# Reduced order modeling and parameter identification for coupled nonlinear PDE systems

**Dissertation**

zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

Oliver Lass

an der


Mathematisch-naturwissenschaftliche Sektion
Fachbereich Mathematik und Statistik

Tag der Mündlichen Prüfung: 21.02.2014

1. Referent: Prof. Dr. Stefan Volkwein
2. Referent: Prof. Dr. Bernard Haasdonk

# Abstract

In this work mathematical systems arising from the modeling of lithium ion batteries are investigated. These models are expressed in terms of highly nonlinear and coupled partial differential equations (PDEs) of different types. There are several parameters in the PDE system which are not known a-priori or which cannot be determined experimentally. Hence, efficient numerical algorithms to estimate unknown parameters are needed. For this purpose a parameter identification problem is formulated as a nonlinear least squares problem. To investigate the parameter depending behavior of the nonlinear system output a sensitivity analysis is carried out. By utilizing a subset selection method the relevant parameters for the optimization process are determined. To speed up the optimization algorithms a model reduction approach based on proper orthogonal decomposition (POD) is applied. Different techniques for the realization of the reduced order models and the parameter estimation are discussed. Numerical examples are presented to illustrate the efficiency of the proposed methods.

# Acknowledgement

First I would like to thank my advisor Prof. Dr. Stefan Volkwein for his guidance during the progress and development of this PhD thesis. All the fruitful discussions, the valuable insights and encouragements in both professional and personal aspects are highly appreciated. I also want to express my gratitude to Prof. Dr. Bernard Haasdonk for carefully reading my thesis and providing me with valuable comments and suggestions.

To all my colleagues, my sincerest thanks for the helpful and interesting discussions not only on mathematical issues but also for all the time spent together in and out of the university. I would also like to express my thanks to all the other members of the Fachbereich Mathematik und Statistik for a very pleasant working atmosphere.

This PhD thesis is dedicated to my parents. I would have not pursued my cherished mathematics career if not for them. Thanks for all the unceasing support, encouragement and unwavering belief in me. Special thanks to Michelle who greatly inspired and supported me during my PhD and made me feel like anything is possible. Furthermore, I would like to express my heartfelt gratitude to all the other people who contributed to the success of this PhD thesis with their professional or personal support.

I gratefully acknowledge the support by the German Science Fund *Numerical and Analytical Methods for Elliptic-Parabolic Systems Appearing in the Modeling of Lithium-Ion Batteries* (Excellence Initiative) and the support by the Zukunftsfonds Steiermark project *Prognosesicheres Batteriemodell für Elektro- und Hybridfahrzeuge*.

# Contents

# List of Figures

# List of Tables

# Introduction

In this work a nonlinear parameter estimation problem is investigated. The underlying equations are given by a nonlinear system of partial differential equations (PDEs). Efficient methods are developed to perform the parameter estimation. This involves the development of a reduced order model that can be solved at less computational expenses.

We consider an elliptic-parabolic PDE system consisting of two elliptic and one parabolic equation. Coupled multi component systems of this type can be viewed as generalizations of mathematical models for lithium ion batteries; see, e.g. [33, 72, 86]. The elliptic equations in the nonlinear system of PDEs model the potentials in liquid and solid phase and the parabolic equation the concentration of lithium ions. The three equations are coupled by a strong nonlinear term involving the hyperbolic sine, the square root and the logarithmic function. Additionally, the equations are coupled by a nonlinear diffusion coefficient. These coupling make the system hard to solve. Due to the nature of the nonlinearities the solvers for the nonlinear system have to be well adjusted. A finite element discretization will be utilized to solve the nonlinear PDE system numerically. Moreover different strategies to obtain stable solvers are presented.

The discretization of the nonlinear system of PDEs using the finite element techniques, lead to very large systems that are expensive to solve. The goal is to develop a reduced order model for the nonlinear system of PDEs that is cheap to evaluate. This is motivated by applications like parameter estimations, optimal control and design, where repeated evaluations of the nonlinear systems are required. Therefore, the spatial approximation is realized by the Galerkin scheme using proper orthogonal decomposition (POD); see, e.g. [50, 56, 77]. POD is used to generate a basis of a subspace that expresses the characteristics of the expected solution. This is in contrast to more general approximation methods, such as the finite element method, that do not correlate to the dynamics of the underlying system. The method of snapshots to obtain the POD basis is outlined and different computational strategies are presented.

Further, the POD basis generation under the assumption that several simulations of the nonlinear systems have already been performed is investigated. Two strategies to compute POD basis are presented. The standard POD approach is compared with an algorithm based on the greedy procedure. The goal is to extract the best possible POD basis from the given data. The strategy involving the greedy procedure generates the POD basis functions iteratively. Similar approaches are also used in reduced basis methods [44, 45, 70].

Additionally, we investigate a non-intrusive model order reduction technique. This allows the construction of reduced order models independent from the original discretization techniques. Hence, for the construction of the reduced order model black box solvers can

be used. This is different to the projection approach used e.g. in [13,62], where the reduced order models are obtained by projecting the original algebraic systems onto the subspaces spanned by the POD basis. The advantage of this method is outlined by applying it to an adaptive finite element solution.

To obtain an efficient reduced order model an empirical interpolation method (EIM) is applied [4]. This method is often used in the combination with the reduced basis approach; see, e.g. [39,65,70]. The EIM methods are very effective in reducing the complexity of the evaluation of the nonlinear terms. Numerical examples are presented to demonstrate the effectiveness of the EIM when applied to the introduced nonlinear system.

After obtaining an efficient reduced order model we want to utilize it in a parameter estimation problem. The nonlinear systems arising from modeling of lithium ion batteries contain a variety of parameters. Those parameters have to be identified in order to calibrate the model. The parameters can be correlated and not all are sensitive, hence a strategy has to be applied in order to identify the parameters best suited for the identification process. For this the sensitivities are computed and with the help of a subset selection method the sensitive parameters are extracted [6, 12, 31]. Here the sensitivity based characterisation of the identifiability is exploited [73]. The obtained nonlinear least squares problem is then solved by numerical methods for optimization problems [55, 68, 82].

When using a reduced order model in the optimization process an error is introduced. Therefore, an a-posteriori error estimator has to be developed in order to quantify the quality of the obtained solution. We here use results from [29, 53]. Further, it is important to understand that the obtained reduced order model obtained by the POD method is only a local approximation of the nonlinear system. Hence, it is necessary to guarantee that the approximation is good throughout the optimization process. For this we make use of a simplified error indicator known from the reduced basis strategies [39]. Using this error indicator an adaptive reduced order model strategy is proposed to solve the optimization problem in an efficient way. This is achieved by combining the finite element discretization with the reduced order model during the optimization process.

The work is organized in the following manner:

- In Chapter 1 the nonlinear elliptic-parabolic system is formulated. The discretization using the finite element method is outlined and a-priori error estimators are derived. The Newton method used to solve the nonlinear system is introduced and required modifications are described. The chapter is concluded with some numerical results using the proposed method.

- Chapter 2 is devoted to the development of the reduced order model. The POD method is introduced and the reduced order model is generated in the continuous and discrete setting. As in the finite element case a-priori error estimators are derived. To obtain efficient reduced order models the EIM is introduced and the application to the presented system is described. The computation of the POD basis is outlined and numerical examples utilizing the different techniques are presented. Additionally, different scenarios for the POD basis computation are investigated, such as the POD basis computation from multiple snapshot sets using a POD-Greedy method and the

POD basis computation in the case of adaptive finite element solutions. Lastly, the chapter is concluded by presenting numerical examples utilizing the reduced order model. The effectiveness of the approach is outlined and different error measures are presented.

- Chapter 3 introduces the parameter estimation problem. Basic optimality results are presented. The concept of sensitivity analysis is introduced and the subset selection method is outlined. A model problem is considered to demonstrate the theoretical results. Different approaches for the computation of first and second order derivatives are presented. Lastly, the adaptive optimization strategy is introduced by combining the results from Chapter 1 and Chapter 2. The computation of the a-posteriori error estimator is outlined. Numerical examples demonstrate the efficiency of the proposed strategy.

- Lastly, a conclusion is drawn to summarize the solved and open problems and give an outline of possible extensions.

# Chapter 1

# The nonlinear elliptic-parabolic system

In this chapter we present the coupled nonlinear system of elliptic-parabolic partial differential equations investigated in this work. After introducing some basic functional analysis tools the full nonlinear system is stated. Some properties and analytical results are outlined. Furthermore, the numerical approach to solve the system of equations using the finite element method is described and some numerical results are presented.

## 1.1 Basic functional analysis

Let us first have a look at the spaces of integrable functions, the Lebesgue spaces denoted by $L^p$. Let $\Omega \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$, denote an open and bounded domain with boundary $\Gamma = \partial\Omega$. Then the Lebesgue spaces $L^p(\Omega)$ are given by

$$L^p(\Omega) = \left\{ \varphi : \Omega \to \mathbb{R} \,\middle|\, \varphi \text{ is measurable and } \int_\Omega \left|\varphi(\mathbf{x})\right|^p \mathrm{d}\mathbf{x} < \infty \right\}$$

for $1 \leq p < \infty$. Together with the endowed $L^p$-norm given by

$$\|\varphi\|_{L^p(\Omega)} = \left( \int_\Omega |\varphi(\mathbf{x})|^p \,\mathrm{d}\mathbf{x} \right)^{\frac{1}{p}} \quad \text{for} \quad \varphi \in L^p(\Omega)$$

we get a Banach space. In particular, $L^2(\Omega)$ is the space of square integrable functions in $\Omega$. We endow $L^2(\Omega)$ with the inner product and the induced norm

$$\langle \varphi, \psi \rangle_{L^2(\Omega)} = \int_\Omega \varphi(\mathbf{x})\psi(\mathbf{x}) \,\mathrm{d}\mathbf{x} \quad \text{and} \quad \|\varphi\|_{L^2(\Omega)} = \sqrt{\langle \varphi, \varphi \rangle_{L^2(\Omega)}},$$

respectively. Hence, $L^2(\Omega)$ is a Hilbert space denoted by $H^0(\Omega)$. Further, the Lebesgue space for the case $p = \infty$ is given by

$$L^\infty(\Omega) = \left\{ \varphi : \Omega \to \mathbb{R} \,\middle|\, \varphi \text{ is measurable and } \operatorname*{ess\,sup}_{\mathbf{x}\in\Omega} \left|\varphi(\mathbf{x})\right| \right\}$$

with the endowed norm

$$\|\varphi\|_{L^\infty(\Omega)} = \operatorname*{ess\,sup}_{\mathbf{x}\in\Omega} |\varphi(\mathbf{x})|.$$

Let us recall the following inequalities which we will utilize later.

i. *Hölder's inequality*. Assume $1 \leq p, q \leq \infty, \frac{1}{p} + \frac{1}{q} = 1$. Then if $u \in L^p(\Omega), v \in L^q(\Omega)$, we have

$$\int_\Omega |uv| \mathrm{d}\mathbf{x} \leq \|u\|_{L^p(\Omega)} \|v\|_{L^q(\Omega)}.$$

ii. *Minkowski's inequality*. Assume $1 \leq p \leq \infty$ and $u, v \in L^p(\Omega)$. Then

$$\|u + v\|_{L^p(\Omega)} \leq \|u\|_{L^p(\Omega)} + \|v\|_{L^p(\Omega)}.$$

Next we introduce a subspace of $L^2(\Omega)$ that contains smoother functions. Let us recall the Sobolev space

$$W^{1,p}(\Omega) = \Big\{ \varphi \in L^p(\Omega) \text{ and } \int_\Omega |\nabla \varphi(\mathbf{x})|_2^p \, \mathrm{d}\mathbf{x} < \infty \Big\}, \quad 1 \leq p < \infty,$$

where $|\cdot|_2$ denotes the Euclidean norm in $\mathbb{R}^d$. In particular, for $p = 2$ we set $H^1(\Omega) = W^{1,2}(\Omega)$. The space $H^1$ is a Hilbert space supplied with the inner product

$$\begin{aligned} \langle \varphi, \psi \rangle_{H^1(\Omega)} &= \int_\Omega \varphi(\mathbf{x}) \psi(\mathbf{x}) + \nabla \varphi(\mathbf{x}) \cdot \nabla \psi(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ &= \langle \varphi, \psi \rangle_{L^2(\Omega)} + \langle \nabla \varphi, \nabla \psi \rangle_{L^2(\Omega)^d} \quad \text{for} \quad \varphi, \psi \in H^1(\Omega) \end{aligned}$$

and the induced norm $\|\varphi\|_{H^1(\Omega)} = \sqrt{\langle \varphi, \varphi \rangle_{H^1(\Omega)}}$ for $\varphi \in H^1(\Omega)$. Additionally, we introduce the Hölder space as

$$C^{0,\beta}(\overline{\Omega}) = \left\{ \varphi \in C(\overline{\Omega}) \text{ and } \sup_{\substack{x,y \in \Omega \\ x \neq y}} \frac{|\varphi(x) - \varphi(y)|}{|x - y|^\beta} < \infty \right\}$$

for $\beta \in (0, 1]$. We refer the reader, e.g. to [1,30] for more details on Lebesgue and Sobolev spaces. Next we introduce the dual space.

**Definition 1.1.** *Let $X$ and $Y$ be real normed linear spaces. Then*

1. *A mapping $A : X \to Y$ is a* linear operator *provided*

$$A[\lambda u + \mu v] = \lambda A u + \mu A v$$

*for all $u, v \in X$ and $\lambda, \mu \in \mathbb{R}$.*

2. *A linear operator $A : X \to Y$ is* bounded *if*

$$\|A\| := \sup\{\|Au\|_Y \mid \|u\|_X \leq 1\} < \infty.$$

3. *A bounded linear operator $A : X \to \mathbb{R}$ is called a* bounded linear functional *on $X$.*

4. *Let $X^*$ be the collection of all bounded linear functionals on $X$. Then $X^*$ is called the* dual space *of $X$.*

The dual pairing between the dual space $X^*$ and $X$ will be denoted by $\langle \cdot, \cdot \rangle_{X^*,X}$. We introduce the *dual norm* as in [20, 30, 40] by

$$\|g\|_{X^*} := \sup_{v \in X \setminus \{0\}} \frac{g(v)}{\|v\|_X} \quad \text{for} \quad g \in X^* \quad \text{and} \quad v \in X.$$

For completeness we state the Riesz representation theorem.

**Theorem 1.2** (Riesz Representation Theorem)**.** *Let $X$ be a Hilbert space. For each $g(v) \in X^*$ there exists a unique element $v_g \in X$ such that*

$$g(v) = \langle v_g, v \rangle_X$$

*holds for all $v \in X$. The mapping $g(v) \mapsto v_g$ is a linear isomorphism from $X^*$ onto $X$. Moreover one has $\|g\|_{X^*} = \|v_g\|_X$. Furthermore we have $(X^*)^* = X$.*

In the numerical part this theorem can be used to compute the dual norms.

## 1.2 The model

In this section we formulate the nonlinear elliptic-parabolic system. Suppose that $\Omega = (a, b) \subset \mathbb{R}$, $a < b$, is the spatial domain with boundary $\Gamma = \{a, b\}$. We set $H = L^2(\Omega)$, $V = H^1(\Omega)$ and

$$V_a = \{\varphi \in H^1(\Omega) \,|\, \varphi(a) = 0\}.$$

For the terminal time $T > 0$ let $Q = (0, T) \times \Omega$ and $\Sigma = (0, T) \times \Gamma$. The space $L^2(0, T; V)$ stands for the space (equivalence classes) of measurable abstract functions $\varphi : [0, T] \to V$, which are square integrable, i.e.

$$\int_0^T \|\varphi(t)\|_V^2 \, \mathrm{d}t < \infty.$$

When $t$ is fixed, the expression $\varphi(t)$ stands for the function $\varphi(t, \cdot)$ considered as a function in $\Omega$ only. Recall that

$$W(0, T; V) = \{\varphi \in L^2(0, T; V) \,|\, \varphi_t \in L^2(0, T; V)\}$$

is a Hilbert space supplied with its corresponding inner product; see [2, 21].

After introducing the required functional settings let us state the nonlinear elliptic-parabolic system under investigation. First we introduce the admissible set $\mathcal{M}_{ad}$ for the parameter $\mu$ by

$$\mathcal{M}_{ad} = \left\{(\mu_1, \mu_2, \mu_3, \mu_4) \in \mathbb{R}^4 \,\middle|\, 1 < \mu_1 \leq 3/2,\ \mu_2 < 0, \mu_3 < 0 \text{ and } \underline{\mu}_4 \leq \mu_4 \leq \overline{\mu}_4\right\}$$

with $\underline{\mu}_4, \overline{\mu}_4 \in \mathbb{R}$ and $\underline{\mu}_4 \leq \overline{\mu}_4$. For a given parameter $\mu = (\mu_1, \mu_2, \mu_3, \mu_4) \in \mathcal{M}_{ad}$ the triple $(y, p, q) : Q \to \mathbb{R}$ satisfies the nonlinear system

$$y_t(t, \mathbf{x}) - \big(c_1(\mathbf{x}) y_{\mathbf{x}}(t, \mathbf{x})\big)_{\mathbf{x}} + \mathcal{N}(\mathbf{x}, y(t, \mathbf{x}), p(t, \mathbf{x}), q(t, \mathbf{x}); \mu) = 0, \tag{1.1a}$$

$$-\big(c_2(y(t, \mathbf{x}); \mu) p_{\mathbf{x}}(t, \mathbf{x})\big)_{\mathbf{x}} + \mathcal{N}(\mathbf{x}, y(t, \mathbf{x}), p(t, \mathbf{x}), q(t, \mathbf{x}); \mu) = 0, \tag{1.1b}$$

$$-\big(c_3(\mathbf{x}) q_{\mathbf{x}}(t, \mathbf{x})\big)_{\mathbf{x}} - \mathcal{N}(\mathbf{x}, y(t, \mathbf{x}), p(t, \mathbf{x}), q(t, \mathbf{x}); \mu) = 0 \tag{1.1c}$$

for almost all (f.a.a.) $(t, \mathbf{x}) \in Q$ together with the homogeneous Neumann boundary conditions for $y$ and $p$

$$y_{\mathbf{x}}(t,a) = y_{\mathbf{x}}(t,b) = p_{\mathbf{x}}(t,a) = p_{\mathbf{x}}(t,b) = 0, \tag{1.1d}$$

Dirichlet-Neumann condition for the variable $q$

$$q(t,a) = 0 \quad \text{and} \quad q_{\mathbf{x}}(t,b) = \mathcal{I}(t) \tag{1.1e}$$

f.a.a. $t \in (0,T)$ and the initial condition

$$y(0,\mathbf{x}) = y_\circ(\mathbf{x}) \tag{1.1f}$$

f.a.a. $\mathbf{x} \in \Omega$ with $y_\circ : \Omega \to \mathbb{R}$ positive and bounded. The diffusion coefficients $c_1$ and $c_3$ are supposed to be piecewise constant and positive. Further, the diffusion coefficient $c_2$ is continuously differentiable, positive and nonlinearly dependent on the variable $y$ and the parameter $\mu_4$. Later, in the numerical experiments a polynomial in $y$ and $\mu_4$ is considered. Let us introduce the variable $z = (y,p,q) \in \mathcal{Z}_{ad}$ and the corresponding admissible set

$$\mathcal{Z}_{ad} = \big\{(y,p,q) \in \mathbb{R}^3 \,\big|\, y \geq y_{\min}\big\}.$$

The nonlinear coupling term $\mathcal{N} : \Omega \times \mathcal{Z}_{ad} \times \mathcal{M}_{ad} \to \mathbb{R}$ introduced in the model (1.1) is given by

$$\mathcal{N}(\mathbf{x}, z; \mu) = \chi(\mathbf{x}; \mu)\sqrt{y}\sinh\big(\mu_1(q-p) - \ln y\big), \tag{1.2a}$$

where $\chi$ is an indicator function of the form

$$\chi(\mathbf{x};\mu) = \begin{cases} \mu_2 & \text{for} \quad \mathbf{x} \in [a, s_1] \\ 0 & \text{for} \quad \mathbf{x} \in (s_1, s_2) \\ \mu_3 & \text{for} \quad \mathbf{x} \in [s_2, b] \end{cases} \tag{1.2b}$$

with $a < s_1 \leq s_2 < b$. With these choices (1.1) can be seen as a generalization of a mathematical model for lithium ion batteries; see, e.g. [33, 72, 86].

**Remark 1.3.** *1. The positivity of the $y$ component is needed to evaluate the terms $\sqrt{y}$ and $\ln y$ in the nonlinearity.*

*2. Notice that $\mathcal{N}(\mathbf{x}, \cdot\,; \mu) : \mathcal{Z}_{ad} \to \mathbb{R}$ is continuously differentiable for any parameter $\mu \in \mathcal{M}_{ad}$. Furthermore, we get*

$$\frac{\partial \mathcal{N}}{\partial p}(\mathbf{x}, z; \mu) = -\mu_1 \chi(\mathbf{x}, \mu)\sqrt{y}\cosh(\mu_1(q-p) - \ln y) = -\frac{\partial \mathcal{N}}{\partial q}(\mathbf{x}, z; \mu) > 0. \tag{1.3}$$

Let us next look at the weak formulation of (1.1). For this we multiply (1.1) by test-functions $\varphi \in V$ and $\varphi \in V_a$, respectively. Then integrating by parts the weak formulation is given by

$$\int_\Omega y_t(t)\varphi + c_1 y_{\mathbf{x}}(t)\varphi' + \mathcal{N}(\mathbf{x}, y(t), p(t), q(t); \mu)\varphi \,\mathrm{d}\mathbf{x} = 0 \quad \text{for all } \varphi \in V, \tag{1.4a}$$

$$\int_\Omega c_2(y(t);\mu)p_{\mathbf{x}}(t)\varphi' + \mathcal{N}(\mathbf{x}, y(t), p(t), q(t); \mu)\varphi \,\mathrm{d}\mathbf{x} = 0 \quad \text{for all } \varphi \in V, \tag{1.4b}$$

$$\int_\Omega c_3 q_{\mathbf{x}}(t)\varphi' - \mathcal{N}(\mathbf{x}, y(t), p(t), q(t); \mu)\varphi \,\mathrm{d}\mathbf{x} - \int_\Gamma \mathcal{I}(t)\varphi \,\mathrm{d}S = 0 \quad \text{for all } \varphi \in V_a. \tag{1.4c}$$

The triple $(y, p, q)$ solving (1.4) is called a *weak solution* to (1.1) if $z = (y, p, q) \in Z$, $y(0) = y_\circ$ in $H$ with

$$Z = \big(W(0,T;V) \times L^2(0,T;V) \times L^2(0,T;V_a)\big) \cap L^\infty(Q)^3.$$

In the following theorem we state the analytical results about existence and uniqueness of weak solutions to the nonlinear system (1.1).

**Theorem 1.4.** *Let $\mu \in \mathcal{M}_{ad}$, $y_\circ \geq y_{\min} > 0$ and $y_\circ \in C^{0,\beta}(\bar{\Omega})$ for some $\beta > 0$. We assume there exist $k_1 \leq 3/2 + \mu_1$ and $k_2 > 0$ such that*

$$y^{k_1} \leq c_2(y;\mu) \leq k_2 y^{3/2+\mu_1}, \quad y \leq 1.$$

*Then there exists a unique global weak solution to the system given by* (1.1) *with* (1.2) *for all $T > 0$. Furthermore, there exists a constant $C > 0$ such that*

$$\|p\|_{L^\infty(\Omega)} \leq Ce^{Ct}, \quad \|q\|_{L^\infty(\Omega)} \leq Ce^{Ct} \quad \textit{and} \quad y_{\min} = \frac{1}{C} \leq y(t,\mathbf{x}) \leq C$$

*holds.*

*Sketch of Proof.* The proof to this theorem is presented in [86] for the case of all homogenous Neumann boundary conditions for the variable $q$, i.e. $q_{\mathbf{x}}(t,a) = q_{\mathbf{x}}(t,b) = 0$ together with

$$\int_\Omega q(t,\mathbf{x})\,\mathrm{d}\mathbf{x} = 0.$$

The introduced constraints on the parameters $\mu_1$, $\mu_2$ and $\mu_3$ in the admissible set $\mathcal{M}_{ad}$ are essential for the proof. By using the results from [75] the proof can be extended to the case of the boundary conditions presented in (1.1e). □

Using this result we get that the nonlinear solution operator $\mathcal{S} : \mathcal{M}_{ad} \to Z$ is well-defined, where $z = \mathcal{S}(\mu)$ is the weak solution to (1.1) for the parameter value $\mu \in \mathcal{M}_{ad}$.

## 1.3 The Galerkin approximation and discretization of the nonlinear system

Let us next introduce the Galerkin scheme for (1.1). To discretize the nonlinear system (1.1) in space we use a Galerkin scheme. For this we introduce the finite dimensional subspaces

$$V^{g,N_{\mathbf{x}}} = \text{span}\{\varphi_1^g, \ldots, \varphi_{dof}^g\} \subset V$$

with $g = \{y, p\}$, where the $\varphi_i^g$'s denote the $dof$ basis functions that are used to approximate the space $V$. Analogously we proceed with $V_a^{q,N_{\mathbf{x}}} \subset V_a$. At this point we do not make a restriction of what type of basis functions are being used. With the superindex $g$ we emphasize the possibility of using different types of basis functions for each variable. In this chapter we will use the same basis functions for each variable and hence omit the

superindex $g$ from now on. We proceed by introducing the standard Galerkin ansatz of the form

$$y^N(t,\mathbf{x}) = \sum_{i=1}^{dof} \mathrm{y}_i^N(t)\varphi_i(\mathbf{x}), \quad p^N(t,\mathbf{x}) = \sum_{i=1}^{dof} \mathrm{p}_i^N(t)\varphi_i(\mathbf{x}), \quad \varphi \in V^{N_\mathbf{x}}, \tag{1.5a}$$

and

$$q^N(t,\mathbf{x}) = \sum_{i=1}^{dof} \mathrm{q}_i^N(t)\varphi_i(\mathbf{x}), \quad \varphi \in V_a^{N_\mathbf{x}}. \tag{1.5b}$$

Here $\mathrm{y}_i^N$, $\mathrm{p}_i^N$ and $\mathrm{q}_i^N$ denote the time-dependent Galerkin coefficients corresponding to the variables $y$, $p$ and $q$. Inserting the Galerkin ansatz into the weak form (1.4) we get for $1 \le i \le dof$ the system

$$\int_\Omega y_t^N(t)\varphi_i + c_1 y_\mathbf{x}^N(t)\varphi_i' + \mathcal{N}(\mathbf{x}, y^N(t), p^N(t), q^N(t); \mu)\varphi_i \,\mathrm{d}\mathbf{x} = 0, \tag{1.6a}$$

$$\int_\Omega c_2(y^N(t);\mu) p_\mathbf{x}^N(t)\varphi_i' + \mathcal{N}(\mathbf{x}, y^N(t), p^N(t), q^N(t); \mu)\varphi_i \,\mathrm{d}\mathbf{x} = 0, \tag{1.6b}$$

$$\int_\Omega c_3 q_\mathbf{x}^N(t)\varphi_i' - \mathcal{N}(\mathbf{x}, y^N(t), p^N(t), q^N(t); \mu)\varphi_i \,\mathrm{d}\mathbf{x} - \int_\Gamma \mathcal{I}(t)\varphi_i \,\mathrm{d}S = 0, \tag{1.6c}$$

with $\varphi_i \in V^{N_\mathbf{x}}$ for (1.6a)-(1.6b) and $\varphi_i \in V_a^{N_\mathbf{x}}$ for (1.6c). Note that here also the test functions $\varphi_i$ are chosen from the same space as the ansatz functions. There are also techniques to choose them from different spaces which then results in a Petrov-Galerkin ansatz. Utilizing the structure of the Galerkin ansatz for $y^N$, $p^N$ and $q^N$ the obtained system can be written in matrix-vector form. For this we introduce

$$\left(\left(\mathrm{M}_f^N\right)\right)_{ij} = \int_\Omega f(\mathbf{x})\varphi_j\varphi_i \,\mathrm{d}\mathbf{x}, \qquad \text{(mass matrix)} \tag{1.7a}$$

$$\left(\left(\mathrm{S}_f^N\right)\right)_{ij} = \int_\Omega f(\mathbf{x})\varphi_j'\varphi_i' \,\mathrm{d}\mathbf{x}, \qquad \text{(stiffness matrix)} \tag{1.7b}$$

$$\left(\mathcal{N}^N(\mathrm{y}^N(t), \mathrm{p}^N(t), \mathrm{q}^N(t); \mu)\right)_i = \int_\Omega \mathcal{N}(\mathbf{x}, y^N(t), p^N(t), q^N(t); \mu)\varphi_i \,\mathrm{d}\mathbf{x}, \tag{1.7c}$$

$$\left(\mathcal{I}^N(t)\right)_i = \int_\Gamma \mathcal{I}(t)\varphi_i \,\mathrm{d}S, \tag{1.7d}$$

$$\left(\mathcal{Y}_0^N\right)_i = \int_\Omega y_\circ(\mathbf{x})\varphi_i \,\mathrm{d}S, \tag{1.7e}$$

for $1 \le i, j \le dof$, $t \in [0,T]$, where $f$ denotes a positive weight function and

$$\mathrm{y}^N(t) = \left(\mathrm{y}_i^N(t)\right)_{1\le i\le dof}, \quad \mathrm{p}^N(t) = \left(\mathrm{p}_i^N(t)\right)_{1\le i\le dof}, \quad \mathrm{q}^N(t) = \left(\mathrm{q}_i^N(t)\right)_{1\le i\le dof}.$$

Hence, the weak form (1.4) can be written in semi-discrete form using matrix vector notation together with the initial condition as

$$\mathrm{M}_1^N \mathrm{y}_t^N(t) + \mathrm{S}_{c_1}^N \mathrm{y}^N(t) + \mathcal{N}^N(\mathrm{y}^N(t), \mathrm{p}^N(t), \mathrm{q}^N(t); \mu) = 0 \qquad \text{f.a.a. } t \in [0,T], \quad (1.8a)$$

$$\mathrm{M}_1^N \mathrm{y}^N(0) = \mathcal{Y}_0^N, \quad (1.8b)$$

$$\mathrm{S}_{c_2(\mathrm{y}^N(t);\mu)}^N \mathrm{p}^N(t) + \mathcal{N}^N(\mathrm{y}^N(t), \mathrm{p}^N(t), \mathrm{q}^N(t); \mu) = 0 \qquad \text{f.a.a. } t \in [0,T], \quad (1.8c)$$

$$\mathrm{S}_{c_3}^N \mathrm{q}^N(t) - \mathcal{N}^N(\mathrm{y}^N(t), \mathrm{p}^N(t), \mathrm{q}^N(t); \mu) = \mathcal{I}^N(t) \quad \text{f.a.a. } t \in [0,T]. \quad (1.8d)$$

Note that (1.8) is a nonlinear semi-discrete system since only the space domain has been discretized. Furthermore, the dimension of the matrices $\mathrm{M}^N$ and $\mathrm{S}^N$ is $dof \times dof$, i.e. $\mathrm{M}^N, \mathrm{S}^N \in \mathbb{R}^{dof \times dof}$. Consequently the semi-discrete nonlinear system (1.8) is of dimension $3\,dof$.

### 1.3.1 Discretization of the nonlinear system

Starting from (1.8) we introduce the discretization for (1.1). In the following we use results from [7, 10, 34, 40, 60, 67]. We will utilize the finite element method for the spatial discretization. First let us discretize the domain $\Omega = (a, b)$ with $N_\mathbf{x}$ points. The discretization is given by $\{\bar{\mathbf{x}}_i\}_{i=1}^{N_\mathbf{x}}$ with

$$a = \bar{\mathbf{x}}_1 < \bar{\mathbf{x}}_2 < \ldots < \bar{\mathbf{x}}_{N_\mathbf{x}-1} < \bar{\mathbf{x}}_{N_\mathbf{x}} = b.$$

Further, we decompose $\Omega$ into the subdomains $\Omega_i = (\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_{i+1})$ for $i = 1, \ldots, N_\mathbf{x} - 1$. In this work we will use piecewise quadratic ansatz functions. Therefore, let us shortly introduce the finite element approximation for this case. For the quadratic ansatz functions we consider the Galerkin ansatz of the form

$$y^N(t, \mathbf{x}) = \sum_{i=1}^{N_\mathbf{x}} \mathrm{y}_i^N \bar{\varphi}_i(\mathbf{x}) + \sum_{i=1}^{N_\mathbf{x}-1} \mathrm{y}_{i+1/2}^N \bar{\varphi}_{i-1/2}(\mathbf{x}) \quad (1.9)$$

with the global continuous basis functions $\bar{\varphi}_i$ and $\bar{\varphi}_{i+1/2}$. Analogously, the Galerkin ansatz is given for $p^N$ and $q^N$. The piecewise quadratic and globally continuous basis functions have to satisfy the following conditions:

1. $\bar{\varphi}_i(\mathbf{x})|_{\Omega_i}$, $\bar{\varphi}_i(\mathbf{x})|_{\Omega_{i+1}}$ and $\bar{\varphi}_{i+1/2}|_{\Omega_i}$ are quadratic polynomials,
2. $\bar{\varphi}_i(\bar{\mathbf{x}}_k) = \delta_{jk}$ and $\bar{\varphi}_i(\bar{\mathbf{x}}_{k+1/2}) = 0$,
3. $\bar{\varphi}_{i+1/2}(\bar{\mathbf{x}}_k) = 0$ and $\bar{\varphi}_{i+1/2}(\bar{\mathbf{x}}_{k+1/2}) = \delta_{jk}$,

where the $\bar{\mathbf{x}}_{i+1/2} := \frac{1}{2}(\bar{\mathbf{x}}_i + \bar{\mathbf{x}}_{i+1})$ denote the midpoints of the subintervals $\Omega_i$ for $i = 1, \ldots, N_\mathbf{x} - 1$ and $\delta_{jk}$ stands for the Kronecker symbol, i.e. $\delta_{jk} = 0$ for $k \neq j$ and $\delta_{jj} = 1$. Following these conditions we can write down the globally continuous piecewise quadratic basis functions explicitly as

$$\bar{\varphi}_i(\mathbf{x}) = \begin{cases} \frac{2(\mathbf{x}-\bar{\mathbf{x}}_i)(\mathbf{x}-\bar{\mathbf{x}}_{i+1/2})}{(\bar{\mathbf{x}}_{i+1}-\bar{\mathbf{x}}_i)^2} & \text{for } \mathbf{x} \in \overline{\Omega}_i, \\ \frac{2(\bar{\mathbf{x}}_{i+2}-\mathbf{x})(\bar{\mathbf{x}}_{i+1/2}-\mathbf{x})}{(\bar{\mathbf{x}}_{i+2}-\bar{\mathbf{x}}_{i+1})^2} & \text{for } \mathbf{x} \in \Omega_{i+1}, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\bar{\varphi}_{i-1/2}(\mathbf{x}) = \begin{cases} \frac{4(\mathbf{x}-\bar{\mathbf{x}}_i)(\bar{\mathbf{x}}_{i+1}-\mathbf{x})}{(\bar{\mathbf{x}}_{i+1}-\bar{\mathbf{x}}_i)^2} & \text{for } \mathbf{x} \in \Omega_i, \\ 0 & \text{otherwise.} \end{cases}$$

A basis of this type is called a *nodal basis*. Let us next transform the basis back to the notation used for the Galerkin ansatz (1.5). For this we perform a reindexing of the form

$$\varphi_{2i-1} = \bar{\varphi}_i \quad \text{and} \quad \varphi_{2i} = \bar{\varphi}_{i+1/2},$$

and hence obtain the finite dimensional finite element subspace

$$V^{N_\mathbf{x}} = \text{span}\{\varphi_1, \ldots, \varphi_{dof}\} \subset V$$

with $dof = 2N_\mathbf{x} - 1$. We also apply this reindexing to the discretization of the domain $\Omega$, i.e.

$$\mathbf{x}_{2i-1} = \bar{\mathbf{x}}_i \quad \text{and} \quad \mathbf{x}_{2i} = \bar{\mathbf{x}}_{i+1/2}.$$

Further, we set $h_i = |\Omega_i|$ the length of the interval $\Omega_i$ and $h = \max_{1\le i\le N_\mathbf{x}} |\Omega_i|$, where $\Omega_i = (\mathbf{x}_{2i-1}, \mathbf{x}_{2i+1})$ for $i = 1, \ldots, N_\mathbf{x} - 1$.

**Remark 1.5.** *When using globally continuous piecewise linear ansatz functions we obtain the classical hat functions. In this case $dof \equiv N_\mathbf{x}$ and no reindexing is needed.*

Let us have a look at the error we introduce by applying the finite element discretization. We impose the following assumption.

**Assumption 1.6.** *Suppose that for any $\mu \in \mathcal{M}_{ad}$ the system* (1.8) *admits a unique solution satisfying*

$$y^N(t) = \sum_{i=1}^{dof} \mathrm{y}_i^N(t)\varphi_i(\mathbf{x}) \ge y_{\min} \quad \textit{f.a.a.} \quad (t, \mathbf{x}) \in Q.$$

*Further, there exists a constant $C > 0$ independent of $h$ such that the following estimates hold:*

$$\|p^N(t)\|_{L^\infty(\Omega)} \le Ce^{Ct}, \quad \|q^N(t)\|_{L^\infty(\Omega)} \le Ce^{Ct} \quad \textit{and} \quad y_{\min} = \frac{1}{C} \le y^N(t) \le C.$$

This assumption should in general be met if the discretization is of good quality and represents the system (1.4) well. Next we want to develop an a-priori error estimator for the discrete nonlinear system. The goal is to estimate the error

$$\int_0^T \|y(t) - y^h(t)\|_V^2 + \|p(t) - p^h(t)\|_V^2 + \|q(t) - q^h(t)\|_V^2 \,\mathrm{d}t.$$

**Remark 1.7.** *The proof of the a-priori error estimate works also for $\Omega \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$. For this reason we use a notation which is appropriate also for the three-dimensional case.*

To introduce appropriate projections we define the symmetric, bounded, coercive bilinear forms for $y$ and $q$ as

$$\begin{aligned} a^y(\varphi,\psi) &= \int_\Omega c_1 \nabla\varphi\cdot\nabla\psi + \varphi\psi \,\mathrm{d}\mathbf{x} \quad \text{for} \quad \varphi,\psi\in V,\\ a^q(\varphi,\psi) &= \int_\Omega c_3 \nabla\varphi\cdot\nabla\psi \mathrm{d}\mathbf{x} \qquad\quad \text{for} \quad \varphi,\psi\in V_a. \end{aligned} \tag{1.10a}$$

Recall that

$$\langle\varphi,\psi\rangle_{V_a} = \int_\Omega \nabla\varphi\cdot\nabla\phi \,\mathrm{d}\mathbf{x} \quad \text{for} \quad \varphi,\psi\in V_a$$

is an inner product on $V_a$. Moreover, the piecewise constant coefficient functions $c_1$ and $c_3$ are strictly positive. Hence, there exists a constant $c_V \geq 0$ such that

$$a^y(\varphi,\varphi)\geq c_V\|\varphi\|_V^2 \quad \text{and} \quad a^q(\varphi,\varphi)\geq c_V\|\varphi\|_V^2 \tag{1.10b}$$

holds. Additionally, we introduce the parametrized, symmetric, bounded, coercive bilinear form for $p$ as

$$a^p(\varphi,\psi;y,\mu) = \int_\Omega c_2(y;\mu)\nabla\varphi\cdot\nabla\psi+\varphi\psi\,\mathrm{d}\mathbf{x} \quad \text{for} \quad \varphi,\psi\in V, \tag{1.10c}$$

where $y$ and $\mu$ are considered as parameters for the nonlinear diffusion coefficient. Further, the coefficient $c_2(y;\mu)$ is strictly positive. Since $y$ and $\mu_4$ are bounded we get that $\underline{c}\leq c_2(\,\cdot\,;\,\cdot\,)\leq\overline{c}$ holds. Then there exists a constant $c_V\geq 0$ independent of $y$ and $\mu$ such that

$$a^p(\varphi,\varphi;\,\cdot\,)\geq c_V\|\varphi\|_V^2 \tag{1.10d}$$

holds. Let us define the projections $\mathcal{P}^{N_\mathbf{x},y}: V\to V^{N_\mathbf{x}}$ and $\mathcal{P}^{N_\mathbf{x},p}: V\to V^{N_\mathbf{x}}$ by

$$a^y(\mathcal{P}^{N_\mathbf{x},y}\varphi,\psi) = a^y(\varphi,\psi) \quad \text{and} \quad a^p(\mathcal{P}^{N_\mathbf{x},p}\varphi,\psi;\,\cdot\,) = a^p(\varphi,\psi;\,\cdot\,) \tag{1.11a}$$

for all $\psi\in V^{N_\mathbf{x}}$. Analogously, we introduce the mapping $\mathcal{P}_a^{N_\mathbf{x},q}: V_a\to V_a^{N_\mathbf{x}}$ by

$$a^q(\mathcal{P}_a^{N_\mathbf{x},q}\varphi,\psi) = a^q(\varphi,\psi) \quad \text{for all } \psi\in V_a^{N_\mathbf{x}}. \tag{1.11b}$$

Using the projection the error for the component $y$ is decomposed as

$$y(t)-y^N(t) = y(t)-\mathcal{P}^{N_\mathbf{x},y}y(t)+\mathcal{P}^{N_\mathbf{x},y}y(t)-y^N(t) = \varrho^{N_\mathbf{x},y}(t)+\vartheta^{N_\mathbf{x},y}(t). \tag{1.12a}$$

where $\varrho^{N_\mathbf{x},y}(t) = y(t)-\mathcal{P}^{N_\mathbf{x},y}y(t)$ and $\vartheta^{N_\mathbf{x},y}(t) = \mathcal{P}^{N_\mathbf{x},y}y(t)-y^N(t)$. Analogously, let

$$p(t)-p^N(t) = p(t)-\mathcal{P}^{N_\mathbf{x},p}p(t)+\mathcal{P}^{N_\mathbf{x},p}p(t)-p^N(t) = \varrho^{N_\mathbf{x},p}(t)+\vartheta^{N_\mathbf{x},p}(t), \tag{1.12b}$$

$$q(t)-q^N(t) = q(t)-\mathcal{P}_a^{N_\mathbf{x},q}q(t)+\mathcal{P}_a^{N_\mathbf{x},q}q(t)-q^N(t) = \varrho^{N_\mathbf{x},q}(t)+\vartheta^{N_\mathbf{x},q}(t). \tag{1.12c}$$

Note that the projections $\mathcal{P}^{N_\mathbf{x},y}$ and $\mathcal{P}^{N_\mathbf{x},q}$ are Ritz projections. First we tackle the approximation error given by the $\varrho$-terms. We can now formulate the following lemma:

**Lemma 1.8.** *Suppose that $y \in W(0,T;H^{1+s}(\Omega))$ and $p,\, q \in L^2(0,T;H^{1+s}(\Omega))$. Then the approximation error terms $\varrho^{N_\mathbf{x},y}$, $\varrho^{N_\mathbf{x},p}$ and $\varrho^{N_\mathbf{x},q}$ satisfy*

$$\int_0^T \|\varrho^{N_\mathbf{x},y}(t)\|_V^2 \,\mathrm{d}t = \int_0^T \|y(t) - \mathcal{P}^{N_\mathbf{x},y} y(t)\|_V^2 \,\mathrm{d}t \le Ch^s \|y\|^2_{L^2(0,T;H^{1+s}(\Omega))},$$
$$\int_0^T \|\varrho^{N_\mathbf{x},q}(t)\|_V^2 \,\mathrm{d}t = \int_0^T \|q(t) - \mathcal{P}_a^{N_\mathbf{x},q} q(t)\|_V^2 \,\mathrm{d}t \le Ch^s \|q\|^2_{L^2(0,T;H^{1+s}(\Omega))}$$

*with $s = 1$ for piecewise linear and $s = 2$ for piecewise quadratic finite element spaces and $C$ independent of $h$, $t \in [0,T]$ and the variable $y$, $p$ and $q$. Additionally, we get*

$$\int_0^T \|\varrho_t^{N_\mathbf{x},y}(t)\|_V^2 \,\mathrm{d}t = \int_0^T \|y_t(t) - \mathcal{P}^{N_\mathbf{x},y} y_t(t)\|_V^2 \,\mathrm{d}t \le Ch^s \|y_t\|^2_{L^2(0,T;H^{1+s}(\Omega))}.$$

*Further, for $0 \le \underline{c} \le c_2(y;\mu) \le \overline{c}$ there exists a constant $C_1$ depending on $\underline{c}, \overline{c}$ and independent of $h$, $t \in [0,T]$ and the variable $y$, $p$ and $q$ such that*

$$\int_0^T \|\varrho^{N_\mathbf{x},p}(t)\|_V^2 \,\mathrm{d}t = \int_0^T \|p(t) - \mathcal{P}^{N_\mathbf{x},p} p(t)\|_V^2 \,\mathrm{d}t \le C_1(\underline{c},\overline{c}) h^s \|p\|^2_{L^2(0,T;H^{1+s}(\Omega))}$$

*holds.*

*Proof.* For the proof to this lemma see, e.g. [7, 10, 19, 80] and the references therein. □

In the following lemma we present an error estimate for the discretization error terms $\vartheta^{N_\mathbf{x},p}(t)$ and $\vartheta^{N_\mathbf{x},q}(t)$. The proof is based on (1.4), (1.6) and the properties of the projections $\mathcal{P}^{N_\mathbf{x},p}$ and $\mathcal{P}_a^{N_\mathbf{x},q}$.

**Lemma 1.9.** *Let $z$ and $z^N$ be the solution to* (1.1) *and* (1.8)*, respectively and let Assumption 1.6 be satisfied. Moreover, suppose that $y \in W(0,T;H^{1+s}(\Omega))$ and $p,\, q \in L^2(0,T;H^{1+s}(\Omega))$. Then there exists a constant $C > 0$ independent of $h$ and $t \in [0,T]$ such that*

$$\begin{aligned} &\|\vartheta^{N_\mathbf{x},p}(t)\|_V^2 + \|\vartheta^{N_\mathbf{x},q}(t)\|_V^2 \\ &\qquad \le C\left(\|\varrho^{N_\mathbf{x},y}(t)\|_V^2 + \|\varrho^{N_\mathbf{x},p}(t)\|_V^2 + \|\varrho^{N_\mathbf{x},q}(t)\|_V^2 + \|\vartheta^{N_\mathbf{x},y}(t)\|_H^2\right) \end{aligned} \tag{1.13}$$

*holds f.a.a. $t \in [0,T]$.*

*Proof.* Recall that the mapping $\mathcal{N}$ is continuously differentiable on $\mathcal{Z}_{ad} \times \mathcal{M}_{ad}$. Applying (1.12b), the property of the mapping $\mathcal{P}^{N_\mathbf{x},p}$, (1.4b), (1.6b) and the integral form of the mean

value theorem we obtain that

$$
\begin{aligned}
a^p(\vartheta^{N_{\mathbf{x}},p}(t),\varphi;y^N(t),\mu) &= a^p(\mathcal{P}^{N_{\mathbf{x}},p}p(t),\varphi;y^N(t),\mu) - a^p(p^N(t),\varphi;y^N(t),\mu)\\
&= a^p(p(t),\varphi;y^N(t),\mu) - a^p(p^N(t),\varphi;y^N(t),\mu)\\
&= \langle \mathcal{N}(\,\cdot\,,z^N(t);\mu),\varphi\rangle_H + \langle p(t)-p^N(t),\varphi\rangle_H + \int_\Omega c_2(y^N(t);\mu)\nabla p\nabla\varphi\,\mathrm{d}\mathbf{x}\\
&= \langle \mathcal{N}(\,\cdot\,,z^N(t);\mu),\varphi\rangle_H + \langle p(t)-p^N(t),\varphi\rangle_H + \int_\Omega c_2(y(t);\mu)\nabla p\nabla\varphi\,\mathrm{d}\mathbf{x}\\
&\quad + \int_\Omega (c_2(y^N(t);\mu)-c_2(y(t);\mu))\nabla p\nabla\varphi\,\mathrm{d}\mathbf{x}\\
&= \langle \mathcal{N}(\,\cdot\,,z^N(t);\mu)-\mathcal{N}(\,\cdot\,,z(t);\mu),\varphi\rangle_H + \langle p(t)-p^N(t),\varphi\rangle_H\\
&\quad + \int_\Omega (c_2(y^N(t);\mu)-c_2(y(t);\mu))\nabla p\nabla\varphi\,\mathrm{d}\mathbf{x}\\
&= \int_0^1 \left\langle \frac{\partial\mathcal{N}}{\partial y}(\,\cdot\,,\zeta(t,s);\mu)(y^N(t)-y(t)) + \frac{\partial\mathcal{N}}{\partial p}(\,\cdot\,,\zeta(t,s);\mu)(p^N(t)-p(t)),\varphi\right\rangle_H \mathrm{d}s\\
&\quad + \int_0^1 \left\langle \frac{\partial\mathcal{N}}{\partial q}(\,\cdot\,,\zeta(t,s);\mu)(q^N(t)-q(t)),\varphi\right\rangle_H \mathrm{d}s + \langle p(t)-p^N(t),\varphi\rangle_H\\
&\quad + \int_0^1 \left\langle \frac{\partial c_2}{\partial y}(\zeta^y(t,s);\mu)(y^N(t)-y(t))\nabla p,\nabla\varphi\right\rangle_H \mathrm{d}s
\end{aligned}
$$

for all $\varphi \in V^{N_{\mathbf{x}}}$ and f.a.a. $t \in [0,T]$, where $\zeta(t,s) = z(t) + s(z^N(t)-z(t))$ and $\zeta^y(t,s) = y(t) + s(y^N(t)-y(t))$, $(t,s) \in [0,T]\times[0,1]$, parametrize the straight lines between $z(t)$ and $z^N(t)$ and $y(t)$ and $y^N(t)$, respectively. Choosing $\varphi = \vartheta^{N_{\mathbf{x}},p}(t)$ and using (1.10d), (1.3) we find that

$$
\begin{aligned}
c_V \left\|\vartheta^{N_{\mathbf{x}},p}(t)\right\|_V^2 \le{}& \left\langle \varrho^{N_{\mathbf{x}},y}(t),\vartheta^{N_{\mathbf{x}},p}(t)\right\rangle_H + \left\langle \vartheta^{N_{\mathbf{x}},y}(t),\vartheta^{N_{\mathbf{x}},p}(t)\right\rangle_H\\
&- \int_0^1 \left\langle \frac{\partial\mathcal{N}}{\partial y}(\,\cdot\,,\zeta(t,s);\mu)(\varrho^{N_{\mathbf{x}},y}(t)+\vartheta^{N_{\mathbf{x}},y}(t)),\vartheta^{N_{\mathbf{x}},p}(t)\right\rangle_H \mathrm{d}s\\
&- \int_0^1 \left\langle \frac{\partial\mathcal{N}}{\partial p}(\,\cdot\,,\zeta(t,s);\mu)(\varrho^{N_{\mathbf{x}},p}(t)+\vartheta^{N_{\mathbf{x}},p}(t)),\vartheta^{N_{\mathbf{x}},p}(t)\right\rangle_H \mathrm{d}s\\
&+ \int_0^1 \left\langle \frac{\partial\mathcal{N}}{\partial p}(\,\cdot\,,\zeta(t,s);\mu)(\varrho^{N_{\mathbf{x}},q}(t)+\vartheta^{N_{\mathbf{x}},q}(t)),\vartheta^{N_{\mathbf{x}},p}(t)\right\rangle_H \mathrm{d}s\\
&+ \int_0^1 \left\langle \frac{\partial c_2}{\partial y}(\zeta^y(t,s);\mu)(\varrho^{N_{\mathbf{x}},y}(t)+\vartheta^{N_{\mathbf{x}},y}(t))\nabla p(t),\vartheta^{N_{\mathbf{x}},p}(t)\right\rangle_H \mathrm{d}s
\end{aligned}
\tag{1.14}
$$

f.a.a. $t \in [0,T]$. Analogously, we proceed for $\vartheta^{N_{\mathbf{x}},q}(t)$. From (1.12c), (1.4c), (1.6c) and

(1.3) we deduce that

$$\begin{aligned} a^q(\vartheta^{N_{\mathbf{x}},q}(t),\varphi) =& \langle \mathcal{N}(\,\cdot\,,z(t);\mu) - \mathcal{N}(\,\cdot\,,z^N(t);\mu),\varphi\rangle_H \\ &= \int_0^1 \left\langle \frac{\partial \mathcal{N}}{\partial y}(\,\cdot\,,\zeta(t,s);\mu)(y(t)-y^N(t)),\varphi \right\rangle_H \mathrm{d}s \\ &\quad + \int_0^1 \left\langle \frac{\partial \mathcal{N}}{\partial p}(\,\cdot\,,\zeta(t,s));\mu)(p(t)-p^N(t)),\varphi \right\rangle_H \mathrm{d}s \\ &\quad - \int_0^1 \left\langle \frac{\partial \mathcal{N}}{\partial p}(\,\cdot\,,\zeta(t,s));\mu)(q(t)-q^N(t)),\varphi \right\rangle_H \mathrm{d}s \end{aligned}$$

for all $\varphi \in V^{N_{\mathbf{x}}}$ and f.a.a. $t \in [0,1]$. Taking $\varphi = \vartheta^{N_{\mathbf{x}},q}$ and using (1.10b) it follows that

$$\begin{aligned} c_V \left\|\vartheta^{N_{\mathbf{x}},q}(t)\right\|_V^2 \leq & \int_0^1 \left\langle \frac{\partial \mathcal{N}}{\partial y}(\,\cdot\,,\zeta(t,s);\mu)(\varrho^{N_{\mathbf{x}},y}(t)+\vartheta^{N_{\mathbf{x}},y}(t)),\vartheta^{N_{\mathbf{x}},q}(t) \right\rangle_H \mathrm{d}s \\ &+ \int_0^1 \left\langle \frac{\partial \mathcal{N}}{\partial p}(\,\cdot\,,\zeta(t,s));\mu)(\varrho^{N_{\mathbf{x}},p}(t)+\vartheta^{N_{\mathbf{x}},p}(t)),\vartheta^{N_{\mathbf{x}},q}(t) \right\rangle_H \mathrm{d}s \\ &- \int_0^1 \left\langle \frac{\partial \mathcal{N}}{\partial p}(\,\cdot\,,\zeta(t,s));\mu)(\varrho^{N_{\mathbf{x}},q}(t)+\vartheta^{N_{\mathbf{x}},q}(t)),\vartheta^{N_{\mathbf{x}},q}(t) \right\rangle_H \mathrm{d}s \end{aligned} \tag{1.15}$$

f.a.a. $t \in [0,1]$. Due to Assumption 1.6 we have $\zeta(t,x) \in \mathcal{Z}_{ad}$ f.a.a. $(t,x) \in Q$ and $\|\zeta\|_{L^\infty(Q)^3} \leq C$ with a constant independent of $h$. Therefore, there exists a constant $c_{\mathcal{N}} > 0$ independent of $h$ such that

$$\operatorname*{esssup}_{t\in[0,T]} \left( \max\left\{ \int_0^1 \left\| \frac{\partial \mathcal{N}}{\partial y}(\,\cdot\,,\zeta(t,s)) \right\|_{L^\infty(\Omega)} \mathrm{d}s, \int_0^1 \left\| \frac{\partial \mathcal{N}}{\partial p}(\,\cdot\,,\zeta(t,s)) \right\|_{L^\infty(\Omega)} \mathrm{d}s \right\} \right) \leq c_{\mathcal{N}}.$$

Furthermore, there exists a constant $c_c > 0$ independent of $h$ such that

$$\operatorname*{esssup}_{t\in[0,T]} \left( \int_0^1 \left\| \frac{\partial c_3}{\partial y}(\zeta^y(t,s)) \right\|_{L^\infty(\Omega)} \mathrm{d}s \right) \leq c_c.$$

Adding (1.14) and (1.15) we obtain that

$$
\begin{aligned}
&c_V \left( \|\vartheta^{N_{\mathbf{x}},p}(t)\|_V^2 + \|\vartheta^{N_{\mathbf{x}},q}(t)\|_V^2 \right) \\
&\leq \int_0^1 \left\langle \frac{\partial \mathcal{N}}{\partial y}(\zeta(t,s);\mu), (\varrho^{N_{\mathbf{x}},y}(t) + \vartheta^{N_{\mathbf{x}},y}(t))(\vartheta^{N_{\mathbf{x}},q}(t) - \vartheta^{N_{\mathbf{x}},p}(t)) \right\rangle_H \mathrm{d}s \\
&\quad + \int_0^1 \left\langle -\frac{\partial \mathcal{N}}{\partial p}(\,\cdot\,,\zeta(t,s);\mu), (\vartheta^{N_{\mathbf{x}},p}(t) - \vartheta^{N_{\mathbf{x}},q}(t))^2 \right\rangle_H \mathrm{d}s \\
&\quad + \int_0^1 \left\langle \frac{\partial \mathcal{N}}{\partial p}(\,\cdot\,,\zeta(t,s);\mu), \varrho^{N_{\mathbf{x}},p}(t)\vartheta^{N_{\mathbf{x}},q}(t) + \varrho^{N_{\mathbf{x}},q}(t)\vartheta^{N_{\mathbf{x}},p}(t) \right\rangle_H \mathrm{d}s \\
&\quad - \int_0^1 \left\langle \frac{\partial \mathcal{N}}{\partial p}(\,\cdot\,,\zeta(t,s);\mu), \varrho^{N_{\mathbf{x}},p}(t)\vartheta^{N_{\mathbf{x}},p}(t) + \varrho^{N_{\mathbf{x}},q}(t)\vartheta^{N_{\mathbf{x}},q}(t) \right\rangle_H \mathrm{d}s \\
&\quad + \left\langle \varrho^{N_{\mathbf{x}},y}(t), \vartheta^{N_{\mathbf{x}},p}(t) \right\rangle_H + \left\langle \vartheta^{N_{\mathbf{x}},y}(t), \vartheta^{N_{\mathbf{x}},p}(t) \right\rangle_H \\
&\quad + \int_0^1 \left\langle \frac{\partial c_2}{\partial y}(\zeta^y(t,s);\mu)\nabla p(t), \varrho^{N_{\mathbf{x}},y}(t)\vartheta^{N_{\mathbf{x}},p}(t) + \vartheta^{N_{\mathbf{x}},y}(t)\vartheta^{N_{\mathbf{x}},p}(t) \right\rangle_H \mathrm{d}s \\
&\leq c_{\mathcal{N}} \left( \|\varrho^{N_{\mathbf{x}},y}(t)\|_H + \|\vartheta^{N_{\mathbf{x}},y}(t)\|_H \right) \left( \|\vartheta^{N_{\mathbf{x}},q}(t)\|_H + \|\vartheta^{N_{\mathbf{x}},p}(t)\|_H \right) \\
&\quad + c_{\mathcal{N}} \left( \|\varrho^{N_{\mathbf{x}},p}(t)\|_H \|\vartheta^{N_{\mathbf{x}},q}(t)\|_H + \|\varrho^{N_{\mathbf{x}},q}(t)\|_H \|\vartheta^{N_{\mathbf{x}},p}(t)\|_H \right) \\
&\quad + c_{\mathcal{N}} \left( \|\varrho^{N_{\mathbf{x}},p}(t)\|_H \|\vartheta^{N_{\mathbf{x}},p}(t)\|_H + \|\varrho^{N_{\mathbf{x}},q}(t)\|_H \|\vartheta^{N_{\mathbf{x}},q}(t)\|_H \right) \\
&\quad + \|\varrho^{N_{\mathbf{x}},y}(t)\|_H \|\vartheta^{N_{\mathbf{x}},p}(t)\|_H + \|\vartheta^{N_{\mathbf{x}},y}(t)\|_H \|\vartheta^{N_{\mathbf{x}},p}(t)\|_H \\
&\quad + c_c \left( \|\nabla p(t)\|_H \|\varrho^{N_{\mathbf{x}},y}(t)\|_H \|\vartheta^{N_{\mathbf{x}},p}(t)\|_H + \|\nabla p(t)\|_H \|\vartheta^{N_{\mathbf{x}},y}(t)\|_H \|\vartheta^{N_{\mathbf{x}},p}(t)\|_H \right)
\end{aligned}
$$

f.a.a. $t \in [0,T]$. Since $V$ and $V_a$ are continuously embedded into $H$, there exists a constant $c_H > 0$ such that $\|\varphi\|_H \leq c_H \|\varphi\|_V$ for all $\varphi \in V$. Additionally, since $p(t) \in H^{1+s}(\Omega)$ we can bound the norm $\|\nabla p(t)\|_H \leq c_p$. Consequently,

$$
\begin{aligned}
&c_V \left( \|\vartheta^{N_{\mathbf{x}},p}(t)\|_V^2 + \|\vartheta^{N_{\mathbf{x}},q}(t)\|_V^2 \right) \\
&\leq C_1 \left( \|\varrho^{N_{\mathbf{x}},y}(t)\|_V \|\vartheta^{N_{\mathbf{x}},q}(t)\|_V + \|\vartheta^{N_{\mathbf{x}},y}(t)\|_H \|\vartheta^{N_{\mathbf{x}},q}(t)\|_V + 3\|\varrho^{N_{\mathbf{x}},y}(t)\|_V \|\vartheta^{N_{\mathbf{x}},p}(t)\|_V \right) \\
&\quad + C_1 \left( \|\varrho^{N_{\mathbf{x}},p}(t)\|_V \|\vartheta^{N_{\mathbf{x}},p}(t)\|_V + \|\varrho^{N_{\mathbf{x}},p}(t)\|_V \|\vartheta^{N_{\mathbf{x}},q}(t)\|_V + \|\varrho^{N_{\mathbf{x}},q}(t)\|_V \|\vartheta^{N_{\mathbf{x}},p}(t)\|_V \right) \\
&\quad + C_1 \left( 3\|\vartheta^{N_{\mathbf{x}},y}(t)\|_H \|\vartheta^{N_{\mathbf{x}},p}(t)\|_V + \|\varrho^{N_{\mathbf{x}},q}(t)\|_V \|\vartheta^{N_{\mathbf{x}},q}(t)\|_V \right)
\end{aligned}
$$

f.a.a. $t \in [0,T]$ and with $C_1 = \max(1, c_{\mathcal{N}}, c_c) \max(c_H, c_H^2) \max(1, c_p)$. Using Young's

inequality [2, p. 242] we obtain

$$\begin{aligned} c_V \left( \|\vartheta^{N_{\mathbf{x}},p}(t)\|_V^2 + \|\vartheta^{N_{\mathbf{x}},q}(t)\|_V^2 \right) \\ \leq & \frac{C_1^2}{\varepsilon} \left( 5 \|\varrho^{N_{\mathbf{x}},y}(t)\|_V^2 + 5 \|\vartheta^{N_{\mathbf{x}},y}(t)\|_H^2 + \|\varrho^{N_{\mathbf{x}},p}(t)\|_V^2 + \|\varrho^{N_{\mathbf{x}},q}(t)\|_V^2 \right) \\ & + 2\varepsilon \left( \|\vartheta^{N_{\mathbf{x}},q}(t)\|_V^2 + \|\vartheta^{N_{\mathbf{x}},p}(t)\|_V^2 \right) \end{aligned}$$

for an arbitrary $\varepsilon > 0$. By choosing $\varepsilon = c_V/4$ and setting $C = 40C_1^2/c_V$ we find that

$$\begin{aligned} \|\vartheta^{N_{\mathbf{x}},p}(t)\|_V^2 + \|\vartheta^{N_{\mathbf{x}},q}(t)\|_V^2 \\ \leq C \left( \|\varrho^{N_{\mathbf{x}},y}(t)\|_V^2 + \|\varrho^{N_{\mathbf{x}},p}(t)\|_V^2 + \|\varrho^{N_{\mathbf{x}},q}(t)\|_V^2 + \|\vartheta^{N_{\mathbf{x}},y}(t)\|_H^2 \right) \end{aligned} \tag{1.13}$$

f.a.a. $t \in [0,T]$ which proves the lemma. □

The first three terms on the right-hand side of (1.13) shall be bounded by using Lemma 1.8. From the parabolic equation an estimate is derived for the term $\|\vartheta^{N_{\mathbf{x}},y}(t)\|_H^2$. Let us mention that it is essential that $\vartheta^{N_{\mathbf{x}},y}(t)$ occurs in the $H$- and not in the $V$-norm. Next we turn to an estimate for the difference $\vartheta^{N_{\mathbf{x}},y}(t) = \mathcal{P}^{N_{\mathbf{x}},y} y(t) - y^N(t)$.

**Lemma 1.10.** *Let $z$ and $z^N$ be the solution to* (1.1) *and* (1.8)*, respectively and let Assumption 1.6 be satisfied. Moreover, suppose that $y \in W(0,T;H^{1+s}(\Omega))$ and $p, q \in L^2(0,T;H^{1+s}(\Omega))$. Then there exists a constant $C > 0$ independent of $h$ and $t \in [0,T]$ such that*

$$\|\vartheta^{N_{\mathbf{x}},y}\|_{L^\infty(0,T;H)}^2 + \|\vartheta^{N_{\mathbf{x}},y}\|_{L^2(0,T;V)}^2 \leq C \left( h^s + \|\mathcal{P}^{N_{\mathbf{x}},y} y_\circ - y^N(0)\|_H^2 \right) \tag{1.16}$$

*holds.*

*Proof.* Using (1.12), (1.10b), (1.4a), (1.6a) we find

$$\begin{aligned} & \frac{\mathrm{d}}{\mathrm{d}t} \langle \vartheta^{N_{\mathbf{x}},y}(t), \varphi \rangle_H + a^y(\vartheta^{N_{\mathbf{x}},y}(t), \varphi) \\ & = \langle \mathcal{P}^{N_{\mathbf{x}},y} y_t(t) - y_t^N(t), \varphi \rangle_H + a^y(\mathcal{P}^{N_{\mathbf{x}},y} y(t) - y^N(t), \varphi) \\ & = \langle \mathcal{N}(\,\cdot\,, z^N(t); \mu) - \mathcal{N}(\,\cdot\,, z(t); \mu) + y(t) - y^N(t) + \mathcal{P}^{N_{\mathbf{x}},y} y_t(t) - y_t(t), \varphi \rangle_H \\ & = \int_0^1 \left\langle \frac{\partial \mathcal{N}}{\partial y}(\,\cdot\,, \zeta(t,s); \mu)(y^N(t) - y(t)) + \frac{\partial \mathcal{N}}{\partial p}(\,\cdot\,, \zeta(t,s); \mu)(p^N(t) - p(t)), \varphi \right\rangle_H \mathrm{d}s \\ & \quad - \int_0^1 \left\langle \frac{\partial \mathcal{N}}{\partial p}(\,\cdot\,, \zeta(t,s)); \mu)(q^N(t) - q(t)), \varphi \right\rangle_H \mathrm{d}s \\ & \quad + \int_0^1 \langle y(t) - y^N(t) + \mathcal{P}^{N_{\mathbf{x}},y} y_t(t) - y_t(t), \varphi \rangle_H \, \mathrm{d}s, \end{aligned}$$

where $\zeta(t,s) = z(t) + s(z^N(t) - z(t))$, $(t,s) \in [0,T] \times [0,1]$, parametrizes the straight line between $z(t)$ and $z^N(t)$. Recall that we have introduced the constant $c_{\mathcal{N}}$ in the proof of Lemma 1.9. Choosing $\varphi = \vartheta^{N_{\mathbf{x}},y}(t)$, $C_1 = \max(c_{\mathcal{N}} \max(1, c_H), c_H, 1)/2$ and applying (1.10b) and Young's inequality we find that

$$\begin{aligned}
&\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H^2 + c_V\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_V^2 \\
&\qquad \leq 2C_1\left(\left\|\varrho^{N_{\mathbf{x}},y}(t)\right\|_V\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H + \left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H^2\right) \\
&\qquad\quad + 2C_1\left(\left\|\varrho^{N_{\mathbf{x}},p}(t)\right\|_V\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H + \left\|\varrho^{N_{\mathbf{x}},q}(t)\right\|_V\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H\right) \\
&\qquad\quad + 2C_1\left(\left\|\vartheta^{N_{\mathbf{x}},q}(t)\right\|_V\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H + \left\|\vartheta^{N_{\mathbf{x}},p}(t)\right\|_V\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H\right) \\
&\qquad\quad + 2C_1\left(\left\|\varrho^{N_{\mathbf{x}},y}(t)\right\|_V\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H + \left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H^2\right) \\
&\qquad\quad + 2C_1\left(\left\|\mathcal{P}^{N_{\mathbf{x}},y}y_t(t) - y_t(t)\right\|_H\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H\right) \\
&\qquad \leq C_1^2\left(2\left\|\varrho^{N_{\mathbf{x}},y}(t)\right\|_V^2 + \left\|\varrho^{N_{\mathbf{x}},p}(t)\right\|_V^2 + \left\|\vartheta^{N_{\mathbf{x}},p}(t)\right\|_V^2 + \left\|\varrho^{N_{\mathbf{x}},q}(t)\right\|_V^2\right) \\
&\qquad\quad + C_1^2\left(\left\|\vartheta^{N_{\mathbf{x}},q}(t)\right\|_V^2 + 11\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H^2 + \left\|\mathcal{P}^{N_{\mathbf{x}},y}y_t(t) - y_t(t)\right\|_H^2\right).
\end{aligned}$$

From (1.13) we infer that

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H^2 + 2c_V\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_V^2 \\
&\qquad \leq C_2\left(\left\|\varrho^{N_{\mathbf{x}},y}(t)\right\|_V^2 + \left\|\varrho^{N_{\mathbf{x}},p}(t)\right\|_V^2 + \left\|\varrho^{N_{\mathbf{x}},q}(t)\right\|_V^2\right) \\
&\qquad\quad + C_2\left(\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H^2 + \left\|\mathcal{P}^{N_{\mathbf{x}},y}y_t(t) - y_t(t)\right\|_H^2\right),
\end{aligned}$$

where $C_2 = \max(C_1^2 \max(C, 22))$. Integrating over time and using Lemma 1.8 we get

$$\begin{aligned}
&\left\|\vartheta^{N_{\mathbf{x}},y}(t)\right\|_H^2 + 2\int_0^t c_V\left\|\vartheta^{N_{\mathbf{x}},y}(s)\right\|_V^2 \mathrm{d}s \\
&\qquad \leq C_2\int_0^T \left\|\varrho^{N_{\mathbf{x}},y}(t)\right\|_V^2 + \left\|\varrho^{N_{\mathbf{x}},p}(t)\right\|_V^2 + \left\|\varrho^{N_{\mathbf{x}},q}(t)\right\|_V^2 \mathrm{d}t + \left\|\vartheta^{N_{\mathbf{x}},y}(0)\right\|_H^2 \\
&\qquad\quad + C_2\int_0^t \left\|\vartheta^{N_{\mathbf{x}},y}(s)\right\|_H^2 \mathrm{d}t + C_2\int_0^T \left\|\mathcal{P}^{N_{\mathbf{x}},y}y_t(t) - y_t(t)\right\|_H^2 \mathrm{d}t \qquad (1.17) \\
&\qquad \leq C_3 h^s + \left\|\mathcal{P}^{N_{\mathbf{x}},y}y_\circ - y^N(0)\right\|_H^2 \\
&\qquad\quad + C_2\left(\left\|\mathcal{P}^{N_{\mathbf{x}},y}y_t - y_t\right\|_{L^2(0,T;H)}^2 + \int_0^t \left\|\vartheta^{N_{\mathbf{x}},y}(s)\right\|_H^2 \mathrm{d}s\right)
\end{aligned}$$

with $C_3 = 3C_2C\max(\|y\|_{L^2(0,T;H^{1+s}(\Omega))}, \|p\|_{L^2(0,T;H^{1+s}(\Omega))}, \|q\|_{L^2(0,T;H^{1+s}(\Omega))})$. Applying Lemma 1.8 on the term involving $y_t$ we get

$$\left\|\mathcal{P}^{N_{\mathbf{x}},y}y_t - y_t\right\|_{L^2(0,T;H)}^2 \leq c_H\left\|\mathcal{P}^{N_{\mathbf{x}},y}y_t - y_t\right\|_{H^1(0,T;H)}^2 \leq Cc_Hh^s\|y_t\|_{L^2(0,T;H^{1+s}(\Omega))} \leq C_4h^s.$$

Hence, we have

$$\|\vartheta^{N_{\mathbf{x}},y}(t)\|_H^2 + 2\int_0^t c_V \|\vartheta^{N_{\mathbf{x}},y}(s)\|_V^2 \,\mathrm{d}s \\ \leq C_5 h^s + \|\mathcal{P}^{N_{\mathbf{x}},y} y_\circ - y^N(0)\|_H^2 + C_2 \int_0^t \|\vartheta^{N_{\mathbf{x}},y}(s)\|_H^2 \,\mathrm{d}s$$

with $C_5 = \max(C_3, C_4)$. By using Gronwall's inequality [41, 79], there exists a constant $C_G > 0$ depending on $T$, and we get

$$\|\vartheta^{N_{\mathbf{x}},y}(t)\|_H^2 \leq C_6 \left(h^s + \|\mathcal{P}^{N_{\mathbf{x}},y} y_\circ - y^N(0)\|_H^2\right) \tag{1.18}$$

f.a.a. $t \in [0,T]$ with $C_6 = C_G \max(C_5, 1)$. Furthermore,

$$\|\vartheta^{N_{\mathbf{x}},y}(t)\|_{L^2(0,T;V)}^2 \leq C_6 \left(h^s + \|\mathcal{P}^{N_{\mathbf{x}},y} y_\circ - y^N(0)\|_H^2\right) \tag{1.19}$$

holds. Adding the results (1.18) and (1.19) concludes the proof. □

Combing the results from Lemmas 1.8, 1.9 and 1.10 we formulate the following theorem for the a-priori error estimate.

**Theorem 1.11.** *Let $z$ and $z^N$ be the solution to* (1.1) *and* (1.8)*, respectively and let Assumption 1.6 be satisfied. Moreover, suppose that $y \in W(0,T;H^{1+s}(\Omega))$ and $p, q \in L^2(0,T;H^{1+s}(\Omega))$. Then there exists a constant $C > 0$ which is independent of $N_{\mathbf{x}}$ and $\mu$ such that*

$$\int_0^T \|y(t) - y^N(t)\|_V^2 + \|p(t) - p^N(t)\|_V^2 + \|q(t) - q^N(t)\|_V^2 \,\mathrm{d}t \\ \leq C\big(\|\mathcal{P}^{N_{\mathbf{x}},y} y_\circ - y^N(0)\|_H^2 + h^s\big)$$

*with $s = 1$ for piecewise linear and $s = 2$ for piecewise quadratic finite element ansatz functions.*

In the implementation we will not distinguish between the discrete spaces for the variables $y$, $p$ and $q$ as in (1.5). To incorporate the Dirichlet boundary for the variable $q$ into the finite element approximation we utilize the boundary penalty method [5]. The idea of this method is to replace the Dirichlet boundary (1.1e) by a Robin boundary condition of the form

$$c_3(a) q_{\mathbf{x}}(t,a) + \gamma q(t,a) = \gamma g(t). \tag{1.20}$$

When choosing $\gamma$ large the Dirichlet boundary is dominant and the Neumann boundary has no influence. In the numerical experiments it can be observed that choosing

$$\gamma = c_3(a)/h^3 \quad \text{with} \quad h = \max_{i \in \{1,\ldots,N_{\mathbf{x}}-1\}} (\mathbf{x}_{2i+1} - \mathbf{x}_{2i-1}) \tag{1.21}$$

delivers very good results. This observation coincides with the results from [5]. Using this approach has the benefit of not needing to treat Dirichlet boundaries in a special way. Commercial software packages often implement this approach.

Given the finite element basis $\{\varphi_i\}_{i=1}^{dof}$ we can now compute the matrices and vectors given in (1.7). Additionally, to realize the Dirichlet boundary using the introduced penalty method the matrices $\mathcal{Q}^N$ and the vector $\mathcal{G}^N$ given by

$$\big(\big(\mathcal{Q}^N\big)\big)_{ij} = \int_\Gamma \varphi_j \varphi_i \,\mathrm{d}S \quad \text{and} \quad \big(\mathcal{G}^N(t)\big)_i = \int_\Gamma g(t)\varphi_i \,\mathrm{d}S, \quad \text{for } 1 \le i,j \le dof \tag{1.22}$$

have to be computed. For the efficient assembling of these matrices and vectors the structure of the basis can be exploited. The presented finite element approach uses basis functions with local support. Hence, the integrals over $\Omega$ reduce to integrals over the subdomains $\Omega_i$. Using the knowledge about the basis functions one can develop numerical integration rules which integrate the desired quantities exact. In the case of the previously introduced quadratic ansatz functions we here make the distinction between the case of constant coefficients and non constant coefficients. The numerical integration is realized by a Gaussian quadrature rule. Recall that Gaussian quadrature rules of order $n$ integrate polynomials of order $2n-1$ exact [26, 60, 78]. We here consider the Gauss-Legendre quadrature points and weights. Let us first discuss the matrix computation. In the case of constant coefficients we utilize the numerical integration rule of order three to integrate the obtained polynomial of fourth order exact. For non constant coefficients we assume that locally the coefficient can be approximated well by a cubic polynomial and hence we use an integration rule of order four to integrate the seventh order polynomial exact. For the computation of the right hand side integration of order two and three is sufficient under the same assumptions. The locally computed matrices and vectors are assembled and then added together to get the matrix of full dimension. By introducing a reference element $\hat{\Omega} = [0,1]$ the integration process can be handled very efficiently since the evaluation of the basis functions only has to be done once at the integration points and the integration weights are known. In the case of the one dimensional problem the transformation of $\Omega_i$ to $\hat{\Omega}$ is just shifting and scaling and hence very easy to compute. Details on this and other strategies for the finite element implementation can be found in [34, 60, 67].

**Remark 1.12.** *When using globally continuous piecewise linear ansatz functions numerical integration of order two and three for matrices and one and three for right hand sides is sufficient for the two choices of coefficients.*

Utilizing (1.8) and penalization strategy (1.20) and (1.21) we end up with the semi-discrete nonlinear system

$$\mathrm{M}_1^N \mathrm{y}_t^N(t) + \mathrm{S}_{c_1}^N \mathrm{y}^N(t) + \mathcal{N}^N(\mathrm{y}^N(t), \mathrm{p}^N(t), \mathrm{q}^N(t); \mu) = 0, \tag{1.23a}$$

$$\mathrm{M}_1^N \mathrm{y}^N(0) = \mathcal{Y}_0^N, \tag{1.23b}$$

$$\mathrm{S}_{c_2(\mathrm{y}^N(t);\mu)}^N \mathrm{p}^N(t) + \mathcal{N}^N(\mathrm{y}^N(t), \mathrm{p}^N(t), \mathrm{q}^N(t); \mu) = 0, \tag{1.23c}$$

$$(\mathrm{S}_{c_3}^N + \gamma \mathcal{Q}^N)\mathrm{q}^N(t) - \mathcal{N}^N(\mathrm{y}^N(t), \mathrm{p}^N(t), \mathrm{q}^N(t); \mu) = \mathcal{I}^N(t) + \gamma \mathcal{G}^N(t), \tag{1.23d}$$

f.a.a. $t \in [0,T]$. The time interval $[0,T]$ is still infinite dimensional. By introducing a discretization for the time variable we obtain a fully discrete system.

The semi-discrete system obtained by the discretization of the space variable can be interpreted as a nonlinear system of ordinary differential equations. There are a variety of

one-step and multi-step methods available to approach this type of systems [27, 37, 40]. Not all methods are equally suited since systems arising from the discretization of partial differential equations in general are stiff. In this work we do not want to focus on these issues. Hence, we choose a standard method for the time discretization in the context of partial differential equations. The method chosen in this work is the implicit Euler method. For this the time derivative is numerically approximated by

$$\mathrm{M}_1^N \mathrm{y}_t^N(t) \approx \mathrm{M}_1^N \frac{\mathrm{y}^N(t+\delta t) - \mathrm{y}^N(t)}{\delta t}.$$

For simplicity we use an equidistant discretization of the time interval for the fully implicit approach. Using a time adaptive strategy is possible but not the focus here. For a constant time step $\delta t$ the time discretization is given by $t_k = t_{k-1} + \delta t$ for $k = 1, \dots N_t$ and $t_0 = 0$. We set $\mathrm{y}^{N,k} \approx \mathrm{y}^N(t_k)$ and apply the implicit Euler method to (1.23a) and obtain

$$\mathrm{M}_1^N \mathrm{y}^{N,k} + \delta t \mathrm{S}_{c_1}^N \mathrm{y}^{N,k} + \delta t \mathcal{N}^N(\mathrm{y}^{N,k}, \mathrm{p}^{N,k}, \mathrm{q}^{N,k}; \mu) = \mathrm{M}_1^N \mathrm{y}^{N,k-1}. \tag{1.24}$$

Combining (1.24), (1.23c) and (1.23d) we obtain a fully discretized system. The resulting algebraic system is coupled and nonlinear:

$$(\mathrm{M}_1^N + \delta t \mathrm{S}_{c_1}^N)\mathrm{y}^{N,k} + \delta t \mathcal{N}^N(\mathrm{y}^{N,k}, \mathrm{p}^{N,k}, \mathrm{q}^{N,k}; \mu) = M_1^N \mathrm{y}^{n,k-1}, \tag{1.25a}$$

$$\mathrm{M}_1^N \mathrm{y}^{N,0} = \mathcal{Y}_0^N, \tag{1.25b}$$

$$\mathrm{S}_{c_2(\mathrm{y}^{N,k};\mu)}^N \mathrm{p}^{N,k} + \mathcal{N}^N(\mathrm{y}^{N,k}, \mathrm{p}^{N,k}, \mathrm{q}^{N,k}; \mu) = 0, \tag{1.25c}$$

$$(\mathrm{S}_{c_3}^N + \gamma \mathcal{Q}^N)\mathrm{q}^{N,k} - \mathcal{N}^N(\mathrm{y}^{N,k}, \mathrm{p}^{N,k}, \mathrm{q}^{N,k}; \mu) - \mathcal{I}^{N,k} - \gamma \mathcal{G}^{N,k} = 0, \tag{1.25d}$$

for $k = 1, \dots, N_t$. This system is a fully implicit discretized nonlinear system. At each time step a discrete nonlinear system of dimension $(3\,dof) \times (3\,dof)$ has to be solved. Further, there are $N_t$ time steps, this gives a total number of $(3\,dof) \times (3\,dof) \times N_t$ unknowns.

**Remark 1.13.** *We will not derive an a-priori error estimate for the discretized system involving the discretization of the time variable since the focus of this work is on the spatial discretization. Error estimates for the fully discrete case can be developed by utilizing methods from [60, 80].*

### 1.3.2 Numerical method

After introducing the discretization in Section 1.3.1 we now focus on the numerical methods for solving the arising nonlinear systems of equations. In every time step a nonlinear system of equations has to be solved numerically. For this the Newton method is applied [24, 26, 78]. Let us introduce the nonlinear function $F : \mathbb{R}^{3\,dof} \to \mathbb{R}^{3\,dof}$ as

$$F(\mathrm{y}^{N,k}, \mathrm{p}^{N,k}, \mathrm{q}^{N,k}) = \begin{pmatrix} F^y(\mathrm{y}^{N,k}, \mathrm{p}^{N,k}, \mathrm{q}^{N,k}) \\ F^p(\mathrm{y}^{N,k}, \mathrm{p}^{N,k}, \mathrm{q}^{N,k}) \\ F^q(\mathrm{y}^{N,k}, \mathrm{p}^{N,k}, \mathrm{q}^{N,k}) \end{pmatrix} \tag{1.26}$$

with

$$
\begin{aligned}
F^y(\mathrm{y}^{N,k},\mathrm{p}^{N,k},\mathrm{q}^{N,k}) &= (\mathrm{M}_1^N+\delta t\mathrm{S}_{c_1}^N)\mathrm{y}^{N,k}+\delta t\mathcal{N}^N(\mathrm{y}^{N,k},\mathrm{p}^{N,k},\mathrm{q}^{N,k};\mu)-\mathrm{M}_1^N\mathrm{y}^{n,k-1},\\
F^p(\mathrm{y}^{N,k},\mathrm{p}^{N,k},\mathrm{q}^{N,k}) &= \mathrm{S}_{c_2(\mathrm{y}^{N,k};\mu)}^N\mathrm{p}^{N,k}+\mathcal{N}^N(\mathrm{y}^{N,k},\mathrm{p}^{N,k},\mathrm{q}^{N,k};\mu),\\
F^q(\mathrm{y}^{N,k},\mathrm{p}^{N,k},\mathrm{q}^{N,k}) &= (\mathrm{S}_{c_3}^N+\gamma\mathcal{Q}^N)\mathrm{q}^{N,k}-\mathcal{N}^N(\mathrm{y}^{N,k},\mathrm{p}^{N,k},\mathrm{q}^{N,k};\mu)-\mathcal{I}^{N,k}.
\end{aligned}
$$

The goal is to find the root of $F$ using the Newton method. Note that the function $F$ corresponds to the residual of the nonlinear system which is zero when the solution is inserted. The Newton method requires the derivative of $F$. Hence, the next step is to compute the Jacobian $F':\mathbb{R}^{3\,dof}\to\mathbb{R}^{3\,dof\times 3\,dof}$ given as

$$
F'(\mathrm{y}^{N,k},\mathrm{p}^{N,k},\mathrm{q}^{N,k})=\begin{pmatrix}
\frac{\partial F^y}{\partial \mathrm{y}^{N,k}} & \frac{\partial F^y}{\partial \mathrm{p}^{N,k}} & \frac{\partial F^y}{\partial \mathrm{q}^{N,k}}\\
\frac{\partial F^p}{\partial \mathrm{y}^{N,k}} & \frac{\partial F^p}{\partial \mathrm{p}^{N,k}} & \frac{\partial F^p}{\partial \mathrm{q}^{N,k}}\\
\frac{\partial F^q}{\partial \mathrm{y}^{N,k}} & \frac{\partial F^q}{\partial \mathrm{p}^{N,k}} & \frac{\partial F^q}{\partial \mathrm{q}^{N,k}}
\end{pmatrix} \tag{1.27}
$$

with

$$
\begin{aligned}
&\frac{\partial F^y}{\partial \mathrm{y}^k}=\mathrm{M}_1^N+\delta t\mathrm{S}_{c_1}^N+\delta t\mathrm{M}_{\frac{\partial\mathcal{N}}{\partial y}(k)}^N, && \frac{\partial F^y}{\partial \mathrm{p}^{N,k}}=\delta t\mathrm{M}_{\frac{\partial\mathcal{N}}{\partial p}(k)}^N, && \frac{\partial F^y}{\partial \mathrm{q}^{N,k}}=\delta t\mathrm{M}_{\frac{\partial\mathcal{N}}{\partial q}(k)}^N,\\
&\frac{\partial F^p}{\partial \mathrm{y}^{N,k}}=\mathrm{C}_{\frac{\partial c_2}{\partial y}(k)p^N(t^k)_{\mathrm{x}}}^N+\mathrm{M}_{\frac{\partial\mathcal{N}}{\partial y}(k)}^N, && \frac{\partial F^p}{\partial \mathrm{p}^{N,k}}=\mathrm{S}_{c_2}^N+\mathrm{M}_{\frac{\partial\mathcal{N}}{\partial p}(k)}^N, && \frac{\partial F^p}{\partial \mathrm{q}^{N,k}}=\mathrm{M}_{\frac{\partial\mathcal{N}}{\partial q}(k)}^N,\\
&\frac{\partial F^q}{\partial \mathrm{y}^{N,k}}=-\mathrm{M}_{\frac{\partial\mathcal{N}}{\partial y}(k)}^N, && \frac{\partial F^q}{\partial \mathrm{p}^{N,k}}=-\mathrm{M}_{\frac{\partial\mathcal{N}}{\partial p}(k)}^N, && \frac{\partial F^q}{\partial \mathrm{q}^{N,k}}=\mathrm{S}_{c_3}^N-\mathrm{M}_{\frac{\partial\mathcal{N}}{\partial q}(k)}^N
\end{aligned}
$$

and

$$
\left(\left(\mathrm{C}_f^N\right)\right)_{ij}=\int_\Omega f(x)\varphi_j(\varphi_i)'\,\mathrm{d}x,\quad \text{for } 1\le i,j\le dof.
$$

Note that the term $\frac{\partial\mathcal{N}}{\partial y}(k)$ stands short for $\frac{\partial\mathcal{N}}{\partial y}(\mathrm{x},y^N(t^k);\mu)$. Analogously, the same holds for the other derivatives of the nonlinear terms. The derivatives of the nonlinear terms are given in the Appendix A.1.

By introducing the iteration counter $\nu=0,\dots,\nu_{\max}$, with $\nu_{\max}\in\mathbb{N}_0$, for the Newton method we can write down the individual computation steps. First the Newton correction $\delta\mathrm{d}^\nu$ is computed by solving the Newton equation

$$
F'(\mathrm{y}^{N,k,\nu},\mathrm{p}^{N,k,\nu},\mathrm{q}^{N,k,\nu})\delta\mathrm{d}^\nu=-F(\mathrm{y}^{N,k,\nu},\mathrm{p}^{N,k,\nu},\mathrm{q}^{N,k,\nu}). \tag{1.28}
$$

Then the Newton step or update is computed by

$$
(\mathrm{y}^{N,k,\nu+1},\mathrm{p}^{N,k,\nu+1},\mathrm{q}^{N,k,\nu+1})=(\mathrm{y}^{N,k,\nu},\mathrm{p}^{N,k,\nu},\mathrm{q}^{N,k,\nu})+\delta\mathrm{d}^\nu. \tag{1.29}
$$

Lastly, a stopping criterion is evaluated. In this work we will use the residual as a stopping criterion for the Newton method of the form

$$
\|F^{y,\nu}\|_{V^*}+\|F^{p,\nu}\|_{V^*}+\|F^{q,\nu}\|_{V^*}<\varepsilon^{\text{Newton}}
$$

where $F^{y,\nu}$ corresponds to the evaluation $F^{y}(\mathrm{y}^{N,k,\nu}, \mathrm{p}^{N,k,\nu}, \mathrm{q}^{N,k,\nu})$, for $F^{p,\nu}$ and $F^{q,\nu}$, respectively. Note that the dual norms are required which by Theorem 1.2 involves the solving of linear systems. Additionally, the Newton method is stopped if the number of updates exceeds $\nu_{\max}$ iterations. This criterion is important in order to avoid computational expenses when the method diverges. From the theory it is well known that the Newton method is a local method with quadratic convergence. For completeness let us state one variant of the Newton theorem, the so called *refined Newton-Mysovskikh theorem* [24].

**Theorem 1.14.** *Let $D \subset \mathbb{R}^n$ be open and convex and $F : D \to \mathbb{R}^n$ a continuously differentiable mapping. Suppose that $F'(\mathrm{z})$ is invertible for each $\mathrm{z} \in D$. Assume that the following affine covariant Lipschitz condition holds*

$$\|F'(\mathrm{z})^{-1}(F'(\tilde{\mathrm{z}}) - F'(\mathrm{z}))(\tilde{\mathrm{z}} - \mathrm{z})\| \leq \omega \|\tilde{\mathrm{z}} - \mathrm{z}\|^2$$

*for $\mathrm{z}, \tilde{\mathrm{z}} \in D$. Let $F(\mathrm{z}) = 0$ have a solution $\mathrm{z}^\star$. For the initial guess $\mathrm{z}^0$ assume that the closed ball $\bar{B}(\mathrm{z}^\star, \|\mathrm{z}^0 - \mathrm{z}^\star\|) \subset D$ and that*

$$\omega \|\mathrm{z}^0 - \mathrm{z}^\star\| < 2. \tag{1.30}$$

*Then the ordinary Newton iteration defined by*

$$F'(\mathrm{z}^\nu)\delta\mathrm{d}^\nu = -F(\mathrm{z}^\nu), \quad \mathrm{z}^{\nu+1} = \mathrm{z}^\nu + \delta\mathrm{d}^\nu, \quad \nu = 0, 1, \ldots$$

*remains in the open ball $B(\mathrm{z}^\star, \|\mathrm{z}^0 - \mathrm{z}^\star\|)$ and converges to $\mathrm{z}^\star$ at the estimated rate*

$$\|\mathrm{z}^{\nu+1} - \mathrm{z}^\star\| \leq \frac{\omega}{2} \|\mathrm{z}^\nu - \mathrm{z}^\star\|^2.$$

*Moreover, the solution $\mathrm{z}^\star$ is unique in the open ball $B(\mathrm{z}^\star, 2/\omega)$.*

The proof to this theorem can be found in many books, e.g. [24–27,78]. By defining $\mathrm{z} = (\mathrm{y}^{N,k}, \mathrm{p}^{N,k}, \mathrm{q}^{N,k})$ Theorem 1.14 can easily be put in relation with (1.26) and the Newton method described for it. Note that the conditions of this theorem are hard to check and only provide convergence for an initial guess in a closed ball around the solution, i.e. local convergence. Since in general the solution $\mathrm{z}^\star$ is not known other strategies have to be developed in order to guarantee convergence. Therefore, globalization methods have to be applied to obtain robust algorithms. We will have a look at the different strategies that are utilized in this work.

For the convergence of the Newton method the initial guess is essential, see the condition given by (1.30). If the initial guess is sufficiently close to the solution the convergence is guaranteed by Theorem 1.14. To obtain a good initial guess for $\mathrm{y}^{N,k}$, $\mathrm{p}^{N,k}$ and $\mathrm{q}^{N,k}$ the following approach is used. Instead of using the fully implicit discretization for the variable $y$ as introduced in (1.24) we replace it by a semi-implicit discretization to compute an initial guess for the Newton method. In the semi-implicit discretization the nonlinear term is evaluated at the previous time step. Given the $(k-1)$-st step the arising system for the initial guess of the $k$-th step is given by

$$(\mathrm{M}_1^N + \delta t \mathrm{S}_{c_1}^N)\mathrm{y}^{N,k,0} = \mathrm{M}_1^N \mathrm{y}^{n,k-1} - \delta t \mathcal{N}^N(\mathrm{y}^{N,k-1}, \mathrm{p}^{N,k-1}, \mathrm{q}^{N,k-1}; \mu). \tag{1.31}$$

Then by inserting $y^{N,k,0}$ into the equation (1.25c) and (1.25d) we obtain the nonlinear elliptic system for initial guesses $p^{N,k,0}$ and $q^{N,k,0}$ of the form

$$\mathrm{S}^N_{c_2(\mathrm{y}^{N,k,0};\mu)}\mathrm{p}^{N,k,0} + \mathcal{N}^N(\mathrm{y}^{N,k,0}, \mathrm{p}^{N,k,0}, \mathrm{q}^{N,k,0}; \mu) = 0, \tag{1.32a}$$

$$(\mathrm{S}^N_{c_3} + \gamma\mathcal{Q}^N)\mathrm{q}^{N,k,0} - \mathcal{N}^N(\mathrm{y}^{N,k,0}, \mathrm{p}^{N,k,0}, \mathrm{q}^{N,k,0}; \mu) - \mathcal{I}^{N,k} = 0. \tag{1.32b}$$

Note that (1.31) and (1.32) are solved independently. This reduces the computational cost since one solves two smaller systems compared to one big system. Choosing a fully implicit approach, i.e. also implicit for the variables $p$ and $q$ fails since the system decouples and delivers inaccurate results. Using the solution of the previous time step as initial guess fails in many cases. Hence, the proposed approach is a good trade-off between computational expenses and robustness. The described strategy does not guarantee convergence of the Newton method but provides an improvement for the initial guess.

After computing a more suitable initial guess for the Newton method the next step is to apply a globalization strategy. As we have seen in Theorem 1.14 the condition (1.30) is essential for the convergence. The method might converge if this condition is not satisfied but it is not guaranteed. By exploiting an additional global structure of $F$ like convexity, convergence of the Newton method can be obtained. For a general mapping $F$ a globalization strategy has to be utilized. The two most prominent approaches are the trust region methods and their variants and the Newton method with damping strategy. In this work we focus on the approach involving a damping strategy. A damped Newton iteration is given by replacing (1.29) by

$$(\mathrm{y}^{N,k,\nu+1}, \mathrm{p}^{N,k,\nu+1}, \mathrm{q}^{N,k,\nu+1}) = (\mathrm{y}^{N,k,\nu}, \mathrm{p}^{N,k,\nu}, \mathrm{q}^{N,k,\nu}) + \tau_\nu \delta\mathrm{d}^\nu \tag{1.33}$$

or in more compact notation $\mathrm{z}^{\nu+1} = \mathrm{z}^\nu + \tau_\nu \delta\mathrm{d}^\nu$. The step length $\tau_\nu \in (0, 1]$ is chosen by some suitable criteria in order to achieve global convergence. The criterion should be easy and inexpensive to evaluate. In practice a monotonicity test is used to decide whether to accept or reject an iterate. The two most common ones are the *residual* monotonicity test and the *natural* monotonicity test. These monotonicity tests are easy to check and can be used as convergence criteria in algorithms. Let us have a short look at both of them.

The residual based approach uses the residual as a measure for the convergence. Solving $F(\mathrm{z}) = 0$ is equivalent to minimizing the residual. Hence, one can expect that the residual is monotonically decreasing, i.e.

$$\|F(\mathrm{z}^{\nu+1})\|_{V^*} \leq \Theta \|F(\mathrm{z}^\nu)\|_{V^*} \quad \text{for} \quad \nu = 0, 1, \ldots \tag{1.34}$$

and some positive $\Theta < 1$. Note that this type of criteria is not affine invariant. This results in the fact that when multiplying $F$ by an invertible matrix $A$ the result of the monotonicity can be changed arbitrarily. Hence, it is required to have a more robust approach which is affine invariant. For this the natural monotonicity test is introduced as

$$\|F'(\mathrm{z}^\nu)^{-1} F(\mathrm{z}^{\nu+1})\|_V \leq \Theta \|F'(\mathrm{z}^\nu)^{-1} F(\mathrm{z}^\nu)\|_V \quad \text{for} \quad \nu = 0, 1, \ldots$$

and some positive $\Theta < 1$. This condition can also be written in terms of the Newton correction $\delta d^\nu$ and reads as

$$\|\overline{\delta d}^\nu\|_V \leq \Theta \|\delta \mathrm{d}^\nu\|_V, \quad F'(\mathrm{z}^\nu)\overline{\delta \mathrm{d}}^\nu = F(\mathrm{z}^{\nu+1}) \quad \text{for} \quad \nu = 0, 1, \ldots. \tag{1.35}$$

Note that $\overline{\delta \mathrm{d}}^{\nu}$ is known as the *simplified* Newton correction since only the right hand side of the Newton equation is updated. This approach is affine invariant and is the preferred approach in problems arising from discretization of nonlinear partial differential equations. The difficulties occur especially in situations where the Jacobian matrix is ill-conditioned. In this work a damping strategy using the natural monotonicity test will be used.

Let us next have a look at the numerical realization of the introduced damped Newton method. First we demonstrate how the natural monotonicity test is utilized to extend the local convergence of the Newton method. We follow the strategy presented in [26].

To obtain convergence of the Newton method damped steps of the form (1.33) are performed. Here the choice of $\tau_\nu$, the step length or *damping factor*, is important. For the choice of the damping factor the natural monotonicity test (1.35) is utilized. We introduce $\Theta = 1 - \tau_\nu/2$ and require

$$\|\overline{\delta \mathrm{d}}^{\nu}(\tau_\nu)\|_V \leq \left(1 - \frac{\tau_\nu}{2}\right) \|\delta \mathrm{d}^{\nu}\|_V, \tag{1.36}$$

where $\overline{\delta \mathrm{d}}^{\nu}(\tau_\nu)$ is the simplified Newton correction given by the linear system

$$F'(\mathrm{z}^{\nu})^{-1} \overline{\delta \mathrm{d}}^{\nu}(\tau_\nu) = F(\mathrm{z}^{\nu} + \tau_\nu \delta \mathrm{d}^{\nu}).$$

Note that the choice of $\Theta$ relaxes the monotonicity test since $\tau_\nu \in (0, 1]$ and from theory [24] it is known that the simplified Newton correction satisfies the condition

$$\|\overline{\delta \mathrm{d}}^{\nu}\|_V \leq \frac{1}{4} \|\delta \mathrm{d}^{\nu}\|_V.$$

In the implementation a monotonic decreasing sequence of $\tau_\nu$ is generated for which (1.36) is checked. The Newton update is then performed with the largest $\tau_\nu$ for which (1.36) is satisfied. The most simple implementation can be realized by a backtracking strategy, where the sequence for $\tau_\nu$ is given by

$$\tau_\nu^\rho = \gamma^\rho, \quad \rho = 0, 1, \ldots.$$

A typical choice for $\gamma$ is $1/2$. It is advised to introduce a $\rho_{\max}$ to avoid too small damping factors $\tau_\nu$ and non converging loops during the implementation.

After introducing the computation of a good initial guess for the Newton method and a damping strategy to achieve a global convergence we next utilize some properties of the equation itself to obtain a robust solver. The nonlinear term (1.2a) is crucial when solving the system (1.1). Together with the results obtained by Theorem 1.4 we introduce the following safeguards. In every iteration of the Newton method it has to be ensured that the variable $y^N(t_k, \mathrm{x})$ is positive, or more strictly, the condition $y_{\min} \leq y^N(t_k)$ has to be satisfied. Since nodal ansatz functions are used this condition can be written as

$$y_{\min} \leq \mathrm{y}^{N,k}. \tag{1.37}$$

With this bound the evaluation of the square root and logarithm is secured. Additionally, to the lower bound for $\mathrm{y}^{N,k}$ it is advised to incorporate a bound for the argument of the hyperbolic sine. Since the hyperbolic sine exhibits an exponential growth, too large arguments can lead to significant numerical problems. For this reason a bound of the form

$$\max \left| \mu_1(\mathrm{q}^{N,k} - \mathrm{p}^{N,k}) - \ln \mathrm{y}^{N,k} \right| \leq C_{\sinh} \tag{1.38}$$

is introduced. With these two safeguards the numerical stability of the Newton method and the nonlinear solver is further enforced. Note that these safeguards have to be checked for every Newton iterations, i.e. for every $\mathrm{y}^{N,k,\nu}$, $\mathrm{p}^{N,k,\nu}$ and $\mathrm{q}^{N,k,\nu}$. With these safeguards it is guaranteed that for every Newton update the nonlinear term can be evaluated.

Lastly, we have a look at the computational expenses of the Newton method utilizing the proposed damping strategy. Obviously, solving for the simplified Newton correction is as expensive as solving for the Newton correction which makes the method expensive. Even if no damping is needed two sets of Newton equations have to be solved. On the other hand, when looking at the structure of the Jacobian matrix we will observe that this can be done very efficiently using for example a LU-decomposition. The LU-decomposition is the method of choice in this case since the Jacobian is not symmetric. By applying a reordering algorithm to the Jacobian one can speed up the computation significantly. Here we will apply the `colamd` algorithm available in `MATLAB`. This algorithm implements the column approximate minimum degree permutation [22, 23]. Also other choices are available but are not considered here. In Figure 1.1 and 1.2 the effect of the reordering is demonstrated. It can be observed that the bandwidth of the Jacobian is reduced significantly and by applying a LU-decomposition the number of non zeros is much lower compared to the non reordered variant. By using this approach the linear systems can be solved very efficiently. Note that the reordering only has to be computed once since the Jacobian always has the same structure. The natural monotonicity test hence only involves additional forward and backward substitutions which are cheap. The computationally expensive LU-decomposition only has to be computed once in every Newton iteration and is reused in the damping process.

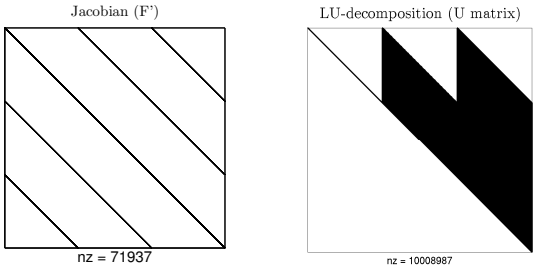


Figure 1.1: Structure of the Jacobian matrix (1.27) (left) and the corresponding upper triangular matrix obtained by the LU-decomposition (right).

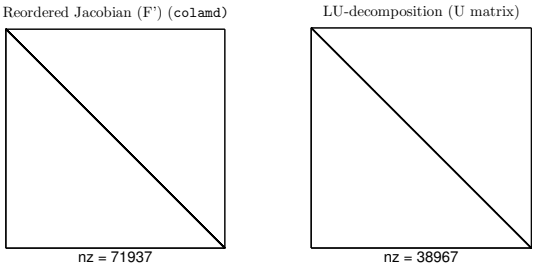


Figure 1.2: Structure of the reordered Jacobian matrix (1.27) (left) and the corresponding upper triangular matrix obtained by the LU-decomposition (right).

To summarize the procedure for solving the full discrete nonlinear system given by (1.25) the strategy is shortly outlined as a pseudo code in Algorithm 1. The main idea is to sequentially apply the damped Newton method, given by Algorithm 2, for each time step. For the computation of the initial condition, first $\mathrm{y}^{N,k,0}$ is computed by (1.31). Then by applying Algorithm 2 to (1.32) the initial conditions $\mathrm{p}^{N,k,0}$ and $\mathrm{q}^{N,k,0}$ are obtained.

**Algorithm 1** (Solver for nonlinear system)

**Require:** $\mathrm{y}^{N,0}$

Compute $\mathrm{p}^{N,0}$ and $\mathrm{q}^{N,0}$ by solving (1.32) using Algorithm 2 with fixed $\mathrm{y}^{N,0}$
$k \leftarrow 1$
**while** $k \leq N_t$ **do**
  Compute initial guess $\mathrm{y}^{N,k,0}$ solving

$$(\mathrm{M}_1^N + \delta t \mathrm{S}_{c_1}^N)\mathrm{y}^{N,k,0} = \mathrm{M}_1^N \mathrm{y}^{n,k-1} - \delta t \mathcal{N}^N(\mathrm{y}^{N,k-1}, \mathrm{p}^{N,k-1}, \mathrm{q}^{N,k-1}; \mu)$$

  Compute $\mathrm{p}^{N,k,0}$ and $\mathrm{q}^{N,k,0}$ by solving (1.32) using Algorithm 2 with fixed $\mathrm{y}^{N,k,0}$
  Compute $\mathrm{y}^{N,k}$, $\mathrm{p}^{N,k}$ and $\mathrm{p}^{N,k}$ using Algorithm 2
  $k \leftarrow k + 1$
**end while**
**return** $\mathrm{Y}^N \leftarrow [\mathrm{y}^{N,0}, \ldots, \mathrm{y}^{N,N_t}]$, $\mathrm{P}^N \leftarrow [\mathrm{p}^{N,0}, \ldots, \mathrm{p}^{N,N_t}]$, $\mathrm{Q}^N \leftarrow [\mathrm{q}^{N,0}, \ldots, \mathrm{q}^{N,N_t}]$

**Algorithm 2** (Damped Newton method with natural monotonicity test)

**Require:** Initial guess $\mathrm{y}^{N,k,0}$, $\mathrm{p}^{N,k,0}$ and $\mathrm{q}^{N,k,0}$

Set $\nu \leftarrow 0$.
**while** $\|F^{y,\nu}\|_{V^*} + \|F^{p,\nu}\|_{V^*} + \|F^{q,\nu}\|_{V^*} > \varepsilon^{\text{Newton}}$ and $\nu \leq \nu_{\max}$ **do**
  Solve for the Newton correction

$$F'(\mathrm{y}^{N,k,\nu}, \mathrm{p}^{N,k,\nu}, \mathrm{q}^{N,k,\nu})\delta\mathrm{d}^\nu = -F(\mathrm{y}^{N,k,\nu}, \mathrm{p}^{N,k,\nu}, \mathrm{q}^{N,k,\nu})$$

  Set $(\mathrm{y}^{N,k,\nu+1}, \mathrm{p}^{N,k,\nu+1}, \mathrm{q}^{N,k,\nu+1}) \leftarrow (\mathrm{y}^{N,k,\nu}, \mathrm{p}^{N,k,\nu}, \mathrm{q}^{N,k,\nu}) + \delta\mathrm{d}^\nu$
  Check the safeguards (1.37) and (1.38) for $(\mathrm{y}^{N,k,\nu+1}, \mathrm{p}^{N,k,\nu+1}, \mathrm{q}^{N,k,\nu+1})$.
  Solve for the simplified Newton correction

$$F'(\mathrm{y}^{N,k,\nu}, \mathrm{p}^{N,k,\nu}, \mathrm{q}^{N,k,\nu})\overline{\delta\mathrm{d}}^\nu = -F(\mathrm{y}^{N,k,\nu+1}, \mathrm{p}^{N,k,\nu+1}, \mathrm{q}^{N,k,\nu+1})$$

  Set $\tau_\nu \leftarrow 1$ and $\rho \leftarrow 1$
  **while** $\|\overline{\delta\mathrm{d}}^\nu(\tau_\nu)\|_V > \left(1 - \frac{\tau_\nu}{2}\right)\|\delta\mathrm{d}^\nu\|_V$ and $\rho \leq \rho_{\max}$ **do**
    Reduce step size $\tau_\nu \leftarrow \frac{1}{2}\tau_\nu$
    Set $(\mathrm{y}^{N,k,\nu+1}, \mathrm{p}^{N,k,\nu+1}, \mathrm{q}^{N,k,\nu+1}) \leftarrow (\mathrm{y}^{N,k,\nu}, \mathrm{p}^{N,k,\nu}, \mathrm{q}^{N,k,\nu}) + \tau_\nu\delta\mathrm{d}^\nu$
    Check the safeguards (1.37) and (1.38) for $(\mathrm{y}^{N,k,\nu+1}, \mathrm{p}^{N,k,\nu+1}, \mathrm{q}^{N,k,\nu+1})$.
    Solve for the simplified Newton correction

$$F'(\mathrm{y}^{N,k,\nu}, \mathrm{p}^{N,k,\nu}, \mathrm{q}^{N,k,\nu})\overline{\delta\mathrm{d}}^\nu = -F(\mathrm{y}^{N,k,\nu+1}, \mathrm{p}^{N,k,\nu+1}, \mathrm{q}^{N,k,\nu+1})$$

    Set $\rho \leftarrow \rho + 1$
  **end while**
  Set $\nu \leftarrow \nu + 1$
**end while**
**return** $\mathrm{y}^{N,k} \leftarrow \mathrm{y}^{N,k,\nu}$, $\mathrm{p}^{N,k} \leftarrow \mathrm{p}^{N,k,\nu}$, $\mathrm{q}^{N,k} \leftarrow \mathrm{q}^{N,k,\nu}$

### 1.3.3 Numerical results

In this section we present some numerical results using the finite element method. For this let us first introduce the settings. We start by introducing the domain $\Omega = [0, 5]$. Further, for the partitioning of the domain we set $s_1 = 2$ and $s_2 = 3$. Hence, we get $\Omega = \Omega_l \cup \Omega_c \cup \Omega_r$ with $\Omega_l = [0, 2]$, $\Omega_c = (2, 3)$ and $\Omega_r = [3, 5]$. This partitioning is motivated by the structure found in lithium ion batteries; see, e.g. [33, 72, 86].

Next let us consider the missing quantities for system (1.1). We start by introducing the diffusion coefficients. The piecewise constant diffusion coefficients are given as

$$c_1(x) = \begin{cases} 3 & \text{for} \quad \mathbf{x} \in \Omega_l, \\ 4 & \text{for} \quad \mathbf{x} \in \Omega_c, \\ 2 & \text{for} \quad \mathbf{x} \in \Omega_r, \end{cases} \quad \text{and} \quad c_3(x) = \begin{cases} 1 & \text{for} \quad \mathbf{x} \in \Omega_l, \\ 10^{-3} & \text{for} \quad \mathbf{x} \in \Omega_c, \\ 5 & \text{for} \quad \mathbf{x} \in \Omega_r. \end{cases} \tag{1.39a}$$

For the nonlinear diffusion coefficient we choose a cubic polynomial in $y$ and $\mu_4$ of the form

$$c_2(y; \mu) = (1 + \mu_4 y)^3 - 1. \tag{1.39b}$$

For the bounds in the admissible set we choose $\underline{\mu}_4 = 0$ and $\overline{\mu}_4 = 3$. Note that this choice for $c_2$ fulfills the conditions required in Theorem 1.4 for the existence and uniqueness of the solution. The required derivatives for the Newton method are given by (A.3) in Appendix A. The nonlinear term $\mathcal{N}$ is chosen as introduced in (1.2a) together with the indicator function (1.2b). Further, let us introduce the boundary value for (1.1e) by

$$\mathcal{I}(t) = \frac{t}{2} \sin(2\pi t). \tag{1.39c}$$

Moreover we set the initial condition (1.1f) for the variable $y$ as

$$y_\circ(x) = 1. \tag{1.39d}$$

In the numerical experiment we choose different values for $T$ to compare the computational time. In all experiments we choose $\delta t = 1/100$, i.e. $N_t = 100\,T$. As already mentioned in Section 1.3.1 we use second order finite elements. We set $N_{\mathbf{x}} = 1000$. Hence, we end up with $1999$ degrees of freedom $(dof)$. For the Newton method the tolerance $\varepsilon^{\text{Newton}} = 10^{-10}$ is chosen. The safeguards are set to $y_{\min} = 0.01$ and $C_{\sinh} = 10$. In the presented results we choose two settings for the parameter

$$\bar{\mu} = [1.1, -0.9, -0.2, 0.1] \quad \text{and} \quad \tilde{\mu} = [1.4, -1.6, -0.3, 1.6]$$

to demonstrate some of the dynamics in the system. For details on the computer and software used to obtain these numerical results see Appendix B.1.

In Figures 1.3 and 1.4 the numerical solution for the variables $y$, $p$ and $q$, solving (1.1) are presented for the parameters $\bar{\mu}$ and $\tilde{\mu}$, respectively. From the figures it can be seen that the change in the parameter results in very different scaled solutions. While the basic shape of the solutions share some similarities the values are very different, especially in the variables $p$ and $q$. The computational performance for different values of $T$ is presented in Table 1.1. The proposed method for solving the nonlinear system performs very well

and the Newton method needs for both parameter settings in average two to three iterations in every time step to solve the nonlinear system. The computation of the initial guess in average takes three Newton iterations on the nonlinear elliptic system (1.32). It can be observed that the damping strategy is most active when the sign in the boundary term $\mathcal{I}(t)$ changes. Further, the introduced safeguards are never active in these two parameter settings. From the computational time the slight fluctuation in the number of Newton iteration can be observed. With 'Setup FEM' everything that can be precomputed is summarized. This involves the generation of the time and space discretization, the reordering using `colamd`, the generation of the reference element and the assembling of matrices and right hand sides that are time independent.

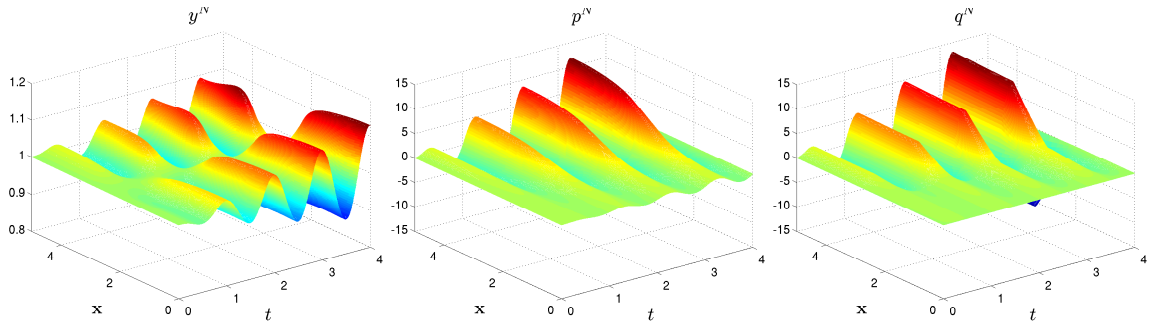


Figure 1.3: Numerical solution $y$ (left), $p$ (center) and $q$ (right) to nonlinear elliptic-parabolic system (1.1) for $(t, \mathbf{x}) \in [0, 4] \times [0, 5]$ for parameter $\bar{\mu}$.

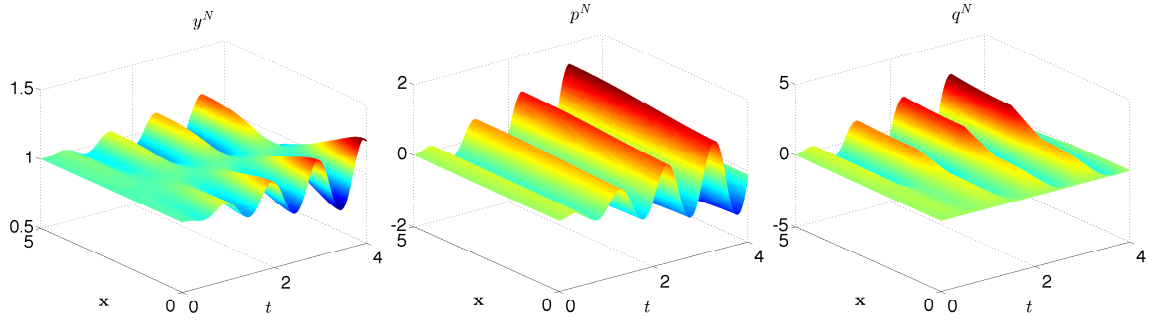


Figure 1.4: Numerical solution $y$ (left), $p$ (center) and $q$ (right) to nonlinear elliptic-parabolic system (1.1) for $(t, \mathbf{x}) \in [0, 4] \times [0, 5]$ for parameter $\tilde{\mu}$.

| | Setup FEM | $T = 1$ | $T = 2$ | $T = 3$ | $T = 4$ |
|---|---|---|---|---|---|
| CPU time (seconds) | 0.029 | 27.8 | 57.5 | 88.4 | 132.0 |

Table 1.1: Computation time using the finite element method to solve (1.1) for $\bar{\mu}$ and different time intervals.

When pushing the parameter $\mu_1$ out of the introduced bounds numerical results can still be obtained. The number of Newton iterations increase and fluctuate strongly. In certain cases divergence can be observed and the safeguards become active. For the parameters $\mu_2$ and $\mu_3$ it is important to keep them away from zero. In the case of zero the system decouples and can no longer be solved due to the choice of the boundary conditions. If $\mu_2$ and $\mu_3$ are chosen close to zero the computation time increases dramatically since many Newton iterations are needed and the safeguards become active. Setting $\mu_2 = 0.05$, solving the linear system takes up to $2000$ seconds, opposed to the $132$ presented in Table 1.1.

# Chapter 2

# The Proper Orthogonal Decomposition (POD) method

In this chapter we introduce the POD method and develop the POD Galerkin scheme for (1.1). Moreover, an a-priori error estimate will be derived. The realization of the reduced order approach is outlined. To evaluate the nonlinearity efficiently in our reduced order model the *empirical interpolation method* (EIM) will be utilized. Finally, numerical results are presented to demonstrate the efficiency of the proposed strategy.

## 2.1 The continuous version of the POD method

The goal is to construct a reduced order model to solve (1.1). For this the ansatz functions in the Galerkin scheme for the nonlinear system should be chosen in an optimal way. In this section we introduce the POD method for the infinite dimensional nonlinear system. Assume that $z = (y, p, q) = \mathcal{S}(\mu)$ is the weak solution to (1.1) for a chosen parameter $\mu \in \mathcal{M}_{ad}$. The computation of the POD basis for the first solution component $y$ is explained. For the other variables the procedure is analogous. Suppose that $\mathcal{H}$ denotes either the space $H$ or the space $V$. We set $\mathcal{V}^y = \operatorname{span}\{y(t) \,|\, t \in [0,T]\} \subset V$ with $d_y = \dim \mathcal{V}^y \leq \infty$. Notice that for $p$ and $q$ we choose $H$ or $V$ and $V_a$, respectively. The goal is to construct a low dimensional basis by solving the optimization problem

$$\begin{cases} \min\limits_{\psi_1^y,\dots,\psi_{\ell_y}^y \in \mathcal{H}} \displaystyle\int_0^T \Big\| y(t) - \sum_{i=1}^{\ell_y} \langle y(t), \psi_i^y \rangle_{\mathcal{H}} \, \psi_i^y \Big\|_{\mathcal{H}}^2 \, \mathrm{d}t \\ \quad \text{subject to (s.t.)} \quad \langle \psi_i^y, \psi_j^y \rangle_{\mathcal{H}} = \delta_{ij} \quad \text{for} \quad 1 \leq i, j \leq \ell_y, \end{cases} \tag{2.1}$$

where $\delta_{ij}$ stands for the Kronecker symbol, i.e. $\delta_{ij} = 0$ for $i \neq j$ and $\delta_{ii} = 1$. To obtain the basis $\psi_i$, $i = 1, \dots, \ell_y$ the infinite dimensional optimization problem (2.1) has to be solved. For an easier representation we reformulate the problem as a maximization problem of the

form

$$\begin{cases} \max\limits_{\psi_1^y,\ldots,\psi_{\ell_y}^y \in \mathcal{H}} \displaystyle\int_0^T \sum_{i=1}^{\ell_y} |\langle y(t), \psi_i^y \rangle_{\mathcal{H}}|^2 \, \mathrm{d}t \\ \quad \text{s.t.} \quad \langle \psi_i^y, \psi_j^y \rangle_{\mathcal{H}} - \delta_{ij} = 0 \quad \text{for} \quad 1 \le i, j \le \ell_y. \end{cases} \tag{2.2}$$

Note that these two formulations are equivalent [43,83]. This problem can be solved by applying the Lagrangian framework for optimization [48, 81]. We define $\Psi^y = (\psi_1^y, \ldots, \psi_{\ell_y}^y)$ and set up the Lagrange functional $\mathcal{L}$, with Lagrange multipliers $((\Lambda^y))_{ij} = \lambda_{ij}^y$, $i, j = 1, \ldots, \ell_y$ as

$$\mathcal{L}(\Psi^y, \Lambda^y) = \sum_{i=1}^{\ell_y} \langle y(t), \psi_i^y \rangle_{\mathcal{H}} \, \mathrm{d}t + \sum_{i=1}^{\ell_y} \sum_{j=1}^{\ell_y} \lambda_{ij}^y \langle \psi_i^y, \psi_j^y \rangle_{\mathcal{H}} - \delta_{ij}.$$

By computing the first and second order optimality conditions we end up with the integral operator $\mathcal{R}^y : \mathcal{H} \to \mathcal{H}$ given by

$$\mathcal{R}^y \psi^y = \int_0^T \langle y(t), \psi^y \rangle_{\mathcal{H}} \, y(t) \, \mathrm{d}t \quad \text{for} \quad \psi^y \in \mathcal{H} \tag{2.3}$$

and $y \in L^2(0, T; \mathcal{H})$. Clearly, $\mathcal{R}^y$ is a linear, bounded, nonnegative and self-adjoint operator. The obtained optimality results are summarized in the following proposition:

**Proposition 2.1.** *The solution to* (2.1) *is given by the eigenfunctions corresponding to the* $\ell_y \le d_y$ *largest eigenvalues* $\lambda_1^y \ge \lambda_2^y \ge \ldots \ge \lambda_{\ell_y}^y > 0$ *of the eigenvalue problem*

$$\mathcal{R}^y \psi_i^y = \int_0^T \langle y(t), \psi_i^y \rangle_{\mathcal{H}} \, y(t) \, \mathrm{d}t = \lambda_i^y \psi_i^y \qquad \textit{for} \quad i = 1, \ldots, \ell_y, \tag{2.4a}$$

$$\langle \psi_i^y, \psi_j^y \rangle_{\mathcal{H}} = \delta_{ij} \qquad \textit{for} \quad i, j = 1, \ldots, \ell_y. \tag{2.4b}$$

*In addition, we have the* POD approximation error

$$\int_0^T \Big\| y(t) - \sum_{i=1}^{\ell_y} \langle y(t), \psi_i^y \rangle_{\mathcal{H}} \, \psi_i^y \Big\|_{\mathcal{H}}^2 \, \mathrm{d}t = \sum_{i=\ell_y+1}^{d_y} \lambda_i^y.$$

For a detailed proof of the proposition we refer for instance to [43, 50, 83].

**Remark 2.2.** *The decay of the eigenvalues* $\{\lambda_i^y\}_{i=1}^{d_y}$ *is essential to obtain a good POD approximation. If the eigenvalues decay too slow, either the resulting reduced order model is of large dimension or one obtains large approximation errors.*

We shall utilize the POD basis $\{\psi_i^y\}_{i=1}^{\ell_y}$ with respect to $\mathcal{H} = H$ or $\mathcal{H} = V$. The POD-subspace for the variable $y$ is then denoted by

$$V^{\ell_y} = \operatorname{span}\{\psi_1^y, \ldots, \psi_{\ell^y}^y\}.$$

Note that $\psi_i^y \in V$ holds also for $\mathcal{H} = H$. This follows from (2.4a) using that $y \in L^2(0,T;V)$.

For the solution components $p$ and $q$ we follow the same approach as for $y$. Hence, we obtain POD bases $\{\psi_i^p\}_{i=1}^{\ell_p}$ and $\{\psi_i^q\}_{i=1}^{\ell_q}$ in an analogous way. We introduce indices and super-indices $y$, $p$ and $q$ for $\ell$ and $\psi$, respectively, to emphasize that the bases for the three solution components are computed independently and $\ell$ may be different for each of the components. This is done to obtain the best possible basis for each variable and to capture as much dynamics as possible. For the POD Galerkin scheme we make the ansatz

$$y^\ell(t) = \sum_{i=1}^{\ell_y} \mathrm{y}_i^\ell(t)\psi_i^y, \quad p^\ell(t) = \sum_{i=1}^{\ell_p} \mathrm{p}_i^\ell(t)\psi_i^p, \quad q^\ell(t) = \sum_{i=1}^{\ell_q} \mathrm{q}_i^\ell(t)\psi_i^q,$$

replace $y$, $p$ and $q$ in (1.4) by $y^\ell$, $p^\ell$ and $q^\ell$ and choose the POD basis functions as test functions. Hence, the POD Galerkin scheme is given by

$$\int_\Omega y_t^\ell(t)\varphi_i + c_1 y_{\mathbf{x}}^\ell(t)(\psi_i^y)' + \mathcal{N}(\mathbf{x}, y^\ell(t), p^\ell(t), q^\ell(t); \mu)\psi_i^y \,\mathrm{d}\mathbf{x} = 0, \tag{2.5a}$$

$$\int_\Omega c_2(y^\ell(t);\mu)p_{\mathbf{x}}^\ell(t)(\psi_i^p)' + \mathcal{N}(\mathbf{x}, y^\ell(t), p^\ell(t), q^\ell(t); \mu)\psi_i^p \,\mathrm{d}\mathbf{x} = 0, \tag{2.5b}$$

$$\int_\Omega c_3 q_{\mathbf{x}}^\ell(t)(\psi_i^q)' - \mathcal{N}(\mathbf{x}, y^\ell(t), p^\ell(t), q^\ell(t); \mu)\psi_i^q \,\mathrm{d}\mathbf{x} - \int_\Gamma \mathcal{I}(t)\psi_i^q \,\mathrm{d}S = 0, , \tag{2.5c}$$

for $1 \le i \le \ell_\wp$, $\wp \in \{y, p, q\}$, respectively and

$$\langle y^\ell(0), \psi_i^y\rangle_H = \langle y_\circ, \psi_i^y\rangle_H \quad \text{for} \quad 1 \le i \le \ell_y. \tag{2.5d}$$

System (2.5) is a reduced order model for (1.1). The presented approach is a Galerkin scheme and hence very similar to the finite element method presented in Section 1.3.1. We make use of the following assumption.

**Assumption 2.3.** *Suppose that for any $\mu \in \mathcal{M}_{ad}$ the system* (2.5) *admits a unique solution satisfying*

$$y^\ell(t) = \sum_{i=1}^{\ell_y} \mathrm{y}_i^\ell(t)\psi_i^y(\mathbf{x}) \ge y_{\min} \quad \textit{f.a.a.} \quad (t,\mathbf{x}) \in Q.$$

*Further, there exists a constant $C > 0$ independent of $\ell_y, \ell_p$ and $\ell_q$ such that the following estimates hold:*

$$\|p^\ell(t)\|_{L^\infty(\Omega)} \le Ce^{Ct}, \quad \|q^\ell(t)\|_{L^\infty(\Omega)} \le Ce^{Ct} \quad \textit{and} \quad y_{\min} = \frac{1}{C} \le y^\ell(t) \le C.$$

Utilizing the structure of the Galerkin ansatz we can write the reduced order model in

matrix-vector form. We define the matrices and vectors

$$\left(\left(\mathrm{M}_f^{\ell_\wp}\right)\right)_{ij} = \int_\Omega f(\mathbf{x})\psi_j^\wp\psi_i^\wp \,\mathrm{d}\mathbf{x}, \tag{2.6a}$$

$$\left(\left(\mathrm{S}_f^{\ell_\wp}\right)\right)_{ij} = \int_\Omega f(\mathbf{x})(\psi_j^\wp)'(\psi_i^\wp)' \,\mathrm{d}\mathbf{x}, \tag{2.6b}$$

$$\left(\mathcal{N}^{\ell_\wp}(\mathrm{y}^\ell(t),\mathrm{p}^\ell(t),\mathrm{q}^\ell(t);\mu)\right)_i = \int_\Omega \mathcal{N}(\mathbf{x},y^\ell(t),p^\ell(t),q^\ell(t);\mu)\psi_i^\wp \,\mathrm{d}\mathbf{x}, \tag{2.6c}$$

$$\left(\mathcal{I}^{\ell_q}(t)\right)_i = \int_\Gamma \mathcal{I}(t)\psi_i^q \,\mathrm{d}S, \tag{2.6d}$$

$$\left(\mathcal{Y}_0^{\ell_y}\right)_i = \int_\Omega y_\circ(\mathbf{x})\psi_i^y \,\mathrm{d}S, \tag{2.6e}$$

for $1 \le i,j \le \ell_\wp$, $\wp \in \{y,p,q\}$, $t \in [0,T]$, where $f$ denotes a positive weight function and

$$\mathrm{y}^\ell(t) = \left(\mathrm{y}_i^\ell(t)\right)_{1\le i\le \ell_y}, \quad \mathrm{p}^\ell(t) = \left(\mathrm{p}_i^\ell(t)\right)_{1\le i\le \ell_p}, \quad \mathrm{q}^\ell(t) = \left(\mathrm{q}_i^\ell(t)\right)_{1\le i\le \ell_q}.$$

Inserting the Galerkin ansatz into (2.5) we derive the nonlinear system

$$\mathrm{M}_1^{\ell_y}\mathrm{y}_t^\ell(t) + S_{c_1}^{\ell_y}\mathrm{y}^\ell(t) + \mathcal{N}^{\ell_y}(\mathrm{y}^\ell(t),\mathrm{p}^\ell(t),\mathrm{q}^\ell(t);\mu) = 0 \qquad \text{f.a.a.} \quad t\in[0,T], \tag{2.7a}$$

$$\mathrm{M}_1^{\ell_y}\mathrm{y}^\ell(0) = \mathcal{Y}_0^{\ell_y}, \tag{2.7b}$$

$$\mathrm{S}_{c_2(\mathrm{y}^\ell(t);\mu)}^{\ell_p}\mathrm{p}^\ell(t) + \mathcal{N}^{\ell_p}(\mathrm{y}^\ell(t),\mathrm{p}^\ell(t),\mathrm{q}^\ell(t);\mu) = 0 \qquad \text{f.a.a.} \quad t\in[0,T], \tag{2.7c}$$

$$\mathrm{S}_{c_3}^{\ell_q}\mathrm{q}^\ell(t) - \mathcal{N}^{\ell_q}(\mathrm{y}^\ell(t),\mathrm{p}^\ell(t),\mathrm{q}^\ell(t);\mu) = \mathcal{I}^{\ell_q}(t) \quad \text{f.a.a.} \quad t\in[0,T]. \tag{2.7d}$$

Due to Assumption 2.3 there exists a unique solution triple $\mathrm{z}^\ell = (\mathrm{y}^\ell,\mathrm{p}^\ell,\mathrm{q}^\ell)$ to (2.7) for every $\mu \in \mathcal{M}_{ad}$.

## 2.2 The discrete version of the POD method

To compute the POD bases $\{\psi_i^y\}_{i=1}^{\ell_y}$, $\{\psi_i^p\}_{i=1}^{\ell_p}$ and $\{\psi_i^q\}_{i=1}^{\ell_q}$ as described in Section 2.1 we need the snapshots $y(t,\mathbf{x})$, $p(t,\mathbf{x})$ and $q(t,\mathbf{x})$ for $t\in[0,T]$. This is realized numerically by computing approximations using a spatial and temporal discretization method. As in Section 2.1 the approach is illustrated in detail for the $y$-variable. The procedure for the $p$- and $q$-variable is analogous. First, we consider the case that the snapshots are given by finite element approximations $V^{N_\mathrm{x}} \ni y^N(t) \approx y(t) \in V$ for $t\in[0,T]$. In a second step we turn to the temporal discretization.

### 2.2.1 The spatial discretization

Suppose that Assumption 1.6 holds. Let the discrete solution operator for (1.8) be given by $\mathcal{S}^{N_\mathrm{x}}$. Then $\mathrm{z}^N(t) = (\mathrm{y}^N(t),\mathrm{p}^N(t),\mathrm{q}^N(t)) = \mathcal{S}^{N_\mathrm{x}}(\mu)$ is the solution triple for a given $\mu\in\mathcal{M}_{ad}$. We define the following weighted inner product in $\mathbb{R}^{dof}$:

$$\langle \mathrm{u}^N,\mathrm{v}^N\rangle_\mathrm{W} = (\mathrm{u}^N)^\top \mathrm{W}\mathrm{v}^N = \sum_{i=1}^{dof}\sum_{j=1}^{dof}\mathrm{u}_i^N\mathrm{W}_{ij}\mathrm{v}_j^N$$

for $\mathrm{u}^N = (\mathrm{u}_i^N)_{1\le i\le dof}$ and $\mathrm{v}^N = (\mathrm{v}_i^N)_{1\le i\le dof}$, where $\mathrm{W} = ((\mathrm{W}_{ij}))_{1\le i,j\le dof}$ denotes a symmetric, positive definite matrix. Moreover, we set $|\mathrm{u}^N|_{\mathrm{W}} = (\langle \mathrm{u}^N, \mathrm{u}^N\rangle_{\mathrm{W}})^{1/2}$.

**Remark 2.4.** *Suppose that* $\mathrm{u}^N = (\mathrm{u}_i^N)_{1\le i\le dof}$ *and* $\mathrm{v}^N = (\mathrm{v}_i^N)_{1\le i\le dof}$ *are two arbitrary vectors in* $\mathbb{R}^{dof}$*. Then,*

$$u^N(x) = \sum_{i=1}^{dof} \mathrm{u}_i^N \varphi_i(x) \quad \textit{and} \quad v^N(x) = \sum_{i=1}^{dof} \mathrm{v}_i^N \varphi_i(x)$$

*are elements in the finite element space* $V^{N_{\mathtt{x}}}$*. Further, we have*

$$\langle u^N, v^N\rangle_H = \langle \mathrm{u}^N, \mathrm{v}^N\rangle_{\mathrm{W}} \quad \textit{and} \quad \|u^N\|_H = |\mathrm{u}^N|_{\mathrm{W}}$$

*with* $\mathrm{W} = \mathrm{M}_1^N$*, where the symmetric, positive definite mass matrix has been introduced in Section 1.3. Analogously, we obtain*

$$\langle u^N, v^N\rangle_V = \langle \mathrm{u}^N, \mathrm{v}^N\rangle_{\mathrm{W}} \quad \textit{and} \quad \|u^N\|_V = |\mathrm{u}^N|_{\mathrm{W}}$$

*with* $\mathrm{W} = \mathrm{M}_1^N + \mathrm{S}_1^N$*, where the symmetric, positive definite stiffness matrix has been also introduced in Section 1.3. Summarizing, the weighted inner product* $\langle\cdot,\cdot\rangle_{\mathrm{W}}$ *is used to replace the inner products in the* $dof$*-dimensional finite element space* $V^{N_{\mathtt{x}}}$ *by an inner product in* $\mathbb{R}^{dof}$ *for the finite element nodal coefficients.* ◊

Let $\mathcal{V}^{y,N_{\mathtt{x}}} = \operatorname{span}\{\mathrm{y}^N(t)\,|\,t\in[0,T]\} \subset \mathbb{R}^{dof}$ and $\mathrm{d}_y = \dim \mathcal{V}^{y,N_{\mathtt{x}}} \le dof$. For any $\ell_y \in \{1,\dots,\mathrm{d}_y\}$ we construct a low-dimensional orthonormal basis by solving the optimization problem

$$\begin{cases} \min\limits_{\Psi_1^{y,N},\dots,\Psi_{\ell_y}^{y,N}\in\mathbb{R}^{dof}} \displaystyle\int_0^T \Big|\mathrm{y}^N(t) - \sum_{i=1}^{\ell_y} \langle \mathrm{y}^N(t), \Psi_i^{y,N}\rangle_{\mathrm{W}}\, \Psi_i^{y,N}\Big|_{\mathrm{W}}^2 \,\mathrm{d}t \\ \quad \text{s.t.} \quad \langle \Psi_i^{y,N}, \Psi_j^{y,N}\rangle_{\mathrm{W}} = \delta_{ij} \quad \text{for} \quad 1\le i,j\le \ell_y. \end{cases} \tag{2.8}$$

As for (2.1) the solution to (2.8) is given by an eigenvalue decomposition. In contrast to (2.3) let us define the linear, bounded, non-negative and self-adjoint operator $\mathcal{R}^{y,N} : \mathbb{R}^{dof} \to \mathbb{R}^{dof}$ by

$$\mathcal{R}^{y,N}\Psi^{y,N} = \int_0^T \langle \mathrm{y}^N(t), \Psi^{y,N}\rangle_{\mathrm{W}}\, \mathrm{y}^N(t)\,\mathrm{d}t \quad \text{for} \quad \Psi^{y,N}\in\mathbb{R}^{dof}.$$

Now the solution to (2.8) is given by the eigenvectors corresponding to the $\ell_y$ largest (positive) eigenvalues $\lambda_1^{y,N} \ge \lambda_2^{y,N} \ge \dots \ge \lambda_{\ell_y}^{y,N} \ge \dots \ge \lambda_{\mathrm{d}_y}^{y,N} > 0$ solving the $dof \times dof$ eigenvalue problem

$$\mathcal{R}^{y,N}\Psi_i^{y,N} = \lambda_i^{y,N}\Psi_i^{y,N} \quad \text{for} \quad i = 1,\dots,\ell_y. \tag{2.9}$$

Again, we can quantify the POD approximation error as

$$\int_0^T \Big\|\mathrm{y}^N(t) - \sum_{i=1}^{\ell_y} \langle \mathrm{y}^N(t), \Psi_i^{y,N}\rangle_{\mathrm{W}}\, \Psi_i^{y,N}\Big\|_{\mathrm{W}}^2 \,\mathrm{d}t = \sum_{i=\ell_y+1}^{\mathrm{d}_y} \lambda_i^{y,N}.$$

These results are analogous to the results obtained in the continuous case. For more details see, e.g. [43, 83]. For the solution components $\mathrm{p}^N$ and $\mathrm{q}^N$ we follow the same approach as for the component $\mathrm{y}^N$. Hence, we obtain POD bases $\{\Psi_i^{p,N}\}_{i=1}^{\ell_p}$ and $\{\Psi_i^{q,N}\}_{i=1}^{\ell_q}$, respectively.

Suppose that we have determined three POD bases $\{\Psi_i^{\wp,N}\}_{i=1}^{\ell_\wp}$ for $\wp \in \{y,p,q\}$. Let us define the matrices

$$\Psi^{\wp,N} = \left(\Psi_1^{\wp,N} \,\middle|\, \ldots \,\middle|\, \Psi_{\ell_\wp}^{\wp,N}\right) \in \mathbb{R}^{dof\times\ell_\wp} \quad \text{for} \quad \wp \in \{y,p,q\}.$$

As explained in Remark 2.4 the vectors $\Psi_i^\wp$, $\wp \in \{y,p,q\}$ can be interpreted as finite element coefficients of $\mathbf{x}$-dependent POD basis functions:

$$\psi_i^{\wp,N}(\mathbf{x}) = \sum_{j=1}^{dof} \left(\Psi_i^{\wp,N}\right)_j \varphi_j(\mathbf{x}) \quad \text{for} \quad i = 1,\ldots,\ell_\wp \quad \text{with} \quad \wp \in \{y,p,q\}.$$

This implies

$$\begin{aligned} y^{\ell,N}(t,\mathbf{x}) &= \sum_{i=1}^{\ell_y} \mathrm{y}_i^{\ell,N}(t)\psi_i^{y,N}(\mathbf{x}) \\ &= \sum_{i=1}^{\ell_y} \mathrm{y}_i^{\ell,N}(t) \sum_{j=1}^{dof} \left(\Psi_i^{y,N}\right)_j \varphi_j(x) = \sum_{j=1}^{dof} \left(\Psi^{y,N}\mathrm{y}^\ell(t)\right)_j \varphi_j(\mathbf{x}). \end{aligned} \tag{2.10}$$

Thus, the vector $\Psi^{y,N}\mathrm{y}^\ell(t)$ contains the finite element nodal coefficients of the function $y^{\ell,N}$ at time $t \in [0,T]$. An analogous result follows for the $p$- and $q$-variable. Moreover, the matrices and vectors for the reduced order model are then given by

$$\begin{aligned} \mathrm{M}_f^{\ell_\wp,N} &= \left(\Psi^{\wp,N}\right)^\top \mathrm{M}_f^N \Psi^{\wp,N}, \quad \mathrm{S}_f^{\ell_\wp,N} = \left(\Psi^{\wp,N}\right)^\top \mathrm{S}_f^N \Psi^{\wp,N}, \\ \mathcal{I}^{\ell_q,N}(t) &= \left(\Psi^{q,N}\right)^\top \mathcal{I}^N(t), \quad \mathcal{Y}_0^{\ell_y,N} = \left(\Psi^{y,N}\right)^\top \mathcal{Y}_0^N \end{aligned} \tag{2.11a}$$

and

$$\mathcal{N}^{\ell_\wp,N}(\mathrm{y}^\ell(t),\mathrm{p}^\ell(t),\mathrm{q}^\ell(t);\mu) = \left(\Psi^{\wp,N}\right)^\top \mathcal{N}^N\left(\Psi^{y,N}\mathrm{y}^\ell(t), \Psi^{p,N}\mathrm{p}^\ell(t), \Psi^{q,N}\mathrm{q}^\ell(t);\mu\right), \tag{2.11b}$$

for $\wp \in \{y,p,q\}$, where the finite element matrices and vectors are given by (1.7). The reduced order model then reads as

$$\mathrm{M}_1^{\ell_y,N}\mathrm{y}_t^{\ell,N}(t) + \mathrm{S}_{c_1}^{\ell_y,N}\mathrm{y}^{\ell,N}(t) + \mathcal{N}^{\ell_y,N}(\mathrm{y}^{\ell,N}(t),\mathrm{p}^{\ell,N}(t),\mathrm{q}^{\ell,N}(t);\mu) = 0, \tag{2.12a}$$

$$\mathrm{M}_1^{\ell_y,N}\mathrm{y}^{\ell,N}(0) = \mathcal{Y}_0^{\ell_y,N}, \tag{2.12b}$$

$$\mathrm{S}_{c_2(\mathrm{y}^{\ell,N}(t);\mu)}^{\ell_p,N}\mathrm{p}^{\ell,N}(t) + \mathcal{N}^{\ell_p,N}(\mathrm{y}^{\ell,N}(t),\mathrm{p}^{\ell,N}(t),\mathrm{q}^{\ell,N}(t);\mu) = 0, \tag{2.12c}$$

$$\mathrm{S}_{c_3}^{\ell_q,N}\mathrm{q}^{\ell,N}(t) - \mathcal{N}^{\ell_q,N}(\mathrm{y}^{\ell,N}(t),\mathrm{p}^{\ell,N}(t),\mathrm{q}^{\ell,N}(t);\mu) = \mathcal{I}^N(t), \tag{2.12d}$$

f.a.a. $t \in [0,T]$. This obtained reduced order model is based on the finite element model introduced in Section 1.3.1 but is of much smaller dimension. The model is obtained by the projection of the finite element matrices and vectors onto the subspace computed through the POD method. The system is of dimension $\ell_y + \ell_p + \ell_q$ opposed to the $3\,dof$ in the finite element case. If the eigenvalues decay fast this results in small systems that can be solved efficiently.

**Assumption 2.5.** *Suppose that for any $\mu \in \mathcal{M}_{ad}$ the system* (2.12) *admits a unique solution satisfying*

$$y^{\ell,N}(t) = \sum_{i=1}^{\ell_y} \mathrm{y}_i^{\ell,N}(t)\psi_i^y(\mathbf{x}) \geq y_{\min} \quad \textit{f.a.a.} \quad (t,\mathbf{x}) \in Q.$$

*Further, there exists a constant $C > 0$ independent of $h, \ell_y, \ell_p$ and $\ell_q$ such that the following estimates hold:*

$$\|p^{\ell,N}(t)\|_{L^\infty(\Omega)} \leq Ce^{Ct}, \quad \|q^{\ell,N}(t)\|_{L^\infty(\Omega)} \leq Ce^{Ct} \quad \textit{and} \quad y_{\min} = \frac{1}{C} \leq y^{\ell,N}(t) \leq C.$$

### 2.2.2 The temporal discretization

Let us now consider the more applied case. In practical computations, we do not have the whole trajectory $\mathcal{V}^{y,N_\mathtt{x}}$ at hand, but rather snapshots $\mathbb{R}^{dof} \ni \mathrm{y}^{N,k} \approx \mathrm{y}^N(t_j)$, $0 \leq j \leq N_t$, at discrete times

$$0 = t_0 < t_1 < \ldots < t_{N_t} = T.$$

This setting coincides with the results obtained by the finite element discretization introduced in Section 1.3.1. Note that we will reuse the notation from the previous subsection in order to avoid an overhead in the notation. We set $\mathcal{V}^{y,N_\mathtt{x}} = \operatorname{span}\{\mathrm{y}^{N,0}, \ldots, \mathrm{y}^{N,N_t}\} \subset \mathbb{R}^{dof}$ with $\mathrm{d}_y = \dim \mathcal{V}^{y,N_\mathtt{x}} \leq \min(dof, N_t)$. Then, a POD basis $\{\Psi_i^{y,N}\}_{i=1}^{\ell_y}$ is given by the solution to

$$\begin{cases} \min\limits_{\Psi_1^y,\ldots,\Psi_{\ell_y}^y \in \mathbb{R}^{dof}} \sum\limits_{k=0}^{N_t} \alpha_k \left| \mathrm{y}^{N,k} - \sum\limits_{i=1}^{\ell_y} \langle \mathrm{y}^{N,k}, \Psi_i^{y,N} \rangle_{\mathrm{W}} \, \Psi_i^{y,N} \right|_{\mathrm{W}}^2 \mathrm{d}t \\ \quad \text{s.t. } \langle \Psi_i^{y,N}, \Psi_j^{y,N} \rangle_{\mathrm{W}} = \delta_{ij} \quad \text{for} \quad 1 \leq i,j \leq \ell_y \end{cases} \tag{2.13}$$

with $\alpha_k$ nonnegative weights. We here set the $\alpha_k$ to the trapezoidal weights for the integration over the discretized time interval, i.e.

$$\alpha_0 = \frac{t_1 - t_0}{2}, \quad \alpha_k = \frac{t_{k+1} - t_{k-1}}{2}, \quad k = 1, \ldots, N_t - 1, \quad \text{and} \quad \alpha_{N_t} = \frac{t_{N_t} - t_{N_t-1}}{2}.$$

Let us introduce the matrices

$$\mathrm{D} = \operatorname{diag}\left(\alpha_0, \ldots, \alpha_{N_t}\right) \in \mathbb{R}^{(N_t+1)\times(N_t+1)} \quad \text{and} \quad \mathrm{Y}^N = \left(\mathrm{y}^{N,0} \,\middle|\, \ldots \,\middle|\, \mathrm{y}^{N,N_t}\right) \in \mathbb{R}^{dof\times(N_t+1)}.$$

Then we can write the operator $\mathcal{R}^{y,N}$ arising from the optimization problem (2.13) as

$$\mathcal{R}^{y,N}\Psi^{y,N} = \sum_{k=0}^{N_t} \alpha_k \langle \mathrm{y}^{N,k}, \Psi^{y,N} \rangle_{\mathrm{W}} \mathrm{y}^{N,k} = Y^N D \left(Y^N\right)^\top \mathrm{W}\Psi^{y,N} \quad \text{for } \Psi^{y,N} \in \mathbb{R}^{dof}$$

which leads to the unsymmetric eigenvalue problem

$$Y^N D \left(Y^N\right)^\top \mathrm{W}\Psi_i^{y,N} = \lambda^{y,N}\Psi_i^{y,N} \quad \text{for } i = 1, \ldots, \mathrm{d}_y. \tag{2.14}$$

This eigenvalue problem has to be solved in order to obtain the solution to the optimal control problem (2.13). In order to obtain a symmetric eigenvalue problem we introduce the matrix

$$\bar{\mathrm{Y}}^N = \mathrm{W}^{1/2}\mathrm{Y}^N\mathrm{D}^{1/2} \in \mathbb{R}^{dof\times(N_t+1)}.$$

Recall that W is supposed to be symmetric and positive definite. Thus, W possesses an eigenvalue decomposition of the form $\mathrm{W} = \mathrm{P}\Lambda\mathrm{P}^\top$, where $\Lambda$ is a diagonal matrix containing the positive eigenvalues of W and $\mathrm{P} \in \mathbb{R}^{dof\times dof}$ is an orthogonal matrix. Hence, we can define the square root of W by setting $\mathrm{W}^{1/2} = \mathrm{P}\Lambda^{1/2}\mathrm{P}^\top$, where $\Lambda^{1/2}$ is a diagonal matrix containing the square roots of the eigenvalues. Note that $\mathrm{D}^{1/2} = \operatorname{diag}(\alpha_0^{1/2},\dots,\alpha_{N_t}^{1/2})$ holds, but usually the computation of the square root $\mathrm{W}^{1/2}$ is more involved. Hence, the solution $\{\Psi_i^{y,N}\}_{i=1}^{\ell_y}$ to (2.13) can be determined by three different ways (see, e.g. [43, 83]):

1) Solve the symmetric $dof \times dof$ eigenvalue problem

$$\bar{\mathcal{R}}\mathrm{u}_i = \bar{\mathrm{Y}}^N\big(\bar{\mathrm{Y}}^N\big)^\top\mathrm{u}_i = \lambda_i^y\mathrm{u}_i, \quad 1 \le i \le \ell_y, \tag{2.15}$$

and set $\Psi_i^{y,N} = \mathrm{W}^{-1/2}\mathrm{u}_i$ for $1 \le i \le \ell_y$.

2) Determine the solution to the symmetric $(N_t+1)\times(N_t+1)$ eigenvalue problem

$$\bar{\mathcal{K}}\mathrm{v}_i = \big(\bar{\mathrm{Y}}^N\big)^\top\bar{\mathrm{Y}}^N\mathrm{v}_i = \lambda_i^y\mathrm{v}_i, \quad 1 \le i \le \ell_y, \tag{2.16}$$

and set $\Psi_i^{y,N} = \mathrm{Y}\mathrm{D}^{1/2}\mathrm{v}_i/\sqrt{\lambda_i^y}$ for $1 \le i \le \ell_y$.

3) Compute the singular value decomposition for $\bar{\mathrm{Y}}$, i.e.,

$$\mathrm{U}^\top\bar{\mathrm{Y}}\mathrm{V} = \left(\begin{array}{ccc|c} \sigma_1^y & & & \\ & \ddots & & 0 \\ & & \sigma_{\mathrm{d}_y}^y & \\ \hline & 0 & & 0 \end{array}\right)$$

with orthogonal matrices $\mathrm{U} = (\mathrm{u}_1^y\,|\,\dots\,|\,\mathrm{u}_{dof}^y) \in \mathbb{R}^{dof\times dof}$, $\mathrm{V} \in \mathbb{R}^{(N_t+1)\times(N_t+1)}$ and singular values $\sigma_1^y \ge \sigma_2^y \ge \dots \ge \sigma_{\mathrm{d}_y}^y > 0$. Then, set $\Psi_i^{y,N} = \mathrm{W}^{-1/2}\mathrm{u}_i$ for $1 \le i \le \ell_y$ as in 1) above.

Notice that $\Psi_i^{y,N} = \mathrm{W}^{-1/2}\mathrm{u}_i$ is realized by solving the linear system $\mathrm{W}^{1/2}\Psi_i^{y,N} = \mathrm{u}_i$. Moreover, $(\sigma_i^y)^2 = \lambda_i^y$ holds true for $1 \le i \le \mathrm{d}_y$. In the case $N_t + 1 \ll dof$ the approach 2) turns out to be the fastest strategy.

Analogous to the previous results the POD basis is given by the eigenvectors corresponding to the $\ell_y$ largest eigenvalues. Since this approach will be the one used in the numerical implementation let us state the following theorem [43, Theorem 2.7] for completeness.

**Theorem 2.6.** *Suppose that* $\{\lambda_i^{y,N}\}_{i=1}^{\mathrm{d}_y}$ *and* $\{\Psi_i^{y,N}\}_{i=1}^{\mathrm{d}_y}$ *denote the positive eigenvalues and associated orthonormal eigenvectors of* $\mathcal{R}^{y,N}$ *satisfying* (2.14)*. Then the optimal solution to* (2.13) *is given by the eigenvectors corresponding to the* $\ell_y \leq \mathrm{d}_y$ *largest eigenvalues. Moreover, the POD approximation error is given by*

$$\sum_{k=0}^{N_t} \alpha_k \left\| \mathrm{y}^{N,k} - \sum_{i=1}^{\ell_y} \langle \mathrm{y}^{N,k}, \Psi_i^{y,N} \rangle_{\mathrm{W}} \, \Psi_i^{y,N} \right\|_{\mathcal{H}}^2 \mathrm{d}t = \sum_{i=\ell_y+1}^{\mathrm{d}_y} \lambda_i^{y,N}.$$

For the proof of this theorem we refer the reader to [43]. Essentially, the proof is done by a finite induction over $\ell_y \in \{1, \ldots, \mathrm{d}_y\}$.

By applying the obtained POD basis to the system generated by the finite element discretization we can compute the reduced order model. The matrices in reduced form are given by (2.11) together with the reduced order model in semi-discrete form (2.12). Note that the time discretization is not indicated since this approach is independent of the underlying time integration method. For the numerical results the implicit Euler method, as introduced in Section 1.3.1, will be applied.

## 2.3 A-priori error estimates

In this section we present a-priori error estimates for the POD Galerkin schemes introduced in the previous sections of this chapter. For that purpose we also utilize the a-priori error estimate for the finite element model (1.8) from Section 1.3.1. As for the finite element case the proofs rely essentially on the properties of the nonlinear function $\mathcal{N}$; see Remark 1.3. Moreover, we require that the solutions are essentially bounded in $L^\infty(Q)$. In the context of the previous sections we choose $\mathcal{H} = V$ and $\mathcal{H} = V_a$ to compute POD bases for the $y$, $p$ and $q$ variables, respectively.

### 2.3.1 A-priori error estimates for the continuous POD model

For an arbitrarily chosen $\mu \in \mathcal{M}_{ad}$ let $z = \mathcal{S}(\mu)$ denote the unique solution to (1.1). We assume that Assumption 2.3 holds. As in the finite element case as presented in Section 1.3.1 we want to estimate the error

$$\int_0^T \|y(t) - y^\ell(t)\|_V^2 + \|p(t) - p^\ell(t)\|_V^2 + \|q(t) - q^\ell(t)\|_V^2 \, \mathrm{d}t.$$

**Remark 2.7.** *The proof of the a-priori error estimate works also for* $\Omega \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$. *For that reason we use a notation which is appropriate also for the three-dimensional case.*

The procedure to develop the a-priori error estimates is very similar to the techniques used in the finite element case. We will use the symmetric, bounded, coercive bilinear forms (1.10a) for $y$ and $q$ and (1.10c) for $p$ together with the coercivity conditions (1.10b) and (1.10d), respectively. We define the Ritz projection $\mathcal{P}^{\ell_y} : V \to V^{\ell_y}$ and $\mathcal{P}_a^{\ell_q} : V_a \to V_a^{\ell_q}$ by

$$a^y(\mathcal{P}^{\ell_y}\varphi, \psi) = a^y(\varphi, \psi) \quad \text{and} \quad a^q(\mathcal{P}_a^{\ell_q}\varphi, \psi) = a^q(\varphi, \psi) \tag{2.17a}$$

for all $\psi \in V^{\ell_y}$ and $\psi \in V_a^{\ell_q}$. Analogously, we introduce the mapping $\mathcal{P}^{\ell_p} : V \to V^{\ell_p}$ by

$$a^p(\mathcal{P}^{\ell_p}\varphi, \psi; \cdot) = a^p(\varphi, \psi; \cdot) \quad \text{for all} \quad \psi \in V^{\ell_p}. \tag{2.17b}$$

Using this projections we decompose the error in the $y$ component as

$$y(t) - y^\ell(t) = y(t) - \mathcal{P}^{\ell_y} y(t) + \mathcal{P}^{\ell_y} y(t) - y^\ell(t) = \varrho^{\ell_y}(t) + \vartheta^{\ell_y}(t) \tag{2.18a}$$

where $\varrho^{\ell_y}(t) = y(t) - \mathcal{P}^{\ell_y} y(t)$ and $\vartheta^{\ell_y}(t) = \mathcal{P}^{\ell_y} y(t) - y^\ell(t)$. Analogously, let

$$p(t) - p^\ell(t) = \big(p(t) - \mathcal{P}^{\ell_p} p(t)\big) + \big(\mathcal{P}^{\ell_p} p(t) - p^\ell(t)\big) = \varrho^{\ell_p}(t) + \vartheta^{\ell_p}(t), \tag{2.18b}$$

$$q(t) - q^\ell(t) = \big(q(t) - \mathcal{P}^{\ell_q} q(t)\big) + \big(\mathcal{P}^{\ell_q} q(t) - q^\ell(t)\big) = \varrho^{\ell_q}(t) + \vartheta^{\ell_q}(t). \tag{2.18c}$$

The next result characterizing the projection error follow from the construction of the POD basis [56, Lemma 3].

**Lemma 2.8.** *The approximation error terms $\varrho^{\ell_y}$, $\varrho^{\ell_p}$ and $\varrho^{\ell_p}$ satisfy*

$$\int_0^T \|\varrho^{\ell_y}(t)\|_V^2 \,\mathrm{d}t = \int_0^T \|y(t) - \mathcal{P}^{\ell_y} y(t)\|_V^2 \,\mathrm{d}t = \sum_{i=\ell_y+1}^{d_y} \lambda_i^y,$$

$$\int_0^T \|\varrho^{\ell_p}(t)\|_V^2 \,\mathrm{d}t = \int_0^T \|p(t) - \mathcal{P}^{\ell_p} p(t)\|_V^2 \,\mathrm{d}t = \sum_{i=\ell_p+1}^{d_p} \lambda_i^p,$$

$$\int_0^T \|\varrho^{\ell_q}(t)\|_V^2 \,\mathrm{d}t = \int_0^T \|q(t) - \mathcal{P}^{\ell_q} q(t)\|_V^2 \,\mathrm{d}t = \sum_{i=\ell_q+1}^{d_q} \lambda_i^q.$$

In the next lemma we present an error estimate for the discretization error terms $\vartheta^{\ell_p}(t)$ and $\vartheta^{\ell_q}(t)$.

**Lemma 2.9.** *Suppose that Assumption 2.3 holds. Then there exists a constant $C > 0$ independent of $\ell_y, \ell_p, \ell_q$ and $t \in [0, T]$ so that*

$$\|\vartheta^{\ell_p}(t)\|_V^2 + \|\vartheta^{\ell_q}(t)\|_V^2 \le C\big(\|\varrho^{\ell_y}(t)\|_V^2 + \|\varrho^{\ell_p}(t)\|_V^2 + \|\varrho^{\ell_q}(t)\|_V^2 + \|\vartheta^{\ell_y}(t)\|_H^2\big) \tag{2.19}$$

*f.a.a. $t \in [0, T]$.*

The proof of this lemma is analogous to the proof of Lemma 1.9. It is based on (1.4), (2.5) and the properties of the projections $\mathcal{P}^{\ell_p}$, $\mathcal{P}^{\ell_q}$. The first three terms on the right-hand side of (2.19) shall be bounded by using Lemma 2.8. From the parabolic equation an estimate is derived for the term $\|\vartheta^{\ell_y}(t)\|_H^2$. Let us mention that it is essential that $\vartheta^{\ell_y}(t)$ occurs in the $H$- and not in the $V$-norm. Now we turn to an estimate for the difference $\vartheta^{\ell y}(t) = \mathcal{P}^{\ell_y} y(t) - y^\ell(t)$.

**Lemma 2.10.** *Suppose that Assumption 2.3 holds. Then there exists a constant $C > 0$ independent of $\ell_y, \ell_p, \ell_q$ and $t \in [0,T]$ so that*

$$\begin{aligned}\|\vartheta^{\ell_y}\|^2_{L^\infty(0,T;H)} + \|\vartheta^{\ell_y}\|^2_{L^2(0,T;V)} &\\ \leq C\big(\|\mathcal{P}^{\ell_y} y_\circ - y^\ell(0)\|^2_H + \|\mathcal{P}^{\ell_y} y_t - y_t\|^2_{L^2(0,T;H)}\big) &\\ + C\bigg(\sum_{i=\ell_y+1}^{d_y} \lambda_i^y + \sum_{i=\ell_p+1}^{d_p} \lambda_i^p + \sum_{i=\ell_q+1}^{d_q} \lambda_i^q\bigg).\end{aligned} \tag{2.20}$$

*Proof.* The proof is essentially the same as for Lemma 1.10. Let us point out the differences. While we have an a-priori error estimator for the time derivative in the finite element case, Lemma 1.8, this is missing in the POD case, Lemma 2.8. Hence, starting from (1.17) of the proof for Lemma 1.10 we get

$$\begin{aligned}\|\vartheta^{\ell_y}(t)\|^2_H + 2\int_0^t c_V \|\vartheta^{\ell_y}(s)\|^2_V \,\mathrm{d}s &\\ \leq C_3\bigg(\sum_{i=\ell_y+1}^{d_y} \lambda_i^y + \sum_{i=\ell_p+1}^{d_p} \lambda_i^p + \sum_{i=\ell_q+1}^{d_q} \lambda_i^q\bigg) + \|\mathcal{P}^{\ell_y} y_\circ - y^\ell(0)\|^2_H &\\ + C_2\left(\|\mathcal{P}^{\ell_y} y_t - y_t\|^2_{L^2(0,T;H)} + \int_0^t \|\vartheta^{\ell_y}(s)\|^2_H \,\mathrm{d}s\right)\end{aligned} \tag{2.21}$$

which by using Gronwall's inequality [41, 79] gives

$$\begin{aligned}\|\vartheta^{\ell_y}(t)\|^2_H \leq C_2\bigg(\sum_{i=\ell_y+1}^{d_y} \lambda_i^y + \sum_{i=\ell_p+1}^{d_p} \lambda_i^p + \sum_{i=\ell_q+1}^{d_q} \lambda_i^q\bigg) &\\ + C_2\bigg(\|\mathcal{P}^{\ell_y} y_\circ - y^\ell(0)\|^2_H + \|\mathcal{P}^{\ell_y} y_t - y_t\|^2_{L^2(0,T;H)}\bigg)\end{aligned}$$

f.a.a. $t \in [0,T]$ and

$$\begin{aligned}\|\vartheta^{\ell_y}(t)\|^2_{L^2(0,T;V)} \leq C_2\bigg(\sum_{i=\ell_y+1}^{\infty} \lambda_i^y + \sum_{i=\ell_p+1}^{\infty} \lambda_i^p + \sum_{i=\ell_q+1}^{\infty} \lambda_i^q\bigg) &\\ + C_2\bigg(\|\mathcal{P}^{\ell_y} y_\circ - y^\ell(0)\|^2_H + \|\mathcal{P}^{\ell_y} y_t - y_t\|^2_{L^2(0,T;H)}\bigg).\end{aligned}$$

This concludes the proof. □

From Lemmas 2.8, 2.9 and 2.10 we infer the following theorem.

**Theorem 2.11.** *Let $\mu \in \mathcal{M}_{ad}$ be chosen arbitrarily and the POD bases be generated as described in Section 2.1. Suppose that Assumption 2.3 is valid. Then there exists a constant*

*$C > 0$ independent of $\ell_y$, $\ell_p$, $\ell_q$ and $\mu$ such that*

$$\begin{aligned}\int_0^T \|y(t) - y^\ell(t)\|_V^2 + \|p(t) - p^\ell(t)\|_V^2 + \|q(t) - q^\ell(t)\|_V^2 \,\mathrm{d}t \\ \leq C\big(\|\mathcal{P}^{\ell_y} y_\circ - y^\ell(0)\|_H^2 + \|\mathcal{P}^{\ell_y} y_t - y_t\|_{L^2(0,T;H)}^2\big) \\ + C\bigg(\sum_{i=\ell_y+1}^{d_y} \lambda_i^y + \sum_{i=\ell_p+1}^{d_p} \lambda_i^p + \sum_{i=\ell_q+1}^{d_q} \lambda_i^q\bigg).\end{aligned}$$

**Remark 2.12.** *1. The term $\|\mathcal{P}^{\ell_y} y_\circ - y^\ell(0)\|_H^2$ expresses how well the initial condition is approximated by the POD basis. It follows from [49, Remark 3.3] that*

$$\|\mathcal{P}^{\ell_y} y_\circ - y^\ell(0)\|_H^2 \leq \|(\mathcal{P}^{\ell_y} - \mathcal{T}^{\ell_y}) y_\circ\|_H^2 \overset{\ell_y \to \infty}{\longrightarrow} 0,$$

*where $\mathcal{T}^{\ell_y}\varphi = \sum_{i=1}^{\ell_y} \langle \varphi, \psi_i^y \rangle_H \psi_i^y$ for $\varphi \in H$.*

*2. Note that the error depends on the $L^2(0,T;H)$ norm of the difference $\mathcal{P}^{\ell_y} y_t - y_t$. To avoid this dependence we have to include time derivatives into our snapshot set, see [56–58, 74]. Utilizing techniques introduced in [15, 76] an a-priori error bound can be derived without including the time derivatives into the snapshot subspace.*

### 2.3.2 A-priori error estimates for the discrete POD model

We choose an arbitrary $\mu \in \mathcal{M}_{ad}$. Suppose that Assumptions 1.6 and 2.5 hold. Let $\mathrm{z}^N = (\mathrm{y}^N, \mathrm{p}^N, \mathrm{q}^N)$ be the associated solution to (1.8). Next we compute POD bases as described in Section 2.3.2. By $\mathrm{z}^{\ell,N} = (\mathrm{y}^{\ell,N}, \mathrm{p}^{\ell,N}, \mathrm{q}^{\ell,N})$ we denote the associated solution to (2.7) for the parameter $\mu$. As in Sections 1.3.1 and 2.3.2 we write $z^N = (y^N, p^N, q^N)$ and $z^{\ell,N} = (y^{\ell,N}, p^{\ell,N}, q^{\ell,N})$ for the associated finite element representations. Then we formulate the next theorem.

**Theorem 2.13.** *Let $\mu \in \mathcal{M}_{ad}$ be arbitrary. Assume that Assumptions 1.6 and 2.5 hold. Furthermore, the POD bases are computed as described in Section 2.2.1. Then there exists a constant $C > 0$ independent of $h$, $\ell_y$, $\ell_p$, $\ell_q$ and $\mu$ satisfying*

$$\begin{aligned}\int_0^T \|y^N(t) - y^{\ell,N}(t)\|_V^2 + \|p^N(t) - p^{\ell,N}(t)\|_V^2 + \|q^N(t) - q^{\ell,N}(t)\|_V^2 \,\mathrm{d}t \\ \leq C\big(\|\mathcal{P}^{\ell_y} y^N(0) - y^{\ell,N}(0)\|_H^2 + \|\mathcal{P}^{\ell_y} y_t^N - y_t^N\|_{L^2(0,T;H)}^2\big) \\ + C\bigg(\sum_{i=\ell_y+1}^{\mathrm{d}_y} \lambda_i^{y,N} + \sum_{i=\ell_p+1}^{\mathrm{d}_p} \lambda_i^{p,N} + \sum_{i=\ell_q+1}^{\mathrm{d}_q} \lambda_i^{q,N}\bigg).\end{aligned}$$

*Proof.* The proof follows the arguments of the proof of Theorem 1.11 or 2.11. Note that (1.10) is valid if we replace the spaces $V$ and $V_a$ by their corresponding finite element subspaces, which are endowed with the same topology as in $V$ and $V_a$, respectively. Then the claim follows since $z^N$ and $z^{\ell,N}$ are essentially bounded on $Q$ by Assumptions 1.6 and 2.5. □

**Remark 2.14.** *Let Assumptions 1.6 and 2.5 hold. We choose an arbitrary $\mu \in \mathcal{M}_{ad}$. Note that*

$$\int_0^T \|y(t) - y^{\ell,N}(t)\|_V^2 \,\mathrm{d}t \leq 2\int_0^T \|y(t) - y^N(t)\|_V^2 \,\mathrm{d}t + 2\int_0^T \|y^N(t) - y^{\ell,N}(t)\|_V^2 \,\mathrm{d}t.$$

*Thus, the error between a solution to the original variational system* (1.4) *and its POD-Galerkin approximation built on* (2.7) *with the settings in (2.6) can be bounded by applying Theorems 1.11 and 2.13. In fact, we find*

$$\begin{aligned}
&\int_0^T \|y(t) - y^{\ell,N}(t)\|_V^2 \,\mathrm{d}t + \|p^N(t) - p^{\ell,N}(t)\|_V^2 + \|q^N(t) - q^{\ell,N}(t)\|_V^2 \,\mathrm{d}t \\
&\quad \leq C\Big(\|\mathcal{P}^{N_{\mathbf{x}},y} y_\circ - y^N(0)\|_H^2 + h^s + \|\mathcal{P}^{\ell_y} y^N(0) - y^{\ell,N}(0)\|_H^2 + \|\mathcal{P}^{\ell_y} y_t^N - y_t^N\|_{L^2(0,T;H)}^2\Big) \\
&\qquad + C\Big(\sum_{i=\ell_y+1}^{\mathrm{d}_y} \lambda_i^{y,N} + \sum_{i=\ell_p+1}^{\mathrm{d}_p} \lambda_i^{p,N} + \sum_{i=\ell_q+1}^{\mathrm{d}_q} \lambda_i^{q,N}\Big)
\end{aligned}$$

*with a constant $C > 0$ independent of $h$, $\ell_y$, $\ell_p$, $\ell_q$ and $\mu$ and $s = 1$ for piecewise linear and $s = 2$ for piecewise quadratic finite element ansatz functions. Analogously, we can derive error estimates for the differences $p - p^{\ell,N}$ and $q - q^{\ell,N}$.*

### 2.3.3 Full discrete POD Galerkin schemes

The previous a-priori error estimates are derived for models which are not discretized in the time variable. As in Section 1.3.1 we do not investigate the time discretization. To extend the analysis to the full discrete case one has to utilize the techniques in [56,58] and [57,74] for the implicit Euler method and the Crank-Nicolson scheme, respectively. Since the approach is not new, but the technical details are quite involved, we do not include this a-priori error analysis into this work.

## 2.4 POD basis computation

In this section we want to present some results related to the POD basis computation. Different techniques for the basis generation are outlined. A comparison of the approximation errors is made. Additionally, some outlines for an improved POD basis computation are given. Finally, the empirical interpolation method is introduced in order to overcome the numerical complexity when evaluating the nonlinear term in the reduced order model.

### 2.4.1 Single snapshot set

First we have a look at the POD basis computation as described in Section 2.2.2. For the POD basis computation we set $\mathcal{H} = V$, i.e. $\mathrm{W} = \mathrm{S}_1^N + \mathrm{M}_1^N$ in (2.13). When $\mathcal{H} = H$ similar results are obtained. We present numerical results utilizing the eigenvalue decomposition of $\bar{\mathcal{K}}$ and the singular value decomposition of $\bar{\mathrm{Y}}$. Let the finite element solution be given

by the settings introduced in Section 1.3.3 utilizing parameter $\bar{\mu}$. First we investigate the eigenvalue decomposition of $\bar{\mathcal{K}}$ to determine the POD bases. In the left plot of Figure 2.1 the decay of the normalized eigenvalues $\{\lambda_i^{\wp,N}\}_{i=1}^{50}$, $\wp \in \{y, p, q\}$, is shown. Utilizing the

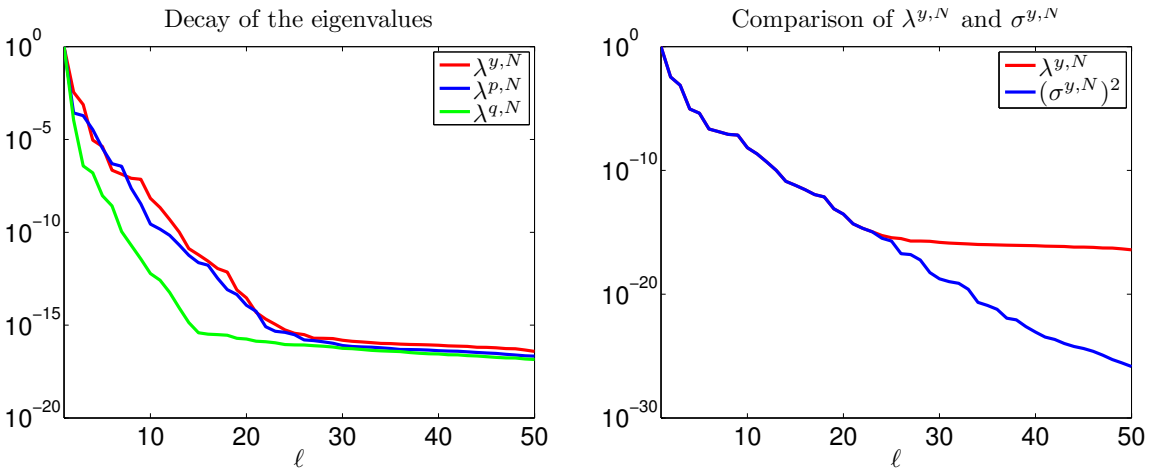


Figure 2.1: Decay of the normalized eigenvalues for the POD basis for $y$, $p$ and $q$ (left) and a comparison of the decay of the singular values and eigenvalues for $y$ (right).

singular values decomposition we compute the singular values $\{\sigma_i^{y,N}\}_{i=1}^{50}$; see right plot of Figure 2.1. In the comparison it can be seen that the first values $(\sigma_i^{\wp,N})^2$ and $\lambda_i^{\wp,N}$ coincide. The arising difference after approximately 20 values is a numerical issue since both methods work to the same accuracy. The difference only occurs since the singular values are squared for comparison. This issue is not influencing the POD method since we are only interested in the largest eigenvalues with $\ell_y, \ell_p$ and $\ell_q$ small. Hence, the gain in accuracy by using the singular value decomposition is not significant. The kink in Figure 2.1 (left) can be seen as a guess for the number of basis functions needed for a good approximation. In Figure 2.2 we compare the first six POD bases for the variables $y, p$ and $q$ obtained by the eigenvalue and singular value decomposition. It can be seen that they are more or less the same. The only difference that can be spotted is the sign difference in some cases. For a better comparison of the two approaches we have a look at the approximation errors. For this we introduce the following errors:

$$err_{\lambda}^{\wp}(\ell) = \sqrt{\sum_{i=\ell+1}^{\mathrm{d}_{\wp}} \lambda_i^{\wp,N}} \quad \text{and} \quad err_{\mathcal{P}}^{\wp}(\ell) = \sqrt{\sum_{k=0}^{N_t} \alpha_k \left| \mathrm{y}^{N,k} - \sum_{i=1}^{\ell} \langle \mathrm{y}^{N,k}, \Psi_i^{y,N} \rangle_{\mathrm{W}} \, \Psi_i^{y,N} \right|_{\mathrm{W}}^2}$$

for $\wp \in \{y, p, q\}$ and the $\alpha_k$ the trapezoidal weights as introduced in Section 2.2.2. Note that $err_{\lambda}^{\wp}(\ell_{\wp})$ can also be computed using the trace for the matrix $\bar{\mathcal{K}}$, i.e.

$$err_{\lambda}^{\wp}(\ell) = \sqrt{\mathrm{trace}(\bar{\mathcal{K}}) - \sum_{i=1}^{\ell} \lambda_i^{\wp,N}}.$$

We compare the approximation errors for the three variables using the POD bases obtained by the eigenvalue and the singular value decomposition. In Figure 2.3 the decay is shown.

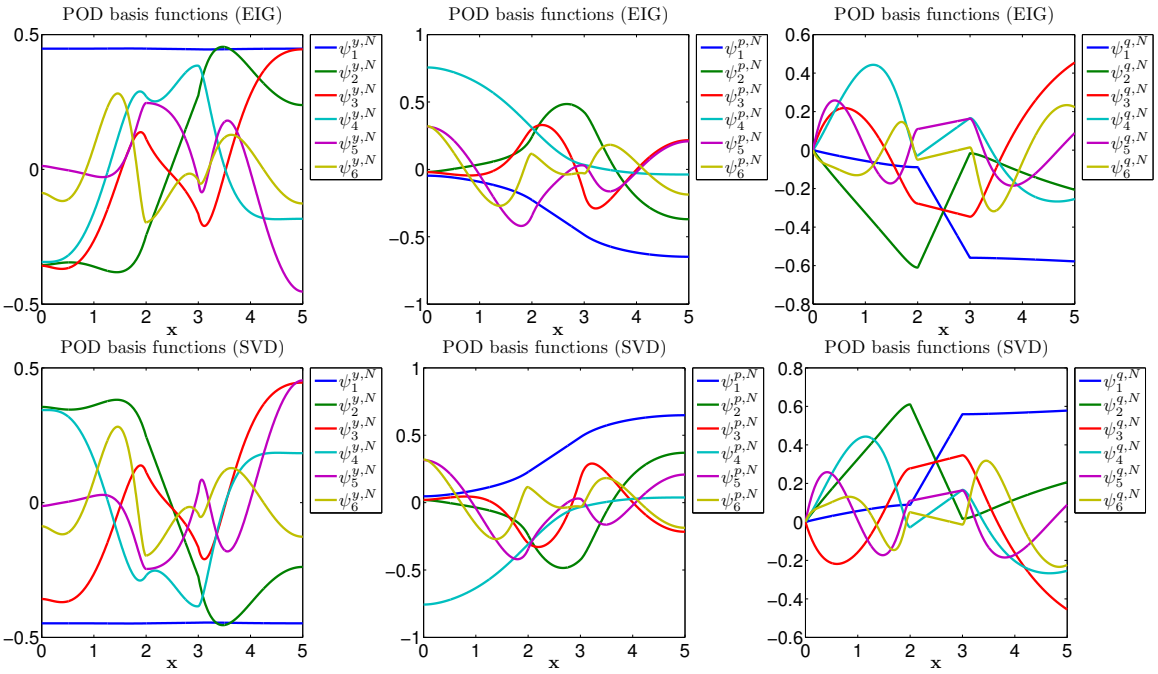


Figure 2.2: The first six POD basis functions for $y$ (left), $p$ (center) and $q$ (right) computed using the matrix $\bar{\mathcal{K}}$ (top) and the singular value decomposition (bottom).

It can be observed that both methods deliver very similar results. The higher computational accuracy from the singular value decomposition has no influence in the approximation errors. In the case that only a few basis functions are required the two presented approaches deliver essentially the same results. Comparing it to the theoretical result, i.e. $err_{\lambda}^{\wp}(\ell_{\wp})$, it can be observed that it can be matched until the accuracy of the eigenvalue solver is exceeded. Since we compute the error by subtracting the eigenvalues from the trace we end up with a plateau. This is caused by the fact that the sum is not equal to the trace but a deviation of approximately $10^{-16}$. This corresponds to $10^{-8}$ in the error computation since a square root is applied. Further, note that the accuracy in the projection error is of similar magnitude as the projection error of the finite element method.

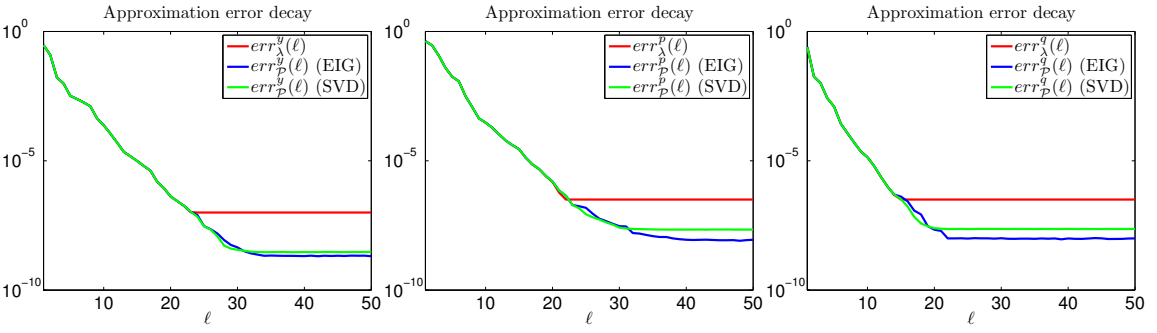


Figure 2.3: Comparison of the decay of the approximation errors for the three variables $y$ (left), $p$ (center) and $q$ (right).

**Remark 2.15.** *To guarantee numerically that the obtained basis is orthonormal applying an orthonormalization strategy after computing the POD basis is advised. Then it is ensured that the condition*

$$\langle \Psi_i^{\wp,N}, \Psi_j^{\wp,N} \rangle_W = \delta_{ij}, \quad \textit{for} \quad 1 \leq i,j \leq \ell_\wp$$

*and $\wp \in \{y,p,q\}$ is satisfied. In the presented results the Gram-Schmidt process [32, 38] is utilized to orthonormalize the obtained POD basis. Alternatively also methods using the QR-method can be applied at a higher computational cost [38, 78]. The improvement of the basis is outlined in Figure 2.4.*

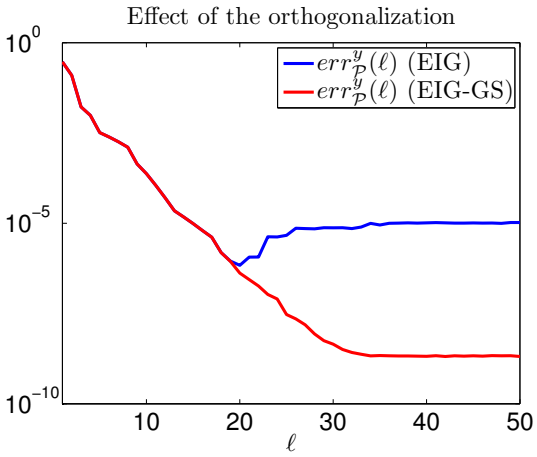


Figure 2.4: Comparison of the approximation error $err^y_{\mathcal{P}}(\ell)$ for $y$ when applying the Gram-Schmidt orthogonalization (red) and when utilizing the basis obtained directly by the eigenvalue solver (blue).

Looking at the computational cost let us mention that the POD basis computation using the singular value decomposition is more expensive than the eigenvalue decomposition. This is due to the need of the matrix $W^{1/2}$ and $W^{-1/2}$. In practice not the whole eigenvalue or singular value decomposition has to be computed but it is enough to compute the first $\ell$ elements. Hence, iterative algorithms like the Lanczos algorithm [26, 38] can be applied. These methods deliver very good approximations and are computationally more affordable.

**Remark 2.16.** *When computing the eigenvalues and eigenvectors using the* `eigs` *command in* MATLAB *only the largest $\ell$ eigenvalues and eigenvectors can be computed. To obtain better results it turns out that computing $\ell + 5$ eigenvalues and eigenvectors and using only $\ell$ generates better numerical results. This is done in all the presented numerical experiments.*

The POD basis can be further enhanced by improving the time discretization. This can be done by an optimal snapshot location approach [59]. This approach has been applied to the nonlinear model (1.1) with $c_2$ piecewise constant and positive. To avoid a computational

overhead linear combination of the snapshots were chosen to generate new snapshots. The results are very promising [9]. This approach can also be extended to an adaptive strategy [63], where the POD basis is enhanced by adding information about the system behavior when the parameter is varied.

### 2.4.2 Multiple snapshot sets

Let us now consider the case that multiple snapshot sets are available. In the previous section we look at the case for a single snapshot set. Here different values for the parameter $\mu$ are considered. This is the case if already several simulations of the nonlinear system have been carried out for different parameter settings of $\mu$. Now all this data should be used to generate the best possible reduced order model. We denote the snapshot sets as $\mathrm{Y}_j = [y_{\mu^j}^{N,0}, \ldots, y_{\mu^j}^{N,N_t}]$ for $i = 1, \ldots, N_s$. Note that we indicate the dependency of $y^{N,k}$ on the parameter $\mu^j$ explicitly by a subindex for clarity. Let us assume that $N_t$ is the same for all the snapshot sets. Further, we assume that the spatial discretization is the same for all the samples. A strategy to treat adaptive meshes is outlined in the next section. These are not constraints but rather simplifications for the notation. We want to look at two approaches for the computation of the POD basis. One approach is to apply the POD method straight to the snapshots while the other uses a greedy algorithm, motivated by the reduced basis method [11, 39, 44–46].

Let us start with POD method applied to multiple snapshot sets. The idea is to treat all the sets equally by considering them as one set. For this let us introduce $\mathbb{Y} = [\mathrm{Y}_1, \ldots, \mathrm{Y}_{N_s}]$. Then the eigenvalue approach delivers the matrix

$$\tilde{\mathcal{K}} = \tilde{\mathrm{D}}^{\frac{1}{2}} \mathbb{Y}^\top \mathrm{W} \mathbb{Y} \tilde{\mathrm{D}}^{\frac{1}{2}} \in \mathbb{R}^{N_s(N_t+1) \times N_s(N_t+1)},$$

where $\tilde{\mathrm{D}} = \mathrm{diag}(\mathrm{D}, \ldots, \mathrm{D}) \in \mathbb{R}^{N_s(N_t+1) \times N_s(N_t+1)}$ is a block diagonal matrix and $D$ as previously introduced. The matrix $\tilde{\mathcal{K}}$ can become very large if $N_s$ is large and $dof \ll (N_t + 1)N_s$ holds. For this reason using the approach

$$\tilde{\mathcal{R}} = \mathrm{W}^{\frac{1}{2}} \mathbb{Y} \tilde{\mathrm{D}} \mathbb{Y}^\top \mathrm{W}^{\frac{1}{2}} \in \mathbb{R}^{dof \times dof},$$

can be a better choice and lead to smaller dimensions. Still the matrices $\mathrm{W}^{1/2}$ and $\mathrm{W}^{-1/2}$ are needed but due to the high number of snapshots the computational cost can become negligible compared to the approach using the $\tilde{\mathcal{K}}$ matrix. Alternatively, also the singular value decomposition can be applied as described previously. The POD basis will be obtained as described before. In the numerical results we will focus on applying the eigenvalue decomposition since it delivers accurate results for the first POD bases and we assume that not so many will be needed. These approaches suffer from the problem of computing the matrices $\tilde{\mathcal{K}}$ or $\tilde{\mathcal{R}}$. The computational cost are very high when $N_s$ is large and might therefore not be feasible in some cases.

Let us now look at the greedy type approach. Computing POD basis functions for the reduced order model utilizing the greedy method is a standard approach in the reduced basis method for parametrized systems. In the case of time dependent problems this technique is combined with the POD method [39, 44, 45]. In the presented case the greedy

algorithm will be used to extract a basis from the given snapshot sets. This procedure belongs to the group of *strong* greedy procedures. Compared to the more frequently used *weak* greedy procedures this method does not benefit from rapid computable error estimators [11, 39, 70]. In the presented approach the true projection error is used as an error indicator. The advantage of this strategy is that the equation under investigation can easily be exchanged without changing the algorithm. Further, since it is used to extract a basis of already computed snapshots it is also computationally efficient. The presented approach is similar to [46]. The main difference to the standard reduced basis method is that the snapshot sets are given and are not computed during the procedure. By using a greedy technique the basis computation is split into subproblems and the basis $\{\psi_i^{y,N}\}_{i=1}^{\ell_y}$ will be generated iteratively. Given $N_s$ snapshot sets, in every iteration the worst approximated snapshot set is chosen to enrich the POD basis. As an indicator we use the projection error over the space-time cylinder, i.e. $Q = \Omega \times [\mathrm{t}_0, \mathrm{t}_{N_t}]$. This approach has the advantage that no further information other than the given data is required.

**Algorithm 3** (The POD-Greedy Algorithm)

**Require:** $N_s$ snapshot sets $\mathrm{Y}_k = [\mathrm{y}^{N,0}, \ldots, \mathrm{y}^{N,N_t}]$ associated to different parameters $\mu_k$, tolerance $\epsilon^{\mathrm{POD}}$ for the projection error

1: $\Psi \leftarrow POD_L(\mathrm{Y}_1), \quad \ell_y = L$

2: Compute projection error for all $N_s$ snapshot sets

$$\mathrm{E}_j = \|\mathrm{Y}_j - \mathcal{P}_\Psi(\mathrm{Y}_j)\|^2_{(\mathrm{D},\mathrm{W})}, \quad j = 1, \ldots, N_s$$

3: **while** $\max(\mathrm{E}_k) > \varepsilon^{\mathrm{POD}}$ **do**

4: $\quad k \leftarrow \arg\max_{j=1,\ldots,N_s} \mathrm{E}_j$

5: $\quad \overline{\Psi} \leftarrow POD_L(\mathrm{Y}_k - \mathcal{P}_\Psi(\mathrm{Y}_k)), \quad \Psi \leftarrow \Psi \cup \overline{\Psi}, \quad \ell_y = \ell_y + L$

6: $\quad$ Compute projection errors $\mathrm{E}_j$ for all $N_s$ snapshot sets using $\Psi$

7: **end while**

8: **return** POD basis $\Psi$

The detailed algorithm is given in Algorithm 3. By $\mathcal{P}_\Psi$ the orthogonal projection on the subspace spanned by $\Psi$ with respect to the inner product W is denoted, i.e. $\langle Y_j, \Psi\rangle_W \Psi$. The norm $\|\cdot\|^2_{(\mathrm{D},\mathrm{W})}$ equals to

$$\|\mathrm{Y}_j - \mathcal{P}_\Psi(\mathrm{Y}_j)\|^2_{(\mathrm{D},\mathrm{W})} = \sum_{k=0}^{N_t} \alpha_k \left| \mathrm{y}_{\mu^j}^{N,k} - \sum_{i=1}^{\ell_y} \langle \mathrm{y}_{\mu^j}^{N,k}, \Psi_i^{y,N} \rangle_{\mathrm{W}} \, \Psi_i^{y,N} \right|^2_{\mathrm{W}}.$$

This corresponds to the discrete version of the norm $\|\cdot\|^2_{L^2(0,T;V)}$. Note that this is just the previously introduced approximation error $err^y_{\mathcal{P}}(\ell_y)$ for the snapshot set $\mathrm{Y}_j$. Let us denote this error by $err^y_{\mathcal{P}}(\ell_y, j)$ in order to have a short notation. Further, by $POD_L$ the routine extracting the POD basis functions corresponding to the $L$ largest eigenvalues or singular values obtained by the best fit property (2.13) is denoted. Here either of the approaches introduced in Section 2.4.1 can be used to obtain the POD basis. For the numerical experiments we again utilize the approach using the matrix $\bar{\mathcal{K}}$ and the eigenvalue decomposition.

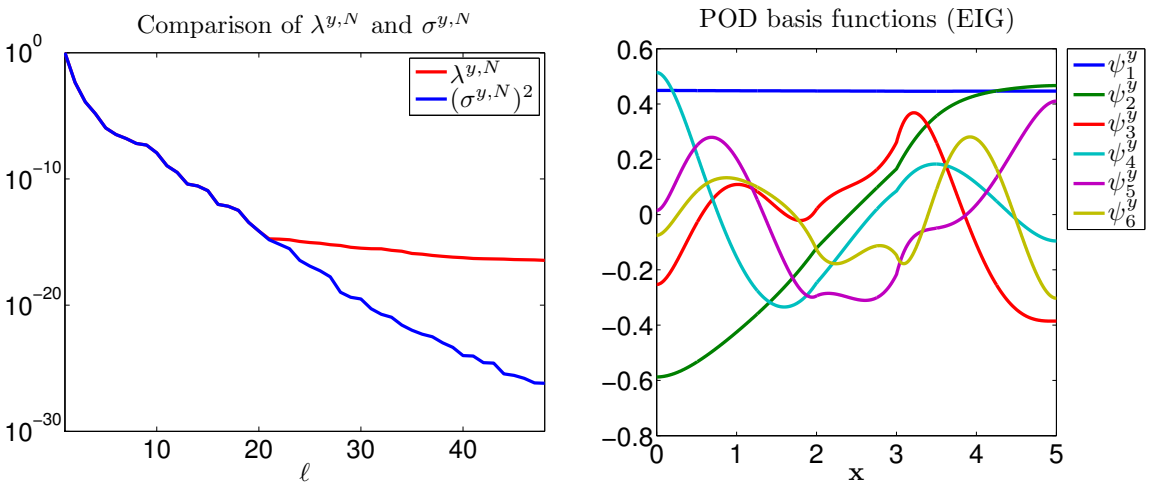


Figure 2.5: Decay of the normalized eigenvalues for the variable $y$ (left) and the corresponding first six POD basis functions obtained by the eigenvalue decomposition (right).

Algorithm 3 always return an orthonormal basis and no further orthogonalization has to be applied since the basis is computed of the projection residuals which are orthogonal to the previously obtained basis [44]. The orthonormality of the basis is even preserved in the case that a snapshot set is selected more than once. Note that the computation of the projection error in Algorithm 3 can be implemented very efficiently since each $\mathrm{E}_j$ only depends on $\mathrm{Y}_j$ and hence the evaluation can be done in parallel. Moreover, the dimension of the POD basis is determined by the algorithm and does not have to be set priorly.

**Remark 2.17.** *As stated in Remark 2.15 also in the case of the POD-Greedy approach the application of an orthogonalization process will improve the numerical results. This is a purely numerical issue and applying the orthogonalization improves the results obtained by the eigenvalue solver of the POD subproblems.*

Let us shortly look at the obtained POD basis from the two described methods. We start by introducing the settings. Let us consider the parameter domain

$$(\mu_1, \mu_2, \mu_3, \mu_4) \in \{1.25\} \times \{-0.75, -0.25\} \times \{-0.75, -0.25\} \times \{0.5, 1\}.$$

For these eight parameters we compute the finite element solution with the settings described in Section 1.3.3. We will focus only on the variable $y$ since similar results can be expected for the variables $p$ and $q$. To measure the performance of the two approaches we define the following error measures:

$$err^y_{\max}(\ell) = \max_{j=1,\dots,N_s} (err^y_{\mathcal{P}}(\ell, j)) \quad \text{and} \quad err^y_{\sum}(\ell) = \sqrt{\sum_{j=1}^{N_s} err^y_{\mathcal{P}}(\ell, j)^2}.$$

First we have a look at the standard POD approach using the snapshot matrix $\mathbb{Y}$. As in the previous section we utilize the eigenvalue and the singular value decomposition. Figure 2.5

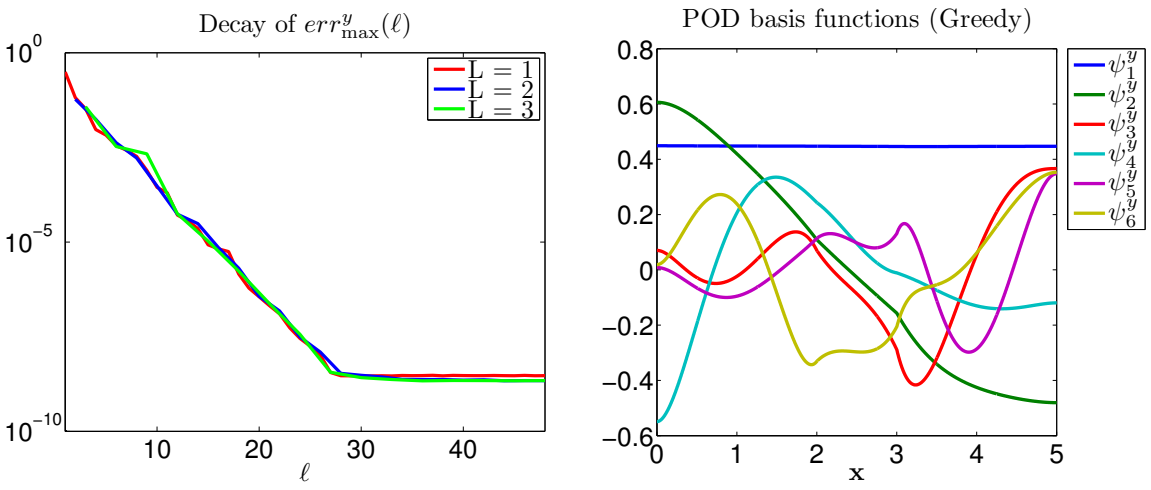


Figure 2.6: Decay of the projection error in $y$ for different values of $L$ (left) and the first six POD basis functions obtained by the POD-Greedy algorithm using $L = 1$ (right).

shows the decay of the first $48$ eigenvalues and singular values for the variable $y$ (left) together with the first six POD basis functions (right). For the POD-Greedy approach we additionally investigate the number of added POD bases in each iteration. For this we choose $L \in \{1, 2, 3\}$. In Figure 2.6 (left) the decay of the maximum projection error $(err^y_{\max}(\ell))$ for the variable $y$ is shown for different choices of $L$. It can be seen that the projection error decays equally fast for all the three choices. When computing the basis a larger $L$ can achieve a faster basis computation but can also lead to a larger basis. The first six POD bases obtained for $L = 1$ are shown in Figure 2.6 (right). Both methods deliver very similar shaped POD bases. Note that by using a different $L$ in Algorithm 3 the basis might be obtained in a different order.

Lastly, we want to compare the performance directly. For this we look at the errors $err^y_{\max}(\ell)$ for both approaches. Additionally, we look at the error $err^y_{\sum}(\ell)$ and compare it with the theoretical approximation error $err^y_{\lambda}(\ell)$. For the case using the POD-Greedy algorithm we fix $L = 1$ in this comparison. The results are outlined in Figure 2.7. Both methods deliver similar accuracy although when comparing $err^y_{\sum}(\ell)$ it can be seen that the direct POD approach can achieve the accuracy given by $err^y_{\lambda}(\ell)$ while for the POD-Greedy the error is always slightly larger. This is due to the fact that the POD basis is optimal and is designed to minimize the error given by $err^y_{\sum}(\ell)$. The POD-Greedy approach is not too far off. Due to the capability of computing the basis in parallel the computational time can be reduced, especially if $L$ is chosen larger than $1$. Note that both methods reach the accuracy of the finite element discretization.

### 2.4.3 POD for adaptive meshes

In this section we want to focus on the computation of a POD basis for a solution given on an adaptive spatial mesh. We assume that the snapshots are given as $\mathrm{y}^{N,1} \in \mathbb{R}^{dof_1}, \ldots,$ $\mathrm{y}^{N,N_t} \in \mathbb{R}^{dof_{N_t}}$. In this case we cannot set up the matrix Y directly. Under the assumption

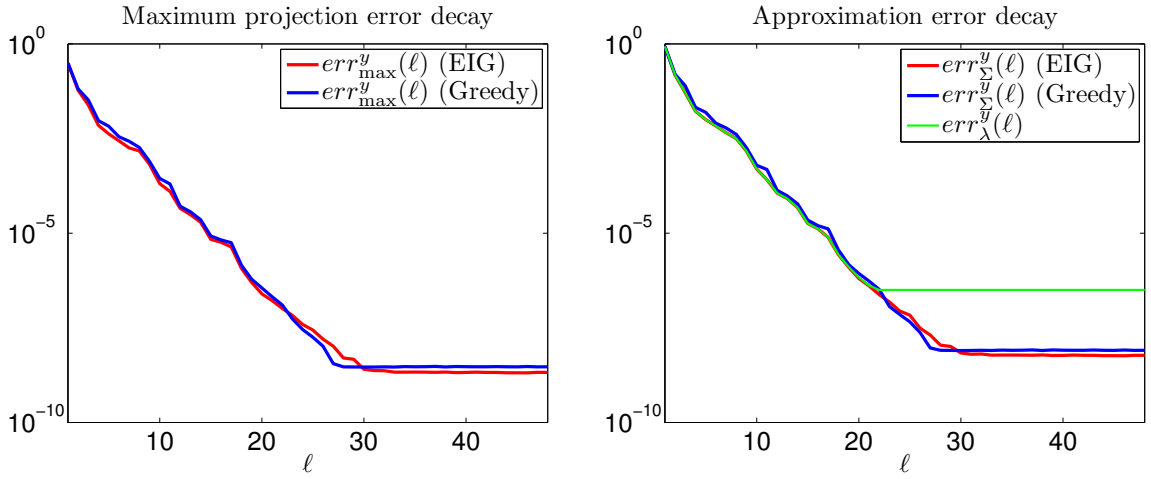


Figure 2.7: Comparison of the error decay for the POD approach and the POD-Greedy approach measured in $err^y_{\max}(\ell)$ (left) and $err^y_{\sum}(\ell)$ (right).

that $\mathrm{y}^{N,k}$ are finite element coefficients it would be possible to use the Galerkin representation to evaluate the variable $y$ on an unified grid, then compute the weight matrix $\mathrm{W}$ and continue as described in Section 2.2.2. The approach we want to outline here does not depend whether the solution is a finite element solution or for example a finite difference or finite volume solution. The idea is to develop an approach that is independent of the original discretization method and only requires solution vectors $\mathrm{y}^{N,k}$ and the corresponding grids as data.

We outline the computation of the basis functions for $y$. For this we want to apply the POD method introduced in the previous sections. Since we do not know the underlying discretization approach we use a numerical interpolation method to generate a continuous representation of the solution. This makes the method independent of the original discretization approach since they will all be handled in the same way. We here use the notation $y^{\mathrm{pol}}(t,x)$ for the (piecewise) polynomial representation of the snapshots with the interpolation property

$$y^{\mathrm{pol}}(t_k, \mathbf{x}_i) = \mathrm{y}_i^{N,k} \quad \text{for} \quad i = 1, \ldots, dof_k \quad \text{and} \quad k = 0, \ldots, N_t.$$

Here $t_k$ represents the discretization points in time and $\mathbf{x}_i$ the discretization points in space. Hence, it is guaranteed that the information of the data can be extracted exactly from the polynomials. We now compute the POD basis by reformulating (2.1) in the semi-discrete form:

$$\begin{cases} \min\limits_{\{\psi_i^y\}_{i=1}^{\ell_y}} \sum\limits_{k=0}^{N_t} \alpha_k \left\| y^{\mathrm{pol}}(t_k, \cdot) - \sum\limits_{i=1}^{\ell_y} \langle y^{\mathrm{pol}}(t_k, \cdot), \psi_i^{y,\mathrm{pol}}(\cdot) \rangle_{\mathcal{H}} \, \psi_i^{y,\mathrm{pol}}(\cdot) \right\|_{\mathcal{H}}^2 \\ \text{subject to} \quad \langle \psi_i^{y,\mathrm{pol}}(\cdot), \psi_j^{y,\mathrm{pol}}(\cdot) \rangle_{\mathcal{H}} = \delta_{ij} \quad \text{for} \quad 1 \le i, j \le \ell, \end{cases} \tag{2.22}$$

with $\alpha_k$ nonnegative trapezoidal weights as introduced in Section 2.3.3. Note that here the time is discretized but the spatial variable is infinite dimensional. Hence, we indicate the

norms and inner products by $\mathcal{H}$. Assuming $(N_t+1) \ll dof$ we determine the POD basis, as presented in Section 2.2.2 by using the $(N_t+1)\times(N_t+1)$ matrix

$$\mathcal{K}_{ij} = \sqrt{\alpha_i}\sqrt{\alpha_j}\langle y^{\mathrm{pol}}(t_j,\cdot), y^{\mathrm{pol}}(t_i,\cdot)\rangle_{\mathcal{H}}.$$

We assume the $\ell_y$ largest eigenvalues of $\mathcal{K}$ are given in the form $\lambda_1^{y,\mathrm{pol}} \geq \ldots \geq \lambda_{\ell_y}^{y,\mathrm{pol}} > 0$ and hence the POD basis is given by

$$\psi_i^{y,\mathrm{pol}}(\mathbf{x}) = \frac{1}{\sqrt{\lambda_i^{y,\mathrm{pol}}}} \sum_{k=0}^{N_t} \sqrt{\alpha_k}(v_i^y)_k y^{\mathrm{pol}}(t_k,\mathbf{x}), \tag{2.23}$$

where $v_i^y \in \mathbb{R}^{N_t+1}$ are the eigenvectors of $\mathcal{K}$ to the corresponding eigenvalues $\lambda_i^y$. Note that the POD basis is given as a linear combination of polynomials and hence are also polynomials. The approximation error for this approach is given by

$$\sum_{k=0}^{N_t} \alpha_k \left\| y^{\mathrm{pol}}(t_k,\cdot) - \sum_{i=1}^{\ell_y} \langle y^{\mathrm{pol}}(\mathrm{t}_k,\cdot), \psi_i^{y,\mathrm{pol}}(\cdot)\rangle_{\mathcal{H}} \psi_i^{y,\mathrm{pol}}(\cdot) \right\|_{\mathcal{H}}^2 = \sum_{i=\ell_y+1}^{N_t} \lambda_i^{y,\mathrm{pol}}. \tag{2.24}$$

Alternatively, if $dof \ll N_t+1$ the approach using the operator $\mathcal{R}$ can be used as previously introduced.

Let us next give some details on the numerical realization. We will use cubic splines to eliminate the spatial discretization of the snapshots. There are different variants of interpolation methods to avoid oscillation. In the numerical experiments we utilize the implementation provided by the GNU SCIENTIFIC LIBRARY (GSL) using the `akima` routines. Alternatively the MATLAB routine `interp1` with the option `pchip` can be used. The implementation is a realization of the piecewise cubic Hermite interpolation while the GSL implementation realizes a non-rounded corner algorithm for the cubic spline interpolation. The advantage of the GSL variant is that it can also provide derivatives of the polynomials. The representation of the data by polynomials allows the construction of the semi-discrete POD method since the variables are infinite dimensional in space. This is the first step to make the POD computation from data and independent from the original spatial discretization method.

To obtain a fully discrete POD method we have to realize the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. This can be done by applying numerical integration and one obtains for $\mathcal{H} = H$

$$\langle y^{\mathrm{pol}}(t_j,\cdot), y^{\mathrm{pol}}(t_j,\cdot)\rangle_H = y^{\mathrm{pol}}(t_j,\mathbf{x}^{\mathrm{int}})^\top \mathrm{W} y^{\mathrm{pol}}(t_j,\mathbf{x}^{\mathrm{int}}) = \sum_{k=1}^{N_{\mathrm{int}}} w_k y^{\mathrm{pol}}(t_j,\mathbf{x}_k^{\mathrm{int}})^2,$$

where $\mathbf{x}^{\mathrm{int}}$ are appropriately chosen points for the numerical integration of the polynomials and

$$\mathrm{W} = \mathrm{diag}(w_1,\ldots,w_{N_{\mathrm{int}}})$$

the weight matrix containing the $N_{\mathrm{int}}$ associated integration weights $w_i$ in the diagonal. For the integration we use the Gauss-Legendre quadrature. Note that we obtain an equality

since the numerical integration of polynomials can be carried out exactly. In our case we use piecewise cubic polynomials and hence a Gaussian quadrature rule of order four is required. Note that this approach is not limited to $\mathcal{H} = H$ but also $\mathcal{H} = V$ is possible. In this case additionally the derivatives of the polynomials are included in the inner products.

To compute the POD basis the results from Section 2.2.2 can be directly applied. Again for the data it is assumed that $N_t + 1 \ll dof$. Then the $\bar{\mathcal{K}}$ matrix is given as

$$\bar{\mathcal{K}} = \mathrm{D}^{\frac{1}{2}} \mathrm{Y}^\top \mathrm{W} \mathrm{Y} \mathrm{D}^{\frac{1}{2}} \in \mathbb{R}^{(N_t+1)\times(N_t+1)},$$

where $\mathrm{Y} = [y^{\mathrm{pol}}(t_0, \mathbf{x}^{\mathrm{int}}), \ldots, y^{\mathrm{pol}}(t_{N_t}, \mathbf{x}^{\mathrm{int}})]$ and $\mathrm{D} = \mathrm{diag}(\alpha_0, \ldots, \alpha_{N_t})$ a diagonal matrix containing the weights for the trapezoidal integration over time. The POD basis is then obtained by

$$\psi_i^{y,\mathrm{pol}}(\mathbf{x}^{\mathrm{int}}) = \frac{1}{\sqrt{\lambda_i^{y,\mathrm{pol}}}} \mathrm{Y} \mathrm{D}^{\frac{1}{2}} v_i \quad \text{for } i = 1, \ldots, \ell_y,$$

where $v_i$ is the $i$-th eigenvector of $\bar{\mathcal{K}}$ corresponding to the $i$-th largest eigenvalue $\lambda_i$. In the case of $dof \ll N_t + 1$ a similar approach using the operator $\bar{\mathcal{R}}$ is used. We then get

$$\bar{\mathcal{R}} = \mathrm{W}^{\frac{1}{2}} \mathrm{Y} \mathrm{D} \mathrm{Y}^\top \mathrm{W}^{\frac{1}{2}} \in \mathbb{R}^{dof \times dof}.$$

Then the POD basis is obtained by

$$\psi_i^{y,\mathrm{pol}}(\mathbf{x}^{\mathrm{int}}) = \mathrm{W}^{-\frac{1}{2}} u_i \quad \text{for } i = 1, \ldots, \ell_y,$$

where $u_i$ is the $i$-th eigenvector of $\bar{\mathcal{R}}$ corresponding to the $i$-th largest eigenvalue $\lambda_i^{y,\mathrm{pol}}$. Note that the computation of $\mathrm{W}^{\frac{1}{2}}$ and $\mathrm{W}^{-\frac{1}{2}}$ is very cheap since these matrices are diagonal matrices. This is in contrast to the approach presented in Section 2.2.2. Moreover, both approaches are linked by the singular value decomposition of $\bar{\mathrm{Y}} = \mathrm{W}^{\frac{1}{2}} \mathrm{Y} \mathrm{D}^{\frac{1}{2}}$ with the property $\bar{\mathrm{Y}} v_i = \sigma_i^{y,\mathrm{pol}} u_i$ and $\sigma_i^{y,\mathrm{pol}} = \sqrt{\lambda_i^{y,\mathrm{pol}}}$. A comparison has already been presented in Section 2.4.1.

The POD basis $\{\psi_i^{y,\mathrm{pol}}\}_{i=1}^{\ell_y}$ obtained by the above introduced methods are evaluation of polynomials at the integration points. Using equation (2.23) the polynomial representation can also be obtained. Due to the polynomial origin of the basis functions the derivatives can be computed easily. Using this we can now generate the discrete reduced order model simply by evaluating the integrals and obtaining the matrices (2.6). These integrals can again be computed by introducing integration points and corresponding weights. This construction makes the reduced order model independent of the original spatial discretization method. The reduced order model is computed entirely from the data provided through the snapshots. This is in contrast to other approach presented in Section 2.2.1, where the availability of the system matrices from the finite element discretization are required to perform a projection, compare (2.11) or [13, 62, 84], where the standard POD method is applied to similar systems.

Alternatively, if noisy data is provided, e.g. measurements this method can be applied as a smoothing step. Then the (piecewise) polynomial interpolation is replaced by a (piecewise) polynomial approximation. The rest of the steps are the same. Hence, a model can be extracted directly from the noisy data without setting up a high dimensional finite element model first.

### 2.4.4 Empirical interpolation methods

The reduced order model introduced in (2.7) is a nonlinear system. Hence, the problem with the POD Galerkin approach is the complexity of the evaluation of the nonlinearities. In the following we will outline the procedure for the nonlinearity $\mathcal{N}$. All the described steps are analogous for the nonlinear diffusion coefficient $c_2$. We illustrate the evaluation expenses for the nonlinearity $\mathcal{N}$ in (2.7a). Thus, we can write

$$\mathcal{N}^{\ell_y,N}(\mathrm{y}^\ell(t),\mathrm{p}^\ell(t),\mathrm{q}^\ell(t);\mu) = \big(\Psi^{y,N}\big)^\top \mathcal{N}^N\big(\Psi^{y,N}\mathrm{y}^\ell(t),\Psi^{p,N}\mathrm{p}^\ell(t),\Psi^{q,N}\mathrm{q}^\ell(t);\mu\big).$$

This can be interpreted in the way that the variables $\mathrm{y}^\ell, \mathrm{p}^\ell$ and $\mathrm{q}^\ell$ are first expanded to a vector of dimension of the finite element discretization, i.e. of dimension $dof$. Then the nonlinearity $\mathcal{N}(\Psi^{y,N}\mathrm{y}^\ell(t),\Psi^{p,N}\mathrm{p}^\ell(t),\Psi^{q,N}\mathrm{q}^\ell(t);\mu)$ is evaluated and integrated with the finite element ansatz functions $\varphi_i$ to get $\mathcal{N}^N$. At last the result is reduced back to the low dimension $\ell_y$ of the reduced order model by projection using $\Psi^{y,N}$. This is computationally expensive. Further, this means that our reduced order model is not independent of the full dimension $dof$ and additionally the finite element assembler is needed. Note that when applying a Newton method to the system (2.7) the Jacobian of the nonlinearity is also needed. For instance, we have

$$\begin{aligned}\mathcal{N}^{\ell_y,N}_{\mathrm{y}^\ell}(\mathrm{y}^\ell(t),\mathrm{p}^\ell(t),\mathrm{q}^\ell(t);\mu) &= \frac{\partial \mathcal{N}^{\ell_y,N}}{\partial \mathrm{y}^\ell}(\mathrm{y}^\ell(t),\mathrm{p}^\ell(t),\mathrm{q}^\ell(t);\mu)\\ &= \big(\Psi^{y,N}\big)^\top \mathcal{N}^N_{(\Psi^{y,N}\mathrm{y}^\ell)}\big(\Psi^{y,N}\mathrm{y}^\ell(t),\Psi^{p,N}\mathrm{p}^\ell(t),\Psi^{q,N}\mathrm{q}^\ell(t);\mu\big)\Psi^{y,N},\end{aligned}$$

for $t \in [0,T]$, where

$$\begin{aligned}&\mathcal{N}^N_{(\Psi^{y,N}\mathrm{y}^\ell)}\big(\Psi^{y,N}\mathrm{y}^\ell(t),\Psi^{p,N}\mathrm{p}^\ell(t),\Psi^{q,N}\mathrm{q}^\ell(t);\mu\big)\\ &\quad= \left(\int_\Omega \frac{\partial \mathcal{N}}{\partial y}\big(\mathbf{x},y^{\ell,N}(t,\mathbf{x}),p^{\ell,N}(t,\mathbf{x}),q^{\ell,N}(t,\mathbf{x});\mu\big)\varphi_j(\mathbf{x})\varphi_i(\mathbf{x})\,\mathrm{d}\mathbf{x}\right)_{1\le i,j\le dof}.\end{aligned}$$

Analogously, the other nonlinear parts required in the Jacobian matrices are generated. Again the same problem can be observed. Note that here the computational expenses are even larger since the Jacobian submatrix $\mathcal{N}^N_{(\Psi^{y,N}\mathrm{y}^\ell)}$ is of dimension $dof \times dof$. Hence, a finite element matrix has to be assembled and then reduced to the dimension of the reduced order model by pre and post multiplying with $\Psi^{y,N}$. Note that this expensive evaluations have to be carried out throughout the simulation process and hence also the computational complexity of the reduced order model will be high. To avoid this computational expensive evaluation the empirical interpolation method (EIM) was introduced [4]. This method is often used in combination with the reduced basis approach [39, 65]. The second approach we will have a look is the discrete empirical interpolation method (DEIM) as introduced in [16, 17]. While the EIM implementation is based on a greedy algorithm (same strategy as introduced in Section 2.4.2) the DEIM implementation is based on a POD approach combined with a greedy algorithm. We will now describe both methods in the discrete setting. We define

$$n(t;\mu) = \mathcal{N}(\mathbf{x},\Psi^{y,N}\mathrm{y}^\ell(t),\Psi^{p,N}\mathrm{p}^\ell(t),\Psi^{q,N}\mathrm{q}^\ell(t);\mu) \in \mathbb{R}^{dof}.$$

The goal is to approximate $n(t;\mu)$ by a Galerkin ansatz utilizing $\ell_{\mathcal{N}}^{EIM}$ linearly independent functions $\phi^1,\dots,\phi^{\ell_{\mathcal{N}}^{EIM}} \in \mathbb{R}^{dof}$, i.e.

$$n(t;\mu) \approx \sum_{k=1}^{\ell_{\mathcal{N}}^{EIM}} \phi^k c_k(t;\mu) = \Phi c(t;\mu) \tag{2.25}$$

with $c(t;\mu) = (c_1(t;\mu),\dots,c_{\ell_{\mathcal{N}}^{EIM}}(t;\mu))^\top \in \mathbb{R}^{\ell_{\mathcal{N}}^{EIM}}$ and the matrix $\Phi = \left(\phi^1 \,\middle|\, \dots \,\middle|\, \phi^{\ell_{\mathcal{N}}^{EIM}}\right) \in \mathbb{R}^{dof\times \ell_{\mathcal{N}}^{EIM}}$. Note that by the subindex $\mathcal{N}$ we indicate the association to the nonlinearity $\mathcal{N}$. We can now write the approximation of $\mathcal{N}^{\ell_y,N}(\mathrm{y}^\ell(t),\mathrm{p}^\ell(t),\mathrm{q}^\ell(t);\mu)$ as

$$\mathcal{N}^{\ell_y,N}(\mathrm{y}^\ell(t),\mathrm{p}^\ell(t),\mathrm{q}^\ell(t);\mu) \approx \sum_{k=1}^{\ell_{\mathcal{N}}^{EIM}} \left(\Psi^y\right)^\top \theta^k c_k(t;\mu)$$

with

$$\theta^k = \left(\int_\Omega \phi^{k,N}(\mathbf{x})\varphi_i(\mathbf{x})\,\mathrm{d}\mathbf{x}\right)_{1\le i\le dof} \quad \text{and} \quad \phi^{k,N}(\mathbf{x}) = \sum_{i=1}^{dof} \phi_i^k \varphi_i(\mathbf{x}).$$

Here, $\ell_{\mathcal{N}}^{EIM}$ denotes the number of basis functions chosen for the interpolation as well as the number of interpolation points that will be associated to the basis functions. Note that the approximation of $\mathcal{N}^{\ell_y,N}(\mathrm{y}^\ell(t),\mathrm{p}^\ell(t),\mathrm{q}^\ell(t);\mu)$ involves the approximation (2.25) and the integration of basis vectors $\phi^k$. In order to perform the integration of $\phi^k$ the entries of the vector are interpreted as finite element coefficients. This introduced an approximation error that can be neglected if the discretization is fine enough.

The question arising is how to compute the matrix $\Phi$ and the vector $c(t;\mu)$. Let $\wp^{EIM}$ be an index vector and $\mathrm{A} \in \mathbb{R}^{dof\times \ell_{\mathcal{N}}^{EIM}}$ a given matrix. Then by $\mathrm{A}_{\{\wp^{EIM}\}}$ we denote the submatrix consisting of the rows of A corresponding to the indices in $\wp^{EIM}$. Obviously, if we choose $\ell_{\mathcal{N}}^{EIM}$ indices then the overdetermined system $n(t;\mu) = \Phi c(t;\mu)$ can be solved by choosing $\ell_{\mathcal{N}}^{EIM}$ rows of $n(t;\mu)$ and $\Phi$. Here it is assumed that $\Phi_{\{\wp^{EIM}\}}$ is invertible.

Assuming we have computed $\Phi$ and $\wp^{EIM}$ by an algorithm. Then we proceed as follows. For simplicity we introduce here the matrix $\mathrm{P} = (\mathrm{e}_{\wp_1^{EIM}} \,|\, \dots \,|\, \mathrm{e}_{\wp_{\ell^{EIM}}^{EIM}})$, where $\mathrm{e}_{\wp_i^{EIM}} = (0,\dots,0,1,0,\dots,0)^\top \in \mathbb{R}^{dof}$ is a vector with all zeros and at the $\wp_i^{EIM}$-th row a one. Note that $\Phi_{\{\wp^{EIM}\}} = \mathrm{P}^\top\Phi$ holds. To evaluate the approximate nonlinearity we need $c(t;\mu)$. Since we know $\Phi$ and the index vector $\wp^{EIM}$ we can compute

$$c(t;\mu) = \left(\mathrm{P}^\top\Phi\right)^{-1}\mathrm{P}^\top n(t;\mu) = \left(\mathrm{P}^\top\Phi\right)^{-1}\mathrm{P}^\top\mathcal{N}(\mathbf{x},\Psi^{y,N}\mathrm{y}^\ell(t),\Psi^{p,N}\mathrm{p}^\ell(t),\Psi^{q,N}\mathrm{q}^\ell(t);\mu),$$

for $t\in[0,T]$. The nonlinearity on which we focus in this work can be evaluated pointwise and they only depend on spatial variable $\mathbf{x}$, the variables $y$, $p$, $q$ and the parameter $\mu$. Hence, the matrix P can be moved into the nonlinearity and we obtain

$$\begin{aligned}
&\mathrm{P}^\top\mathcal{N}(\mathbf{x},\Psi^{y,N}\mathrm{y}^\ell(t),\Psi^{p,N}\mathrm{p}^\ell(t),\Psi^{q,N}\mathrm{q}^\ell(t);\mu)\\
&\quad= \left(\mathcal{N}(\mathbf{x},\Psi^{y,N}\mathrm{y}^\ell(t),\Psi^{p,N}\mathrm{p}^\ell(t),\Psi^{q,N}\mathrm{q}^\ell(t);\mu)\right)_{i\in\{\wp^{EIM}\}}\\
&\quad= \mathcal{N}(P^\top\mathbf{x},P^\top\Psi^{y,N}\mathrm{y}^\ell(t),P^\top\Psi^{p,N}\mathrm{p}^\ell(t),P^\top\Psi^{q,N}\mathrm{q}^\ell(t);\mu).
\end{aligned}$$

An extension for general nonlinearities is shown in [16]. Let us now have a look at the computational expenses. The matrices $\mathrm{P}^\top \Psi^{y,N} \in \mathbb{R}^{\ell_{\mathcal{N}}^{EIM} \times \ell_y}$, and $(\mathrm{P}^\top \Phi)^{-1} \in \mathbb{R}^{\ell_{\mathcal{N}}^{EIM} \times \ell_{\mathcal{N}}^{EIM}}$ can be precomputed. Further, we set $\Theta = (\theta^1 \,\big|\, \ldots \,\big|\, \theta^{\ell_{\mathcal{N}}^{EIM}}) \in \mathbb{R}^{\ell_y \times dof}$ and hence we can precompute $\left(\Psi^{y,N}\right)^\top \Theta \in \mathbb{R}^{\ell_y \times \ell_{\mathcal{N}}^{EIM}}$. All the precomputed quantities are independent of the finite element dimension $dof$. Additionally, during the iterations the nonlinearity only has to be evaluated at the interpolation points, i.e. only at $\ell_{\mathcal{N}}^{EIM}$ points. This allows the reduced order model to be completely independent of the finite element dimension. Note that the used method is an interpolation and therefore is exact at the interpolation points. For the Jacobian the approach is similar. To summarize the results we now state the computation procedure for the evaluation of the nonlinearity

$$\begin{aligned}&\mathcal{N}^{\ell_y,N}(\mathrm{y}^\ell(t), \mathrm{p}^\ell(t), \mathrm{q}^\ell(t); \mu)\\ &\quad \approx \left(\Psi^{y,N}\right)^\top \Theta \left(\mathrm{P}^\top \Phi\right)^{-1} \mathcal{N}(P^\top \mathbf{x}, P^\top \Psi^{y,N} \mathrm{y}^\ell(t), P^\top \Psi^{p,N} \mathrm{p}^\ell(t), P^\top \Psi^{q,N} \mathrm{q}^\ell(t); \mu)\end{aligned}$$

and the Jacobian

$$\mathcal{N}_{\mathrm{y}^\ell}^{\ell_y,N}(\mathrm{y}^\ell(t), \mathrm{p}^\ell(t), \mathrm{q}^\ell(t); \mu) \approx \sum_{k=1}^{\ell^{EIM}} \left(\Psi^{y,N}\right)^\top \Xi^k \Psi^{y,N} d_k$$

with

$$d = \left(\mathrm{P}^\top \Phi\right)^{-1} \frac{\partial \mathcal{N}}{\partial y}(P^\top \mathbf{x}, P^\top \Psi^{y,N} \mathrm{y}^\ell(t), P^\top \Psi^{p,N} \mathrm{p}^\ell(t), P^\top \Psi^{q,N} \mathrm{q}^\ell(t); \mu)$$

and

$$\Xi^k = \left( \int_\Omega \phi^{k,N}(x) \varphi_j(x) \varphi_i(x) \mathrm{d}x \right)_{1 \le i,j \le dof}$$

for the reduced order model. Note that the quantities $\left(\Psi^{y,N}\right)^\top \Xi^k \Psi^{y,N} \in \mathbb{R}^{\ell_{\mathcal{N}}^{EIM} \times \ell_{\mathcal{N}}^{EIM}}$ can be precomputed and the computation of $d \in \mathbb{R}^{\ell_{\mathcal{N}}^{EIM}}$ is independent of the finite element dimension $dof$. Hence, also the evaluation of the Jacobian can be realized very efficiently. For the other two equations in (2.7) the approach is the same. Note that $\Phi$ is independent of the three variables and only depends on the nonlinearity. Therefore, the computation has to be performed only once. The additional terms that are required are the reduction with respect to the other POD basis $\Psi^{p,N}$ and $\Psi^{q,N}$. For the nonlinear diffusion coefficient the approach is the same as for the submatrices for the Jacobian. Hence, another set of $\ell_{c_2}^{EIM}$ basis functions have to be computed.

Let us remark here that in the case of the EIM the strategy can be described in a continuous setting [4, 39]. We here show it in a discrete setting in order to show the similarities. In the numerical implementation both methods are utilized in a discrete setting hence the description in a discrete setting is closer to the implementation.

We now turn to the EIM and DEIM algorithms. When (1.1) is solved using the finite element method the nonlinearities $\mathcal{N}$ and $c_2$ are evaluated for each time step. If these evaluations are stored the procedure to determine the EIM basis $\Psi$ and the index vector $\wp^{EIM}$ does not involve any further evaluations of the nonlinearities. We denote by $\mathrm{E} \in \mathbb{R}^{dof \times (N_t+1)}$ the matrix with columns $\mathcal{N}(\mathbf{x}, y^{N,k}, p^{N,k}, q^{N,k}; \mu) \in \mathbb{R}^{dof}$ or $c_2(y^{N,k}; \mu) \in$

$\mathbb{R}^{dof}$ for $k = 0, \ldots, N_t$. The matrix $\mathrm{E}$ can be seen as a snapshot matrix. Moreover, by $\mathrm{E}_k$ the $k-$th column is denoted. Next let us have a look at the two algorithms of interest.

In the algorithms $\|\cdot\|_\infty$ stands for the maximum norm in $\mathbb{R}^{dof}$ and the operation 'arg max' returns the index, where the maximum entry occurs. In Algorithm 4 we state the EIM using a greedy algorithm. Here the basis $\phi^i$, $i = 1, \ldots, \ell^{EIM}$, is chosen from the provided snapshots of the nonlinearities by scaling and shifting. The obtained basis is not orthonormal. The advantage of this method is that the submatrix $\Phi_{\{\wp^{EIM}\}}$ is a lower triangular matrix. Hence, solving for $c(t;\mu)$ is computationally cheap. The drawback of this method is that the computation of the basis is more expensive than the DEIM algorithm presented in Algorithm 5. The DEIM algorithm on the other hand generates the basis using the POD approach. Here the previously introduced POD approach is applied to the snapshots of the nonlinearities to compute $\Phi$. The matrix $\Phi_{\{\wp^{EIM}\}}$ obtained by the DEIM method has no special structure. Hence, evaluating the nonlinearity using DEIM is more expensive compared to EIM. The computational cost can be reduced by precomputing a LU decomposition of $\Phi_{\{\wp^{EIM}\}}$. Then the evaluation of the nonlinearity using DEIM involves two solves compared to one solve for the EIM. Further, when comparing the two algorithms it can be seen that the computation for the EIM basis is more expensive compared to the DEIM basis. This can be seen when comparing line 13 in Algorithm 4 and line six in Algorithm 5. In each iteration of Algorithm 4 one has to solve $N_t + 1$ linear systems compared to one linear system in Algorithm 5. A detailed comparison with numerical results is presented in [62], where both methods are applied to a similar nonlinear system.

**Algorithm 4** (The empirical interpolation method (EIM))

**Require:** Tolerance $\varepsilon^{\mathrm{EIM}}$ and matrix $\mathrm{E} \in \mathbb{R}^{dof \times (N_t+1)}$ containing the snapshots

1: $k \leftarrow \arg\max_{j=1,\ldots,N_t+1} \|\mathrm{E}_j\|_\infty$
2: $\xi \leftarrow \mathrm{E}_k$
3: idx $\leftarrow \arg\max_{j=1,\ldots,dof} |\xi_j|$
4: $\phi_1 \leftarrow \xi/\xi_{\{\mathrm{idx}\}}$
5: $\Phi = [\phi^1]$ and $\wp^{EIM} = \mathrm{idx}$
6: $c_j \leftarrow \left(\Phi_{\{\wp^{EIM}\}}\right)^{-1}\left(\mathrm{E}_j\right)_{\{\wp^{EIM}\}}$ for $j = 1, \ldots, N_t + 1$
7: $k \leftarrow \arg\max_{j=1,\ldots,N_t+1} \|\mathrm{E}_j - \Phi c_j\|_\infty$
8: **while** $\|\mathrm{E}_k - \Phi c_k\|_\infty > \varepsilon^{\mathrm{EIM}}$ **do**
9: $\quad \xi \leftarrow \mathrm{E}_k$
10: $\quad$ idx$\leftarrow \arg\max_{j=1,\ldots,dof} |(\xi - \Phi c_k)_{\{j\}}|$
11: $\quad \phi^i \leftarrow (\xi - \Phi c_k)/(\xi - \Phi c_k)_{\{\mathrm{idx}\}}$
12: $\quad \Phi \leftarrow [\Phi, \phi^i]$ and $\wp^{EIM} \leftarrow [\wp^{EIM}, \mathrm{idx}\,]$
13: $\quad c_j \leftarrow \left(\Phi_{\{\wp^{EIM}\}}\right)^{-1}\left(\mathrm{E}_j\right)_{\{\wp^{EIM}\}}$ for $j = 1, \ldots, N_t + 1$
14: $\quad k \leftarrow \arg\max_{j=1,\ldots,N_t+1} \|\mathrm{E}_j - \Phi c_j\|_\infty$
15: **end while**
16: **return** $\Phi$ and $\wp^{EIM}$

In Figure 2.8 we compare the first six basis vectors obtained by each of the two methods. It can be seen that they are quite different which is due to their different properties. When

**Algorithm 5** (The discrete empirical interpolation method (DEIM))

**Require:** $\ell^{EIM}$ and matrix $\mathrm{E} \in \mathbb{R}^{dof \times (N_t+1)}$ containing the snapshots

1: Compute POD basis $\Phi = [\phi^1, \ldots, \phi^{\ell^{EIM}}]$ for $E$
2: idx$\leftarrow \arg\max_{j=1,\ldots,dof} |(\phi^1)_{\{j\}}|$
3: $U = [\phi^1]$ and $\wp^{EIM} =$ idx
4: **for** $i = 2$ to $\ell^{EIM}$ **do**
5: $\quad u \leftarrow \phi^i$
6: $\quad c \leftarrow \left(U_{\{\wp^{EIM}\}}\right)^{-1} u_{\{\wp^{EIM}\}}$
7: $\quad r \leftarrow u - Uc$
8: $\quad$ idx$\leftarrow \arg\max_{j=1,\ldots,dof} |(r)_{\{j\}}|$
9: $\quad U \leftarrow [U, u]$ and $\wp^{EIM} \leftarrow [\wp^{EIM}$, idx $]$
10: **end for**
11: **return** $\Phi$ and $\wp^{EIM}$

looking at the obtained basis more closely it can be observed that the basis obtained by DEIM has the maximum always at two or three, the two points where the nonlinearity $\mathcal{N}$ has a jump. At these two points the basis functions generated by the DEIM method will exhibit spikes of large magnitude. This is not the case for the basis obtained by EIM. This issue causes the DEIM method to fail when applied to the reduced order model. A similar phenomenon can also be observed when the POD method is applied to nonsmooth data. In [43] a similar observation is made when applying POD to linear parabolic PDEs with nonsmooth initial conditions. To overcome this issue the nonlinearity $\mathcal{N}$ can be divided into three parts and the DEIM method can be applied on the two nonzero parts. This leads to a more complex code which should be avoided. The EIM on the other hand does not suffer from this issue. The EIM uses the data directly to generate the basis functions, hence it can recover nonsmooth properties very well. Therefore, it is the method of choice in the numerical example presented in the next section. For the nonlinear term $c_2$ both methods perform equally well.

The DEIM approximation error can be incorporated into the a-priori error estimators. The tools needed for this are presented in [16–18]. For the nonlinear term $\mathcal{N}$ this is outlined in [62]. Additionally, there are recent results for a-posteriori error estimators [85]. Also for the EIM method a-priori error estimators are available [65]. In the context of the reduced basis method a-posteriori error estimators are developed [39]. Note that the error can be checked during the simulation process. In the algorithm a test can be included of the form

$$err_{L^\infty}^{\mathcal{N}} = \|\mathcal{N}(\mathbf{x}, \Psi^{y,N}\mathrm{y}^\ell(t), \Psi^{p,N}\mathrm{p}^\ell(t), \Psi^{q,N}\mathrm{q}^\ell(t); \mu) - \mathcal{N}^{EIM}(t; \mu)\|_{L^\infty(\Omega)},$$

where $\mathcal{N}^{EIM}(t; \mu)$ corresponds to the evaluation of the nonlinearity using one of the interpolation methods. This comes at the cost of an evaluation of the nonlinearity on the finite element discretization but no assembling is required. If the nonlinear term is cheap to evaluate this can be included in the code as a safeguard. This can be particularly useful in order to avoid divergence of the reduced order solver.

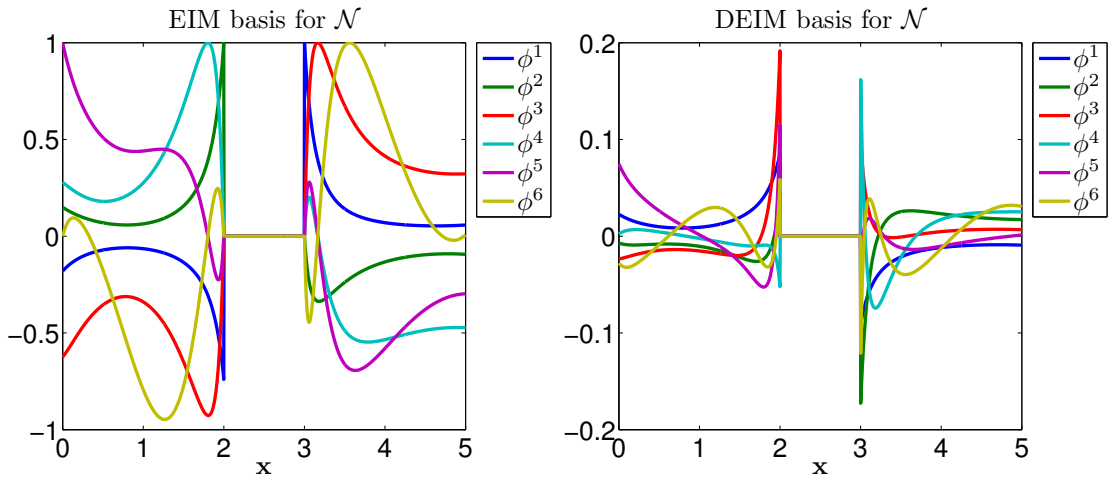


Figure 2.8: The first six basis vectors obtained by the empirical interpolation method (EIM) (left) and the discrete empirical interpolation method (DEIM) (right) for the nonlinearity $\mathcal{N}$.

## 2.5 Numerical experiments

In this section we present numerical results for the methods introduced in the previous sections. Three different experiments will be performed. In the first run we demonstrate the effectiveness of the POD Galerkin scheme to solve the nonlinear elliptic parabolic system. In the second run we consider the extension using multiple snapshot sets. Finally, numerical results utilizing the reduction strategy of Section 2.4.3 are presented.

### 2.5.1 Run 1 (Single snapshot set)

Let us start with setting up the problem. The settings for the nonlinear system (1.1) are given by (1.2) and (1.39) in Chapter 1. Note that in this run we look at the scenario involving $\bar{\mu}$ presented in Section 1.3.3. For the finite element solutions we choose the same settings as introduced in Section 1.3.3. The solver for the reduced order model is the same as in the finite element case, i.e. Algorithm 1 with the reduced order model instead of the finite element model. Hence, also the same settings are used as described for the finite element case. In the simulation we only consider the case $T = 4$ with $400$ time steps.

Next we discuss the settings for the simulation utilizing the POD method. We will report on the performance of the simulation using POD with and without the use of the empirical interpolation method (EIM). Let us start with giving the dimension of the basis for the POD. Recall the results from Section 2.4.1. Looking at Figure 2.1 we can see that the eigenvalues decrease rapidly. Here we choose $\ell_y = 18$, $\ell_p = 20$ and $\ell_q = 13$. This is a reasonable choice since it corresponds to approximately the value, where the kink can be observed. Further, we set $\varepsilon^{EIM} = 10^{-11}$. This results in $\ell_{\mathcal{N}}^{\text{EIM}} = 45$ for the nonlinearity $\mathcal{N}$ and $\ell_{c_2}^{\text{EIM}} = 38$ for the nonlinear diffusion coefficient $c_2$.

To measure the accuracy of the POD method with respect to the finite element method

the average relative error is introduced in the form

$$\varepsilon_{\mathcal{H}}^{\wp} = \sqrt{\frac{1}{N_t+1}\sum_{i=0}^{N_t}\frac{\|\wp^N(t_i)-\wp^{\ell,N}(t_i)\|_{\mathcal{H}}^2}{\|\wp^N(t_i)\|_{\mathcal{H}}^2}} \tag{2.26}$$

for $\wp \in \{y,p,q\}$, $N_t$ the number of time steps and $t_i$ the discretization points in time. For the norms we choose $\mathcal{H}=H$ or $\mathcal{H}=V$. Note that we can define the average relative error in this way since we choose an equidistant discretization in time. Additionally, we define

$$\varepsilon_{L^\infty}^{\wp} = \|\wp^N - \wp^{\ell,N}\|_{L^\infty(Q)}. \tag{2.27}$$

Lastly, we introduce the norm of the residual as an additional error indicator. The error is obtained by inserting the reduced order solution into the finite element discretization, i.e.

$$\begin{aligned}
\mathrm{res}_y(t) =& \mathrm{M}_1^N \Psi^{y,N} \mathrm{y}_t^{\ell,N}(t) + \mathrm{S}_{c_1}^N \Psi^{y,N} \mathrm{y}^{\ell,N}(t) \\
&+ \mathcal{N}^N(\Psi^{y,N}\mathrm{y}^{\ell,N}(t), \Psi^{p,N}\mathrm{p}^{\ell,N}(t), \Psi^{q,N}\mathrm{q}^{\ell,N}(t);\mu), \\
\mathrm{res}_p(t) =& \mathrm{S}_{c_2(\Psi^{y,N}\mathrm{y}^{\ell,N}(t);\mu)}^N \Psi^{p,N}\mathrm{p}^{\ell,N}(t) \\
&+ \mathcal{N}^N(\Psi^{y,N}\mathrm{y}^{\ell,N}(t), \Psi^{p,N}\mathrm{p}^{\ell,N}(t), \Psi^{q,N}\mathrm{q}^{\ell,N}(t);\mu), \\
\mathrm{res}_q(t) =& \mathrm{S}_{c_3}^N \Psi^{q,N}\mathrm{q}^{\ell,N}(t) \\
&- \mathcal{N}^N(\Psi^{y,N}\mathrm{y}^{\ell,N}(t), \Psi^{p,N}\mathrm{p}^{\ell,N}(t), \Psi^{q,N}\mathrm{q}^{\ell,N}(t);\mu) - \mathcal{I}^N(t).
\end{aligned}$$

Then computing the dual norm of the residuals we get

$$\varepsilon_{\mathrm{res}}^{\wp} = \sqrt{\frac{1}{N_t+1}\sum_{i=0}^{N_t}\|\mathrm{res}_{\wp}(t_i)\|_{V^*}^2} \tag{2.28}$$

for $\wp \in \{y,p,q\}$. This error indicates how well the reduced order solution solves the nonlinear system using the finite element discretization. In Tables 2.1 and 2.2 we look at the different error measures. We compare the performance of the reduced order model with and without the EIM. It can be observed that in both cases the results of the reduced order model can reconstruct the finite element solution very good. Note that the error indicator using the residual is not too far from the other errors. In the case of the variable $y$ the residual is slightly smaller than the error while in the case of the variables $p$ and $q$ the residual is greater than the actual error. Note that the errors are all in the magnitude of the finite element discretization which is approximately $10^{-6}$ in the $V$ norm.

| | $y$ | $p$ | $q$ |
|---|---|---|---|
| $\varepsilon_{L^2}^{\wp}$ | $6.4970 \times 10^{-8}$ | $5.1929 \times 10^{-8}$ | $3.8924 \times 10^{-8}$ |
| $\varepsilon_{H^1}^{\wp}$ | $6.8669 \times 10^{-7}$ | $4.5816 \times 10^{-7}$ | $2.4038 \times 10^{-7}$ |
| $\varepsilon_{L^\infty}^{\wp}$ | $3.7024 \times 10^{-7}$ | $2.8142 \times 10^{-7}$ | $2.7566 \times 10^{-7}$ |
| $\varepsilon_{\mathrm{res}}^{\wp}$ | $4.4354 \times 10^{-8}$ | $5.3310 \times 10^{-7}$ | $2.6668 \times 10^{-6}$ |

Table 2.1: Run 1: Comparison of the different errors when solving the reduced order model.

| | Solve FEM | POD Basis | EIM Basis | Solve ROM | Solve ROM-EIM |
|---|---|---|---|---|---|
| CPU time | 132.00 | $3 \times 0.15$ | $2 \times 1.30$ | 46.00 | 3.81 |

Table 2.3: Run 1: Summary of the performance for the finite element and reduced order model (ROM) with and without the empirical interpolation method (EIM) measured in seconds.

| | $y$ | $p$ | $q$ |
|---|---|---|---|
| $\varepsilon^{\wp}_{L^2}$ | $6.4976 \times 10^{-8}$ | $5.3562 \times 10^{-8}$ | $4.0466 \times 10^{-8}$ |
| $\varepsilon^{\wp}_{H^1}$ | $6.8673 \times 10^{-7}$ | $4.5880 \times 10^{-7}$ | $2.4054 \times 10^{-7}$ |
| $\varepsilon^{\wp}_{L^\infty}$ | $3.7021 \times 10^{-7}$ | $3.1106 \times 10^{-7}$ | $3.1592 \times 10^{-7}$ |
| $\varepsilon^{\wp}_{\text{res}}$ | $4.4358 \times 10^{-8}$ | $5.3564 \times 10^{-7}$ | $2.6669 \times 10^{-6}$ |

Table 2.2: Run 1: Comparison of the different errors when solving the reduced order model with the empirical interpolation method.

Further, in Figure 2.9 the absolute error between the solution obtained by the finite element method and the reduced order model using EIM is shown. It can be seen that the

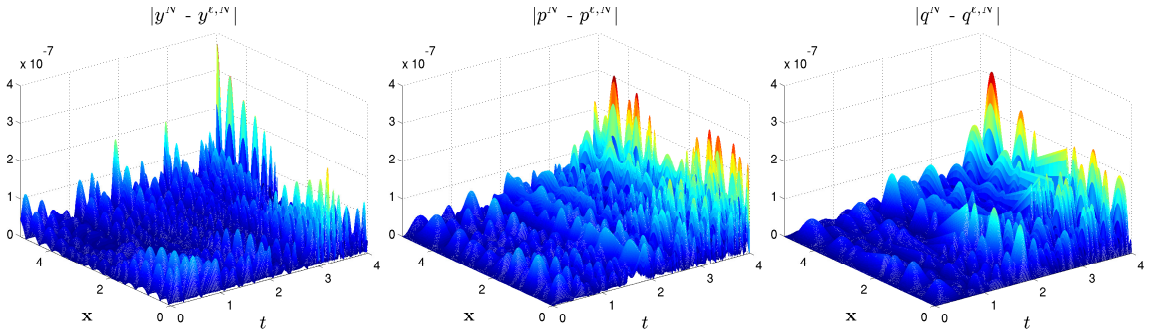


Figure 2.9: Run 1: Absolute error between the finite element solution and the POD solution using the empirical interpolation method for $y$ (left), $p$ (center) and $q$ (right).

POD method delivers nearly the same results as the finite element method and the error is distributed more or less equally over the whole space-time domain.

To compare the performance we state the CPU time in seconds. The computational performance of reduced order method compared to the finite element method is summarized in Table 2.3. It can be seen that the reduced order model can be solved very fast. This was expected when looking at the dimensions of the problem. Recall that the finite element discretization leads to a system with $3 \times 1999$ unknowns. In comparison the reduced order model is of dimension $18 + 20 + 13$. This is a significant reduction in dimension. Let us remark that the matrices in the reduced order model are full, hence speed up strategies as introduced for the sparse matrices arising from the finite element method cannot be ap-

plied. Note that the speedup when using EIM is significant. When no EIM is available the speedup is approximately of factor three while we obtain a speedup factor of approximately $35$ when the EIM is utilized. This indicates the computational complexity that arises from the nonlinear terms in the reduced order model as outlined in Section 2.4.4. For completeness, setting up the finite element system takes $0.029$ seconds. In the case of the reduced order model we get $0.01$ seconds when EIM is not used and $0.97$ seconds when EIM is applied. This time summarizes the generation of all matrices that do not change during the simulation process. In EIM case this includes the computation of the quantities $\Theta$ and $\Xi$ needed in the interpolation process.

Additionally, to the average relative error over the whole domain $\Omega$ we have a look at the average relative boundary error. This is in particular interesting in the application, where one can only measure certain quantities at the boundary. Therefore, it is important that the POD approximation is accurate on the boundary. Let us define the average relative error for the boundary as

$$\varepsilon_b^q = \frac{1}{N_t + 1} \sum_{i=0}^{N_t} \frac{|q^N(t_i, b) - q^{\ell,N}(t_i, b)|}{|q^N(t_i, b)|}. \tag{2.29}$$

This error is interesting for the variable $q(t, b)$ which is the measurable quantity in application. In the context of lithium ion batteries modeling this corresponds to the potential that can be measured. In Table 2.4 we summarize the results.

| | ROM | ROM-EIM |
|---|---|---|
| $\varepsilon_b^q$ | $5.2866e \times 10^{-10}$ | $1.4767 \times 10^{-8}$ |

Table 2.4: Run 1: Average relative errors on the boundary when using the reduced order model (ROM) with and without EIM.

It can be seen that the POD approach delivers good results also when looking only at one particular point of the domain $\Omega$, in our case the boundary. This is of great interest since when utilizing the reduced order model in an optimization only the boundary point will be of interest.

To summarize this numerical experiment let us recall that the reduced order model can recover the finite element solution very well. The EIM contributes significantly to the speedup of the reduced order model and the errors are small on the whole space time domain as well as on the boundary.

### 2.5.2 Run 2 (Multiple snapshot sets)

In this run we focus on the case that the parameter $\mu$ is changed. This is in contrast to Run $1$, where the parameter $\mu$ was fixed for all experiments. This scenario occurs when using the reduced order model in an optimization or parameter estimation process. For this experiment we introduce two sets, one sample set for which finite element solutions

are available or are generated. Additionally, we introduce a distinct test set to evaluate the reliability of the simulation using the reduced order model. As in Run 1 we use the settings for the nonlinear system, the finite element solver and the reduced order solver as described in Chapter 1.

Next let us introduce the settings for the reduced order model. We choose $\ell_y = 28$, $\ell_p = 28$ and $\ell_q = 20$. This setting corresponds to setting $\varepsilon^{\text{POD}} = 10^{-8}$ in the POD-Greedy approach. The EIM basis is computed using $\varepsilon^{\text{EIM}} = 10^{-11}$, which results in $\ell_{\mathcal{N}}^{EIM} = 42$ and $\ell_{c_2}^{EIM} = 45$. Next we introduce the sample and test set. We choose the two disjoint sets. The sample set is as defined in Section 2.4.2, i.e.

$$\mathcal{M}_{sample} = \{1.25\} \times \{-0.75, -0.25\} \times \{-0.75, -0.25\} \times \{0.5, 1\}.$$

The test set is given by

$$\begin{aligned}\mathcal{M}_{test} = \{1.00,&1.50\} \times \{-1.00, -0.50, -0.05\} \\ &\times \{-1.00, -0.50, -0.05\} \times \{0.25, 0.75, 1.25\}.\end{aligned}$$

The choice of $\mathcal{M}_{sample}$ gives us eight possible combinations for the sample set on which the PDE will be solved using the finite element discretization. These solutions are used as the snapshots to compute the POD basis utilizing the two strategies introduced in Section 2.4.2. Moreover, the EIM is computed using the data obtained by these finite element solves. The test set will give us $54$ possible combinations on which the reduced order model will be solved using the POD method together with EIM. We will not report on the reduced order model without EIM since there are too many possible combinations. From Run 1 we saw that using the EIM the solution of the reduced order model is accurate enough. Hence, there is no need to perform these experiments again. In the numerical results we will utilize the two strategies for the POD basis computation and compare their performance.

To illustrate the performance the average relative error (2.26) is compared. Additionally also the errors (2.27) and the residual (2.28) are used to measure the accuracy of the reduced order model with respect to the finite element model. In Figure 2.10 the results for the different error measures are summarized for the sample set $\mathcal{M}_{sample}$ for the two basis generation strategies. For the test set $\mathcal{M}_{test}$ the errors are compared in Figure 2.11. Note that all results are generated using EIM for the evaluation of the nonlinear terms. When looking at the plots it can be seen that both strategies perform equally good. Only slight differences can be spotted. As in Run 1 the residual for the variable $y$ is again smaller than the errors while for the variables $p$ and $q$ it is larger. To summarize, the reduced order model performs very well on the test set $\mathcal{M}_{test}$ considering that it is not a reconstruction since the two sets are disjoint.

Let us next have a look at the performance in terms of computational time. For this we have a look at the CPU time required to generate the basis for the reduced order model. This is outlined in Table 2.5. It can be seen that applying the POD method to all snapshots at once is faster than the POD-Greedy approach. This is due to the fact that the number of sample parameters is small. Additionally, the parallel implementation provided by MATLAB cannot be applied to this problem in a sophisticated way. Hence, not the full potential of the POD-Greedy method can be demonstrated. When adding more sample parameters

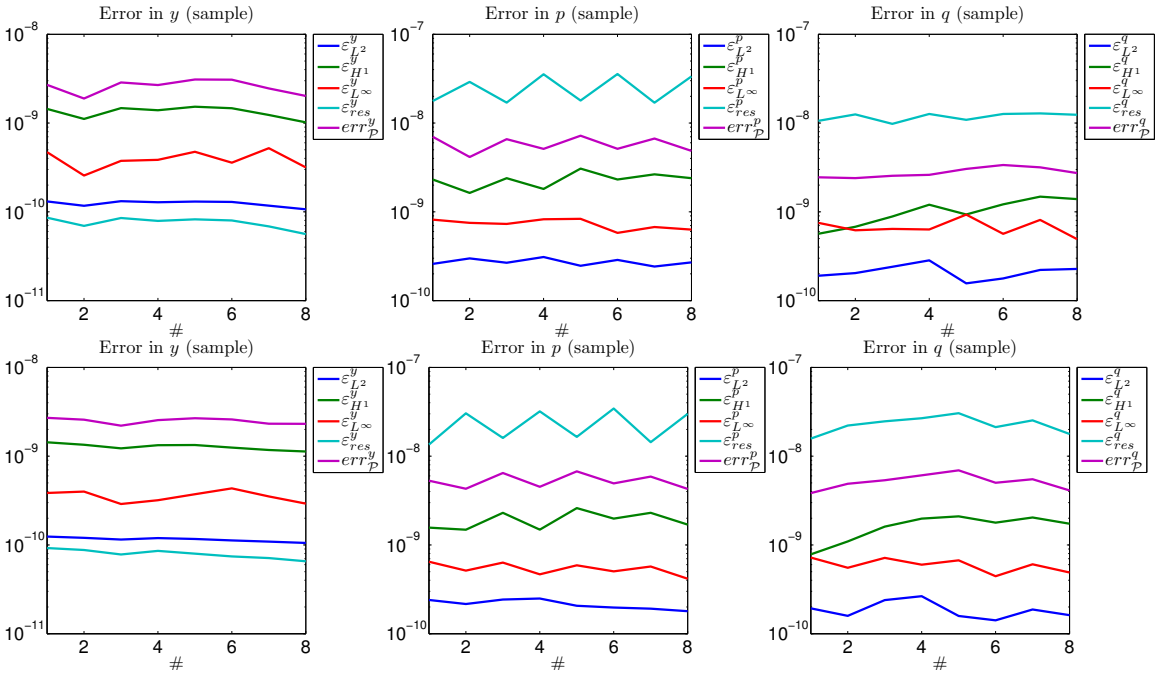


Figure 2.10: Run 2: Different error measures for the variables $y$ (left), $p$ (center) and $q$ (right) for the sample set $\mathcal{M}_{sample}$ utilizing a POD basis (top) and POD-Greedy basis (bottom).

the computation time using the POD method directly will approach the POD-Greedy strategy in means of computation time.

| | POD Basis (EIG) | POD-Greedy | EIM Basis |
|---|---|---|---|
| CPU time | $3 \times 6.93$ | $3 \times 40.27$ | $2 \times 11.81$ |

Table 2.5: Run 2: Summary of the computation time for the POD basis generation using POD (EIG) and POD-Greedy and the EIM basis generation measured in seconds.

In Table 2.6 we summarize the performance of the reduced order model with respect to the computational time. It can be seen that we again achieve a significant speedup. Here the speedup is even larger compared to the results obtained in Run 1. This is due to the fact that the finite element solver does require significantly more computation time for different parameter settings. On the other hand the reduced order solver requires approximately four seconds for each solve independent of the chosen parameter. Hence, we end up with speedup factors between 30 and 500 depending on the parameter. Note that again the EIM gives a huge speed up. In the case that EIM is not utilized the speedup factor is only around 2 to 40.

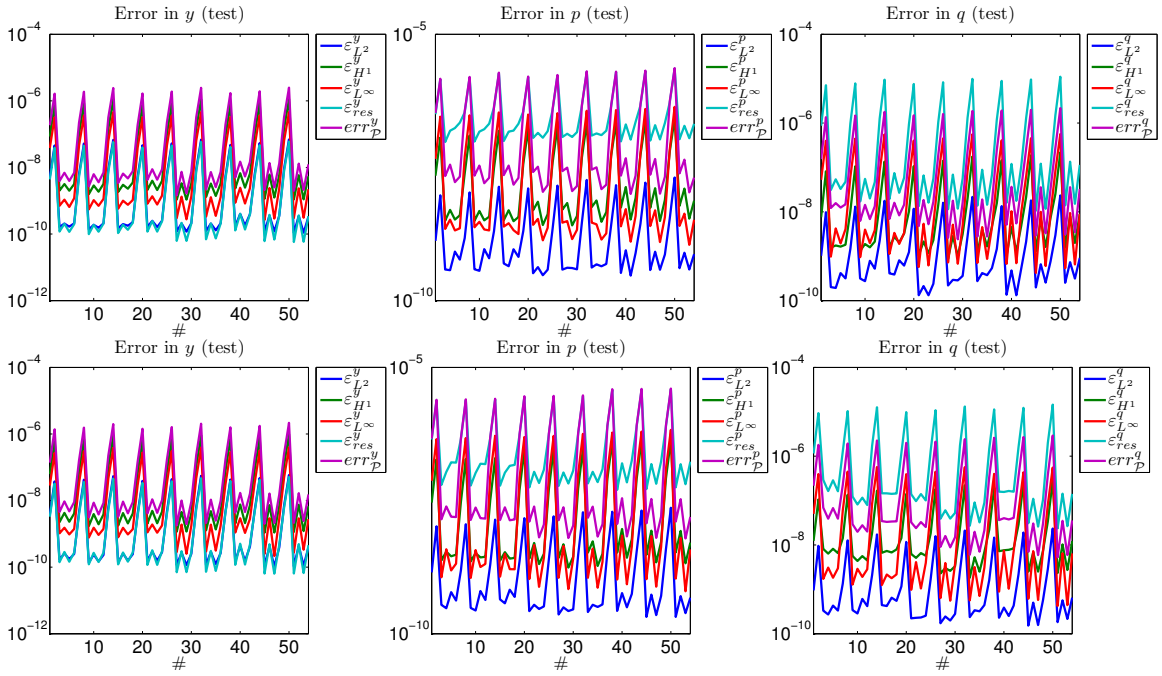


Figure 2.11: Run 2: Different error measures for the variables $y$ (left), $p$ (center) and $q$ (right) for the test set $\mathcal{M}_{sample}$ utilizing a POD basis (top) and POD-Greedy basis (bottom).

| | Solve FEM (sample) | Solve FEM (test) | Solve ROM | Solve ROM-EIM |
|---|---|---|---|---|
| CPU time | $\sim 120 - 160^{(*)}$ | $\sim 120 - 2000^{(**)}$ | $\sim 54.8^{(*,**)}$ | $\sim 3.79^{(*,**)}$ |

Table 2.6: Run 2: Summary of the performance for the finite element and reduced order model (ROM) with and without EIM measured in seconds ((*) for each sample parameter, (**) for each test parameter).

Lastly, we again have a look at the error on the boundary. For this we plot the relative error defined by (2.29). As described in Run 1 this error is interesting in application since it will correspond to the value that can be measured. In Figure 2.12 the error introduced by the reduced order model with and without EIM is compared.

As already observed for the error over the whole space-time domain the reduced order model performed very well. Even though the basis is only computed on the sample set the solutions obtained by the reduced order model on the test set are very good. Both methods for the POD basis generation deliver almost the same results.

In this experiment we showed that the different strategies for generating the POD basis deliver reduced order models that perform equally well. When the EIM is applied very good speedup factors can be achieved.

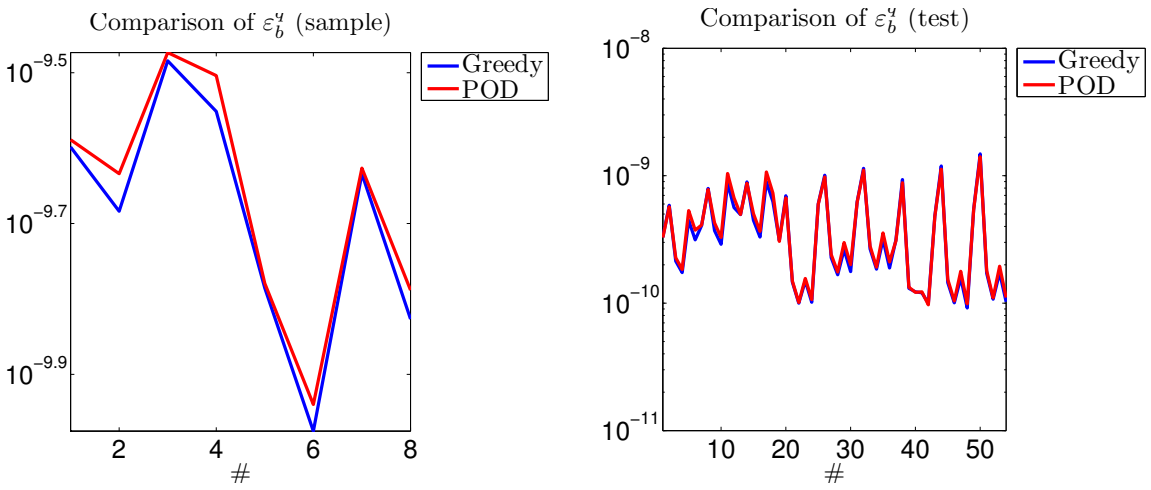


Figure 2.12: Run 2: Average relative boundary error for the sample (left) and the test (right) set using POD and POD-Greedy with EIM.

### 2.5.3 Run 3 (Adaptive meshes)

In this run we want to demonstrate the performance of the strategy introduced in Section 2.4.3. Since we do not have an adaptive finite element solver we generate the 'adaptive' solution in a post processing step. Grid points are removed by the following strategy. We compute the difference of the neighboring points. Then we compute the difference of the neighboring differences. If that absolute difference is less than $10^{-7}$ we remove the corresponding point. This can be realized by the MATLAB command

```
pt_rm = abs(diff(diff(FEM))) <= 1e-7,
```

where `FEM` is the finite element solution on the equidistant mesh. To avoid that the initial condition (which is constant) is removed we keep the first nine time steps completely. Additionally, we make sure that the boundary points are kept. In Figure 2.13 it is indicated which points are removed. Note that although it looks like whole regions are removed, this is only visually due to the high density of points. In the case of the variable $y$ we remove $144055$ points which corresponds to approximately $18\%$ of the total points. For the variables $p$ and $q$ we get $27729$ and $119139$ points to remove, which corresponds to $3.5\%$ and $14.8\%$, respectively.

By applying this strategy to the finite element solution obtained in Section 1.3.3 we obtain the adaptive solution. The settings for the finite element model and reduced order model are as described in Run 1, Section 2.5.1. The only difference is the implementation. We follow the strategy described in Section 2.4.3. Let us right away report on the numerical results.

First we state the errors (2.29) and the residual (2.28) in Tables 2.7 and 2.8. We again observe the same behavior as in the case where the reduced order model is generated by projection. The errors are in average approximately two magnitude more but still very good. Especially when comparing to the expected finite element error which is of magnitude $10^{-6}$ in the $V$ norm we see that the proposed strategy delivers very good results. The errors are

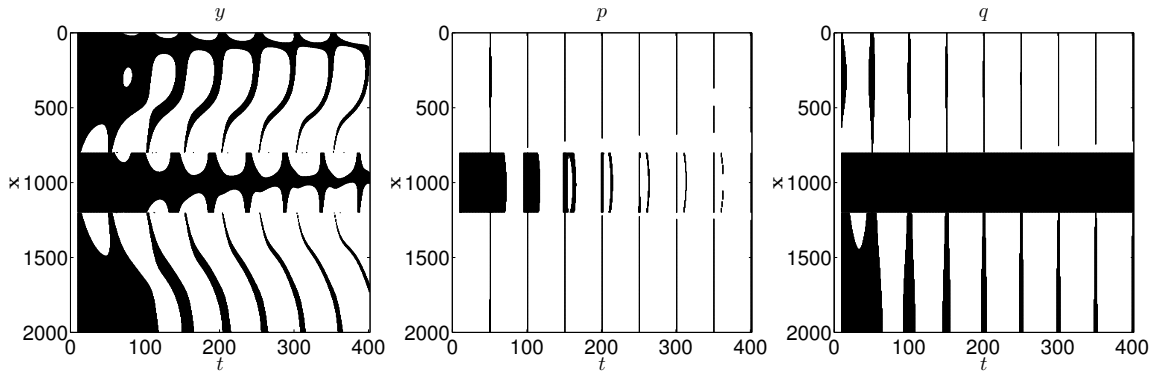


Figure 2.13: Run 3: Points removed from the equidistant discretization (in black) in order to obtain adaptive finite element solutions solving the nonlinear system (1.1).

| | $y$ | $p$ | $q$ |
|---|---|---|---|
| $\varepsilon^{\varphi}_{L^2}$ | $2.5751 \times 10^{-6}$ | $2.6030 \times 10^{-6}$ | $2.8246 \times 10^{-6}$ |
| $\varepsilon^{\varphi}_{H^1}$ | $2.6045 \times 10^{-5}$ | $4.8163 \times 10^{-6}$ | $3.1413 \times 10^{-6}$ |
| $\varepsilon^{\varphi}_{L^\infty}$ | $9.5834 \times 10^{-6}$ | $2.4332 \times 10^{-5}$ | $2.1790 \times 10^{-5}$ |
| $\varepsilon^{\varphi}_{\text{res}}$ | $1.8525 \times 10^{-6}$ | $1.4611 \times 10^{-5}$ | $1.7450 \times 10^{-5}$ |

Table 2.7: Run 3: Comparison of the different errors when solving the reduced order model with integration.

only introduced by the interpolation using the cubic splines and the POD approximation. All other integrals that are required to obtain the reduced order model can be evaluated exact by the Gauss-Legendre quadrature rule. The accuracy can be increased by adding more basis functions. In order to have a good comparison to the results of Run 1 we opted to use the same settings in both experiments. Note that also in this run the EIM works very well and delivers almost the same results as using the finite element method for the assembling of the nonlinear terms. Moreover, we again observe the same behavior of the residual as in the previous two numerical experiments.

| | $y$ | $p$ | $q$ |
|---|---|---|---|
| $\varepsilon^{\varphi}_{L^2}$ | $2.5739 \times 10^{-6}$ | $2.5985 \times 10^{-6}$ | $2.8252 \times 10^{-6}$ |
| $\varepsilon^{\varphi}_{H^1}$ | $2.6014 \times 10^{-5}$ | $4.9295 \times 10^{-6}$ | $3.1410 \times 10^{-6}$ |
| $\varepsilon^{\varphi}_{L^\infty}$ | $9.5813 \times 10^{-6}$ | $2.3712 \times 10^{-5}$ | $2.1787 \times 10^{-5}$ |
| $\varepsilon^{\varphi}_{\text{res}}$ | $1.8499 \times 10^{-6}$ | $1.5034 \times 10^{-5}$ | $1.7140 \times 10^{-5}$ |

Table 2.8: Run 3: Comparison of the different errors when solving the reduced order model with integration and EIM.

Let us now look at the computational time. How does the adaptivity of the finite element solution influence the performance of the reduced order model with respect to computational expenses? In Table 2.9 the CPU times are reported for the different computations needed. The computational expenses for the POD basis computation almost doubles when compared to the results in Table 2.3. In the EIM computation there is no difference. The computational complexity of the reduced order model using EIM is the same when compared to the results of Run 1. Lastly, we look at the reduced order model not using EIM. In this case the approach using the integration technique is faster by a factor of two. This means we get a speedup of approximately five when compared to the finite element solver. The reason for the speedup is due to the fact that no finite element assembling is needed. There is no need to evaluate the nonlinear term on the reference element in order to perform the integration. In the presented approach the integral can be carried out directly. Note that the computational cost of the integration is approximately the same as the cost for the projection of the finite element vectors and matrices. Hence, the computational advantage is that no finite element assembling costs are required.

For completeness we state the CPU times necessary for the precomputable quantities. For the finite element method we get $0.029$ seconds. In the case of the reduced order model without using EIM $0.06$ seconds are needed. When EIM is utilized we require $0.85$ seconds. This includes the generation of all matrices that do not change during the simulation process. In EIM case this includes the computation of the quantities $\Theta$ and $\Xi$ needed in the interpolation process. These times are very comparable to the results of Run 1.

| | Solve FEM | POD Basis | EIM Basis | Solve ROM | Solve ROM-EIM |
|---|---|---|---|---|---|
| CPU time | 132.00 | $3 \times 0.29$ | $2 \times 1.40$ | 24.00 | 3.97 |

Table 2.9: Run 3: Summary of the performance for the finite element and reduced order model (ROM) with and without EIM measured in seconds.

Finally, we present the relative boundary error (2.29) in Table 2.10 which is the important quantity for the underlying application. Again we observe that also in the case of this implementation the error on the boundary is small.

| | ROM | ROM-EIM |
|---|---|---|
| $\varepsilon_b^q$ | $4.7238e \times 10^{-5}$ | $4.7253 \times 10^{-5}$ |

Table 2.10: Run 3: Average relative errors on the boundary when using the reduced order model (ROM) with integration with and without EIM.

To conclude this numerical experiment let us summarize our findings. First of all a different implementation is used in order to obtain the reduced order model. This approach made it possible to successfully apply the POD model order reduction to an adaptive finite

element solution. There is no requirement to have any access to the finite element matrices nor to any finite element specific information other than the solution and the corresponding grids. This makes this method very flexible and independent of the underlying discretization method. In the numerical experiments this implementation shows to be competitive to the standard projection based approach. Moreover, it turns out that the proposed strategy is more efficient when no EIM is available.

# Chapter 3

# Parameter estimation

In this chapter the parameter estimation problem is introduced. A first optimize then discretize approach is pursued. A nonlinear least squares problem is formulated. The obtained optimization problem is interpreted as a PDE constrained optimization problem. We start by recalling basic results from the optimization theory. Then the basic concept of a sensitivity analysis is introduced. We outline how to utilize the obtained information for the parameter estimation problem. Further, different techniques for computing first and second order derivatives are investigated. Finally, the application to the nonlinear elliptic parabolic system from Chapter 1 is described. An algorithm to incorporate the techniques of Chapter 2 is outlined. Numerical results are presented for different settings to underline the efficiency of the proposed strategy.

## 3.1 Infinite dimensional optimization

In this section we will present the basic tools for optimization. For the proofs and further discussions we refer the reader to [48, 64, 66, 81].

### 3.1.1 Differentiability

First we introduce some basic results on differentiability.

**Definition 3.1.** *Let $X$ and $Y$ be Banach spaces. Further, let $U$ be a nonempty subset of $X$. Then the mapping $F : U \to Y$ is called*

*a)* directionally differentiable *at a point $x \in U$ if the limit*

$$F'(x)h = \lim_{t \to 0^+} \frac{F(x+th) - F(x)}{t}$$

*exists for all $h \in X$. $F'(x)h$ is then called the* directional derivative *of $F$ at $x$ in the direction $h$,*

*b)* Gâteaux differentiable *at a point $x \in U$ if the directional derivative exists for all $h \in X$ and the mapping $F'(x) : X \ni h \mapsto F'(x)h \in Y$ is linear and bounded,*

*c)* Fréchet differentiable *at a point $x \in U$ if $F$ is Gâteaux differentiable and in addition*

$$\lim_{\|h\|_X \to 0} \frac{\|F(x+h) - F(x) - F'(x)h\|_Y}{\|h\|_X} = 0 \tag{3.1}$$

*holds. Then $F'(x)$ is called the* Fréchet derivative *of $F$ at the point $x$.*

A mapping $F$ is called continuous if for every $\varepsilon > 0$ there exists a $\delta > 0$ such that $\|x - x_0\| < \delta$ implies $\|F(x) - F(x_0)\| < \varepsilon$. If the mapping $F : U \to Y$ is Fréchet differentiable at the point $x \in U$ then $F'$ is unique and continuous at the point $x$. If $F$ is Fréchet differentiable at the point $x \in U$ then the Gâteaux derivative exists at $x$ and the Fréchet derivative and Gâteaux derivative are equal.

### 3.1.2 Optimality results

Let us consider the general constraint optimization problem of the form

$$\min_{(z,\mu)\in Z\times\mathcal{M}} J(z,\mu) \quad \text{subject to (s.t.)} \quad e(z,\mu) = 0,\ \mu \in \mathcal{D}_{ad}, \tag{3.2}$$

where $\mathcal{M}$, $Z$, $\Lambda$ are Banach spaces, $J : Z \times \mathcal{M} \to \mathbb{R}$ is the cost function, $e : Z \times \mathcal{M} \to \Lambda$ is a nonlinear operator and $\mathcal{D}_{ad} \subset \mathcal{M}$ is a nonempty convex set.

The mapping $e(z,\mu) = 0$ is an operator that represents the PDE constraint. Assume $e : Z \times \mathcal{M} \to \Lambda$ is continuously Fréchet differentiable and its partial derivative $e_z(z,\mu)$ has an inverse. Under these conditions it can be shown that $e(z,\mu) = 0$ defines locally a continuously Fréchet differentiable mapping $\mu \mapsto z(\mu)$. This result can be obtained by the implicit function theorem.

**Theorem 3.2.** (Implicit function theorem) *Let $\mathcal{M}$, $Z$ and $\Lambda$ be Banach spaces and $G \subset Z \times \mathcal{M}$ an open subset. Further, let the mapping $e : G \to \Lambda$ be continuous Fréchet differentiable. Let $(\bar{z}, \bar{\mu}) \in G$ satisfying $e(\bar{z}, \bar{\mu}) = 0$ and $e_z(\bar{z}, \bar{\mu})$ be an invertible mapping. Then there exist a neighborhood $B = B_Z(\bar{z}) \times B_{\mathcal{M}}(\bar{\mu})$ of the point $(\bar{z}, \bar{\mu})$ and a unique continuous function $\phi : B_{\mathcal{M}}(\bar{\mu}) \to Z$ such that*

$$e(z,\mu) = 0 \Leftrightarrow z = \phi(\mu), \quad \forall\, (z,\mu) \in B \subset G \subset Z \times \mathcal{M}.$$

*We write*

$$z = z(\mu), \quad \textit{such that} \quad e(z(\mu),\mu) = 0.$$

*Moreover, the Fréchet derivative of $z$ is given by*

$$z'(\mu) = -e_z(z(\mu),\mu)^{-1} e_\mu(z(\mu),\mu).$$

We assume that the cost function $J$ and the constraint function $e$ are continuously Fréchet differentiable and for each $\mu \in \mathcal{M}$ the state equation

$$e(z,\mu) = 0$$

has a corresponding unique solution $z(\mu) \in Z$. Thus, we obtain a solution operator $\mu \in \mathcal{M} \mapsto z(\mu) \in Z$. Further, we assume that $e_z(z(\mu), \mu)$ is continuously invertible. Then it follows from the implicit function theorem that $z(\mu)$ is continuously differentiable. By differentiating the equation $e(z(\mu), \mu) = 0$ with respect to $\mu$ we obtain an equation for the derivative $z'(\mu)$ given by

$$e_z(z(\mu), \mu)z'(\mu) + e_\mu(z(\mu), \mu) = 0. \tag{3.3}$$

For ease of notation we now summarize the assumptions we have utilized.

**Assumption 3.3.** *Let $\mathcal{M}$, $Z$ and $\Lambda$ be Banach spaces. Further, $\mathcal{D}_{ad} \subset \mathcal{M}$ and $V \subset \mathcal{M}$ is a neighborhood of $\mathcal{D}_{ad}$ ($\mathcal{D}_{ad} \subset V$).*

1. *$\mathcal{D}_{ad}$ is nonempty, convex and closed.*
2. *$J : Z \times \mathcal{M} \to \mathbb{R}$ and $e : J : Z \times \mathcal{M} \to \Lambda$ are continuously Fréchet differentiable.*
3. *The state equation $e(z, \mu) = 0$ has a unique solution $z = z(\mu) \in Z$ for all $\mu \in V$.*
4. *The partial derivative $e_z(z(\mu), \mu)$ is invertible for all $\mu \in V$.*

Let Assumption 3.3 hold. By inserting $z(\mu)$ in (3.2) we can formulate the reduced problem

$$\min_{\mu \in \mathcal{M}} \hat{J}(\mu) \quad \text{s.t.} \quad \mu \in \mathcal{D}_{ad} \tag{3.4}$$

with the reduced cost function

$$\hat{J}(\mu) := J(z(\mu), \mu),$$

where $\mu \in V \mapsto z(\mu) \in Z$ is the solution operator of the state equation. Next we formulate the following general results.

**Definition 3.4.** *A parameter $\mu^\star \in \mathcal{D}_{ad}$ is called* optimal *and the corresponding $z^\star = z(\mu^\star)$ is called* optimal state *if*

$$\hat{J}(\mu^\star) \leq \hat{J}(\mu) \quad \forall \mu \in \mathcal{D}_{ad}.$$

The *Lagrange function* $\mathcal{L} : Z \times \mathcal{M} \times \Lambda^* \to \mathbb{R}$ associated with (3.2) is given by

$$\mathcal{L}(z, \mu, \lambda) = J(z, \mu) + \langle \lambda, e(z, \mu) \rangle_{\Lambda^*, \Lambda}$$

where $\lambda$ denotes the *Lagrange multiplier*. We now state the first order necessary optimality conditions.

**Theorem 3.5.** *Let Assumption* 3.3 *hold and $(z^\star, \mu^\star)$ be an optimal solution to problem* (3.2)*. Then there exists a unique associated Lagrange multiplier $\lambda^\star \in \Lambda^*$ such that the following optimality conditions hold*

$$\mathcal{L}_\lambda(z^\star, \mu^\star, \lambda^\star) = e(z^\star, \mu^\star) = 0, \tag{3.5a}$$
$$\mathcal{L}_z(z^\star, \mu^\star, \lambda^\star) = 0, \tag{3.5b}$$
$$\langle \mathcal{L}_\mu(z^\star, \mu^\star, \lambda^\star), \mu - \mu^\star \rangle_{\mathcal{M}^*, \mathcal{M}} \geq 0 \quad \forall\, \mu \in \mathcal{D}_{ad}. \tag{3.5c}$$

The optimality conditions given in (3.5) can also be written in the equivalent variational form

$$\langle \lambda, \mathcal{L}_\lambda(z^\star, \mu^\star, \lambda^\star)\rangle_{\Lambda^*,\Lambda} = \langle \lambda, e(z^\star, \mu^\star)\rangle_{\Lambda^*,\Lambda} = 0 \qquad \forall\, \lambda \in \Lambda^*, \tag{3.6a}$$

$$\langle \mathcal{L}_z(z^\star, \mu^\star, \lambda^\star), z\rangle_{Z^*,Z} = 0 \qquad \forall\, z \in Z, \tag{3.6b}$$

$$\langle \mathcal{L}_\mu(z^\star, \mu^\star, \lambda^\star), \mu - \mu^\star\rangle_{\mathcal{M}^*,\mathcal{M}} \geq 0 \qquad \forall\, \mu \in \mathcal{D}_{ad}. \tag{3.6c}$$

For the reduced problem (3.4) the corresponding optimality conditions can be formulated as follows.

**Theorem 3.6.** *Let Assumption* 3.3 *hold. If $\mu^\star$ is a local solution of the reduced problem* (3.4) *then $\mu^\star$ satisfies the variational inequality*

$$\langle \hat{J}'(\mu^\star), \mu - \mu^\star\rangle_{\mathcal{M}^*,\mathcal{M}} \geq 0 \quad \forall\, \mu \in \mathcal{D}_{ad}.$$

For completeness we state the second order sufficient optimality condition for the local minimum. Let us introduce for this purpose the Banach space $X = Z \times \mathcal{M}$.

**Theorem 3.7.** *The twice continuously Fréchet differentiable function $J$ has a local minimum at the point $x^\star = (z^\star, \mu^\star)$ under the constraint $e(x) = 0$, if there exist $\lambda^\star \in \Lambda^*$ satisfying* (3.6) *and $\kappa > 0$ such that*

$$\langle \mathcal{L}_{xx}(x^\star, \lambda^\star)v, v\rangle_{X^*,X} \geq \kappa \|v\|_X^2 \quad \forall\, v \in \mathit{ker}(e'(x^\star)).$$

## 3.2 Sensitivity analysis

In this section we have a look at the sensitivity analysis of a dynamical system. This is an interesting tool to study the parameter dependent system behavior. For this we consider a dynamical system that models a physical phenomenon of the form

$$\dot{z}(t) = \mathcal{F}(t, z(t), \mu), \quad z(0) = z_\circ(\mu), \quad t \in [0, T], \tag{3.1}$$

where $z(t)$ is the state variable and $\mu \in \mathbb{R}^p$ is a vector of system parameters. Note that the nonlinear elliptic parabolic system introduced in Chapter 1 can be written in this form. Further, let us introduce the equation

$$\eta(t) = h(t, z(t), \mu), \tag{3.2}$$

to describe the observable or measurable outputs $\eta(t)$. Let us, without loss of generality, consider the case where $\eta(t) \in \mathbb{R}$. With this assumption we reduce the amount of indices in the notation. Further, we assume that in the following $\mathcal{F}$, $h$ and $\eta$ are sufficiently smooth/regular so that the following analysis can be carried out. We want to investigate the sensitivity of the model output with respect to the parameters. We can then identify the parameter to which the model is most or least sensitive. Let us introduce the *first order sensitivity function*, also known as the *traditional sensitivity function* (TSF) [3,54]. For this

we have to build the partial derivatives of the output with respect to the parameters and we get

$$s_k(t,\mu) = \frac{\partial \eta}{\partial \mu_k}(t,\mu), \quad k = 1,\dots,p. \tag{3.3}$$

Applying the total derivative and the chain rule we get

$$s_k(t,\mu) = \frac{\partial h}{\partial z}(t,z(t,\mu),\mu)\frac{\partial z}{\partial \mu_k}(t,z(t,\mu),\mu) + \frac{\partial h}{\partial \mu_k}(t,z(t,\mu),\mu), \quad 0 \le t \le T,$$

where $Z_k(t) = \frac{\partial z}{\partial \mu_k}$ as a function of $t$ satisfies the *first order sensitivity equations*

$$\begin{aligned} \dot{Z}_k(t,\mu) &= \mathcal{F}_z(t,z(t,\mu),\mu)Z_k(t,\mu) + \mathcal{F}_{\mu_k}(t,z(t,\mu),\mu), \\ Z(0,\mu) &= \frac{\partial z_\circ}{\partial \mu_k}(\mu) \end{aligned} \tag{3.4}$$

and $z(t,\mu)$ is the solution to the initial value problem (3.1). Note that (3.4) is a linear equation and the computed sensitivity depends on the parameter $\mu$ and the time $t$. The sensitivity function (3.3) is the derivative of the output function $\eta$ with respect to the parameters and describes the changes of the output when the parameter is varied. Note that the sensitivities describe local properties.

Additionally, we have a look at the second order sensitivity. This is of particular interest when investigating parameter estimation problems. The link will be given later. The second order sensitivity function is defined by

$$\sigma_{k,m}(t,\mu) = \frac{\partial^2 \eta}{\partial \mu_k \partial \mu_m}(t,\mu), \quad k,m = 1,\dots,p.$$

Again by applying the total derivative and the chain rule we get

$$\sigma_{k,m}(t,\mu) = \frac{\partial^2 h}{\partial y^2}\frac{\partial z}{\partial \mu_k}\frac{\partial z}{\partial \mu_m} + \frac{\partial^2 h}{\partial z \partial \mu_m} + \frac{\partial h}{\partial z}\frac{\partial^2 z}{\partial \mu_k \partial \mu_m} + \frac{\partial^2 h}{\partial \mu_k \partial \mu_m}.$$

By setting $Z_{k,m} = \frac{\partial^2 z}{\partial \mu_k \partial \mu_m}$ we can formulate the second order sensitivity equations as

$$\dot{Z}_{k,m}(t,\mu) = \mathcal{F}_z Z_{k,m} + \mathcal{F}_{zz} Z_k Z_m + \mathcal{F}_{z\mu_m} Z_k + \mathcal{F}_{\mu_k z} Z_m + \mathcal{F}_{\mu_k \mu_m}, \tag{3.5}$$

together with the initial condition

$$Z_{k,m}(0,\mu) = \frac{\partial^2 z_\circ}{\partial \mu_k \partial \mu_m}(\mu).$$

Having the sensitivities $s_k$ and $\sigma_{k,m}$ one now has the possibility to make interpretations about the system. High absolute first order sensitivities mean large changes in the model output when there is a variation in the parameter. Additionally, if similar or same sensitivities are obtained for the sensitivities with respect to two different parameters same model responses can be expected.

**Remark 3.1.** *Solving the first order sensitivity equations* (3.4) *is equivalent to solving* (3.3) *when* $e(z,\mu)$ *is chosen as*

$$e(z,\mu) = \dot{z}(t) - \mathcal{F}(t,z(t),\mu).$$

### 3.2.1 The model problem: Sensitivity analysis

Next we have a look at a simple model problem to demonstrate the introduced sensitivity analysis. We consider a nonlinear parametrized heat equation. Let us introduce the parameter vector $\mu = (\mu_1, \mu_2, \mu_3) \in \mathbb{R}^3$. Then the nonlinear heat equation under consideration is given as

$$\frac{\partial z}{\partial t}(t, \mathbf{x}) - \mu_1 \Delta z(t, \mathbf{x}) = \mu_2 \sinh(\mu_3 z(t, \mathbf{x})), \tag{3.6a}$$

for $t \in [0, 1]$ and $\mathbf{x} \in \Omega = [0, 1]$ together with boundary condition and initial condition

$$z(t, 0) = 0, \quad \mu_1 \frac{\partial z}{\partial \mathrm{x}}(t, 1) = t \sin(2\pi t) \quad \text{and} \quad z(0, \mathbf{x}) = 1. \tag{3.6b}$$

Furthermore, we introduce the output function as

$$\eta(t, \mu) = h(t, z(t, \mathbf{x}); \mu) = \mathcal{P}_{\Gamma_1} z(t, \mathbf{x}) = z(t, 1). \tag{3.6c}$$

Here $\mathcal{P}_{\Gamma_1}$ is the restriction operator that restricts the variable to the boundary, i.e. it evaluates the variable at $\mathbf{x} = 1$. We are interested in the sensitivity of the output function with respect to the parameter $\mu$. For this we want to compute the sensitivity function $s(t, \mu)$. We set

$$\mathcal{F}(t, z, \mu) = \mu_1 \Delta z + \mu_2 \sinh(\mu_3 z)$$

and compute the directional derivative with respect to the variable $z$ in direction $\delta_z$

$$\frac{\partial \mathcal{F}}{\partial z}(t, z, \mu)\delta_z = \mathcal{F}_z \delta_z = \mu_1 \Delta \delta_z + \mu_2 \mu_3 \cosh(\mu_3 z)\delta_z.$$

The derivatives of $\mathcal{F}$ with respect to the parameter $\mu$ are given as

$$\begin{aligned}
\frac{\partial \mathcal{F}}{\partial \mu_1}(t, z, \mu) &= \mathcal{F}_{\mu_1} = \Delta z, \\
\frac{\partial \mathcal{F}}{\partial \mu_2}(t, z, \mu) &= \mathcal{F}_{\mu_2} = \sinh(\mu_3 z), \\
\frac{\partial \mathcal{F}}{\partial \mu_3}(t, z, \mu) &= \mathcal{F}_{\mu_3} = \mu_2 \cosh(\mu_3 z) z.
\end{aligned}$$

Inserting these into (3.4) we get for the sensitivity equations

$$\begin{aligned}
\dot{Z}_1(t, \mathbf{x}) &= \mu_1 \Delta Z_1(t, \mathbf{x}) + \mu_2 \mu_3 \cosh(\mu_3 z(t, \mathbf{x})) Z_1(t, \mathbf{x}) + \Delta z(t, \mathbf{x}), \\
\dot{Z}_2(t, \mathbf{x}) &= \mu_1 \Delta Z_2(t, \mathbf{x}) + \mu_2 \mu_3 \cosh(\mu_3 z(t, \mathbf{x})) Z_2(t, \mathbf{x}) + \sinh(\mu_3 z(t, \mathbf{x})), \\
\dot{Z}_3(t, \mathbf{x}) &= \mu_1 \Delta Z_3(t, \mathbf{x}) + \mu_2 \mu_3 \cosh(\mu_3 z(t, \mathbf{x})) Z_3(t, \mathbf{x}) + \mu_2 \cosh(\mu_3 z(t, \mathbf{x})) z(t, \mathbf{x}),
\end{aligned}$$

together with the boundary conditions

$$\mu_1 \frac{\partial Z_1}{\partial \mathbf{x}}(t, 1) = -\frac{\partial z}{\partial \mathbf{x}}(t, 1), \quad \mu_1 \frac{\partial Z_i}{\partial \mathbf{x}}(t, 1) = 0 \quad \text{for} \quad i = \{2, 3\}.$$

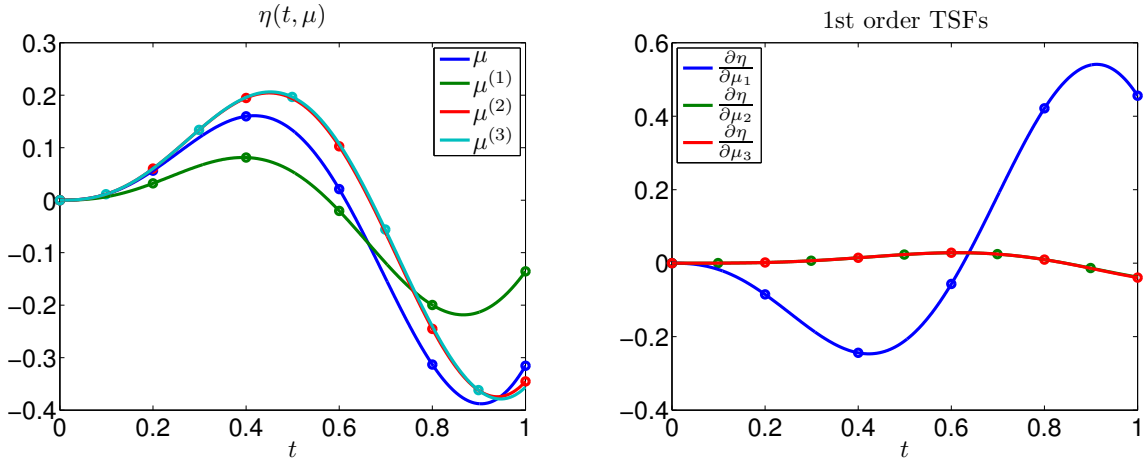


Figure 3.1: Output $\eta$ for different values of $\mu$ (left) and the first order traditional sensitivity functions (right).

and $Z_i(t,0) = 0$, for $i = 1,\dots,3$ and the initial conditions

$$Z_i(0,\mathbf{x}) = 0 \quad \text{for} \quad i = 1,\dots,3.$$

Utilizing this results we can write down the traditional sensitivity function as

$$S(t,\mu) = (\mathcal{P}_{\Gamma_1} Z_1(t,\mathbf{x}), \mathcal{P}_{\Gamma_1} Z_2(t,\mathbf{x}), \mathcal{P}_{\Gamma_1} Z_3(t,\mathbf{x})) = (Z_1(t,1), Z_2(t,1), Z_3(t,1))\,.$$

Let us next look at some numerical results. Since we want to investigate the sensitivity of the output function with respect to the parameter $\mu$ we do numerical simulations for different $\mu$ and plot the output. For this we set

$$\mu = (1,1,1), \quad \mu^{(1)} = (3,1,1), \quad \mu^{(2)} = (1,3,1), \quad \mu^{(3)} = (1,1,3)$$

and take a look at the different behavior of our output function. In Figure 3.1 (left) the numerical results are shown. It can be seen that for $\mu^{(2)}$ and $\mu^{(3)}$ nearly the same outputs are obtained. Looking at the sensitivity function, Figure 3.1 (right), it can be seen that the sensitivities at $\mu$ with respect to $\mu_2$ and $\mu_3$ are the same. This verifies the observation. The computed sensitivities provide good information of the local system behavior with respect to changes in the parameter.

Let us next have a look at the second order sensitivity for this particular example. For this we compute the second order derivatives of $\mathcal{F}$. We first look at the directional derivatives involving the variable $z$:

$$\begin{aligned}
\frac{\partial^2 \mathcal{F}}{\partial z \partial \mu_1}(t,z,\mu)\delta_z &= \mathcal{F}_{z\mu_1}\delta_z = \Delta\delta_z = \mathcal{F}_{\mu_1 z}\delta_z,\\
\frac{\partial^2 \mathcal{F}}{\partial z \partial \mu_2}(t,z,\mu)\delta_z &= \mathcal{F}_{z\mu_2}\delta_z = \cosh(\mu_3 z)\delta_z = \mathcal{F}_{\mu_2 z}\delta_z,\\
\frac{\partial^2 \mathcal{F}}{\partial z \partial \mu_3}(t,z,\mu)\delta_z &= \mathcal{F}_{z\mu_3}\delta_z = \mu_2\cosh(\mu_3 z)\delta_z + \mu_2\mu_3\sinh(\mu_3 z) z\delta_z = \mathcal{F}_{\mu_3 z}\delta_z,\\
\frac{\partial^2 \mathcal{F}}{\partial z^2}(t,z,\mu)\delta_z\bar{\delta}_z &= \mathcal{F}_{zz}\delta_z\bar{\delta}_z = \mu_2\mu_3^2\sinh(\mu_3 z)\delta_z\bar{\delta}_z.
\end{aligned}$$

Next the derivatives involving only the parameter $\mu$.

$$\begin{aligned}
\frac{\partial^2 \mathcal{F}}{\partial \mu_1 \partial \mu_1}(t,z,\mu) &= \mathcal{F}_{\mu_1\mu_1} = 0,\\
\frac{\partial^2 \mathcal{F}}{\partial \mu_1 \partial \mu_2}(t,z,\mu) &= \mathcal{F}_{\mu_1\mu_2} = 0 = \mathcal{F}_{\mu_2\mu_1},\\
\frac{\partial^2 \mathcal{F}}{\partial \mu_1 \partial \mu_3}(t,z,\mu) &= \mathcal{F}_{\mu_1\mu_3} = 0 = \mathcal{F}_{\mu_3\mu_1},\\
\frac{\partial^2 \mathcal{F}}{\partial \mu_2 \partial \mu_2}(t,z,\mu) &= \mathcal{F}_{\mu_2\mu_2} = 0,\\
\frac{\partial^2 \mathcal{F}}{\partial \mu_2 \partial \mu_3}(t,z,\mu) &= \mathcal{F}_{\mu_2\mu_3} = \cosh(\mu_3 z)z = \mathcal{F}_{\mu_3\mu_2},\\
\frac{\partial^2 \mathcal{F}}{\partial \mu_3 \partial \mu_3}(t,z,\mu) &= \mathcal{F}_{\mu_3\mu_3} = \mu_2 \sinh(\mu_3 z)zz.
\end{aligned}$$

Note that the symmetries are already indicated. Inserting these terms into (3.5) we get second order sensitivity equations. They read as

$$\begin{aligned}
\dot{Z}_{11}(t,\mathbf{x}) =& \mu_1 \Delta Z_{11}(t,\mathbf{x}) + \mu_2\mu_3 \cosh(\mu_3 z(t,\mathbf{x}))Z_{11}(t,\mathbf{x})\\
&+ \mathcal{F}_{zz} Z_1(t,\mathbf{x})Z_1(t,\mathbf{x}) + \Delta Z_1(t,\mathbf{x}) + \Delta Z_1(t,\mathbf{x}),\\
\dot{Z}_{12}(t,\mathbf{x}) =& \mu_1 \Delta Z_{12}(t,\mathbf{x}) + \mu_2\mu_3 \cosh(\mu_3 z(t,\mathbf{x}))Z_{12}(t,\mathbf{x})\\
&+ \mathcal{F}_{zz} Z_1(t,\mathbf{x})Z_2(t,\mathbf{x}) + \mathcal{F}_{z\mu_2} Z_1(t,\mathbf{x}) + \Delta Z_2(t,\mathbf{x}),\\
\dot{Z}_{13}(t,\mathbf{x}) =& \mu_1 \Delta Z_{13}(t,\mathbf{x}) + \mu_2\mu_3 \cosh(\mu_3 z(t,\mathbf{x}))Z_{13}(t,\mathbf{x})\\
&+ \mathcal{F}_{zz} Z_1(t,\mathbf{x})Z_3(t,\mathbf{x}) + \mathcal{F}_{z\mu_3} Z_1(t,\mathbf{x}) + \Delta Z_3(t,\mathbf{x}),\\
\dot{Z}_{22}(t,\mathbf{x}) =& \mu_1 \Delta Z_{22}(t,\mathbf{x}) + \mu_2\mu_3 \cosh(\mu_3 z(t,\mathbf{x}))Z_{22}(t,\mathbf{x})\\
&+ \mathcal{F}_{zz} Z_2(t,\mathbf{x})Z_2(t,\mathbf{x}) + \mathcal{F}_{z\mu_2} Z_2(t,\mathbf{x}) + \mathcal{F}_{\mu_2 z} Z_2(t,\mathbf{x}),\\
\dot{Z}_{23}(t,\mathbf{x}) =& \mu_1 \Delta Z_{23}(t,\mathbf{x}) + \mu_2\mu_3 \cosh(\mu_3 z(t,\mathbf{x}))Z_{23}(t,\mathbf{x})\\
&+ \mathcal{F}_{zz} Z_2(t,\mathbf{x})Z_3(t,\mathbf{x}) + \mathcal{F}_{z\mu_3} Z_2(t,\mathbf{x}) + \mathcal{F}_{\mu_2 z} Z_3(t,\mathbf{x}) + \mathcal{F}_{\mu_2\mu_3},\\
\dot{Z}_{33}(t,\mathbf{x}) =& \mu_1 \Delta Z_{33}(t,\mathbf{x}) + \mu_2\mu_3 \cosh(\mu_3 z(t,\mathbf{x}))Z_{33}(t,\mathbf{x})\\
&+ \mathcal{F}_{zz} Z_3(t,\mathbf{x})Z_3(t,\mathbf{x}) + \mathcal{F}_{z\mu_3} Z_3(t,\mathbf{x}) + \mathcal{F}_{\mu_3 z} Z_3(t,\mathbf{x}) + \mathcal{F}_{\mu_3\mu_3}
\end{aligned}$$

and

$$Z_{12}(t,\mathbf{x}) = Z_{21}(t,\mathbf{x}), \quad Z_{13}(t,\mathbf{x}) = Z_{31}(t,\mathbf{x}), \quad Z_{23}(t,\mathbf{x}) = Z_{32}(t,\mathbf{x}).$$

This is together with the boundary conditions $Z_{ij}(t,0) = 0$ for $1 \le i,j \le 3$ and

$$\mu_1 \frac{\partial Z_{11}}{\partial \mathbf{x}}(t,1) = -2\frac{\partial Z_1}{\partial \mathbf{x}}(t,1), \quad \mu_1 \frac{\partial Z_{ij}}{\partial \mathbf{x}}(t,1) = 0$$

for all $1 \le i,j \le 3$ except for $i = j = 1$. The initial conditions are given by

$$Z_{ij}(0,\mathbf{x}) = 0, \quad \text{for} \quad 1 \le i,j \le 3.$$

Hence, it follows that six additional linear PDEs have to be solved.

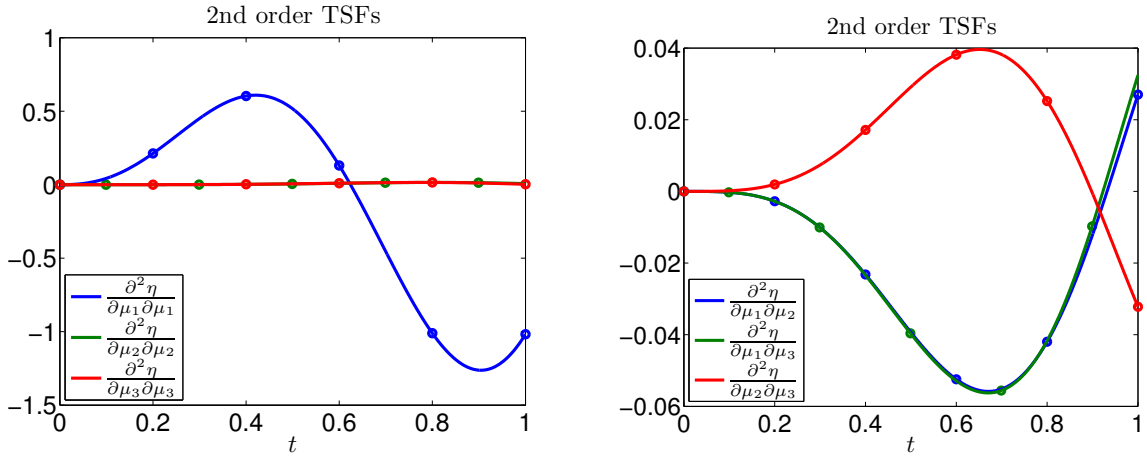


Figure 3.2: Second order sensitivities for the model problem.

In Figure 3.2 the six second order traditional sensitivity equations are shown. In the left plot a similar result can be seen as in the right of Figure 3.1. The second derivatives with respect to the parameter $\mu_2$ and $\mu_3$ are the same. In the left plot of Figure 3.2 the mixed derivatives are shown. In this case the derivatives involving $\mu_1$ and $\mu_2$ or $\mu_3$ deliver the same second order sensitivity. This underlines again that a change in the parameter $\mu_2$ and $\mu_3$ deliver the same output, even for the derivative $\frac{\partial\eta}{\partial\mu_1}$. The second derivatives are a good tool to verify the results of the first order sensitivities. In most cases the computational expenses are too high and therefore are not computed.

## 3.3 Subset selection

To address the question of identifiability of parameters we apply a subset selection method. These methods extract parameters that are either insensitive or correlated. There are different approaches to handle this task and comparisons have been carried out [69, 71]. In certain cases the one or the other method fails which is outlined in [42]. We apply the strategy of using a singular value decomposition (eigenvalue decomposition) followed by a QR factorization with column pivoting which corresponds to the ideas introduced in [35, 36]. In our application this method turns out to be computationally very efficient. This method has also been used in various applications [12, 31, 71].

The methods rely on the sensitivity Jacobian matrix $S$ which is given in our case in the following form:

$$S(t,\mu) = \begin{pmatrix} s_1(t,\mu) & \cdots & s_p(t,\mu) \end{pmatrix}.$$

Considering the case that the sensitivities are given at specific time points $t_0 < t_1 < \ldots < t_{N_t}$, e.g. as numerical solutions, we write the sensitivity Jacobian as

$$S(\mu) = \begin{pmatrix} s_1(t_0,\mu) & \cdots & s_p(t_0,\mu) \\ \vdots & & \vdots \\ s_1(t_{N_t},\mu) & \cdots & s_p(t_{N_t},\mu) \end{pmatrix}.$$

In some cases it might be necessary to use the normalized sensitivities [6]. In our application both worked equally well. By $\delta\mu = \mu - \mu^{(0)}$ we denote a small perturbation in the parameter $\mu$. The corresponding perturbation in the output is then given by $\delta\eta = \eta(t,\mu) - \eta(t,\mu^{(0)})$. Using the Taylor expansion we get

$$\delta\eta = S\delta\mu + \mathcal{O}(\|\delta\mu\|^2).$$

Under the assumption that the perturbation $\delta\mu$ is small we can write this as

$$\delta\eta \approx S\delta\mu.$$

In the notion of sensitivity identifiability [73] the system parameters are said to be identifiable if and only if $S$ has full rank, i.e. $\operatorname{rank}(S) = p$. This is equivalent to $\det(S^\top S) \neq 0$. The matrix $S^\top S$ is also referred as the *Fisher information matrix* [3]. In the later algorithm and optimization context, this matrix will also be referred as the Gauss-Newton matrix. The matrix $S$ is evaluated at a nominal parameter, hence the obtained results and properties are valid only locally. In the least squares setting, where the output $\eta$ should track given data $\eta^d$ over time, $S$ corresponds to the derivative of the residual $\eta - \eta^d$. Hence, a weighing can be applied. In the continuous case this weighing corresponds to the integration over time. Hence, $S^\top S$ can then be interpreted as

$$\big(\big(H(\mu)\big)\big)_{ij} = \int_0^T \frac{\partial\eta}{\partial\mu_i}(t,\mu)\frac{\partial\eta}{\partial\mu_j}(t,\mu)\,\mathrm{d}t, \tag{3.7}$$

for $1 \leq i,j \leq p$. Note that $H$ is a symmetric and positive semidefinite matrix. The idea of the subset selection method is now to identify parameters which are not identifiable. Hence, the eigenvalues of $S^\top S$ or the singular values of $S$ are computed. Numerically an accuracy tolerance $\varepsilon_{ss}$ is introduced in order to decide which eigenvalues to neglect. The associated parameters are then omitted in the identification process. To match the eigenvalues to the parameters a QR decomposition is performed. The idea is to use a column pivoting strategy to compute a permutation matrix $\Pi$ so that the matrix satisfies $H\Pi = QR$. The permutation matrix $\Pi$ is then used to reorder the parameters. Let us summarize the subset selection method to eliminate non-identifiable parameters in Algorithm 6 [12, 38]. By applying the subset selection method it is guaranteed that the matrix $H$ is invertible. In the Gauss-Newton context this means that the search direction can be computed.

**Algorithm 6** (Subset selection method)

**Require:** Given $H(\mu) \in \mathbb{R}^{p\times p}$ symmetric positive semidefinite, $\mu$ and $\varepsilon_{ss}$

1: Compute the nonnegative eigenvalues $\{\lambda_i\}_{i=1}^p$ of $H = V\Lambda V^\top$ and truncate at $\varepsilon_{ss} > 0$, i.e.

$$\lambda_1 \geq \ldots \geq \lambda_k \geq \varepsilon_{ss} \quad \text{with} \quad k \in \{1,\ldots,p\}.$$

2: Use the $QR$ method with column-pivoting to reorder $\mu$, i.e.

$$H\Pi = QR \quad \text{and} \quad \bar{\mu} = \Pi^\top\mu.$$

3: Select the first $k$ parameters of $\bar{\mu}$ as new parameter and set the others to academic/reasonable values.

### 3.3.1 The model problem: Subset selection

As we have seen in Section 3.1 multiple parameters can locally describe the same model behavior. Hence, in an optimization process, e.g. least squares problem, some parameters might not be identifiable.

Let us now apply the subset selection strategy using the QR method to the example introduced in Section 3.1. For this we first compute the matrix $H$ by using the sensitivity functions. For $\mu = (1, 1, 1)$ we get

$$H = \begin{pmatrix} 7.696145426910515 & -0.184957719825766 & -0.192520839855786 \\ -0.184957719825766 & 0.030774713253229 & 0.031145691717736 \\ -0.192520839855786 & 0.031145691717736 & 0.031529750791884 \end{pmatrix}$$

together with the eigenvalues

$$7.705470551987061 \geq 0.052977210205481 \geq 0.000002128763087 \geq 0.$$

We consider the last eigenvalue to be small and hence neglect it. So we split the eigenvalues into the groups $\{\lambda_1, \lambda_2\}$ and $\{\lambda_3\}$. Applying the subset selection strategy we get that the parameters corresponding to these eigenvalues have the index $1$ and $3$. Hence, the parameter $\mu_2$ should be fixed. With this knowledge a parameter estimation should only be formulated using the parameters $\mu_1$ and $\mu_3$. This result did not come with a surprise since we already observed in the sensitivities that the output function has the same response for a change in the parameters $\mu_2$ and $\mu_3$.

## 3.4 Nonlinear least squares problem

In this section we formulate a least squares problem and investigate two techniques for computing first and second order derivatives. Let us here consider a nonlinear least squares parameter estimation problem of the form

$$\mu^\star = \arg\min_{\mu} \hat{J}(\mu) = \frac{1}{2} \int_0^T \|\eta(t, \mu) - \eta^d(t)\|^2 \, \mathrm{d}t. \tag{3.8}$$

Note that $\eta(t, \mu)$ is an observable or measurable output (3.2) to an initial value problem of the type (3.1). We assume that the target $\eta^d$ is given as a function of time. More commonly it is assumed that $\eta^d$ is only given at particular points in time denoted by $t_i$ , $i = 0, \ldots, N_t$. Then writing (3.8) in the discrete form we get

$$\mu^\star = \arg\min_{\mu} \hat{J}(\mu) = \frac{1}{2} \sum_{i=0}^{N_t} \alpha_i \|\eta(t_i, \mu) - \eta^d(t_i)\|^2, \tag{3.9}$$

where $\alpha_i$ correspond to the integration weights associated to $t_i$ in order to approximate the integral. Trapezoidal weights of the form

$$\alpha_0 = \frac{t_1 - t_0}{2}, \quad \alpha_i = \frac{t_{i+1} - t_{i-1}}{2}, \quad i = 1, \ldots, N_t - 1, \quad \text{and} \quad \alpha_0 = \frac{t_{N_t} - t_{N_t - 1}}{2}$$

can be used. To solve the minimization problem we recall the optimality conditions. Let us now investigate two possibilities of computing the derivative of the continuous formulation of the least squares problem (3.8).

### 3.4.1 The sensitivity approach

In the so-called sensitivity approach we compute the derivative of the cost with respect to the parameter $\mu \in \mathbb{R}^p$ by applying the chain and product rule. We get

$$\frac{\partial \hat{J}(\mu)}{\partial \mu_i} = \int_0^T \big(\eta(t,\mu) - \eta(t)\big) \frac{\partial \eta}{\partial \mu_i}(t,\mu)\,\mathrm{d}t \quad \text{for } i = 1,\dots,p. \tag{3.10}$$

Recall the results from Section 3.2, the quantity $\frac{\partial \eta}{\partial \mu_i}$ corresponds to the introduced sensitivities $s_i(t,\mu)$. The gradient is then given as

$$\hat{J}'(\mu) = \left( \frac{\partial \hat{J}(\mu)}{\partial \mu_1}, \dots, \frac{\partial \hat{J}(\mu)}{\partial \mu_p} \right)^\top.$$

Applying the chain and the product rule another time the components of the Hessian are given as

$$\frac{\partial^2 \hat{J}(\mu)}{\partial \mu_i \partial \mu_j} = \int_0^T \frac{\partial \eta}{\partial \mu_i}(t,\mu) \frac{\partial \eta}{\partial \mu_j}(t,\mu)\,\mathrm{d}t + \int_0^T \big(\eta(t,\mu) - \eta^d(t)\big) \frac{\partial^2 \eta}{\partial \mu_i \partial \mu_j}(t,\mu)\,\mathrm{d}t \tag{3.11}$$

for $1 \le i,j \le p$ and the Hessian reads as

$$\hat{J}''(\mu) = \begin{pmatrix} \frac{\partial^2 \hat{J}(\mu)}{\partial \mu_1 \partial \mu_1} & \cdots & \frac{\partial^2 \hat{J}(\mu)}{\partial \mu_1 \partial \mu_p} \\ \vdots & & \vdots \\ \frac{\partial^2 \hat{J}(\mu)}{\partial \mu_p \partial \mu_1} & \cdots & \frac{\partial^2 \hat{J}(\mu)}{\partial \mu_p \partial \mu_p} \end{pmatrix}.$$

For the Hessian additionally the second order sensitivities $\sigma_{ij}(t,\mu) = \frac{\partial^2 \eta}{\partial \mu_i \partial \mu_j}(t,\mu)$ are required. The second order sensitivities are computed as outlined in Section 3.2. Note that the computational expenses increase. For the gradient it is required to solve $p$ first order sensitivity equations. In order to obtain the Hessian we additionally need to solve $(p+1)p/2$ second order sensitivities (under the assumption the Hessian is symmetric). This becomes unfeasible already for small numbers of parameters. Recall that all the sensitivities are obtained by solving linear systems.

### 3.4.2 The Lagrangian-based adjoint approach

Now we want to derive the gradient using the Lagrange function. For this we formulate the nonlinear least squares problem in the form

$$\min_{(z,\mu) \in Z \times \mathcal{M}} J(z,\mu) \quad \text{s.t.} \quad e(z,\mu) = 0 \tag{3.12}$$

with $J : Z \times \mathcal{M} \to \mathbb{R}$ the cost function

$$J(z, \mu) = \frac{1}{2} \int_0^T \|h(t, z, \mu) - \eta^d(t)\|^2 \mathrm{d}t$$

and $e : Z \times \mathcal{M} \to \Lambda$ an operator between Banach spaces given by

$$e(z, \mu) = \dot{z}(t) - \mathcal{F}(t, z, \mu).$$

By assuming that $J$ and $e$ are continuously Fréchet-differentiable and the state equation $e(z, \mu) = 0$ possesses for each $\mu \in \mathcal{M}$ a unique corresponding solution $z(\mu) \in Z$ we can introduce the solution operator $\mu \in \mathcal{M} \mapsto z(\mu) \in Z$. Then (3.12) can be written in the reduced form

$$\min_{\mu \in \mathcal{M}} \hat{J}(\mu) := J(z(\mu), \mu)$$

which then again corresponds to (3.8). To compute the first and second order derivatives we here want to look at the Lagrangian-based adjoint approach. For this we define the Lagrange function $\mathcal{L} : Z \times \mathcal{M} \times \Lambda^* \to \mathbb{R}$,

$$\mathcal{L}(z, \mu, \lambda) = J(z, \mu) + \langle \lambda, e(z, \mu) \rangle_{\Lambda^*, \Lambda}.$$

Now by choosing an arbitrary $\lambda \in \Lambda^*$ and setting $z = z(\mu)$ we get

$$\hat{J}(\mu) = J(z(\mu), \mu) = J(z(\mu), \mu) + \langle \lambda, e(z(\mu), \mu) \rangle_{\Lambda^*, \Lambda} = \mathcal{L}(z(\mu), \mu, \lambda).$$

Hence, we can build the directional derivative in direction $\delta_\mu^1 \in \mathcal{M}$ of the reduced cost functional and obtain

$$\langle \hat{J}'(\mu), \delta_\mu^1 \rangle_{\mathcal{M}^*, \mathcal{M}} = \langle \mathcal{L}_z(z(\mu), \mu, \lambda), z'(\mu)\delta_\mu^1 \rangle_{Z^*, Z} + \langle \mathcal{L}_\mu(z(\mu), \mu, \lambda), \delta_\mu^1 \rangle_{\mathcal{M}^*, \mathcal{M}}. \tag{3.13}$$

Choosing now $\lambda = \lambda(\mu)$ such that

$$\mathcal{L}_z(z(\mu), \mu, \lambda(\mu)) = 0. \tag{3.14}$$

Note that this is in fact the adjoint equation

$$\begin{aligned} \langle \mathcal{L}_z(z, \mu, \lambda), \delta_z^1 \rangle_{Z^*, Z} &= \langle J_z(z, \mu), \delta_z^1 \rangle_{Z^*, Z} + \langle \lambda, e_z(z, \mu)\delta_z^1 \rangle_{\Lambda^*, \Lambda} \\ &= \langle J_z(z, \mu) + e_z(z, \mu)^* \lambda, \delta_z^1 \rangle_{Z^*, Z} \end{aligned}$$

with $\delta_z^1 \in Z$. Hence, we get

$$\mathcal{L}_z(z(\mu), \mu, \lambda) = J_z(z(\mu), \mu) + e_z(z(\mu), \mu)^* \lambda.$$

Choosing $\lambda$ as in (3.14) we obtain for the derivative of the reduced cost function

$$\hat{J}'(\mu) = \mathcal{L}_\mu(z(\mu), \mu, \lambda(\mu)) = J_\mu(z(\mu), \mu) + e_\mu(z(\mu), \mu)^* \lambda(\mu). \tag{3.15}$$

Note that in the investigated case the partial derivatives of $J$ are given as

$$J_z(z, \mu)\delta_z^1 = \int_0^T (h(t, z, \mu) - \eta^d(t)) \frac{\partial h}{\partial z}(t, z, \mu)\delta_z^1 \, \mathrm{d}t$$

and

$$J_\mu(z,\mu)\delta_\mu^1 = \int_0^T (h(t,z,\mu) - \eta^d(t))\frac{\partial h}{\partial \mu}(t,z,\mu)\delta_\mu^1 \,\mathrm{d}t.$$

Let us now turn to the second derivative. Starting from (3.13) we differentiate once again with direction $\delta_\mu^2 \in \mathcal{M}$.

$$\begin{aligned}\langle \hat{J}''(\mu)\delta_\mu^2, \delta_\mu^1\rangle_{\mathcal{M}^*,\mathcal{M}} =& \langle \mathcal{L}_z(z(\mu),\mu,\lambda), z''(\mu)(\delta_\mu^1,\delta_\mu^2)\rangle_{Z^*,Z} \\ &+ \langle \mathcal{L}_{zz}(z(\mu),\mu,\lambda)z'(\mu)\delta_\mu^2, z'(\mu)\delta_\mu^1\rangle_{Z^*,Z} \\ &+ \langle \mathcal{L}_{z\mu}(z(\mu),\mu,\lambda)\delta_\mu^2, z'(\mu)\delta_\mu^1\rangle_{Z^*,Z} \\ &+ \langle \mathcal{L}_{\mu z}(z(\mu),\mu,\lambda)z'(\mu)\delta_\mu^2, \delta_\mu^1\rangle_{\mathcal{M}^*,\mathcal{M}} \\ &+ \langle \mathcal{L}_{\mu\mu}(z(\mu),\mu,\lambda)\delta_\mu^2, \delta_\mu^1\rangle_{\mathcal{M}^*,\mathcal{M}}.\end{aligned}$$

Now again choosing $\lambda = \lambda(\mu)$ in the way that (3.14) is satisfied the term involving $z''(\mu)$ drop out and we get

$$\begin{aligned}\langle \hat{J}''(\mu)\delta_\mu^2, \delta_\mu^1\rangle_{\mathcal{M}^*,\mathcal{M}} =& \langle \mathcal{L}_{zz}(z(\mu),\mu,\lambda(\mu))z'(\mu)\delta_\mu^2, z'(\mu)\delta_\mu^1\rangle_{Z^*,Z} \\ &+ \langle \mathcal{L}_{z\mu}(z(\mu),\mu,\lambda(\mu))\delta_\mu^2, z'(\mu)\delta_\mu^1\rangle_{Z^*,Z} \\ &+ \langle \mathcal{L}_{\mu z}(z(\mu),\mu,\lambda(\mu))z'(\mu)\delta_\mu^2, \delta_\mu^1\rangle_{\mathcal{M}^*,\mathcal{M}} \\ &+ \langle \mathcal{L}_{\mu\mu}(z(\mu),\mu,\lambda(\mu))\delta_\mu^2, \delta_\mu^1\rangle_{\mathcal{M}^*,\mathcal{M}}.\end{aligned}$$

Hence, we arrive at

$$\begin{aligned}\hat{J}''(\mu) =& z'(\mu)^*\mathcal{L}_{zz}(z(\mu),\mu,\lambda(\mu))z'(\mu) + z'(\mu)^*\mathcal{L}_{z\mu}(z(\mu),\mu,\lambda(\mu)) \\ &+ \mathcal{L}_{\mu z}(z(\mu),\mu,\lambda(\mu))z'(\mu) + \mathcal{L}_{\mu\mu}(z(\mu),\mu,\lambda(\mu))\end{aligned} \tag{3.16}$$

for the second derivative. Alternatively (3.16) can be written as

$$\hat{J}''(\mu) = T(\mu)^* L_{xx}(y(\mu),\mu,p(\mu))T(\mu)$$

with

$$T(\mu) = \begin{pmatrix} z'(\mu) \\ I_{\mathcal{M}} \end{pmatrix} \quad \text{and} \quad \mathcal{L}_{xx} = \begin{pmatrix} \mathcal{L}_{zz} & \mathcal{L}_{z\mu} \\ \mathcal{L}_{\mu z} & \mathcal{L}_{\mu\mu} \end{pmatrix},$$

where $I_{\mathcal{M}}$ is the identity on $\mathcal{M}$ and $z'(\mu)$ solves the sensitivity equation of the form

$$z'(\mu) = -e_z(z(\mu),\mu)^{-1}e_\mu(z(\mu),\mu),$$

which is equivalent to (3.4). Note that by introducing the adjoint equation the sensitivities of second order are not needed. This is computationally a big advantage of this approach. Further, the second derivative in a direction $s$, i.e. $\hat{J}''(\mu)s$ can also be computed without the use of $z'(\mu)$. Since we will here use a combination of the two introduced approaches for computing the first and second derivatives we refer the reader to [48] for more details.

Let us remark that to compute the Hessian using the Lagrangian-based approach we require $p$ first order sensitivities and the solution to the adjoint equation. Comparing this to the computational effort required to obtain the Hessian using the sensitivity approach this is much more affordable and feasible.

## 3.5 Numerical methods

This section is devoted to two numerical algorithms to solve the nonlinear parameter estimation problem. There are several different strategies available [55, 68, 82]. Out of the many in this work we will utilize the Gauss-Newton and the Newton method. For this we shortly introduce these two strategies. Additionally, we shortly describe the SQP method to outline a possible incorporation of bound constraints on the parameter.

### 3.5.1 Gauss-Newton method

The problem under consideration is of least squares type. We have seen how to compute first and second derivatives of the cost function with respect to the parameters. Let us recall the representation of the Hessian (3.11) in the sensitivity approach presented in Section 3.4.1. The Gauss-Newton approximation of the Hessian is then given by

$$\big(\big(H(\mu)\big)\big)_{ij} = \int_0^T \frac{\partial \eta}{\partial \mu_i}(t,\mu)\frac{\partial \eta}{\partial \mu_j}(t,\mu)\,\mathrm{d}t \quad \text{for} \quad 1 \leq i,j \leq p. \tag{3.17}$$

In this approximation the term involving the second order sensitivities is neglected which is a computational advantage since it is cheaper to compute than the full Hessian. Note that when computing the gradient using (3.10) for the gradient computation all quantities for the Gauss-Newton matrix are already available. The matrix $H$ corresponds to the Fisher information matrix introduced in Section 3.3. Note that the matrix $H$ is positive semidefinite, and it is positive definite if the sensitivity Jacobian has full rank. Recall the subset selection method introduced in Section 3.3. In the numerical experiments this method will guarantee that the matrix is positive definite. Hence, it is ensured that the Gauss-Newton step

$$H(\mu)d_{\mathrm{GN}} = -\hat{J}'(\mu)$$

is a descent direction, i.e. $\langle J'(\mu), d_{\mathrm{GN}}\rangle < 0$. Further, since $H$ is positive definite efficient iterative solvers can be used to solve the Gauss-Newton step if large scale problems are considered, e.g. the conjugate gradient method.

The Gauss-Newton approximation is obtained by neglecting the second order sensitivities. This is done under the assumption that $\eta(t,\mu) - \eta(t)$ is small or tends to zero. If this is true the first order term in the Hessian dominates the second order term and a good approximation is obtained. Hence, local quadratic convergence can be observed. In the case that this assumption does not hold the rate may be linear or the method might not converge if $\eta(t,\mu) - \eta(t)$ is too large. An extension to the Gauss-Newton method is the *Levenberg-Marquardt method*, where the Hessian is approximated by $H + \gamma I$ with a parameter $\gamma \geq 0$.

### 3.5.2 Newton method

Opposed to the Gauss-Newton method in the Newton method the full Hessian is used. This means that the full Hessian has to be computed aside of the gradient. This makes the

Newton method more expensive and also requires that the Hessian can be solved accurately. The Newton step is then computed by

$$\hat{J}''(\mu)d_{\mathrm{N}} = -\hat{J}(\mu).$$

The Hessian might not always be positive definite, hence the obtained $d_{\mathrm{N}}$ must not be a search direction. Therefore, a globalization strategy has to be applied. The obtained convergence is again locally quadratic. Similar theorems as Theorem 1.14 can be formulated [68]. To globalize the Newton method strategies like the *Armijo condition* or *trust region methods* are utilized [55, 68].

### 3.5.3 SQP method

The idea of the *sequential quadratic programming* (SQP) method is to replace the nonlinear constrained optimization problem locally by quadratic approximations. This is similar to the Gauss-Newton and Newton method. The main difference is that additionally inequality constraints can be incorporated. For this we formulate the problem as follows:

$$\min_{\mu\in\mathcal{M}} \hat{J}(\mu) \quad \text{s.t.} \quad h(\mu) \geq 0. \tag{3.18}$$

We are in particular interested in inequality constraints on the parameter $\mu$. These are constraints that can be given by the application or even by analytical results. We consider constraints of the following form

$$\underline{\mu_i} \leq \mu_i \leq \overline{\mu_i} \quad \text{for } i = 1, \ldots, p$$

with $\underline{\mu_i}, \overline{\mu_i} \in \mathbb{R}$ and $\underline{\mu_i} \leq \overline{\mu}_i$. This can then be formulated with the help of the function $h$ or simply by the introduction of an admissible set. We then rewrite (3.18) as

$$\min_{\mu\in\mathcal{M}} \hat{J}(\mu) \quad \text{s.t.} \quad \mu \in \mathcal{D}_{ad}, \tag{3.19}$$

where $\mathcal{D}_{ad} = \{\mu \in \mathbb{R}^p \,|\, \underline{\mu_i} \leq \mu_i \leq \overline{\mu}_i\} \subseteq \mathcal{M}$. We define $\mu^{k+1} = \mu^k + d_{\mathrm{SQP}}$ and now replace (3.19) by a sequence of quadratic problems of the type

$$\min_{d_{\mathrm{SQP}}\in\mathcal{M}} \frac{1}{2}(d_{\mathrm{SQP}})^\top \hat{J}''(\mu^k)d_{\mathrm{SQP}} + (\hat{J}'(\mu^k))^\top d_{\mathrm{SQP}} + J(\mu^k) \quad \text{s.t.} \quad \mu^{k+1} \in \mathcal{D}_{ad},$$

where in each iteration a quadratic optimal control problem has to be solved. There are several strategies to solve these problems, we refer the reader to e.g. [68]. The problem can then be reformulated in $d_{\mathrm{SQP}}$ as

$$\min_{d_{\mathrm{SQP}}\in\mathcal{M}} \frac{1}{2}(d_{\mathrm{SQP}})^\top \hat{J}''(\mu^k)d_{\mathrm{SQP}} + (\hat{J}'(\mu^k))^\top d_{\mathrm{SQP}} \quad \text{s.t.} \quad d_{\mathrm{SQP}} \in \mathcal{D}_{ad,k},$$

where all constant terms with respect to $d_{\mathrm{SQP}}$ are neglected and $\mathcal{D}_{ad,k} = \{d \in \mathbb{R}^p \,|\, \underline{\mu_i} - \mu_i^k \leq d_i \leq \overline{\mu}_i - \mu_i^k\}$. In the quadratic problems the Hessian $\hat{J}''(\mu)$ can then again be replaced by the Gauss-Newton matrix. Alternatively to the proposed strategy the optimization problem (3.19) can be solved directly by a projected Newton method [55] or semi-smooth Newton method [51]. Since there are no equality constraints considered in (3.19) these methods only differ in some technical aspects.

## 3.6 The parameter estimation problem

In this section we introduce the parameter estimation problem involving the nonlinear system (1.1) introduced in Chapter 1. The problem will be formulated as a nonlinear least squares problem. The gradient will be derived using the sensitivity and the adjoint approach. A numerical algorithm for solving the parameter estimation utilizing the reduced order model from Chapter 2 will be introduced. Numerical examples will be presented to underline the efficiency of the proposed method.

### 3.6.1 Problem formulation

Let us start by introducing the parameter identification problem. The parameters to identify are given by $\mu = (\mu_1, \mu_2, \mu_3, \mu_4) \in \mathbb{R}^4$ and the target is given by $q^d(t) \in \mathbb{R}$, for $t \in (0, T]$. Note that we only have a target on $q$ at the point $b$. This setting is motivated by the application in the modeling of lithium ion batteries, where only the difference of the potential can be measured, which is given as the difference between the two boundary points of $q$ [13, 33, 84]. We consider a cost functional of the form

$$J(z, \mu) = \frac{1}{2} \int_0^T |q(t, b) - q^d(t)|^2 \, \mathrm{d}t,$$

where $q$ is part of the solution triple $z = (y, p, q)$ that solves the nonlinear system (1.1) from Chapter 1. Let us additionally impose constraints on the parameters $\mu$ of the form

$$\underline{\mu}_i \leq \mu_i \leq \overline{\mu}_i \quad \text{for} \quad i = 1, \ldots, 4$$

and define the admissible set as

$$\mathcal{D}_{ad} = \{\mu \in \mathbb{R}^4 \,|\, \underline{\mu}_i \leq \mu_i \leq \overline{\mu}_i \,\text{for}\, i = 1, \ldots, 4\} \subseteq \mathcal{M}_{ad}.$$

This constraint is introduced in order to derive more general results. The minimization problem can be formulated as

$$\min J(z, \mu) \quad \text{s.t.} \quad (z, \mu) \text{ solves (1.1) and } \mu \in \mathcal{D}_{ad} \tag{3.20}$$

and in reduced form as

$$\min \hat{J}(\mu) \quad \text{s.t.} \quad \mu \in \mathcal{D}_{ad}, \tag{3.21}$$

where $z(\mu) = \mathcal{S}(\mu)$ is the solution operator for the nonlinear system (1.1). Note that for every $\mu \in \mathcal{M}_{ad}$ there exists a unique $z(\mu)$ following Theorem 1.4. To derive formally the first order optimality conditions for (3.20) we need to compute derivatives. Let us apply the Lagrange approach. Recall that the first order optimality conditions [48, 81] are given by

$$\mathcal{L}_z(z^\star, \mu^\star, \lambda^*)\delta z = 0 \qquad \text{for all } \delta z \in \mathcal{Z}_{ad}, \tag{3.22}$$

$$\mathcal{L}_\mu(z^\star, \mu^\star, \lambda^\star)(\mu - \mu^*) \geq 0 \qquad \text{for all } \mu \in \mathcal{D}_{ad}, \tag{3.23}$$

where (3.22) is the adjoint equation, (3.23) the variational inequality and $^\star$ indicates a local optimal solution. Note that following Assumption 3.3 a unique Lagrange multiplier exists to every solution pair $(z^\star, \mu^\star)$. For this we define $e(z,\mu) = (e^y(z,\mu), e^p(z,\mu), e^q(z,\mu))^\top$, where each component corresponds to (1.1a), (1.1b) and (1.1c), respectively. The Lagrange function for (3.20) is then given as

$$\begin{aligned}\mathcal{L}(z,\mu,\lambda) =& \frac{1}{2}\int_0^T |q(t,b) - q^d(t)|^2 \,\mathrm{d}t \\ &+ \int_0^T \int_\Omega y_t(t)\lambda^y(t) + c_1 y_{\mathbf{x}}(t)\lambda^y_{\mathbf{x}}(t) + \mathcal{N}(\,\cdot\,, y(t), p(t), q(t); \mu)\lambda^y(t)\,\mathrm{d}\mathbf{x}\mathrm{d}t \\ &+ \int_0^T \int_\Omega c_2(y(t);\mu)p_{\mathbf{x}}(t)\lambda^p_{\mathbf{x}}(t) + \mathcal{N}(\,\cdot\,, y(t), p(t), q(t); \mu)\lambda^p(t)\,\mathrm{d}\mathbf{x}\mathrm{d}t \\ &+ \int_0^T \int_\Omega c_3 q_{\mathbf{x}}(t)\lambda^q_{\mathbf{x}}(t) - \mathcal{N}(\,\cdot\,, y(t), p(t), q(t); \mu)\lambda^q(t)\,\mathrm{d}\mathbf{x}\mathrm{d}t \\ &+ \int_0^T \int_\Gamma \left(\frac{\partial q}{\partial n} - \mathcal{I}(t)\right)\lambda^q(t)\,\mathrm{d}S\mathrm{d}t + \int_\Omega (y(0) - y_\circ)\lambda^y(0)\,\mathrm{d}\mathbf{x}\end{aligned}$$

for $\lambda = (\lambda^y, \lambda^p, \lambda^q)^\top$. Note that we do not introduce additional Lagrange multipliers for the boundary and initial conditions but rather apply known results from [81]. Let us next derive the adjoint system for our system which is given by the derivative of $\mathcal{L}$ with respect to the variable $z = (y, p, q)$. All the required derivatives of the nonlinear terms are given in Appendix A. We get

$$\begin{aligned}&\int_0^T \int_\Omega \delta y_t(t)\lambda^y(t) + c_1\delta y_{\mathbf{x}}(t)\lambda^y_{\mathbf{x}} + c_{2,y}(y(t);\mu)\delta y(t)p_{\mathbf{x}}(t)\lambda^p_{\mathbf{x}}(t)\,\mathrm{d}\mathbf{x}\mathrm{d}t \\ &\qquad + \int_0^T \int_\Omega \mathcal{N}_y(\,\cdot\,, y(t), p(t), q(t); \mu)\delta y(t)(\lambda^y(t) + \lambda^p(t) - \lambda^q(t))\,\mathrm{d}\mathbf{x}\mathrm{d}t \\ &\qquad\qquad + \int_\Omega \delta y(0)\lambda^y(0)\,\mathrm{d}\mathbf{x} = 0,\end{aligned}$$

$$\begin{aligned}&\int_0^T \int_\Omega c_2(y(t);\mu)\delta p_{\mathbf{x}}(t)\lambda^p_{\mathbf{x}}(t)\,\mathrm{d}\mathbf{x}\mathrm{d}t \\ &\qquad + \int_0^T \int_\Omega \mathcal{N}_p(\,\cdot\,, y(t), p(t), q(t); \mu)\delta p(t)(\lambda^y(t) + \lambda^p(t) - \lambda^q(t))\,\mathrm{d}\mathbf{x}\mathrm{d}t = 0,\end{aligned}$$

$$\begin{aligned}&\int_0^T (q(t,b) - q^d(t))\delta q(t)\mathrm{d}t + \int_0^T \int_\Omega c_3\delta q_{\mathbf{x}}(t)\lambda^q_{\mathbf{x}}(t)\,\mathrm{d}\mathbf{x}\mathrm{d}t + \int_0^T \int_\Gamma \frac{\partial \delta q}{\partial n}\lambda^q(t)\,\mathrm{d}S\mathrm{d}t \\ &\qquad + \int_0^T \int_\Omega \mathcal{N}_q(\,\cdot\,, y(t), p(t), q(t); \mu)\delta q(t)(\lambda^y(t) + \lambda^p(t) - \lambda^q(t))\,\mathrm{d}\mathbf{x}\mathrm{d}t = 0,\end{aligned}$$

for all $\delta z(t) = (\delta y(t), \delta p(t), \delta q(t)) \in Z$. Hence, by applying integration by parts on $t$ and $\mathbf{x}$ the adjoint system can be written in the strong form as

$$
\begin{aligned}
&-\lambda_t^y(t,\mathbf{x}) - \big(c_1(\mathbf{x})\lambda_{\mathbf{x}}(t,\mathbf{x})\big)_{\mathbf{x}} + c_y(y(t,\mathbf{x});\mu)p_{\mathbf{x}}(t,\mathbf{x})\lambda_{\mathbf{x}}^p(t,\mathbf{x}) \\
&\qquad + \mathcal{N}_y(\mathbf{x}, y(t,\mathbf{x}), p(t,\mathbf{x}), q(t,\mathbf{x});\mu)(\lambda^y(t,\mathbf{x}) + \lambda^p(t,x) - \lambda^q(t,\mathbf{x})) = 0, \quad (3.24a) \\
&-\big(c_2(y(t,\mathbf{x});\mu)\lambda_{\mathbf{x}}^p(t,\mathbf{x})\big)_{\mathbf{x}} \\
&\qquad + \mathcal{N}_p(\mathbf{x}, y(t,\mathbf{x}), p(t,\mathbf{x}), q(t,\mathbf{x});\mu)(\lambda^y(t,\mathbf{x}) + \lambda^p(t,\mathbf{x}) - \lambda^q(t,\mathbf{x})) = 0, \quad (3.24b) \\
&-\big(c_3(\mathbf{x})\lambda_{\mathbf{x}}^q(t,\mathbf{x})\big)_{\mathbf{x}} \\
&\qquad + \mathcal{N}_q(\mathbf{x}, y(t,\mathbf{x}), p(t,\mathbf{x}), q(t,\mathbf{x});\mu)(\lambda^y(t,\mathbf{x}) + \lambda^p(t,\mathbf{x}) - \lambda^q(t,\mathbf{x})) = 0 \quad (3.24c)
\end{aligned}
$$

together with the boundary conditions

$$\lambda_{\mathbf{x}}^y(t,a) = \lambda_{\mathbf{x}}^y(t,b) = \lambda_{\mathbf{x}}^p(t,a) = \lambda_{\mathbf{x}}^p(t,b) = 0 \quad (3.24d)$$

and

$$\lambda^q(t,a) = 0 \quad \text{and} \quad \lambda_{\mathbf{x}}^q(t,b) = q^d(t) - q(t,b) \quad (3.24e)$$

and the terminal condition

$$\lambda(T,\mathbf{x}) = 0. \quad (3.24f)$$

From the variational inequality we get

$$
\begin{aligned}
\Bigg( \int_0^T \int_\Omega \mathcal{N}_\mu(\,\cdot\,, y(t), p(t), q(t);\mu)(\lambda^y(t) + \lambda^p(t) - \lambda^q(t))\,\mathrm{d}\mathbf{x}\mathrm{d}t \\
+ \int_0^T \int_\Omega \big(c_2\big)_\mu(y(t);\mu)p_{\mathbf{x}}(t)\lambda_{\mathbf{x}}^p(t)\,\mathrm{d}\mathbf{x}\mathrm{d}t \Bigg)(\mu - \mu^*) \geq 0
\end{aligned}
$$

with

$$\mathcal{N}_\mu(\mathbf{x}, z;\mu) = \begin{pmatrix} \mathcal{N}_{\mu_1}(\mathbf{x}, z;\mu) \\ \mathcal{N}_{\mu_2}(\mathbf{x}, z;\mu) \\ \mathcal{N}_{\mu_3}(\mathbf{x}, z;\mu) \\ 0 \end{pmatrix} \quad \text{and} \quad \big(c_2\big)_\mu(y;\mu) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \big(c_2\big)_{\mu_4}(y;\mu) \end{pmatrix}.$$

Next let us state the sensitivity equations. We will utilize them later in the subset selection and also in order to compute the Hessian and Hessian approximation in our numerical experiments. Following the strategy in Section 3.2 we get

$$
\begin{aligned}
&\big(s_i^y\big)_t(t) - \big(c_1\big(s_i^y\big)_{\mathbf{x}}(t)\big)_{\mathbf{x}} \\
&\qquad + \mathcal{N}_y(\,\cdot\,, y(t), p(t), q(t);\mu)\big(s_i^y(t) + s_i^p(t) + s_i^q(t)\big) = \mathcal{G}_i^y(t), \quad (3.25a) \\
&-\big(c_2(y(t);\mu)\big(s_i^p\big)_{\mathbf{x}}(t)\big)_{\mathbf{x}} + c_y(y(t);\mu)p_{\mathbf{x}}(t)\big(s_i^y\big)_{\mathbf{x}}(t) \\
&\qquad + \mathcal{N}_p(\,\cdot\,, y(t), p(t), q(t);\mu)\big(s_i^y(t) + s_i^p(t) + s_i^q(t)\big) = \mathcal{G}_i^p(t), \quad (3.25b) \\
&-\big(c_3\big(s_i^q\big)_{\mathbf{x}}(t)\big)_{\mathbf{x}} - \mathcal{N}_q(\,\cdot\,, y(t), p(t), q(t);\mu)\big((s_i^y(t) + s_i^p(t) + s_i^q(t)\big) = \mathcal{G}_i^q(t) \quad (3.25c)
\end{aligned}
$$

with

$$\begin{aligned}
&\mathcal{G}_i^y(t) = -\mathcal{N}_{\mu_i}(\,\cdot\,, y(t), p(t), q(t); \mu) \text{ for } i = 1,2,3, \quad \mathcal{G}_4^y(t) = 0 \\
&\mathcal{G}_i^p(t,\mathbf{x}) = \mathcal{G}_y^i(t) \text{ for } i = 1,2,3, \quad \mathcal{G}_4^p(t) = -\big((c_2(y(t);\mu))_{\mu_4} y_{\mathbf{x}}(t)\big)_{\mathbf{x}} \\
&\mathcal{G}_i^q(t) = -\mathcal{G}_i^y(t) \text{ for } i = 1,\dots,4
\end{aligned} \tag{3.25d}$$

together with the boundary conditions

$$\big(s_i^y\big)_{\mathbf{x}}(t,a) = \big(s_i^y\big)_{\mathbf{x}}(t,b) = \big(s_i^p\big)_{\mathbf{x}}(t,a) = \big(s_i^p\big)_{\mathbf{x}}(t,b) = 0 \quad \text{f.a.a. } t \in (0,T) \tag{3.25e}$$

and

$$s_i^q(t,a) = 0 \quad \text{and} \quad \big(s_i^q\big)_{\mathbf{x}}(t,b) = 0 \quad \text{f.a.a. } t \in (0,T) \tag{3.25f}$$

and the initial conditions

$$s_i^y(0,\mathbf{x}) = 0 \quad \text{f.a.a. } \mathbf{x} \in \Omega \tag{3.25g}$$

for $i = 1,\dots,4$. Hence, we can write the two representations for the gradient as follows

$$\begin{aligned}
\hat{J}'_{adj}(\mu) = \int_0^T \int_\Omega \mathcal{N}_\mu(\,\cdot\,, y(t), p(t), q(t); \mu)(\lambda^y(t) + \lambda^p(t) - \lambda^q(t))\,\mathrm{d}\mathbf{x}\mathrm{d}t \\
+ \int_0^T \int_\Omega \big(c_2\big)_\mu(y(t);\mu) p_{\mathbf{x}}(t) \lambda^p_{\mathbf{x}}(t)\,\mathrm{d}\mathbf{x}\mathrm{d}t
\end{aligned} \tag{3.26}$$

and

$$\big(\hat{J}'_{sens}(\mu)\big)_i = \int_0^T \big(q(t,b) - q^d(t)\big) s_i^q(t,b)\,\mathrm{d}t \quad \text{for} \quad i = 1,\dots,4, \tag{3.27}$$

where $adj$ and $sens$ indicate the strategy for the derivation. To obtain the Hessian $\hat{J}''(\mu) \in \mathbb{R}^{4\times 4}$ we combine the two approaches as described in Section 3.4.2. The variant involving the second order sensitivities will not be investigated here since it is numerically not feasible.

Recalling the results of Section 3.4.2 the Hessian is given by (3.16). To be able to apply this result directly to the presented problem we recall that $z = (y,p,q)$, $\mu = (\mu_1,\mu_2,\mu_3,\mu_4)$ and $e = (e^y, e^p, e^q)^\top$. The first and second order derivatives of $J$ and $e$ are given in Appendix A.3. We define the matrices

$$\mathcal{L}_{zz} = \begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yp} & \mathcal{L}_{yq} \\ \mathcal{L}_{py} & \mathcal{L}_{pp} & \mathcal{L}_{pq} \\ \mathcal{L}_{qy} & \mathcal{L}_{qp} & \mathcal{L}_{qq} \end{pmatrix}, \quad \mathcal{L}_{z\mu} = \begin{pmatrix} \mathcal{L}_{y\mu_1} & \mathcal{L}_{y\mu_2} & \mathcal{L}_{y\mu_3} & \mathcal{L}_{y\mu_4} \\ \mathcal{L}_{p\mu_1} & \mathcal{L}_{p\mu_2} & \mathcal{L}_{p\mu_3} & \mathcal{L}_{p\mu_4} \\ \mathcal{L}_{q\mu_1} & \mathcal{L}_{q\mu_2} & \mathcal{L}_{q\mu_3} & \mathcal{L}_{q\mu_4} \end{pmatrix},$$

and

$$\mathcal{L}_{\mu\mu} = \begin{pmatrix} \mathcal{L}_{\mu_1\mu_1} & \mathcal{L}_{\mu_1\mu_2} & \mathcal{L}_{\mu_1\mu_3} & \mathcal{L}_{\mu_1\mu_4} \\ \mathcal{L}_{\mu_2\mu_1} & \mathcal{L}_{\mu_2\mu_2} & \mathcal{L}_{\mu_2\mu_3} & \mathcal{L}_{\mu_2\mu_4} \\ \mathcal{L}_{\mu_3\mu_1} & \mathcal{L}_{\mu_3\mu_2} & \mathcal{L}_{\mu_3\mu_3} & \mathcal{L}_{\mu_3\mu_4} \\ \mathcal{L}_{\mu_4\mu_1} & \mathcal{L}_{\mu_4\mu_2} & \mathcal{L}_{\mu_4\mu_3} & \mathcal{L}_{\mu_4\mu_4} \end{pmatrix}.$$

Note that the derivatives of $\mathcal{L}$ are given by the derivatives of $J$ and $e$. For example let us state the case of the mixed derivatives:

$$\begin{aligned}\mathcal{L}_{\wp\mu_i}(z(\mu),\mu,,\lambda(\mu)) =& J_{\wp\mu_i}(z(\mu),\mu) + \langle\lambda, e_{\wp\mu_i}(z(\mu),\mu)\rangle_{\Lambda',\Lambda}\\ =& J_{\wp\mu_i}(z(\mu),\mu) + \langle\lambda^y(\mu), e^y_{\wp\mu_i}(z(\mu),\mu)\rangle_{\Lambda',\Lambda}\\ &+ \langle\lambda^p(\mu), e^p_{\wp\mu_i}(z(\mu),\mu)\rangle_{\Lambda',\Lambda} + \langle\lambda^q(\mu), e^q_{\wp\mu_i}(z(\mu),\mu)\rangle_{\Lambda',\Lambda}\end{aligned}$$

for $i = 1,\dots,4$ and $\wp \in \{y,p,q\}$. Note that inserting this into (3.16) we get

$$\begin{aligned}\big(\big(\hat{J}''(\mu)\big)\big)_{ij} = \int_0^T & s_i^q(t,b)s_j^q(t,b)\,\mathrm{d}t\\ &+ (s_i^y)^*\mathcal{L}_{yy}(z,\mu,\lambda)s_j^y + (s_i^y)^*\mathcal{L}_{yp}(z,\mu,\lambda)s_j^p + (s_i^y)^*\mathcal{L}_{yq}(z,\mu,\lambda)s_j^q\\ &+ (s_i^p)^*\mathcal{L}_{py}(z,\mu,\lambda)s_j^y + (s_i^p)^*\mathcal{L}_{pp}(z,\mu,\lambda)s_j^p + (s_i^p)^*\mathcal{L}_{pq}(z,\mu,\lambda)s_j^q\\ &+ (s_i^q)^*\mathcal{L}_{qy}(z,\mu,\lambda)s_j^y + (s_i^q)^*\mathcal{L}_{qp}(z,\mu,\lambda)s_j^p + (s_i^q)^*\mathcal{L}_{qq}(z,\mu,\lambda)s_j^q\\ &+ (s_i^y)^*\mathcal{L}_{y\mu_i}(z,\mu,\lambda) + (s_i^p)^*\mathcal{L}_{p\mu_i}(z,\mu,\lambda) + (s_i^q)^*\mathcal{L}_{q\mu_i}(z,\mu,\lambda)\\ &+ \mathcal{L}_{\mu_j,y}(z,\mu,\lambda)s_j^y + \mathcal{L}_{\mu_j p}(z,\mu,\lambda)s_j^p + \mathcal{L}_{\mu_j q}(z,\mu,\lambda)s_j^q\\ &+ e_{\mu_i\mu_j}(z(\mu),\mu)\end{aligned}$$

for $1 \le i,j \le 4$, $z = z(\mu)$ and $\lambda = \lambda(\mu)$. As it can be seen the Hessian for the presented problem is very involved. To compute the Hessian one nonlinear system, one adjoint system and four sensitivity systems have to be solved. Recall that the adjoint and sensitivity equations are linear. Moreover, the first part of the Hessian corresponds to the Gauss-Newton or Fisher information matrix.

Before we continue to the numerical realization of the parameter estimation let us recall the assumption of Fréchet differentiability. Let us now have a look at one particular variable and check the Fréchet differentiability. The differentiability results of the variables $y$ and $p$ with respect to $\mu$ are known. Results for this type of equations are available in [48, 81]. The equation for $q$ is not so common. Hence, we investigate the Fréchet differentiability of $p$ with respect to the parameter $\mu$.

**Theorem 3.2.** *Suppose* $(y,p,q)$ *is a solution to* (1.1) *with* (1.2) *and* $\|\frac{\partial}{\partial\mu}c_2(y;\mu)\|_{L^\infty(Q)} < \overline{c}$. *Then* $p$ *is Fréchet differentiable with respect to* $\mu$.

*Sketch of Proof.* The proof follows from [8] with some modifications. We rewrite the problem in the form

$$-\nabla\cdot(c(\mu)\nabla p) = f(\mu), \quad \frac{\partial p}{\partial n} = 0 \tag{3.28}$$

with $f \in L^2(\Omega)$, $f = \mathcal{N}(\mathbf{x},\bar{y},p(\mu),q(\mu);\mu)$ and $c(\mu) = c_2(\bar{y};\mu)$ for a fixed $\bar{y}$. Note that the obtained nonlinear elliptic system has a unique solution [75, 86], which is also required in the proof of Theorem 1.4. By writing the nonlinear diffusion coefficient in the form

$$c(\mu+\delta\mu) = c(\mu) + c'(\bar{\mu})\delta\mu$$

for a $\bar{\mu} \in (\mu, \mu + \delta\mu)$ and inserting into (3.28) we can directly apply the results presented in [8]. In the proof the Poincaré's inequality is utilized. This is not possible in the presented setting since we have all Neumann boundaries. To overcome this we can apply [75, Lemma 3.14]. For this the requirement $f \in L^2(\Omega)$ is essential. This concludes the proof. □

### 3.6.2 A-posteriori error estimator

In the numerical realization of the parameter estimation we want to utilize the numerical techniques introduced in Chapter 1 and 2. By discretizing the nonlinear least squares problem (3.21) using the finite element method we obtain $\hat{J}^N$ and the corresponding optimal solution $\mu^{N,\star}$. Analogously, when using the reduced order model obtained by the POD Galerkin scheme we get $\hat{J}^\ell$ and $\mu^{\ell,\star}$. In the optimization process the goal is to replace the finite element model by the reduced order model. The obtained solution is suboptimal, i.e. $\mu^{\ell,\star}$ is not an optimal solution to the reference finite element problem $\hat{J}^N$. The aim is to find bounds for the error $\|\mu^{N,\star} - \mu^{\ell,\star}\|_2$. Thereby the local optimal solution $\mu^{\ell,\star}$ obtained by the reduced order model can be certified. For this we reformulate the results from [29, 53] for the settings introduced in this work.

The result – obtained in [53] by a second-order analysis – are in a more general setting including bound constraints for $\mu$ which are omitted in the numerical part of this work. Let us restate the results for our specific settings [53, Theorem 3.4].

**Theorem 3.3.** *Let $\mu^{N,\star} \in \mathcal{M}_{ad}$ be an inactive, local minimizer of* (3.21) *utilizing the finite element discretization, i.e. $\nabla \hat{J}^N(\mu^{N,\star}) \equiv 0$ holds true. Further, there exists a $\delta > 0$ such that the second-order sufficient optimality condition*

$$v^\top \nabla^2 \hat{J}^N(\mu^{N,\star}) v \geq \delta \|v\|_2^2 \quad \forall v \in \mathbb{R}^4$$

*is satisfied. If $\mu^{\ell,\star} \in \mathcal{M}_{ad}$ is given such that $\|\mu^{N,\star} - \mu^{\ell,\star}\|_2$ is sufficiently small. Then the error estimator*

$$\|\mu^{N,\star} - \mu^{\ell,\star}\|_2 \leq \frac{2}{\delta} \|\nabla \hat{J}^N(\mu^{\ell,\star})\|_2$$

*holds.*

This gives an upper bound for the error when utilizing the reduced order model with respect to the solution obtained when using the finite element model. The open question with this estimator is the value for $\delta$. In [53] the constant $\delta$ was chosen as the smallest eigenvalue of the Hessian $\nabla^2 \hat{J}^N(\mu^{\ell,\star})$ which turned out to work very well. The error estimator then reads as

$$\|\mu^{N,\star} - \mu^{\ell,\star}\|_2 \leq 2 \|\nabla^2 \hat{J}^N(\mu^{\ell,\star})^{-1}\|_2 \|\nabla \hat{J}^N(\mu^{\ell,\star})\|_2. \tag{3.29}$$

Note that in order to compute the error estimator the Hessian is needed. This can become computationally expensive since it involves finite element solves. In the context of reduced basis a similar result, which is based on the inverse and implicit function theorem [14], is obtained. Similar results can be obtained from the Newton-Kantorovich theory. For our setting the result reads as follows [29, Proposition 4]:

**Proposition 3.4.** *Let $\mu^{\ell,\star} \in \mathcal{M}_{ad}$ be the optimal solution found by solving the least squares problem* (3.21) *utilizing the reduced order model* (2.12) *satisfying the stopping criterion $\|\nabla \hat{J}^\ell(\mu^{\ell,\star})\|_2 \leq \varepsilon_{opt}$. Further, let us introduce*

$\bar{B}(\mu^{\ell,\star}, \alpha)$ *a closed ball around $\mu^{\ell,\star}$ with radius $\alpha$,*

*the Hessian $\nabla^2 \hat{J}^N(\mu^{\ell,\star})$ is regular,*

$\gamma := \|\nabla^2 \hat{J}^N(\mu^{\ell,\star})^{-1}\|_2$,

$\varepsilon := \Delta_{\nabla \hat{J}(\mu^{\ell,\star})} + \varepsilon_{opt}$ *with* $\Delta_{\nabla \hat{J}(\mu)} = \nabla \hat{J}^N(\mu) - \nabla \hat{J}^\ell(\mu)$,

$L(\alpha) := \sup_{\mu \in \bar{B}(\mu^{\ell,\star},\alpha)} \|\nabla^2 \hat{J}^N(\mu^{\ell,\star}) - \nabla^2 \hat{J}^N(\mu)\|_2$.

*If the condition*

$$2\gamma L(2\gamma\epsilon) \leq 1$$

*holds, then there exists a unique solution $\mu^{N,\star}$ to the optimization problem* (3.21) *utilizing the finite element model* (1.8) *with $\mu^{N,\star} \in \bar{B}(\mu_\ell^\star, 2\gamma\varepsilon)$ and the rigorous error bound*

$$\|\mu^{N,\star} - \mu^{\ell,\star}\|_2 \leq 2\|\nabla^2 \hat{J}^N(\mu^{\ell,\star})^{-1}\|_2 (\Delta_{\nabla \hat{J}(\mu^{\ell,\star})} + \varepsilon_{opt}) \tag{3.30}$$

*holds.*

When comparing the results it can be seen that they are very similar although the derivations are very different. Furthermore, this also verifies the choice made in (3.29) by setting $\delta = \|\nabla^2 \hat{J}^N(\mu^{\ell,\star})^{-1}\|_2$.

### 3.6.3 Numerical strategy

As a numerical method to solve the introduced parameter estimation problem (3.21) we apply the Gauss-Newton method [55, 68, 82]. In the Gauss-Newton method the Hessian matrix $\nabla^2 \hat{J}(\mu)$ is approximated by the matrix $H(\mu) \in \mathbb{R}^{4\times 4}$ given by (3.17). Note that for the evaluation of the approximated Hessian only the first order sensitivities are needed which are used to evaluate the gradient $\nabla \hat{J}(\mu)$. Further, recall that $H$ corresponds to the Fisher information matrix which is utilized in the subset selection method. By applying the subset selection method it is ensured that the inverse of $H$ exists which is essential in computing the search direction in the proposed optimization strategy. In every iteration $k$ the Gauss-Newton search direction $d_{\mathrm{GN}}^{(k)}$ is computed as the solution of the linear system

$$H(\mu^{(k)}) d_{\mathrm{GN}}^{(k)} = -\nabla \hat{J}(\mu^{(k)}).$$

In order to speed up the optimization process we will replace the finite element model (1.8) by the reduced order model (2.12). This will allow us to solve the parameter estimation problem at less computational costs.

When looking at the construction of the reduced order model it can be seen that the POD basis is computed with respect to a given parameter $\mu$. This implies that when varying the parameter $\mu$ we do not have a guarantee that the reduced order model is of good quality.

Hence, it can be necessary to recompute the POD basis. For this we need a measure for the error introduced by the reduced order model. In this work we utilize the residual as an error indicator. In particular we look at the error obtained by inserting the reduced order solution into the finite element model. The norm of the residuals is then given by (2.28). In order not to distinguish between the three variables, since the finite element solution can only be obtained simultaneously, we define

$$\rho(\Psi, \mu) = \varepsilon^y_{\text{res}} + \varepsilon^p_{\text{res}} + \varepsilon^q_{\text{res}}. \tag{3.31}$$

This type of error indicator is motivated by error estimators which are used in the reduced basis context [39, 44, 70]. The indicator measures the error of the given reduced order solution with respect to the finite element discretization for a given parameter $\mu$. This error indicator is used to ensure that the reduced order model is sufficiently accurate during the optimization process. In Section 2.5 this type of error indicator was investigated numerically and it showed to be always very close to the other error measures.

Let us next describe how the different modules are used in the optimization process [61]. For a given initial value $\tilde{\mu}$ the finite element model (1.8) is solved and a POD basis is computed for each variable using (2.16). By utilizing the obtained POD basis the reduced order model (2.12) is generated. Additionally, reduced order models are generated for the sensitivity equations. The subset selection is performed to determine the parameters that are identifiable. Non identifiable parameters are set to some reasonable or academic value and hence the parameter vector $\mu^{(0)}$ is obtained. Starting with this parameter the optimization strategy is started. During the optimization process the reduced order model is utilized to solve the nonlinear system (1.1). After each solution process of (2.12) the error indicator (3.31) is evaluated in order to quantify the accuracy of the reduced order model. If the error indicator is too large (i.e. $\rho(\Psi, \mu) > \varepsilon_{res}$), the finite element model (1.8) is solved, new POD bases are determined and the associated reduced order models are generated. An Armijo backtracking strategy is applied for the step size control during the optimization process. The optimization method is stopped when a predefined tolerance $\varepsilon_{opt}$ is reached (i.e. $\|\nabla \hat{J}(\mu)\| < \varepsilon_{opt}$). Lastly, the error estimator (3.29) or (3.30) can be computed to verify that the obtained optimal solution is of good quality. The described strategy is summarized in Algorithm 7 in detail. A similar approach is investigated independently in [87].

The introduced algorithm ensured that the solution to the finite element model and the reduced order model do not deviate too much from each other. Possible extension to the described strategy are to introduce additional POD bases for the sensitivity equations. In the presented examples this is not required in order to obtain accurate results. Note that in the presented algorithm the sensitivity equations are always solved through a reduced order model. The POD basis computation can be enhanced by including strategies involving the sensitivity information [47, 52]. This could lead to a better basis for the optimization strategy. Furthermore, the subset selection method could be applied in every iteration of the optimization problem. This particular scenario is not investigated in this work. The focus here is to apply the subset selection method once to obtain a reasonable identifiable system.

**Algorithm 7** (Adaptive optimization strategy)

**Require:** Initial guess $\tilde{\mu}$, residual tolerance $\varepsilon_{res}$, stopping criteria $\varepsilon_{opt}$, maximum number of iteration $k_{\max}$ and $j_{\max}$

$[Y^N, P^N, Q^N] \leftarrow$Solve FE model (1.8) for $\tilde{\mu}$
$\Psi = [\Psi^y, \Psi^p, \Psi^q] \leftarrow$Compute POD basis by solving (2.16) for $Y^N$, $P^N$, and $Q^N$
Solve the sensitivity equation (3.25) using POD for $\tilde{\mu}$ and $\Psi$
$\mu^{(0)} \leftarrow$Perform the subset selection for $\tilde{\mu}$
$[Y^\ell, P^\ell, Q^\ell] \leftarrow$Solve reduced order model (2.12) for $\mu^{(0)}$ and $\Psi$
$\rho(\Psi, \mu^{(0)}) \leftarrow$ Evaluate error indicator (3.31) for $\Psi$ and $\mu^{(0)}$
**if** $\rho(\Psi; \mu^{(0)}) > \varepsilon_{res}$ **then**
  $[Y^N, P^N, Q^N] \leftarrow$Solve FE model (1.8) for $\mu^{(0)}$
  $\Psi = [\Psi^y, \Psi^p, \Psi^q] \leftarrow$Compute POD basis by (2.16) for $Y^N$, $P^N$, and $Q^N$
**end if**
Solve the sensitivity equation (3.25) using POD for $\mu^{(0)}$ and $\Psi$
Set $k \leftarrow 0$ and evaluate cost $\hat{J}^{(k)} \leftarrow \hat{J}(\mu^{(k)})$ and $\nabla \hat{J}^{(k)} \leftarrow \nabla \hat{J}(\mu^{(k)})$
**while** $\nabla \hat{J}^{(k)} > \varepsilon_{opt}$ and $k \leq k_{\max}$ **do**
  Evaluate Gauss-Newton matrix $H^{(k)}$
  Solve search direction $d_{\mathrm{GN}}^{(k)}$ given by $H^{(k)} d_{\mathrm{GN}}^{(k)} = -\nabla \hat{J}^{(k)}$
  Set $\alpha \leftarrow 1$ and $\bar{\mu} \leftarrow \mu^{(k)} + \alpha d_{\mathrm{GN}}^{(k)}$
  $[Y^\ell, P^\ell, Q^\ell] \leftarrow$Solve reduced order model (2.12) for $\bar{\mu}$ and $\Psi$
  $\rho(\Psi; \bar{\mu}) \leftarrow$ Evaluate error indicator (3.31) for $\Psi$ and $\bar{\mu}$
  **if** $\rho(\Psi, \bar{\mu}) > \varepsilon_{res}$ **then**
    $[Y^N, P^N, Q^N] \leftarrow$Solve FE model (1.8) for $\bar{\mu}$
    $\Psi = [\Psi^y, \Psi^p, \Psi^q] \leftarrow$Compute POD basis by (2.16) for $Y^N$, $P^N$, and $Q^N$
  **end if**
  Solve the sensitivity equation (3.25) using POD for $\bar{\mu}$ and $\Psi$
  Evaluate cost $\bar{J} \leftarrow \hat{J}(\bar{\mu})$ and $\nabla \bar{J} \leftarrow \nabla \hat{J}(\bar{\mu})$
  **while** $\bar{J} \geq \hat{J}^{(k)} + \sigma\alpha\langle \nabla \hat{J}^{(k)}, d_{\mathrm{GN}}^{(k)} \rangle_2$ and $j \leq j_{\max}$ **do**
    Set $\alpha \leftarrow \frac{1}{2}\alpha$ and $\bar{\mu} \leftarrow \mu^k + \alpha d_{\mathrm{GN}}^{(k)}$
    Solve reduced order model (2.12) for $\bar{\mu}$ and $\Psi$
    $\rho(\Psi, \bar{\mu}) \leftarrow$ Evaluate error indicator (3.31) for $\Psi$ and $\bar{\mu}$
    **if** $\rho(\Psi, \bar{\mu}) > \varepsilon_{res}$ **then**
      $[Y^N, P^N, Q^N] \leftarrow$Solve FE model (1.8) for $\bar{\mu}$
      $\Psi = [\Psi^y, \Psi^p, \Psi^q] \leftarrow$Compute POD basis by (2.16) for $Y^N$, $P^N$, and $Q^N$
    **end if**
    Solve the sensitivity equation (3.25) using POD for $\bar{\mu}$ and $\Psi$
    Evaluate cost $\bar{J} \leftarrow \hat{J}(\bar{\mu})$ and $\nabla \bar{J} \leftarrow \nabla \hat{J}(\bar{\mu})$
    Set $j \leftarrow j + 1$
  **end while**
  Set $\mu^{(k+1)} \leftarrow \bar{\mu}$, $\hat{J}^{(k+1)} \leftarrow \bar{J}$ and $\nabla \hat{J}^{(k+1)} \leftarrow \nabla \bar{J}$
  Set $k \leftarrow k + 1$;
**end while**
**return** Optimal solution $\mu^\star \leftarrow \mu^{(k)}$ ($\varepsilon \leftarrow$Error estimator (3.29) or (3.30) (Optional))

The Gauss-Newton method can be replaced by any other optimization strategy. Especially, in order to incorporate inequality constraints on the parameter $\mu$ a SQP method can be utilized, where the search direction $d_{\mathrm{SQP}}$ is solved by a quadratic programming problem in every iteration. The process has been outlined in Section 3.5 where the Newton and SQP method were introduced. In the numerical experiments these two methods did not provide any additional advantage and hence are not considered in the presented results.

## 3.7 Numerical results

To show the efficiency of the described algorithm we will present some numerical examples. We will consider three scenarios. First we will have a look at the case that the initial guess is close to the optimal solution. In the second case we choose an initial guess that is further away from the optimal solution. In both scenarios the obtained solution of Algorithm 7 is compared to a strategy using only the finite element model. This strategy is obtained by replacing the reduced order solvers by finite element solvers in Algorithm 7. Additionally, after the optimization the error estimator is evaluated. Lastly, we will present numerical results where noise is added to the data/target $q^d(t)$ to show the robustness of the proposed method.

Let us introduce the settings used in the numerical experiments. The missing quantities for the nonlinear system are given by (1.39) together with the nonlinearity (1.2), introduced in Chapter 1. For the finite element solver we choose the settings introduced in Section 1.3.3. Note that we set $T = 1$ and $N_t = 100$ in the presented results. For the reduced order model the number of basis functions for the three variables are set to $\ell_y = 19$, $\ell_p = 19$ and $\ell_q = 17$. These numbers are obtained by solving (2.16) under consideration of the accuracy of the eigenvalue solver. The nonlinear terms are evaluated with EIM as shown in Section 2.5. For the nonlinearity $\mathcal{N}$ we utilize $22$ basis functions and for $c_3$ we use $23$. This corresponds to a maximum interpolation error of $10^{-11}$ during the construction.

Next let us introduce the settings for the optimization procedure. As the tolerance for the stopping criteria we choose $\varepsilon_{opt} = 10^{-6}$ and set the maximum number of iterations to $k_{\max} = 100$. Further, we set the tolerance for the error indicator to $\varepsilon_{res} = 10^{-4}$. The settings for the Armijo condition are $\sigma = 0.01$ and $j_{\max} = 10$. The target $q^d(t)$ is generated by solving the nonlinear system (1.1) for $\mu^\star = (1.1, -0.7, -0.1, 0.4)$ which then should be recovered by the parameter estimation.

In the subset selection procedure (Algorithm 6) parameters associated to eigenvalues larger than $\varepsilon_{ss} = 10^{-6}$ are selected for the optimization. This is a reasonable choice and coincides with the accuracy of the finite element discretization and eigenvalue solver. The parameters excluded from the optimization are set to the corresponding exact values of $\mu^\star$. This is done in order to show the performance of the optimization algorithm and omit problems arising from the ill posedness of the problem.

All tests are performed on the same computer and software. To compare the performance we state the CPU times for the computation and the number of required finite element solves for the nonlinear system and for the sensitivity equations in parentheses.

### 3.7.1 Run 1 (Parameter identification)

For the first numerical experiment we consider the case that the deviation of the initial parameter $\tilde{\mu}$ is in the neighborhood of the exact solution $\mu^\star$. For this we choose

$$\tilde{\mu} = (1.3 \cdot \mu_1^\star, 1.5 \cdot \mu_2^\star, 1.5 \cdot \mu_3^\star, 1.5 \cdot \mu_4^\star) = (1.43, -1.05, -0.15, 0.60).$$

This corresponds to a deviation of $30\%$ in the first parameter and $50\%$ in the other parameters. By applying the subset selection the parameter $\mu_2$ is marked as non identifiable and therefore set to $\mu_2^{(0)} = -0.7$. This is also indicated in the numerical results in Table 3.1 by a dagger. The obtained results of the optimization are presented in Table 3.1. Further, the computational expenses for the two approaches are outlined in the last two columns in form of CPU time and finite element solves. In Figure 3.3 (left plot) the target function $q^d(t)$ together with the output corresponding to the initial guess $\mu^{(0)}$, i.e. after the subset selection, are shown. To compare the output corresponding to the optimal solution obtained by the two methods the absolute difference is plotted in Figure 3.3 (right plot). Lastly, in Table 3.2 the error estimator is presented.

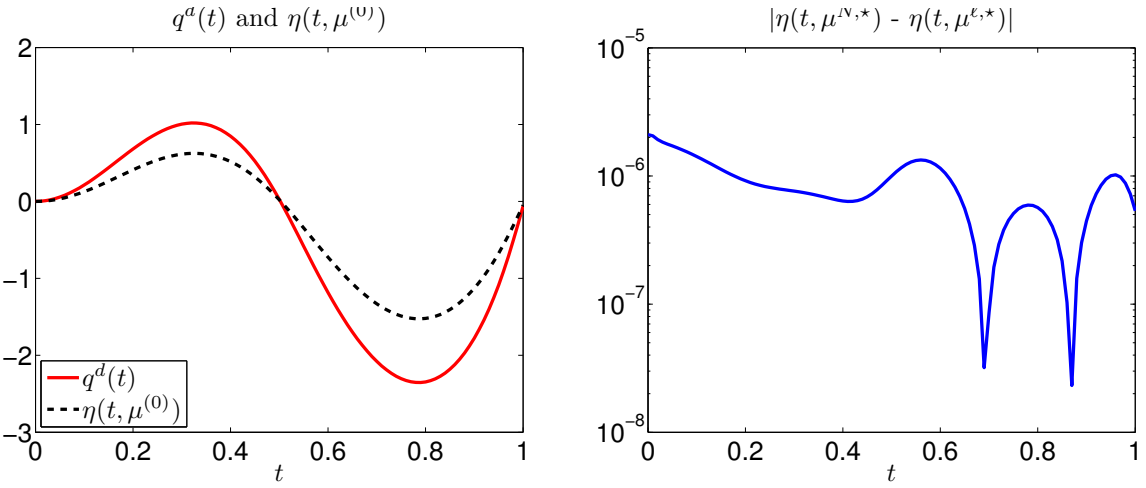


Figure 3.3: Run 1: Target function $q^d$ and output function $\eta(t,\mu^{(0)})$ (right plot) and difference between the output $\eta(t,\mu)$ for the optimal solutions $\mu^\star$ obtained by using finite elements and adaptive reduced order models (right plot).

When looking at the results it can be seen that the optimization using adaptive reduced order models is very efficient. Only one finite element solution is required. This is due to the fact that we started close to the exact solution $\mu^\star$. The optimization strategy terminates after five iterations. Hence, a speedup of more than seven is a very good result. Further, let us note that computing a single finite element solution takes approximately $27$ seconds. This implies that the optimization using the reduced order model takes nine seconds which corresponds to a speedup of factor $26$. The accuracy of the optimization using the reduced order model is very good when looking at the right plot presented in Figure 3.3. The obtained optimal parameter $\mu^{N,\star}$ and $\mu^{\ell,\star}$ by both strategies deliver almost the same values.

Comparing it to the exact $\mu^\star$ it can be seen that the optimization strategies could recover the parameters almost exact in both cases.

| Method | $\mu^{N,\star}$ and $\mu^{\ell,\star}$ | FE solves | CPU time |
|---|---|---|---|
| FE | $(1.100000, -0.700000^\dagger, -0.100000, 0.400000)$ | $8(21)$ | 260 |
| ROM | $(1.100003, -0.700000^\dagger, -0.100000, 0.399995)$ | $1(0)$ | 36 |

Table 3.1: Run 1: Optimal solution obtained by the finite element (FE) approach and the adaptive algorithm using reduced order models (ROM). The † indicates the parameter fixed by the subset selection. In the last two columns the total expenses of the two approaches is compared.

Lastly, we look at the error estimator. We want to avoid computing the optimization using only finite element solvers. Further, the exact solution $\mu^\star$ is in general not known. Hence, we need to estimate the error in the parameter $\mu^{\ell,\star}$ obtained by the adaptive strategy. For this we evaluate the error estimator introduced in Section 3.6.2. The evaluation of this estimator is expensive. The computational time is approximately $39$ seconds and involves the solution of the nonlinear system, the sensitivity equations and the adjoint equation using the finite element solvers. The obtained solution by the error estimator is very satisfying since it indicates that the error is below $1\%$. Note that the estimator is an upper bound and the actual error is lower. The efficiency of the error estimator is comparable to the results presented in [29, 53]. The computational cost for the estimator can be reduced by appropriate reduced order models and efficient error estimators [28].

| $\|\mu^{N,\star} - \mu^{\ell,\star}\|_2$ | (3.29) | (3.30) |
|---|---|---|
| $0.000005$ | $0.005945$ | $0.008495$ |

Table 3.2: Run 1: Error estimator for the obtained solution $\mu^{\ell,\star}$ by the adaptive reduced order model optimization.

### 3.7.2 Run 2 (Adaptive POD)

In the second numerical experiment we will use an initial guess that is further from the optimal solution. For this we set

$$\tilde{\mu} = (1.3 \cdot \mu_1^\star, 2 \cdot \mu_2^\star, 2 \cdot \mu_3^\star, 2 \cdot \mu_4^\star) = (1.43, -1.40, -0.20, 0.80).$$

Note that the first parameter $\tilde{\mu}_1$ is again deviated by $30\%$. The reason for this is that we need to stay in the admissible set $\mathcal{M}_{ad}$ in order to satisfy the conditions for Theorem 1.4. The other parameters are deviated by $100\%$. The subset selection method again selects the

parameter $\mu_2$ which is then set to the exact value $-0.7$ in the initial value $\mu^{(0)}$. Compared to the previous experiment we expect that now the adaptivity of the algorithm will be active.

The results are outlined in Table 3.3. It can be seen that the adaptive strategy is active since now the nonlinear system has to be solved two times using finite elements. This leads to the fact that now the adaptive strategy became more expensive compared to the first scenario. The POD bases are recomputed in the second iteration. Still a speed up factor of approximately five is obtained. The optimization procedure again requires five iteration. This time the Armijo condition is active more often, hence the higher computational cost in the finite element case. The optimization using the adaptive reduced order models takes approximately ten seconds when excluding the time required by the finite element solves (27 seconds each). This corresponds to a speedup factor of 31. Comparing this to the previous run it can be seen that the cost for the optimization stayed the same only more finite element evaluations are needed to update the POD bases. In the case that the POD bases have to be updated more frequently, the optimization using adaptive reduced order models will have roughly the same computational cost as the optimization using finite elements. When introducing additional POD bases for the sensitivity equations and setting $\varepsilon_{res} = 0$ the adaptive strategy is equivalent to using only finite element solvers.

| Method | $\mu^{N,\star}$ and $\mu^{\ell,\star}$ | FE solves | CPU time |
|---|---|---|---|
| FE | $(1.100000, -0.700000^{\dagger}, -0.100000, 0.400000)$ | $10(21)$ | $315$ |
| ROM | $(1.099997, -0.700000^{\dagger}, -0.100001, 0.399998)$ | $2(0)$ | $64$ |

Table 3.3: Run 2: Optimal solution obtained by the finite element (FE) approach and the adaptive algorithm using reduced order models (ROM). The † indicates the parameter fixed by the subset selection. In the last two columns the total expenses of the two approaches is compared.

Lastly, we again look at the error estimator. As in the previous case the estimators give an upper bound for the error in the optimal parameter $\mu^{\ell,\star}$. The results are again very good and of equal quality and computational expenses.

| $\|\mu^{N,\star} - \mu^{\ell,\star}\|_2$ | (3.29) | (3.30) |
|---|---|---|
| $0.000004$ | $0.006375$ | $0.002471$ |

Table 3.4: Run 2: Error estimator for the obtained solution $\mu^{\ell,\star}$ by the adaptive reduced order model optimization.

### 3.7.3 Run 3 (Noise)

In the last experiment we want to consider the case that there is noise added to the target. We consider the case of $5\%$ noise. The noise is generated using the MATLAB command

`randn`. All other settings are the same as introduced in Run 1. The obtained noise is then added to the target generated by $\mu^\star$. In Figure 3.4 the target with noise and the output function corresponding to the initial value $\mu^{(0)}$ is shown. We will further investigate the influence of $\varepsilon_{ss}$ on the obtained solution. For this we perform two experiments on the identical settings and noise where we set $\varepsilon_{ss}$ to $10^{-6}$ and $10^{-4}$ and compare the obtained results.

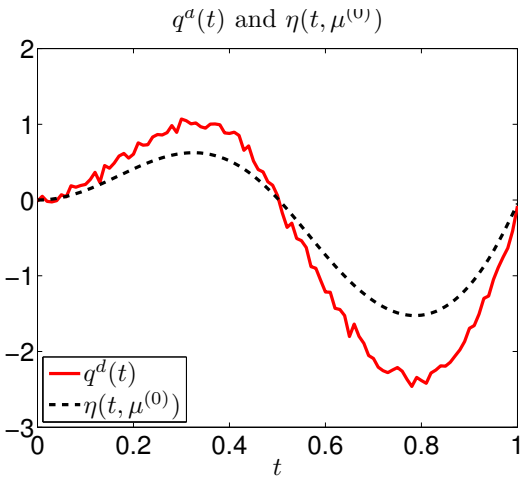


Figure 3.4: Run 3: Target function $q^d$ with noise and output function $\eta(t,\mu^{(0)})$.

By applying the subset selection with $\varepsilon_{ss} = 10^{-6}$ the parameter $\mu_2$ is marked as non identifiable and therefore set to $\mu_2^{(0)} = -0.7$. In the case of $\varepsilon_{ss} = 10^{-4}$ additionally the parameter $\mu_4$ is selected and set to $\mu_4^{(0)} = 0.4$. This is also indicated in the numerical results in Table 3.5 by a dagger. In Table 3.5 the obtained results of the optimization are presented. The computational expenses for the two approaches are given in form of CPU time and finite element solves. In Figures 3.5 and 3.6 the absolute difference between the finite element solution and the reduced order model solution for the output function is shown (left plot). In the right plots the fitting of the output function $\eta$ to the target is shown for the reduced order model output.

| $\varepsilon_{ss}$ | Method | $\mu^{N,\star}$ and $\mu^{\ell,\star}$ | FE solves | CPU time |
|---|---|---|---|---|
| $10^{-6}$ | FE | $(0.903072, -0.7000^\dagger, -0.114817, 0.576732\ )$ | 8(21) | 260 |
| $10^{-6}$ | ROM | $(0.904330, -0.7000^\dagger, -0.114756, 0.574235\ )$ | 1(0) | 36 |
| $10^{-4}$ | FE | $(1.071003, -0.7000^\dagger, -0.105129, 0.400000^\dagger)$ | 8(14) | 220 |
| $10^{-4}$ | ROM | $(1.070810, -0.7000^\dagger, -0.105162, 0.400000^\dagger)$ | 1(0) | 35 |

Table 3.5: Run 3: Optimal solution obtained by the finite element (FE) approach and the adaptive algorithm using reduced order models (ROM). The † indicates the parameter fixed by the subset selection. In the last two columns the total expenses of the two approaches is compared.

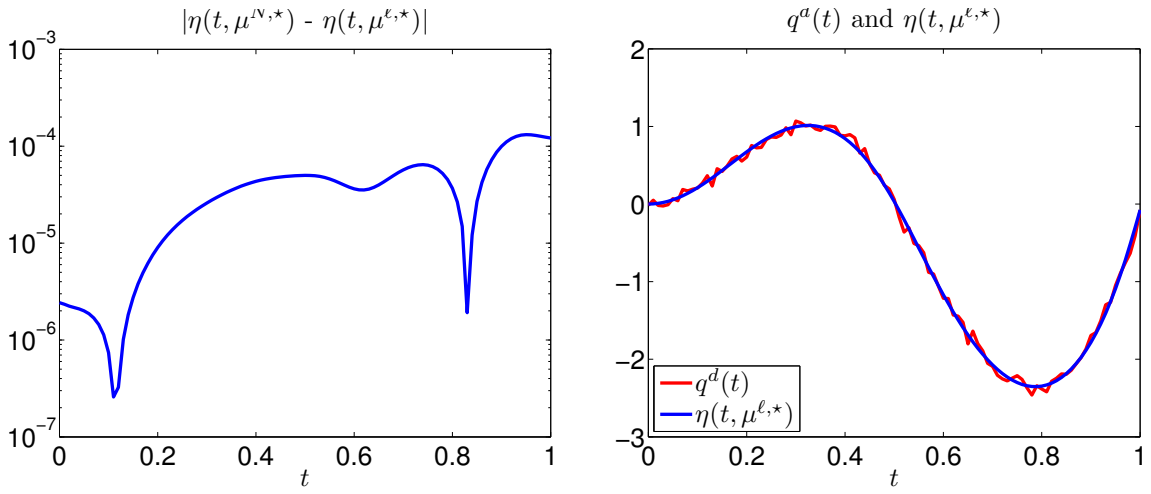


Figure 3.5: Run 3: Difference between the output $\eta(t,\mu)$ for the optimal solutions $\mu^\star$ obtained by using finite elements and adaptive reduced order models (left plot) and comparison of the target with the output function $\eta(t,\mu^{\ell,\star})$ (right plot) for $\varepsilon_{ss} = 10^{-6}$.

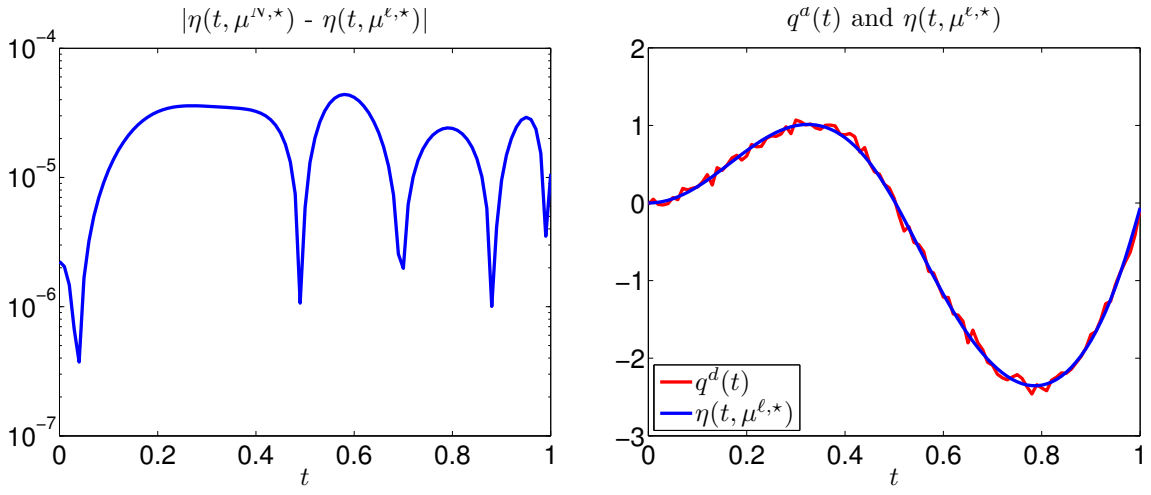


Figure 3.6: Run 3: Difference between the output $\eta(t,\mu)$ for the optimal solutions $\mu^\star$ obtained by using finite elements and adaptive reduced order models (left plot) and comparison of the target with the output function $\eta(t,\mu^{\ell,\star})$ (right plot) for $\varepsilon_{ss} = 10^{-4}$.

Looking at the results we see that for a target with noise it is crucial to carefully choose the parameters for the identification process. Although the eigenvalue corresponding to the parameter $\mu_4$ is of magnitude $10^{-5}$ it cannot be identified well. It even interferes with the computation of the other parameters and results in inaccurate solutions. When $\varepsilon_{ss} = 10^{-4}$ the situation becomes much better, see Table 3.5. The difference is not visible when looking at the right plots in Figures 3.5 and 3.6. In terms of computational performance we obtain the same results as in Run 1. The adaptive strategy is not active since the initial guess was

chosen close enough to the exact solution. In the case that the initial guess is chosen further from the exact solution the same results can be observed combined with the performance results from Run 2.

Lastly, we investigate the error estimator. As in the previous two cases the estimators give an upper bound for the error in the optimal parameter $\mu^{\ell,\star}$. The results for the two settings are shown in Table 3.6. For both settings of $\varepsilon_{ss}$ the error estimator returns good results. It detects very well the accuracy of the reduced order approximation. When comparing the values it can be seen that the magnitude of the estimator changes as well as in the real error. The computational expenses for the estimator are as introduced in the previous two runs.

| $\varepsilon_{ss}$ | $\|\mu^{N,\star} - \mu^{\ell,\star}\|_2$ | (3.29) | (3.30) |
|---|---|---|---|
| $10^{-6}$ | 0.002797 | 0.087951 | 0.085666 |
| $10^{-4}$ | 0.000195 | 0.008078 | 0.008089 |

Table 3.6: Run 3: Error estimator for the obtained solution $\mu^{\ell,\star}$ by adaptive reduced order model optimization.

# Conclusion

In this work an efficient strategy for solving a nonlinear parameter estimation problem is outlined. The underlying system of equations is motivated by mathematical models for lithium ion batteries. A stable nonlinear solver for the nonlinear system is described and an efficient reduced order model using POD and EIM is developed. By combining the reduced order model and the finite element discretization an efficient solver for the parameter estimation problem is introduced.

In the course of the development of the different components, solutions to various problems are developed. There are still some open questions. On the one hand there is the question whether the studied problem or class of problems is suited as the model for a lithium ion battery. This issue was not addressed in this work since it is related to modeling. The inclusion of sensitivity information into the generation of the basis of the reduced order model would be an interesting extension to the proposed strategy. This could allow a dynamic change of the POD basis during the optimization process. Furthermore, the number of parameters used in the estimation problem can be increased which would bring the described problem closer to real world application. An open problem is also the error indicator used in the optimization strategy. A sophisticated error indicator can be developed that provides a rigorous error bound for the reduced order model. Lastly, the a-posteriori error estimator can be incorporated into the optimization strategy. This would allow to reduce the number of assumptions made in order to get the error bound.

# Appendix A

# Derivatives

## A.1 First order derivatives

In this section the first order derivatives for the nonlinear terms used in nonlinear system (1.1) are presented. Let us start with the derivatives of the nonlinearity $\mathcal{N}$ given by (1.2a) and the indicator function (1.2b) introduced in Chapter 1. We will use the compact notation and use $z = (y, p, q)$. The derivatives with respect to the variables $y$, $p$ and $q$ are given by

$$\begin{aligned}
\frac{\partial \mathcal{N}}{\partial y}(\mathbf{x}, z; \mu) &= \mathcal{N}_y = \chi(\mathbf{x}; \mu) \frac{\sinh\big(\mu_1(q-p) - \ln y\big) - 2\cosh\big(\mu_1(q-p) - \ln y\big)}{2\sqrt{y}}, \\
\frac{\partial \mathcal{N}}{\partial p}(\mathbf{x}, z; \mu) &= \mathcal{N}_p = -\chi(\mathbf{x}; \mu)\mu_1\sqrt{y}\cosh\big(\mu_1(q-p) - \ln y\big), \\
\frac{\partial \mathcal{N}}{\partial q}(\mathbf{x}, z; \mu) &= \mathcal{N}_q = \chi(\mathbf{x}; \mu)\mu_1\sqrt{y}\cosh\big(\mu_1(q-p) - \ln y\big)
\end{aligned} \tag{A.1}$$

and

$$\frac{\partial \chi}{\partial y}(\mathbf{x}; \mu) = \frac{\partial \chi}{\partial p}(\mathbf{x}; \mu) = \frac{\partial \chi}{\partial q}(\mathbf{x}; \mu) = 0.$$

Further, the derivatives with respect to the parameter $\mu$ read as

$$\begin{aligned}
\frac{\partial \mathcal{N}}{\partial \mu_1}(\mathbf{x}, z; \mu) &= \mathcal{N}_{\mu_1} = \chi(\mathbf{x}; \mu)\sqrt{y}\cosh\big(\mu_1(q-p) - \ln y\big)(q-p) \\
\frac{\partial \mathcal{N}}{\partial \mu_2}(\mathbf{x}, z; \mu) &= \mathcal{N}_{\mu_2} = \frac{\partial \chi}{\partial \mu_2}(\mathbf{x}; \mu)\sqrt{y}\sinh\big(\mu_1(q-p) - \ln y\big) \\
\frac{\partial \mathcal{N}}{\partial \mu_3}(\mathbf{x}, z; \mu) &= \mathcal{N}_{\mu_3} = \frac{\partial \chi}{\partial \mu_3}(\mathbf{x}; \mu)\sqrt{y}\sinh\big(\mu_1(q-p) - \ln y\big) \\
\frac{\partial \mathcal{N}}{\partial \mu_4}(\mathbf{x}, z; \mu) &= \mathcal{N}_{\mu_4} = 0
\end{aligned} \tag{A.2}$$

with

$$\frac{\partial \chi}{\partial \mu_2}(\mathbf{x}; \mu) = \begin{cases} 1 & \text{for} \quad \mathbf{x} \in [a, s_1], \\ 0 & \text{for} \quad \mathbf{x} \in (s_1, s_2), \\ 0 & \text{for} \quad \mathbf{x} \in [s_2, b], \end{cases} \quad \text{and} \quad \frac{\partial \chi}{\partial \mu_3}(\mathbf{x}; \mu) = \begin{cases} 0 & \text{for} \quad \mathbf{x} \in [a, s_1], \\ 0 & \text{for} \quad \mathbf{x} \in (s_1, s_2), \\ 1 & \text{for} \quad \mathbf{x} \in [s_2, b], \end{cases}$$

and

$$\frac{\partial \chi}{\partial \mu_1}(\mathbf{x};\mu) = \frac{\partial \chi}{\partial \mu_4}(\mathbf{x};\mu) = 0.$$

Next we state the derivatives of the nonlinear diffusion coefficient introduced by (1.39b) in Chapter 1. With respect to the variables $y$, $p$ and $q$ we get

$$\frac{\partial c_2}{\partial y}(y;\mu) = (c_2)_y = 3\mu_4(1+\mu_4 y)^2 \tag{A.3}$$

and

$$\frac{\partial c_2}{\partial p}(y;\mu) = \frac{\partial c_2}{\partial q}(y;\mu) = 0.$$

The derivatives with respect to the parameter $\mu$ are given by

$$\frac{\partial c_2}{\partial \mu_4}(y;\mu) = (c_2)_{\mu_4} = 3y(1+\mu_4 y)^2 \tag{A.4}$$

and

$$\frac{\partial c_2}{\partial \mu_1}(y;\mu) = \frac{\partial c_2}{\partial \mu_2}(y;\mu) = \frac{\partial c_2}{\partial \mu_3}(y;\mu) = 0.$$

## A.2 Second order derivatives

Here we present the second order derivatives required in Chapter 3. The derivatives with respect to the variables and the parameters for the nonlinearities $\mathcal{N}$ and $c_2$ are presented. For a more compact notation we summarize the variable as $z = (y, p, q)$ when ever possible. The appearing first order derivatives are not restated since they are all introduced in Appendix A.1. Let us start with the derivatives for the nonlinear term $\mathcal{N}$ given by (1.2a). First off the derivatives with respect to the variables $y$, $p$ and $q$.

$$\begin{aligned}
\frac{\partial^2 \mathcal{N}}{\partial y^2}(\mathbf{x},z;\mu) &= \mathcal{N}_{yy} = \chi(\mathbf{x};\mu)\frac{3\sinh\big(\mu_1(q-p)-\ln y\big)}{4y^{3/2}},\\
\frac{\partial^2 \mathcal{N}}{\partial y \partial p}(\mathbf{x},z;\mu) &= \mathcal{N}_{yp} = -\chi(\mathbf{x};\mu)\mu_1\frac{\cosh\big(\mu_1(q-p)-\ln y\big)-2\sinh\big(\mu_1(q-p)-\ln y\big)}{2\sqrt{y}},\\
\frac{\partial^2 \mathcal{N}}{\partial y \partial q}(\mathbf{x},z;\mu) &= \mathcal{N}_{yq} = \chi(\mathbf{x};\mu)\mu_1\frac{\cosh\big(\mu_1(q-p)-\ln y\big)-2\sinh\big(\mu_1(q-p)-\ln y\big)}{2\sqrt{y}},\\
\frac{\partial^2 \mathcal{N}}{\partial p^2}(\mathbf{x},z;\mu) &= \mathcal{N}_{pp} = \chi(\mathbf{x};\mu)\mu_1^2\sqrt{y}\sinh\big(\mu_1(q-p)-\ln y\big),\\
\frac{\partial^2 \mathcal{N}}{\partial p \partial q}(\mathbf{x},z;\mu) &= \mathcal{N}_{pq} = -\chi(\mathbf{x};\mu)\mu_1^2\sqrt{y}\sinh\big(\mu_1(q-p)-\ln y\big),\\
\frac{\partial^2 \mathcal{N}}{\partial q^2}(\mathbf{x},z;\mu) &= \mathcal{N}_{qq} = \chi(\mathbf{x};\mu)\mu_1^2\sqrt{y}\sinh\big(\mu_1(q-p)-\ln y\big).
\end{aligned}$$

Note that all second derivatives with respect to $y$, $p$ and $q$ of $\chi(\mathbf{x};\mu)$ are equal to zero. Additionally, we get

$$\mathcal{N}_{yp} = \mathcal{N}_{py}, \quad \mathcal{N}_{yq} = \mathcal{N}_{qy} \quad \text{and} \quad \mathcal{N}_{pq} = \mathcal{N}_{qp}.$$

Next we have a look at the derivatives with respect to the parameter $\mu$.

$$\begin{aligned}
\frac{\partial^2 \mathcal{N}}{\partial \mu_1^2}(\mathbf{x}, z; \mu) &= \mathcal{N}_{\mu_1\mu_1} = \chi(\mathbf{x}; \mu)\sqrt{y}\cosh\big(\mu_1(q-p) - \ln y\big)(q-p)^2, \\
\frac{\partial^2 \mathcal{N}}{\partial \mu_1 \partial \mu_2}(\mathbf{x}, z; \mu) &= \mathcal{N}_{\mu_1\mu_2} = \frac{\partial \chi}{\partial \mu_2}(\mathbf{x}; \mu)\sqrt{y}\cosh\big(\mu_1(q-p) - \ln y\big)(q-p), \\
\frac{\partial^2 \mathcal{N}}{\partial \mu_1 \partial \mu_3}(\mathbf{x}, z; \mu) &= \mathcal{N}_{\mu_1\mu_3} = \frac{\partial \chi}{\partial \mu_3}(\mathbf{x}; \mu)\sqrt{y}\cosh\big(\mu_1(q-p) - \ln y\big)(q-p)
\end{aligned}$$

together with

$$\mathcal{N}_{\mu_1\mu_2} = \mathcal{N}_{\mu_2\mu_1} \quad \text{and} \quad \mathcal{N}_{\mu_1\mu_3} = \mathcal{N}_{\mu_3\mu_1}.$$

Note the all second derivatives of $\chi(\mathbf{x}; \mu)$ with respect to $\mu$ are equal to zero. Furthermore, the remaining derivatives are

$$\begin{aligned}
\mathcal{N}_{\mu_1\mu_4} &= \mathcal{N}_{\mu_4\mu_1} = \mathcal{N}_{\mu_2\mu_2} = \mathcal{N}_{\mu_2\mu_3} = \mathcal{N}_{\mu_3\mu_2} \\
&= \mathcal{N}_{\mu_2\mu_4} = \mathcal{N}_{\mu_4\mu_2} = \mathcal{N}_{\mu_3\mu_3} = \mathcal{N}_{\mu_3\mu_4} = \mathcal{N}_{\mu_4\mu_3} = 0.
\end{aligned}$$

Lastly, we have a look at the mixed derivative, i.e. derivatives involving the variables $y$, $p$ and $p$ as well as the parameter $\mu$. We get

$$\begin{aligned}
\frac{\partial^2 \mathcal{N}}{\partial y \partial \mu_1}(\mathbf{x}, z; \mu) &= \mathcal{N}_{y\mu_1} \\
&= \chi(\mathbf{x}; \mu)\frac{(q-p)\big(\cosh\big(\mu_1(q-p) - \ln y\big) - 2\sinh\big(\mu_1(q-p) - \ln y\big)\big)}{2\sqrt{y}}, \\
\frac{\partial^2 \mathcal{N}}{\partial y \partial \mu_2}(\mathbf{x}, z; \mu) &= \mathcal{N}_{y\mu_2} \\
&= \frac{\partial \chi}{\partial \mu_2}(\mathbf{x}; \mu)\frac{\sinh\big(\mu_1(q-p) - \ln y\big) - 2\cosh\big(\mu_1(q-p) - \ln y\big)}{2\sqrt{y}}, \\
\frac{\partial^2 \mathcal{N}}{\partial y \partial \mu_3}(\mathbf{x}, z; \mu) &= \mathcal{N}_{y\mu_3} \\
&= \frac{\partial \chi}{\partial \mu_3}(\mathbf{x}; \mu)\frac{\sinh\big(\mu_1(q-p) - \ln y\big) - 2\cosh\big(\mu_1(q-p) - \ln y\big)}{2\sqrt{y}},
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 \mathcal{N}}{\partial p \partial \mu_1}(\mathbf{x}, z; \mu) &= \mathcal{N}_{p\mu_1} \\
&= -\chi(\mathbf{x}; \mu)\big(\sqrt{y}\cosh\big(\mu_1(q-p) - \ln y\big) + \mu_1\sinh\big(\mu_1(q-p) - \ln y\big)(q-p)\big), \\
\frac{\partial^2 \mathcal{N}}{\partial p \partial \mu_2}(\mathbf{x}, z; \mu) &= \mathcal{N}_{p\mu_2} \\
&= -\frac{\partial \chi}{\partial \mu_2}(\mathbf{x}; \mu)\mu_1\sqrt{y}\cosh\big(\mu_1(q-p) - \ln y\big), \\
\frac{\partial^2 \mathcal{N}}{\partial p \partial \mu_3}(\mathbf{x}, z; \mu) &= \mathcal{N}_{p\mu_3} \\
&= -\frac{\partial \chi}{\partial \mu_3}(\mathbf{x}; \mu)\mu_1\sqrt{y}\cosh\big(\mu_1(q-p) - \ln y\big),
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 \mathcal{N}}{\partial q \partial \mu_1}(\mathbf{x}, z; \mu) &= \mathcal{N}_{q\mu_1} \\
&= \chi(\mathbf{x}; \mu)\big(\sqrt{y}\cosh\big(\mu_1(q-p) - \ln y\big) + \mu_1 \sinh\big(\mu_1(q-p) - \ln y\big)(q-p)\big), \\
\frac{\partial^2 \mathcal{N}}{\partial q \partial \mu_2}(\mathbf{x}, z; \mu) &= \mathcal{N}_{q\mu_2} \\
&= \frac{\partial \chi}{\partial \mu_2}(\mathbf{x}; \mu)\mu_1\sqrt{y}\cosh\big(\mu_1(q-p) - \ln y\big), \\
\frac{\partial^2 \mathcal{N}}{\partial q \partial \mu_3}(\mathbf{x}, z; \mu) &= \mathcal{N}_{q\mu_3} \\
&= \frac{\partial \chi}{\partial \mu_3}(\mathbf{x}; \mu)\mu_1\sqrt{y}\cosh\big(\mu_1(q-p) - \ln y\big)
\end{aligned}$$

and

$$\begin{aligned}
&\mathcal{N}_{y\mu_1} = \mathcal{N}_{\mu_1 y}, \quad \mathcal{N}_{y\mu_2} = \mathcal{N}_{\mu_2 y}, \quad \mathcal{N}_{y\mu_3} = \mathcal{N}_{\mu_3 y}, \\
&\mathcal{N}_{p\mu_1} = \mathcal{N}_{\mu_1 p}, \quad \mathcal{N}_{p\mu_2} = \mathcal{N}_{\mu_2 p}, \quad \mathcal{N}_{p\mu_3} = \mathcal{N}_{\mu_3 p}, \\
&\mathcal{N}_{q\mu_1} = \mathcal{N}_{\mu_1 q}, \quad \mathcal{N}_{q\mu_2} = \mathcal{N}_{\mu_2 q}, \quad \mathcal{N}_{q\mu_3} = \mathcal{N}_{\mu_3 q}.
\end{aligned}$$

Furthermore, we have that the derivatives involving the parameter $\mu_4$ are given all equal to zero, i.e.

$$\mathcal{N}_{y\mu_4} = \mathcal{N}_{\mu_4 y} = \mathcal{N}_{p\mu_4} = \mathcal{N}_{\mu_4 p} = \mathcal{N}_{q\mu_4} = \mathcal{N}_{\mu_4 q} = 0$$

The mixed derivatives of $\chi(\mathbf{x}, \mu)$ are as well equal to zero. This complete the required second derivatives of the nonlinear term $\mathcal{N}$.

Next we state the second derivatives of the nonlinear diffusion coefficient $c_2$ given by (1.39b). Since this nonlinearity only depends on the variable $y$ and the parameter $\mu_4$ we right away state all the derivatives. We get

$$\begin{aligned}
\frac{\partial^2 c_2}{\partial y^2} &= (c_2)_{yy} = 6\mu_4^2(1 + \mu_4 y), \\
\frac{\partial^2 c_2}{\partial \mu_4^2} &= (c_2)_{\mu_4\mu_4} = 6y^2(1 + \mu_4 y), \\
\frac{\partial^2 c_2}{\partial y \partial \mu_4} &= (c_2)_{y\mu_4} = 3(1 + 3\mu_4 y)(1 + \mu_4 y)
\end{aligned}$$

and

$$(c_2)_{y\mu_4} = (c_2)_{\mu_4 y}.$$

Note that all the derivatives involving the variable $p$, $q$ and $\mu_i$, $i = 1 \ldots, 3$ are equal to zero.

## A.3 Derivatives for the Hessian

In this section we present the derivatives required to compute the Hessian for the least squares problem introduced in Section 3.6.1. The used derivatives of the nonlinear term

can be found in Appendix A.1 and A.2. All derivatives are given in a compact form, i.e. subindices indicate the derivative. Additionally, the dependency on the spatial variable $\mathbf{x}$ and the time variable $t$ are not indicated. Furthermore, the variables $y$, $p$ and $q$ are summarized to $z$ when ever possible.

We start by computing the derivatives for the cost function $J$. For this, let us first state the cost function again. We have

$$J(z,\mu) = \frac{1}{2}\int_0^T |q(t,b) - q^d(t)|^2 \,\mathrm{d}t.$$

The first and second order derivatives of $J$ are then given as

$$J_q(z,\mu)\delta q \int_0^T (q(t,b) - q^d(t))\delta q(t,b)\,\mathrm{d}t \quad \text{and} \quad J_{qq}(z,\mu)\delta q\tilde{\delta q} = \int_0^T \delta q(t,b)\delta q(\tilde{t},b)\,\mathrm{d}t.$$

All other derivatives of $J$ are equal to zero. Now let us next recall the definition of $e$:

$$e(z,\mu) = \begin{pmatrix} e^y(z,\mu) \\ e^p(z,\mu) \\ e^q(z,\mu) \end{pmatrix} = \begin{pmatrix} y_t - (c_1 y_{\mathbf{x}})_{\mathbf{x}} + \mathcal{N}(\,\cdot\,, z;\mu) \\ -(c_2(y;\mu)p_{\mathbf{x}})_{\mathbf{x}} + \mathcal{N}(\,\cdot\,, z;\mu) \\ -(c_3 q_{\mathbf{x}})_{\mathbf{x}} - \mathcal{N}(\,\cdot\,, z;\mu) \end{pmatrix}.$$

For the first order derivatives with the respect to $z$ and $\mu$ are given as

$$\begin{aligned}
&e^y_y\delta y = \delta y_t - (c_1\delta y_{\mathbf{x}})_{\mathbf{x}} + \mathcal{N}_y\delta y, && e^p_y\delta y = -((c_2)_y\delta y\, p_{\mathbf{x}})_{\mathbf{x}} + \mathcal{N}_y\delta y, && e^q_y = -\mathcal{N}_y\delta y,\\
&e^y_p\delta p = \mathcal{N}_p\delta p, && e^p_p\delta p = -(c_2\delta p_{\mathbf{x}})_{\mathbf{x}} + \mathcal{N}_p\delta p, && e^q_p = -\mathcal{N}_y\delta p,\\
&e^y_q\delta q = \mathcal{N}_q\delta q, && e^p_q\delta q = \mathcal{N}_q\delta q, && e^q_q = (c_3\delta q_{\mathbf{x}})_{\mathbf{x}} - \mathcal{N}_q\delta q
\end{aligned}$$

and

$$\begin{aligned}
&e^y_{\mu_1} = \mathcal{N}_{\mu_1}, && e^y_{\mu_2} = \mathcal{N}_{\mu_2}, && e^y_{\mu_3} = \mathcal{N}_{\mu_3}, && e^y_{\mu_4} = 0,\\
&e^p_{\mu_1} = \mathcal{N}_{\mu_1}, && e^p_{\mu_2} = \mathcal{N}_{\mu_2}, && e^p_{\mu_3} = \mathcal{N}_{\mu_3}, && e^p_{\mu_4} = -((c_2)_{\mu_4}p_{\mathbf{x}})_{\mathbf{x}},\\
&e^q_{\mu_1} = \mathcal{N}_{\mu_1}, && e^q_{\mu_2} = \mathcal{N}_{\mu_2}, && e^q_{\mu_3} = \mathcal{N}_{\mu_3}, && e^q_{\mu_4} = 0.
\end{aligned}$$

Next we state the second derivatives. We start with second derivatives with respect to $z$ or $\mu$. The mixed derivatives are given in a second step. The derivatives read as

$$\begin{aligned}
&e^y_{yy}\delta y\tilde{\delta y} = \mathcal{N}_{yy}\delta y\tilde{\delta y}, && e^p_{yy}\delta y\tilde{\delta y} = -((c_2)_{yy}\delta y\tilde{\delta y}\, p_{\mathbf{x}})_{\mathbf{x}} + \mathcal{N}_{yy}\delta y\tilde{\delta y}, && e^q_{yy}\delta y\tilde{\delta y} = -\mathcal{N}_{yy}\delta y\tilde{\delta y},\\
&e^y_{yp}\delta y\delta p = \mathcal{N}_{yp}\delta y\delta p, && e^p_{yp}\delta y\delta p = -((c_2)_y\delta y\,\delta p_{\mathbf{x}})_{\mathbf{x}} + \mathcal{N}_{yp}\delta y\delta p, && e^q_{yp}\delta y\delta p = -\mathcal{N}_{yp}\delta y\delta p,\\
&e^y_{yq}\delta y\delta q = \mathcal{N}_{yq}\delta y\delta q, && e^p_{yq}\delta y\delta q = \mathcal{N}_{yq}\delta y\delta q, && e^q_{yq}\delta y\delta q = -\mathcal{N}_{yq}\delta y\delta q,\\
&e^y_{py}\delta p\delta y = \mathcal{N}_{py}\delta p\delta y, && e^p_{py}\delta p\delta y = -((c_2)_y\delta y\,\delta p_{\mathbf{x}})_{\mathbf{x}} + \mathcal{N}_{py}\delta p\delta y, && e^q_{py}\delta p\delta y = -\mathcal{N}_{py}\delta p\delta y,\\
&e^y_{pp}\delta p\tilde{\delta p} = \mathcal{N}_{pp}\delta p\tilde{\delta p}, && e^p_{pp}\delta p\tilde{\delta p} = \mathcal{N}_{pp}\delta p\tilde{\delta p}, && e^q_{pp}\delta p\tilde{\delta p} = -\mathcal{N}_{pp}\delta p\tilde{\delta p},\\
&e^y_{pq}\delta p\delta q = \mathcal{N}_{pq}\delta p\delta q, && e^p_{pq}\delta p\delta q = \mathcal{N}_{pq}\delta p\delta q, && e^q_{pq}\delta p\delta q = -\mathcal{N}_{pq}\delta p\delta q,\\
&e^y_{qy}\delta q\delta y = \mathcal{N}_{qy}\delta q\delta y, && e^p_{qy}\delta y\delta q = \mathcal{N}_{qy}\delta q\delta y, && e^q_{qy}\delta q\delta y = -\mathcal{N}_{qy}\delta q\delta y,\\
&e^y_{qp}\delta q\tilde{\delta p} = \mathcal{N}_{qp}\delta q\tilde{\delta p}, && e^p_{qp}\delta q\delta p = \mathcal{N}_{qp}\delta q\delta p, && e^q_{qp}\delta q\tilde{\delta p} = -\mathcal{N}_{qp}\delta q\tilde{\delta p},\\
&e^y_{qq}\delta q\tilde{\delta q} = \mathcal{N}_{qq}\delta q\tilde{\delta q}, && e^p_{qq}\delta q\tilde{\delta q} = \mathcal{N}_{qq}\delta q\tilde{\delta q}, && e^q_{qq}\delta q\tilde{\delta q} = -\mathcal{N}_{qq}\delta q\tilde{\delta q}
\end{aligned}$$

and

$$
\begin{array}{llll}
e^y_{\mu_1\mu_1} = \mathcal{N}_{\mu_1\mu_1}, & e^y_{\mu_1\mu_2} = \mathcal{N}_{\mu 1\mu_2}, & e^y_{\mu_1\mu_3} = \mathcal{N}_{\mu_1\mu_3}, & e^y_{\mu_1\mu_4} = 0, \\
e^y_{\mu_2\mu_1} = \mathcal{N}_{\mu_2\mu_1}, & e^y_{\mu_2\mu_2} = \mathcal{N}_{\mu 2\mu_2}, & e^y_{\mu_2\mu_3} = \mathcal{N}_{\mu_2\mu_3}, & e^y_{\mu_2\mu_4} = 0, \\
e^y_{\mu_3\mu_1} = \mathcal{N}_{\mu_3\mu_1}, & e^y_{\mu_3\mu_2} = \mathcal{N}_{\mu 3\mu_2}, & e^y_{\mu_3\mu_3} = \mathcal{N}_{\mu_3\mu_3}, & e^y_{\mu_3\mu_4} = 0, \\
e^y_{\mu_4\mu_1} = 0, & e^y_{\mu_4\mu_2} = 0, & e^y_{\mu_4\mu_3} = 0, & e^y_{\mu_4\mu_4} = 0, \\
e^p_{\mu_1\mu_1} = \mathcal{N}_{\mu_1\mu_1}, & e^p_{\mu_1\mu_2} = \mathcal{N}_{\mu 1\mu_2}, & e^p_{\mu_1\mu_3} = \mathcal{N}_{\mu_1\mu_3}, & e^p_{\mu_1\mu_4} = 0, \\
e^p_{\mu_2\mu_1} = \mathcal{N}_{\mu_2\mu_1}, & e^p_{\mu_2\mu_2} = \mathcal{N}_{\mu 2\mu_2}, & e^p_{\mu_2\mu_3} = \mathcal{N}_{\mu_2\mu_3}, & e^p_{\mu_2\mu_4} = 0, \\
e^p_{\mu_3\mu_1} = \mathcal{N}_{\mu_3\mu_1}, & e^p_{\mu_3\mu_2} = \mathcal{N}_{\mu 3\mu_2}, & e^p_{\mu_3\mu_3} = \mathcal{N}_{\mu_3\mu_3}, & e^p_{\mu_3\mu_4} = 0, \\
e^p_{\mu_4\mu_1} = 0, & e^p_{\mu_4\mu_2} = 0, & e^p_{\mu_4\mu_3} = 0, & e^p_{\mu_4\mu_4} = -((c_2)_{\mu_4\mu_4} p_{\mathbf{x}})_{\mathbf{x}}, \\
e^q_{\mu_1\mu_1} = -\mathcal{N}_{\mu_1\mu_1}, & e^q_{\mu_1\mu_2} = -\mathcal{N}_{\mu 1\mu_2}, & e^q_{\mu_1\mu_3} = -\mathcal{N}_{\mu_1\mu_3}, & e^q_{\mu_1\mu_4} = 0, \\
e^q_{\mu_2\mu_1} = -\mathcal{N}_{\mu_2\mu_1}, & e^q_{\mu_2\mu_2} = -\mathcal{N}_{\mu 2\mu_2}, & e^q_{\mu_2\mu_3} = -\mathcal{N}_{\mu_2\mu_3}, & e^q_{\mu_2\mu_4} = 0, \\
e^q_{\mu_3\mu_1} = -\mathcal{N}_{\mu_3\mu_1}, & e^q_{\mu_3\mu_2} = -\mathcal{N}_{\mu 3\mu_2}, & e^q_{\mu_3\mu_3} = -\mathcal{N}_{\mu_3\mu_3}, & e^q_{\mu_3\mu_4} = 0, \\
e^q_{\mu_4\mu_1} = 0, & e^q_{\mu_4\mu_2} = 0, & e^q_{\mu_4\mu_3} = 0, & e^q_{\mu_4\mu_4} = 0.
\end{array}
$$

Lastly, let us state the mixed second order derivatives. We get

$$
\begin{array}{llll}
e^y_{y\mu_1} = \mathcal{N}_{y\mu_1}\delta y, & e^y_{y\mu_2} = \mathcal{N}_{y\mu_2}\delta y, & e^y_{y\mu_3} = \mathcal{N}_{y\mu_3}\delta y, & e^y_{y\mu_4}\delta y = 0, \\
e^y_{p\mu_1} = \mathcal{N}_{p\mu_1}\delta p, & e^y_{p\mu_2} = \mathcal{N}_{p\mu_2}\delta p, & e^y_{p\mu_3} = \mathcal{N}_{p\mu_3}\delta p, & e^y_{p\mu_4}\delta p = 0, \\
e^y_{q\mu_1} = \mathcal{N}_{q\mu_1}\delta q, & e^y_{q\mu_2} = \mathcal{N}_{q\mu_2}\delta q, & e^y_{q\mu_3} = \mathcal{N}_{q\mu_3}\delta q, & e^y_{q\mu_4}\delta q = 0, \\
e^p_{y\mu_1} = \mathcal{N}_{y\mu_1}\delta y, & e^p_{y\mu_2} = \mathcal{N}_{y\mu_2}\delta y, & e^p_{y\mu_3} = \mathcal{N}_{y\mu_3}\delta y, & e^p_{y\mu_4}\delta y = -((c_2)_{y\mu_4}\delta y\, p_{\mathbf{x}})_{\mathbf{x}}, \\
e^p_{p\mu_1} = \mathcal{N}_{p\mu_1}\delta p, & e^p_{p\mu_2} = \mathcal{N}_{p\mu_2}\delta p, & e^p_{p\mu_3} = \mathcal{N}_{p\mu_3}\delta p, & e^p_{p\mu_4}\delta p = -((c_2)_{\mu_4}\delta y_{\mathbf{x}})_{\mathbf{x}}, \\
e^p_{q\mu_1} = \mathcal{N}_{q\mu_1}\delta q, & e^p_{q\mu_2} = \mathcal{N}_{q\mu_2}\delta q, & e^p_{q\mu_3} = \mathcal{N}_{q\mu_3}\delta q, & e^p_{q\mu_4}\delta q = 0, \\
e^q_{y\mu_1} = -\mathcal{N}_{y\mu_1}\delta y, & e^q_{y\mu_2} = -\mathcal{N}_{y\mu_2}\delta y, & e^q_{y\mu_3} = -\mathcal{N}_{y\mu_3}\delta y, & e^q_{y\mu_4}\delta y = 0, \\
e^q_{p\mu_1} = -\mathcal{N}_{p\mu_1}\delta p, & e^q_{p\mu_2} = -\mathcal{N}_{p\mu_2}\delta p, & e^q_{p\mu_3} = -\mathcal{N}_{p\mu_3}\delta p, & e^q_{p\mu_4}\delta p = 0, \\
e^q_{q\mu_1} = -\mathcal{N}_{q\mu_1}\delta q, & e^q_{q\mu_2} = -\mathcal{N}_{q\mu_2}\delta q, & e^q_{q\mu_3} = -\mathcal{N}_{q\mu_3}\delta q, & e^q_{q\mu_4}\delta q = 0.
\end{array}
$$

Note that the mixed derivatives are symmetric, i.e. $e_{\wp\mu_i}\delta\wp = e_{\mu_i\wp}\delta\wp$ for $i = 1, \ldots, 4$ and $\wp \in \{y, p, q\}$, hence they are not stated twice.

# Appendix B

# Miscellaneous

## B.1 Computer Information

| | |
|---|---|
| OS | openSUSE 11.4 (i586) |
| CPU | Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz |
| RAM | 4GB |
| MATLAB | 7.5.0.338 (R2007b) |

Table B.1: Information about the computer and software used for the numerical simulations.

# Deutsche Zusammenfassung

In dieser Arbeit wird ein nichtlineares Parameteridentifikationsproblem untersucht. Die zugrunde liegenden Gleichungen sind gegeben durch ein nichtlineares System von partiellen Differentialgleichungen (PDG). Effiziente Methoden zum Lösen des Identifikationsproblems werden entwickelt. Hierfür wird ein reduziertes Modell entwickelt, um eine Lösungsstrategie mit weniger Rechenaufwand zu erhalten.

Wir betrachten ein elliptisch-parabolisches PDG-System bestehend aus zwei elliptischen und einer parabolischen Gleichung. Gekoppelte Systeme dieser Art können als Verallgemeinerung von mathematischen Modellen für Lithium-Ionen-Batterien gesehen werden [33, 72, 86]. Die elliptischen Gleichungen in dem nichtlinearen PDG-System modellieren die Potentiale im Flüssig- und Festkörper, während die parabolische Gleichung die Konzentration der Lithium-Ionen entspricht. Die drei Gleichungen sind durch eine starke Nichtlinearität gekoppelt, welche durch die Verknüpfung vom Sinus-Hyperbolicus, der Wurzelfunktion und dem Logarithmus gegeben ist. Des weiteren sind die Gleichungen durch einen nichtlinearen Diffusionskoeffizienten gekoppelt. Gekoppelte Systeme dieser Art sind im Allgemeinen schwer zu lösen. Auf Grund der betrachteten Nichtlinearitäten muss der nichtlineare Löser angepasst werden. Eine Finite-Elemente-Methode wird verwendet, um das nichtlineare PDG-System numerisch zu lösen. Verschiedene Modifikationen, um einen stabilen Löser zu erhalten, werden beschrieben.

Die Diskretisierung des nichtlinearen PDG-Systems durch die Finite-Elemente-Methode führt zu einem hochdimensionalen diskreten System, welches teuer zu lösen ist. Das Ziel ist ein reduziertes Modell für das nichlineare PDG-System zu entwickeln, welches schnell und effizient berechnet werden kann. Dies ist motiviert durch Anwendungen wie Identifikations-, Steuerungs- oder Designprobleme. In diesen Aufgabenstellungen muss das nichtlineare System wiederholt gelöst werden. Somit ersetzen wir die Diskretisierung (räumlich) durch einen Galerkinansatz unter Verwendung der *Proper Orthogonal Decomposition* (POD) Methode [50, 56, 77]. POD wird verwendet, um eine Basis zu generieren, welche einen Unterraum aufspannt, der die Charakteristiken der erwarteten Lösungen beschreibt. Die Finite-Elemente-Methode im Gegensatz berücksichtigt bei der Basiswahl nicht die Dynamik der zugrundeliegenden Gleichungen. Die Methode der Snapshots zum Berechnen der POD-Basis wird beschrieben und verschiedene Varianten der Berechung aufgezeigt.

Des weiteren wird die POD-Basisberechnung unter der Annahme beschrieben, dass mehrere Lösungen des nichtlinearen PDG-System vorhanden sind. Zwei Strategien zur Berechnung der Basis werden präsentiert. Der Standard-POD-Ansatz wird mit einem Algorithmus basierend auf der *POD-Greedy*-Strategie verglichen. Das Ziel ist es, die bestmög-

liche POD-Basis aus den gegebenen Daten zu extrahieren. Die Methode – basierend auf der Greedy-Strategie – erzeugt die Basis iterativ. Ähnliche Ansätze werden auch bei den reduzierten Basen Methoden verwendet [44, 45, 70].

Zusätzlich betrachten wir eine alternative numerische Umsetzung der Reduktionstechnik. Diese erlaubt es, ein reduziertes Modell unabhängig von der Ausgangsdiskretisierungsmethode zu konstruieren. Somit kann für die Konstruktion ein Black-Box-Löser verwendet werden. Dies steht im Gegensatz zu dem Projektionsansatz, der in [13, 62] verwendet wird. Dort wird das reduzierte Modell durch Projektion des hochdimensionalen algebraischen Systems auf den POD-Unterraum generiert. Der Vorteil des neuen Ansatzes wird an Hand einer Lösung mit adaptivem Gitter gezeigt.

Um ein effizientes reduziertes Modell zu erhalten, wenden wir die *Empirical Interpolation Method* (EIM) an [4]. Diese Methode wird oft in Zusammenhang mit der reduzierten Basen-Methode und nichtlinearen Problemen verwendet [39, 65, 70]. Die EIM Methoden sind sehr effektiv um die Komplexität der Auswertung von nichtlinearen Termen zu reduzieren. Numerische Beispiele werden präsentiert, um die Effektivität der EIM Methode zu demonstrieren.

Nach dem Erzeugen eines effizienten reduzierten Modells wollen wir dies in unserem Parameteridentifikationsproblem anwenden. Die mathematischen Modelle für Lithium--Ionen-Batterien beinhalten eine Vielzahl an Parametern. Diese Parameter müssen bestimmt werden, um das Modell zu kalibrieren. Die Parameter können korreliert sein oder sind nicht alle sensitiv, also muss eine Strategie angewandt werden um geeignete Parameter für den Identifikationsprozess auszuwählen. Hierfür werden die Sensitivitäten berechnet und mit Hilfe der *Subset-Selection*-Methode die sensitiven Parameter extrahiert [6, 12, 31]. Wir nutzen hier, dass die Identifizierbarkeit anhand der Sensitivitäten charakterisiert werden kann [73]. Somit erhalten wir ein nichtlineares Ausgleichsproblem, welches durch numerische Optimierungsmethoden gelöst werden kann [55, 68, 82].

Wenn im Optimierungsprozess ein reduziertes Modell verwendet wird, führen wir einen Fehler ein, da das reduzierte Modell nicht exakt ist. Daher wird ein A-Posteriori-Fehlerschätzer entwickelt, um den Fehler zu schätzen. Wir verwenden die Resultate von [29, 53]. Des weiteren ist das reduzierte Modell erzeugt durch die POD Methode nur eine lokale Approximation des nichtlinearen Systems. Somit ist es notwendig sicherzustellen, dass die Approximation für den Verlauf der Optimierung gut ist. Hierfür verwenden wir einen vereinfachten Fehlerindikator, welcher aus der reduzierten Basen-Methode bekannt ist [39]. Unter Verwendung dieses Fehlerindikators wird eine adaptive Optimierungsstrategie mit reduzierten Modellen entwickelt. Diese kann das Identifikationsproblem effizient lösen. Dies wird durch die Kombination der Finite-Elemente-Methode und der reduzierten Modelle im Laufe des Optimierungsprozesses erreicht.

Die Arbeit ist wie folgt gegliedert:

- In Kapitel 1 wird das nichtlineare elliptisch-parabolische System formuliert und die Diskretisierung unter Verwendung der Finite-Elemente-Methode eingeführt. Die dazugehörenden A-Priori-Fehlerschätzer werden hergeleitet. Das Newton-Verfahren zum Lösen des nichtlinearen Systems wird eingeführt und die notwendigen Ände-

rungen erläutert. Das Kapitel wird abgerundet mit numerischen Lösungen des nichtlinearen Systems.

- Kapitel 2 beschäftigt sich mit der Entwicklung des reduzierten Modells. Die POD Methode wird eingeführt und das reduzierte Modell wird für den kontinuierlichen und den diskreten Fall hergeleitet. Um ein effizientes reduziertes Modell zu erhalten, wird die EIM Methode präsentiert und die Verknüpfung mit dem nichtlinearen System erläutert. Die POD-Basisberechnung wird veranschaulicht und numerische Resultate für die verschiedenen Ansätze präsentiert. Zusätzlich wird die Basisberechnung aus mehreren Finite-Elemente-Lösungen mittels POD-Greedy-Verfahren und für adaptive Finite-Elemente-Lösungen gezeigt. Schlussendlich werden numerische Resultate präsentiert, die die Effektivität des reduzierten Modells unterstreichen.

- Kapitel 3 führt das Identifikationsproblem ein. Grundlegende Resultate aus der Optimierung werden präsentiert. Das Konzept der Sensitivitätsanalyse wird vorgestellt und die Subset-Selection-Methode erläutert. Ein vereinfachtes Modell wird eingeführt, um die theoretischen Resultate zu demonstrieren. Verschiedene Ansätze für die Berechnung der ersten und zweiten Ableitung werden gezeigt. Schlussendlich wird die adaptive Optimierungsstrategie unter Verwendung der Resultate aus Kapitel 1 und Kapitel 2 entwickelt. Die Berechnung des A-Posterori-Fehlerschätzers wird dargelegt. Numerische Beispiele werden präsentiert, um die Effizienz der vorgeschlagenen Strategie zu demonstrieren.

- Die Arbeit wird abgerundet mit einer kurzen Schlussbemerkung, in welcher gelöste und noch offene Problemstellungen dargelegt werden.

# Bibliography


[1] R.A. Adams and J.J.F. Fournier. *Sobolev Spaces*. Pure and Applied Mathematics. Academic Press, Oxford, 2003.

[2] H.W. Alt. *Lineare Funktionalanalysis: Eine anwendungsorientierte Einführung*. Springer, Berlin Heidelberg, 5th edition, 2006.

[3] H.T. Banks, S. Dediu, S.L. Ernstberger, and F. Kappel. Generalized sensitivity and optimal experimental design. *Journal of Inverse and III-posed Problems*, 18:25–83, 2010.

[4] M. Barrault, Y. Maday, N.G. Nguyen, and A.T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique*, 339(9):667–672, 2004.

[5] J.W. Barrett and C.M. Elliott. Finite Element Approximation of the Dirichlet Problem Using the Boundary Penalty Method. *Numerische Mathematik*, 49:343–366, 1986.

[6] J.J. Batzel, S. Fürtinger, M. Bachar, M. Fink, and F. Kappel. Sensitivity identifiability of a baroreflex control system model. Technical report, Graz University of Technology, 2009.

[7] S.C. Benner and L.R. Scott. *The Mathematical Theory of Finite Element Methods*. Texts in Applied Mathematics 15. Springer-Verlag, New York, 3rd edition, 2008.

[8] J. Borggaard and V.L. Nunes. Fréchet Sensitivity Analysis for Partial Differential Equations with Distributed Parameters. *American Control Conference*, pages 1789–1794, 2011.

[9] M. Brack. *Optimale Snapshotwahl bei Proper-Orthogonal-Decomposition am Beispiel eines Batteriemodells*. Staatsexamsarbeit, University of Konstanz, 2012.

[10] D. Braess. *Finite Elemente*. Springer-Verlag, Berlin Heidelberg, 4th edition, 2007.

[11] A. Buffa, Y. Maday, A.T. Patera, C. Prud'homme, and G. Turinici. A priori convergence of the Greedy algorithm for the parametrized reduced basis method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:595–603, 2012.

[12] M. Burth, G.C. Verghese, and M. Vélez-Reyes. Subset selection for Improved parameter estimation in on-line identification of a synchronous generators. *IEEE Transactions on Power Systems*, 14(1):218–225, 1999.

[13] L. Cai and R. E. White. Reduction of Model Order Based on Proper Orthogonal Decomposition for Lithium-Ion Battery Simulations. *Journal of The Electrochemical Society*, 156(3):A154–A161, 2009.

[14] G. Caloz and J. Rappaz. *Numerical analysis for nonlinear and bifurcation problems*, volume V of *Techniques of Scientific Computing (Part 2)*. Elsevier, Amsterdam, 1997.

[15] D. Chapelle, A. Gariah, and J. Saint-Marie. Galerkin approximation with proper orthogonal decomposition: new error estimates and illustrative examples. *ESIAM: Mathematical Modelling and Numerical Analysis*, 46:731–757, 2012.

[16] S. Chaturantabut and D.C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[17] S. Chaturantabut and D.C. Sorensen. Application of POD and DEIM on a dimension reduction of nonlinear miscible viscous fingering in porous media. Technical Report 4, 2011.

[18] S. Chaturantabut and D.C. Sorensen. A state space estimate for POD-DEIM nonlinear model reduction. *SIAM Journal on Numerical Analysis*, 50(1):46–63, 2012.

[19] P.G. Ciarlet. *The finite element method for elliptic problems*, volume 4 of *Studies in Mathematics and its Applications*. North-Holland Publishing Company, New York, 1st edition, 1978.

[20] F. Clarke. *Functional Analysis, Calculus of Variations and Optimal Control*. Graduate Texts in Mathematics 264. Springer, 2013.

[21] R. Dautray and J.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology. Volume 5: Evolution Problems I*. Series Computational Mathematics 35. Addison-Wesley, Springer-Verlag, 1992.

[22] T.A. Davis, J.R. Gilbert, S.I. Larimore, and E.G. Ng. A column approximate minimum degree ordering algorithm. *ACM Transactions on Mathematical Software*, 30(3):353–376, 2004.

[23] T.A. Davis, J.R. Gilbert, S.I. Larimore, and E.G. Ng. Algorithm 836: COLAMD, a column approximate minimum degree ordering algorithm. *ACM Transactions on Mathematical Software*, 30(3):377–380, 2004.

[24] P. Deuflhard. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*. Series Computational Mathematics 35. Springer-Verlag, Berlin Heidelberg, 2004.

[25] P. Deuflhard and G. Heindl. Affine Invariant Convergence Theorems for Newton's Method and Extensions to Related Methods. *SIAM Journal on Numerical Analysis*, 16(1):1–10, 1979.

[26] P. Deuflhard and A. Hohmann. *Numerische Mathematik 1*. Walter de Gruyter, Berlin New York, 4th edition, 2008.

[27] P. Deuflhard and A. Hohmann. *Numerische Mathematik 2*. Walter de Gruyter, Berlin New York, 3rd edition, 2008.

[28] M. Dihlmann and B. Haasdonk. Certified nonlinear parameter optimization with reduced basis surrogate models. *Proceedings in Applied Mathematics and Mechanics*, 13:3–6, 2013.

[29] M. Dihlmann and B. Haasdonk. Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems. *Submitted*, 2013.

[30] L.C. Evans. *Partial Differential Equations*. American Mathematical Society, Providence, Rhode Island, 2002.

[31] M. Fink, A. Attarian, and H. Tran. Subset selection for parameter estimation in an HIV model. *Proceedings in Applied Mathematics and Mechanics*, 7(1):1121501–1121502, 2007.

[32] G. Fischer. *Lineare Algebra*. Eine Einführung für Studienanfänger. Vieweg+Teubner, Wiesbaden, 2010.

[33] T. Fuller, M. Doyle, and J. Newman. Modeling of galvanostatic charge and discharge of the lithium ion battery/polymer/insertion cell. *Journal of the Chemical Society*, 140(6):1526–1533, 1993.

[34] M.S. Gockenbach. *Understanding and Implementing the Finite Element Method*. Society for Industrial and Applied Mathematics, Philadelphia, 2006.

[35] G. Golub. Numerical methods for solving least squares problems. *Numerische Mathematik*, 7:206–216, 1965.

[36] G. Golub, V. Klema, and G.W. Stewart. Rank Degeneracy and Least Squares Problems. Technical report, Stanford University, 1977.

[37] G.H. Golub and J.M. Orthega. *Scientific Computing and Differential Equations*. Academic Press, Boston, 1st edition, 1992.

[38] G.H. Golub and C.F. van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 3 edition, 1996.

[39] M. Grepl. Certified reduced basis method for nonaffine linear time-varying and nonlinear parabolic partial differential equations. *WSPC: Mathematical Models and Methods in Applied Sciences*, 22(3):40, 2012.

[40] C. Grossmann and H.G. Roos. *Numerical Treatment of Partial Differential Equations*. Springer-Verlag, Berlin Heidelberg, 2007.

[41] L. Grüne and O. Junge. *Gewöhnliche Differentialgleichungen: Eine Einführung aus der Perspektive der dynamischen Systeme*. Vieweg+Teubner, Wiesbaden, 1st edition, 2009.

[42] M. Gu and S.C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal on Scientific Computing*, 17:848–869, 1996.

[43] M. Gubisch and S. Volkwein. Proper Orthogonal Decomposition for Linear-Quadratic Optimal Control. *Submitted*, 2013.

[44] B. Haasdonk. Convergence rates of the POD-greedy method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(3):859–873, 2013.

[45] B. Haasdonk and M. Ohlberger. Reduced Basis Method for Finite Volume Approximations of Parametrized Linear Evolution Equations. *Mathematical Modelling and Numerical Analysis*, 42(2):277–302, 2008.

[46] B. Haasdonk, J. Salomon, and B. Wohlmuth. A Reduced Basis Method for the Simulation of American Options. In *ENUMATH 2011 Proceedings*, 2012.

[47] A. Hay, J. Borggaard, and D. Pelletier. Local improvement to reduced-order models using sensitivity analysis of the proper orthogonal decomposition. *Journal of Fluid Mechanics*, 629:41–72, 2009.

[48] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*. Mathematical Modelling: Theory and Application 23. Springer, Netherlands, 2008.

[49] M. Hinze and S. Volkwein. Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition. *Computational Optimization and Application*, 39:319–345, 2008.

[50] P. Holmes, J.L. Lumley, G. Berkooz, and C.W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge Monographs on Mechanics. Cambridge University Press, Cambridge, 2nd edition, 2012.

[51] K. Ito and K. Kunisch. On a semi-smooth Newton method and its globalization. *Mathematical Programming*, 118(2):347–370, 2009.

[52] C. Jarvis. Reduced Order Model Study of Burger's Equation using Proper Orthogonal Decomposition. Master's thesis, Virginia Polytechnic Institute and State University, 2012.

[53] E. Kammann, F. Tröltzsch, and S. Volkwein. A method of a-posteriori error estimation with application to proper orthogonal decomposition. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47:555–581, 2013.

[54] F. Kappel. Modeling the Dynamics of the Cardiovascular-respiratory System (CVRS) in Humans, a Survey. *Mathematical Modelling of Natural Phenomena*, 7(5):65–77, 2012.

[55] C.T. Kelley. *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, 1999.

[56] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische Mathematik*, 90:117–148, 2001.

[57] K. Kunisch and S. Volkwein. Crank-Nicolson Galerkin proper orthogonal decomposition approximations for a general equation in fluid dynamics. *In Proceedings of the 18th GAMM Seminar on Multigrid and related methods for optimization problems, Leipzig, Germany, January 24-26, 2002, M. Griebel and W. Hackbusch eds.*, pages 97–114, 2002.

[58] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40:492–515, 2002.

[59] K. Kunisch and S. Volkwein. Optimal snapshot location for computing POD basis functions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(3):509–529, 2010.

[60] M.G. Larson and F. Bengzon. *The Finite Element Method: Theory, Implementation, and Practice*. Springer-Verlag, Berlin Heidelberg, 2010.

[61] O. Lass and S. Volkwein. Parameter identification for nonlinear elliptic-parabolic systems with application in lithium-ion battery modeling. *Submitted*, 2013.

[62] O. Lass and S. Volkwein. POD Galerkin schemes for nonlinear elliptic-parabolic systems. *SIAM Journal on Scientific Computing*, 35(3):A1271–A1298, 2013.

[63] O. Lass and S. Volkwein. Adaptive POD basis computation for parametrized nonlinear systems using optimal snapshot location. *Computational Optimization and Application*, pages 1–33, 2014.

[64] D.G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, Inc., New York, 1969.

[65] Y. Maday, N.C. Nguyen, A.T. Patera, and G.S.H. Pau. A general, multipurpose interpolation procedure: the magic points. *Communications on Pure and Applied Analysis*, 8(1):383 – 404, 2009.

[66] H. Maurer and J. Zowe. First and second order necessary and sufficient optimality conditions for infinite-dimensional programming problems. *Mathematical Programming*, 16(1):98–110, 1979.

[67] M. Merkel and A. Öchsner. *Eindimensionale Finite Elemente*. Springer-Verlag, Berlin Heidelberg, 2010.

[68] J. Nocedal and S.J. Wright. *Numerical optimization*. Series in Operations Research. Springer-Verlag, New York, 2006.

[69] M.S. Olufsen and J.T. Ottesen. A practical approach to parameter estimation applied to model predicting heart rate regulation. *Journal of Mathematical Biology*, 67:39–68, 2013.

[70] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*. version 1.0. MIT Pappalardo Graduate Monographs in Mechanical Engineering, MIT, 2007.

[71] S.R. Pope. *Parameter Identification in Lumped Compartment Cardiorespiratory Models*. PhD thesis, North Carolina State University, 2009.

[72] P. Popov, Y. Vutov, S. Margenov, and O. Iliev. Finite volume discretization of equations describing nonlinear diffusion in li-ion batteries. *Numerical Methods and Applications, Lecture Notes in Computer Science*, 6046:338–346, 2011.

[73] J.G. Reid. Structural identifiability in linear time-invariant systems. *IEEE Trans Automat Control*, 22:242–246, 1977.

[74] E.W. Sachs and M. Schu. A-priori error estimates for reduced order models in finance. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(2):449–469, 2013.

[75] T. Seger. *Elliptic-Parabolic Systems with Application to Lithium-Ion Battery Models*. PhD thesis, University of Konstanz, 2013.

[76] J.R. Singler. New POD error expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs. *Submitted*, 2013.

[77] L. Sirovich. Turbulence and the dynamics of coherent structures. Parts I-II. *Quarterly of Applied Mathematics*, XVL:561–590, 1987.

[78] J. Stoer. *Numerische Mathematik 1*. Springer-Verlag, Berlin Heidelberg, 2005.

[79] G. Teschl. *Ordinary Differential Equations and Dynamical Systems*, volume 140 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, Rhode Island, 2012.

[80] V. Thomée. *Galerkin Finite Element Methods for Parabolic Problems*. Springer Series in Computational Mathematics 25. Springer-Verlag, Berlin Heidelberg, 2nd edition, 2006.

[81] F. Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, volume 120 of *Graduate Studies in Mathematics*. American Mathematical Society, 2010.

[82] C.R. Vogel. *Computational Methods for Inverse Problems*. Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 2002.

[83] S. Volkwein. Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling. Lecture Notes, University of Konstanz, 2012.

[84] S. Volkwein and A. Wesche. The reduced basis method applied to transport equations of a lithium-ion battery. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 32(5):1760–1772, 2013.

[85] D. Wirtz, D. Sorensen, and B. Haasdonk. A-posteriori error estimation for DEIM reduced nonlinear dynamical systems. *Submitted*, 2012.

[86] J. Wu, J. Xu, and H. Zou. On the well posedness of mathematical model for lithium-ion battery systems. *Methods and Applications of Analysis*, 13:275–298, 2006.

[87] M.J. Zahr, D. Amsallem, and C. Farhat. Construction of Parametrically-Robust CFD-Based Reduced-Order Models for PDE-Constrained Optimization. In *21st AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 2013.