X} = \sigma_1 u_1 v_1^T + \dots + \sigma_k u_k v_k^T. \tag{6.4}$$

Lemma 6.2.4. *Given A of* rank r *for all X of* rank *less than k, it holds*

$$\|A - X\|_2 \geq \sigma_{k+1}(A).$$

Proof (Lemma). Let $y_i \in \mathbb{R}^m$, $i = 1 \dots m-k$ be a basis of $\ker X$, and $A = U\Sigma V^T = U\Sigma \begin{bmatrix} v_1 & \cdots & v_m \end{bmatrix}^T$. Then the intersection

$$\operatorname{span}\{y_1, \dots, y_{m-k}\} \cap \operatorname{span}\{v_1, \dots, v_{k+1}\}$$

is not empty. Let z be in this intersection and $z^Tz = 1$, then

$$\begin{aligned}\|A - X\|_2^2 &\geq \|(A - X)z\|_2^2 = \|Az\|_2^2 \\ &= \sum_{i=1}^{k+1} \sigma_i^2 (v_i^T z)^2 \geq \sigma_{k+1}^2.\end{aligned}$$

□

Proof. For the proof of the theorem it remains only to show that the lower bound is attained, this can easily be checked by inserting (6.4) □

Example 6.2.5. *To demonstrate the SVD at an arbitrary matrix, we interpret the image from Figure 6.2a as matrix $A = (a_{ij})_{ij}$ where a_{ij} is represented by a corresponding color value at position (i, j). As an example of the minimal rank property obtained in Theorem 6.2.3, we show the corresponding minimal rank approximations in Figure 6.2 of the full image.*

Figure 6.1 shows the singular values of A and the decreasing relative error

$$r = \frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=1}^{n} \sigma_i},$$

while the number of used basis vectors is increased. The image has a size of

$$2000 \times 3000 = 6 \cdot 10^6 \text{ pixels.}$$

For a minimal representation of rank r, only the data of

$$r \cdot (2000 + 3000)$$

elements is needed. The error and size of the different approximations is given in Table 6.1. However, please notice that the singular value decomposition is not a particular good method for image compression.

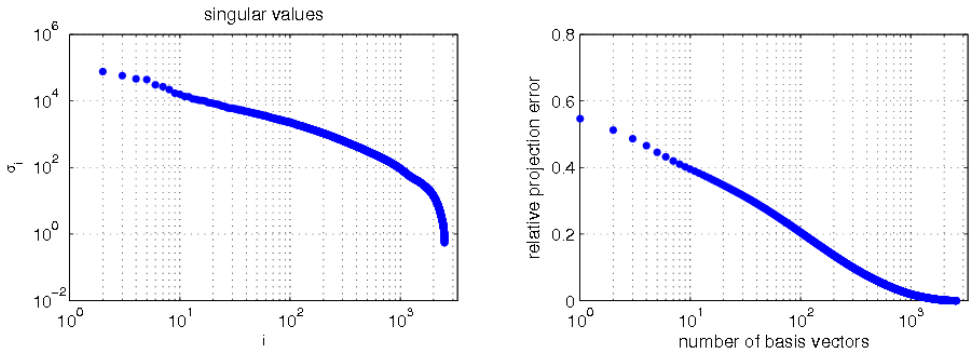


Figure 6.1: Singular values and relative error of projection

rank	reduced size	reduced to full size ratio	rel. error	σ_{k+1}
1	5,000	0.08%	.547	$7.600 \cdot 10^4$
10	50,000	0.8%	.395	$1.360 \cdot 10^4$
100	500,000	8%	.205	$2.223 \cdot 10^3$
1,000	5,000,000	80%	.019	$9.130 \cdot 10^1$

Table 6.1: Comparison of different approximations seen Figure 6.2

We can further say how sensitive the singular value decomposition is to perturbations [Ste90],

Theorem 6.2.6 (Weyl [Wey12]). *Let $\tilde{A} = A + E$ and denote the singular values of $\tilde{A}$ and A, respectively, by $\tilde{\sigma}_i$ and σ_i for $i = 1 \dots n$. Then it holds that*

$$|\tilde{\sigma}_i - \sigma_i| \leq \|E\|_2.$$

Theorem 6.2.7 (Mirsky [Mir60]). *Using the notation of Theorem 6.2.6, it holds that*

$$\sqrt{\sum_i (\tilde{\sigma}_i - \sigma_i)^2} \leq \|E\|_F,$$

for $\|E\|_F = \sqrt{\sum_{i,j} |e_{ij}|^2}$ being the Frobenius norm.

Remark 6.2.8. *There is no restriction to the size of $\|E\|$ in the above theorems.*

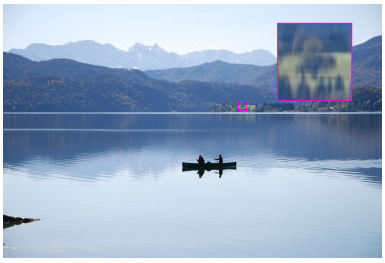
For singular vectors it is getting more complicated. Let

$$A = \begin{bmatrix} U_1 & U_2 & U_3 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^H \\ V_2^H \end{bmatrix}$$

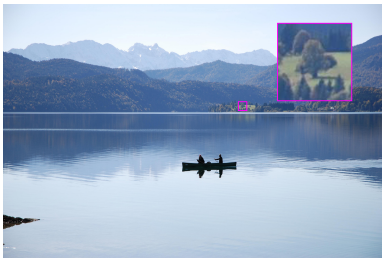
(a) original

(b) rank 1

(c) rank 10



(d) rank 100



(e) rank 1000

Figure 6.2: Low rank approximations by singular value decomposition

and

$$\tilde{A} = \begin{bmatrix} \tilde{U}_1 & \tilde{U}_2 & \tilde{U}_3 \end{bmatrix} \begin{bmatrix} \tilde{\Sigma}_1 & 0 \\ 0 & \tilde{\Sigma}_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_1^H \\ \tilde{V}_2^H \end{bmatrix}.$$

Theorem 6.2.9 (Wedin [Wed72]). *Suppose that δ, α, and β with $0 < \delta \leq \alpha \leq \beta$ are such that the eigenvalues of $\tilde{\Sigma}$ lie in $[\alpha, \beta]$ while the eigenvalues of Σ_2 are outside of $(\alpha - \delta, \beta + \delta)$. Then*

$$\|\tilde{U}_1^H \begin{bmatrix} U_2 & U_3 \end{bmatrix}\| \leq \sqrt{2} \frac{\max(\|R\|_2, \|S\|_2)}{\delta},$$
$$\|\tilde{V}_1^H V_2\| \leq \sqrt{2} \frac{\max(\|R\|_2, \|S\|_2)}{\delta},$$

where the size of perturbations to A is measured by the size of the residuals

$$R = A\tilde{V}1 - \tilde{U}_1\tilde{\Sigma}_1 \qquad \text{and} \qquad S = A^H\tilde{U}1 - \tilde{V}_1\tilde{\Sigma}_1.$$

Example 6.2.10. *Consider the perturbed matrices*

$$A_1 = \begin{bmatrix} 1+\epsilon & 0 \\ 0 & 1-\epsilon \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 1 & \epsilon \\ \epsilon & 1 \end{bmatrix},$$

the singular vectors of A_1 are

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

where those of A_2 are

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

We see that in the case of A_2 the whole subspace has turned by 45 degrees due to a perturbation of ϵ, while it was unaffected by the perturbation in the case of A_1.

6.3 POD

Let us explain the POD method following [Ant05]. Given a function $x : \mathbb{R} \to \mathbb{R}^n$ of time t, we denote a *time-snapshot* at time t_i by

$$x_i = x(t_i) \in \mathbb{R}^n. \tag{6.5}$$

We are looking for a set of orthonormal basis vectors $u_j \in \mathbb{R}^n$, $j = 1 \ldots N$ such that

$$x_i = \sum_{j=1}^{N} \gamma_{ji} u_j, \qquad i = 1 \ldots N.$$

For more time samples $t = t_1 \ldots t_N$ this is

$$\underbrace{\begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \gamma_{11} & \cdots & \gamma_{1N} \\ \vdots & \ddots & \vdots \\ \gamma_{N1} & \cdots & \gamma_{NN} \end{bmatrix}}_{\Gamma} \quad \text{with} \quad U^T U = \mathrm{Id}.$$

We truncate the elements to

$$\hat{x}_i = \sum_{j=1}^{k} \gamma_{ji} u_j, \qquad i = 1 \dots N$$

so that the snapshots are reconstructed only by the first k vectors u_j. Considering $\hat{X} = \begin{bmatrix} \hat{x}_1 & \cdots & \hat{x}_N \end{bmatrix}$, what is an optimal basis U such that the 2-induced norm of the difference $\|X - \hat{X}\|_2$ is minimized?

This problem is exactly the one solved by Schmidt-Eckard-Young-Mirsky, Theorem 6.2.3. Consider the SVD of $X = U\Sigma V^T$. Then $\Gamma = \Sigma V^T$ and so we have found an orthonormal basis U such that $X = U\Gamma$.

For a general dynamical system

$$\dot{x}(t) = f(x(t), u(t))$$
$$y(t) = g(x(t), u(t))$$

we can use a projection $T = VW^T$, where $W^T V = \mathrm{Id}_k$, to obtain a reduced order dynamical system of order k

$$\dot{\hat{x}}(t) = W^T f(V\hat{x}(t), u(t)) \tag{6.6}$$
$$y(t) = g(V\hat{x}(t), u(t)). \tag{6.7}$$

The trajectories of the reduced system $\hat{x} = W^T x$ evolve in a k-dimensional subspace. If $V = W$ the columns of V form an orthonormal set and T is orthogonal.

For a given system with given input trajectories $u(t)$ we build time-snapshots of a specific solution of the dynamical equations. Using the matrix of time-snapshots X, we obtain an optimal basis in the discussed way such that $X = U\Gamma$. For a dimension reduction of the system we may take only the first k elements of this basis so that $U_k = \begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix}$ is used for the projection with

$$V = W = U_k \in \mathbb{R}^{n \times k}. \tag{6.8}$$

$U_k \hat{x} = U_k U_k^T x$ is the projection of x onto $\mathrm{span}\{u_1, \dots, u_k\}$. The projection error is $\|\hat{x} - x\|_2 \geq \sigma_{k+1}$ as described by Lemma 6.2.4. Typically, k is chosen in a way such that the ratio of singular values

$$r = \frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=1}^{n} \sigma_i}$$

is of a given size (e.g., $r = 99.9\%$). r is used to describe the relative size of the projection error.

Definition 6.3.1. *For a dynamical system*

$$\dot{x} = f(x, u)$$

we denote by proper orthogonal decomposition *or* POD *the process*

1. *Generate a matrix of time-snapshots $t_i \in [t_0, t_{end}]$, $i = 1 \dots n$ for some input u,*

$$X = \begin{bmatrix} x(t_1) & \cdots & x(t_n) \end{bmatrix}. \tag{6.9}$$

2. *Calculate an orthogonal projection basis U_k for some chosen k*

$$X = U^T \Sigma V, \qquad U = \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix}, \qquad U_k = \begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix}. \tag{6.10}$$

3. *Construct the reduced system*

$$\dot{\hat{x}} = U_k^T f(U_k \hat{x}, u). \tag{6.11}$$

We will decorate the reduced system with a hat and denote the projection basis of dimension k by U_k, or U if the dimension is obvious from the context.

Remark 6.3.2. *The columns u_i of U_k are also called* POD-modes*.*

Remark 6.3.3. *The POD method is also known as* Karhuen-Loève transformation *or* principal component analysis*.*

Beside this general approach, POD is used in many fields and has different varieties, for example, in parametric systems [HDO11], fluid dynamics [KV03], parabolic systems [KV01], inverse design [BTDW04], dynamic analysis [LLL+02] or missing point evaluation [AWWB08].

We are going to apply the POD method to structural dynamical problems. For examples of our applications have a look at Chapter 7.

Connection of POD and balanced truncation

Let us consider the linear system (6.1) with the input matrix $B \in \mathbb{R}^{n \times m}$. By solving the initial value problem $x(t)$ for $x(0) \equiv 0$ while the input $u(t)$ is only a Dirac impulse $\delta(t)$ in every space dimension i,

$$u_i(t) = \delta(t) e_i, \qquad\qquad i = 1 \dots m,$$

and e_i the i-th unit vector, we get as solution

$$x(t) = \begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{pmatrix}, \qquad x_i(t) = e^{At} B_i, \qquad i = 1 \dots n.$$

In this way, the reachability Gramian has the following connection to the exact solution

$$P = \int_0^\infty e^{At} B B^T e^{At} dt = \int_0^\infty x(t) x(t)^T dt.$$

Transferring this observation to a discrete vector of time instances, we define the empirical reachability Gramian as follows.

Definition 6.3.4. *Let $x(t)$ be the solution of a differential equation, then we define for a matrix of time-snapshots $X = \begin{bmatrix} x(t_1) & \cdots & x(t_N) \end{bmatrix}$, $t_1 < t_2 < \ldots < t_N$, the* empirical reachability Gramian

$$P_e = \sum_{i=1}^{N} x(t_i) x(t_i)^T = X X^T, \qquad\qquad P_e \in \mathbb{R}^{N \times N}.$$

Now considering the POD method, we do a singular value decomposition of X (Theorem 6.2.1) and thus a decomposition by the eigenvalues of P_e is calculated. So the POD method can be interpreted as: Find a basis U such that for the projection of $\hat{P}_e = U^T P_e U$ holds that

$$\|P_e - \hat{P}_e\|_2$$

is minimized with $k = \operatorname{rank}\hat{P} < \operatorname{rank} P_e$. So by POD we estimate the empirical reachability Gramian in dependence of some given input u.

Remark 6.3.5. *In a statistical setup the empirical reachability Gramian can also be interpreted as the covariance matrix of the data as seen in the principal component analysis [Jol02].*

The parallels to balanced truncation ends because of the missing link to the observability Gramian. Nevertheless, we may also define the empirical observability Gramian in terms of a measured output y [LMG02]:

Definition 6.3.6. *Let $y(t) = C\,x(t)$, $C \in \mathbb{R}^{s\times n}$ be the measured output of a differential equation, then we define for a matrix of output time-snapshots*

$$Y = \begin{bmatrix} y(t_1) & \cdots & y(t_N) \end{bmatrix}$$

for $t_1 < t_2 < \ldots < t_N$ the empirical observability Gramian *as*

$$Q_e = Y^T Y, \qquad Q_e \in \mathbb{R}^{N\times N}.$$

But by the POD method the observability is not considered, thus we have no knowledge about the output energy of the neglected information. This is in contrary to balanced truncation, where the observability can be utilized to obtain an error-estimate. In the following we won't consider an output map to y for our systems.

Error propagation

For an orthogonal projection $T \in \mathbb{R}^{n\times n}$ such that $TT = T$ and $T^T = T$ we want to compare (following [RP03]) the systems

$$\begin{aligned} \dot{x} &= f(x,t), \\ \dot{\hat{x}} &= Tf(\hat{x},t), \end{aligned}$$

by introducing the error function

$$e(t) = \hat{x}(t) - x(t). \tag{6.12}$$

Further we split $e(t)$ in an orthogonal part $e_o(t)$ such that $Te_o(t) = 0$ and an in-plane part $e_i(t)$, for which it holds that $Te_i = e_i$. Thus e_o is the error orthogonal to the projection, while e_i is the accumulated error inside the subspace. Differentiation of (6.12) yields

$$\begin{aligned} \dot{e}(t) &= \dot{e}_i(t) + \dot{e}_o \\ &= Tf(\hat{x},t) - f(x,t). \end{aligned}$$

Multiplication by T on both sides using $T^2 = T$ and $T\dot{e} = \dot{e}_i$, we obtain an initial value problem for the in plane error

$$\begin{aligned}\dot{e}_i(t) &= T(f(e_i(t) + e_o(t) + x(t)) - f(x,t)),\\ e_i(0) &= 0.\end{aligned}$$

Locally we are looking at this in the linear case for an asymptotically stable system $\dot{x} = Ax$. We have

$$\dot{e}_i = TAe_i + TAe_o, \qquad e_i(0) = 0,$$

so as long as e_o is small also e_i will be small due to the stability of the system. While e_o can be controlled by the relative projection error.

In general side effects of the nonlinear system while applying a projection method are possible, and they may amplify and grow with the projection error. For a general estimate Gronwall's lemma may be used.

Remark 6.3.7. *Other works [HV08, KV03, Her08] show how the projection error can be amplified by the integration scheme and thus produce several "special" POD integration methods, e.g., the POD-Backward Euler or POD-Newmark schemes. We step back form describing a POD-generalized$-\alpha$ method or POD-Rosenbrock methods. In our view, POD is only an exchange of the discretized system and is thus independent of the time-integration scheme used, as long the reduced system is stable.*

Systems of second order

For the reduction of mechanical systems it is necessary to consider systems of second order like

$$\ddot{q} = f(q, \dot{q}) \qquad \text{for} \qquad q \in \mathbb{R}^n.$$

Because of the separated q and $\dot{q}$ variables we now have multiple possibilities of applying the POD method to the system.

We can construct a projection basis by snapshots of q such that the snapshot matrix is

$$X_q = \begin{bmatrix} q(t_1) & \cdots & q(t_N) \end{bmatrix}, \qquad X_q \in \mathbb{R}^{n\times N},$$

calculate the projection basis of U from X_q and apply the Galerkin projection like

$$\ddot{\hat{q}} = U^T f(U\hat{q}, U\dot{\hat{q}}). \tag{6.13}$$

But we could also use $x = \begin{pmatrix} q & v \end{pmatrix}^T$ and bring the system to its first order form

$$\dot{x} = \begin{pmatrix} v \\ f(q,v) \end{pmatrix}. \tag{6.14}$$

We can then apply the POD method to (6.14), use the snapshot matrix

$$X = \begin{bmatrix} \begin{pmatrix} q(t_1) \\ v(t_1) \end{pmatrix} & \cdots & \begin{pmatrix} q(t_N) \\ v(t_N) \end{pmatrix} \end{bmatrix}, \qquad X \in \mathbb{R}^{2n\times N} \tag{6.15}$$

and calculate the projection basis U_x from X. By separating

$$U_x = \begin{bmatrix} U_q \\ U_v \end{bmatrix}, \qquad U_q, U_v \in \mathbb{R}^{n\times k},$$

into a basis for q and v we end up at the projected system

$$\begin{aligned} \dot{\hat{x}} &= U_x^T \begin{pmatrix} U_q\hat{x} \\ f(U_q\hat{x}, U_v\hat{x}) \end{pmatrix} \\ &= U_q^T U_q\hat{x} + U_v^T f(U_q\hat{x}, U_v\hat{x}) \end{aligned}, \qquad x \approx U_x\hat{x}. \tag{6.16}$$

We see that in the reduced system (6.16) the previously separated position and velocity information is mixed up because of the projection, by this the invariant relation $\dot{q} = v$ can be destroyed.

Applying POD to the first order form has several consequences, the velocity components have got a larger absolute value and are thus preferred by the singular value decomposition of X. To circumvented this, we can calculate separate projection bases $U_q \in \mathbb{R}^{n\times k_q}$ and $U_v \in \mathbb{R}^{n\times k_v}$ by considering the snapshot matrices X_q and $X_v = \begin{bmatrix} v(t_1) & \cdots & v(t_n) \end{bmatrix}$. Afterwards, we put U_q and U_v as uncorrelated components together into the orthogonal projection basis

$$U_x = \begin{bmatrix} U_q & 0 \\ 0 & U_v \end{bmatrix}, \qquad U_x \in \mathbb{R}^{2n\times(k_q+k_v)}.$$

Remark 6.3.8. *Position and velocity is not uncorrelated, since by a finite difference approximation we have that*

$$v(t) = \frac{q(t+h) - q(t)}{h} + O(h^2)$$

and thus as linear combination already contained in the basis U_q. Neglecting a correlation where one exists, increases the size of the basis and thus the size of the projected system, but beside of this does no harm to the method as long the projection matrices stay orthogonal.

Numerical studies suggest that a direct handling of the second order system like in (6.13) gives better results then bringing the system to first order form. The corresponding simulations were discussed in the recent work of Joachim Krenciszek at the University of Kaiserslautern (to be published).

In our work we will consider the POD method always for the second order systems (6.13).

Definition 6.3.9. *By POD for a second order dynamical system*

$$\ddot{q} = f(q, \dot{q}, u)$$

we denote the process of

1. *Generate a matrix of time-snapshots $t_i \in [t_0, t_{end}]$, $i = 1 \ldots n$ for some input u,*

$$X = \begin{bmatrix} q(t_1) & \cdots & q(t_n) \end{bmatrix}. \tag{6.17}$$

2. *Calculate an orthogonal projection basis U_k for some chosen k*

$$X = U^T \Sigma V, \qquad U = \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix}, \qquad U_k = \begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix}. \tag{6.18}$$

3. *Construct the reduced system*

$$\ddot{\hat{q}} = U_k^T f(U_k \hat{q}, U_k \dot{\hat{q}}, u). \tag{6.19}$$

Remark 6.3.10. *Sticking to the second order system, we have the additional advantage that the structure of the system remains in second order form which allows us to use the optimized integration methods, discussed in Chapter 5 also for the reduced systems equations.*

Singularly perturbed systems

For the singular singularly perturbed systems considered in Chapter 3,

$$\begin{aligned} \ddot{q} &= f(q) - G^T(q)\lambda \\ \epsilon^2 \lambda &= g(q) \end{aligned}, \qquad \epsilon^2 \ll 1, \tag{6.20}$$

we first want to look at an expansion of the solution

$$q(t) = q_0(t) + \epsilon^2 q_1(t) + O(\epsilon^4),$$

as given by Theorem 3.1.10. In the snapshot matrix for $t_i \in [0, t_{end}], i = 1 \dots n$ we have

$$X = \begin{bmatrix} q(t_0) & \cdots & q(t_n) \end{bmatrix} = \underbrace{\begin{bmatrix} q_0(t_0) & \cdots & q_0(t_n) \end{bmatrix}}_{X_0} + \epsilon^2 \underbrace{\begin{bmatrix} q_1(t) & \cdots & q_1(t_n) \end{bmatrix}}_{X_1}$$

such that also the snapshot matrix can be decomposed into a part coming from the solution to the $\epsilon = 0$ system and a perturbation by ϵ^2. Unfortunately a perturbation by ϵ can have a tremendous effect on the calculated basis, as seen in Theorem 6.2.9 and Example 6.2.10. Thus it is not possible to conclude to the projection basis by considering the smooth system via X_0. This means, for snapshot generation the perturbed system has to be simulated. We have shown how this is effectively done in the case of a second order perturbed systems in Part I.

Our next concern is how to do a model reduction for system (6.20). The problem is separated into two sets of variables, namely those of q and those of λ. For the application of POD we have again several choices:

(a) The naive approach is to treat $z = \begin{pmatrix} q & \lambda \end{pmatrix}^T$ as the whole system state and use it for a snapshot matrix like

$$X = \begin{bmatrix} z(t_1) & \cdots & z(t_N) \end{bmatrix}.$$

The projection basis $U \in \mathbb{R}^{(n+n_\lambda \times k)}$ calculated from X splits similarly to the second order case into

$$U = \begin{bmatrix} U_k \\ L_k \end{bmatrix}, \qquad U_k \in \mathbb{R}^{n \times k}, \quad L_k \in \mathbb{R}^{n_\lambda \times k}.$$

Trying to calculate the Galerkin projection, we have to bring the $\epsilon^2\lambda$-term to the right hand side of the equation and obtain

$$\begin{aligned} U_k^T U_k \ddot{\hat{z}} &= U^T \begin{pmatrix} f(U_k\hat{z}) - G(U_k\hat{z})^T L_k \hat{z} \\ g(U_k\hat{z}) - \epsilon^2 L_k \hat{z} \end{pmatrix} \\ &= U_k^T f(U_k\hat{z}) + U_k^T G(U_k\hat{z})^T L_k\hat{z} + L_k^T g(U_k\hat{z}) - \epsilon^2 L_k^T L_k \hat{z} \end{aligned} \quad , \quad \text{for} \quad \hat{z} \in \mathbb{R}^k.$$

In general, all structure of the perturbed system is lost. Especially the constraint is mixed into the system, and the interpretation of λ as a Lagrange multiplier is gone. We are not able to distinguish between q and λ quantities, so a change in λ may affect also q, i.e.,

$$U_k U^T \begin{pmatrix} q \\ \lambda \end{pmatrix} \neq U_k U^T \begin{pmatrix} q \\ 0 \cdot \lambda \end{pmatrix}.$$

This all brings an error into the two times differentiated q components while the constraint is most likely to be unsatisfied. Also notice the non-identity mass matrix $U_k^T U_k$ appearing by the projection of

$$\begin{bmatrix} U_k^T & L_k^T \end{bmatrix} \begin{bmatrix} \mathrm{Id}_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_k \\ L_k \end{bmatrix} \tag{6.21}$$

on the left hand side of the equation.

We think these properties ***disqualify*** *the naive approach.*

(b) Respecting the structure of the equations we want λ to be retained such that we can assure that all constraints are acknowledged. We calculate the projection basis only through snapshots of q using

$$X_q = \begin{bmatrix} q(t_1) & \cdots & q(t_N) \end{bmatrix} \tag{6.22}$$

to determine $U_k \in \mathbb{R}^{n\times k}$ and consider the reduced system

$$\begin{aligned} \ddot{\hat{q}} &= U_k^T f(U_k\hat{q}) - G^T(U_k\hat{q})\lambda \\ \epsilon^2\lambda &= g(U_k\hat{q}). \end{aligned} \tag{6.23}$$

Observe that in this way all structure is retained but the dimension reduction suffers if many constraints $n_\lambda \gg 1$ have to be considered.

(c) We can use a separated basis for q and λ. Consider X_q and the snapshot matrices of λ,

$$X_\lambda = \begin{bmatrix} \lambda(t_1) & \cdots & \lambda(t_N) \end{bmatrix},$$

for a basis $U_k \in \mathbb{R}^{n\times k}$ of q and a basis $L_m \in \mathbb{R}^{n_\lambda \times m}$ of size m for λ. By a Galerkin projection we obtain

$$\begin{aligned} \ddot{\hat{q}} &= U_k^T f(U_k\hat{q}) - G^T(U_k\hat{q}) L_m \hat{\lambda} \\ \epsilon^2\hat{\lambda} &= L_m^T g(U_k\hat{q}). \end{aligned}$$

By this approach the projections for q and λ are obtained separately. One can interpret it as if we are doing a second Galerkin projection of system (6.23). The system is reduced to size $k+m$.

Remark 6.3.11. *Finally we could handle the constraint by avoiding it through a transfer into the index 0 formulation (Remark 3.1.9)*

$$\dot{q} = f(q) - \frac{1}{\epsilon^2} G^T(q) g(q).$$

which is then projected to

$$\dot{\hat{q}} = U_k^T f(U_k \hat{q}) - \frac{1}{\epsilon^2} U_k^T G^T(U_k \hat{q}) g(U_k \hat{q}),$$

using again the snapshots of q like in X_q. This approach is also equivalent to the index 0 formulation of (6.23)*. Nevertheless, in this thesis we are going to stick to the index 1 form.*

A numerical test of the described approach (b) and (c) will be done in Chapter 7 (see page 103).

6.4 POD in structural dynamics

The POD method in connection with structural dynamical systems was considered, for example, by [LKM03, KLM01, Wri08]. For the description of nonlinear structural dynamical systems we consider again the semi-discretized equations of Chapter 2, in a simplified notation, there we had

$$M\ddot{q} - f_a(q) = f_c(t), \qquad q \in \mathbb{R}^n. \tag{6.24}$$

We apply the POD method to the second order systems (Definition 6.3.9) by considering snapshots of q and a projection basis $U_k \in \mathbb{R}^{n \times k}$ of size k, we have the Galerkin projection of (6.24)

$$U_k^T M U_k \ddot{\hat{q}} - U_k^T f_a(U_k \hat{q}) = U_k^T f_c(t). \tag{6.25}$$

For the linearization of $f_a(q)$ around q_0, i.e., the stiffness matrix, evaluated at q_0, we have $K_{q_0} = \frac{\partial f_a}{\partial q}|q_0$, its projection is

$$\hat{K}_{q_0} = U_k^T K_{q_0} U_k, \qquad \hat{K}_{q_0} \in \mathbb{R}^{k \times k}.$$

Locally the reduced system is

$$U_k^T M U_q \ddot{\hat{q}} - U_k^T f_a(U_k \hat{q}_0) - \hat{K}_{U_k \hat{q}_0}(\hat{q} - \hat{q}_0) = U_k^T f_c(t),$$

simplified to the case $q_0 \equiv 0$, $f_a(q_0) \equiv 0$, we have

$$U_k^T M U_q \ddot{\hat{q}} - \hat{K}_{U_k \hat{q}_0}(\hat{q} - \hat{q}_0) = U_k^T f_c(t). \tag{6.26}$$

Boundary conditions

The boundary conditions of the reduced system are a crucial aspect. By boundary conditions we mean

- external forces $f_c(t)$,
- constraints which prescribe some state of q.

How are they considered in the reduced model, and how can they be retained accessible in the reduced model, as we may want to alter the external force interactively without doing a new projection?

Generally for boundary conditions, it holds that we have to assure that they are contained inside the projection subspace. For a force $f_c(t)$ and its projection $\hat{f}_c = U_k^T f_c(t)$. The force may be projected to zero if $f_c \perp U_k$ as the basis is calculated without considering f_c. We have a look at two scenarios:

1. A force is only applied to a few discrete points $\ll n$, e.g., a single node or the six degrees of freedom of a rigid body.

2. Many points are subjected to an external load.

For case 1 we add these points explicitly into the basis by an identity. Let U_k be the projection basis, and let without loss of generality the last node be subjected to a force. Then we will consider the projection

$$\begin{bmatrix} U_k[1\dots(n-1), 1\dots(k)] & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{n\times k+1},$$

with $U_k[1\dots(n-1), 1\dots(k)] \in \mathbb{R}^{n-1\times k}$ being the sub-matrix of U_k consisting of the rows $1\dots(n-1)$ and the columns $1\dots k$. In this way the selected components stay accessible in the reduced system and the external force can easily be altered.

For case 2, the overhead of adding all degrees of freedom separately into the basis might be to high. So an alternative approach is to take also snapshots of the force vectors

$$X_f = \begin{bmatrix} f_c(t_1) & \cdots & f_c(t_N) \end{bmatrix} \in \mathbb{R}^{n\times N}$$

and calculate a projection basis U_m^f of size m based on a singular value decomposition of X_f. To obtain an orthogonal projection basis, we build a combined matrix

$$\check{U} = \begin{bmatrix} U_k & U_m^f \end{bmatrix} \in \mathbb{R}^{n\times(k+m)}$$

with POD basis U_k and orthogonalize the result by another singular value decomposition

$$\check{U} = \begin{bmatrix} U_i & U_r \end{bmatrix}^T \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V, \qquad \Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_i \end{bmatrix},$$

and get the projection basis U_i, $i \leq (k+m)$. Notice that $i \leq (k+m)$ because both bases may span the same subspace. The basis U_i contains all basis vectors coming from the snapshots of q and all basis vectors out of the forces f_c.

Remark 6.4.1. *Although, because of the singular value decomposition, the basis is still ordered by some singular values, these can not anymore be interpreted in the sense of Theorem 6.2.3. Nevertheless both selected bases U_k and U_m^f can be interpreted correspondingly, and it is possible to project the calculated solution back into these. Let $\hat{q}(t) : \mathbb{R} \to \mathbb{R}^i$ be the solution of the reduced system using the Galerkin projection U_i, then we can represent the solution in coordinates of U_k by considering $U_k^T U_i \hat{q}(t)$ and in the coordinates of U_m^f by $(U_m^f)^T U_i \hat{q}(t)$.*

By this approach the force can only be altered as long as it stays inside the projected subspace.

Let us come to prescribed displacements and first consider the snapshot matrix X of q and a corresponding projection basis U_k. A node q_i which is fixed in its position $q_i(t) = 0$ for all t gives rise to a zero row in U_k, thus the constraint will be automatically satisfied by the projection.

Remark 6.4.2. *On the other hand, consider a subset of nodes which are constrained to follow the movement of a rigid body, because of a combination of displacements and rotations it may not be possible to describe the deformations by a linear combination of basis vectors, but since a rigid body consists only of six degrees of freedom we already know the relation between the nodes and the rigid body. In this case it may be advantageous to remove the constrained nodes from the basis generation and recover their position by the six degrees of freedom of the rigid body. In this way we saved degrees of freedom for the reduced model.*

In general a prescribed displacement should also be handled as a constraint, we have

$$\begin{aligned} M\ddot{q} &= f_a(q) + f_c(t) - G^T\lambda \\ 0 &= g(q,t) \end{aligned} \qquad \text{with} \qquad G = g_q$$

such that the same things said about f_c are also true for the force coming through $G^T\lambda$, this should be considered in a similar way inside the basis U_k to obtain the Galerkin projected system

$$\begin{aligned} 0 &= \hat{M}\ddot{q} + U_k^T f_a(U_k\hat{q}, U_k\dot{\hat{q}}) + U_k^T f_c(t) - U_k^T G^T\lambda \\ 0 &= g(U_k^T\hat{q}, t). \end{aligned}$$

How much can be saved by a projected system

To quantify how much the complexity is reduced by POD we compare the costs while solving the reduced system (6.25) and the full system (6.24). This depends surely on the integration method used to solve either one. For a rough estimate we try to compare the atomic operations of time-integration which are solutions of linear systems for implicit methods and matrix multiplications in explicit methods. So we want to compare one implicit time-step and one explicit time-step.

An implicit time-step consists mainly of a Newton step (compare Chapter 5.2)

$$\begin{aligned} S_t\Delta q &= f_a(q) \\ S_t &= \left[M\beta' + K\right]. \end{aligned}$$

In the *full system* case the stiffness matrix K is evaluated element-wise which scales with the per element degrees of freedom and is then assembled to obtain the complete stiffness matrix K_{q_0} evaluated at q_0, which scales with the total number of elements. We obtain $O(n)$ for the complete assembly step. The estimate of f_a and f_c is also obtained in $O(n)$. The linear solve step is hard to estimate, it depends roughly on the number of non-zero elements of the iteration matrix S_t.

For the *reduced system* we need also the full assembled stiffness matrix evaluated at q_0, which is the back-projection of the current state $\hat{q}_0$. Additionally we have to do the projection by post- and pre-multiplication of U to obtain the reduced matrix of size $k \times k$. The terms f_a and f_c need also to be projected $\hat{f}_a = U^T f_a$ and $\hat{f}_c = U^T f_c$, these projections imply additional costs of

step	full system	step	reduced system
		back-projection to obtain q_0	$O(nk)$
assembly of K_{q_0}	$O(n)$	$\hat{K}_{\hat{q}_0}$	$O(n)+O(nk)+O(nk^2)$
f_a, f_c	$O(n)$	$\hat{f}_a$, $\hat{f}_c$	$O(n)+O(nk)$
sparse linear solve	$O(nnz(K))$	linear solve	$O(k^3)$

Table 6.2: comparison of implicit Newton step

step	full system	step	reduced system
		back-projection to obtain q_0	$O(nk)$
f_a, f_c	$O(n)$	$\hat{f}_a, \hat{f}_c$	$O(n)+O(nk)$

Table 6.3: comparison explicit methods

- three matrix vector multiplications $O(kn)$ to obtain $\hat{f}_a$, $\hat{f}_c$, and q_0,
- one sparse matrix-matrix multiplication $K_{q_0}U$, $O(nk)$,
- one matrix-matrix multiplication $U^T[K_{q_0}U]$, $O(nk^2)$.

In the linear solve we have got the dense (not sparse), but smaller, matrix $\hat{K}_{\hat{q}_0}$ which takes $O(k^3)$ for a single solution. Albeit reduced, the smaller system still needs a strong connection to the full system, in every time-step we have to back-project and evaluate the equations in the full space. The only advantage comes in the solution of the linear system, but there is also some overhead from the projections, this means that we only gain an advantage if $k \ll n$ and maybe an additional advantage if the smaller system can be integrated using a larger time-step.

For an explicit time-step we consider the central difference scheme, as given by

$$\left(M + \frac{h}{2}D\right) q_{n+1} = h^2(f_c - f_a(q_n)) + \frac{h}{2}Dq_{n+1} + M(2q_n - 2q_{n-1}). \tag{6.27}$$

The comparison in Table 6.3 reveals even poorer performance improvements. Since we do not solve a linear system, the reduced size manifests only in smaller matrix-matrix multiplications. A factorization of M and D may be reused in every step for constant matrices.

We find that the reduction of a nonlinear structural mechanical problem only through a Galerkin projection has the drawback that the computation time is reduced only by a small factor. A similar result was previously found in [LKM03].

6.5 Lookup methods

We saw that the computation has its bottleneck while back-projecting into the full system for evaluation of $\hat{f}_c$, $\hat{f}_a$, and $\hat{K}_{\hat{q}}$ while the costs for solving the equations is,

in comparison, eliminated in the reduced system. Following the idea of [Her08], we want to avoid back-projections into the full state-space by pre-computing those values in an offline step such that we can reuse the already once evaluated function calls in an optimal way.

The used lookup methods are nothing more than an interpolation method in a high dimensional setting. A general interpolation method for a given function

$$\begin{aligned} f : \mathbb{R}^n &\to \mathbb{R}^n \\ q &\mapsto f(q) \end{aligned}$$

using the well known Taylor rule in the point $q = q_i + \delta q$ is

$$f(q) = f(q_i) + (q - q_i)\frac{\partial f}{\partial q}|_{q_i} + \frac{1}{2}(q - q_i)^T \frac{\partial^2 f}{\partial^2 q}|_{q_i}(q - q_i) + O(\delta q^3).$$

This means, knowing the function value at some points q_i and its first and second derivative allows to estimate points q close to q_i with an error of the order of $\|q - q_i\|^3$, and knowing only its first derivative gives an estimate of order $\|q - q_i\|^2$. These are the defining properties of Lookup 1 and Lookup 2.

Definition 6.5.1 (Lookup 1)**.** *Given a function $f : \mathbb{R}^n \to \mathbb{R}^m$, a set of evaluation points $q_i \in \mathbb{R}^n, i = 1 \dots l$, corresponding function values $f(q_i) \in \mathbb{R}^m, i = 1 \dots l$ and the derivatives $K_{q_i} = \frac{\partial f}{\partial q}|_{q_i}, i = 1 \dots l$, then* Lookup 1 *is the estimate $\tilde{f}$ to f constructed via*

$$\begin{aligned} \tilde{f}(q) &= f(q_s) + K_{q_s}(q - q_s), \\ s &= \text{argmin}_{i=1\dots l} \|q_i - q\|_2, \end{aligned}$$

its linearization is

$$\frac{\partial \tilde{f}}{\partial q} = \tilde{K}|_q = K_{q_s}.$$

Definition 6.5.2 (Lookup 2)**.** *Using the same notation as in Lookup 1 plus the second derivative information $H|_{q_i}$ in all known points q_i gives the estimate*

$$\begin{aligned} \tilde{f}(q) &= f(q_s) + K_{q_s}(q - q_s) + \frac{1}{2}(q - q_s)^T H|_{q_s}(q - q_s), \\ s &= \text{argmin}_{i=1\dots l} \|q_i - q\|_2 \end{aligned}$$

called Lookup 2*. For its linearization holds*

$$\frac{\partial \tilde{f}}{\partial q} = \tilde{K}|_q = K_{q_s} + (q - q_s)^T H|_{q_s}.$$

Remark 6.5.3. *Higher derivative information is often not easy to obtain. The $H|_{q_i}$ information in direction of some point q_{i+1} may be approximated through a finite difference using $K|_{q_i}$ and $K|_{q_{i+1}}$*

$$K|_{q_{i+1}} \approx K|_{q_i} + (q_{i+1} - q_i)^T H|_{q_i}.$$

An alternative approach, instead of using higher derivative information, is to utilize more of the known points q_i in the estimation of $f(q)$.

Definition 6.5.4 (TPWL [Rew03])**.** *We borrow the notation of Definition 6.5.1. An estimation to f via the s-point* trajectory piecewise-linear *approximation* (TPWL) *is*

$$\tilde{f}(q) = \sum_{i=1}^{s} \omega_i(q)(f(q_i) + K|_{q_i}(q - q_i)) \tag{6.28}$$

with linearization

$$\tilde{K}|q = \sum_{i=1}^{s} \dot{\omega}_i(q) K|_{q_i}.$$

The weights $\omega_i(q)$ are normalized such that $\sum_{i=0}^{s-1} \omega_i(q) = 1$ and the representation of the q_i is sorted such that it fulfills

$$\|q_0 - q\|_2 \leq \|q_1 - q\|_2 \leq \ldots \leq \|q_l - q\|_2. \tag{6.29}$$

Remark 6.5.5. *If higher derivative information is available it can also be included into TPWL, as for Lookup 2.*

Remark 6.5.6. *Weights ω_i, for example, can be computed dependent on the distance using an exponential kernel. For the current evaluation point q let d_i be the distance $d_i = \|q - q_i\|_2$ while the q_i are ordered like 6.29. Then we can compute*

$$\tilde{\omega}_i = e^{\left(-\beta \frac{d_i}{d_1}\right)}, \qquad \beta \in \mathbb{R},$$

and normalize afterwards to obtain

$$w_i = \frac{\tilde{w}_i}{\sum_{i=1}^{s} \tilde{w}_i}.$$

The error $\|f(q) - \tilde{f}(q)\|_2$ of both techniques, for a general function f, is only small in a region where the distance between evaluation point q and lookup point q_i is small. The choice of q_i, or the relation between those points at which f has to be evaluated and the q_i, is crucial for the success of a lookup method.

Example 6.5.7. *For the sake of easy visualization we choose small dimensions. The lookup methods are intended for higher dimensions where polynomial or spline interpolation is more complicated than in two dimensions.*

We are going to show the described lookup methods applied to

$$\begin{aligned} f : \mathbb{R}^2 &\to \mathbb{R} \\ \begin{pmatrix} x \\ y \end{pmatrix} &\mapsto \sin(x)cos(y). \end{aligned}$$

with 35 given interpolation points $\begin{pmatrix} x_i \\ y_i \end{pmatrix}$ given at a straight line starting from $y_i = 10$ and equally space up to $x_{34} = 9.9$,

$$x_i = 0 + (i-1) \cdot 0.3 \qquad \text{for} \qquad i = 1 \ldots 34$$

plus one additional point at $\begin{pmatrix} 18 \\ 6 \end{pmatrix}$. These are points drawn from a hypothetical trajectory. We use the lookup methods to evaluate all missing points in a 20×20

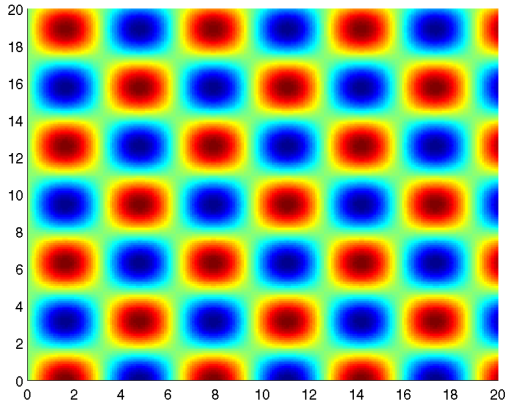


Figure 6.3: Ground truth of Example 6.5.7

square. In the result of Lookup 1 (Figure 6.4), as expected, we see that the method could only recover the local behavior of the function, while Lookup 2 (Figure 6.5) is able to recover a little bit more. The result of both methods suffers from being not differentiable. This is due to the norm we use to determine the closest point which is not differentiable.

For TPWL we can circumvent the problem of differentiability by using all points for the interpolation as seen in Figure 6.6. Using more than one point, for example, five (Figure 6.7), helps already to obtain a locally smoother result than Lookup 1 and Lookup 2 could generate.

Choosing the weights for TPWL as described in 6.5.6, a smaller value β leads to slower decay of $e^{-\beta d}$, which leads to a wider area of influence, whereas a bigger value of β leads to a locally smaller influence.

In combination with the reduced basis representation, the lookup methods may be directly used for the reduced system. This is beneficial through the much smaller state-space which needs to be approximated.

In general, the lookup method can be employed for reduced or full system. One can easily see that the application of Lookup 1 and the Galerkin projection commute:

Lemma 6.5.8. *Let*

$$\hat{\tilde{f}} = U_k^T f(U_k q_s) + U_k^T K_{q_s} U_k (q - q_s), \tag{6.30}$$

$$s = \text{argmin}_{i=1\dots l} \|U_k q_i - U_k q\|_2, \tag{6.31}$$

be the Galerkin projection using U_k applied to Lookup 1 and

$$\tilde{\hat{f}} = \hat{f}(\hat{q}_s) + \hat{K}_{q_s}(U_k q - \hat{q}_s), \tag{6.32}$$

$$s = \text{argmin}_{i=1\dots l} \|\hat{q}_i - U_k q\|_2, \tag{6.33}$$

be Lookup 1 applied after the Galerkin projection. Then it holds that

$$\tilde{\hat{f}} = \hat{\tilde{f}}.$$

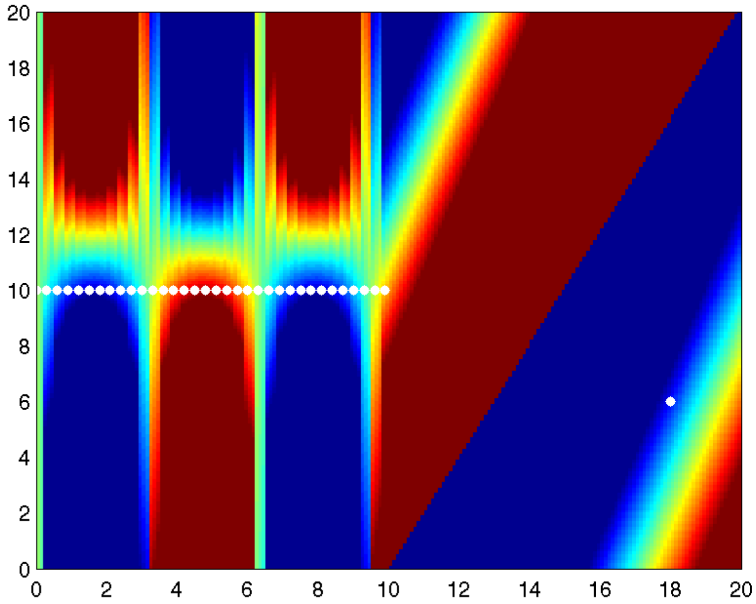


Figure 6.4: Result obtained by Lookup 1

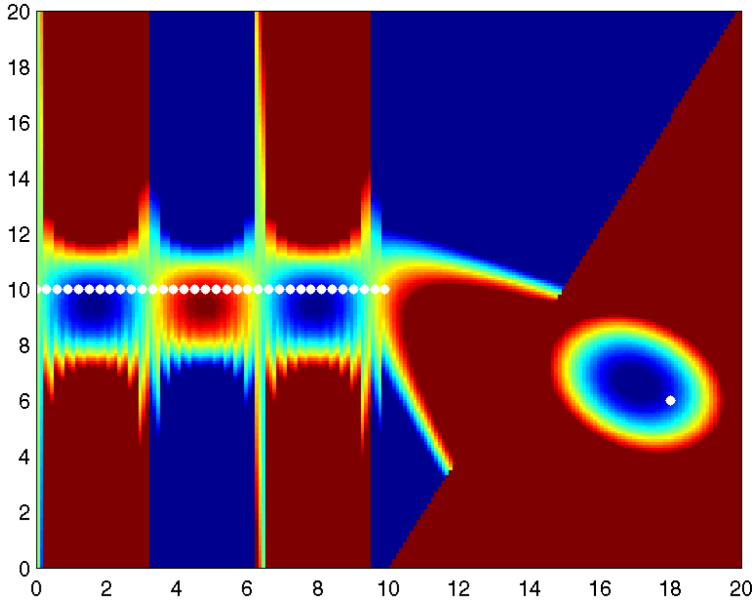


Figure 6.5: Result obtained by Lookup 2

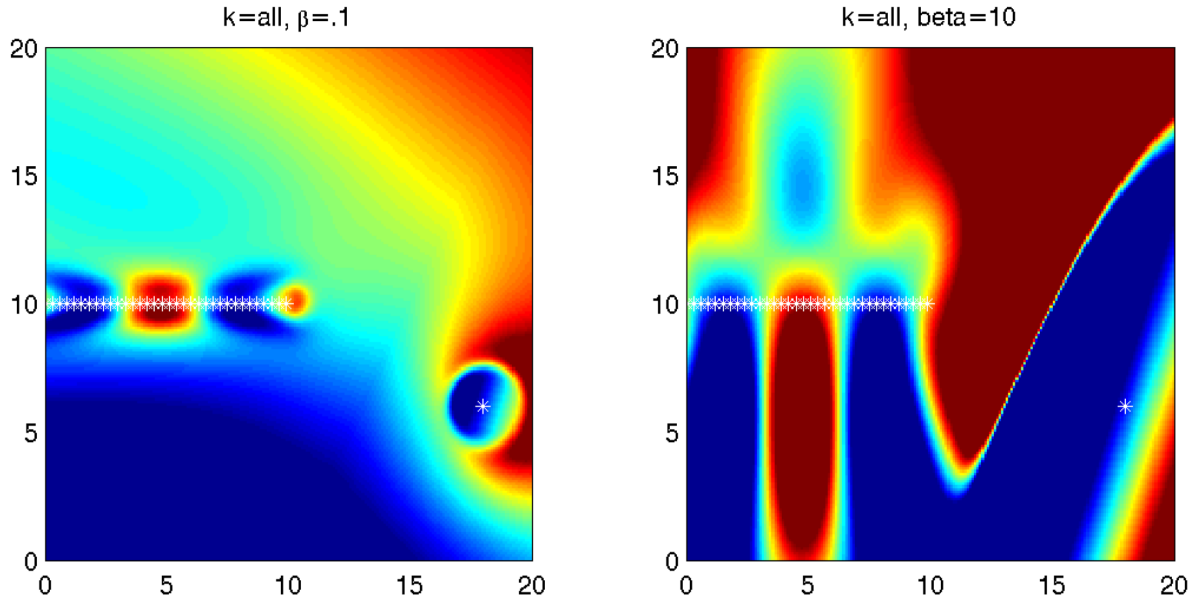


Figure 6.6: Result of the TPWL method using all available points

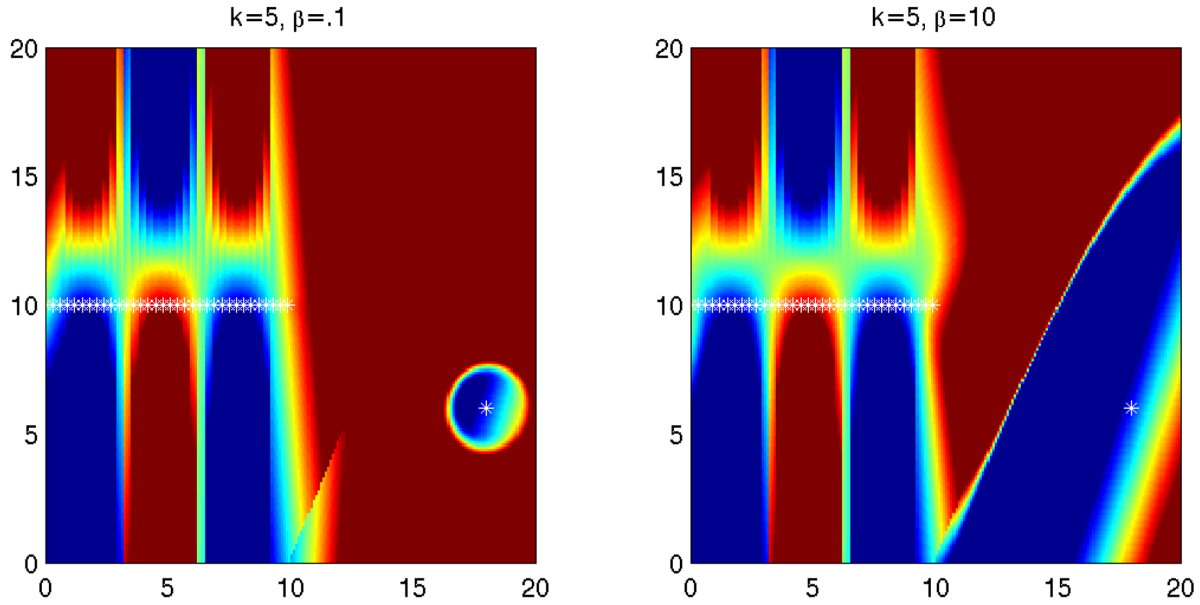


Figure 6.7: Result of the 5 point TPWL method

Remark 6.5.9. *It can be checked that the lemma holds also for Lookup 2 and TPWL.*

Although $\tilde{\hat{f}} = \hat{\tilde{f}}$ holds, their computational and memory costs differ. $\tilde{\hat{f}}$ has only the memory consumption of $k \cdot l$ for l times $\hat{f}(\hat{q}_s)$ plus $k \cdot l$ for l times $\hat{q}_s$ plus $l \cdot k^2$ for $\hat{K}_{q_s}$. All projections can be done in a "data collecting phase" of the method, even $U_k q$ needs not be computed, as it already is available as state of the reduced system.

The computational complexity using either lookup method, splits up into an offline and an online part. The offline computation consists for l points of calculating $f(q_i)$, $i = 1 \dots l$, for some given q_i. This is still the same amount of work as seen in the last section, but during the simulation of the reduced model, only the online part accounts to the run-time of the method, which is much less. For Lookup 1 and Lookup 2 we have to do l comparisons $\|q - q_l\|$ to find the closest point denoted by q_s. After this only a matrix vector multiplication by K remains, which is for the projected matrices of order $O(k^3)$. We saved all projections and the evaluations in the full space of dimension $n \gg k$. Beside the consistent dimension reduction the method allows a reuse of previous function evaluations.

Remark 6.5.10. *The presented lookup methods are kind of a brute force approach. In an application it may be advisable not to tackle the full system equations by one lookup method, but rather split the problem into independent parts and apply a lookup method to each of these, or even combine the lookup method with some analytic knowledge of the function to be approximated.*

DEIM

In difference to a lookup method, another recently discussed approach to efficiently overcome the difficulty of back-projecting into the full-state space is the *discrete empirical interpolation method* (DEIM) [CS10]. To approximate a nonlinear function $f : \mathbb{R}^n \to \mathbb{R}^n$ the idea is to project this function, as for POD, onto a space that is spanned by a much smaller basis $m \ll n$, and captures the nonlinearity. Let

$$\left\{v_1, \dots, v_m\right\} \subset \mathbb{R}^n$$

be such a basis. Then

$$f(t) \approx V c(t),$$

where $V = \begin{bmatrix} v_1 & \cdots & v_m \end{bmatrix} \in \mathbb{R}^{n \times m}$ and $c(t)$ is the corresponding coefficient vector.

To determine $c(t)$ we select m rows of the system $f(t) = Vc(t)$. Consider

$$P = \begin{bmatrix} e_{p_1} & \cdots & e_{p_m} \end{bmatrix},$$

where e_{p_n} is the p_n-th unit vector. If $P^T V$ is non singular we determine $c(t)$ by

$$P^T f(t) = P^T V c(t). \tag{6.34}$$

So that f is only evaluated at the indices p_i. The interpolation of $f(t)$ is then

$$f(t) \approx V(P^T V)^{-1} P^T f(t). \tag{6.35}$$

We have two choices left:

1. The projection basis $\begin{bmatrix} v_1 & \dots & v_m \end{bmatrix}$.
2. The interpolation indices $\begin{bmatrix} p_1 & \cdots & p_m \end{bmatrix}$.

We select the interpolation indices by applying the inductive Algorithm 3. The idea is to add an interpolation point at the index p_i which has the maximal error in the basis considered up to component i. We iterate by considering more basis components and adding one point for each basis vector.

Algorithm 3 DEIM

$p_1 = \text{index\_of\_maximum}(|v_1|)$
$V = [v_1],\ P = [e_{p_1}],\ p = [p_1]$
for $i = 2$ to m **do**
  Solve $c = (P^T V)^{-1} P^T v_i$
  residuum $= v_i - Vc$
  $p_i = \text{index\_of\_maximum}(|\text{residuum}|)$
  $V = [V, v_1],\ P = [P, e_{p_1}],\ p = [p, p_1]$
end for

Lemma 6.5.11. *[CS10] Let $f \in \mathbb{R}^n$, $\{v_1, \dots, v_m | v_i \in \mathbb{R}^n\}$ be a set of orthonormal vectors and*

$$\tilde{f} = V(P^T V)^{-1} P f,$$

with $V = \begin{bmatrix} v_1 & \dots & v_m \end{bmatrix}$ and $P = \begin{bmatrix} e_{p_1} & \cdots & e_{p_m} \end{bmatrix}$. An error bound for $\tilde{f}$ is given by

$$\|f - \tilde{f}\|_2 \leq C \mathscr{E}_*(f),$$

where

$$C = \|(P^T V)^{-1}\|_2, \qquad \mathscr{E}_*(f) = \|(\mathrm{Id} - VV^T) f\|_2.$$

Lemma 6.5.12. *[CS10] By using Algorithm 3 for the selection of the projection P in Lemma 6.5.11 the approximation of f is the best 2-norm approximation from the space* $\mathrm{Im}(V)$*. Furthermore we get a bound on C by*

$$C \leq \frac{(1+\sqrt{2n})^{m-1}}{|e_{p_1}^T u_1|} = (1+\sqrt{2n})^{m-1} \|u_1\|_\infty^{-1}.$$

Remark 6.5.13. *The main saving in computation time of this method is due to the projection $P^T f$ in* (6.35)*. This selection of m rows out of the n components of f allows to reduce the evaluation to these indices (the value at the others is approximated by the used method) such that if the components of f can be computed independently we save much of the computation time. However, the more components of f are needed to evaluate one selected index of the function the more we will recover the original evaluation time.*

For model reduction we apply the method to the system

$$\dot{x} = f(x)$$

and obtain the reduced system using the POD projection basis U_k

$$\begin{aligned} & \dot{\hat{x}} = U_k^T \tilde{f}(U_k \hat{x}) \\ \Leftrightarrow \quad & \dot{\hat{x}} = U_k^T V (P^T V)^{-1} P^T f(U_k \hat{x}). \end{aligned}$$

The matrix $U_k^T V (P^T V)^{-1}$ may be pre-computed.

Remark 6.5.14. *For bringing DEIM to the application in structural mechanics we struggle on how to evaluate only some selected components since, by the mesh, everything seems to be connected.*

Chapter Seven

Simulation Examples

After getting the POD and lookup methods in the last chapter, we will show a few examples of the methods applied to structural dynamical problems.

7.1 Training

As of now we have not talked about what to do in the case of a variable input u, which u shall be used for the training of the time-snapshots matrix

$$X = \begin{bmatrix} x(t_1, u(t_1)) & \cdots & x(t_n, u(t_n)) \end{bmatrix} . \tag{7.1}$$

This is a crucial aspect, as the POD reduced system will only be valid if the system's dynamics are captured in the projected subspace as seen at the end of Section 6.3.

Since a projection basis U_k (6.18) depends only on the observed time-snapshots (7.1) for a different input signal $\tilde{u}$, we expect the reduced model to be valid only if the states of the solution trajectory $\phi(x, \tilde{u}, t)$ can be expressed by the projection basis without loosing much of their information. So the projection basis clearly is not independent of the system's inputs while snapshot generation. Let's assume we know an input signal $u_d(t)$ to our system, in advance of the simulation (for $t \in [0, t_{end}]$). Is there a way of obtaining X (7.1) for u_d? Obviously yes, since X can be computed via a full-system simulation. Taking this X as usual to calculate the POD basis leads to an appropriate projection basis U_k. However, this procedure is very expensive for large time-periods.

Remark 7.1.1. *The input signal u_d is obtained by following a hierarchical approach. Remember that in our motivation the reduced structural mechanical model is only a part of a bigger system. A structural mechanical sub-system can be approximated by a linearization (or in other terms by springs and dampers). Simulating the larger system using this approximated sub-system gives some input signal u_d, which can be used to train a reduced model. Following this way the reduced system is another approximation to the full sub-system and the procedure may be repeated.*

Definition 7.1.2 (Input dependent projection-basis)**.** *For a differential equation $\dot{x} = f(x, u)$ with given initial values, we denote by $\Xi(u)_k$ the POD projection basis of size k generated using $X = \begin{bmatrix} x(0, u(0)) & \cdots & x(t_n, u(t_n)) \end{bmatrix}$ for sufficiently many equally spaced time-steps $t_1 \dots t_n$, $t_1 = 0$.*

The challenge is to synthesize an input $u_T(t)$ for $t \in [0, t_T]$ with $t_T \ll t_{end}$ such that $\Xi_k(u_T) = \Xi_k(u_d)$. Denote by T_k the set of all input-signals u which give rise to the same projection basis

$$T_k = \{u | \Xi_k(u) = \Xi_k(u_d)\} .$$

Then we are searching for those $u(t) \in T_k$ having minimal "end-time" t_T.

Remark 7.1.3. *The requirements to u may be relaxed by considering that the snapshot-matrix X may be composed of different inputs u_i, $i = 1 \dots n$*

$$X = \bigcup_{i=1}^{n} X_i,$$
$$X_i = \begin{bmatrix} x(0, u_i(0)) & \cdots & x(t_{n_i}, u_i(t_{n_i})) \end{bmatrix}.$$

Synthesizing u_T seems to be very difficult and leads through an inverse problem. Consider the singular value decomposition of Ξ_k

$$\Xi_k = \begin{bmatrix} \Xi_k & \zeta \end{bmatrix}^T \begin{bmatrix} \mathrm{Id}_k \\ 0 \end{bmatrix} \mathrm{Id}_k \qquad \text{with} \qquad \zeta \perp \Xi_k.$$

We see that an optimal input u_T could be the input for which the time-snapshot matrix is already equal to the projection basis, so we are searching the inverse input to the POD modes which are unknown up to this point. This seems to be very difficult.

Remark 7.1.4. *Another brief idea could be formulated in the spirit of balanced truncation. We have to remove those inputs or time-intervals from u_d which don't generate much energy in the output and retain those which give rise to large singular values.*

These approaches and others to obtain optimal input signals for snapshot generation is out of the scope of this thesis. In structural mechanics we will in the following give an example for which it was possible to use a reduced model for a longer time period than the corresponding training. A similar observation was made in [Her08].

For the selection of this training input we rely on some heuristics and model properties, in this way we selected a seemingly characteristic excitation and used it for the training.

Remark 7.1.5. *Even if we don't know how to choose a training, we can use the knowledge on how sensitive our projection basis is to the training by looking into the classic theorems on the singular value decomposition, for example, the Theorem of Wedin 6.2.9 and the Theorem of Mirsky 6.2.7.*

7.2 Example: 2D bushing

As our first example of POD we modeled a 2D rubber part under the assumptions of plane strain. The part we are going to simulate (Figure 7.1) consists of three rings where the outer rings shall be steel rings while the inner part represents a compressible rubber-like material. The material of the smaller and outer rings (coloured in red) are modeled as nearly rigid by a linear elastic isotropic material with a Young's modulus of $200.0[GPa]$ and Poisson ratio of 0.33. In the central ring (coloured gray) we use a compressible hyperelastic material model with bulk modulus $\kappa = 30[MPa]$. The isochore energy strain function is chosen as the relation of Mooney-Rivlin (Example 2.2.5), where we used the parameters $c_{10} = 0.4[MPa]$ and $c_{01} = 0.1[MPa]$. The density of the hyperelastic part is $1.1 \cdot 10^{-9}[\frac{t}{mm^3}]$ while both other rings have got $7.85 \cdot 10^{-9}[\frac{t}{mm^3}]$. The radius of the inner ring is $10[mm]$ while the outer ring is of radius $25[mm]$.

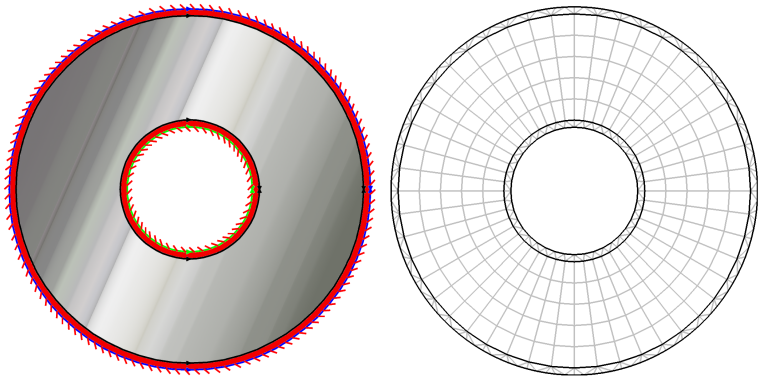


Figure 7.1: 2D rubber part and mesh

The boundary conditions of the simulation are a fixed position of the outer ring while the inner ring is displaced (as a whole) in horizontal and vertical direction. The structure is discretized using linear finite elements. The mesh, represented in Figure 7.1, is composed of 496 elements and carries 1248 degrees of freedom.

For a demonstration of the POD method we follow the steps of Definition 6.3.1 and generate the matrix of time-snapshots using a displacement via

$$u(t) = \begin{bmatrix} A_1 \sin(2\pi f_1 t) \\ A_2 \cos(2\pi f_2 t)\left(1 - \frac{1}{(1+t)^6}\right) \end{bmatrix} \qquad \text{with} \qquad t \in [0,1], \tag{7.2}$$

$A_1 =, A_2 = 9[mm]$, $f_1 = f_2 = 3[Hz]$, of the inner ring.

The calculation is first performed for the full-system. We utilize the full simulation to construct the snapshot matrix

$$X = \begin{bmatrix} q(t_1, u(t_1)) & \cdots & q(t_N, u(t_N)) \end{bmatrix}$$

and the reduction basis U_k. Additionally we use the solution $q(t)$ as a reference solution to compare the simulation results of full a reduced models. In the Figures 7.2 and 7.3 we show some of the basis vectors (POD modes) selected through the training, i.e., columns of U_k.

Doing a simulation of the projected system and comparing it to the full simulations, we plot the relative difference in Figure 7.4. We see that it decreases with the number of used basis vectors. For $k = 200$ we observe a relative error of $\leq 0.1\%$ in position and velocity components. The system represented in the reduced coordinates $\hat{q}$ is depicted in Figure 7.5. We see the contribution of the different basis components (also corresponding to the bases, partly seen in Figure 7.2 and Figure 7.3). Notice how the absolute value of contribution decreases while the index of the bases increases. This is due to the minimal rank property of the singular value decomposition (Theorem 6.2.3). Also observe that the first and second POD mode are the only needed to describe the movement of the inner ring.

For a qualitative comparison between the different projections we plotted (Figure 7.6) the trajectory of one selected degree of freedom inside the bushing using different basis sizes within the POD method.

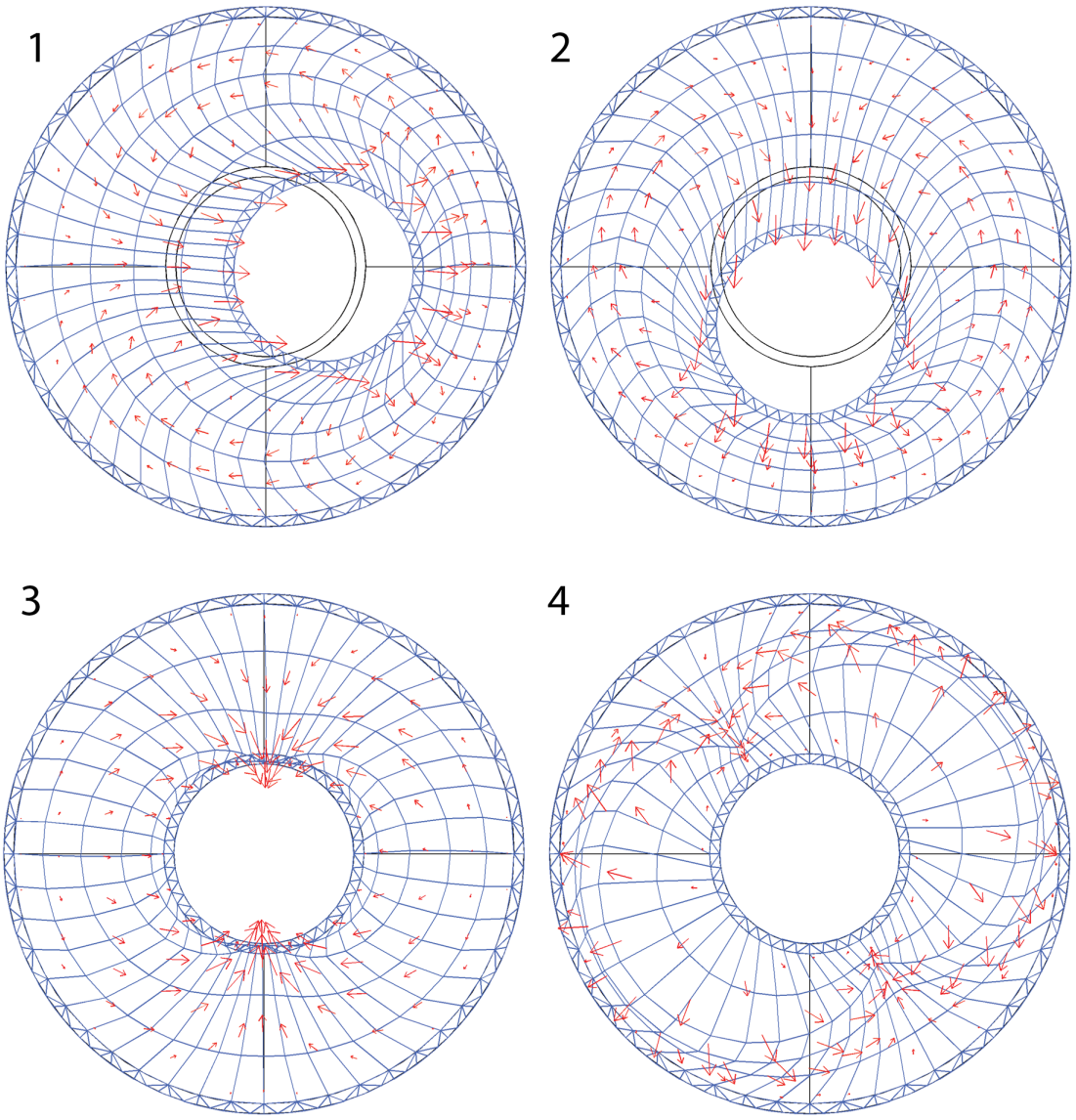


Figure 7.2: POD basis vectors numbered by corresponding singular value

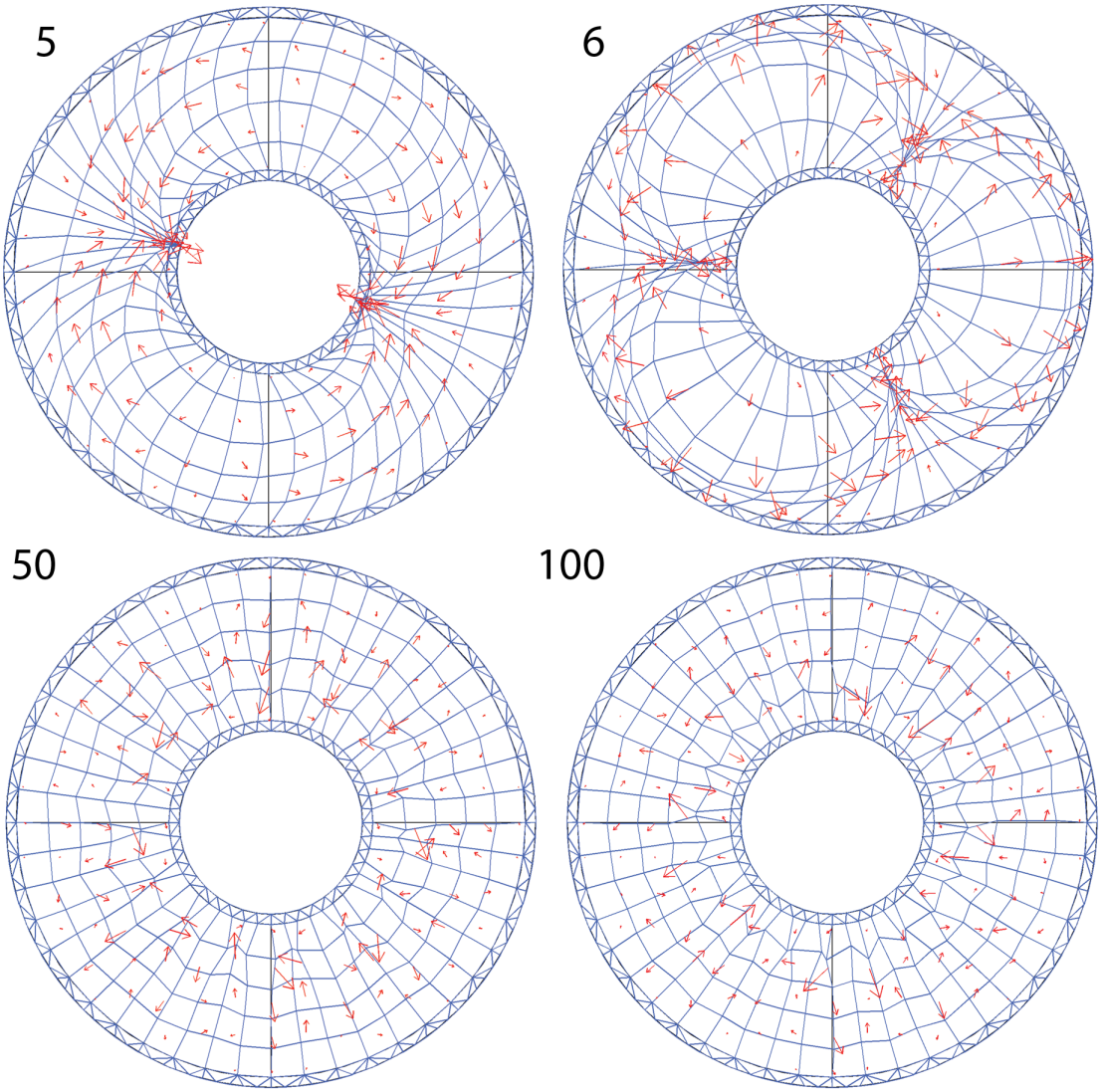


Figure 7.3: continued: POD basis vectors

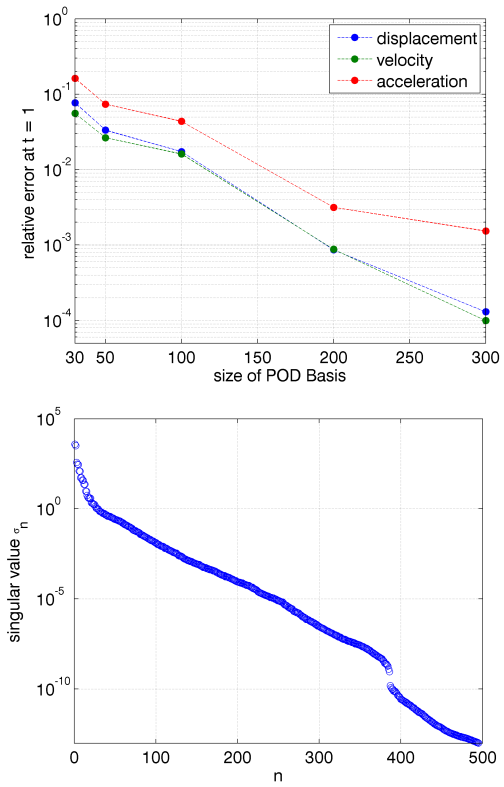


Figure 7.4: Relative Error of the POD method with respect to the used basis size and decay of singular values

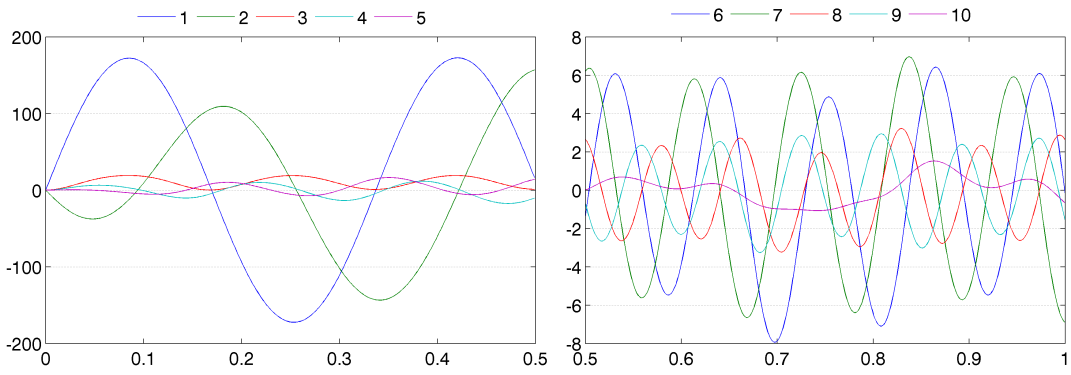


Figure 7.5: Solution of 2D bushing example represented in reduced coordinates

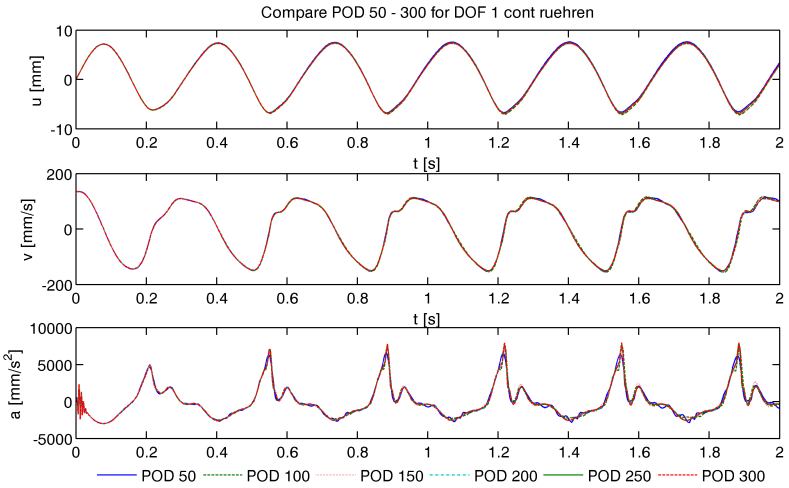


Figure 7.6: Qualitative comparison of different POD approximations in the 2D bushing example

Example mixed formulation

For a reduction of the mixed singularly singular perturbed system as discussed in Part I we proceed like described in the subsection on POD for perturbed systems on page 84.

The example is computed for the same geometry and material parameters used at the begin of this section (Figure 7.1), but we use a mixed formulation to handle deformation q and pressure p separately as described in Section 2. The choice of finite elements has to be adapted to attain a well-defined system as discussed in Section 3.2. So to fulfill the inf-sup condition we choose a second order approximation of displacement quantities q and a linear one for the pressure p. By this the number of degrees of freedoms increases from 1248 to 3264 while the number of elements is retained at 496.

For obtaining a perturbed system we increase the bulk-modulus to

$$\kappa = 3 \cdot 10^5 [MPa]$$

(which also is a typical, physically observed value for rubber materials [Tab94]), and get a Poisson ratio of $\nu = 0.499998$.

The excitation used in the example is again a displacement of the inner ring by

$$u(t) = 9 \left(\frac{1}{1+e^{-10t+8}} - \frac{1}{1+e^{8}} \right) \sin(2\pi f t)[mm].$$

in up-down direction while the outer ring is fixed. The calculated reference solution for all nodes in q and p is plotted in Figure 7.7.

We consider approach (b) and (c) as discussed on page 84. For the projection basis U_k snapshots of the exact solution q are used. In the following k, the size of the basis, will be fixed to 100.

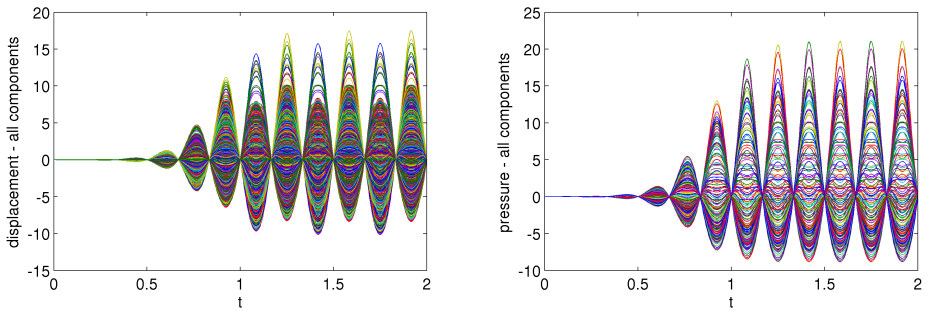


Figure 7.7: Referenced solution of Displacement and pressure of all components over time

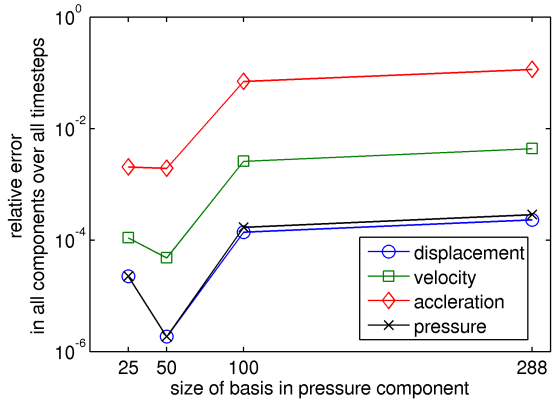


Figure 7.8: Relative error using different size of bases in pressure while fixing the basis to 100 elements in displacement quantities

For approach (b) all 288 p variables are retained, and a simulation of the reduced model is done using the same input trajectory $u(t)$, used in the full simulation. In rightmost point of Figure 7.8 we see the obtained relative error in comparison to the reference simulation.

Introducing a second POD basis using snapshots of p

$$X_\lambda = \begin{bmatrix} p(t_1, u(t_1)) & \cdots & p(t_N, u(t_N)) \end{bmatrix},$$

we construct a separate projection basis L_m for the pressure components following approach (c). The relative error of the reduced model for different projection sizes m is plotted in Figure 7.8. We are astonished to see a decreasing error for decreasing size of m. Since less information is used the error shall increase.

Remark 7.2.1. *In our test implementation the computation time was not dominated by the solution of the nonlinear system, hence we used no lookup method and had a communication overhead while coupling multiple simulation tools. Thus our savings were as estimated in Section 6.4 small. In the next section we see an example with significant savings due to the lookup method.*

7.3 Example: Detailed 3D bushing

For a the next example we want to come to the model of a real rubber-made bushing (Figures 7.9, 7.10).

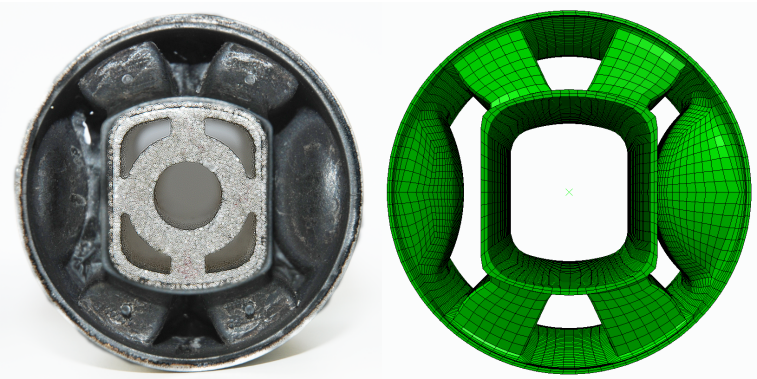


Figure 7.9: Real bushing and its finite element model

The corresponding finite element model as provided by Vibracoustic GmbH&Co KG consists of $\approx 30,000$ elements yielding ≈ 100.000 degrees of freedom. As material model we used again a description by Mooney-Rivlin (Example 2.2.5).

Full system simulation

For discretization and full system simulations we relied on a commercial finite element package, in this way we could include the inner contact areas and the rigid body constraints for inner and outer elements into the simulation. From an input to output view, the model consists of one force transmission point connected to the inner surface, and one force transmission point connected to the outer surface. Thus the model inputs and outputs are, respectively, only the 6 degrees of freedom (displacements in all 3 directions and rotations around 3 axis) of the two connecting rigid bodies. We choose the classical formulation (**not** the mixed form) while decreasing the bulk modulus reasonably and consider standard linear finite elements.

For the simulation, forces and moments are prescribed at the inner rigid body

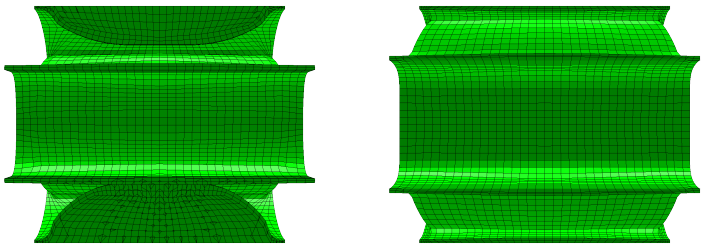


Figure 7.10: Side view of horizontal and vertical cut through the middle plane of finite element model.

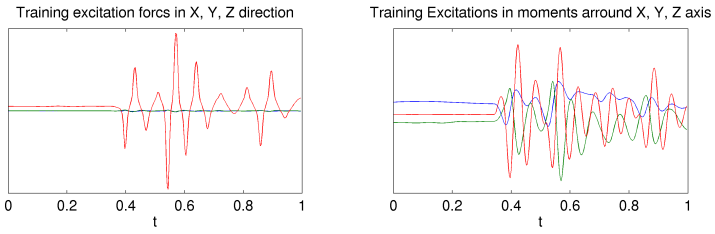


Figure 7.11: Selected training excitation

while the position of the outer one is fixed in all degrees of freedom. The simulation-results are thus the generalized deformations of the middle reference point.

To obtain also a realistic excitation we took a multibody simulation in which the bushing of choice was modeled using only some characteristic curves of the part. We will bring a part of these excitation to the finite element model and will use them for the training of the reduced model. The used force and moment trajectories can be seen in Figure 7.11.

All results of the full model are calculated by a commercial software, using an explicit variant of Newmarks method and step-size $h = 10^{-6}[s]$. Explicit integration was a necessary choice in our simulation tool because of the huge number of freedoms, also the explicit integrator forced us to select the compressible material model. Unluckily we couldn't interface the tool for implementing the presented integration methods of Chapter 5.

The simulation time of the example is only 1 second. In Figure 7.12 one can see the displacements corresponding to the inputs of Figure 7.11. We see how large the deformations inside the part are and how different states of contact are realized.

Reduced system simulation

Consider the discretized system to be of the form

$$M\ddot{q} = f(q) + f_{ext}(t). \tag{7.3}$$

The described full-simulation is used to obtain the following data at a number of time-steps t_i, $i = 1 \dots n$:

- time-snapshots of deformation states $q_i = q(t_i)$,
- inner-force vector $f_i = f(q_i)$,
- linearization at different time-steps

$$K_i = \frac{\partial f(q)}{\partial q}|_{q_i}. \tag{7.4}$$

Remark 7.3.1. *In our example the matrices* (7.4) *have roughly* $3 \cdot 10^6$ *nonzero elements, obtaining them from a commercial tool for projection is rather technical but unluckily not trivial, since they are usually only created and needed while solving the system and are additionally not directly available for export. We had to save intermediate steps while solving the full dynamics and do a re-initialization using the saved states to start the matrix extractions.*

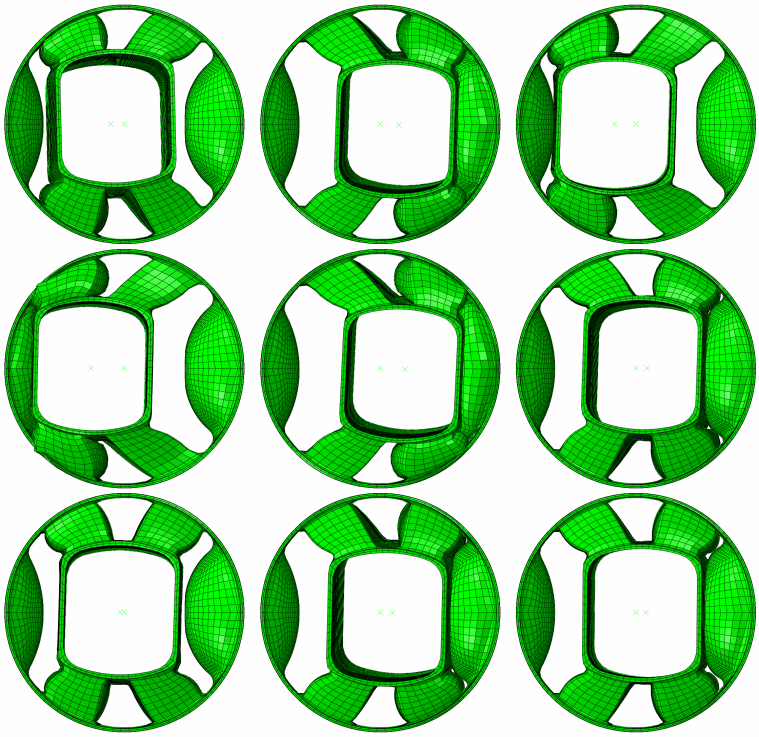


Figure 7.12: Snapshots of unscaled displacements of full simulation using the excitation seen in Figure 7.11 at time-steps (from top left to bottom right) $t = [0.39, 0.43, 0.47, 0.55, 0.57, 0.72, 0.77, 0.89]$

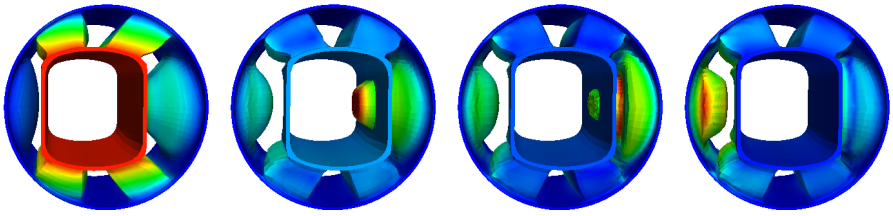


Figure 7.13: First four POD-modes calculated via the full displacement snapshots, note that modes do not automatically obey the constraint

In our example we used the data of 300 equidistantly distributed time-steps in the one second simulation interval. The time-snapshots

$$X = \begin{bmatrix} q_1 & \cdots & q_{300} \end{bmatrix}$$

are used to calculate projection basis U_k, the first four POD-modes can be seen in Figure 7.13. The force is only applied to one rigid body node, consisting of 6 degrees of freedom. We added this node and the rigid body output node as described in Chapter 6.4 (page 84) by an identity into the projection basis.

The collected inner-force vector f_i, at q_i and linearization K_i also at q_i for $i = 1 \dots 300$ are projected to

$$\tilde{f}_i = U_k f_i, \qquad \tilde{q}_i = U_k q_i, \qquad \tilde{K}_i = U_k K_i U_k^T \tag{7.5}$$

and used to construct a lookup table for the TPWL approximation (Definition 6.5.4)

$$\begin{aligned} \hat{f}(\tilde{q}) &= \sum_{l=1}^{s} \omega_l(\tilde{q})(\tilde{f}_l + K_l(\tilde{q} - \tilde{q}_l)), \\ \omega_l(\tilde{q}) &= \|\tilde{q} - \tilde{q}_l\| \left(\sum_{j=1}^{s} \|\tilde{q} - \tilde{q}_j\| \right)^{-1}, \end{aligned} \tag{7.6}$$

with indices sorted such that

$$\|\tilde{q} - \tilde{q}_1\| \leq \|\tilde{q} - \tilde{q}_2\| \leq \ldots \leq \|\tilde{q} - \tilde{q}_s\|.$$

Putting lookup and projection together into (7.3) we end up at the reduced system

$$\tilde{M}\ddot{\tilde{q}} = \hat{f}(\tilde{q}) + \tilde{f}_{ext}(t) \tag{7.7}$$

with precalculated values of

$$\tilde{M} = U_k M U_k^T, \qquad \tilde{f}_{ext} = U_k f_{ext}(t).$$

By choosing the projection of size $k = 250$ we obtain system (7.7) with only 250 degrees of freedom and due to precalculated projections of (7.5) and the used lookup (7.6) we have no need for back-projecting into the full model. Thus the reduced system can be solved independently of the full system, as long as the lookup yields a good approximation, and the reduced system is valid as long as the used projection basis can capture the displacements.

For the reduced system we have all data at hand and so use the generalized-α method from Section 5.2 to solve the system. The solution is calculated not only for the used training excitation (Figure 7.11), but also for an extended 4 second signal depicted in Figure 7.14. The additional 3 seconds arise from the same multibody simulation and are not equal to first second.

The obtained trajectory of the inner rigid-body in all its six degrees of freedom is plotted in Figure 7.15. For comparison we did another simulation of the full model for the whole 4 second interval which can be seen as the reference solution in Figure 7.15. We see a good accordance between reduced and full simulation. Especially, the more significant components (larger deformations) are captured adequately. Components of smaller magnitude are overall still following the reference solution but the reduced solution leaves off more often.

In Table 7.1 we listed the times which had to be spent for the different steps. Comparing the simulation times of the reduced System using the lookup method, which were $126[s]$ including back projection, and the simulation of the full model for the 4 second signal, which took $45[h]$ on 4 CPU cores, we see a significant saving of a factor ≈ 1.000. However, we also have to take into account the necessary steps for obtaining the reduced system. The full simulation to train the reduction basis and lookup method took $12[h]$. Because of the bad interface (Remark

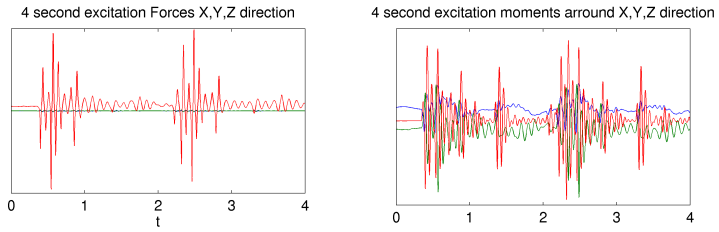


Figure 7.14: Extended excitation for the reduced system

7.3.1), extracting the required data from the simulation took almost as long as the training simulation. After this, we left the commercial tool behind and calculated the projection basis in $3[h]$. So in total we spend $27[h]$ to obtain the reduced system which can then be solved in $126[s]$.

step	CPU- time
full system simulation, for 1 [s]	12 [h]
data extraction for basis and lookup u_i, f_i, K_i	12 [h]
calculation of projection basis $3[h]$ reduced system simulation	126 [s]
full system simulation, for 4 [s]	45 [h]

Table 7.1: Comparison of computational costs at the different steps

7.4 Conclusion

For a model reduction of finite element models we have presented in Part II the POD method which can be applied to a rather general class of nonlinear systems. We discussed how the technique can be applied to a structural mechanical problem. The need of a simulation using the full system couldn't be completely removed, but still an advantage is gained since the full system can be simulated very efficiently in the case of nearly incompressible hyperelastic materials using the optimized methods of Part I.

We saw that the nonlinear model reduction can be applied to our class of problems, but to save computational costs it has to be split into two parts. One part handles the pure dimension reduction. The other part needed is an efficient interpolation of the nonlinear equations so that POD with lookup gave us significant savings. The method has got a high potential. However a further development of interpolation techniques could lead to improved results by attaining larger areas of validity.

Also for a model reduction of the singularly perturbed system we showed how the POD method can be applied to the constrained system and did a few successful numerical experiments. In general a better understanding on how to handle constraints inside of a reduced model is necessary.

The reduced finite element model can now be included into a multibody system. It needs to be investigated how large the area of validity has to be for specific

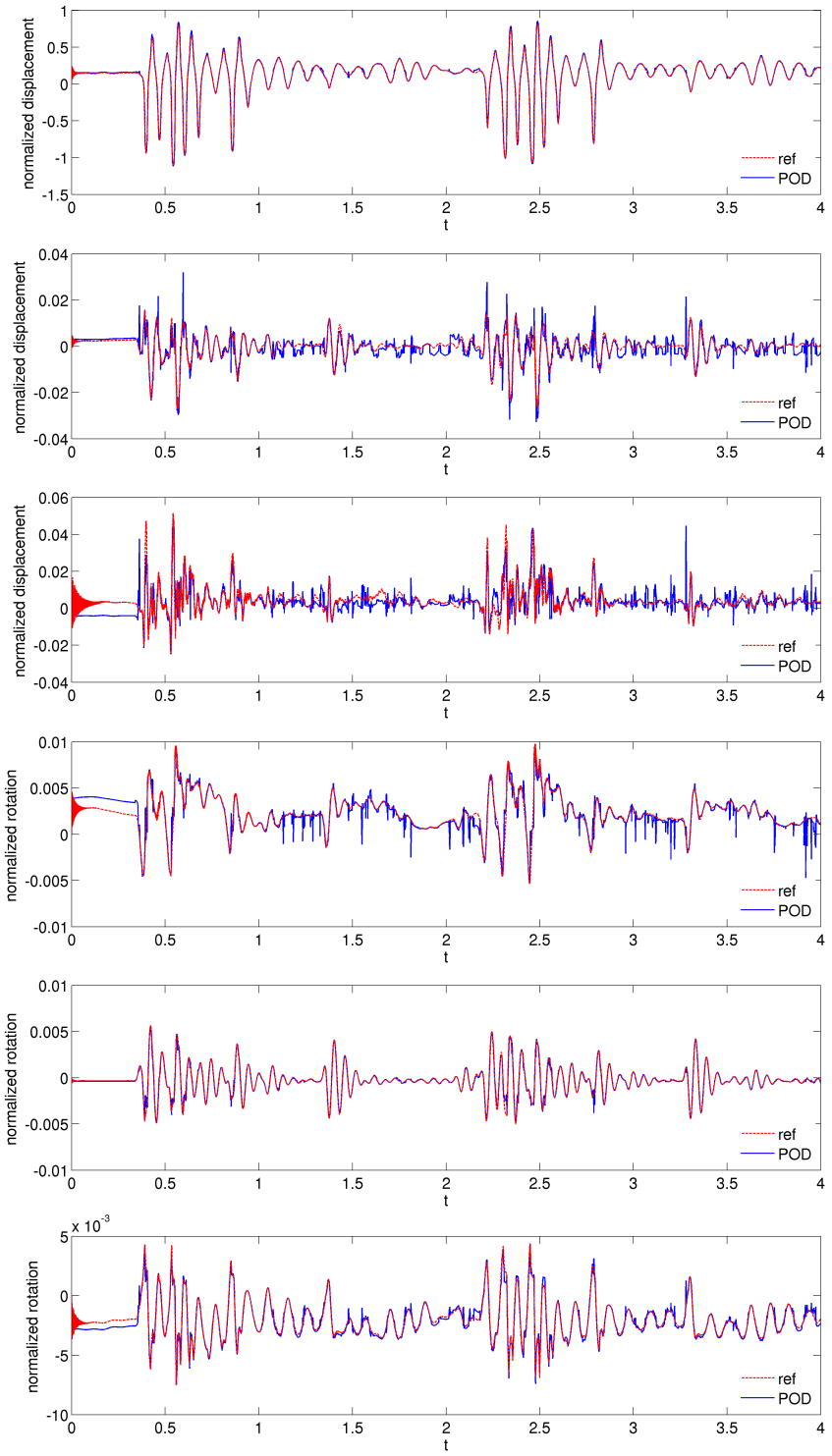


Figure 7.15: Result of POD Lookup simulation in comparison to reference solution for all six components. Notice that the POD Lookup method was trained using only the first second of the full simulation. Excitation is taken from Figure 7.14

use-cases. The accuracy of the used model reduction can be adapted by changing the training excitation, the size of the reduced system and the accuracy of the used lookup method.

A further extension of our work can be done by considering also damping effects inside the rubber-bushing. The techniques for dimension reduction should be similar to the presented methods, but a different approach may be needed in the lookup method if one also wants to include velocity information.

Bibliography

[AB07] M. Arnold and O. Brüls. Convergence of the generalized-alpha scheme for constrained mechanical systems. *Multibody Sys Dyn*, 18:185–202, 2007.

[AB08] M. Arnold and O. Brüls. The generalized-alpha scheme as a linear multistep integrator: Toward a general mechatronic simulator. *Jornal of Computational and Nonlinear Dynamics*, 3:041007, 2008.

[ADKL01] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[AGLP06] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156, 2006.

[Ant05] A. C. Antoulas. *Approximation of large-scale dynamical systems.* SIAM, 2005.

[AWWB08] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control*, 53(10):2237–2251, 2008.

[BB11] T. Breiten and P. Benner. Interpolation-based $\mathscr{H}_2$ model reduction of bilinear control systems. *submitted*, 2011. Available from `http://www.mpi-magdeburg.mpg.de/mpcsc/breiten/`.

[BBD10] P. Benner, T. Breiten, and T. Damm. Krylov subspace methods for model order reduction of bilinear discrete-time control systems. *PAMM*, 10(1):601–602, 2010.

[BD10] T. Breiten and T. Damm. Krylov subspace methods for model order reduction of bilinear control systems. *Systems & Control Letters*, 59(8):443–450, 2010.

[BDT08] C. Bottasso, D. Dopico, and L. Trainelli. On the optimal scaling of index three DAEs in multibody dynamics. *Multibody Syst Dyn*, 19:3–20, 2008.

[Bel73] E. Beltrami. Sulle funzioni bilineari. *Giornale di Matematiche ad Uso degli Stedenti Delle Universita*, (11):98–106, 1873.

[BF91a] F. Brezzi and R. S. Falk. Stability of higher-order Hood-Taylor methods. *SIAM J Numer Anal*, 28(3):581–590, June 1991.

[BF91b] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*. Springer, 1991.

[BMS05] P. Benner, V. L. Mehrmann, and D. C. Sorensen. *Dimension reduction of large-scale systems: proceedings of a workshop held in Oberwolfach, Germany, October 19-25, 2003*. Lecture Notes in Computational Science and Engineering. Springer, 2005.

[BS11] P. Benner and J. Saak. Efficient balancing based MOR for large scale second order systems. *Mathematical and Computer Modeling of Dynamical Systems*, 17(2):123–143, 2011.

[BTDW04] T. Bui-Thanh, M. Damodaran, and K. Wilcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42(8):1505–1516, 2004.

[BVB+11] S. Bruni, J. Vinolas, M. Berg, O. Polach, and S. Stichel. Modeling of suspension components in a rail vehicle dynamics context. *Vehicle System Dynamics*, 49:1021–1072, 2011.

[CH93] J. Chung and G. Hulbert. A time integration algorithm with improved numerical dissipation: the generalized alpha method. *ASME J.Appl.Mech.*, 60:371–375, 1993.

[CS10] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.*, 32(5):2737–2764, 2010.

[Das11] Dassault Systemes. *ABAQUS Theory Manual*, 6.11 edition, 2011.

[Dav04] T. A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, 30(2):165–195, June 2004.

[Dav06] T. A. Davis. *Direct methods for sparse linear systems*. SIAM Book Series on the Fundamentals of Algorithms. SIAM, Philadelphia, Sept. 2006.

[FE11] J. Fehr and P. Eberhard. Simulation process of flexible multibody systems with non-modal model order reduction techniques. *Multibody System Dynamcis*, 25(3):313–334, 2011.

[FS05] K. Fujimoto and J. M. A. Scherpen. Nonlinear input-normal realizations based on the differential eigenstructure of hankel operators. *IEEE Transactions on Automatic Control*, 50(1):2–18, January 2005.

[FS10] K. Fujimoto and J. M. A. Scherpen. Balanced realization and model order reduction for nonlinear systems based on singular value analysis. *SIAM J. Control Optim.*, 48(7):4591–4623, 2010.

[GLS88] K. Gustafsson, M. Lundh, and G. Söderlind. A PI stepsize control for the numerical solution of ordinary differential equations. *BIT Numerical Mathematics*, 28:270–287, 1988.

[GvL96] G. Golub and C. van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

[Hau08] J. Hauth. *Grey-Box modeling for nonlinear systems*. PhD thesis, University of Kaiserslautern, 2008.

[HDO11] B. Haasdonk, M. Dihlmann, and M. Ohlberger. A training set and multiple bases generation approach for parametrized model reduction based on adaptive grids in parameter space. *Mathematical and Computer Modeling of Dynamical Systems*, 17(4):423–442, 2011.

[Her08] S. Herkt. *Model reduction of nonlinear problems in structural mechanics: Towards a finite element tyre Model for multibody simulations*. PhD thesis, University of Kaiserslautern, 2008.

[HLR89a] E. Hairer, C. Lubich, and M. Roche. *The numerical solution of differential-algebraic systems by Runge-Kutta methods*, volume 1409 of *Lecture Notes in Mathematics*. Springer, 1989.

[HLR89b] E. Hairer, Ch. Lubich, and M. Roche. Error of Rosenbrock methods for stiff problems studied via differential algebraic equations. *BIT Numerical Mathematics*, 29:77–90, 1989.

[HNW08] E. Hairer, S. P. Norsett, and G. Wanner. *Solving ordinary differential equations I*. Springer, 2008.

[Hol07] G. A. Holzapfel. *Nonlinear solid mechanics*. Wiley, 2007.

[HT73] P. Hood and C. Taylor. A numerical solution of navier-stokes equation using the finite element technique. *Comp. and Fluids*, 1:73–100, 1973.

[HV08] M. Hinze and S. Volkwein. Error estimates for abstract linear–quadratic optimal control problems using proper orthogonal decomposition. *Computational Optimization and Applications*, 39:319–345, 2008.

[HW96] E. Hairer and G. Wanner. *Solving ordinary differential equations II*. Springer, 1996.

[Jay93] L. Jay. Collocation methods for differential-algebraic equations of index 3. *Numerische Mathematik*, 65:407–421, 1993.

[Joh05] R. S. Johnson. *Singular perturbation theory: Mathematical and analytical techniques with applications to engineering*. Springer, 2005.

[Jol02] I. T. Jolliffe. *Principal component analysis*. Springer Series in Statistics. Springer, 2002.

[KLM01] P. Krysl, S. Lall, and J. E. Marsden. Dimensional model reduction in non-linear finite element dynamics for solids and structures. *Int. J. Numer. Meth. Engng*, 51:479–504, 2001.

[KPB85] P. Kaps, S. Poon, and T. Bui. Rosenbrock methods for stiff ODEs: A comparison of Richardson extrapolation and embedding technique. *Computing*, 34:17–40, 1985.

[KV01] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische Mathematik*, 90:117–148, 2001.

[KV03] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40(2):pp. 492–515, 2003.

[Lju87] L. Ljung. *System identification: Theory for the user*. Prentice-Hall Information and system sciences series. Prentice-Hall, 1987.

[LKM03] S. Lall, P. Krysl, and J. E. Marsden. Structure-preserving model reduction for mechanical systems. *Physica D*, 184:304–318, 2003.

[LLL+02] Y. C. Liang, W. Z. Lin, H. P. Lee, S. P. Lim, K. H. Lee, and H. Sun. Proper orthogonal decomposition and its applications – part II: Model reduction for MEMS dynamical analysis. *Journal of Sound and Vibration*, 256(3):515–532, 2002.

[LMG02] S. Lall, J. E. Marsden, and S. Glavaški. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *Int J Robust Nonlin*, 12(6):519–535, 2002.

[LT08] J. Lang and D. Teleaga. Towards a fully space-time adaptive FEM for magnetoquasistatics. *Magnetics, IEEE Transactions on*, 44(6):1238–1241, June 2008.

[Lub93] C. Lubich. Integration of stiff mechanical systems by Runge-Kutta methods. *ZAMP*, 44:1022–1053, 1993.

[LV01] J. Lang and J. Verwer. ROS3P - an accurate third-order Rosenbrock solver designed for parabolic problems. *BIT Numerical Mathematics*, 41:731–738, 2001.

[MH94] J. E. Marsden and T. J. R. Hughes. *Mathematical foundations of elasticity*. Dover, 1994.

[Mir60] L. Mirsky. Symmetric gage functions and unitariliy invariant norms. *Quarterly Journal of Mathematics*, pages 50–59, 1960.

[Moo81] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *Automatic Control, IEEE Transactions on*, 26(1):17–32, feb 1981.

[New59] N. Newmark. A method for computation for structural dynamics. *ASCE J. Eng. Mech. Div.*, 85:67–94, 1959.

[NKEB12] C. Nowakowski, P. Kürschner, P. Eberhard, and P. Benner. Model reduction of an elastic crankshaft for elastic multibody simulations. Max Planck Institute Magdeburg Preprint MPIMD/12-07, MPI-Magedburg, March 2012. Available from `http://www.mpi-magdeburg.mpg.de/preprints/`.

[Ogd72] R. W. Ogden. Large deformations isotropic elasticity - on the correlation of theory and experiment for compressible rubber-like solids. *Proceedings of the Royal Society of London*, A328:567–583, 1972.

[Ogd97] R. W. Ogden. *Non-linear elastic deformations*. Dover, 1997.

[PB97] D. Pantuso and K.-J. Bathe. On the stability of mixed finite elements in large strain analysis of incompressible solids. *Finite Elements in Analysis and Design*, 28:83–104, 1997.

[PR74] A. Prothero and A. Robinson. On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. *Math. Comp.*, 28:145–162, 1974.

[REOM88] Jr. R. E. O' Malley. On nonlinear singularly perturbed initial value problems. *SIAM Rev.*, 30:193–212, 1988.

[Rew03] M. J. Rewienski. *A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, 2003.

[Roc88] M. Roche. *Méthodes de Runge-Kutta et Rosenbrock Pour Équations Différentielles Algébriques et systèmes différentiels Raides*. PhD thesis, University of Geneva, 1988.

[Ros63] H. H. Rosenbrock. Some general implicit processes for the numerical solution of differential equations. *Computer J.*, 5(4):329–330, 1963.

[RP03] M. Rathinam and L. Petzold. A new look at proper orthogonal decomposition. *SIAM J. Numer. Anal.*, 41(5):1893–1925, 2003.

[RS08] T. Reis and T. Stykel. Balanced truncation model reduction of second-order systems. *Mathematical and Computer Modeling of Dynamical Systems*, 14(5):391–406, 2008.

[Sch89] S. Scholz. Order barriers for the B-convergence of ROW methods. *Computing*, 41:219–235, 1989.

[SDR11] K. Sedlaczek, S. Dronka, and J. Rauh. Advanced modular modeling of rubber bushings for vehicle simulations. *Vehicle System Dynamics*, 49:741–759, 2011.

[SHKH10] A. Schlecht, B. Heißing, H. Krome, and G. Hackenberg. Entwicklung einer schwingungsunempfindlichen Vorderachskinematik. In *VDI-Jahrbuch Fahrzeug- und Verkehrstechnik*, number 2010-03. Springer Vieweg, 2010.

[Sim98] B. Simeon. Order reduction of stiff solvers at elastic multibody systems. *Applied Numerical Mathematics*, 28(2-4):459–475, 1998.

[Sim06] B. Simeon. On lagrange multipliers in flexible multibody dynamics. *Comput. Methods Appl. Engrg.*, pages 6993–7005, 2006.

[SL06] B. Salimbahrami and B. Lohmann. Order reduction of large scale second-order systems using krylov subspace methods. *Linear Algebra and its Applications*, 415(2–3):385–405, 2006. Special Issue on Order Reduction of Large-Scale Systems.

[SM92] J. C. Simo and C. Miehe. Associative coupled thermopalsticity at finite strains: Formulation, numerical analysis and implementation. *Computer Methods in Applied Mechanics and Engineering*, 98(41-104), 1992.

[Ste90] G. W. Stewart. Perturbation theory for the singular value decomposition. Technical Report CS-TR 2539, UMIACS, 1990.

[Ste93] G.W. Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566, 1993.

[Ste95] G. Steinebach. *Die Linienmethode und ROW-Verfahren zur Abfluss- und Prozesssimulation in Fliessgewässern am Beispiel von Rhein und Mosel*. PhD thesis, TU Darmstadt, 1995.

[Stu04] T. Stumpp. *Integration stark gedämpfter mechanischer Systeme mit Runge-Kutta-Verfahren*. PhD thesis, Universität Tübingen, 2004.

[SVB+97] A. Sandu, J. G. Verwer, J. G. Blom, E. J. Spee, G. R. Carmichael, and F. A. Porta. Benchmarking stiff ODE solvers for atmospheric chemistry problems II; solvers. *Atmospheric Enviorment*, 31(20):3459–3472, 1997.

[SW79] T. Steinhaug and A. Wolfbrandt. An attempt to avoid exact Jacobian and nonlinear equations in the numerical solution of stiff differential equations. *Math. Comp.*, 33:521–534, 1979.

[Tab94] D. Tabor. The bulk modulus of rubber. *Polymer*, 35(13):2759–2763, 1994.

[TVS85] A. N. Tikhonov, A. B. Vasiléva, and A. G. Sveshnikov. *Differential equations*. Springer, 1985.

[Wed72] P. A. Wedin. Perturbation bounds in connection with singular value decomposition. *BIT*, pages 99–111, 1972.

[Wen97] J. Wensch. *Zur numerischen Integration differential-algebraischer Systeme - partitionierte ROW-Methoden für Mehrkörpersysteme und Stabilität von Diskretisierungsverfahren für Index-2 Systeme*. PhD thesis, Martin-Luther-Universitäet Halle-Wittenberg, 1997.

[Wen98] J. Wensch. An eight stage fourth order partitioned Rosenbrock method for multibody systems in index-3 formulation. *Applied Numerical Mathematics*, 27(2):171–183, 1998.

[Wey12] H. Weyl. Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf der Theorie der Hohlraumstrahlung). *Mathematische Annalen*, 71:441–479, 1912.

[Wri08] P. Wriggers. *Nonlinear finite element methods*. Springer, 2008.

[YBBK12] Yue Yu, Hyoungsu Baek, M. L. Bittencourt, and G. E. Karniadakis. Mixed spectral/hp element formulation for nonlinear elasticity. *Comput. Methods Appl. Mech. Engrg.*, 213-216:42–57, 2012.

Scientific Career

Urs Becker

2004- 2009, study of techno-mathematics with application to mechanical engineering, at the Technical University of Kaiserslautern.

2009, diploma degree “Diplom-Technomathematiker (Dipl.-Math. techn.)” with focus on “Modeling and Scientific Computing”, diploma thesis “Iterativ lernende Regelung und Invariante Anregung”, Technical University of Kaiserslautern.

2009- 2012, scholarship of the Fraunhofer society at the Fraunhofer Institute for Industrial Mathematics ITWM, Kaiserslautern.

2009- 2012, PhD student at the Technical University of Kaiserslautern, Department of Mathematics.

Wissenschaftlicher Werdegang

Urs Becker

2004- 2009, Studium Technonmathematik im Anwendungsfach Maschinenbau an der Technischen Universität Kaiserslautern.

2009, "Diplom-Technomathematik (Dipl.-Math. techn.)" der Technischen Universität Kaiserslautern, Diplomarbeit "Iterativ lernede Regelung und invariante Anregung".

2009- 2012, Stipendium der Fraunhofer Gesellschaft, am Fraunhofer Institut für Techno- und Wirtschaftsmathematik ITWM, Kaiserslautern.

2009- 2012, Promotionsstudent der Technischen Universität Kaisreslautern am Fachbereich Mathematik.