Optimal Input Design for Large-Scale Inverse Problems using PDE-Constrained Optimization

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

vorgelegt von

Andrea Petrocchi

an der

Universität Konstanz

Mathematisch-Naturwissenschaftliche Sektion

Fachbereich für Mathematik und Statistik

Tag der mündlichen Prüfung: 18. Juni 2024

1. Referent: Prof. Dr. Stefan Volkwein
2. Referent: Prof. Dr. Mario Ohlberger

Abstract

In this thesis we consider the optimal input design of inverse problems using partial differential equation (PDE)-constrained optimization. While the analysis and numerical solution of inverse problems has made extensive progress in the last decades, some of these problems might depend on the model *input* or *control*. The field of optimal input/experiment design is an established framework for generating experiments or input functions in order to satisfy certain theoretical conditions. In our case we aim to optimize the *observability* of the parameters, i.e. their impact the model output. We analyze two classes of inverse problems, one driven by a general linear parabolic system and the other by a nonlinearly coupled elliptic-parabolic system. Then, we consider an optimal input design algorithm, whose purpose is to improve the parameter estimation process. Numerical experiments of the design algorithm conclude this work, and deliver satisfying results to both the mathematical problem and the industrial application.

We use, in particular, the reduced basis (RB) approximation in the framework of model order reduction (MOR) to define fast-computable approximated PDE-solutions. To solve the finite-dimensions inverse problem, we consider a trust-region optimization method, which requires RB error estimates in order to be properly defined and to guarantee its convergence. While the study and optimization of general linear parabolic systems is already well-studied, the nonlinear model lacks efficient and certified error estimates. Hence, one of the novelties of this dissertation is the approximation of hierarchical error estimates and especially their use in the optimization algorithm.

For the optimal input design we describe a standard algorithm and propose a variation for an industrial application. The standard formulation relies on an adaptive argument, where one finds a sequence of input functions that maximize the overall observability of the parameters, and therefore makes the convergence to the true parameters easier. Even though the input design problem lacks a theoretical framework, we observe a general success in its purpose, namely gaining trust in the estimated parameter. Furthermore, the algorithm is independent of the model and can be applied to general inverse problems.

Finally, we apply the input design algorithm to the industrial problem of estimating parameters for Lithium-ion battery cell systems. Since these systems are very complex and depend on many parameters, the estimation process is usually long and resource-expensive. Hence, the input design finds a shorter optimal input function, in this case the battery current profile. We apply two variations of the optimal input design algorithm previously defined in order to solve this problem.

Zusammenfassung

In dieser Arbeit befassen wir uns mit dem Design von optimalen Inputs für inverse Optimierungsprobleme mit partiellen Differentialgleichungen (PDE) als Nebenbedingung. Während die Analyse und numerische Lösung von inversen Problemen in den letzten Jahrzehnten große Fortschritte gemacht hat, können einige dieser Probleme von der Input- oder Kontroll-Funktionen. Das Gebiet des optimalen Input-/Experiment-Designs ist eine etablierte Methodik zur Generierung von Experimenten oder Input-Funktionen, um bestimmte theoretische Bedingungen zu erfüllen. In unserem Fall zielen wir darauf ab, die Beobachtbarkeit der Parameter, d.h. ihren Effekt auf den Output des Modells, zu optimieren. Wir betrachten dafür zwei Klassen inverser Probleme, eine bestimmt durch ein allgemeines lineares parabolisches System und die andere durch ein nichtlinear gekoppeltes elliptisch-parabolisches System. Anschließend betrachten wir einen Algorithmus für das optimale Input-Design, der das Ziel hat, den Parameterschätzungsprozess zu verbessern. Numerische Experimente des Algorithmus schließen diese Arbeit ab und liefern zufriedenstellende Ergebnisse sowohl für das mathematische Problem als auch für die industrielle Anwendung.

Zur effizienten Berechnung approximativer Lösungen der PDEs verwenden wir insbesondere den Ansatz der Reduzierten Basen (RB) im Rahmen der Modellreduktion (MOR). Um die eindlichdimensionalen inversen Probleme zu lösen, betrachten wir ein Trust-Region-Verfahren, welches eine genaue Definition und die Garantie der Konvergenz von RB-Fehlerschätzern voraussetzt. Während die Analyse und Optimierung allgemeiner linearer parabolischer Systeme bereits weitgehend erforscht ist, fehlt es im Fall des nichtlinearen Modells an effizienten und zertifizierten Fehlerschätzern. Daher ist die Approximation hierarchischer Fehlerschätzern und insbesondere deren Verwendung im Optimierungsalgorithmus eine der Neuheiten dieser Dissertation.

Für das Problem des optimalen Input-Designs wird ein Standardalgorithmus erörtert und eine Variation für eine industrielle Anwendung vorgeschlagen. Die Standardformulierung stützt sich dabei auf ein adaptives Argument, mit dem eine Folge von Input-funktionen gefunden wird, welche die allgemeine Beobachtbarkeit der Parameter maximiert und somit die Konvergenz zu den wahren Parametern erleichtert. Obwohl dem Problem des Input-Designs eine theoretische Grundlage fehlt, beobachten wir, dass es im Allgemeinen seinen Zweck erfüllt, Vertrauen in den geschätzten Parameter zu gewinnen. Zudem ist der Algorithmus unabhängig vom Modell und kann auf weitere, allgemeine inverse Probleme angewendet werden.

Schließlich wenden wir den Input-Design-Algorithmus auf das industrielle Problem der Parameterschätzung für Lithium-Ionen-Batteriezellensysteme an. Da diese Systeme sehr komplex sind und von vielen Parametern abhängen, ist der Schätzungsprozess in der Regel langwierig und ressourcenaufwendig. Daher findet das Input-Design eine kürzere Input-Funktion, in diesem Fall das Batteriestromprofil. Zur Lösung dieses Problems wenden wir zwei Varianten des zuvor definierten Algorithmus zum optimalen Input-Design an.

Acknowledgments

First and foremost, I would like to thank my supervisor Stefan Volkwein. He welcomed me in his group in July 2020 and guided me with interest and trust for more than three years. I am grateful for having had such a great mentor and boss.

I would like to thank Mario Ohlberger for his warm hospitality in Münster, his cordiality, and his availability for being a reviewer of this work.

Furthermore, I would like to thank Matthias Scharrer for having been a great guide and support to the industrial side of this project. He has shown genuine interest in my research and has been a great host every time I have visited Graz, where he has made me feel part of the team for the whole time.

Next, I thank Behzad Azmi for always being available to share his mathematical expertise and other interesting discussions in sincere and long talks.

I would like to especially thank the patient and willing proof-readers: Christian Jäkle, Simon Buchwald, Luca Petrocchi, and Svetlana Kapustina.

I acknowledge with gratitude the financial support by the Austrian Federal Ministry for Climate Action, the Austrian Federal Ministry for Digital and Economic Affairs, the Province of Styria (Dept. 12) and the Styrian Business Promotion Agency within the COMET K2 Competence Centers for Excellent Technologies, as well as the Deutsche Forschungsgemeinschaft for the project Localized Reduced Basis Methods for PDE-constrained Parameter Optimization under contract VO 1658/6-1.

Finally, I am thankful for the support of my family and my girlfriend, who have always believed in me, and my friends, who have made this very-long-vacation-out-of-Italy more bearable.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The topic of lithium-ion batteries has been of interest in the recent past in many electronic applications, for example electric mobility. Many companies and research institutes are therefore focusing on their mathematical modeling and testing, often involving great effort and cost in time and computational resources. The derivation of mathematical models that are both reliable and computationally cheap is now one of the biggest challenges in this field.

The most common mathematical models for lithium-ion batteries come from the study of the electrochemical processes, and consist of several coupled nonlinear equations describing the transport of lithium ions and the electrical potential in the so-called solid phases and electrolyte, in the different section of the battery, like the electrodes (i.e., the cathode and the anode) and the separators. Some more sophisticated models assume further conditions, for example external and thermal effects. In the last fifty years different approaches for the modelization have been proposed. A macro-homogeneous approach is developed in the pioneering work [42] and forms the basis for further investigations; cf. [17, 23, 24, 56, 62], for instance.

For more recent work about the modeling and discretization of lithium-ion battery cells, see, e.g., [47, 38, 39, 63, 20]. Well-posedness is studied, e.g., in [68, 61]. A lot of the interest is of course in approximating or estimating hidden parameters or states; cf., e.g., [60, 21, 57].

Our estimation process is based on a fitting argument. Chosen a certain input current, we observe the output voltage of the battery cell. The parameter estimation is then an optimization problem based on minimizing the discrepancy between the mathematical model and the real experimental response of the battery, by tweaking the parameters in the mathematical model. In this framework the input current is arbitrarily chosen, and, as observed in many works (e.g. [41]), the current impacts strongly on the identification of some of the parameters.

Therefore, in the context of optimal design of experiments – or in this case optimal design of inputs – we find an input or a collection of inputs that minimizes the uncertainty of the parameters, or that maximizes their observability, in order to estimate them better. For

more information about classic design of experiment, see [25, 44, 54, 50, 3, 49]. For the design of inputs applied to parameter identification in batteries see, e.g., [48].

In our work we will further use model order reduction techniques to speed up evaluations of our mathematical model. In particular, the reduced basis (RB) scheme is used; see, e.g., [30, 53] and [37, 40, 43, 67] for battery models. In this framework, we apply an adaptive trust-region method that does not need any a-priori computation of an RB subspace on an offline phase, but builds the RB approximation online based on computable a-posteriori error estimates; see [69, 51, 33, 45, 6, 5]. Furthermore, we would like to refer to earlier work on methods based on TR proper orthogonal decomposition (TR-POD); see [2, 59, 58].

1.1 Outline

This work is based on three previously published works: in the first one, [45], we consider a parameter-dependent evolution problem, where the parameters are unknown. A bi-level optimization input design algorithm is taken into consideration, where in the inner optimization problem the parameter vector is optimized in order to minimize a cost function measuring the discrepancy from some given data. In the outer cycle a new input is found in order to maximize some observability measure, evaluated through the Fisher information matrix. The new contribution of this paper was the use of the trust-region RB algorithm mentioned above (together with some novel ideas) in the context of parabolic equation.

In the second work, [4], a more complex mathematical model is taken into consideration, in order to slightly resemble a battery model. There we deal with a coupled nonlinear parabolic-elliptic system, where the nonlinear coupling function is given as an exponential function motivated by the Butler Volmer equation. While the finite element approximation and RB scheme follow standard arguments, we need fast-evaluable a-posteriori RB error estimates, usually not available for such nonlinear models. We develop approximated hierarchical error bounds, and show the good performances in terms of accuracy and efficiency of such estimate. Such approximated estimates also give satisfying results when used in the trust-region framework.

In the third work, available on [46], we do not study any mathematical models or develop any model order reduction techniques. Instead, with the support of our project partners we use a standard adaptive input design algorithm to develop a current profile in order to better estimate 9 parameters in a real battery model. The model is the one described and used in [57], with some of the parameters blocked at a reasonable "middle" value. We use such model as a black-box: we assume we can only control the battery parameters and current data as inputs, and observe the battery voltage as output. We show then the numerical results for two separate cases; in the first case we build a collection of inputs to maximize the observation in different situations (namely, in different battery "states", such as full and low battery charge); in the second case, on the other hand, we build a single long input current by stacking optimal pieces – where "optimal" here stands as maximizing the information. The reason for this is that resetting batteries to the initial configuration after each run takes some time, as a long resting interval is usually needed. With the one-long-input case, we wanted to avoid the resting pause and just develop a single run

experiment that could optimize the overall experiment time and still provide the necessary resting periods.

The thesis is developed as follows:

- In the rest of the Chapter 1, we introduce notation and preliminary concepts, such as finite element (FE) discretization, reduced order (RO) approximation and the discrete empirical interpolation method (DEIM).
- In Chapter 2 we introduce the linear parameter-dependent evolution problem like the one in [45]. This chapter will serve mostly as an introduction to the next one, and we show how the concepts of FE discretization, RO approximation, and the trust-region optimization algorithm are applied in this case.
- In Chapter 3 we study the nonlinear coupled parabolic-elliptic system as done in [4], where well-posedness is fully analyzed and better discussed. Then we consider the optimization problem and discretize it. In the RO approximation we overcome the difficulties deriving from the nonlinearity and finally we define the hierarchical approximated error estimates and their use in the trust region optimization algorithm.
- In Chapter 4 we talk about the input design algorithm and its application to the coupled model and the industrial model. We finally show numerical results for both cases and for a variation of the input design derived from the industrial needs.
- In Chapter 5 we conclude this dissertation with final remarks and discuss the outlooks in the different fields that this work touches.

1.2 Notation and preliminary concepts

In this section, we introduce the notation used in this thesis, as well as some concepts well discussed in other works; the introduction of basic concepts of functional analysis might be overlooked.

In general, we use lowercase *italic* for scalar, such as i, k, N, and functions, such as y, q, v, and lowercase roman for vectors, i.e. y, q, v. Matrices are indicated by roman uppercase letters, i.e. M, A, F. Two exceptions are the identity matrix in $\mathbb{R}^{N\times N}$, denoted by $\mathbb{I}_N$, and the parameter vector $\boldsymbol{\mu} \in \mathbb{R}^d = (\mu_i)_{i=1}^d$.

Overall, we append further information in the superscript and in the subscript of some quantities. In the superscript we might append the letters h or ℓ to specify the finite element and reduced order approximation, respectively. In the superscript will also appear the time step taken into consideration, usually indexed with the letter k. In the subscript we might instead append the parameter vector $\boldsymbol{\mu}$ or the letter $\mathrm{y}, \mathrm{q}, \mathrm{p}$ to explicit that a certain quantity refers to a state variable such as y or q, or to an adjoint variable such as p.

Starting from $\mathbb{R}$ the set of real numbers, the d-dimensional Euclidean space, the inner product, and the Euclidean norm are denoted by $\mathbb{R}^d$, $\langle\cdot,\cdot\rangle_{\mathbb{R}^d}$, and $\|\cdot\|_{\mathbb{R}^d}$, respectively. For a symmetric, positive definite matrix $\mathrm{M} \in \mathbb{R}^{d\times d}$ we define $\langle \mathrm{x}, \mathrm{y}\rangle_{\mathrm{M}} := \langle \mathrm{Mx}, \mathrm{y}\rangle_{\mathbb{R}^d}$ for any $\mathrm{x}, \mathrm{y} \in \mathbb{R}^d$. The induced norm is then defined as $\|\cdot\|_{\mathrm{M}} := \langle\cdot,\cdot\rangle_{\mathrm{M}}$.

Basic functional analysis Let $\Omega \subset \mathbb{R}$ denote an open and bounded interval and let $\partial\Omega$ denote its boundary. Many ideas in this section are well established for higher dimensions, but since we are going to be working with intervals we keep the notation relevant to our thesis. The Lebesgue spaces $L^p(\Omega)$ for $1 \leq p < \infty$ are defined as

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

with endowed norm

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

while setting $p = \infty$ we have

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

with endowed norm

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

For $1 \leq p \leq \infty$, $L^p(\Omega)$ is a Banach space, while in particular only $L^2(\Omega)$ (also denoted by $H^0(\Omega)$) is a Hilbert space with inner product

$$\langle \varphi, \psi \rangle_{L^2(\Omega)} := \int_\Omega \varphi(x)\psi(x)\mathrm{d}x.$$

For $1 \leq p < \infty$, the Sobolev spaces are defined as

$$W^{1,p}(\Omega) = \left\{ \varphi \in L^p(\Omega) \;\middle|\; \int_\Omega |\varphi'(x)|^p \mathrm{d}x < \infty \right\},$$

and in particular for $p = 2$ we define the Hilbert space $H^1(\Omega) := W^{1,2}(\Omega)$ with inner product

$$\langle \varphi, \psi \rangle_{H^1(\Omega)} := \int_\Omega \varphi(x)\psi(x)\mathrm{d}x + \int_\Omega \varphi'(x)\psi'(x)\mathrm{d}x \qquad \text{for } \varphi, \psi \in H^1(\Omega).$$

As they are fundamental in the study of parabolic equation, we define for the rest of this work $H := L^2(\Omega)$ and $V := H^1(\Omega)$. The sequence of continuous and dense embeddings $V \hookrightarrow H \cong H^* \hookrightarrow V^*$ is then called Gelfand triple (see, e.g., [31]).

1.2.1 Finite element discretization

Let us consider the linear evolution problem

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\langle y(t), \varphi \rangle_H + a(y(t), \varphi) &= \langle f, \varphi \rangle_H \quad \text{for all } \varphi \in V, \text{ f.a.a. } t \in [0, T], \\ y(0) &= y_\circ \qquad\qquad \text{in } H, \end{aligned} \tag{1.1}$$

for a bilinear form a and a function $f \in H$. Assuming existence and uniqueness conditions for $y \in W(0,T;V;V^*)$, let us introduce the discretization that we are going to use in the following chapters. We consider a finite-dimensional but potentially high-dimensional space $V^h \subset V$, with basis $\{\varphi_1, \dots, \varphi_N\}$, endowed with the same topology of V, and we introduce the Galerkin ansatz

$$y^h(t) = \sum_{j=1}^{N} \mathrm{y}_j^h(t)\varphi_j \qquad \text{f.a.a. } t \in [0,T], \tag{1.2}$$

where the vector $\mathrm{y}^h(t) \in \mathbb{R}^N$ is called coordinate vector (in the basis $\{\varphi_1, \dots, \varphi_N\}$). In particular, we choose one-dimensional linear Lagrangian finite elements (see, e.g., [52]) as basis set. Then, $y^h(t)$ is the solution of

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\langle y^h(t), \varphi_j\rangle_H + a(y^h(t), \varphi_j) &= \langle f, \varphi_j\rangle_H, \qquad \text{for } j = 1, \dots, N \text{ f.a.a. } t \in [0,T], \\ \langle y^h(0) - y_\circ^h, \varphi_j\rangle_H &= 0 \qquad \text{for } j = 1, \dots, N, \end{aligned} \tag{1.3}$$

where $y_\circ^h = \mathcal{P} y_\circ$, and $\mathcal{P}$ is the projection operator onto V^h defined as

$$\mathcal{P} y_\circ = \operatorname{argmin} \left\{ \|y_\circ - \varphi\|_H \;\middle|\; \varphi \in V^h \right\}. \tag{1.4}$$

Inserting (1.2) into (1.3) we get the finite element (FE) problem

$$\begin{aligned} \mathrm{M}\dot{\mathrm{y}}^h(t) + \mathrm{A}\mathrm{y}^h(t) &= \mathrm{f} \qquad \text{f.a.a. } t \in [0,T], \\ \mathrm{M}\mathrm{y}^h(0) &= \mathrm{y}_\circ^h, \end{aligned} \tag{1.5}$$

where $\mathrm{M} = ((\langle \varphi_j, \varphi_i\rangle_H)) \in \mathbb{R}^{N\times N}$, $\mathrm{A} = ((a(\varphi_j, \varphi_i))) \in \mathbb{R}^{N\times N}$, $\mathrm{f} = (\langle f, \varphi_i\rangle_H) \in \mathbb{R}^N$, and $\mathrm{y}_\circ^h = (\langle y_\circ, \varphi_i\rangle_H) \in \mathbb{R}^N$.

In order to discretize in time – namely, fully discretize – system (1.5), we first define a uniform time grid $t_k = k\delta$, for $k = 0, \dots, K$ and $\delta = T/K$. Then, we use a θ-scheme [52, Section 5.1], where a generic dynamical system

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\langle u(t), v\rangle_H + \mathcal{A}(u(t), v) &= \mathcal{F}(t, v) \qquad \forall v \in V, \text{ f.a.a. } t \in (0,T], \\ u(0) &= u_\circ, \end{aligned}$$

is discretized for $k = 1, \dots, K$ and $\theta \in [0,1]$ as

$$\frac{1}{\delta}\langle u^k - u^{k-1}, v\rangle_H + \mathcal{A}(\theta u^k + (1-\theta)u^{k-1}, v) = \theta\mathcal{F}(t_k, v) + (1-\theta)\mathcal{F}(t_{k-1}, v)$$

and $u^0 = u_\circ$. Among θ-methods, we use implicit Euler (IE) by choosing $\theta = 1$. Thus, (1.5) is discretized as finding $\mathrm{y}^{h,0}, \dots, \mathrm{y}^{h,K} \in \mathbb{R}^N$ solutions of

$$\begin{aligned} (\mathrm{M} + \delta\mathrm{A})\mathrm{y}^{h,k} &= \mathrm{M}\mathrm{y}^{h,k-1} + \delta\mathrm{f}, \qquad k = 1, \dots, K, \\ \mathrm{M}\mathrm{y}^{h,0} &= \mathrm{y}_\circ^h. \end{aligned} \tag{1.6}$$

We call system (1.6) full order (FO) model for (1.1). Then, $\mathrm{y}^{h,k} \in \mathbb{R}^N$ is the coefficient vector of $y^{h,k} \in V^h$, i.e.

$$y^{h,k} = \sum_{j=1}^{N} \mathrm{y}_j^{h,k} \varphi_j \in V^h \qquad \text{for } k = 0, \dots, K,$$

and the FO solution $y^{h,k}$ is an approximation of the FE solution $y(t_k) \in V$ solution of (1.1), and therefore we use FO solution and FO approximation interchangeably. We furthermore assume that the FE model is sufficiently accurate and sufficiently close to the exact solution with a reasonable tolerance, i.e. $\|y^h - y\|_{L^2(0,T;V)} < \varepsilon^{\mathsf{FE}}$.

1.2.2 Newton's method

In Chapter 3 we deal with a nonlinear problem, whose full discretization is not a linear system as in (1.6). On the opposite, let us assume a semi-discretized system of the form

$$\begin{aligned} \mathrm{M}\dot{\mathrm{y}}^h(t) + \mathrm{A}\mathrm{y}^h(t) &= \mathrm{M}f(\mathrm{y}^h(t)) \qquad \text{f.a.a. } t \in [0,T], \\ \mathrm{M}\mathrm{y}^h(0) &= \mathrm{y}_\circ^h \in \mathbb{R}^N, \end{aligned} \tag{1.7}$$

where f is scalar nonlinear function applied component-wise, namely

$$f(\mathrm{y}^h(t)) = [f(\mathrm{y}_1^h(t)), \dots, f(\mathrm{y}_N^h(t))]^\top \in \mathbb{R}^N.$$

After discretization (Implicit Euler) in time, the time stepping of (1.7) becomes, for $k = 1, \dots, K$

$$(\mathrm{M} + \delta\mathrm{A})\mathrm{y}^{h,k} = \delta\mathrm{M}f(\mathrm{y}^{h,k}) + \mathrm{M}\mathrm{y}^{h,k-1}, \tag{1.8}$$

or, written as an implicit problem,

$$\mathrm{F}^k(\mathrm{y}^{h,k}) = 0. \tag{1.9}$$

To solve (1.9) we use Newton's method described in Algorithm 1, which finds the root using the Jacobian information.

Algorithm 1: Newton's method

Input: Initial point $\mathrm{x}_0 \in \mathbb{R}^n$, function $\mathrm{f} : \mathbb{R}^n \to \mathbb{R}^n$, Jacobian matrix $\mathrm{J_f} : \mathbb{R}^n \to \mathbb{R}^{n \times n}$;

1 Set $k = 0$;
2 **while** $\|\mathrm{f}(\mathrm{x}_k)\|_2 > \tau_{\mathsf{Newton}}$ **do**
3 Find $\mathrm{d} \in \mathbb{R}^n$ solution of $\mathrm{J_f}(\mathrm{x}_k)\mathrm{d} = -\mathrm{f}(\mathrm{x}_k)$;
4 Generate new point $\mathrm{x}_{k+1} := \mathrm{x}_k + \mathrm{d}$;
5 Set $k = k + 1$;

Let us also use a globalization strategy for Newton's method, namely the damped Newton method, which is essentially a combination of the standard Newton method and the Armijo rule, a line search technique, in order to guarantee monotonicity in the residual decrease. While the globalization strategy is a good safeguard, its use slows the performance, although not significantly. We omit the algorithm and the discussion for simplicity; more details are available in [15].

1.2.3 Reduced Order Modeling

In order to make calculations faster, we construct a reduced order (RO) space V^ℓ of dimension $L \ll N$ and a basis $\{\psi_1, \dots, \psi_L\} \subset V^\ell \subset V^h$, whose construction will be explained later. We define the coefficient matrix $\Psi \in \mathbb{R}^{N\times L}$ satisfying $\psi_j = \sum_{i=1}^N \Psi_{ij}\varphi_i$ for $j = 1, \dots, L$. The RO model for (1.6) is given by the Galerkin projection and it reads as follows: find $\{\hat{\mathrm{y}}^1, \dots, \hat{\mathrm{y}}^K\} \subset \mathbb{R}^L$ such that

$$\begin{aligned} (\mathrm{M}^\ell + \delta \mathrm{A}^\ell)\hat{\mathrm{y}}^k &= \mathrm{M}^\ell \hat{\mathrm{y}}^{k-1} + \delta \mathrm{f}, \qquad k = 1, \dots, K, \\ \mathrm{M}^\ell \hat{\mathrm{y}}^0 &= \mathrm{y}_\circ^\ell, \end{aligned} \tag{1.10}$$

where the RO arrays $\mathrm{M}^\ell \in \mathbb{R}^{L\times L}$, $\mathrm{A}^\ell \in \mathbb{R}^{L\times L}$, $\mathrm{f}^\ell \in \mathbb{R}^L$ and $\mathrm{y}_\circ^\ell \in \mathbb{R}^L$ are defined as

$$\begin{aligned} &\mathrm{M}^\ell := \Psi^\top \mathrm{M}\Psi, && \mathrm{A}^\ell := \Psi^\top \mathrm{A}\Psi, \\ &\mathrm{f}^\ell := \Psi^\top \mathrm{f} \text{ for } k = 1, \dots, K, && \mathrm{y}_\circ^\ell := \Psi^\top \mathrm{y}_\circ^h. \end{aligned}$$

The solution $\{\hat{\mathrm{y}}^k\}_{k=0}^K$ of (1.10) exists uniquely (see [35]) and it is interpreted as a reduced order approximation for $\{\mathrm{y}^k\}_{k=0}^K$:

$$\mathrm{y}^k \approx \mathrm{y}^{\ell,k} := \Psi \hat{\mathrm{y}}^k \in \mathbb{R}^N.$$

For $0 \le k \le K$ we will also use the additional notation

$$y^{\ell,k} := \sum_{i=1}^L \hat{\mathrm{y}}_i^k \psi_i = \sum_{i=1}^N \mathrm{y}_i^{\ell,k} \varphi_i \in V^\ell.$$

1.2.4 Proper Orthogonal Decomposition (POD)

To select an RO space we use the method of *proper orthogonal decomposition* (POD), that uses the state variable snapshots $\{\mathrm{y}^{h,0}, \dots, \mathrm{y}^{h,K}\} \subset \mathbb{R}^N$, or equivalently $\{y^{h,0}, \dots, y^{h,K}\} \subset V^h$, to define the RO bases $\{\psi_j\}_{j=1}^L$ satisfying the problem

$$\begin{aligned} &\min_{\psi_1,\dots,\psi_L \in V} \int_0^T \Big\| y^h(t) - \sum_{i=1}^L \langle y^h(t), \psi_j \rangle_V \, \psi_j \Big\|_V^2 \\ &\text{subject to } \langle \psi_i, \psi_j \rangle_V = \delta_{ij} \qquad \forall\, 1 \le i, j \le L. \end{aligned}$$

or, rather, its discretized version

$$\begin{aligned} &\min_{\psi_1,\dots,\psi_L \in V} \sum_{k=0}^K \alpha_k \Big\| y^{h,k} - \sum_{i=1}^L \langle y^{h,k}, \psi_j \rangle_V \, \psi_j \Big\|_V^2 \\ &\text{subject to } \langle \psi_i, \psi_j \rangle_V = \delta_{ij} \qquad \forall\, 1 \le i, j \le L, \end{aligned} \tag{1.11}$$

where $\alpha_0, \dots, \alpha_K$ are the trapezoidal weights used to approximate the integral, i.e. $\alpha_0 = \alpha_K = \delta/2$ and $\alpha_k = \delta$ for $k = 1, \dots, K-1$.

We will discuss the discretized version of the POD only; all the details regarding the continuous version can be found, e.g., in [27]. A solution to (1.11) is given by evaluating

the eigenvectors corresponding to the L biggest eigenvalues of the matrix $\mathrm{R} = \mathrm{Y}\mathrm{D}\mathrm{Y}^\top\mathrm{W}$, where

$$\mathrm{Y} = [\mathrm{y}^{h,0}, \dots, \mathrm{y}^{h,K}] \in \mathbb{R}^{N\times(K+1)}$$

is the matrix of snapshots, D is the diagonal matrix of the trapezoidal weights, namely

$$\mathrm{D} = \begin{pmatrix} \alpha_0 & & \\ & \ddots & \\ & & \alpha_K \end{pmatrix},$$

and $\mathrm{W} = ((\langle \varphi_j, \varphi_i \rangle_V)) \in \mathbb{R}^{N\times N}$.

As discussed in [36], there are numerous ways of finding the bases, and consequently Ψ. Assuming $K < N$, we discuss the two main approaches, one being more accurate and the other faster in computation time. We start by defining

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

The square root of W is well-defined since it is symmetric and positive definite, while the square root of diagonal matrices are merely the diagonal matrices with element-wise square roots. In particular, it is possible to decompose W using a diagonal matrix Λ and an orthogonal matrix Q as $\mathrm{W} = \mathrm{Q}\Lambda\mathrm{Q}^\top$, and in this case we can write $\mathrm{W}^{1/2} := \mathrm{Q}\Lambda^{1/2}\mathrm{Q}^\top$.

The matrix $\bar{\mathrm{Y}}$ will appear in both ways, but only in one of them it will be necessary to actually evaluate $\mathrm{W}^{1/2}$.

The method of snapshots In this case we define the symmetric $(K+1)\times(K+1)$ matrix

$$\bar{\mathrm{R}} := \bar{\mathrm{Y}}^\top\bar{\mathrm{Y}}.$$

Then, the RO basis is given by solving the eigenvalue problem

$$\begin{aligned} &\bar{\mathrm{R}}\mathrm{v}_i = \lambda_i \mathrm{v}_i, && 1 \le i \le L, \\ &\text{s.t. } \langle \mathrm{v}_i, \mathrm{v}_j \rangle_{\mathbb{R}^N} = \delta_{ij}, && 1 \le i, j \le L, \end{aligned}$$

and then defining the vectors

$$\Psi_i := \frac{1}{\sqrt{\lambda_i}}\mathrm{Y}\mathrm{D}^{1/2}\mathrm{v}_i \in \mathbb{R}^N.$$

These will be the coordinate vectors of the new basis w.r.t. the old one, or equivalently the columns of Ψ. Let us observe that the evaluation of $\bar{\mathrm{R}} = \mathrm{D}^{1/2}\mathrm{Y}^\top\mathrm{W}\mathrm{Y}\mathrm{D}^{1/2}$ only involves the cheap evaluation of square roots of diagonal matrices. This method is faster, but it can be inaccurate when evaluating, e.g., the smallest eigenvalues of $\bar{\mathrm{R}}$, and it can be subject to numerical errors.

The SVD method In this case we compute the singular value decomposition for $\bar{\mathrm{Y}}$, i.e.

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

for the orthogonal matrix $\mathrm{U} = [\mathrm{u}_1, \ldots, \mathrm{u}_N]$, then set $\Psi_i := \mathrm{W}^{-1/2}\mathrm{u}_i$ and $\lambda_i := \sigma_i^2$ for $1 \le i \le L$. Observe that this means solving L linear problems, i.e. $\mathrm{W}^{1/2}\Psi_i = \mathrm{u}_i$, and hence it will be slower, but it holds in general a better accuracy. In this work we have preferred this method: the slowdown is not prohibitive and we prefer keeping the better numerical results.

On the quality of the reduced basis we have the following error formula (cf., e.g., [35]):

$$\sum_{k=0}^{K} \alpha_k \left\| y^{h,k} - \sum_{i=1}^{L} \langle y^{h,k}, \psi_i \rangle_V \ \psi_i \right\|_V^2 = \sum_{j=L+1}^{K} \lambda_i. \tag{1.12}$$

As discussed in [35] and more recently in [18], we also modify the set of snapshots to improve the RO approximation of the time derivative of the state variable. Let us define the finite differences

$$\bar{\partial} y^{h,k} := \frac{y^{h,k} - y^{h,k-1}}{\delta} \qquad k = 1, \ldots, K,$$

and let us consider the problem

$$\begin{aligned} &\min_{\{\psi_j\}_{j=1}^{L}} \sum_{k=0}^{K} \alpha_k \left\| y^{h,k} - \sum_{i=1}^{L} \langle y^{h,k}, \psi_i \rangle_V \ \psi_i \right\|_V^2 + \sum_{k=0}^{K} \delta \left\| \bar{\partial} y^{h,k} - \sum_{i=1}^{L} \langle \bar{\partial} y^{h,k}, \psi_i \rangle_V \ \psi_i \right\|_V^2 \\ &\text{s.t.} \langle \psi_\mathrm{i}, \psi_\mathrm{j} \rangle_\mathrm{V} = \delta_\mathrm{ij} \qquad \forall\ 1 \le \mathrm{i}, \mathrm{j} \le \mathrm{L}. \end{aligned} \tag{1.13}$$

Problem (1.13) is solved using the eigenvalues and eigenvectors of the matrix $\mathrm{R} = \mathrm{Z}\tilde{\mathrm{D}}\mathrm{Z}^\top \mathrm{W}$, where now the snapshot matrix is

$$\mathrm{Z} := [\mathrm{y}^0, \ldots, \mathrm{y}^K, \bar{\partial}\mathrm{y}^1, \ldots, \bar{\partial}\mathrm{y}^K] \in \mathbb{R}^{N \times (2K+1)}, \tag{1.14}$$

we set

$$\tilde{\mathrm{D}} := \begin{pmatrix} \alpha_0 & & & \\ & \ddots & & \\ & & \alpha_K & \\ & & & \delta \mathbb{I}_K \end{pmatrix} \in \mathbb{R}^{(2K-1)\times(2K-1)} \tag{1.15}$$

and $\bar{\partial}\mathrm{y}^{h,k} := (\mathrm{y}^{h,k} - \mathrm{y}^{h,k-1})/\delta$. Similarly to (1.12), in this setting we get

$$\sum_{k=0}^{K} \alpha_k \left\| y^{h,k} - \sum_{i=1}^{L} \langle y^{h,k}, \psi_i \rangle_V \ \psi_i \right\|_V^2 + \sum_{k=0}^{K} \delta \left\| \bar{\partial} y^{h,k} - \sum_{i=1}^{L} \langle \bar{\partial} y^{h,k}, \psi_i \rangle_V \ \psi_i \right\|_V^2 = \sum_{j=L+1}^{K} \lambda_i.$$

1.2.5 Discrete empirical interpolation method

The discrete empirical interpolation method (DEIM) is a dimension reduction method introduced in [12] that approximates the nonlinearity in a reduced dimension subspace. It was formulated as a modification of the empirical interpolation method (EIM), (for more details, see [8]). In the following, we review the main steps of the algorithm; for more details, see [12, 11].

Let us go back to our N-dimensional nonlinear time-stepping equation (1.8). Let us assume we have found a suitable L-dimensional RO basis associated with a matrix $\Psi \in \mathbb{R}^{N\times L}$. Then, by standard Galerkin projection, we have for $k = 1, \dots, K$

$$(\mathrm{M}^\ell + \delta \mathrm{A}^\ell)\mathrm{y}^{h,k} = \delta \Psi^\top \mathrm{M} f(\Psi \mathrm{y}^{h,k}) + \mathrm{M}^\ell \mathrm{y}^{h,k}. \tag{1.16}$$

Equation (1.16) relies on the RO dimension L, but it also depends on N, hence making the evaluation of $\mathrm{y}^{h,k}$ not *online efficient*.

The DEIM projects the vector $\mathrm{f}^k := f(\Psi \mathrm{y}^{h,k}) \in \mathbb{R}^N$ onto a smaller space of dimension M. Assuming this space is spanned by the basis $\{\mathrm{u}_1, \dots, \mathrm{u}_M\}$, and defining the matrix $\mathrm{U} \in \mathbb{R}^{N\times M}$ of the coordinate vectors of such basis, the approximation is then

$$\mathrm{f}^k \approx \mathrm{U}\mathrm{c}^k, \tag{1.17}$$

where $\mathrm{c}^k \in \mathbb{R}^M$. Pre-evaluating and storing the matrix $\mathrm{M}_\mathrm{f}^\ell := \Psi^\top \mathrm{M}\mathrm{U} \in \mathbb{R}^{L\times M}$, if we can evaluate c^k in a cheap way, then we can also cheaply compute

$$(\mathrm{M}^\ell + \delta \mathrm{A}^\ell)\mathrm{y}^{h,k} = \delta \mathrm{M}_\mathrm{f}^\ell \mathrm{c}^k + \mathrm{M}^\ell \mathrm{y}^{h,k},$$

and the evaluation becomes fully online efficient. Let us start with some notation. The i-th Euclidean basis element in $\mathbb{R}^N$ is denoted by the vector $\mathrm{e}_i = [0, \dots, 0, 1, 0, \dots, 0]^\top \in \mathbb{R}^N$, which is also necessarily the i-th column of the identity matrix $\mathbb{I}_N$. Moreover, let $\mathcal{I} = [i_1, \dots, i_M]$ be a list of M indices, and $\mathrm{v} \in \mathbb{R}^N$ be a vector. We define $\mathrm{v}_\mathcal{I}$ as the vector given by extracting the indices corresponding to $\mathcal{I}$, i.e.

$$\mathrm{v}_\mathcal{I} := [\mathrm{v}_{i_1}, \dots, \mathrm{v}_{i_M}]^\top \in \mathbb{R}^M.$$

If we introduce the matrix $\mathrm{I} = \mathrm{I}_\mathcal{I} \in \mathbb{R}^{N\times M}$ associated with list $\mathcal{I}$ as

$$\mathrm{I} = [\mathrm{e}_{i_1} \ \dots \ \mathrm{e}_{i_M}] \qquad \text{for } \mathcal{I} = [i_1, \dots, i_M],$$

then it holds $\mathrm{v}_\mathcal{I} = \mathrm{I}^\top \mathrm{v}$. By assuming the equation (1.17) correct on M distinct rows, and indexing such rows as $\mathcal{I} = [i_1, \dots, i_M]$, we can write

$$\mathrm{I}^\top \mathrm{f}^k = (\mathrm{I}^\top \mathrm{U})\mathrm{c}^k.$$

Assuming, furthermore, that $\mathrm{I}^\top \mathrm{U}$ is invertible, we can then find c^k as

$$\mathrm{c}^k = (\mathrm{I}^\top \mathrm{U})^{-1} \mathrm{I}^\top \mathrm{f}^k. \tag{1.18}$$

Inserting (1.18) into (1.17) we get

$$\mathrm{f}^k \approx \mathrm{U}(\mathrm{I}^\top \mathrm{U})^{-1} \mathrm{I}^\top \mathrm{f}^k. \tag{1.19}$$

Let us observe that since the linearity is component-wise, then

$$\mathrm{I}^\top \mathrm{f}^k = \mathrm{I}^\top f(\Psi \mathrm{y}^{h,k}) = f(\mathrm{I}^\top \Psi \mathrm{y}^{h,k}).$$

Hence, by pre-evaluating and storing the matrices $\hat{\mathrm{M}}_{\mathrm{f}}^{\ell} := \mathrm{M}_{\mathrm{f}}^{\ell}(\mathrm{I}^\top \mathrm{U})^{-1} = \Psi^\top \mathrm{M}\mathrm{U}(\mathrm{I}^\top \mathrm{U})^{-1} \in \mathbb{R}^{L\times M}$ and $\hat{\Psi} := \mathrm{I}^\top \Psi \in \mathbb{R}^{M\times L}$, we can cheaply solve the online-efficient equation

$$(\mathrm{M}^{\ell} + \delta \mathrm{A}^{\ell})\mathrm{y}^{h,k} = \delta \hat{\mathrm{M}}_{\mathrm{f}}^{\ell} f(\hat{\Psi}\mathrm{y}^{h,k}) + \mathrm{M}^{\ell}\mathrm{y}^{h,k}.$$

In Algorithm 2 we transcribe the algorithm from [12] to evaluate the index matrix I and the matrix U.

Algorithm 2: Algorithm for DEIM

Input: Snapshots $\mathrm{f}^1, \ldots, \mathrm{f}^K \in \mathbb{R}^N$;

Evaluate M linearly independent vectors $\mathrm{u}_1, \ldots \mathrm{u}_M \in \mathbb{R}^N$ using POD;
$i_1 = \arg\max_{j=1,\ldots,N} |(\mathrm{u}_1)_j|$;
$\mathrm{U} = [\mathrm{u}_1]$;
$\mathrm{I} = [\mathrm{e}_{i_1}]$;
for $k = 2$ **to** m **do**
    Solve $(\mathrm{I}^\top \mathrm{U})\,\mathrm{c} = \mathrm{I}^\top \mathrm{u}_k$ for c;
    $\mathrm{r} = \mathrm{u}_k - \mathrm{U}\mathrm{c}$;
    $i_k = \arg\max_{j=1,\ldots,N} |\mathrm{r}_j|$;
    $\mathrm{U} = [\mathrm{U}, \mathrm{u}_k]$;
    $\mathrm{I} = [\mathrm{I}, \mathrm{e}_{i_k}]$;

Chapter 2

Linear single-state model

Let us start from a simplified model that can be studied in detail mathematically. This chapter serves as an introduction to the next, since many of the ideas discussed here will be expanded and applied to a more complex model.

In the introduction we have mentioned different battery models studied over the decades; they all stem from physical modeling, where some assumptions and simplifications can be made in order to change the system of equations. For example, electrodes can be assumed to have a certain shape, or some physical quantities (e.g. diffusion coefficients) can be modeled constant. One of the most common assumptions is the approximation of the electrodes as homogeneously distributed spherical particles, which causes the dynamics to become decoupled one-dimensional equations, as in the renowned DFN model (see [17]). For this reason, we are also going to consider evolution problems in the one-dimensional domain $\Omega := (0, L)$.

Let $T > 0$ be the (finite) time horizon, and $Q_T := (0, T) \times (0, L)$. Battery cell models are equations describing the movements of electrons and the balancing of potential energies, and can be over-simplified, e.g., as a linear scalar reaction-diffusion system

$$\begin{aligned} y_t(t, x) &= D y_{xx}(t, x) + R y(t, x), \\ y_x(t, 0) &= 0, \quad D y_x(t, L) = u(t), \\ y(0, x) &= y_\circ(x), \end{aligned} \tag{2.1}$$

where $D, R > 0$ are positive constant (called, respectively, diffusion constant and reaction constant), $u : [0, T] \to \mathbb{R}$ is an arbitrary control and $y_\circ : \Omega \to \mathbb{R}$ is the initial condition. It is well known that this system models the change in space and time of the concentration of some physical substance and can be found in several problems, e.g. in chemistry, biology and geology.

We can generalize this system to a class of linear abstract parabolic evolution problem in one dimension in space. Furthermore, let us assume that all d scalar parameters are stacked in a parameter vector $\boldsymbol{\mu} \in \mathbb{R}^d$, and that the latter belongs to the nonempty, compact and convex admissible parameter set

$$\mathcal{P}_{\mathsf{ad}} := \{\boldsymbol{\mu} = (\mu_1, \dots, \mu_d) \in \mathcal{P} : \boldsymbol{\mu}_{\mathsf{a}} \leq \boldsymbol{\mu} \leq \boldsymbol{\mu}_{\mathsf{b}} \text{ in } \mathcal{P}\} \subset \mathcal{P} := \mathbb{R}^d,$$

where $\boldsymbol{\mu}_{\mathsf{a}} = (\mu_{\mathsf{a},i})_{1\le i\le 2}$ and $\boldsymbol{\mu}_{\mathsf{b}} = (\mu_{\mathsf{b},i})_{1\le i\le 2}$ are given parameter bounds satisfying $0 < \boldsymbol{\mu}_{\mathsf{a}} \le \boldsymbol{\mu}_{\mathsf{b}}$.

The *state* variable y satisfies the abstract parabolic evolution problem in weak form

$$\begin{aligned} \langle y_t(t), \varphi\rangle_{V^*,V} + a_{\boldsymbol{\mu}}(y(t), \varphi) &= \langle f_{\boldsymbol{\mu}}(t; u), \varphi\rangle_{V^*,V} \\ y(0) &= y_\circ \quad \text{in } H \end{aligned} \tag{2.2}$$

for all $\varphi \in V$ and f.a.a. $t \in (0, T]$, where $y_\circ \in H$ and V, H are separable Hilbert spaces with $V \hookrightarrow H \hookrightarrow V^*$ (Gelfand triple). The *input*, or *control*, function u belongs to a convex, bounded, closed subset $\mathcal{U}_{\mathsf{ad}}$ of a Hilbert space $\mathcal{U}$. Further, for any $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$ the bilinear form $a_{\boldsymbol{\mu}} : V \times V \to \mathbb{R}$ is

- symmetric:

$$a_{\boldsymbol{\mu}}(\varphi, \phi) = a_{\boldsymbol{\mu}}(\phi, \varphi) \quad \text{for all } \varphi, \phi \in V;$$

- continuous: there exists a constant $\overline{\gamma}_{\boldsymbol{\mu}} > 0$ satisfying

$$|a_{\boldsymbol{\mu}}(\varphi, \phi)| \le \overline{\gamma}_{\boldsymbol{\mu}} \|\varphi\|_V \|\phi\|_V \quad \text{for all } \varphi, \phi \in V;$$

- coercive: there is $\underline{\alpha}_{\boldsymbol{\mu}} > 0$ such that

$$a_{\boldsymbol{\mu}}(\varphi, \varphi) \ge \underline{\alpha}_{\boldsymbol{\mu}} \|\varphi\|_V^2 \quad \text{for all } \varphi \in V. \tag{2.3}$$

Moreover, $f_{\boldsymbol{\mu}}(\cdot\,; u) \in L^2(0; T; V^*)$ holds for any $(\boldsymbol{\mu}, u) \in \mathcal{X}_{\mathsf{ad}} = \mathcal{P}_{\mathsf{ad}} \times \mathcal{U}_{\mathsf{ad}}$. It follows that (2.2) admits a unique solution $y = y_{\boldsymbol{\mu}} \in \mathcal{Y} = W(0, T; V, V^*) = L^2(0, T; V) \cap H^1(0, T; V^*)$ for any $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$; cf., e.g., [31, 64]. Furthermore, we assume that $a_{\boldsymbol{\mu}}$ and $f_{\boldsymbol{\mu}}$ depend affinely on the parameters:

$$a_{\boldsymbol{\mu}} = \sum_{l=1}^{m_a} \vartheta_l^a(\boldsymbol{\mu}) \hat{a}_l, \quad f_{\boldsymbol{\mu}}(t; u) = \sum_{l=1}^{m_f} \vartheta_l^f(\boldsymbol{\mu}) \hat{f}_l(t; u) \tag{2.4}$$

for any $(\boldsymbol{\mu}, u) \in \mathcal{X}_{\mathsf{ad}}$ and $t \in [0, T]$. Otherwise, we apply the *empirical interpolation method* to get approximations satisfying (2.4); cf., e.g., [8, 12, 11, 30]. Moreover, we assume that ϑ_l^a and ϑ_l^f are continuous and continuously differentiable for $l = 1, \ldots, m_a$ and $l = 1, \ldots, m_f$, respectively, to ensure that the maps $\boldsymbol{\mu} \to a_{\boldsymbol{\mu}}$ and $\boldsymbol{\mu} \to f_{\boldsymbol{\mu}}$ are continuous and differentiable. Lastly, we also assume the bilinear forms $\hat{a}_l$ to be continuous for all $l = 1, \ldots, m_a$ and all linear operators $\hat{f}_l(t; u)$ to be continuous for all $l = 1, \ldots, m_f$, $t \in [0, T]$ a.e. and all $u \in \mathcal{U}_{\mathsf{ad}}$.

Remark 2.1. The model (2.2) comprehends the reaction-diffusion equation (2.1), with

$$a_{\boldsymbol{\mu}}(y(t), \varphi) = D\langle y_x(t), \varphi_x\rangle_H - R\langle y(t), \varphi\rangle_H,$$

with parameters $\boldsymbol{\mu} = (D, R) \in \mathcal{P}_{\mathsf{ad}}$, and

$$\langle f_{\boldsymbol{\mu}}(t; u), \varphi\rangle_{V^*,V} = u(t)\varphi(L).$$

$\Diamond$

This chapter is organized as follows: in Section 2.1 we introduce the PDE-constrained optimization, define the cost function and the adjoint equation. The finite element discretization is done in Section 2.2, and in Section 2.3 we define the reduced order approximation. In Section 2.4, finally, we briefly describe the trust region optimization problem used in this work.

2.1 PDE-constrained optimization problem

In the context of data-driven parameter estimation, we assume that the model parameters $\boldsymbol{\mu}$ are unknown and consider the optimization problem

$$\min J(y, \boldsymbol{\mu}) \quad \text{subject to } (s.t.) \quad (y, \boldsymbol{\mu}) \in \mathcal{Y} \times \mathcal{P}_{\mathsf{ad}} \text{ satisfying (2.2)}, \tag{P$_1$}$$

where J is a cost function quantifying the discrepancy between the state variable y and some data $w \in L^2(0,T;H)$ (the data will depend on the input u, namely $w = w(u)$):

$$J(y, \boldsymbol{\mu}) = \frac{\alpha_J}{2} \int_0^T \|y - w\|_H^2 \,\mathrm{d}t + \frac{\beta_J}{2} \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2^2, \tag{2.5}$$

where $\|\cdot\|_2$ stands for the Euclidean norm, β_J is a non-negative weight and $\hat{\boldsymbol{\mu}} \in \mathbb{R}^d$ is a reference parameter.

As (2.2) is uniquely solvable, we can define the *reduced cost*

$$\hat{J}(\boldsymbol{\mu}) = J(y_{\boldsymbol{\mu}}, \boldsymbol{\mu}) = \frac{\alpha_J}{2} \int_0^T \|y_{\boldsymbol{\mu}} - w\|_H^2 \,\mathrm{d}t + \frac{\beta_J}{2} \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2^2, \tag{2.6}$$

where $y_{\boldsymbol{\mu}}$ solves (2.2). Then, (P$_1$) is equivalent to

$$\min \hat{J}(\boldsymbol{\mu}) \quad \text{subject to (s.t.)} \quad \boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}. \tag{$\hat{\text{P}}_1$}$$

Existence of an optimal solution $\bar{\boldsymbol{\mu}}$ is proven in Section A.1 of the appendix, but – due to non-convexity – there are possibly many local solutions; cf., e.g., [31]. To characterize the local optimal solutions of ($\hat{\text{P}}_1$), we use first-order necessary optimality conditions, from which we derive the *adjoint variable* $\bar{p} = p_{\bar{\boldsymbol{\mu}}} \in \mathcal{Y}$ associated with the optimal parameter $\bar{\boldsymbol{\mu}}$ and state $\bar{y} = y_{\bar{\boldsymbol{\mu}}}$. The adjoint variable solves the *adjoint equation* in weak form

$$\begin{aligned} \langle p_t(t), \varphi \rangle_{V^*,V} + a_{\boldsymbol{\mu}}(p(t), \varphi) &= \langle w(t) - y_{\boldsymbol{\mu}}(t), \varphi \rangle_H \\ p(T) &= 0, \quad \text{in } H. \end{aligned}$$

for all $\varphi \in V$ and f.a.a. $t \in [0,T)$. We skip the details regarding the derivation of the adjoint equation, but they can be found, e.g., in [64, Chapter 3]. Then, following the *adjoint approach* in [31, Section 1.6.2], the gradient of the cost functional at $\bar{\boldsymbol{\mu}}$ is given as

$$\nabla \hat{J}(\bar{\boldsymbol{\mu}}) = \beta_J(\bar{\boldsymbol{\mu}} - \hat{\boldsymbol{\mu}}) + \int_0^T \sum_{l=1}^{m_a} \nabla \vartheta_l^a(\bar{\boldsymbol{\mu}}) \hat{a}_l\left(\bar{y}(t), \bar{p}(t)\right) \mathrm{d}t - \int_0^T \sum_{l=1}^{m_f} \nabla \vartheta_l^f(\bar{\boldsymbol{\mu}}) \langle \hat{f}_l(t;u), \bar{p}(t) \rangle_{V^*,V} \,\mathrm{d}t.$$

2.2 Finite element discretization

In this section we introduce a high-dimensional approximation, called *full-order* (FO) approximation, based on the finite element (FE) theory, which we assume to be accurate enough, but however expensive to solve. To reduce significantly the computational cost a further approximation is later applied, the *reduced order* approximation, faster to solve but less accurate.

The FO discretization is overview in Section 1.2.1. We can go ahead and define the FO system, where we find $\mathrm{y}_{\boldsymbol{\mu}}^{h,0}, \dots, \mathrm{y}_{\boldsymbol{\mu}}^{h,K} \in \mathbb{R}^N$ solutions of

$$\begin{aligned} (\mathrm{M} + \delta \mathrm{A}_{\boldsymbol{\mu}})\mathrm{y}_{\boldsymbol{\mu}}^{h,k} &= \mathrm{M}\mathrm{y}_{\boldsymbol{\mu}}^{h,k-1} + \delta \mathrm{f}_{\boldsymbol{\mu}}^{h,k}, \qquad k = 1, \dots, K, \\ \mathrm{M}\mathrm{y}_{\boldsymbol{\mu}}^{h,0} &= \mathrm{y}_{\circ}^{h}, \end{aligned} \tag{2.7}$$

where, using the FE space V^h and basis $\{\varphi_1, \dots, \varphi_N\}$ we have used

$$\begin{aligned} &\mathrm{M} \in \mathbb{R}^{N \times N}, && (\mathrm{M})_{i,j} := \langle \varphi_j, \varphi_i \rangle_H, \\ &\mathrm{A}_{\boldsymbol{\mu}} \in \mathbb{R}^{N \times N}, && (\mathrm{A}_{\boldsymbol{\mu}})_{i,j} := a_{\boldsymbol{\mu}}(\varphi_j, \varphi_i), \\ &\mathrm{f}_{\boldsymbol{\mu}}^{h,k} \in \mathbb{R}^N, && (\mathrm{f}_{\boldsymbol{\mu}}^{h,k})_j := \langle f_{\boldsymbol{\mu}}(t_k; u), \varphi_j \rangle_{V^*,V}, \\ &\mathrm{y}_{\circ}^{h} \in \mathbb{R}^N, && (\mathrm{y}_{\circ}^{h})_j := \langle y_{\circ}, \varphi_j \rangle_H. \end{aligned}$$

Then, for $k = 0, \dots, K$, $\mathrm{y}_{\boldsymbol{\mu}}^{h,k} \in \mathbb{R}^N$ is the coefficient vector of $y_{\boldsymbol{\mu}}^{h,k} \in V^h$, i.e.,

$$y_{\boldsymbol{\mu}}^{h,k} = \sum_{j=1}^{N} \mathrm{y}_{\boldsymbol{\mu} i}^{h,k} \varphi_i \in V^h \qquad \text{for } k = 0, \dots, K, \tag{2.8}$$

and $y_{\boldsymbol{\mu}}^{h,k}$ is an approximation of $\mathrm{y}_{\boldsymbol{\mu}}^{h}(t_k) \in V$.

Remark 2.2. Due to (2.4), both $\mathrm{A}_{\boldsymbol{\mu}}$ and $\mathrm{f}_{\boldsymbol{\mu}}$ satisfy

$$\mathrm{A}_{\boldsymbol{\mu}} = \sum_{l=1}^{m_a} \vartheta_l^a(\boldsymbol{\mu}) \hat{\mathrm{A}}_l^h, \quad \mathrm{f}_{\boldsymbol{\mu}}^{h,k} = \sum_{l=1}^{m_f} \vartheta_l^f(\boldsymbol{\mu}) \hat{\mathrm{f}}_l^{h,k}$$

for the FE arrays $\hat{\mathrm{A}}_l^h := ((\hat{a}_l(\varphi_j, \varphi_i)))$ and $\hat{\mathrm{f}}_l^{h,k} := (\langle \hat{f}_l(t_k; u), \varphi_i \rangle_{V^*;V})$, which do not depend on $\boldsymbol{\mu}$. ◊

Example 2.3. Let $\Omega = (0, 4)$ and $T = 1$. We use linear Lagrangian elements on 401 nodes and 201 time steps, so that $\delta = 0.005$. Let us further choose

$$a_{\boldsymbol{\mu}}(y(t), \varphi) := \mu_1 \langle y_x(t), \varphi' \rangle_H + \frac{1}{2} \langle y(t), \varphi \rangle_H$$

and

$$\langle f(t; u), \varphi \rangle_{V^*,V} := \mu_2 u(t) \varphi(L),$$

so that we are solving the parabolic problem

$$\begin{aligned} &y_t(t, x) - \mu_1 y_{xx}(t, x) + \frac{1}{2} y(t, x) = 0, \\ &y_x(t, 0) = 0, \qquad \mu_1 y_x(t, L) = \mu_2 u(t), \\ &y(0, x) = y_{\circ}(x), \end{aligned}$$

Finally, we set $u(t) := \frac{1}{2}\cos(10t)$ and $y_\circ(x) := 5+e^{-10|x-0.5|^2}$. In Figure 2.1 it possible to see the FO approximations for different parameters, namely $\boldsymbol{\mu}_1 = (0.3, 10)$ and $\boldsymbol{\mu}_2 = (8, 0.2)$. We observe that some behaviors change (although not drastically), such as the response to the input function u on the border $x = 4$, much stronger in the first case.

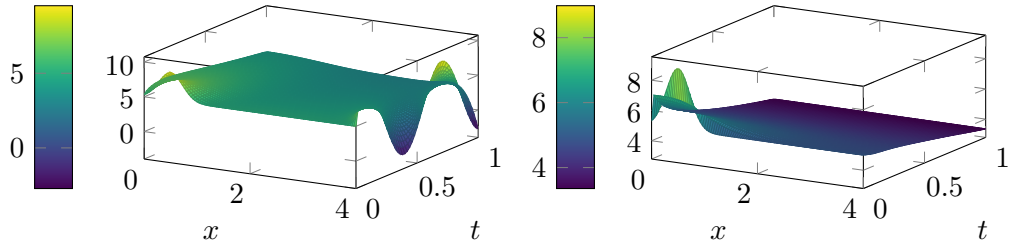


Figure 2.1: Plots of FO state variables corresponding to different parameters.

The parameter optimization will fit the approximation of the state, $y^h_{\boldsymbol{\mu}}$, to the available data, w. In order to get a FO approximation of the reduced cost, we first consider

$$\begin{aligned} w^0 &= \frac{2}{\delta}\int_{t_0}^{t_0+\delta/2} w(s)\mathrm{d}s, \qquad w^K = \frac{2}{\delta}\int_{t_K-\delta/2}^{t_K} w(s)\mathrm{d}s, \\ w^k &= \frac{1}{\delta}\int_{t_k-\delta/2}^{t_k+\delta/2} w(s)\mathrm{d}s \qquad \text{for } k = 1,\dots,K-1 \end{aligned} \tag{2.9}$$

and then we use the projection operator onto V^h defined in (1.4):

$$w^{h,k} = \mathcal{P}^h w^k, \qquad 0 \le k \le K. \tag{2.10}$$

Consider, furthermore, the coefficient vector $\mathrm{w}^{h,k}$ of $w^{h,k}$, the FO approximation of the reduced cost (2.6) is given as

$$\hat{J}^h(\boldsymbol{\mu}) = \frac{\alpha_J}{2}\sum_{k=0}^{K}\alpha_k\|\mathrm{y}^{h,k}_{\boldsymbol{\mu}} - \mathrm{w}^{h,k}\|^2_{\mathrm{M}} + \frac{\beta_J}{2}\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}\|_2^2, \tag{2.11}$$

where α_k are the trapezoidal weights, i.e. $\alpha_0 = \alpha_K = \delta/2$ and $\alpha_k = \delta$ for $k = 1,\dots,K-1$.

Skipping the details, a discrete approximation of the adjoint variable finds $\mathrm{p}^{h,0}_{\boldsymbol{\mu}},\dots,\mathrm{p}^{h,K}_{\boldsymbol{\mu}} \in \mathbb{R}^N$ by solving

$$\begin{aligned} (\mathrm{M}+\delta\mathrm{A}_{\boldsymbol{\mu}})\mathrm{p}^{h,k}_{\boldsymbol{\mu}} &= \mathrm{M}\mathrm{p}^{h,k+1}_{\boldsymbol{\mu}} + \delta\mathrm{M}(\mathrm{w}^{h,k}-\mathrm{y}^{h,k}_{\boldsymbol{\mu}}), \qquad k = 0,\dots,K-1, \\ \mathrm{p}^{h,K}_{\boldsymbol{\mu}} &= 0. \end{aligned} \tag{2.12}$$

Finally, the gradient of the cost function is approximated as

$$\nabla\hat{J}^h(\boldsymbol{\mu}) = \beta_J(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}) + \sum_{k=0}^{K}\alpha_k\sum_{l=1}^{m_a}\nabla\vartheta^a_l(\boldsymbol{\mu})\mathrm{y}^{h,k\top}_{\boldsymbol{\mu}}\hat{\mathrm{A}}^h_l\mathrm{p}^{h,k}_{\boldsymbol{\mu}} - \sum_{k=0}^{K}\alpha_k\sum_{l=1}^{m_f}\nabla\vartheta^f_l(\boldsymbol{\mu})\hat{\mathrm{f}}^{h,k\top}_l\mathrm{p}^{h,k}_{\boldsymbol{\mu}}.$$

2.3 Reduced order discretization

As explained in Section 1.2.3, we construct a reduced order (RO) space $V^\ell = \text{span}\{\psi_1, \ldots, \psi_L\}$ such that $V^\ell \subset V^h$ of dimension $L \ll N$ to built a computationally cheap model. The RO model for (2.7) reads as follows: find $\{\hat{\mathrm{y}}^1_{\boldsymbol{\mu}}, \ldots, \hat{\mathrm{y}}^K_{\boldsymbol{\mu}}\} \subset \mathbb{R}^L$ such that

$$\begin{aligned} (\mathrm{M}^\ell + \delta \mathrm{A}^\ell_{\boldsymbol{\mu}})\hat{\mathrm{y}}^k_{\boldsymbol{\mu}} &= \mathrm{M}^\ell \hat{\mathrm{y}}^{k-1}_{\boldsymbol{\mu}} + \delta \mathrm{f}^{\ell,k}_{\boldsymbol{\mu}}, \qquad k = 1, \ldots, K, \\ \mathrm{M}^\ell \hat{\mathrm{y}}^0_{\boldsymbol{\mu}} &= \mathrm{y}^\ell_\circ, \end{aligned} \tag{2.13}$$

where, using the coefficient matrix $\Psi_\mathrm{y} \in \mathbb{R}^{N \times L}$, we have defined

$$\begin{aligned} &\mathrm{M}^\ell := \Psi_\mathrm{y}^\top \mathrm{M} \Psi_\mathrm{y}, && \mathrm{A}^\ell_{\boldsymbol{\mu}} := \Psi_\mathrm{y}^\top \mathrm{A}_{\boldsymbol{\mu}} \Psi_\mathrm{y}, \\ &\mathrm{f}^{\ell,k}_{\boldsymbol{\mu}} := \Psi_\mathrm{y}^\top \mathrm{f}^{h,k}_{\boldsymbol{\mu}} \quad \text{for } k = 1, \ldots, K, && \mathrm{y}^\ell_\circ := \Psi_\mathrm{y}^\top \mathrm{y}^h_\circ. \end{aligned}$$

Remark 2.4. As in Remark 2.2, due to (2.4),

$$\mathrm{A}^\ell_{\boldsymbol{\mu}} = \sum_{l=1}^{m_a} \vartheta^a_l(\boldsymbol{\mu}) \hat{\mathrm{A}}^\ell_l, \quad \mathrm{f}^{\ell,k}_{\boldsymbol{\mu}} = \sum_{l=1}^{m_f} \vartheta^f_l(\boldsymbol{\mu}) \hat{\mathrm{f}}^{\ell,k}_l$$

holds true for the RO arrays $\hat{\mathrm{A}}^\ell_l := \Psi_\mathrm{y}^\top \hat{\mathrm{A}}^h_l \Psi_\mathrm{y} \in \mathbb{R}^{L \times L}$ and $\hat{\mathrm{f}}^{\ell,k}_l := \Psi_\mathrm{y}^\top \hat{\mathrm{f}}^{h,k}_l \in \mathbb{R}^L$. Hence, once the arrays $\{\hat{\mathrm{A}}^\ell_l\}_{l=1}^{m_a}$ and $\{\hat{\mathrm{f}}^{\ell,k}_l\}_{l=1}^{m_f}$ are evaluated and stored, the evaluation of $\mathrm{A}^\ell_{\boldsymbol{\mu}}$ and $\mathrm{f}^{\ell,k}_{\boldsymbol{\mu}}$ for different parameters $\boldsymbol{\mu}$ does not depend on the FO dimension N. ◊

We reiterate that the solution $\{\hat{\mathrm{y}}^k_{\boldsymbol{\mu}}\}_{k=0}^K$ of (2.13) is interpreted as an RO approximation for $\{\mathrm{y}^{h,k}_{\boldsymbol{\mu}}\}_{k=0}^K$:

$$\mathrm{y}^{h,k}_{\boldsymbol{\mu}} \approx \mathrm{y}^{\ell,k}_{\boldsymbol{\mu}} := \Psi_\mathrm{y} \hat{\mathrm{y}}^k_{\boldsymbol{\mu}} \in \mathbb{R}^N,$$

and for $0 \le k \le K$ we also use

$$y^{\ell,k}_{\boldsymbol{\mu}} := \sum_{i=1}^{L} \hat{\mathrm{y}}^k_{\boldsymbol{\mu} i} \psi_i = \sum_{i=1}^{N} \mathrm{y}^{\ell,k}_{\boldsymbol{\mu} i} \varphi_i \in V^\ell. \tag{2.14}$$

Similarly, we construct a RO space for the adjoint state, $V^{\ell_p} \subset V^h$, of dimension L_p and a basis $\{\psi^p_1, \ldots, \psi^p_{L_p}\}$ associated with coefficient matrix Ψ_p. The RO adjoint variable $\{\hat{\mathrm{p}}^k_{\boldsymbol{\mu}}\}_{k=0}^K$ are then found as

$$\begin{aligned} (\mathrm{M}^{\ell_p} + \delta \mathrm{A}^{\ell_p}_{\boldsymbol{\mu}})\hat{\mathrm{p}}^k_{\boldsymbol{\mu}} &= \mathrm{M}^{\ell_p} \hat{\mathrm{p}}^{k+1}_{\boldsymbol{\mu}} + \delta \mathrm{f}^{\ell_p,k}_p, \qquad k = 0, \ldots, K-1 \\ \hat{\mathrm{p}}^K_{\boldsymbol{\mu}} &= 0. \end{aligned} \tag{2.15}$$

where, in this case, the RO arrays $\mathrm{M}^{\ell_p} \in \mathbb{R}^{L_p \times L_p}$, $\mathrm{A}^{\ell_p}_{\boldsymbol{\mu}} \in \mathbb{R}^{L_p \times L_p}$ and $\mathrm{f}^{\ell_p,k}_p \in \mathbb{R}^{L_p}$ are defined as

$$\begin{aligned} &\mathrm{M}^{\ell_p} := \Psi_\mathrm{p}^\top \mathrm{M} \Psi_\mathrm{p}, \qquad \mathrm{A}^{\ell_p}_{\boldsymbol{\mu}} := \Psi_\mathrm{p}^\top \mathrm{A}_{\boldsymbol{\mu}} \Psi_\mathrm{p}, \\ &\mathrm{f}^{\ell_p,k}_p := \Psi_\mathrm{p}^\top \mathrm{M}(\mathrm{w}^{h,k} - \mathrm{y}^{\ell,k}_{\boldsymbol{\mu}}) = \Psi_\mathrm{p}^\top \mathrm{M} \mathrm{w}^{h,k} - \mathrm{M}^{\ell_p,\ell} \hat{\mathrm{y}}^k_{\boldsymbol{\mu}}, \end{aligned}$$

and $\mathrm{M}^{\ell_p,\ell} := \Psi_\mathrm{p}^\top \mathrm{M} \Psi_\mathrm{y} \in \mathbb{R}^{L_p \times L}$. As in the state variable case, the solution $\{\hat{\mathrm{p}}_{\boldsymbol{\mu}}^k\}_{k=0}^K \subset \mathbb{R}^{L_p}$ of (2.15) exists uniquely and it is interpreted as a RO approximation for $\{\mathrm{p}_{\boldsymbol{\mu}}^{h,k}\}_{k=0}^K$, i.e.

$$\mathrm{p}_{\boldsymbol{\mu}}^{h,k} \approx \mathrm{p}_{\boldsymbol{\mu}}^{\ell_p,k} := \Psi_\mathrm{p} \hat{\mathrm{p}}_{\boldsymbol{\mu}}^k \in \mathbb{R}^N,$$

and for $0 \le k \le K$ we define

$$p_{\boldsymbol{\mu}}^{\ell_p,k} := \sum_{i=1}^{L_p} \hat{\mathrm{p}}_{\boldsymbol{\mu} i}^k \psi_i^p = \sum_{i=1}^{N} \mathrm{p}_{\boldsymbol{\mu} i}^{\ell_p,k} \varphi_i \in V^{\ell_p}.$$

Finally, the reduced cost (2.6) is approximated as

$$\hat{J}^\ell(\boldsymbol{\mu}) := \frac{\alpha_J}{2} \sum_{k=0}^{K} \alpha_k \|\mathrm{y}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\|_\mathrm{M}^2 + \frac{\beta_J}{2} \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2^2, \tag{2.16}$$

and the gradient is given as

$$\nabla \hat{J}^{\ell,\ell_p}(\boldsymbol{\mu}) = \beta_J(\bar{\boldsymbol{\mu}} - \hat{\boldsymbol{\mu}}) + \alpha_J \sum_{k=0}^{K} \alpha_k \sum_{l=1}^{m_a} \nabla \vartheta_l^a(\boldsymbol{\mu}) \hat{\mathrm{y}}_{\boldsymbol{\mu}}^{k\top} \hat{\mathrm{A}}_l^{\ell,\ell_p} \hat{\mathrm{p}}_{\boldsymbol{\mu}}^k - \alpha_J \sum_{k=0}^{K} \alpha_k \sum_{l=1}^{m_f} \nabla \vartheta_l^f(\boldsymbol{\mu}) \hat{\mathrm{f}}_l^{\ell_p,k\top} \hat{\mathrm{p}}_{\boldsymbol{\mu}}^k,$$

where

$$\begin{aligned} \hat{\mathrm{A}}_l^{\ell,\ell_p} &= \Psi_\mathrm{y}^\top \hat{\mathrm{A}}_l^h \Psi_\mathrm{p} \in \mathbb{R}^{L \times L_p} \quad l = 1, \dots, m_a, \\ \hat{\mathrm{f}}_l^{\ell_p,k} &= \Psi_\mathrm{p}^\top \hat{\mathrm{f}}_l^{h,k} \in \mathbb{R}^{L_p} \quad l = 1, \dots, m_f. \end{aligned}$$

Remark 2.5. Observe that the evaluation of the RO reduced cost (2.16) is online-efficient, i.e. it does not depend on the FO dimension N. Indeed,

$$\begin{aligned} \|\mathrm{y}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\|_\mathrm{M}^2 &= \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k\top} \mathrm{M} \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k} - 2\mathrm{w}^{h,k\top} \mathrm{M} \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k} + \mathrm{w}^{h,k\top} \mathrm{M} \mathrm{w}^{h,k} \\ &= \hat{\mathrm{y}}_{\boldsymbol{\mu}}^{k\top} \mathrm{M}^\ell \mathrm{y}_{\boldsymbol{\mu}}^k - \mathrm{J}_1^\top \hat{\mathrm{y}}_{\boldsymbol{\mu}}^k + \mathrm{J}_2, \end{aligned}$$

where $\mathrm{J}_1 = 2\Psi^\top \mathrm{M} \mathrm{w}^{h,k} \in \mathbb{R}^\ell$ and $\mathrm{J}_2 = \mathrm{w}^{h,k\top} \mathrm{M} \mathrm{w}^{h,k} \in \mathbb{R}$ do not depend on $\boldsymbol{\mu}$ and can be pre-evaluated and stored. ◊

2.3.1 Proper orthogonal decomposition

To select an RO space we use the method of POD (see Section 1.2.4) using the state variable snapshots $\{\mathrm{y}_{\boldsymbol{\mu}}^{h,k}\}_{k=0}^K \subset \mathbb{R}^N$ and the finite differences $\{\bar{\partial}\mathrm{y}_{\boldsymbol{\mu}}^{h,k}\}_{k=1}^K \subset \mathbb{R}^N$, or equivalently $\{y_{\boldsymbol{\mu}}^{h,k}\}_{k=0}^K \subset V^h$ and $\{\bar{\partial} y_{\boldsymbol{\mu}}^{h,k}\}_{k=1}^K \subset V^h$.

Regarding the adjoint RO space V^{ℓ_p}, we have two possibilities: we either use the state RO space, i.e. $V^{\ell_p} = V^\ell$, or we create and use a different one, i.e. $V^{\ell_p} \neq V^\ell$. In the following, we try both approaches and observe the differences.

$\boldsymbol{V^{\ell_p} = V^{\ell}}$**:** In this case, since the RO approximation needs to be accurate for both state and adjoint, we include the snapshots of both variables together with their finite differences in time, i.e.

$$\hat{\mathrm{Z}} = [\mathrm{y}^0_{\boldsymbol{\mu}}, \dots, \mathrm{y}^K_{\boldsymbol{\mu}}, \bar{\partial}\mathrm{y}^1_{\boldsymbol{\mu}}, \dots, \bar{\partial}\mathrm{y}^K_{\boldsymbol{\mu}}, \mathrm{p}^0_{\boldsymbol{\mu}}, \dots, \mathrm{p}^K_{\boldsymbol{\mu}}, \bar{\partial}\mathrm{p}^1_{\boldsymbol{\mu}}, \dots, \bar{\partial}\mathrm{p}^K_{\boldsymbol{\mu}}] \in \mathbb{R}^{N\times(4K+2)}.$$

Then, defining

$$\hat{\mathrm{D}} = \begin{pmatrix} \alpha_0 & & & & & & & & \\ & \ddots & & & & & & & \\ & & \alpha_K & & & & & & \\ & & & \delta\mathrm{I}_K & & & & & \\ & & & & \alpha_0 & & & & \\ & & & & & \ddots & & & \\ & & & & & & \alpha_K & & \\ & & & & & & & \delta\mathrm{I}_K \end{pmatrix} \in \mathbb{R}^{(4K-2)\times(4K-2)},$$

we find the eigenvectors corresponding to the L big eigenvalues $\lambda_1, \dots, \lambda_L$ of $\hat{\mathrm{R}} = \hat{\mathrm{Z}}\hat{\mathrm{D}}\hat{\mathrm{Z}}^\top\mathrm{W}$. Then, defining the bases $\psi_1, \dots, \psi_L$ as described previously, the error formula now gives

$$\begin{aligned} &\sum_{k=0}^{K} \alpha_k \left\| y^{h,k}_{\boldsymbol{\mu}} - \sum_{i=1}^{L} \langle y^{h,k}_{\boldsymbol{\mu}}, \psi_i\rangle_V \ \psi_i \right\|^2_V + \sum_{k=1}^{K} \delta \left\| \bar{\partial} y^{h,k} - \sum_{i=1}^{L} \langle \bar{\partial} y^{h,k}, \psi_i\rangle_V \ \psi_i \right\|^2_V \\ &+ \sum_{k=0}^{K} \alpha_k \left\| p^{h,k}_{\boldsymbol{\mu}} - \sum_{i=1}^{L} \langle p^{h,k}_{\boldsymbol{\mu}}, \psi_i\rangle_V \ \psi_i \right\|^2_V + \sum_{k=1}^{K} \delta \left\| \bar{\partial} p^{h,k} - \sum_{i=1}^{L} \langle \bar{\partial} p^{h,k}, \psi_i\rangle_V \ \psi_i \right\|^2_V = \sum_{j=L+1}^{K} \lambda_i. \end{aligned}$$

Hence, $\Psi_{\mathrm{y}} = \Psi_{\mathrm{p}}$, i.e. the coefficient matrices for the state variable and the adjoint variables are the same (because the RO space is the same).

$\boldsymbol{V^{\ell_p} \neq V^{\ell}}$**:** In this case we simply use the POD algorithm on the set of adjoint snapshots

$$\hat{\mathrm{Z}} = [\mathrm{p}^0_{\boldsymbol{\mu}}, \dots, \mathrm{p}^K_{\boldsymbol{\mu}}, \bar{\partial}\mathrm{p}^1_{\boldsymbol{\mu}}, \dots, \bar{\partial}\mathrm{p}^K_{\boldsymbol{\mu}}] \in \mathbb{R}^{N\times(2K+1)}$$

and use $\tilde{\mathrm{D}}$ as defined in (1.15). Then, extracting eigenvalues and eigenvectors of $\hat{\mathrm{R}} := \hat{\mathrm{Z}}\tilde{\mathrm{D}}\hat{\mathrm{Z}}^\top\mathrm{W}$ we find the L_p-dimensional basis $\psi^p_1, \dots, \psi^p_{L_p}$ and from that the coefficient matrix $\Psi_{\mathrm{p}} \in \mathbb{R}^{N\times L_p}$.

One of the most evident consequence of choosing $V^{\ell_p} = V^{\ell}$ is the memory saving: in this case $\mathrm{M}^{\ell} = \mathrm{M}^{\ell_p} = \mathrm{M}^{\ell_p,\ell}$ and $\mathrm{A}^{\ell}_{\boldsymbol{\mu}} = \mathrm{A}^{\ell_p}_{\boldsymbol{\mu}}$. Another positive side to this approach is that the POD algorithm is done only once (but using a bigger snapshots matrix). On the other side, using two separate spaces improves the accuracy of both the state and the adjoint FO approximation. We have favored the second option, as we prefer keeping the accuracy high, and the memory taken by these matrices is not excessive.

2.3.2 A-posteriori error estimate

In the next section we discuss the optimization algorithm in the trust-region framework, where the surrogate model is the RO approximation. In this framework, error estimates

are essential to measure the accuracy of the surrogate RO model without evaluating the FO model, to save computational time and memory. In particular, we find a-posteriori error estimate

$$\|y_{\boldsymbol{\mu}}^h - y_{\boldsymbol{\mu}}^\ell\| \le \Delta_{\mathrm{y}}^\ell(\boldsymbol{\mu}) = \Delta_{\mathrm{y}}^\ell(\boldsymbol{\mu}, y_{\boldsymbol{\mu}}^\ell),$$

i.e. computable without evaluating the FO state but only (eventually) the RO state.

Let us define the $\boldsymbol{\mu}$-dependent space-time energy norm (cf. [28, 26]) for the sequence $\{y_{\boldsymbol{\mu}}^{h,k}\}_{k=0}^K \subset V^h$ up to time step κ as

$$\|y_{\boldsymbol{\mu}}^{h,\kappa}\|_{\boldsymbol{\mu}} := \left(\|y_{\boldsymbol{\mu}}^{h,\kappa}\|_H^2 + \delta \sum_{k=1}^{\kappa} a_{\boldsymbol{\mu}}(y_{\boldsymbol{\mu}}^{h,k}, y_{\boldsymbol{\mu}}^{h,k}) \right)^{1/2}$$

for $\kappa = 1, \dots, K$, while $\|y_{\boldsymbol{\mu}}^{h,0}\|_{\boldsymbol{\mu}} := \|y_{\boldsymbol{\mu}}^{h,0}\|_H$. Furthermore, let us remind that $\underline{\alpha}_{\boldsymbol{\mu}} > 0$ is the $\boldsymbol{\mu}$-dependent coercitivity constant of the bilinear form $a_{\boldsymbol{\mu}}$. Then we can define the error estimate for the state variable in the following proposition.

Proposition 2.6. *Let $\{y_{\boldsymbol{\mu}}^{h,k}\}_{k=0}^K$ and $\{y_{\boldsymbol{\mu}}^{\ell,k}\}_{k=0}^K$ be given as (2.8) and (2.14), respectively. Let $e_{\boldsymbol{\mu}}^{\ell,k} := y_{\boldsymbol{\mu}}^{h,k} - y_{\boldsymbol{\mu}}^{\ell,k} \in V^h$ for $k = 0, \dots, K$. For $\varphi \in V^h$ we define the residual*

$$\langle \mathcal{R}_{\boldsymbol{\mu}}^k, \varphi \rangle_{V^*,V} := \frac{1}{\delta} \langle y_{\boldsymbol{\mu}}^{\ell,k-1} - y_{\boldsymbol{\mu}}^{\ell,k}, \varphi \rangle_H - a_{\boldsymbol{\mu}}(y_{\boldsymbol{\mu}}^{\ell,k}, \varphi) + \langle f_{\boldsymbol{\mu}}(t_k; u), \varphi \rangle_{V^*,V}$$

and its norm $\varepsilon_{\boldsymbol{\mu}}^k := \|\mathcal{R}_{\boldsymbol{\mu}}^k\|_{V^*}$. Then

$$\|e_{\boldsymbol{\mu}}^{\ell,\kappa}\|_{\boldsymbol{\mu}} \le \Delta_{\mathrm{y}}^{\ell,\kappa}(\boldsymbol{\mu}) \qquad \text{for } \kappa = 0, \dots, K, \tag{2.17}$$

where

$$\Delta_{\mathrm{y}}^{\ell,\kappa}(\boldsymbol{\mu}) := \left(\|e_{\boldsymbol{\mu}}^{\ell,0}\|_H^2 + \frac{\delta}{\underline{\alpha}_{\boldsymbol{\mu}}} \sum_{k=1}^{\kappa} \left| \varepsilon_{\boldsymbol{\mu}}^k \right|^2 \right)^{1/2}. \tag{2.18}$$

Proof. It is easy to check that for $k = 2, \dots, K$ the error $e_{\boldsymbol{\mu}}^{\ell,k}$ solves the equation

$$\langle e_{\boldsymbol{\mu}}^{\ell,k}, \varphi \rangle_H + \delta a_{\boldsymbol{\mu}}(e_{\boldsymbol{\mu}}^{\ell,k}, \varphi) - \langle e_{\boldsymbol{\mu}}^{\ell,k-1}, \varphi \rangle_H = \delta \mathcal{R}_{\boldsymbol{\mu}}^k(\varphi) \qquad \forall \varphi \in V^h.$$

Choosing $\varphi = e_{\boldsymbol{\mu}}^{\ell,k} \in V^h$, we get

$$\langle e_{\boldsymbol{\mu}}^{\ell,k}, e_{\boldsymbol{\mu}}^{\ell,k} \rangle_H + \delta a_{\boldsymbol{\mu}}(e_{\boldsymbol{\mu}}^{\ell,k}, e_{\boldsymbol{\mu}}^{\ell,k}) = \langle e_{\boldsymbol{\mu}}^{\ell,k-1}, e_{\boldsymbol{\mu}}^{\ell,k} \rangle_H + \delta \mathcal{R}_{\boldsymbol{\mu}}^k(e_{\boldsymbol{\mu}}^{\ell,k}). \tag{2.19}$$

Then, using the Cauchy-Schwarz and Young inequalities and (2.3), we estimate

$$\langle e_{\boldsymbol{\mu}}^{\ell,k-1}, e_{\boldsymbol{\mu}}^{\ell,k} \rangle_H \le \langle e_{\boldsymbol{\mu}}^{\ell,k-1}, e_{\boldsymbol{\mu}}^{\ell,k-1} \rangle_H^{1/2} \langle e_{\boldsymbol{\mu}}^{\ell,k}, e_{\boldsymbol{\mu}}^{\ell,k} \rangle_H^{1/2} \le \frac{1}{2} \|e_{\boldsymbol{\mu}}^{\ell,k-1}\|_H^2 + \frac{1}{2} \|e_{\boldsymbol{\mu}}^{\ell,k}\|_H^2, \tag{2.20}$$

and

$$\mathcal{R}_{\boldsymbol{\mu}}^k(e_{\boldsymbol{\mu}}^{\ell,k}) \le \varepsilon_{\boldsymbol{\mu}}^k \|e_{\boldsymbol{\mu}}^{\ell,k}\|_V \le \frac{1}{2\underline{\alpha}_{\boldsymbol{\mu}}} |\varepsilon_{\boldsymbol{\mu}}^k|^2 + \frac{\underline{\alpha}_{\boldsymbol{\mu}}}{2} \|e_{\boldsymbol{\mu}}^{\ell,k}\|_V^2 \le \frac{1}{2\underline{\alpha}_{\boldsymbol{\mu}}} |\varepsilon_{\boldsymbol{\mu}}^k|^2 + \frac{1}{2} a_{\boldsymbol{\mu}}(e_{\boldsymbol{\mu}}^{\ell,k}, e_{\boldsymbol{\mu}}^{\ell,k}). \tag{2.21}$$

Hence, inserting (2.20) and (2.21) in (2.19), we get

$$\frac{1}{2}\|e_{\boldsymbol{\mu}}^{\ell,k}\|_H^2 - \frac{1}{2}\|e_{\boldsymbol{\mu}}^{\ell,k-1}\|_H^2 + \frac{\delta}{2}a_{\boldsymbol{\mu}}(e_{\boldsymbol{\mu}}^{\ell,k}, e_{\boldsymbol{\mu}}^{\ell,k}) \leq \frac{\delta}{2\underline{\alpha}_{\boldsymbol{\mu}}}|\varepsilon_{\boldsymbol{\mu}}^k|^2.$$

Now, summing for $k = 1, \dots, \kappa$ and multiplying by two, we get

$$\|e_{\boldsymbol{\mu}}^{\ell,\kappa}\|_H^2 - \|e_{\boldsymbol{\mu}}^{\ell,0}\|_H^2 + \delta\sum_{k=1}^{\kappa} a_{\boldsymbol{\mu}}(e_{\boldsymbol{\mu}}^{\ell,k}, e_{\boldsymbol{\mu}}^{\ell,k}) \leq \frac{\delta}{\underline{\alpha}_{\boldsymbol{\mu}}}\sum_{k=1}^{\kappa}\left|\varepsilon_{\boldsymbol{\mu}}^k\right|^2.$$

Hence, rearranging we get (2.17). □

Remark 2.7. We show now how to evaluate the squared residual norm $\left|\varepsilon_{\boldsymbol{\mu}}^k\right|^2 = \|\mathcal{R}_{\boldsymbol{\mu}}^k\|_{V^*}^2$ for every $k = 1, \dots, K$, using the Riesz representation theorem

$$\langle R_{\boldsymbol{\mu}}^k, \varphi\rangle_V = \langle \mathcal{R}_{\boldsymbol{\mu}}^k, \varphi\rangle_{V^*,V} \qquad \forall \varphi \in V^h,$$

where $R_{\boldsymbol{\mu}}^k \in V^h$ is the Riesz representant of $\mathcal{R}_{\boldsymbol{\mu}}^k$. To compute it efficiently, we solve a-priori the following problems to evaluate $r^{1,k}, r_i^2, r_i^3 \in V^h$ for $k = 0, \dots, K$ and $i = 1, \dots, L$

$$\begin{aligned}
\langle r^{1,k}, \varphi\rangle_V &= \langle f_{\boldsymbol{\mu}}(t_k), \varphi\rangle_{V^*,V} && \forall \varphi \in V^h,\\
\langle r_i^2, \varphi\rangle_V &= -a_{\boldsymbol{\mu}}(\psi_i, \varphi) && \forall \varphi \in V^h,\\
\langle r_i^3, \varphi\rangle_V &= \frac{1}{\delta}\langle \psi_i, \varphi\rangle_H && \forall \varphi \in V^h.
\end{aligned}$$

Then, using (2.14), we can write

$$\begin{aligned}
\langle R_{\boldsymbol{\mu}}^k, \varphi\rangle_V &= \langle \mathcal{R}_{\boldsymbol{\mu}}^k, \varphi\rangle_{V^*,V} = \langle f_{\boldsymbol{\mu}}(t_k), \varphi\rangle_{V^*,V} - a_{\boldsymbol{\mu}}(y_{\boldsymbol{\mu}}^{\ell,k}, \varphi) + \frac{1}{\delta}\langle y_{\boldsymbol{\mu}}^{\ell,k-1} - y_{\boldsymbol{\mu}}^{\ell,k}, \varphi\rangle_H\\
&= \langle f_{\boldsymbol{\mu}}(t_k), \varphi\rangle_{V^*,V} - a_{\boldsymbol{\mu}}\left(\sum_{i=1}^{L}\hat{\mathrm{y}}_{\boldsymbol{\mu}i}^k\psi_i, \varphi\right) + \frac{1}{\delta}\left\langle\sum_{i=1}^{L}(\hat{\mathrm{y}}_{\boldsymbol{\mu}i}^{k-1} - \hat{\mathrm{y}}_{\boldsymbol{\mu}i}^k)\psi_i, \varphi\right\rangle_H\\
&= \left\langle r^{1,k} + \sum_{i=1}^{L}\hat{\mathrm{y}}_{\boldsymbol{\mu}i}^k r_i^2 + \sum_{i=1}^{L}(\hat{\mathrm{y}}_{\boldsymbol{\mu}i}^{k-1} - \hat{\mathrm{y}}_{\boldsymbol{\mu}i}^k)r_i^3, \varphi\right\rangle_V,
\end{aligned}$$

and therefore

$$R_{\boldsymbol{\mu}}^k = r^{1,k} + \sum_{i=1}^{L}\hat{\mathrm{y}}_{\boldsymbol{\mu}i}^k r_i^2 + \sum_{i=1}^{L}(\hat{\mathrm{y}}_{\boldsymbol{\mu}i}^{k-1} - \hat{\mathrm{y}}_{\boldsymbol{\mu}i}^k)r_i^3 \in V^h.$$

In particular, using the coordinate vectors (e.g. $\mathrm{r}^{1,k} \in \mathbb{R}^N$ for $r^{1,k} \in V^h$), we now illustrate how to evaluate the i-th coordinate. We first observe that

$$\begin{aligned}
(\mathrm{Wr}^{1,k})_j &= \langle r^{1,k}, \varphi_j\rangle_V = \mathrm{f}_{\boldsymbol{\mu}j}^{h,k},\\
(\mathrm{Wr}_i^2)_j &= \langle r_i^2, \varphi_j\rangle_V = -a_{\boldsymbol{\mu}}(\psi_i, \varphi_j) = -a_{\boldsymbol{\mu}}\left(\sum_{l=1}^{N}(\Psi_{\mathrm{y}})_{l,i}\varphi_l, \varphi_j\right) = -(\Psi_{\mathrm{y}}^\top \mathrm{A}_{\boldsymbol{\mu}})_{i,j},\\
(\mathrm{Wr}_j^3)_i &= \langle r_i^3, \varphi_j\rangle_V = \frac{1}{\delta}\langle \psi_i, \varphi_j\rangle_H = \frac{1}{\delta}\left\langle\sum_{l=1}^{N}(\Psi_{\mathrm{y}})_{l,i}\varphi_l, \varphi_j\right\rangle_H = \frac{1}{\delta}(\Psi_{\mathrm{y}}^\top \mathrm{M})_{i,j}.
\end{aligned}$$

Hence, the coordinate vector $\mathrm{R}^k_{\boldsymbol{\mu}} \in \mathbb{R}^N$ of $R^k_{\boldsymbol{\mu}} \in V^h$ can be evaluated by the equation

$$\mathrm{WR}^k_{\boldsymbol{\mu}} = (\mathrm{f}^{h,k}_{\boldsymbol{\mu}} - \mathrm{A}_{\boldsymbol{\mu}}\Psi_{\mathrm{y}}\hat{\mathrm{y}}^k_{\boldsymbol{\mu}} + \frac{1}{\delta}\mathrm{M}\Psi_{\mathrm{y}}(\hat{\mathrm{y}}^{k-1}_{\boldsymbol{\mu}} - \hat{\mathrm{y}}^k_{\boldsymbol{\mu}}) = \mathrm{f}^{h,k}_{\boldsymbol{\mu}} - \mathrm{A}_{\boldsymbol{\mu}}\mathrm{y}^{\ell,k}_{\boldsymbol{\mu}} + \frac{1}{\delta}\mathrm{M}(\mathrm{y}^{\ell,k-1}_{\boldsymbol{\mu}} - \mathrm{y}^{\ell,k}_{\boldsymbol{\mu}}).$$

$\Diamond$

In the next proposition we define an error estimate for the RO reduced cost.

Proposition 2.8. *Let $\{y^{h,k}_{\boldsymbol{\mu}}\}_{k=0}^K$ and $\{y^{\ell,k}_{\boldsymbol{\mu}}\}_{k=0}^K$ be given as (2.8) and (2.14), respectively, the reduced cost functions $\hat{J}^h(\boldsymbol{\mu})$ and $\hat{J}^\ell(\boldsymbol{\mu})$ be given as (2.11) and (2.16), respectively, the estimator $\Delta^{\ell,k}_{\mathrm{y}}(\boldsymbol{\mu})$ be defined by (2.18), and the cost error be defined by $E^\ell_J(\boldsymbol{\mu}) := \hat{J}^h(\boldsymbol{\mu}) - \hat{J}^\ell(\boldsymbol{\mu})$. Then, the following inequality holds:*

$$|E^\ell_J(\boldsymbol{\mu})| \leq \Delta^\ell_J(\boldsymbol{\mu}),$$

where

$$\Delta^\ell_J(\boldsymbol{\mu}) := \frac{\alpha_J}{2}\sum_{k=0}^K \alpha_k \left(\Delta^{\ell,k}_{\mathrm{y}}(\boldsymbol{\mu})^2 + 2\Delta^{\ell,k}_{\mathrm{y}}(\boldsymbol{\mu})\,\|y^{\ell,k}_{\boldsymbol{\mu}} - w^{h,k}\|_H\right), \tag{2.22}$$

which, again, does not require the evaluation of the full-order solution $\{y^k_{\boldsymbol{\mu}}\}_{k=0}^K$.

Proof. First, we can write

$$\begin{aligned}
&\|w^{h,k} - y^{h,k}_{\boldsymbol{\mu}}\|^2_H - \|w^{h,k} - y^{\ell,k}_{\boldsymbol{\mu}}\|^2_H \\
&\quad= \langle w^{h,k} - y^{h,k}_{\boldsymbol{\mu}}, w^{h,k} - y^{h,k}_{\boldsymbol{\mu}}\rangle_H - \langle w^{h,k} - y^{\ell,k}_{\boldsymbol{\mu}}, w^{h,k} - y^{\ell,k}_{\boldsymbol{\mu}}\rangle_H \\
&\quad= \langle w^{h,k} - y^{h,k}_{\boldsymbol{\mu}}, y^{\ell,k}_{\boldsymbol{\mu}} - y^{h,k}_{\boldsymbol{\mu}}\rangle_H - \langle y^{h,k}_{\boldsymbol{\mu}} - y^{\ell,k}_{\boldsymbol{\mu}}, w^{h,k} - y^{\ell,k}_{\boldsymbol{\mu}}\rangle_H \\
&\quad= \langle e^{\ell,k}_{\boldsymbol{\mu}}, y^{h,k}_{\boldsymbol{\mu}} + y^{\ell,k}_{\boldsymbol{\mu}} - 2w^{h,k}\rangle_H \\
&\quad= \langle e^{\ell,k}_{\boldsymbol{\mu}}, e^{\ell,k}_{\boldsymbol{\mu}} + 2y^{\ell,k}_{\boldsymbol{\mu}} - 2w^{h,k}\rangle_H \\
&\quad= \|e^{\ell,k}_{\boldsymbol{\mu}}\|^2_H + 2\,\langle e^{\ell,k}_{\boldsymbol{\mu}}, y^{\ell,k}_{\boldsymbol{\mu}} - w^{h,k}\rangle_H.
\end{aligned} \tag{2.23}$$

Moreover, using the Cauchy-Schwarz inequality

$$\left|\langle e^{\ell,k}_{\boldsymbol{\mu}}, y^{\ell,k}_{\boldsymbol{\mu}} - w^{h,k}\rangle_H\right| \leq \|e^{\ell,k}_{\boldsymbol{\mu}}\|_H \|y^{\ell,k}_{\boldsymbol{\mu}} - w^{h,k}\|_H \tag{2.24}$$

Then, using (2.23) and (2.24) we get

$$\begin{aligned}
\left|E^\ell_J(\boldsymbol{\mu})\right| &\leq \frac{\alpha_J}{2}\sum_{k=0}^K \alpha_k \left|\|w^{h,k} - y^{h,k}_{\boldsymbol{\mu}}\|^2_H - \|w^{h,k} - y^{\ell,k}_{\boldsymbol{\mu}}\|^2_H\right| \\
&\leq \frac{\alpha_J}{2}\sum_{k=0}^K \alpha_k \left(\|e^{\ell,k}_{\boldsymbol{\mu}}\|^2_H + 2\,\|e^{\ell,k}_{\boldsymbol{\mu}}\|_H \|y^{\ell,k}_{\boldsymbol{\mu}} - w^{h,k}\|_H\right) \\
&\leq \frac{\alpha_J}{2}\sum_{k=0}^K \alpha_k \left(\Delta^{\ell,k}_{\mathrm{y}}(\boldsymbol{\mu})^2 + 2\Delta^{\ell,k}_{\mathrm{y}}(\boldsymbol{\mu})\,\|y^{\ell,k}_{\boldsymbol{\mu}} - w^{h,k}\|_H\right).
\end{aligned}$$

$\square$

Example 2.9. Let us take in consideration the same system from Example 2.3, and observe the performance of estimators (2.19) and (2.22). We set $\boldsymbol{\mu} = (1.82, 5.055)$, and using POD we consider all the RO approximations with number of bases L going from 1 to 20. We observe the real errors

$$E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu}) := \|y_{\boldsymbol{\mu}}^{h,K} - y_{\boldsymbol{\mu}}^{\ell,K}\|_{\boldsymbol{\mu}} \qquad \text{and} \qquad E_J^{\ell}(\boldsymbol{\mu}) := |\hat{J}^h(\boldsymbol{\mu}) - \hat{J}^{\ell}(\boldsymbol{\mu})|,$$

the estimates $\Delta_{\mathrm{y}}^{\ell,K}(\boldsymbol{\mu})$ and $\Delta_J^{\ell}(\boldsymbol{\mu})$, and the efficiencies

$$\eta_{\mathrm{y}}(\boldsymbol{\mu}) := \frac{\Delta_{\mathrm{y}}^{\ell,K}(\boldsymbol{\mu})}{E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu})} \qquad \text{and} \qquad \eta_J(\boldsymbol{\mu}) := \frac{\Delta_J^{\ell}(\boldsymbol{\mu})}{E_J^{\ell}(\boldsymbol{\mu})}.$$

As we see in Figure 2.2, the errors and estimates generally decrease with the growing of the RB space, but the efficiency does not; for the state variable it stays around two, while for the state it grows, meaning that the estimate decrease with a slower rate than the real error. This does not alarm us, because the error is nonetheless decreasing and it will not impact the result of trust region algorithm in a perceptible way.

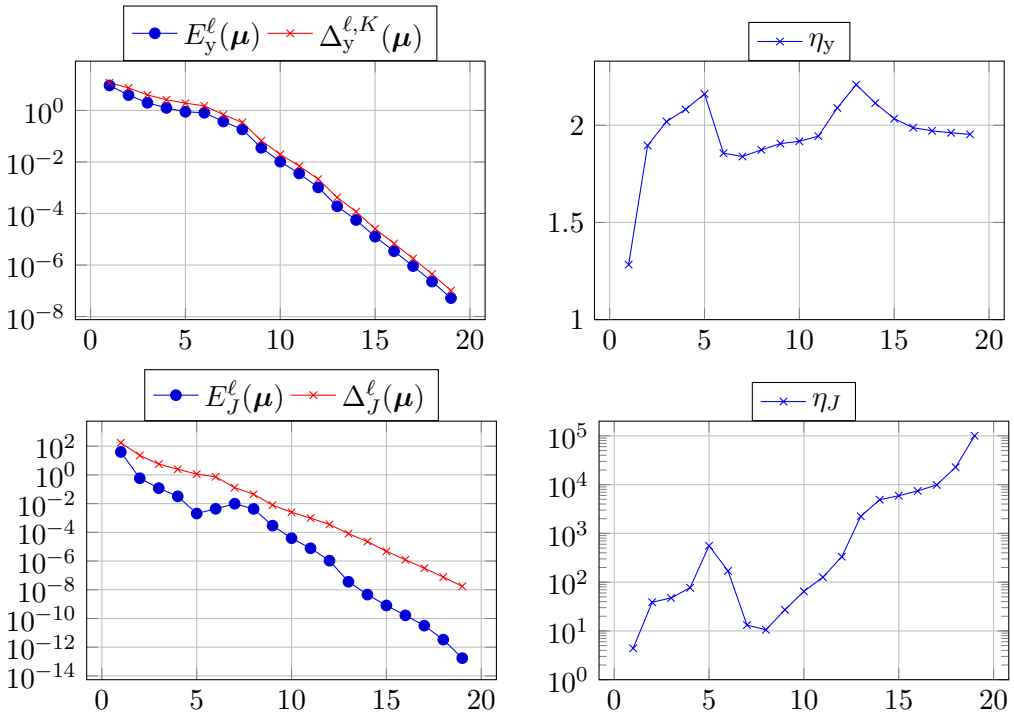


Figure 2.2: Efficiency analysis of estimators (2.19) and (2.22) for an arbitrary $\boldsymbol{\mu}$.

2.4 MOR-based optimization

To solve the optimization problem $(\hat{\mathrm{P}}_1)$ we want to apply a gradient-based optimization method using the RO model to accelerate evaluations of the reduced cost and its gradient.

We then have two possible strategies in order to do this: the offline-online strategy and the adaptive strategy.

In the offline-online strategy, as the name suggests, we have two separate phases: in the offline phase we build a global reduced basis, i.e. accurate for every parameter $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$. Then, in the online phase, the RO model is evaluated in a fast and efficient manner. We will use several time the term online-efficient, meaning evaluations that are independent of the full dimension N and instead rely on the reduced dimension L (and eventually L_p). The adaptive strategy, on the other hand, comes from the fact that the optimization method does not need the RO model to be accurate in every parameter in the admissible parameter set, but only for some. In this case the RO basis is built adaptively during the optimization and the RO model will be accurate for the parameter along the optimization path, but we have no guarantee of accuracy for the other parameters.

We will see both strategies in details, together with some final numerical examples.

2.4.1 Offline-online strategy

Our offline phase is based on the *weak greedy* algorithm described in [26, 28], which builds a RO space by discretizing the admissible parameter space and adding elements to the RO basis until the RO model is accurate for all parameters in the discretized admissible parameter set. The procedure is shown in Algorithm 3. This algorithm shows how to compute a RO space V^ℓ for the state variable. For the adjoint variable, we can repeat the same algorithm and use the RO state in V^ℓ to build V^{ℓ_p}.

Algorithm 3: Weak greedy algorithm for the single-state linear model

Input: Error tolerance $\tau > 0$, initial parameter $\hat{\boldsymbol{\mu}} \in \mathcal{P}_{\mathsf{ad}}$, discretized parameter set $\mathcal{P}_{\mathsf{discr}} \subset \mathcal{P}_{\mathsf{ad}}$, maximum RO dimension $L^{\mathsf{max}} > 0$;

```
Create RO space at μ̂;
while Δ̂ := max_{μ∈P_discr} Δ_y^{ℓ,K}(μ) > τ do
    Set μ̂ := arg max_{μ∈P_discr} Δ_y^{ℓ,K}(μ);
    Enrich the RO space at μ̂;
```

Create the first RO space: The first RO space is created through the POD method at $\hat{\boldsymbol{\mu}}$, meaning using information from the initial parameter $\hat{\boldsymbol{\mu}}$. In particular, we use the POD method to build the matrix Ψ_{y} of dimension L, where L is the smallest integer such that the condition $\Delta_{\mathrm{y}}^{\ell,K}(\hat{\boldsymbol{\mu}}) \leq \tau$ is satisfied. In this way, it is guaranteed that the greedy algorithm will not pick the starting parameter $\hat{\boldsymbol{\mu}}$ again.

The exit criterion: The term *weak* expresses the use of the error estimate as a stopping criterion. The standard (*strong*) algorithm relies on the error $\|y_{\boldsymbol{\mu}}^{h,K} - y_{\boldsymbol{\mu}}^{\ell,K}\|_{\boldsymbol{\mu}}$ to check the accuracy of the RB space. The weak version has both positive and negative consequences: firstly, the estimate $\Delta_{\mathrm{y}}^{\ell,K}(\boldsymbol{\mu})$ is faster to evaluate and the condition $\Delta_{\mathrm{y}}^{\ell,K}(\boldsymbol{\mu}) \leq \tau$ implies $\|y_{\boldsymbol{\mu}}^{h,K} - y_{\boldsymbol{\mu}}^{\ell,K}\|_{\boldsymbol{\mu}} \leq \tau$. On the other side, however, if the estimate is too loose, we might need a much higher RO dimension to reach the exit criterion.

Since in this case the error estimate is tight enough, we choose to use the weak algorithm.

Enrich the RO space: The enrichment of a RO space means adding new elements to the RO basis set. To *enrich* a space in $\hat{\boldsymbol{\mu}}$ means that the new RO basis elements will be evaluated using the POD method on the snapshots derived by the parameter $\hat{\boldsymbol{\mu}}$. In particular, if Z is the snapshot matrix (as in (1.14)) evaluated using the parameter $\hat{\boldsymbol{\mu}}$, then we first orthogonalize the snapshots with respect to V^ℓ, i.e. we evaluate

$$\tilde{\mathrm{Z}} := \mathrm{Z} - \sum_{i=1}^{L} \langle \mathrm{Z}, \Psi_{\mathrm{y}i} \rangle_{\mathrm{W}} \, \Psi_{\mathrm{y}i}.$$

Then, we use the POD method on $\tilde{\mathrm{Z}}$ and evaluate the new L^+ eigenvectors as the columns of the new coefficient matrix $\Psi^+ \in \mathbb{R}^{N \times L^+}$. This matrix, merged together with the old coefficient matrix $\Psi_{\mathrm{y}} \in \mathbb{R}^{N \times L}$ gives the new coefficient matrix, i.e. we now set $\Psi_{\mathrm{y}} := [\Psi_{\mathrm{y}} | \Psi^+] \in \mathbb{R}^{N \times (L+L^+)}$, thus substituting the RO space V^ℓ with a bigger one: $V^\ell := \mathrm{span}\{\psi_1, \dots, \psi_L, \psi_1^+, \dots, \psi_{L^+}^+\}$. Note that the matrices Ψ_{y} and Ψ^+ are already W-orthogonal, namely $\langle \psi_i, \psi_j^+ \rangle_V = \Psi_{\mathrm{y}i}^\top \mathrm{W} \Psi_j^+ = 0$ holds for all $i = 1, \dots, L$ and $j = 1, \dots, L^+$, but it can be enforced numerically during the merging through, e.g., the Gram-Schmidt algorithm. Furthermore, L^+ is chosen as the smaller integer such that the error estimator after enrichment is smaller than the tolerance, i.e. $\Delta_{\mathrm{y}}^{\ell,K}(\hat{\boldsymbol{\mu}}) \leq \tau$ holds true after the enrichment.

2.4.2 Adaptive strategy

In the context of optimization, the offline computation might not be very suitable, because during the optimization method only parameters from a small (but a-priorily unknown) subset of $\mathcal{P}_{\mathsf{ad}}$ are required. In [51, 33, 6] the RB space (for an elliptic problem) is built during a trust region (TR) optimization process, where – following the optimization path – the RB space is enriched if it is necessary. We will utilize these ideas in the context of parabolic problems. The full algorithm is presented in Algorithm 4, and we refer to [6] for further information.

Trust region framework The TR optimization method computes iteratively a first-order critical point of $(\hat{\mathrm{P}}_1)$. At each iteration $n \geq 0$ of the optimization algorithm, we generate a parameter $\boldsymbol{\mu}^{(n)}$, and the sequence of parameters converges to a local optimum. At iteration n we consider a cheaply computable model $m^{(n)}$ (approximation of the reduced cost) that can be trusted to accurately represent the function $\hat{J}^h$ in a neighborhood $\mathcal{T}(\rho^{(n)})$ of $\boldsymbol{\mu}^{(n)}$ depending on the parameter $\rho^{(n)}$, which quantifies our trust in the accuracy of the cheap model. The set $\mathcal{T}(\rho^{(n)})$ and the number $\rho^{(n)}$ take the name, respectively, of *trust region* and *trust region radius*. The TR method finds $\boldsymbol{\mu}^{(n+1)}$ by solving the problem

$$\min_{s \in \mathbb{R}^d} m^{(n)}(s) \quad \text{s.t.} \quad \|s\|_2 \leq \rho^{(n)}, \, \boldsymbol{\mu}^{(n)} + s \in \mathcal{P}_{\mathsf{ad}}$$

Algorithm 4: Adaptive TR-RB optimization

Input: Initial parameter $\boldsymbol{\mu}^{(0)} \in \mathcal{P}_{\mathsf{ad}}$, tolerance $\tau > 0$, initial TR radius $\rho_0 > 0$, maximum number of iterations N_{maxit}, minimum TR radius ρ_{min};

Initialize the RO space at $\boldsymbol{\mu}^{(0)}$. Set $n = 0$, $\rho^{(0)} = \rho_0$, Loop_flag = True;
while Loop_flag ***and*** $\rho^{(n)} > \rho_{\mathsf{min}}$ **do**
  Find $\tilde{\boldsymbol{\mu}}$ solution of the TR sub-problem;
  Set $\rho^{(n+1)} = \rho^{(n)}$;
  if $\hat{J}^{\ell,(n)}(\tilde{\boldsymbol{\mu}}) + \Delta_{\hat{J}}^{\ell,(n)}(\tilde{\boldsymbol{\mu}}) < \hat{J}^{\ell,(n)}(\boldsymbol{\mu}_{\mathsf{AGC}}^{(n)})$ **then**
    Accept parameter $\boldsymbol{\mu}^{(n+1)} = \tilde{\boldsymbol{\mu}}$;
    if $g^h(\boldsymbol{\mu}^{(n+1)}) \leq \tau$ **then**
      Set Loop_flag = False;
    else
      if $\gamma^{(n)} \geq \bar{\gamma}$ **then** Enlarge the TR radius $\rho^{(n+1)} = 2\rho^{(n)}$;
      if ***not*** Skip_enrichment_flag(n) **then** Enrich RO space at $\boldsymbol{\mu}^{(n+1)}$;
  else if $\hat{J}^{\ell,(n)}(\tilde{\boldsymbol{\mu}}) - \Delta_{\hat{J}}^{\ell,(n)}(\tilde{\boldsymbol{\mu}}) > \hat{J}^{\ell,(n)}(\boldsymbol{\mu}_{\mathsf{AGC}}^{(n)})$ **then**
    Reject parameter $\tilde{\boldsymbol{\mu}}$;
    if $\rho^{(n)}/2 \leq \rho_{\mathsf{min}}$ ***or*** Skip_enrichment_flag$(n-1)$ **then**
      Enrich RO space at $\tilde{\boldsymbol{\mu}}$;
    else
      Shrink the TR radius $\rho^{(n+1)} = \rho^{(n)}/2$;
  else
    if $g^h(\tilde{\boldsymbol{\mu}}) \leq \tau$ **then**
      Accept final parameter $\boldsymbol{\mu}^{(n+1)} = \tilde{\boldsymbol{\mu}}$ and set Loop_flag = False;
    else
      if Skip_enrichment_flag(n) ***and*** $\gamma^{(n)} \geq \bar{\gamma}$ **then**
        Accept parameter $\boldsymbol{\mu}^{(n+1)} = \tilde{\boldsymbol{\mu}}$ and set $\rho^{(n+1)} = 2\rho^{(n)}$;
      else if $\hat{J}^h(\boldsymbol{\mu}^{(n+1)}) \leq \hat{J}^{\ell,(n)}(\boldsymbol{\mu}_{\mathsf{ACG}}^{(n)})$ **then**
        Accept parameter $\boldsymbol{\mu}^{(n+1)} = \tilde{\boldsymbol{\mu}}$ and enrich RO space at $\boldsymbol{\mu}^{(n+1)}$;
        if $\gamma^{(n)} \geq \bar{\gamma}$ **then** Set $\rho^{(n+1)} = 2\rho^{(n)}$;
      else
        Reject parameter $\tilde{\boldsymbol{\mu}}$;
        if $\rho^{(n)}/2 \leq \rho_{\mathsf{min}}$ ***or*** Skip_enrichment_flag$(n-1)$ **then**
          Enrich RO space at $\tilde{\boldsymbol{\mu}}$;
        else
          Shrink the TR radius $\rho^{(n+1)} = \rho^{(n)}/2$;
  Set $n = n + 1$;

Setting $\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu}^{(n)} + s$ we define our TR approximation of $(\hat{\mathrm{P}}_1)$ at iteration n as

$$\min_{\tilde{\boldsymbol{\mu}} \in \mathcal{P}_{\mathsf{ad}}} \hat{J}^{\ell,(n)}(\tilde{\boldsymbol{\mu}}) \quad \text{s.t.} \quad \zeta^{(n)}(\tilde{\boldsymbol{\mu}}) := \frac{\Delta_{\hat{J}}^{\ell,(n)}(\tilde{\boldsymbol{\mu}})}{\hat{J}^{\ell,(n)}(\tilde{\boldsymbol{\mu}})} \leq \rho^{(n)}. \tag{2.25}$$

Here and whenever some quantity depends on the iteration n, we show it in the superscript, like the RB cost $\hat{J}^{\ell,(n)}$ (i.e., the RO cost at the n-th iteration of the TR optimization method). The term $\zeta^{(n)}$ quantifies the accuracy of the RB and we use it to define our TR optimization function.

How the RO model behaves during the optimization gives us information about the accuracy of the surrogate model, and we can decide to enrich the basis, reduce the TR radius (in case the model is not accurate enough), enlarge the TR radius (in case the model is already accurate enough), or even skip the enrichment process, in case the model is already accurate enough in the trust region. The overall algorithm stops when the approximated first-order optimality condition for the parameter optimization, namely when

$$g^h(\boldsymbol{\mu}^{(n+1)}) := \|\boldsymbol{\mu}^{(n+1)} - \mathcal{P}_{\mathcal{P}_{\mathsf{ad}}}(\boldsymbol{\mu}^{(n+1)} - \nabla \hat{J}^h(\boldsymbol{\mu}^{(n+1)}))\|_2 \leq \tau,$$

is satisfied.

Create the first RO space: As in the greedy algorithm, the initialization of the RO space is done through the POD method at $\boldsymbol{\mu}^{(0)}$. As before, we choose the eigenvectors corresponding to the biggest L eigenvalues, where L is the smallest integer such that $\Delta_{\mathrm{y}}^{\ell,K}(\boldsymbol{\mu}) \leq \tau_{\mathrm{y}}$, where $\tau_{\mathrm{y}} = 10^{-4}$.

Enrich the RO space: Also the enrichment follows the strategy described previously, and again, we add L^+ new basis elements such that $\Delta_{\mathrm{y}}^{\ell,K}(\hat{\boldsymbol{\mu}}) \leq \tau_{\mathrm{y}}$ holds. However, enriching at each iteration can be counterproductive if the RO dimensions grows too much. For this reason, following [6] we define a Boolean flag that checks the accuracy of the RO model and decide when it is time to enrich the RO space. This flag at iteration n is True when all three these conditions are satisfied:

$$\zeta^{(n)}(\boldsymbol{\mu}^{(n+1)}) \leq \nu \rho^{(n+1)} \tag{2.26a}$$

$$\frac{|g^h(\boldsymbol{\mu}^{(n+1)}) - g^{\ell,(n)}(\boldsymbol{\mu}^{(n+1)})|}{g^{\ell,(n)}(\boldsymbol{\mu}^{(n+1)})} \leq \tau_{\mathsf{g}} \tag{2.26b}$$

$$\frac{\|\nabla \hat{J}^h(\boldsymbol{\mu}^{(n+1)}) - \nabla \hat{J}^{\ell,(n)}(\boldsymbol{\mu}^{(n+1)})\|_2}{\|\nabla \hat{J}^h(\boldsymbol{\mu}^{(n+1)})\|_2} \leq \min\left\{\tau_{\mathsf{grad}}, \nu \rho^{(n+1)}\right\} \tag{2.26c}$$

with $\nu \in (0,1)$ and $g^{\ell,(n)}(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \mathcal{P}_{\mathcal{P}_{\mathsf{ad}}}(\boldsymbol{\mu} - \nabla \hat{J}^{\ell,(n)}(\boldsymbol{\mu}))\|_2$. Inequality (2.26a) indicates that our parameter is well inside the trust region, inequality (2.26b) checks the accuracy of the convergence criterion in the RO model, and inequality (2.26c) checks the RB accuracy of the cost gradient (which gives the descent direction).

TR sub-problem: In each iteration n of the TR optimization, we consider the problem (2.25), called the *TR sub-problem*. A *proposed parameter* $\tilde{\boldsymbol{\mu}}$, solution of (2.25), is evaluated but not necessarily accepted, and in later parts of the algorithm it is

decided whether to finally accept this parameter as the next iteration or to reject it and modify the RO approximation.

In [33] the problem (2.25) is solved by considering the projected Armijo-BFGS algorithm (we refer to [34] for the details), but we apply a more black-box approach, namely plugging the optimization function $\hat{J}^{\ell,(n)}(\boldsymbol{\mu})$, its gradient $\hat{J}^{\ell,(n)}(\boldsymbol{\mu})$, and the nonlinear constraint $\zeta^{(n)}(\boldsymbol{\mu}) \leq \rho^{(n)}$ into the function `minimize` of the library `scipy.optimize`.

Accept or reject the parameter: The acceptance or rejection of the candidate parameter is based on the *error-aware sufficient decrease condition* (EASDC), introduced in [51], which reads

$$\hat{J}^{\ell,(n+1)}(\boldsymbol{\mu}^{(n+1)}) \leq \hat{J}^{\ell,(n)}(\boldsymbol{\mu}^{(n)}_{\mathsf{AGC}}), \tag{2.27}$$

where $\boldsymbol{\mu}^{(n)}_{\mathsf{AGC}}$ is the *approximate generalized Cauchy point*, namely any point satisfying the sufficient decrease condition for the RO model in a descent direction. In particular, this is evaluated by

$$\boldsymbol{\mu}^{(n)}_{\mathsf{AGC}} = \mathcal{P}_{\mathcal{P}_{\mathsf{ad}}}\left(\boldsymbol{\mu}^{(n)} - \alpha_{\mathsf{AGC}} \nabla \hat{J}^{\ell,(n)}(\boldsymbol{\mu}^{(n)})\right),$$

where the projection $\mathcal{P}_{\mathcal{P}_{\mathsf{ad}}}$ onto $\mathcal{P}_{\mathsf{ad}}$ is defined as

$$\mathcal{P}_{\mathcal{P}_{\mathsf{ad}}}(\mu)_i = \begin{cases} \mu_i^a & \text{if } \mu_i < \mu_i^a \\ \mu_i^b & \text{if } \mu_i > \mu_i^b \\ \mu_i & \text{otherwise} \end{cases} \qquad i = 1, \dots, d$$

and $\alpha_{\mathsf{AGC}} := (1/2)^{j_{\mathsf{AGC}}}$, where j_{AGC} is the biggest integer such that the following Armijo sufficient decrease condition

$$\hat{J}^{\ell,(n)}(\mu^{(n)}_{\mathsf{AGC}}) - \hat{J}^{\ell,(n)}(\mu^{(n)}) \leq -\frac{\alpha_{\mathsf{arm}}}{\alpha_{\mathsf{AGC}}} \left\| \mu^{(k)}_{\mathsf{AGC}} - \mu^{(k)} \right\|_2^2,$$

with $\alpha_{\mathsf{arm}} = 10^{-4}$, is satisfied.

We highlight that the term $\hat{J}^{\ell,(n)}$ refers to the reduced model at iteration n, while $\hat{J}^{\ell,(n+1)}$ refers instead to the model after the $(n+1)$-th (eventual) enrichment. It is clear that this condition is not straightforward to verify, and we would like to avoid the enrichment if not needed, since it involves evaluating the FO solution. To ensure that (2.27) is satisfied for the proposed parameter $\tilde{\boldsymbol{\mu}}$, a sufficient condition is analyzed:

$$\hat{J}^{\ell,(n)}(\tilde{\boldsymbol{\mu}}) + \Delta^{\ell,(n)}_{\hat{J}}(\tilde{\boldsymbol{\mu}}) < \hat{J}^{\ell,(n)}(\boldsymbol{\mu}^{(n)}_{\mathsf{AGC}}). \tag{2.28}$$

If (2.28) is satisfied, then the candidate is accepted, i.e. $\boldsymbol{\mu}^{(n+1)} := \tilde{\boldsymbol{\mu}}$. On the other hand, if it is not satisfied, we check a necessary condition for (2.27):

$$\hat{J}^{\ell,(n)}(\tilde{\boldsymbol{\mu}}) - \Delta^{\ell,(n)}_{\hat{J}}(\tilde{\boldsymbol{\mu}}) < \hat{J}^{\ell,(n)}(\boldsymbol{\mu}^{(n)}_{\mathsf{AGC}})$$

If this also fails, it means that the point $\tilde{\boldsymbol{\mu}}$ probably needs an enhancement too big to satisfy the error-aware sufficient decrease condition (2.27). Then it must be rejected, and the RO space must be enriched. If, on the other hand, the sufficient condition fails and the necessary condition holds, we check the value of terms like the $\mathsf{Skip\_enrichment\_flag}$ and the real reduction $\hat{J}^h(\tilde{\boldsymbol{\mu}}) - \hat{J}^{\ell,(n)}(\boldsymbol{\mu}^{(k)}_{\mathsf{AGC}})$ to decide whether to accept the point or not.

Modification of the trust region: During the acceptance or rejection of the candidate parameter, we might realize that the candidate is on the border of the trust region; since the model is not accurate there, we could shrink the TR radius and find a new candidate or just enrich RO space. On the other hand, if the RO model keeps being accurate along the optimization path, we can enlarge the TR radius in order to hasten the converge.

In general, a measure of accuracy is given by the predicted sufficient reduction:

$$\gamma^{(n)} = \frac{\hat{J}^h(\boldsymbol{\mu}^{(n)}) - \hat{J}^h(\boldsymbol{\mu}^{(n+1)})}{\hat{J}^{\ell,(n)}(\boldsymbol{\mu}^{(n)}) - \hat{J}^{\ell,(n)}(\boldsymbol{\mu}^{(n+1)})} \geq \bar{\gamma}, \tag{2.29}$$

where $\bar{\gamma} \in [3/4, 1)$. This condition is called sufficient predicted reduction, because it confronts the reduction of cost $\hat{J}^h$ in the new parameter with the reduction of the RO model $\hat{J}^{\ell,(n)}$.

Convergence of the algorithm: The convergence of the algorithm to a first-order critical point of $\hat{J}^h$ is showed in [33] and [6], Theorem 3.8, for the elliptic case. In our case the algorithm is the same, so we just have to make sure that the assumptions required for the convergence are satisfied. First of all, Assumption I of parameter separability of $a_{\boldsymbol{\mu}}$ and $f_{\boldsymbol{\mu}}$ is satisfied in (2.4). Then, we can assume $a_{\boldsymbol{\mu}}$, $f_{\boldsymbol{\mu}}$ and J to be twice continuously Fréchet differentiable w.r.t. $\boldsymbol{\mu}$, as in Assumption II, which requires all the continuous functions ϑ_l^a and ϑ_l^f to be regular enough for $l = 1, \dots, m_a$ and $l = 1, \dots, m_f$, respectively. Assumption III simply requires the FE discretization to be accurate enough so that we can neglect its error given by the FO approximation. Assumption IV requires the cost functional $J(y, \boldsymbol{\mu})$ to be strictly positive for all $y \in \mathcal{Y}$ and $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$, which we can guaranteed by adding 1 in the cost formula, namely defining

$$\hat{J}(\boldsymbol{\mu}) := 1 + \frac{\alpha_J}{2} \int_0^T \|y_{\boldsymbol{\mu}} - w\|_H^2 \, \mathrm{d}t + \frac{\beta_J}{2} \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2^2, \tag{2.30}$$

and defining the FO and RO approximations accordingly. Assumption V is a technical safeguard that assumes that there exists a radius bigger than machine precision such that there exists a candidate $\tilde{\boldsymbol{\mu}}$ solution of (2.25) satisfying (2.27). The minimum radius δ_{min} is introduced to avoid that the radius reaches machine precision. Assumption VI requires the RO gradient to be uniformly Lipschitz-continuous and its second derivatives to be locally Lipschitz-continuous, guaranteed in our case by the quadratic form (2.16). We also assume the function $\zeta^{(n)}$ to be uniformly continuous in $\mathcal{P}_{\mathsf{ad}}$ uniformly in k as in Assumption VII, which is in our case satisfied by the continuity of $\zeta^{(n)}$ and the compactness of $\mathcal{P}_{\mathsf{ad}}$.

Finally, let us make a comment on Lemma 3.4 of [6]. In the elliptic case, every time the RO space is enriched the following conditions are satisfied

$$\zeta^{(n+1)}(\boldsymbol{\mu}^{(n+1)}) = 0 \qquad \text{and} \qquad \frac{\left| g^{\ell,(n+1)}(\boldsymbol{\mu}^{(n+1)}) - g^h(\boldsymbol{\mu}^{(n+1)}) \right|}{g^{\ell,(n)}(\boldsymbol{\mu}^{(n+1)})} = 0,$$

but in our case this is not the case. For us it is sufficient, though, to require

$$\zeta^{(n+1)}(\boldsymbol{\mu}^{(n+1)}) \leq \nu \rho^{(n+1)}$$

and

$$\frac{\left|g^{\ell,(n+1)}(\boldsymbol{\mu}^{(n+1)}) - g^{h}(\boldsymbol{\mu}^{(n+1)})\right|}{g^{\ell,(n)}(\boldsymbol{\mu}^{(n+1)})} \leq \tau_g$$

every time the model is enriched (i.e. we enforce (2.26a) and (2.26b)) to satisfy Lemma 3.4.

We will now show an application of this optimization method to a problem of the type (2.2). Let us consider, in particular, the same problem and setting in Example 2.3. Furthermore, we define the cost function parameters $\alpha_J := 1$, $\beta_J := 10^{-13}$, and the initial and reference parameter $\boldsymbol{\mu}^{(0)} = \hat{\boldsymbol{\mu}} := (4, 5)$. As data we use a noisy virtual measurement of the state variable corresponding to the hidden parameter $\boldsymbol{\mu}^* := (1, 2)$, namely

$$\mathrm{w}_j^{h,k} := \mathrm{y}_{\boldsymbol{\mu}^* j}^{h,k} + \varepsilon_j^k, \quad \text{where } \varepsilon_j^k \sim \mathcal{N}(0, \sigma_{\mathsf{d}}^2) \quad \text{for any } j = 1, \ldots, N,\ k = 0, \ldots, K, \tag{2.31}$$

where $\sigma_{\mathsf{d}} := 10^{-3}$ is the simulated measurement standard deviation. The algorithm parameters defined in this section are given as

Algorithm tolerance $\tau := 10^{-4}$,
Initial TR radius $\rho_0 := 0.1$,
Maximum number of TR iterations $N_{\mathsf{maxit}} := 30$
Minimum admissible TR radius $\rho_{\mathsf{min}} := 0.1$,
State tolerance for the creation and enrichment $\tau_{\mathrm{y}} := 10^{-4}$,
Parameter in (2.26a) $\nu := 0.5$,
Parameter in (2.26b) $\tau_{\mathsf{g}} := 10^{-2}$,
Parameter in (2.26c) $\tau_{\mathsf{grad}} := 10^{-2}$,
Parameter in (2.29) $\bar{\gamma} := 0.75$.

We compare the performance of the trust region optimization algorithm with the results of the FO optimization problem solved with the function `fmin_l_bfgs_b` from the `scipy.optimize` library (cf., e.g., [10]). The results are shown in Table 2.1, indicating CPU time, number of iterations, number of FO evaluations and error from the true parameter. In particular, these are the norms of the absolute and relative error between the true hidden parameter and the solution of the parameter optimization algorithm, i.e. $\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{abs}} := \|\boldsymbol{\mu}^* - \boldsymbol{\mu}^{\mathsf{opt}}\|_2$ and $\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{rel}} := \|\boldsymbol{\mu}^* - \boldsymbol{\mu}^{\mathsf{opt}}\|_2 / \|\boldsymbol{\mu}^*\|_2$.

	Time	Iterations	FO evaluations	$\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{abs}}$	$\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{rel}}$
FO optimization	9.9 s	34	89	0.0002	0.00009
TR-RB	3.9 s	20	2	0.00039	0.00018

Table 2.1: Comparison between the trust region and the FO optimization.

We observe that the speedup of the TR optimization algorithm in the overall CPU time is about 2.5, factor given by the lower number of iterations and FO evaluations. This will certainly bring a bigger advantage in those cases where the FO evaluation gets too computationally expensive. Since in this case we are dealing with a fast solver for a rather

simple model, we do not notice such big improvement; on the other hand, the speedup will be more evident in the numerical results of the next chapter.

Regarding the accuracy of the optimization, in both cases the algorithm stops sufficiently close to the true parameter. To show the accuracy of the RB approximation, in Figure 2.3 the error between the FO and RO states approximations corresponding to the hidden parameter $\boldsymbol{\mu}^*$ is plotted.

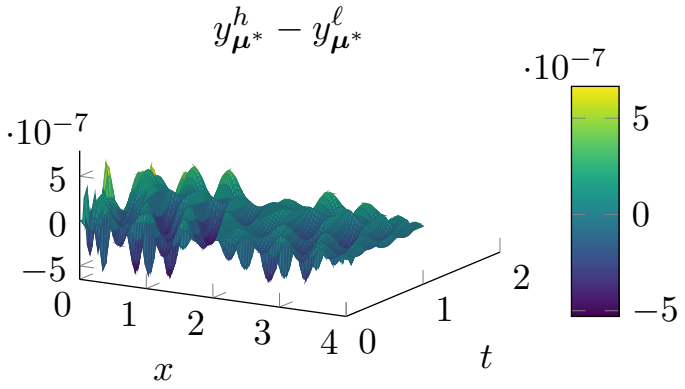


Figure 2.3: Errors between FO and RO approximations.

Chapter 3

Nonlinear coupled model

In this chapter we introduce a new model, more similar to a battery cell system and therefore with a more complex structure. In particular, we want to emulate the dynamics of the concentration of some material in a one-dimensional domain pushed in one direction or the other by some potential energy, which is excited by a control function acting on its boundary. This system can be seen as a drastic over-simplification of an electro-chemical system modeling the concentration of Lithium ions and the electric potential excited by some current density applied to the boundary. For the coupling term we use an exponential nonlinear mapping that is inspired by the renowned Butler-Volmer equation in electrochemistry (see, e.g., [17]).

Battery systems can be modeled in different ways. For example, in [48], a simplified system is driven by a system of ODEs, while elliptic-parabolic systems appear, e.g. in [68, 36, 37, 61]. In these works three states are usually considered, the lithium-ion concentration, the electric potential in the solid phase and the electric potential in the electrolyte phase. We, on the other hand, will only use one state for the electric potential (as in, e.g., [67, 43]), and will not divide the interval Ω in the three battery domain (cathode, separator, anode). This can be done in a later step.

We have worked with this – still simplified – battery system for two reason: on the mathematical side, our aim was to develop online-efficient a-posteriori error estimates, especially hard to evaluate in this case, so that we could use the trust region algorithm. On the other hand, we want to use this model in the input design problem, which is independent of the model used. Hence a different, more complex model can be used in the input design framework that we discuss in the next chapter.

The approximated a-posteriori estimates that we evaluate in [4] are based on the bounds for the elliptic problem in [29]. In this way, we are only able to evaluate approximated estimates, but the trust region algorithm allows for the TR constraint to be inexact; cf., e.g., the so-called *relaxed trust region method* in [32].

This chapter is organized as follows: in Section 3.1 we study the well-posedness and local existence in time of the elliptic-parabolic system and discuss the optimization problem. The full and reduced order approximations are defined in Section 3.2 and Section 3.3,

respectively. The approximated hierarchical a-posteriori error estimator is derived and tested numerically in Section 3.4. The modification of the trust region algorithm is then considered in Section 3.5, where also numerical experiments are presented.

3.1 State equations and optimization problem

As previously, let $T > 0$ be the (finite) time horizon, $\Omega := (0, L) \subset \mathbb{R}$ be a space interval and let $Q_T := (0, T) \times (0, L)$. We consider the following parameter-dependent parabolic-elliptic coupled system for the two state variables $y, q : Q_T \to \mathbb{R}$

$$y_t(t,x) - \mu_1\big(\kappa_1(x)y_x(t,x)\big)_x - \mu_2 f(y(t,x), q(t,x)) = 0 \qquad \text{f.a.a. } (t,x) \in Q_T, \tag{3.1a}$$

$$-\mu_3(\kappa_2(x)q_x(t,x))_x + \mu_4 f(y(t,x), q(t,x)) = 0 \qquad \text{f.a.a. } (t,x) \in Q_T \tag{3.1b}$$

with homogeneous Neumann boundary conditions

$$y_x(t,0) = y_x(t,L) = 0 \quad \text{f.a.a. } t \in (0,T), \tag{3.1c}$$

inhomogeneous Dirichlet-Neumann mixed boundary conditions

$$q(t,0) = 0 \text{ f.a.a. } t \in (0,T), \quad \mu_3\kappa_2(L)q_x(t,L) = u(t) \text{ f.a.a. } t \in (0,T) \tag{3.1d}$$

and initial conditions

$$y(0,x) = y_\circ(x) \quad \text{f.a.a. } x \in \Omega. \tag{3.1e}$$

In the following, we will fix some assumptions for (3.1) that are necessary for our existence results.

Assumption 1.

(1) For given parameter bounds $\boldsymbol{\mu}_\mathsf{a} = (\mu_{\mathsf{a},i})_{1\le i\le 4}$ and $\boldsymbol{\mu}_\mathsf{b} = (\mu_{\mathsf{b},i})_{1\le i\le 4}$ satisfying $0 < \boldsymbol{\mu}_\mathsf{a} \le \boldsymbol{\mu}_\mathsf{b}$ in $\mathcal{P} = \mathbb{R}^4$ an admissible parameter vector $\boldsymbol{\mu} = (\mu_i)_{1\le i\le 4}$ lies in the nonempty, compact and convex set $\mathcal{P}_\mathsf{ad} = \{\boldsymbol{\mu} \in \mathcal{P} : \boldsymbol{\mu}_\mathsf{a} \le \boldsymbol{\mu} \le \boldsymbol{\mu}_\mathsf{b} \text{ in } \mathcal{P}\} \subset \mathcal{P}$. Here, '$\le$' is understood component-wise.

(2) The initial condition $y_\circ$ belongs to $H^1(\Omega)$ and is positive on $\overline{\Omega}$, namely $y_\circ(x) \ge y_\mathsf{a}$ for all $x \in \overline{\Omega}$ and a positive constant y_a.

(3) For $\mathcal{U} := L^2(0,T)$ the set of admissible inputs is $\mathcal{U}_\mathsf{ad} := \{u \in \mathcal{U} : u_\mathsf{a}(t) \le u(t) \le u_\mathsf{b}(t) \text{ f.a.a. } t \in [0,T]\}$ with $u_\mathsf{a}, u_\mathsf{b} \in L^\infty(0,T)$ satisfying $u_\mathsf{a} \le u_\mathsf{b}$ a.e. in $[0,T]$. In particular, $\mathcal{U}_\mathsf{ad} \subset L^\infty(0,T)$ holds, and we have $\|u\|_{L^\infty(0,T)} \le c_\mathcal{U}$ for all $u \in \mathcal{U}$ with $c_\mathcal{U} = \max\{\|u_\mathsf{a}\|_{L^\infty(0,T)}, \|u_\mathsf{b}\|_{L^\infty(0,T)}\}$.

(4) The functions κ_1, κ_2 belong to $C^{0,1}(\overline{\Omega})$ with $\kappa_1(x) \ge \kappa_\mathsf{a}$ and $\kappa_2(x) \ge \kappa_\mathsf{a}$ for all $x \in \overline{\Omega}$ and a positive constant κ_a.

(5) The nonlinearity is defined as $f(\mathrm{y}, \mathrm{q}) := \sqrt{\mathrm{y}}\sinh(\mathrm{q})$ for $\mathrm{y} \in \mathbb{R}_\ge := \{s \in \mathbb{R} \,|\, s \ge 0\}$ and $\mathrm{q} \in \mathbb{R}$.

Remark 3.1.

1) Recall that $H^1(\Omega) \hookrightarrow C(\overline{\Omega})$ holds (cf. [19]). Thus, Assumption 1-(2) implies $y_\circ \in C(\overline{\Omega})$.

2) Note that f is not differentiable at $\mathrm{y} = 0$. In our application, the state y stands for the concentration of lithium-ions in a battery cell. Thus, the situation $\mathrm{y} = y(t,x) \leq 0$ does not have any physical meaning. This non-negativity of y is needed to evaluate $f(\mathrm{y},\mathrm{q})$. ◊

3.1.1 Weak formulation

As before, let $H := L^2(\Omega)$ and $V := H^1(\Omega)$, endowed with the $L^2(\Omega)$ and $H^1(\Omega)$ inner products, respectively. We further define the Hilbert space

$$V_\circ := \{\varphi \in V \,|\, \varphi(0) = 0\}$$

supplied with the inner product

$$\langle \varphi, \phi \rangle_{V_\circ} := \int_\Omega \varphi'(x)\phi'(x)\,\mathrm{d}x \quad \text{for } \varphi, \phi \in V_\circ.$$

Furthermore, let $C_\circ(\overline{\Omega}) = \{\varphi \in C(\overline{\Omega}) : \varphi(0) = 0\}$ supplied with the $C(\overline{\Omega})$-norm, i.e., $\|\varphi\|_{C(\overline{\Omega})} = \max_{x\in\overline{\Omega}} |\varphi(x)|$. Since $V \hookrightarrow C(\overline{\Omega})$ holds, there exists an embedding constant $c_{\mathsf{e}} > 0$ (only dependent on the interval Ω) satisfying

$$\|\varphi\|_{C(\overline{\Omega})} \leq c_{\mathsf{e}} \|\varphi\|_V \quad \text{for all } \varphi \in V \tag{3.2}$$

with $\|\varphi\|_{C(\overline{\Omega})} = \max_{x\in\overline{\Omega}} |\varphi(x)|$. Furthermore, Poincaré's inequality holds on the space $V_\circ$ (see [19, Theorem 3, Page 279]): there exists a constant $c_{\mathsf{P}} > 0$ such that

$$\|\varphi\|_H \leq c_{\mathsf{P}} \|\varphi\|_{V_\circ} \quad \text{for all } \varphi \in V_\circ. \tag{3.3}$$

From (3.2) and (3.3) we infer that

$$\|\varphi\|^2_{C(\overline{\Omega})} \leq c_{\mathsf{e}}^2 \|\varphi\|^2_V = c_{\mathsf{e}}^2 \left(\|\varphi\|^2_H + \|\varphi\|^2_{V_\circ} \right) \leq c_{\mathsf{e}}^2 \left(c_{\mathsf{P}}^2 + 1 \right) \|\varphi\|^2_{V_\circ}.$$

Hence, for $c_{\mathsf{eP}} = c_{\mathsf{e}}(c_{\mathsf{P}}^2 + 1)^{1/2} > 0$ it holds that

$$\|\varphi\|_{C(\overline{\Omega})} \leq c_{\mathsf{eP}} \|\varphi\|_{V_\circ} \quad \text{for all } \varphi \in V_\circ. \tag{3.4}$$

Next, we define the solution spaces

$$\mathcal{Y}^T := W(0,T;V,V^*) \cap C(\overline{Q}_T) \quad \text{and} \quad \mathcal{Q}^T := L^\infty(0,T;V_\circ)$$

endowed by their product topology, where $W(0,T;X,Y) := L^2(0,T;X) \cap H^1(0,T;Y)$; see, e.g., [14] for more details.

Now, the weak formulation of (3.1) is as follows: for $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$ and $u \in \mathcal{U}_{\mathsf{ad}}$ find a solution pair $(y,q) \in \mathcal{Y}^T \times \mathcal{Q}^T$ such that

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle y(t), \varphi^y \rangle_H + a^y_{\boldsymbol{\mu}}(y(t), \varphi^y) + \langle g^y_{\boldsymbol{\mu}}[y(t), q(t)], \varphi^y \rangle_{V^*,V} = 0, \tag{3.5a}$$

$$y(0) = y_\circ \qquad \text{in } H, \tag{3.5b}$$

$$a^q_{\boldsymbol{\mu}}(q(t), \varphi^q) + \langle g^q_{\boldsymbol{\mu}}[y(t), q(t)], \varphi^q \rangle_{V_\circ^*,V_\circ} = \langle b(t), \varphi^q \rangle_{V_\circ^*,V_\circ}, \tag{3.5c}$$

for all $\varphi^y \in V$, $\varphi^q \in V_\circ$ and f.a.a. $t \in (0,T]$, where the bilinear forms are defined as

$$a^y_{\boldsymbol{\mu}}(\varphi,\phi) := \mu_1 \int_\Omega \kappa_1(x)\varphi'(x)\phi'(x)\,\mathrm{d}x =: \mu_1\, \hat{a}^y(\varphi,\phi) \qquad \text{for } \varphi,\phi \in V, \tag{3.6a}$$

$$a^q_{\boldsymbol{\mu}}(\varphi,\phi) := \mu_3 \int_\Omega \kappa_2(x)\varphi'(x)\phi'(x)\,\mathrm{d}x =: \mu_3\, \hat{a}^q(\varphi,\phi) \qquad \text{for } \varphi,\phi \in V_\circ, \tag{3.6b}$$

the nonlinear operators are given by

$$\langle g^y_{\boldsymbol{\mu}}[y(t),q(t)],\varphi\rangle_{V^*,V} := \mu_2 \int_\Omega (-f(y(t,x),q(t,x)))\varphi(x)\,\mathrm{d}x \qquad \text{for } \varphi \in V, \tag{3.7a}$$

$$\langle g^q_{\boldsymbol{\mu}}[y(t),q(t)],\varphi\rangle_{V_\circ^*,V_\circ} := \mu_4 \int_\Omega f(y(t,x),q(t,x))\varphi(x)\,\mathrm{d}x \qquad \text{for } \varphi \in V_\circ, \tag{3.7b}$$

Finally, the Neumann boundary condition appears in the linear operator as

$$\langle b(t),\varphi\rangle_{V_\circ^*,V_\circ} := u(t)\varphi(L) \qquad \text{for } \varphi \in V_\circ \text{ and } t \in (0,T] \text{ a.e.}$$

We further define $\hat{g}^y$ and $\hat{g}^q$ as

$$\langle \hat{g}^y[\varphi^y,\varphi^q],\varphi\rangle_{V^*,V} := \int_\Omega (-f(\varphi^y(x),\varphi^q(x)))\varphi(x)\,\mathrm{d}x \qquad \text{for } \varphi^y \in V,\ \varphi^q \in V_\circ,\ \varphi \in V,$$

$$\langle \hat{g}^q[\varphi^y,\varphi^q],\varphi\rangle_{V_\circ^*,V_\circ} := \int_\Omega f(\varphi^y(x),\varphi^q(x))\varphi(x)\,\mathrm{d}x \qquad \text{for } \varphi^y \in V,\ \varphi^q \in V_\circ,\ \varphi \in V_\circ,$$

and then we can rewrite (3.7) as

$$\langle g^y_{\boldsymbol{\mu}}[y(t),q(t)],\varphi\rangle_{V^*,V} = \mu_2\langle \hat{g}^y[y(t),q(t)],\varphi\rangle_{V^*,V} \qquad \text{for } \varphi \in V,$$

$$\langle g^q_{\boldsymbol{\mu}}[y(t),q(t)],\varphi\rangle_{V_\circ^*,V_\circ} := \mu_4\langle \hat{g}^q[y(t),q(t)],\varphi\rangle_{V_\circ^*,V_\circ} \qquad \text{for } \varphi \in V_\circ.$$

3.1.2 Well-posedness

We discuss briefly the well-posedness of the problem, analyzed thoroughly in [4]. The well-posedness of the two different state variables is proved using the Schauder and the Leray-Schauder fixed point theorems (cf., e.g. [22, p. 189]). These theorems consider the solution operator of the decoupled, linear, weak systems, and find a solution through a fixed point argument. We summarized the main assumptions and existence theorem, together with some stability estimates.

Suppose, initially, that $y \in \mathcal{Y}^T$ with $y > 0$ in Q_T is given. For every $u \in \mathcal{U}_{\mathsf{ad}}$ and f.a.a. $t \in [0,T]$ we consider the nonlinear elliptic problem

$$-\mu_3\,(\kappa_2 q_x(t))_x + \mu_4 f(y(t),q(t)) = 0 \text{ a.e. in } \Omega, \quad q(t,0) = 0, \quad \mu_3\kappa_2(L)q_x(t,L) = u(t). \tag{$\mathbf{E}$}$$

A weak solution of ($\mathbf{E}$) satisfies f.a.a. $t \in [0,T]$

$$\mu_3 \int_\Omega \kappa_2(x)q_x(t,x)\varphi'(x)\,\mathrm{d}x + \mu_4 \int_\Omega f(y(t,x),q(t,x))\varphi(x)\mathrm{d}x = u(t)\varphi(L) \quad \forall \varphi \in V_\circ. \tag{$\mathbf{E}_{\mathsf{w}}$}$$

In order to define the fixed point problem, the following hypothesis is needed.

Assumption 2. For a given $M > 1$ let y belong to the non-empty, closed, bounded, convex set $\mathcal{Y}_M^T$ defined as

$$\mathcal{Y}_M^T := \left\{ y \in C(\overline{Q}_T) : \frac{1}{M} \leq y(t,x) \leq M \text{ for all } (t,x) \in \overline{Q}_T \right\}. \tag{3.8}$$

Then, by using the properties of the solution operator of the de-coupled elliptic equation and the solution space, it is possible to prove the following theorem ([4, Theorem 2.2]):

Theorem 3.2. *Let Assumptions* 1 *and* 2 *hold. Then, for every* $u \in \mathcal{U}_{\mathsf{ad}}$ *there exists a unique solution* $q \in \mathcal{Q}^T$ *of* $(\mathbf{E}_{\mathsf{w}})$ *and a constant* $c(M) > 0$ *(independent of* q *or* u *but dependent on* M*) such that*

$$\|q(t)\|_{C(\overline{\Omega})} \leq c_{\mathsf{eP}} \|q(t)\|_{V_\circ} \leq c(M)|u(t)| \leq c(M)c_{\mathcal{U}} \quad \text{for } t \in [0,T] \text{ a.e.,} \tag{3.9}$$

where the constant c_{eP} *has already been introduced in* (3.4)*, and* $c_{\mathcal{U}}$ *in Assumption 1.*

Existence of the weak solution for the coupled system (3.1) is proved in the theorem ([4, Theorem 2.3]), by using the properties of the solution operator of the de-coupled parabolic equation.

Theorem 3.3. *Let Assumptions* 1 *and* 2 *hold and the constant* M *in Assumption* 2 *be chosen as*

$$M = 2\|y_\circ\|_{C(\overline{\Omega})} + \frac{2}{y_{\mathsf{a}}}. \tag{3.10}$$

Then, there exists a finite time $T_\circ = T_\circ(M) \in (0,T]$ *such that* (3.1) *admits a unique solution pair* $(y,q) \in \mathcal{Y}_M^{T_\circ} \times \mathcal{Q}^{T_\circ}$.

Remark 3.4. Note that, due to (3.10) we can see that $y_\circ \leq M/2$ and $y_\circ \geq y_{\mathsf{a}} \geq 2/M$ for all $x \in \overline{\Omega}$. ◊

3.1.3 PDE-constrained optimization problem

As in the previous chapter, we introduce the nonlinear PDE-constrained parameter optimization problem

$$\min J(y,q,\boldsymbol{\mu}) \quad \text{s.t.} \quad (y,q) \in \mathcal{Y} \times \mathcal{Q} \text{ is a weak solution of (3.1) and } \boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}. \tag{P$_2$}$$

Problem (3.5) is uniquely solvable only locally in time. For that reason, we make use of the following hypothesis.

Assumption 3. For given final time $T > 0$ there exists a unique solution pair $(y,q) \in \mathcal{Y} \times \mathcal{Q}$ to (3.5) for any $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$ which is denoted as $(y_{\boldsymbol{\mu}}, q_{\boldsymbol{\mu}})$.

Choosing a general cost function, this will be the $L^2(0,T;H)$ norm of the error between some observable $z \in L^2(0,T;H)$ depending (eventually) on the state variables and the parameter, i.e. $z = z(y,q,\boldsymbol{\mu})$, and a given data function $w \in L^2(0,T;H)$ that depends on $\boldsymbol{\mu}^* \in \mathcal{P}_{\mathsf{ad}}$, the underlying parameter, plus a regularization term. Then,

$$J(y,q,\boldsymbol{\mu}) := 1 + \frac{\alpha_J}{2}\|z(y,q,\boldsymbol{\mu}) - w\|^2_{L^2(0,T;H)} + \frac{\beta_J}{2}\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2^2 \quad \text{for } \boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}, \tag{3.11}$$

where $\alpha_J > 0$, $\beta_J > 0$ hold, $\hat{\boldsymbol{\mu}} \in \mathcal{P}_{\mathsf{ad}}$ is a reference parameter, and the 1 is added to be compatible with (2.30) and the trust region optimization algorithm. Then, setting $\hat{z}_{\boldsymbol{\mu}} := z(y_{\boldsymbol{\mu}}, q_{\boldsymbol{\mu}}, \boldsymbol{\mu})$, we define the reduced cost function as

$$\hat{J}(\boldsymbol{\mu}) := 1 + \frac{\alpha_J}{2} \|\hat{z}_{\boldsymbol{\mu}} - w\|^2_{L^2(0,T;H)} + \frac{\beta_J}{2} \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2^2. \tag{3.12}$$

In many applications we only have limited information and the observable function cannot be chosen arbitrarily. For example, in battery systems there is no possibility of measuring the concentration of Lithium ions in a certain electrode, and one of the few measurable quantities is the electric potential difference, measuring the energy change between two points. We will try mimicking this reduced information setting, by assuming that we can only observe and measure the q-state, i.e. $\hat{z}_{\boldsymbol{\mu}} := q_{\boldsymbol{\mu}}$. In the last section of the chapter, we see a case where we can observe an even more limited information. Finally, we rewrite the optimization problem in reduced form as

$$\min \hat{J}(\boldsymbol{\mu}) \quad \text{s.t.} \quad \boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}. \tag{$\hat{\mathrm{P}}_2$}$$

The *adjoint variables* $p_y = p_{y,\boldsymbol{\mu}} \in \mathcal{Y}$ and $p_q = p_{q,\boldsymbol{\mu}} \in \mathcal{Q}$ satisfy the linear system in weak form

$$-\frac{\mathrm{d}}{\mathrm{d}t} \langle p_y(t), \varphi^y \rangle_H + \langle g^{py}_{\boldsymbol{\mu}}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t), p_y(t), p_q(t)], \varphi^y \rangle_{V^*,V} = 0, \tag{3.13a}$$

$$\langle p_y(T), \varphi^T \rangle_H = 0, \tag{3.13b}$$

$$\langle g^{pq}_{\boldsymbol{\mu}}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t), p_y(t), p_q(t)], \varphi^q \rangle_{V_\circ^*,V_\circ} = \langle b^p_{\boldsymbol{\mu}}(t), \varphi^q \rangle_{V_\circ^*,V_\circ}, \tag{3.13c}$$

for all $\varphi^y \in V$, $\varphi^q \in V_\circ$, $\varphi^T \in H$ and f.a.a. $t \in (0,T]$, where

$$\begin{aligned}
\langle g^{py}_{\boldsymbol{\mu}}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t), p_y(t), p_q(t)], \varphi^y \rangle_{V^*,V} &:= \mu_1 \hat{a}^y(p_1(t), \varphi^y) \\
&\quad - \mu_2 \langle \partial_y f(y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)) p_y(t), \varphi^y \rangle_H + \mu_4 \langle \partial_y f(y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)) p_q(t), \varphi^y \rangle_H, \\
\langle g^{pq}_{\boldsymbol{\mu}}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t), p_y(t), p_q(t)], \varphi^q \rangle_{V_\circ^*,V_\circ} &:= \mu_3 \hat{a}^q(p_q(t), \varphi^q) \\
&\quad - \mu_2 \langle \partial_q f(y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)) p_y(t), \varphi^q \rangle_H + \mu_4 \langle \partial_q f(y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)) p_2(t), \varphi^q \rangle_H, \\
\langle b^p_{\boldsymbol{\mu}}(t), \varphi^q \rangle_{V_\circ^*,V} &:= \alpha_J \langle w(t) - q_{\boldsymbol{\mu}}(t)), \varphi^q \rangle_H,
\end{aligned}$$

with $\partial_y f$ and $\partial_q f$ being the Fréchet derivative of f with respect to y and q.

Once we evaluate the adjoint variables, the gradient of the cost function (3.12) can be evaluated following the adjoint approach (cf., e.g., [31]) as

$$\nabla \hat{J}(\boldsymbol{\mu}) = \begin{pmatrix} \int_0^T \hat{a}^y(y_{\boldsymbol{\mu}}(t), p_{y,\boldsymbol{\mu}}(t)) \mathrm{d}t \\ -\int_0^T \langle f(y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)), p_{y,\boldsymbol{\mu}}(t) \rangle_H \mathrm{d}t \\ \int_0^T \hat{a}^q(q_{\boldsymbol{\mu}}(t), p_{q,\boldsymbol{\mu}}(t)) \mathrm{d}t \\ \int_0^T \langle f(y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)), p_{q,\boldsymbol{\mu}}(t) \rangle_H \mathrm{d}t \end{pmatrix} + \beta_J(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}). \tag{3.14}$$

3.2 Finite element discretization

Following the steps in Section 1.2.1 and Section 2.2, we discretize problem (3.5) in order to evaluate its numerical solution. Let us discuss discretization keeping $\boldsymbol{\mu}$ fixed in order not to burden the notation; we will later reintroduce the dependence on the parameter when defining the RO error estimates. First, we will introduce a standard Galerkin approximation, which leads to a high-dimensional nonlinear system of ordinary differential equations. Then, we will formulate the reduced order discretization.

Assume that $\{\varphi_i\}_{i=0}^N \subset V$ are linearly independent and that $\{\varphi_i\}_{i=1}^N \subset V_\circ$ holds. We furthermore define the finite dimensional subspaces $V^h = \operatorname{span}\{\varphi_0, \dots, \varphi_N\} \subset V$ and $V_\circ^h = \operatorname{span}\{\varphi_1, \dots, \varphi_N\}$. Then, by Galerkin projection of equations (3.5) onto V^h and $V_\circ^h$, our goal is to find $(y^h, q^h) \in H^1(0,T;V^h) \times L^\infty(0,T;V_\circ^h)$ solving

$$\frac{\mathrm{d}}{\mathrm{d}t}\langle y^h(t), \varphi^y\rangle_H + a^y(y^h(t), \varphi^y) + \langle g^y[y^h(t), q^h(t)], \varphi^y\rangle_{V^*,V} = 0, \tag{3.15a}$$

$$\langle y^h(0) - y_\circ^h, \varphi^y\rangle_H = 0, \tag{3.15b}$$

$$a^q(q^h(t), \varphi^q) + \langle g^q[y^h(t), q^h(t)], \varphi^q\rangle_{V_\circ^*,V_\circ} = \langle b(t), \varphi^q\rangle_{V_\circ^*,V_\circ} \tag{3.15c}$$

for all $\varphi^y \in V^h$, $\varphi^q \in V_\circ^h$, f.a.a. $t \in (0,T]$, where $y_\circ^h$ is a projection of $y_\circ$ on V^h given as

$$y_\circ^h = \operatorname{argmin}\left\{\|y_\circ - \varphi^h\|_H \,|\, \varphi^h \in V^h\right\}.$$

Note that

$$y^h(t) = \sum_{i=0}^N \mathrm{y}_i^h(t)\varphi_i, \qquad q^h(t) = \sum_{i=1}^N \mathrm{q}_i^h(t)\varphi_i \qquad \text{for } t \in [0,T]$$

so that (3.15) reduces into finding the coefficient vectors $\mathrm{y}^h(t) = (\mathrm{y}^h(t))_{0\le i\le N}$ and $\mathrm{q}^h(t) = (\mathrm{q}^h(t))_{1\le i\le N}$ solving the differential algebraic system

$$\mathrm{M_y}\dot{\mathrm{y}}^h(t) + \mu_1\mathrm{A_y}\mathrm{y}^h(t) - \mu_2\mathrm{M_y}\mathrm{f}_\mathrm{y}^h(\mathrm{y}^h(t), \mathrm{q}^h(t)) = 0 \qquad t \in (0,T] \text{ a.e.}, \tag{3.16a}$$

$$\mathrm{M_y}\mathrm{y}^h(0) = \mathrm{y}_\circ^h, \tag{3.16b}$$

$$\mu_3\mathrm{A_q}\mathrm{q}^h(t) + \mu_4\mathrm{M_q}\mathrm{f}_\mathrm{q}^h(\mathrm{y}^h(t), \mathrm{q}^h(t)) + \mathrm{b}^h(t) = 0 \qquad t \in [0,T] \text{ a.e.} \tag{3.16c}$$

for

$$\begin{aligned}
&\mathrm{M_y} = ((\langle\varphi_j, \varphi_i\rangle_H)), && \mathrm{A_y} = ((\hat{a}^y(\varphi_j, \varphi_i))) && \text{for } i,j = 0,\dots,N\\
&\mathrm{y}_\circ^h = (\langle y_\circ, \varphi_i\rangle_H) && && \text{for } i = 0,\dots,N,\\
&\mathrm{M_q} = ((\langle\varphi_j, \varphi_i\rangle_H)), && \mathrm{A_q} = ((\hat{a}^q(\varphi_j, \varphi_i))) && \text{for } i,j = 1,\dots,N,\\
&\mathrm{b}^h(t) = (\langle b(t), \varphi_i\rangle_{V_\circ',V_\circ}) && && \text{for } i = 1,\dots,N.
\end{aligned}$$

Moreover, $\mathrm{f}_\mathrm{y}^h(\mathrm{y}^h(t), \mathrm{q}^h(t)) \in \mathbb{R}^{N+1}$ (resp. $\mathrm{f}_\mathrm{q}^h(\mathrm{y}^h(t), \mathrm{q}^h(t)) \in \mathbb{R}^N$) is the coefficient vector satisfying

$$f(y^h(t,x), q^h(t,x)) \approx \begin{cases} \displaystyle\sum_{j=0}^N \mathrm{f}_{\mathrm{y},j}^h(\mathrm{y}^h(t), \mathrm{q}^h(t))\varphi_j(x) & \text{in (3.16a)},\\ & \text{for } x \in \Omega \text{ and } t \in [0,T] \text{ a.e.}\\ \displaystyle\sum_{j=1}^N \mathrm{f}_{\mathrm{q},j}^h(\mathrm{y}^h(t), \mathrm{q}^h(t))\varphi_j(x) & \text{in (3.16c)}, \end{cases}$$

Note that none of the matrices and vectors depend on $\boldsymbol{\mu}$ due to the affine dependence of the parameters as in (3.6) and (3.7).

For solving (3.16) we apply the implicit Euler method for the time integration (cf., e.g., [52]) on an equidistant time grid $t_k = k\delta$, $k = 0, \dots, K$ and $\delta = T/K$. Then, the problem is to find $\{\mathrm{y}^{h,k}\}_{k=0}^{K} \subset \mathbb{R}^{N+1}$ and $\{\mathrm{q}^{h,k}\}_{k=0}^{K} \subset \mathbb{R}^{N}$ solving

$$\begin{aligned} \mathrm{M_y}(\mathrm{y}^{h,k} - \mathrm{y}^{h,k-1}) + \mu_1 \delta \mathrm{A_y} \mathrm{y}^{h,k} - \mu_2 \delta \mathrm{M_y} \mathrm{f}_{\mathrm{y}}^{h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) = 0 \qquad & \text{for } k = 1, \dots, K, \\ \mathrm{M_y} \mathrm{y}^{h,0} = \mathrm{y}_{\circ}^{h}, & \\ \mu_3 \mathrm{A_q} \mathrm{q}^{h,k} + \mu_4 \, \mathrm{M_q} \mathrm{f}_{\mathrm{q}}^{h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) + \mathrm{b}^{h,k} = 0 \qquad & \text{for } k = 0, \dots, K, \end{aligned} \tag{3.17}$$

where $\mathrm{b}^{h,k} = \mathrm{b}^{h}(t_k)$. The solutions of this system, called *FO solutions* or *approximations*, are approximations of the FE solutions of (3.15) at each time step, namely

$$y^h(t_k) \approx \sum_{i=0}^{N} \mathrm{y}_i^k \varphi_i \quad \text{and} \quad q^h(t_k) \approx \sum_{i=1}^{N} \mathrm{q}_i^k \varphi_i \quad \text{for } k = 0, \dots, K.$$

Remark 3.5. Note that we can evaluate the vectors $\mathrm{f}_{\mathrm{y}}^{h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) \in \mathbb{R}^{N+1}$ and $\mathrm{f}_{\mathrm{q}}^{h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) \in \mathbb{R}^{N}$ at the same time, since they are obtained by applying f component-wise, i.e.

$$\mathrm{f}_{\mathrm{y}}^{h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) = \begin{bmatrix} f(\mathrm{y}_0^{h,k}, 0) \\ f(\mathrm{y}_1^{h,k}, \mathrm{q}_1^{h,k}) \\ \vdots \\ f(\mathrm{y}_N^{h,k}, \mathrm{q}_N^{h,k}) \end{bmatrix}, \qquad \mathrm{f}_{\mathrm{q}}^{h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) = \begin{bmatrix} f(\mathrm{y}_1^{h,k}, \mathrm{q}_1^{h,k}) \\ \vdots \\ f(\mathrm{y}_N^{h,k}, \mathrm{q}_N^{h,k}) \end{bmatrix}.$$

$\lozenge$

To solve the nonlinear system (3.17), we use Newton's method described in Algorithm 1 at each time step $k = 1, \dots, K$. In particular, for k fixed we want to find $\mathrm{x}_k := \begin{pmatrix} \mathrm{y}^{h,k} \\ \mathrm{q}^{h,k} \end{pmatrix}$ root of $\mathrm{F}^k = 0$, with $\mathrm{F}^k : \mathbb{R}^{2N+1} \to \mathbb{R}^{2N+1}$ defined as

$$\mathrm{F}^k(\mathrm{x}_k) := \begin{pmatrix} \mathrm{F}_{\mathrm{y}}^{k}(\mathrm{x}_k) \\ \mathrm{F}_{\mathrm{q}}^{k}(\mathrm{x}_k) \end{pmatrix} \in \mathbb{R}^{2N+1},$$

and

$$\begin{aligned} \mathrm{F}_{\mathrm{y}}^{k}(\mathrm{x}_k) &:= (\mathrm{M_y} + \mu_1 \delta \mathrm{A_y}) \mathrm{y}^{h,k} - \mathrm{M_y} \mathrm{y}^{h,k-1} - \mu_2 \delta \mathrm{M_y} \mathrm{f}_{\mathrm{y}}^{h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) \in \mathbb{R}^{N+1}, \\ \mathrm{F}_{\mathrm{q}}^{k}(\mathrm{x}_k) &:= \mu_3 \mathrm{A_q} \mathrm{q}^{h,k} + \mu_4 \mathrm{M_q} \mathrm{f}_{\mathrm{q}}^{h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) + \mathrm{b}^{h,k} \in \mathbb{R}^{N}. \end{aligned}$$

As initial point for Newton's method we choose the previous iterate $\begin{pmatrix} \mathrm{y}^{h,k-1} \\ \mathrm{q}^{h,k-1} \end{pmatrix}$, and for $k = 0$ we have, in particular,

$$\mathrm{F}_{\mathrm{y}}^{0}(\mathrm{x}_0) := \mathrm{M_y} \mathrm{y}^{h,0} - \mathrm{y}_{\circ}^{h} \in \mathbb{R}^{N+1}, \tag{3.18}$$

$$\mathrm{F}_{\mathrm{q}}^{0}(\mathrm{x}_0) := \mu_3 \mathrm{A_q} \mathrm{q}^{h,0} + \mu_4 \mathrm{M_q} \mathrm{f}_{\mathrm{q}}^{h}(\mathrm{y}^{h,0}, \mathrm{q}^{h,0}) + \mathrm{b}^{h,0} \in \mathbb{R}^{N}. \tag{3.19}$$

We will now derive the Jacobian $\mathrm{dF}^k(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) \in \mathbb{R}^{(2N+1)\times(2N+1)}$ of F^k, with the notation

$$\mathrm{dF}^k := \begin{pmatrix} \mathrm{d_y F_y^k} & \mathrm{d_q F_y^k} \\ \mathrm{d_y F_q^k} & \mathrm{d_q F_q^k} \end{pmatrix},$$

where

$$\mathrm{d_y F_y^k} := \frac{\mathrm{d}}{\mathrm{dy}^{h,k}} \mathrm{F_y^k}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) = \mathrm{M_y} + \mu_1 \delta \mathrm{A_y} - \mu_2 \delta \frac{\mathrm{d}}{\mathrm{dy}^{h,k}} \mathrm{M_y f_y^h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}), \tag{3.20a}$$

$$\mathrm{d_q F_y^k} := \frac{\mathrm{d}}{\mathrm{dq}^{h,k}} \mathrm{F_y^k}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) = -\mu_2 \delta \frac{\mathrm{d}}{\mathrm{dq}^{h,k}} \mathrm{M_y f_y^h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}), \tag{3.20b}$$

$$\mathrm{d_y F_q^k} := \frac{\mathrm{d}}{\mathrm{dy}^{h,k}} \mathrm{F_q^k}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) = \mu_4 \frac{\mathrm{d}}{\mathrm{dy}^{h,k}} \mathrm{M_q f_q^h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}), \tag{3.20c}$$

$$\mathrm{d_q F_q^k} := \frac{\mathrm{d}}{\mathrm{dq}^{h,k}} \mathrm{F_q^k}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) = \mu_3 \mathrm{A_q} + \mu_4 \frac{\mathrm{d}}{\mathrm{dq}^{h,k}} \mathrm{M_q f_q^h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}). \tag{3.20d}$$

First, using the point-wise partial derivative $\partial_\mathrm{y} f(\mathrm{y}, \mathrm{q}) = \frac{\sinh(\mathrm{q})}{2\sqrt{\mathrm{y}}}$ of f it holds that

$$\begin{aligned} \Big(\frac{\mathrm{d}}{\mathrm{dy}^{h,k}} \mathrm{M_y f_y^h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k})\Big)_{i,j} &= \frac{\mathrm{d}}{\mathrm{dy}_j^{h,k}} \int_\Omega f(y^{h,k}(x), q^{h,k}(x)) \varphi_i^y(x) \mathrm{d}x \\ &= \int_\Omega \partial_\mathrm{y} f(y^{h,k}(x), q^{h,k}(x)) \varphi_i^y(x) \frac{\mathrm{d}}{\mathrm{dy}_j^{h,k}} y^{h,k}(x) \mathrm{d}x \\ &= \int_\Omega \partial_\mathrm{y} f(y^{h,k}(x), q^{h,k}(x)) \varphi_i(x) \varphi_j(x) \mathrm{d}x \end{aligned} \tag{3.21a}$$

and, defining $\partial_\mathrm{q} f(\mathrm{y}, \mathrm{q}) = \sqrt{\mathrm{y}} \cosh(\mathrm{q})$, we similarly get

$$\Big(\frac{\mathrm{d}}{\mathrm{dq}^{h,k}} \mathrm{M_y f_y^h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k})\Big)_{i,j} = \int_\Omega \partial_\mathrm{q} f(y^{h,k}(x), q^{h,k}(x)) \varphi_i(x) \varphi_j(x) \mathrm{d}x, \tag{3.21b}$$

$$\Big(\frac{\mathrm{d}}{\mathrm{dy}^{h,k}} \mathrm{M_q f_q^h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k})\Big)_{i,j} = \int_\Omega \partial_\mathrm{y} f(y^{h,k}(x), q^{h,k}(x)) \varphi_i(x) \varphi_j(x) \mathrm{d}x, \tag{3.21c}$$

$$\Big(\frac{\mathrm{d}}{\mathrm{dq}^{h,k}} \mathrm{M_q f_q^h}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k})\Big)_{i,j} = \int_\Omega \partial_\mathrm{q} f(y^{h,k}(x), q^{h,k}(x)) \varphi_i(x) \varphi_j(x) \mathrm{d}x. \tag{3.21d}$$

Let $\partial_\mathrm{y}\mathrm{f}, \partial_\mathrm{q}\mathrm{f} : \mathbb{R}^{N+1} \times \mathbb{R}^{N+1} \to \mathbb{R}^{N+1}$ be the functions evaluating the partial derivatives component-wise, i.e.

$$\partial_\mathrm{y}\mathrm{f}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k})_i = \frac{\sinh(\mathrm{q}_i^{h,k})}{2\sqrt{\mathrm{y}_i^{h,k}}} \quad \text{and} \quad \partial_\mathrm{q}\mathrm{f}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k})_i = \sqrt{\mathrm{y}_i^{h,k}} \cosh(\mathrm{q}_i^{h,k}),$$

with $\mathrm{q}_0^{h,k} = 0$ representing the homogeneous Dirichlet condition for the component vector. Let us further define Θ^l for $l = 0, \ldots, N$ as $\Theta_{i,j}^l := \int_\Omega \varphi_l(x) \varphi_i(x) \varphi_j(x) \mathrm{d}x$. Hence, (3.21)

becomes

$$\frac{\mathrm{d}}{\mathrm{d}\mathrm{y}^{h,k}}\mathrm{M}_\mathrm{y}\mathrm{f}_\mathrm{y}^h(\mathrm{y}^{h,k},\mathrm{q}^{h,k}) = \sum_{l=0}^{N}\partial_\mathrm{y}\mathrm{f}(\mathrm{y}^{h,k},\mathrm{q}^{h,k})_l\,\Theta^l, \tag{3.22a}$$

$$\frac{\mathrm{d}}{\mathrm{d}\mathrm{q}^{h,k}}\mathrm{M}_\mathrm{y}\mathrm{f}_\mathrm{y}^h(\mathrm{y}^{h,k},\mathrm{q}^{h,k}) = \sum_{l=0}^{N}\partial_\mathrm{q}\mathrm{f}(\mathrm{y}^{h,k},\mathrm{q}^{h,k})_l\,\Theta^l, \tag{3.22b}$$

$$\frac{\mathrm{d}}{\mathrm{d}\mathrm{y}^{h,k}}\mathrm{M}_\mathrm{q}\mathrm{f}_\mathrm{q}^h(\mathrm{y}^{h,k},\mathrm{q}^{h,k}) = \sum_{l=0}^{N}\partial_\mathrm{y}\mathrm{f}(\mathrm{y}^{h,k},\mathrm{q}^{h,k})_l\,\Theta^l, \tag{3.22c}$$

$$\frac{\mathrm{d}}{\mathrm{d}\mathrm{q}^{h,k}}\mathrm{M}_\mathrm{q}\mathrm{f}_\mathrm{q}^h(\mathrm{y}^{h,k},\mathrm{q}^{h,k}) = \sum_{l=0}^{N}\partial_\mathrm{q}\mathrm{f}(\mathrm{y}^{h,k},\mathrm{q}^{h,k})_l\,\Theta^l. \tag{3.22d}$$

Then, using (3.22) into (3.20) we get

$$\mathrm{d}_\mathrm{y}\mathrm{F}_\mathrm{y}^k = \frac{\mathrm{d}}{\mathrm{d}\mathrm{y}^{h,k}}\mathrm{F}_\mathrm{y}^k(\mathrm{y}^{h,k},\mathrm{q}^{h,k}) = \mathrm{M}_\mathrm{y} + \mu_1\delta\mathrm{A}_\mathrm{y} - \mu_2\delta\sum_{l=0}^{N}\partial_\mathrm{y}\mathrm{f}(\mathrm{y}^{h,k},\mathrm{q}^{h,k})_l\,\Theta^l, \tag{3.23a}$$

$$\mathrm{d}_\mathrm{q}\mathrm{F}_\mathrm{y}^k = \frac{\mathrm{d}}{\mathrm{d}\mathrm{q}^{h,k}}\mathrm{F}_\mathrm{y}^k(\mathrm{y}^{h,k},\mathrm{q}^{h,k}) = -\mu_2\delta\sum_{l=0}^{N}\partial_\mathrm{q}\mathrm{f}(\mathrm{y}^{h,k},\mathrm{q}^{h,k})_l\,\Theta^l, \tag{3.23b}$$

$$\mathrm{d}_\mathrm{y}\mathrm{F}_\mathrm{q}^k = \frac{\mathrm{d}}{\mathrm{d}\mathrm{y}^{h,k}}\mathrm{F}_\mathrm{q}^k(\mathrm{y}^{h,k},\mathrm{q}^{h,k}) = \mu_4\sum_{l=0}^{N}\partial_\mathrm{y}\mathrm{f}(\mathrm{y}^{h,k},\mathrm{q}^{h,k})_l\,\Theta^l, \tag{3.23c}$$

$$\mathrm{d}_\mathrm{q}\mathrm{F}_\mathrm{q}^k = \frac{\mathrm{d}}{\mathrm{d}\mathrm{q}^{h,k}}\mathrm{F}_\mathrm{q}^k(\mathrm{y}^{h,k},\mathrm{q}^{h,k}) = \mu_3\mathrm{A}_\mathrm{q} + \mu_4\sum_{l=0}^{N}\partial_\mathrm{q}\mathrm{f}(\mathrm{y}^{h,k},\mathrm{q}^{h,k})_l\,\Theta^l. \tag{3.23d}$$

$\Diamond$

Remark 3.6. We comment on the fact that we have not defined the dimension of Θ^l and that, in fact, this matrix changes dimension depending on the equation where it is used. For example, $\mathrm{d}_\mathrm{y}\mathrm{F}_\mathrm{y}^k \in \mathbb{R}^{(N+1)\times(N+1)}$, and hence we need Θ^l to be of dimension $(N+1)\times(N+1)$. On the other hand, $\mathrm{d}_\mathrm{q}\mathrm{F}_\mathrm{y}^k \in \mathbb{R}^{(N+1)\times N}$, and in this case Θ^l will be of dimension $(N+1)\times N$. In practice, we can just save one $(N+1)\times(N+1)$ matrix Θ^l, and depending on the situation we extract the right sub-matrix. For example, in (3.23b) we only use the following highlighted sub-matrix:

$$\Theta^l = \begin{pmatrix} \Theta^l_{0,0} & \boxed{\begin{matrix}\Theta^l_{0,1} & \cdots & \Theta^l_{0,N}\\ \Theta^l_{1,1} & \cdots & \Theta^l_{1,N}\\ \vdots & \ddots & \vdots \\ \Theta^l_{N,1} & \cdots & \Theta^l_{N,N}\end{matrix}} \\ \Theta^l_{1,0} & \\ \vdots & \\ \Theta^l_{N,0} & \end{pmatrix}$$

$\Diamond$

Finally, the algorithm to evaluate the FO state variables $\mathrm{y}^{h,k}, \mathrm{q}^{h,k}$ for $k = 0, \ldots, K$ given parameter $\boldsymbol{\mu}$ is as follows:

- Evaluate $\mathrm{y}^{h,0}$ by projecting the initial value $y_\circ$, namely by solving the linear problem (3.18).
- Get a consistent initial condition for the state q^h, namely compute $\mathrm{q}^{h,0}$ solving the decoupled equation (3.19).
- For $k = 1, \dots, K$, find the root of the nonlinear equation $\mathrm{F}^k(\mathrm{y}^{h,k}, \mathrm{q}^{h,k}) = 0$ by using Newton's method (Algorithm 1).

Example 3.7. To give an idea of what the state variables look like, let us introduce an example in $\Omega = (0, 1)$ and with $T = 1$. We use linear Lagrangian elements on 200 nodes and 201 time steps. The diffusion functions are chosen as $\kappa_1(x) \equiv \kappa_2(x) \equiv 1$, and the initial value as $y_\circ(x) \equiv 6$. We show in Figure 3.1 approximated FO state variables for the input function $u(t) = 0.5 \cdot \cos(10t) + 0.4 \cdot \sin(20t)$. ◊

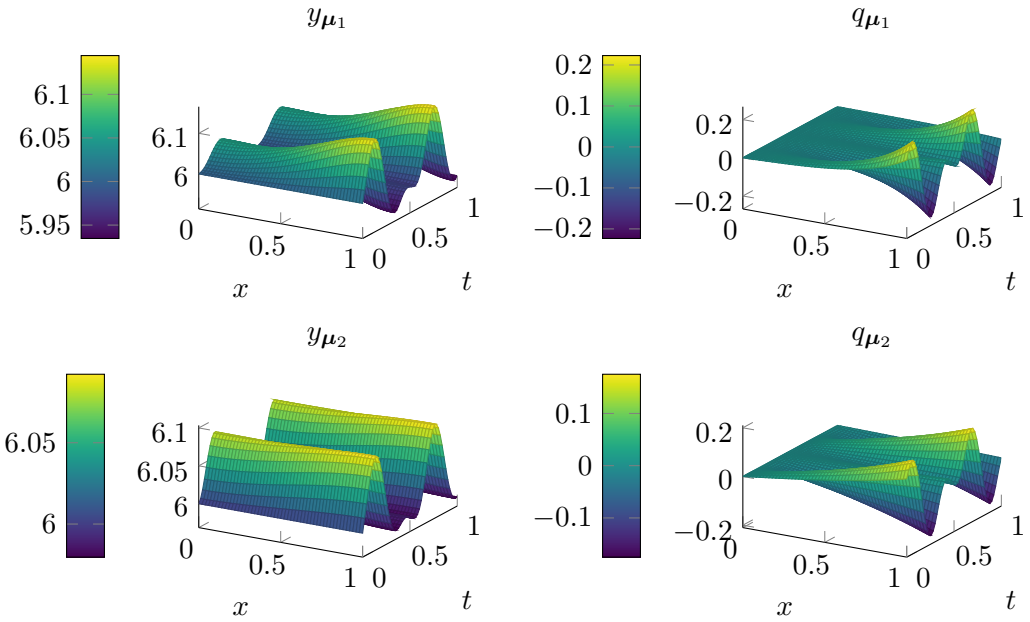


Figure 3.1: State variables for $\boldsymbol{\mu}_1 = (1, 5, 1, 5)$ and $\boldsymbol{\mu}_2 = (5, 3, 4, 2)$.

The FO discretization of the reduced cost function (3.12) is straight-forward, once we define the discretized data $w^{h,k}$ as explained in (2.9) and (2.10), with associated coordinate vector $\mathrm{w}^{h,k} \in \mathbb{R}^N$. Applying the time discretization, we get to the following approximation:

$$\hat{J}^h(\boldsymbol{\mu}) := 1 + \frac{\alpha_J}{2} \sum_{k=0}^{K} \alpha_k \left\| q_{\boldsymbol{\mu}}^{h,k} - w^{h,k} \right\|_H^2 + \frac{\beta_J}{2} \left\| \boldsymbol{\mu} - \hat{\boldsymbol{\mu}} \right\|_2^2. \tag{3.24}$$

Using FO matrices, we can also write

$$\hat{J}^h(\boldsymbol{\mu}) = 1 + \frac{\alpha_J}{2} \sum_{k=0}^{K} \alpha_k \left\| \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{w}^{h,k} \right\|_{\mathrm{M_q}}^2 + \frac{\beta_J}{2} \left\| \boldsymbol{\mu} - \hat{\boldsymbol{\mu}} \right\|_2^2. \tag{3.25}$$

The FO discretization of the adjoint variables follows the same steps of the one of the state variables. The only difference is the time stepping, which is backwards. Given $\{\mathrm{y}^{h,k}, \mathrm{q}^{h,k}\}_{k=0}^{K}$, then $\{\mathrm{p_y}^{h,k}\}_{k=0}^{K} \subset \mathbb{R}^{N+1}$ and $\{\mathrm{p_q}^{h,k}\}_{k=0}^{K} \subset \mathbb{R}^{N}$ are the solutions of

$$\begin{aligned} -\mathrm{M_y}(\mathrm{p_y}^{h,k+1} - \mathrm{p_y}^{h,k}) + \mu_1 \delta \mathrm{A_y} \mathrm{p_y}^{h,k} - \mu_2 \delta \mathrm{G_{yy}}^{k} \mathrm{p_y}^{h,k} + \mu_4 \delta \mathrm{G_{yq}}^{k} \mathrm{p_q}^{h,k} = 0 \quad & \text{for } k = K-1, \dots, 0, \\ \mathrm{p_y}^{h,K} = 0, & \\ \mu_3 \mathrm{A_q} \mathrm{p_q}^{h,k} - \mu_2 \mathrm{G_{qy}}^{k} \mathrm{p_y}^{h,k} + \mu_4 \mathrm{G_{qq}}^{k} \mathrm{p_q}^{h,k} - \mathrm{b}^{\mathrm{p},h,k} = 0 \quad & \text{for } k = K, \dots, 0, \end{aligned} \tag{3.26}$$

where, for any $k = 0, \dots, K$,

$$\begin{aligned} \mathrm{G_{yy}}^{k} &= \sum_{l=0}^{N} \partial_{\mathrm{y}} \mathrm{f}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k})_l \, \Theta^l \in \mathbb{R}^{(N+1)\times(N+1)}, \quad \mathrm{G_{yq}}^{k} = \sum_{l=0}^{N} \partial_{\mathrm{y}} \mathrm{f}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k})_l \, \Theta^l \in \mathbb{R}^{(N+1)\times N}, \\ \mathrm{G_{qy}}^{k} &= \sum_{l=0}^{N} \partial_{\mathrm{q}} \mathrm{f}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k})_l \, \Theta^l \in \mathbb{R}^{N\times(N+1)}, \qquad \mathrm{G_{qq}}^{k} = \sum_{l=0}^{N} \partial_{\mathrm{q}} \mathrm{f}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k})_l \, \Theta^l \in \mathbb{R}^{N\times N}, \end{aligned} \tag{3.27}$$

and $\mathrm{b}^{\mathrm{p},h,k} = \alpha_J \mathrm{M_q}(\mathrm{w}^{h,k} - \mathrm{q}^{h,k})$.

Remark 3.8. System (2.12) is linearly coupled. To solve this system we first project the final condition for the variable p_y, namely $\mathrm{p_y}^{h,k} = 0$, while the final condition for p_q is given by solving a linear problem, namely

$$(\mu_3 \mathrm{A_q} + \mu_4 \mathrm{G_{qq}}^{K}) \mathrm{p_q}^{h,K} = \mathrm{b}^{p,h,K}.$$

Then, for each time step $k = K-1, \dots, 0$ we can transform the coupled equations into a single vectorial equation. Indeed, defined $\tilde{\mathrm{M}} \in \mathbb{R}^{(2N+1)\times(2N+1)}$ and $\tilde{\mathrm{b}} \in \mathbb{R}^{2N+1}$ as

$$\tilde{\mathrm{M}} = \begin{bmatrix} \mathrm{M_y} + \mu_1 \delta \mathrm{A_y} - \mu_2 \delta \mathrm{G_{yy}}^{k} & \mu_4 \delta \mathrm{G_{yq}}^{k} \\ -\mu_2 \mathrm{G_{qy}}^{k} & \mu_3 \mathrm{A_q} + \mu_4 \mathrm{G_{qq}}^{k} \end{bmatrix} \quad \text{and} \quad \tilde{\mathrm{b}} = \begin{bmatrix} \mathrm{M_y} \mathrm{p_y}^{h,k+1} \\ \alpha_J \mathrm{M_q}(\mathrm{w}^{h,k} - \mathrm{q}^{h,k}) \end{bmatrix},$$

then $\mathrm{p_y}^{h,k}$ and $\mathrm{p_q}^{h,k}$ solve

$$\tilde{\mathrm{M}} \begin{pmatrix} \mathrm{p_y}^{h,k} \\ \mathrm{p_q}^{h,k} \end{pmatrix} = \tilde{\mathrm{b}}.$$

$\Diamond$

Finally, the gradient of the reduced cost function is approximated as

$$\nabla \hat{J}^h(\boldsymbol{\mu}) = \begin{pmatrix} \displaystyle\sum_{k=0}^{K} \alpha_k \, \langle \mathrm{y}_{\boldsymbol{\mu}}^{h,k}, \mathrm{p}_{\mathrm{y}\boldsymbol{\mu}}^{h,k} \rangle_{\mathrm{A_y}} \\ \displaystyle -\sum_{k=0}^{K} \alpha_k \, \langle \mathrm{f_y}^{h}(\mathrm{y}_{\boldsymbol{\mu}}^{h,k}, \mathrm{q}_{\boldsymbol{\mu}}^{h,k}), \mathrm{p}_{\mathrm{y}\boldsymbol{\mu}}^{h,k} \rangle_{\mathrm{M_y}} \\ \displaystyle\sum_{k=0}^{K} \alpha_k \, \langle \mathrm{q}_{\boldsymbol{\mu}}^{h,k}, \mathrm{p}_{\mathrm{q}\boldsymbol{\mu}}^{h,k} \rangle_{\mathrm{A_q}} \\ \displaystyle\sum_{k=0}^{K} \alpha_k \, \langle \mathrm{f_q}^{h}(\mathrm{y}_{\boldsymbol{\mu}}^{h,k}, \mathrm{q}_{\boldsymbol{\mu}}^{h,k}), \mathrm{p}_{\mathrm{q}\boldsymbol{\mu}}^{h,k} \rangle_{\mathrm{M_q}} \end{pmatrix} + \beta_J(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}). \tag{3.28}$$

3.3 Reduced order discretization

We construct reduced order spaces $V^\ell \subset V^h$ and $V_\circ^\ell \subset V_\circ^h$, of dimensions respectively $L_\mathrm{y} \ll N$ and $L_\mathrm{q} \ll N$, and a set of basis for each space, $\{\psi_1^y, \dots, \psi_{L_\mathrm{y}}^y\} \subset V^\ell$ and $\{\psi_1^q, \dots, \psi_{L_\mathrm{q}}^q\} \subset V_\circ^\ell$. Furthermore, let us define the coefficient matrices $\Psi_\mathrm{y} \in \mathbb{R}^{(N+1)\times L_\mathrm{y}}$ and $\Psi_\mathrm{q} \in \mathbb{R}^{N\times L_\mathrm{q}}$ of the basis $\{\psi_i^y\}_{i=1}^{L_\mathrm{y}}$ with respect to the basis $\{\varphi_i\}_{i=0}^N$ of V^h and of the basis $\{\psi_i^q\}_{i=1}^{L_\mathrm{q}}$ with respect to the basis $\{\varphi_i\}_{i=1}^N$ of $V_\circ^h$, namely,

$$\psi_j^y = \sum_{i=0}^{N} (\Psi_\mathrm{y})_{ij}\, \varphi_i \text{ for } j = 1, \dots, L_\mathrm{y} \qquad \text{and} \qquad \psi_j^q = \sum_{i=1}^{N} (\Psi_\mathrm{q})_{ij}\, \varphi_i \text{ for } j = 1, \dots, L_\mathrm{q}.$$

As a first step we project all matrices on the reduced spaces so that a reduced order approximation of (3.17) reads as follows: find $\{\hat{\mathrm{y}}^k\}_{k=0}^K \subset \mathbb{R}^{L_\mathrm{y}}$ and $\{\hat{\mathrm{q}}^k\}_{k=0}^K \subset \mathbb{R}^{L_\mathrm{q}}$ such that

$$\begin{aligned} (\mathrm{M}_\mathrm{y}^\ell + \mu_1 \delta \mathrm{A}_\mathrm{y}^\ell)\hat{\mathrm{y}}^k - \mathrm{M}_\mathrm{y}^\ell \hat{\mathrm{y}}^{k-1} - \mu_2 \delta \Psi_\mathrm{y}^\top \mathrm{M}_\mathrm{y} \mathrm{f}_\mathrm{y}^h(\Psi_\mathrm{y}\hat{\mathrm{y}}^k, \Psi_\mathrm{q}\hat{\mathrm{q}}^k) = 0 \qquad & \text{for } k = 1, \dots, K, \\ \mathrm{M}_\mathrm{y}^\ell \hat{\mathrm{y}}^0 = \mathrm{y}_\circ^\ell, & \\ \mu_3 \mathrm{A}_\mathrm{q}^\ell \hat{\mathrm{q}}^k + \mu_4 \Psi_\mathrm{q}^\top \mathrm{M}_\mathrm{q} \mathrm{f}_\mathrm{q}^h(\Psi_\mathrm{y}\hat{\mathrm{y}}^k, \Psi_\mathrm{q}\hat{\mathrm{q}}^k) + \mathrm{b}^{\ell,k} = 0 \qquad & \text{for } k = 0, \dots, K \end{aligned} \tag{3.29}$$

with $\mathrm{M}_\mathrm{y}^\ell = \Psi_\mathrm{y}^\top \mathrm{M}_\mathrm{y} \Psi_\mathrm{y}$, $\mathrm{A}_\mathrm{y}^\ell = \Psi_\mathrm{y}^\top \mathrm{A}_\mathrm{y} \Psi_\mathrm{y}$, $\mathrm{A}_\mathrm{q}^\ell = \Psi_\mathrm{q}^\top \mathrm{A}_\mathrm{q} \Psi_\mathrm{q}$, $\mathrm{b}^{\ell,k} = \Psi_\mathrm{q}^\top \mathrm{b}^{h,k}$ for $k = 0, \dots, K$, and $\mathrm{y}_\circ^\ell = \Psi_\mathrm{y}^\top \mathrm{y}_\circ^h$.

System (3.29) still depends on dimension N through the evaluation of the nonlinearities f_y^h and f_q^h. We use the discrete empirical interpolation method (DEIM) described in Section 1.2.5 to make the evaluation of the system independent of N.

To avoid the confusion with the notation, we reiterate over the main steps. In a nutshell, the vectors $\mathrm{f}_\mathrm{y}^{h,k} := \mathrm{f}_\mathrm{y}(\Psi_\mathrm{y}\hat{\mathrm{y}}^k, \Psi_\mathrm{q}\hat{\mathrm{q}}^k)$ and $\mathrm{f}_\mathrm{q}^{h,k} := \mathrm{f}_\mathrm{q}(\Psi_\mathrm{y}\hat{\mathrm{y}}^k, \Psi_\mathrm{q}\hat{\mathrm{q}}^k)$ for $k = 0, \dots, K$ are projected onto a smaller space of dimension $L_\mathrm{f} \ll N$. For the q-system this is done by finding matrices $\Psi_\mathrm{f} \in \mathbb{R}^{N\times L_\mathrm{f}}$, and $\mathrm{I} \in \mathbb{R}^{N\times L_\mathrm{f}}$ such that

$$\mathrm{f}^k \approx \Psi_\mathrm{f}(\mathrm{I}^\top \Psi_\mathrm{f})^{-1} \mathrm{I}^\top \mathrm{f}^k,$$

where the matrix Ψ_f is given by the POD method using the snapshots f^k for $k = 0, \dots, K$, and the matrix I is composed only by zeros and ones such that $\mathrm{I}^\top \mathrm{v} \in \mathbb{R}^{\ell_\mathrm{f}}$ contains only selected rows of $\mathrm{v} \in \mathbb{R}^N$.

Remark 3.9. For the y-system (of dimension $N+1$) we set

$$\tilde{\Psi}_\mathrm{f} := \begin{bmatrix} 0 \\ \Psi_\mathrm{f} \end{bmatrix} \in \mathbb{R}^{(N+1)\times \ell_\mathrm{f}}, \quad \tilde{\mathrm{I}} := \begin{bmatrix} 0 \\ \mathrm{I} \end{bmatrix} \in \mathbb{R}^{(N+1)\times \ell_\mathrm{f}},$$

and get $\tilde{\mathrm{f}}^k \approx \tilde{\Psi}_\mathrm{f}(\tilde{\mathrm{I}}^\top \tilde{\Psi}_\mathrm{f})^{-1} \tilde{\mathrm{I}}^\top \tilde{\mathrm{f}}^k \in \mathbb{R}^{N+1}$. ◊

Since the evaluation of f^k is done component-wise, we can write

$$\mathrm{I}^\top \mathrm{f}^k = \mathrm{I}^\top \mathrm{f}(\Psi_\mathrm{y}\hat{\mathrm{y}}^k, \Psi_\mathrm{q}\hat{\mathrm{q}}^k) = \mathrm{f}(\tilde{\mathrm{I}}^\top \Psi_\mathrm{y}\hat{\mathrm{y}}^k, \mathrm{I}^\top \Psi_\mathrm{q}\hat{\mathrm{q}}^k) =: \mathrm{f}(\hat{\Psi}_\mathrm{y}\hat{\mathrm{y}}^k, \hat{\Psi}_\mathrm{q}\hat{\mathrm{q}}^k)$$

for $\hat{\Psi}_{\mathrm{y}} = \tilde{\mathrm{I}}^{\top}\Psi_{\mathrm{y}} \in \mathbb{R}^{\ell_{\mathrm{f}} \times L_{\mathrm{y}}}$ and $\hat{\Psi}_{\mathrm{q}} = \mathrm{I}^{\top}\Psi_{\mathrm{q}} \in \mathbb{R}^{\ell_{\mathrm{f}} \times L_{\mathrm{q}}}$. Then, for

$$\mathrm{G}_{\mathrm{y}} = \Psi_{\mathrm{y}}^{\top}\mathrm{M}_{\mathrm{y}}\tilde{\Psi}_{\mathrm{f}}(\tilde{\mathrm{I}}^{\top}\tilde{\Psi}_{\mathrm{f}})^{-1} \in \mathbb{R}^{L_{\mathrm{y}} \times \ell_{\mathrm{f}}} \quad \text{and} \quad \mathrm{G}_{\mathrm{q}} = \Psi_{\mathrm{q}}^{\top}\mathrm{M}_{\mathrm{q}}\Psi_{\mathrm{f}}(\mathrm{I}^{\top}\Psi_{\mathrm{f}})^{-1} \in \mathbb{R}^{L_{\mathrm{q}} \times \ell_{\mathrm{f}}}$$

system (3.29) is approximated by

$$\begin{aligned}(\mathrm{M}_{\mathrm{y}}^{\ell} + \mu_1\delta\mathrm{A}_{\mathrm{y}}^{\ell})\hat{\mathrm{y}}^k - \mathrm{M}_{\mathrm{y}}^{\ell}\hat{\mathrm{y}}^{k-1} - \mu_2\delta\mathrm{G}_{\mathrm{y}}\mathrm{f}(\hat{\Psi}_{\mathrm{y}}\hat{\mathrm{y}}^k, \hat{\Psi}_{\mathrm{q}}\hat{\mathrm{q}}^k) = 0 \qquad &\text{for } k = 1, \dots, K,\\ \mathrm{M}_{\mathrm{y}}^{\ell}\hat{\mathrm{y}}^1 = \mathrm{y}_{\circ}^{\ell}, \qquad &\\ \mu_3\mathrm{A}_{\mathrm{q}}^{\ell}\hat{\mathrm{q}}^k + \mu_4\mathrm{G}_{\mathrm{q}}\mathrm{f}(\hat{\Psi}_{\mathrm{y}}\hat{\mathrm{y}}^k, \hat{\Psi}_{\mathrm{q}}\hat{\mathrm{q}}^k) + \mathrm{b}^{\ell,k} = 0 \qquad &\text{for } k = 0, \dots, K,\end{aligned} \tag{3.30}$$

which is finally independent of N. In the following, we call system (3.30) reduced order (RO) model. The RO solutions $\{\hat{\mathrm{y}}^k\}_{k=0}^K \subset \mathbb{R}^{L_{\mathrm{y}}}$ and $\{\hat{\mathrm{q}}^k\}_{k=0}^K \subset \mathbb{R}^{L_{\mathrm{q}}}$ are interpreted as a reduced order approximations for $\{\mathrm{y}^k\}_{k=0}^K \subset \mathbb{R}^{N+1}$ and $\{\mathrm{q}^k\}_{k=0}^K \subset \mathbb{R}^N$, namely

$$\mathrm{y}^{h,k} \approx \mathrm{y}^{\ell,k} := \Psi_{\mathrm{y}}\hat{\mathrm{y}}^k \qquad \text{and} \qquad \mathrm{q}^{h,k} \approx \mathrm{q}^{\ell,k} := \Psi_{\mathrm{q}}\hat{\mathrm{q}}^k.$$

Let us from now on add the dependence on the parameter $\boldsymbol{\mu}$ on the subscript. Namely, $\{\mathrm{y}_{\boldsymbol{\mu}}^k\}_{k=0}^K$ and $\{\mathrm{q}_{\boldsymbol{\mu}}^k\}_{k=0}^K$ for the FO solutions (solving (3.17)), and $\{\hat{\mathrm{y}}_{\boldsymbol{\mu}}^k\}_{k=0}^K$ and $\{\hat{\mathrm{q}}_{\boldsymbol{\mu}}^k\}_{k=0}^K$ for the ROM solutions (solving (3.30)) with parameter $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$. The RO approximation of the cost (3.25) will then be

$$\hat{J}^{\ell}(\boldsymbol{\mu}) = 1 + \frac{\alpha_J}{2}\sum_{k=0}^{K}\alpha_k \left\|\mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\right\|_{\mathrm{M_q}}^2 + \frac{\beta_J}{2}\left\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\right\|_2^2. \tag{3.31}$$

Remark 3.10. The evaluation of $\hat{J}^{\ell}$ is online efficient, since by pre-evaluating

$$\mathrm{r}_1^k = \Psi_{\mathrm{q}}^{\top}\mathrm{M}_{\mathrm{q}}\mathrm{w}^{h,k} \in \mathbb{R}^{L_{\mathrm{q}}} \qquad \text{and} \qquad \mathrm{r}_2^k = \mathrm{w}^{h,k\top}\mathrm{M}_{\mathrm{q}}\mathrm{w}^{h,k} \in \mathbb{R} \qquad \text{for } k = 0, \dots, K,$$

then we can write

$$\begin{aligned}\left\|\mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\right\|_{\mathrm{M_q}}^2 &= (\Psi_{\mathrm{q}}\hat{\mathrm{q}}_{\boldsymbol{\mu}}^k - \mathrm{w}^{h,k})^{\top}\mathrm{M}_{\mathrm{q}}(\Psi_{\mathrm{q}}\hat{\mathrm{q}}_{\boldsymbol{\mu}}^k - \mathrm{w}^{h,k})\\ &= \hat{\mathrm{q}}_{\boldsymbol{\mu}}^{k\top}\mathrm{M}_{\mathrm{q}}^{\ell}\hat{\mathrm{q}}_{\boldsymbol{\mu}}^k - 2\hat{\mathrm{q}}_{\boldsymbol{\mu}}^{k\top}\mathrm{r}_1^k + \mathrm{r}_2^k.\end{aligned}$$

$\lozenge$

Once we have approximated the state variables, it is also time to approximate the adjoint variables $\{\hat{\mathrm{p}}_{\mathrm{y}}^k, \hat{\mathrm{p}}_{\mathrm{q}}^k\}_{k=0}^K$ (later denoted also by the parameter, e.g. $\hat{\mathrm{p}}_{\mathrm{y}}^k = \hat{\mathrm{p}}_{\mathrm{y},\boldsymbol{\mu}}^k$). Once we build RO spaces $V^{\ell_{\mathrm{p}}} = \mathrm{span}\{\psi_1^{py}, \dots, \psi_{L_{\mathrm{py}}}^{py}\}$ and $V_{\circ}^{\ell_{\mathrm{p}}} = \mathrm{span}\{\psi_1^{pq}, \dots, \psi_{L_{\mathrm{pq}}}^{pq}\}$ with RO coefficient matrices respectively $\Psi_{\mathrm{py}} \in \mathbb{R}^{N \times L_{\mathrm{py}}}$ and $\Psi_{\mathrm{pq}} \in \mathbb{R}^{N \times L_{\mathrm{pq}}}$, system (3.26) gets projected into

$$\begin{aligned}-\mathrm{M}_{\mathrm{y}}^{\ell_{\mathrm{p}}}(\hat{\mathrm{p}}_{\mathrm{y}}^{k+1} - \hat{\mathrm{p}}_{\mathrm{y}}^k) + \mu_1\delta\mathrm{A}_{\mathrm{y}}^{\ell_{\mathrm{p}}}\hat{\mathrm{p}}_{\mathrm{y}}^k - \mu_2\delta\mathrm{G}_{\mathrm{yy}}^{\ell_{\mathrm{p}},k}\hat{\mathrm{p}}_{\mathrm{y}}^k + \mu_4\delta\mathrm{G}_{\mathrm{yq}}^{\ell_{\mathrm{p}},k}\hat{\mathrm{p}}_{\mathrm{q}}^k = 0 \quad &\text{for } k = K-1, \dots, 0,\\ \hat{\mathrm{p}}_{\mathrm{y}}^K = 0, \qquad &\\ \mu_3\mathrm{A}_{\mathrm{q}}^{\ell_{\mathrm{p}}}\hat{\mathrm{p}}_{\mathrm{q}}^k - \mu_2\mathrm{G}_{\mathrm{qy}}^{\ell_{\mathrm{p}},k}\hat{\mathrm{p}}_{\mathrm{y}}^k + \mu_4\mathrm{G}_{\mathrm{qq}}^{\ell_{\mathrm{p}},k}\hat{\mathrm{p}}_{\mathrm{q}}^k - \mathrm{b}^{\mathrm{p},\ell_{\mathrm{p}},k} = 0 \quad &\text{for } k = K, \dots, 0,\end{aligned} \tag{3.32}$$

where

$$\begin{aligned}
&\mathrm{M}_{\mathrm{y}}^{\ell_{\mathrm{p}}} := \Psi_{\mathrm{py}}^{\top}\mathrm{M}\Psi_{\mathrm{py}}, && \mathrm{A}_{\mathrm{y}}^{\ell_{\mathrm{p}}} := \Psi_{\mathrm{py}}^{\top}\mathrm{A}\Psi_{\mathrm{py}},\\
&\mathrm{G}_{\mathrm{yy}}^{\ell_{\mathrm{p}},k} := \Psi_{\mathrm{py}}^{\top}\mathrm{G}_{\mathrm{yy}}^{k}\Psi_{\mathrm{py}}, && \mathrm{G}_{\mathrm{yq}}^{\ell_{\mathrm{p}},k} := \Psi_{\mathrm{py}}^{\top}\mathrm{G}_{\mathrm{yq}}^{k}\Psi_{\mathrm{pq}},\\
&\mathrm{G}_{\mathrm{qy}}^{\ell_{\mathrm{p}},k} := \Psi_{\mathrm{pq}}^{\top}\mathrm{G}_{\mathrm{qy}}^{k}\Psi_{\mathrm{py}}, && \mathrm{G}_{\mathrm{qq}}^{\ell_{\mathrm{p}},k} := \Psi_{\mathrm{pq}}^{\top}\mathrm{G}_{\mathrm{qq}}^{k}\Psi_{\mathrm{pq}},\\
&\mathrm{A}_{\mathrm{q}}^{\ell_{\mathrm{p}}} := \Psi_{\mathrm{pq}}^{\top}\mathrm{A}\Psi_{\mathrm{pq}}, && \mathrm{b}^{\mathrm{p},\ell_{\mathrm{p}},k} := \Psi_{\mathrm{pq}}^{\top}\mathrm{b}^{\mathrm{p},h,k} \text{ for } k = 0,\dots,K.
\end{aligned}$$

Once we find the solutions $\hat{\mathrm{p}}_{\mathrm{y}}^{k}, \hat{\mathrm{p}}_{\mathrm{q}}^{k}$ for $k = 0,\dots,K$, we furthermore define

$$\mathrm{p}_{\mathrm{y}}^{\ell,k} := \Psi_{\mathrm{py}}\hat{\mathrm{p}}_{\mathrm{y}}^{k} \in \mathbb{R}^{N} \qquad \text{and} \qquad \mathrm{p}_{\mathrm{q}}^{\ell,k} := \Psi_{\mathrm{pq}}\hat{\mathrm{p}}_{\mathrm{q}}^{k} \in \mathbb{R}^{N}$$

and

$$p_{y}^{\ell,k} := \sum_{i=1}^{L_{\mathrm{py}}} \hat{\mathrm{p}}_{\mathrm{y}i}^{k}\psi_{i}^{py} = \sum_{i=1}^{N} \mathrm{p}_{\mathrm{y}i}^{\ell,k}\varphi_{i} \in V^{h} \qquad \text{and} \qquad p_{q}^{\ell,k} := \sum_{i=1}^{L_{\mathrm{pq}}} \hat{\mathrm{p}}_{\mathrm{q}i}^{k}\psi_{i}^{pq} = \sum_{i=1}^{N} \mathrm{p}_{\mathrm{q}i}^{\ell,k}\varphi_{i} \in V_{\circ}^{h}$$

Remark 3.11. We observe that we do not actually need to evaluate $4(K+1)$ different matrices $\mathrm{G}_{\mathrm{yy}}^{\ell_{\mathrm{p}},k}, \mathrm{G}_{\mathrm{yq}}^{\ell_{\mathrm{p}},k}, \mathrm{G}_{\mathrm{qy}}^{\ell_{\mathrm{p}},k}, \mathrm{G}_{\mathrm{qq}}^{\ell_{\mathrm{p}},k}$ every time we change $\boldsymbol{\mu}$. Indeed, pre-evaluating just once the products

$$\begin{aligned}
\Theta_{\mathrm{pyy}}^{l} &:= \Psi_{\mathrm{py}}\Theta^{l}\Psi_{\mathrm{py}},\\
\Theta_{\mathrm{pyq}}^{l} &:= \Psi_{\mathrm{py}}\Theta^{l}\Psi_{\mathrm{pq}},\\
\Theta_{\mathrm{pyy}}^{l} &:= \Psi_{\mathrm{pq}}\Theta^{l}\Psi_{\mathrm{py}},\\
\Theta_{\mathrm{pyy}}^{l} &:= \Psi_{\mathrm{pq}}\Theta^{l}\Psi_{\mathrm{pq}}
\end{aligned}$$

for $l = 0,\dots,N$, we can then evaluate, e.g.,

$$\mathrm{G}_{\mathrm{yy}}^{\ell,k} = \sum_{l=0}^{N} \partial_{\mathrm{y}}\mathrm{f}(\mathrm{y}^{h,k}, \mathrm{q}^{h,k})_{l}\Theta_{\mathrm{pyy}}^{l} \in \mathbb{R}^{L_{\mathrm{y}}\times L_{\mathrm{y}}}$$

much faster. Note that the matrix Θ^{l} is sparse and evaluating the products like $\Theta_{\mathrm{pyy}}^{l}$ is not so expensive.

Remark 3.12. As in Remark 3.8, we can compute system (3.32) pretty fast since it is linear. Once we evaluate the first final condition $\hat{\mathrm{p}}_{\mathrm{y}}^{K} = 0$, the second one is given as

$$\left[\mu_{3}\mathrm{A}_{\mathrm{q}}^{\ell_{\mathrm{p}}} + \mu_{4}\mathrm{G}_{\mathrm{qq}}^{\ell_{\mathrm{p}},k}\right]\hat{\mathrm{p}}_{\mathrm{q}}^{k} = \alpha_{J}\Psi_{\mathrm{pq}}^{\top}\mathrm{M}_{\mathrm{q}}(\mathrm{w}^{h,k} - \mathrm{q}^{h,k}).$$

Then, for $k = K-1,\dots,K$, we define

$$\tilde{\mathrm{M}}^{\ell} := \begin{bmatrix} \mathrm{M}_{\mathrm{y}}^{\ell} + \mu_{1}\delta\mathrm{A}_{\mathrm{y}} - \mu_{2}\delta\mathrm{G}_{\mathrm{yy}}^{\ell_{\mathrm{p}},k} & \mu_{4}\delta\mathrm{G}_{\mathrm{yq}}^{\ell,k} \\ -\mu_{2}\mathrm{G}_{\mathrm{qy}}^{\ell_{\mathrm{p}},k} & \mu_{3}\mathrm{A}_{\mathrm{q}} + \mu_{4}\mathrm{G}_{\mathrm{qq}}^{\ell_{\mathrm{p}},k} \end{bmatrix}$$

and

$$\tilde{\mathrm{b}}^{\ell} := \begin{bmatrix} \mathrm{M}_{\mathrm{y}}^{\ell_{\mathrm{p}}}\hat{\mathrm{p}}_{\mathrm{y}}^{k+1} \\ \alpha_{J}\Psi_{\mathrm{pq}}^{\top}\mathrm{M}_{\mathrm{q}}(\mathrm{w}^{h,k} - \mathrm{q}^{h,k}) \end{bmatrix}.$$

Then, $\hat{\mathrm{p}}_{\mathrm{y}}^k$ and $\hat{\mathrm{p}}_{\mathrm{q}}^k$ solve

$$\tilde{\mathrm{M}}^\ell \begin{pmatrix} \hat{\mathrm{p}}_{\mathrm{y}}^k \\ \hat{\mathrm{p}}_{\mathrm{q}}^k \end{pmatrix} = \tilde{\mathrm{b}}^\ell.$$

$\Diamond$

Finally, the RO gradient, approximation of(3.28) is

$$\nabla \hat{J}^\ell(\boldsymbol{\mu}) := \begin{pmatrix} \sum_{k=0}^K \alpha_k \langle \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{p}_{\mathrm{y},\boldsymbol{\mu}}^{\ell,k} \rangle_{\mathrm{A_y}} \\ -\sum_{k=0}^K \alpha_k \langle \mathrm{f}_{\mathrm{y}}^h(\mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}), \mathrm{p}_{\mathrm{y},\boldsymbol{\mu}}^{\ell,k} \rangle_{\mathrm{M_y}} \\ \sum_{k=0}^K \alpha_k \langle \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{p}_{\mathrm{q},\boldsymbol{\mu}}^{\ell,k} \rangle_{\mathrm{A_q}} \\ \sum_{k=0}^K \alpha_k \langle \mathrm{f}_{\mathrm{q}}^h(\mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}), \mathrm{p}_{\mathrm{q},\boldsymbol{\mu}}^{\ell,k} \rangle_{\mathrm{M_q}} \end{pmatrix} + \beta_J(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}). \tag{3.33}$$

Remark 3.13. We point out again the fact that all the evaluations in (3.33) can be made online-efficient, i.e. independent of dimension N. For example, the product

$$\langle \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{p}_{\mathrm{y},\boldsymbol{\mu}}^{\ell,k} \rangle_{\mathrm{A_y}} = \hat{\mathrm{y}}_{\boldsymbol{\mu}}^{k\top} \Psi_{\mathrm{y}}^\top \mathrm{A_y} \Psi_{\mathrm{py}} \hat{p}_{\mathrm{y},\boldsymbol{\mu}}^k$$

can be sped up by pre-evaluating and store the matrix (independent of $\boldsymbol{\mu}$) $\Psi_{\mathrm{y}}^\top \mathrm{A_y} \Psi_{\mathrm{py}}$. $\Diamond$

3.4 Hierarchical a-posteriori error estimates

We want to use the trust region algorithm presented in Section 2.4.2, but we need a-posteriori error estimates. While we can still evaluate projection-based error bounds, these will be not online efficient since the nonlinearity depends on the full dimension N. In this section we will define approximated hierarchical error estimates, extending the approach found in [29].

The hierarchical estimators are well-known, e.g., for adaptive finite elements; cf. [7, 13, 16], for instance. The idea is to use the difference between two approximations with different orders to estimate the RB error. Given RB spaces V^ℓ and $V_\circ^\ell$, with respective reduced bases associated with the matrices Ψ_{y}^ℓ and Ψ_{q}^ℓ, the RB errors are defined as as

$$E_{\mathrm{y}}^\ell(\boldsymbol{\mu}) := \left(\sum_{k=0}^K \alpha_k \, \| \mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k} \|_{\mathrm{W_y}}^2 \right)^{1/2}$$

and

$$E_{\mathrm{q}}^\ell(\boldsymbol{\mu}) := \left(\sum_{k=0}^K \alpha_k \, \| \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} \|_{\mathrm{W_q}}^2 \right)^{1/2},$$

where $\mathrm{W_y} = ((\langle \varphi_j, \varphi_i \rangle_V))_{0 \le i,j \le n}$ and $\mathrm{W_q} = ((\langle \varphi_j, \varphi_i \rangle_{V_\circ}))_{1 \le i,j \le n}$ are positive definite, symmetric weighting matrices. These errors approximate the errors $y^h - y^\ell$ and $q^h - q^\ell$ in the

$L^2(0,T;V)$- and $L^2(0,T;V_\circ)$-norms, respectively. If we had another couple of RB spaces, V^m and $V_\circ^m$ of dimensions respectively M_y and M_q, with $L_\mathrm{y} < M_\mathrm{y} \ll N$ and $L_\mathrm{q} < M_\mathrm{q} \ll N$ such that $E_\mathrm{y}^m(\boldsymbol{\mu}) \le \varepsilon$ and $E_\mathrm{q}^m(\boldsymbol{\mu}) \le \varepsilon$, then we have

$$\begin{aligned} E_\mathrm{y}^\ell(\boldsymbol{\mu})^2 &= \sum_{k=0}^K \alpha_k \left\| \mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k} \right\|_{\mathrm{W_y}}^2 \\ &\le 2 \sum_{k=0}^K \alpha_k \left\| \mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{m,k} \right\|_{\mathrm{W_y}}^2 + 2 \sum_{k=0}^K \alpha_k \left\| \mathrm{y}_{\boldsymbol{\mu}}^{m,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k} \right\|_{\mathrm{W_y}}^2 \\ &\le 2\varepsilon^2 + 2\Delta_\mathrm{y}^{\ell,m}(\boldsymbol{\mu})^2 \end{aligned}$$

with

$$\Delta_\mathrm{y}^{\ell,m}(\boldsymbol{\mu}) := \left(\sum_{k=0}^K \alpha_k \left\| \mathrm{y}_{\boldsymbol{\mu}}^{m,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k} \right\|_{\mathrm{W_y}}^2 \right)^{1/2},$$

and, similarly, $E_\mathrm{q}^\ell(\boldsymbol{\mu})^2 \le 2\varepsilon^2 + 2\Delta_\mathrm{q}^{\ell,m}(\boldsymbol{\mu})^2$ with

$$\Delta_\mathrm{q}^{\ell,m}(\boldsymbol{\mu}) := \left(\sum_{k=0}^K \alpha_k \left\| \mathrm{q}_{\boldsymbol{\mu}}^{m,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} \right\|_{\mathrm{W_q}}^2 \right)^{1/2}.$$

We point out that these estimates are online-efficient (i.e., their evaluation does not depend on the FE dimension N), since

$$\left\| \mathrm{y}_{\boldsymbol{\mu}}^{m,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k} \right\|_{\mathrm{W_y}}^2 = \left(\hat{\mathrm{y}}_{\boldsymbol{\mu}}^{m,k\top} \mathrm{W}_\mathrm{y}^{m,m} \hat{\mathrm{y}}_{\boldsymbol{\mu}}^{m,k} - 2\hat{\mathrm{y}}_{\boldsymbol{\mu}}^{m,k\top} \mathrm{W}_\mathrm{y}^{m,\ell} \hat{\mathrm{y}}_{\boldsymbol{\mu}}^{\ell,k} + \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k\top} \mathrm{W}_\mathrm{y}^{\ell,\ell} \hat{\mathrm{y}}_{\boldsymbol{\mu}}^{\ell,k} \right),$$

and

$$\left\| \mathrm{q}_{\boldsymbol{\mu}}^{m,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} \right\|_{\mathrm{W_q}}^2 = \left(\hat{\mathrm{q}}_{\boldsymbol{\mu}}^{m,k\top} \mathrm{W}_\mathrm{q}^{m,m} \hat{\mathrm{q}}_{\boldsymbol{\mu}}^{m,k} - 2\hat{\mathrm{q}}_{\boldsymbol{\mu}}^{k,m\top} \mathrm{W}_\mathrm{q}^{m,\ell} \hat{\mathrm{q}}_{\boldsymbol{\mu}}^{\ell,k} + \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k\top} \mathrm{W}_\mathrm{q}^{\ell,\ell} \hat{\mathrm{q}}_{\boldsymbol{\mu}}^{\ell,k} \right),$$

where $\{\hat{\mathrm{y}}_{\boldsymbol{\mu}}^{\ell,k}, \hat{\mathrm{q}}_{\boldsymbol{\mu}}^{\ell,k}\}_{k=0}^K$ (resp. $\{\hat{\mathrm{y}}_{\boldsymbol{\mu}}^{m,k}, \hat{\mathrm{q}}_{\boldsymbol{\mu}}^{m,k}\}_{k=0}^K$) are the solution of (3.30) in dimensions L_y and L_q (resp. M_y and M_q), and parameter $\boldsymbol{\mu}$, and

$$\begin{aligned} \mathrm{W}_\mathrm{y}^{m,m} &:= \Psi_\mathrm{y}^{m\top} \mathrm{W}_\mathrm{y} \Psi_\mathrm{y}^m \in \mathbb{R}^{M_\mathrm{y} \times M_\mathrm{y}}, & \mathrm{W}_\mathrm{q}^{m,m} &:= \Psi_\mathrm{q}^{m\top} \mathrm{W}_\mathrm{q} \Psi_\mathrm{q}^m \in \mathbb{R}^{M_\mathrm{q} \times M_\mathrm{q}}, \\ \mathrm{W}_\mathrm{y}^{m,\ell} &:= \Psi_\mathrm{y}^{m\top} \mathrm{W}_\mathrm{y} \Psi_\mathrm{y}^\ell \in \mathbb{R}^{M_\mathrm{y} \times L_\mathrm{y}}, & \mathrm{W}_\mathrm{q}^{m,\ell} &:= \Psi_\mathrm{q}^{m\top} \mathrm{W}_\mathrm{q} \Psi_\mathrm{y}^m \in \mathbb{R}^{M_\mathrm{q} \times M_\mathrm{q}}, \\ \mathrm{W}_\mathrm{y}^{\ell,\ell} &:= \Psi_\mathrm{y}^{\ell\top} \mathrm{W}_\mathrm{y} \Psi_\mathrm{y}^\ell \in \mathbb{R}^{L_\mathrm{y} \times L_\mathrm{y}}, & \mathrm{W}_\mathrm{q}^{\ell,\ell} &:= \Psi_\mathrm{q}^{\ell\top} \mathrm{W}_\mathrm{q} \Psi_\mathrm{q}^\ell \in \mathbb{R}^{L_\mathrm{q} \times L_\mathrm{q}}. \end{aligned}$$

To analyze the hierarchical error estimates it is useful to set $V^\ell \subset V^m$ and $V_\circ^\ell \subset V_\circ^m$ and to assume the following *saturation property* on the bigger RB spaces V^m and $V_\circ^m$: there exist $\sigma_\mathrm{y}, \sigma_\mathrm{q} \in (0,1)$ such that

$$E_\mathrm{y}^m(\boldsymbol{\mu}) \le \sigma_\mathrm{y} E_\mathrm{y}^\ell(\boldsymbol{\mu}) \quad \text{and} \quad E_\mathrm{q}^m(\boldsymbol{\mu}) \le \sigma_\mathrm{q} E_\mathrm{q}^\ell(\boldsymbol{\mu}) \qquad \text{for all } \boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}. \tag{3.34}$$

While it is obvious, for example, that if $V^\ell \subset V^m$ then $E_\mathrm{y}^m(\boldsymbol{\mu}) \le E_\mathrm{y}^\ell(\boldsymbol{\mu})$, we do need the factors σ_y and σ_q to later define the error estimates. This is a natural assumption since one expects the RB error to decrease with the increase of the RB dimension, but from the theoretical point of view it is indeed possible to find a $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$ such that $E_\mathrm{y}^m(\boldsymbol{\mu}) = E_\mathrm{y}^\ell(\boldsymbol{\mu})$. This condition enforces the idea that for all parameters in the admissible parameter space the bigger space is a better-approximating RB space. Hence, since it is not necessarily satisfied, it will be enforced during the construction of the spaces V^m and $V_\circ^m$.

Proposition 3.14. *If* (3.34) *holds, then for all* $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$

$$\frac{\Delta_{\mathrm{y}}^{\ell,m}(\boldsymbol{\mu})}{1+\sigma_{\mathrm{y}}} \leq E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu}) \leq \frac{\Delta_{\mathrm{y}}^{\ell,m}(\boldsymbol{\mu})}{1-\sigma_{\mathrm{y}}}, \tag{3.35a}$$

$$\frac{\Delta_{\mathrm{q}}^{\ell,m}(\boldsymbol{\mu})}{1+\sigma_{\mathrm{q}}} \leq E_{\mathrm{q}}^{\ell}(\boldsymbol{\mu}) \leq \frac{\Delta_{\mathrm{q}}^{\ell,m}(\boldsymbol{\mu})}{1-\sigma_{\mathrm{q}}}. \tag{3.35b}$$

Proof. We only prove (3.35a), since the proof of (3.35b) is similar. If $E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu}) = 0$, then $\Delta_{\mathrm{y}}^{\ell,m}(\boldsymbol{\mu}) = 0$ holds necessarily, and the inequalities (3.35a) follow. In the other case, we prove the following inequalities, that are equivalent to (3.35a):

$$1-\sigma_{\mathrm{y}} \leq \frac{\Delta_{\mathrm{y}}^{\ell,m}(\boldsymbol{\mu})}{E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu})} \leq 1+\sigma_{\mathrm{y}}. \tag{3.36}$$

For $k = 0, \ldots, K$, it holds

$$\|\mathrm{y}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{y}_{\boldsymbol{\mu}}^{m,k}\|_{\mathrm{W_y}}^2 = \|\mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}\|_{\mathrm{W_y}}^2 + \|\mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{m,k}\|_{\mathrm{W_y}}^2 - 2\langle \mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{m,k}\rangle_{\mathrm{W_y}}, \tag{3.37}$$

and, furthermore, using the Cauchy-Schwarz and Young inequalities we can write

$$\left|\langle \mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{m,k}\rangle_{\mathrm{W_y}}\right| \leq \frac{\beta}{2}\|\mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}\|_{\mathrm{W_y}}^2 + \frac{1}{2\beta}\|\mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{m,k}\|_{\mathrm{W_y}}^2 \tag{3.38}$$

for any $\beta > 0$. Let us now consider

$$\frac{\Delta_{\mathrm{y}}^{\ell,m}(\boldsymbol{\mu})^2}{E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu})^2} = \frac{\sum_{k=0}^{K} \alpha_k \, \|\mathrm{y}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{y}_{\boldsymbol{\mu}}^{m,k}\|_{\mathrm{W_y}}^2}{\sum_{k=0}^{K} \alpha_k \, \|\mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}\|_{\mathrm{W_y}}^2}.$$

By using (3.37) and (3.38), and by choosing the same β for all $k = 0, \ldots, K$, we get

$$\begin{aligned}\frac{\Delta_{\mathrm{y}}^{\ell,m}(\boldsymbol{\mu})^2}{E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu})^2} &\leq \frac{\sum_{k=0}^{K} \alpha_k \, \left[(1+\beta)\|\mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}\|_{\mathrm{W_y}}^2 + (1+1/\beta)\|\mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{m,k}\|_{\mathrm{W_y}}^2\right]}{\sum_{k=0}^{K} \alpha_k \, \|\mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}\|_{\mathrm{W_y}}^2} \\ &\leq (1+\beta) + (1+1/\beta)\sigma_{\mathrm{y}}^2.\end{aligned}$$

Since this holds for all choices of $\beta > 0$, it also holds for the minimum value

$$\frac{\Delta_{\mathrm{y}}^{\ell,m}(\boldsymbol{\mu})^2}{E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu})^2} \leq \min_{\beta>0}(1+\beta) + (1+1/\beta)\sigma_{\mathrm{y}}^2 = (1+\sigma_{\mathrm{y}})^2.$$

On the other hand, using (3.37) and (3.38), we get

$$\begin{aligned}\frac{\Delta_{\mathrm{y}}^{\ell,m}(\boldsymbol{\mu})^2}{E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu})^2} &\geq \frac{\sum_{k=0}^{K} \alpha_k \, \left[(1-\beta)\|\mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}\|_{\mathrm{W_y}}^2 + (1-1/\beta)\|\mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{m,k}\|_{\mathrm{W_y}}^2\right]}{\sum_{k=0}^{K} \alpha_k \, \|\mathrm{y}_{\boldsymbol{\mu}}^{h,k} - \mathrm{y}_{\boldsymbol{\mu}}^{\ell,k}\|_{\mathrm{W_y}}^2} \\ &\leq (1-\beta) + (1-1/\beta)\sigma_{\mathrm{y}}^2,\end{aligned}$$

and, by choosing the maximum value over all $\beta > 0$, we obtain

$$\frac{\Delta_{\mathrm{y}}^{\ell,m}(\boldsymbol{\mu})^2}{E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu})^2} \geq \max_{\beta>0}(1-\beta) + (1-1/\beta)\sigma_{\mathrm{y}}^2 = (1-\sigma_{\mathrm{y}})^2,$$

which proves (3.36). □

Therefore, we define the hierarchical error estimators in $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$ as

$$\Delta_{\mathrm{y}}^{\ell}(\boldsymbol{\mu}) = \Delta_{\mathrm{y}}^{\ell}(\boldsymbol{\mu}; m, \sigma_{\mathrm{y}}) = \frac{\Delta_{\mathrm{y}}^{\ell,m}(\boldsymbol{\mu})}{1-\sigma_{\mathrm{y}}} \quad \text{and} \quad \Delta_{\mathrm{q}}^{\ell}(\boldsymbol{\mu}) = \Delta_{\mathrm{q}}^{\ell}(\boldsymbol{\mu}; m, \sigma_{\mathrm{q}}) = \frac{\Delta_{\mathrm{q}}^{\ell,m}(\boldsymbol{\mu})}{1-\sigma_{\mathrm{q}}}. \tag{3.39}$$

Remark 3.15. The effectivities of the estimates in the parameter $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$ are denoted again as

$$\eta_{\mathrm{y}}(\boldsymbol{\mu}) := \frac{\Delta_{\mathrm{y}}^{\ell}(\boldsymbol{\mu})}{E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu})} \quad \text{and} \quad \eta_{\mathrm{q}}(\boldsymbol{\mu}) := \frac{\Delta_{\mathrm{q}}^{\ell}(\boldsymbol{\mu})}{E_{\mathrm{q}}^{\ell}(\boldsymbol{\mu})}. \tag{3.40}$$

Then, we infer from (3.40), (3.35) and (3.39) that

$$1 \leq \eta_{\mathrm{y}}(\boldsymbol{\mu}) \leq \tilde{\eta}_{\mathrm{y}} := \frac{1+\sigma_{\mathrm{y}}}{1-\sigma_{\mathrm{y}}} \quad \text{and} \quad 1 \leq \eta_{\mathrm{q}}(\boldsymbol{\mu}) \leq \tilde{\eta}_{\mathrm{q}} := \frac{1+\sigma_{\mathrm{q}}}{1-\sigma_{\mathrm{q}}} \qquad \text{for all } \boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}. \tag{3.41}$$

Hence we observe that the closer σ_{y} (respectively σ_{q}) is to 0, the more accurate the estimate $\Delta_{\mathrm{y}}^{\ell}(\boldsymbol{\mu})$ (resp. $\Delta_{\mathrm{q}}^{\ell}(\boldsymbol{\mu})$) is for any $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$. ◊

There are two important choices to be made: the first is how to compute σ_{y} and σ_{q}, and the second is how to choose the RB spaces V^m and $V_\circ^m$. For the first issue, we compute approximations $\tilde{\sigma}_{\mathrm{y}}$ and $\tilde{\sigma}_{\mathrm{q}}$ on a training set, namely

$$\sigma_{\mathrm{y}} \approx \tilde{\sigma}_{\mathrm{y}} := \max_{\boldsymbol{\mu}\in\mathcal{P}_{\mathsf{discr}}} \frac{E_{\mathrm{y}}^{m}(\boldsymbol{\mu})}{E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu})} \quad \text{and} \quad \sigma_{\mathrm{q}} \approx \tilde{\sigma}_{\mathrm{q}} := \max_{\boldsymbol{\mu}\in\mathcal{P}_{\mathsf{discr}}} \frac{E_{\mathrm{q}}^{m}(\boldsymbol{\mu})}{E_{\mathrm{q}}^{\ell}(\boldsymbol{\mu})}. \tag{3.42}$$

This is where the hierarchical error estimate is approximated, since it is unfortunately not possible to evaluate σ_{y} and σ_{q} on the full parameter space. The inequality $\tilde{\sigma}_{\mathrm{y}} \leq \sigma_{\mathrm{y}}$ (resp. $\tilde{\sigma}_{\mathrm{q}} \leq \sigma_{\mathrm{q}}$) holds, and $\tilde{\sigma}_{\mathrm{y}} \nearrow \sigma_{\mathrm{y}}$ (resp. $\tilde{\sigma}_{\mathrm{q}} \nearrow \sigma_{\mathrm{q}}$) with $|\mathcal{P}_{\mathsf{discr}}| \to \infty$, namely by enlarging the training set we can improve the accuracy of the estimates. Let us mention that the computation of (3.42) can be done efficiently by parallelization.

Next, the second question is answered by defining V^m (resp. $V_\circ^m$) as an expansion of V^ℓ (resp. $V_\circ^\ell$), namely $V^\ell \subset V^m$ and $V_\circ^\ell \subset V_\circ^m$. In particular, given V^ℓ and $V_\circ^\ell$, we will find $\tilde{V}^m \perp V^\ell$ and $\tilde{V}_\circ^m \perp V_\circ^\ell$ and then define

$$V^m = V^\ell \oplus \tilde{V}^m \qquad \text{and} \qquad V_\circ^m = V_\circ^\ell \oplus \tilde{V}_\circ^m, \tag{3.43}$$

where $\oplus$ is the direct sum (cf., e.g., [7]). A greedy algorithm for the evaluation of global RB spaces is shown in Algorithm 5. The inputs of the algorithm are the tolerance $\tau > 0$, an initial parameter $\boldsymbol{\mu} = (\mu_i)_{1\leq i\leq 4} \in \mathcal{P}_{\mathsf{ad}}$, a training set $\mathcal{P}_{\mathsf{discr}}$ and the maximum RB cardinality $L_{\mathsf{max}} > 0$. The maximum number of bases elements is a safeguard in case the maximum error does not go under tolerance.

Algorithm 5: Weak Greedy algorithm for coupled nonlinear model

Input: $\tau > 0$, $\hat{\boldsymbol{\mu}} \in \mathcal{P}_{\mathsf{ad}}$, $\mathcal{P}_{\mathsf{discr}} \subset \mathcal{P}_{\mathsf{ad}}$, $L_{\mathsf{max}} > 0$;

1 Create RB spaces V^ℓ, $V_\circ^\ell$, $\tilde{V}^m$ and $\tilde{V}_\circ^m$ in $\hat{\boldsymbol{\mu}}$;
2 $V^m \leftarrow V^\ell \oplus \tilde{V}^m$ and $V_\circ^m \leftarrow V_\circ^\ell \oplus \tilde{V}_\circ^m$ as in (3.43) ;
3 Evaluate $\tilde{\sigma}_{\mathrm{y}}$ and $\tilde{\sigma}_{\mathrm{q}}$ as in (3.42) ;
4 **while** $\tilde{\sigma}_{\mathrm{y}} \geq 1$ ***or*** $\tilde{\sigma}_{\mathrm{q}} \geq 1$ **do**
5 Enrich $\tilde{V}^m$ and/or $\tilde{V}_\circ^m$;
6 Set $V^m \leftarrow V^\ell \oplus \tilde{V}^m$ and $V_\circ^m \leftarrow V_\circ^\ell \oplus \tilde{V}_\circ^m$;
7 Evaluate $\tilde{\sigma}_{\mathrm{y}}$ and $\tilde{\sigma}_{\mathrm{q}}$ as in (3.42) ;
8 Evaluate $\hat{\boldsymbol{\mu}} = \arg\max_{\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{discr}}} e^\ell(\boldsymbol{\mu}) := (\Delta_{\mathrm{y}}^\ell(\boldsymbol{\mu}) + \Delta_{\mathrm{q}}^\ell(\boldsymbol{\mu}))/2$ and $\hat{e} = e(\hat{\boldsymbol{\mu}})$;
9 **while** $\hat{e} > \tau$ ***and*** $L_{\mathrm{y}} + L_{\mathrm{q}} < L_{\mathsf{max}}$ **do**
10 **if** $\Delta_{\mathrm{y}}^\ell(\hat{\boldsymbol{\mu}}) > \tau$ **then** Enrich V^ℓ in $\hat{\boldsymbol{\mu}}$;
11 **if** $\Delta_{\mathrm{q}}^\ell(\hat{\boldsymbol{\mu}}) > \tau$ **then** Enrich $V_\circ^\ell$ in $\hat{\boldsymbol{\mu}}$;
12 Set $V^m \leftarrow V^\ell \oplus \tilde{V}^m$ and $V_\circ^m \leftarrow V_\circ^\ell \oplus \tilde{V}_\circ^m$;
13 **while** $\tilde{\sigma}_{\mathrm{y}} \geq 1$ ***or*** $\tilde{\sigma}_{\mathrm{q}} \geq 1$ **do**
14 Enrich $\tilde{V}^m$ and $\tilde{V}_\circ^m$;
15 Set $V^m \leftarrow V^\ell \oplus \tilde{V}^m$ and $V_\circ^m \leftarrow V_\circ^\ell \oplus \tilde{V}_\circ^m$;
16 Evaluate $\tilde{\sigma}_{\mathrm{y}}$ and $\tilde{\sigma}_{\mathrm{q}}$ as in (3.42) ;
17 Evaluate $\hat{\boldsymbol{\mu}} = \arg\max_{\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{discr}}} e^\ell(\boldsymbol{\mu}) := (\Delta_{\mathrm{y}}^\ell(\boldsymbol{\mu}) + \Delta_{\mathrm{q}}^\ell(\boldsymbol{\mu}))/2$ and $\hat{e} = e(\hat{\boldsymbol{\mu}})$;

In line 2 the computation of the initial RB spaces is described. As a first step, we create RB spaces V^ℓ and $V_\circ^\ell$ using POD with the state variables' snapshots $\{\mathrm{y}_{\boldsymbol{\mu}}^{h,k}\}_{k=0}^K$ and $\{\mathrm{q}_{\boldsymbol{\mu}}^{h,k}\}_{k=0}^K$. Then, to define $\tilde{V}^m$ and $\tilde{V}_\circ^m$ we use the sensitivity variables (as suggested in [29]). In Section A.2 we include details on the FO discretization of sensitivity equations and variables. After evaluating the sensitivity vectors $\{\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{y},h,k}, \mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{q},h,k}\}_{k=0}^K$ for $j = 1, \dots, 4$, i.e. the FO sensitivities, we orthogonalize the vectors $\{\mathrm{y}_{\boldsymbol{\mu}}^{h,k}, \mathrm{s}_{i,\boldsymbol{\mu}}^{\mathrm{y},h,k}\}_{k=0}^K$ (resp. $\{\mathrm{q}_{\boldsymbol{\mu}}^{h,k}, \mathrm{s}_{i,\boldsymbol{\mu}}^{\mathrm{q},h,k}\}_{k=0}^K$) for $i = 1, \dots, 4$ w.r.t. V^ℓ (resp. $V_\circ^\ell$). Finally, we use POD on the two sets of vectors to evaluate $\tilde{V}^m$ and $\tilde{V}_\circ^m$. Then, line 6 uses (3.43) to define the spaces V^m and $V_\circ^m$. We observe that by construction $V^\ell \perp \tilde{V}^m$ and $V_\circ^\ell \perp \tilde{V}_\circ^m$, though it is possible to enforce this condition numerically. Furthermore, we add that at first we choose $M_{\mathrm{y}} = L_{\mathrm{y}} + 2$ and $M_{\mathrm{q}} = L_{\mathrm{q}} + 2$, namely we construct the initial $\tilde{V}^m$ and $\tilde{V}_\circ^m$ of dimension 2. This choice has initially been instinctive and arbitrary; it proved to be a good idea in view of the satisfactory results obtained: the m-spaces are bigger enough than the ℓ-spaces to improve the approximation, but not too big to waste computational resources.

Lines 7 and 16 evaluates $\tilde{\sigma}_{\mathrm{y}}$ and $\tilde{\sigma}_{\mathrm{q}}$. If $\tilde{\sigma}_{\mathrm{y}} \geq 1$ or $\tilde{\sigma}_{\mathrm{q}} \geq 1$ (as in line 4), it means that the saturation property is violated. This has so far not happened to us, confirming the expectation that a bigger RB space is a better approximation of the FE space for any $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$. But in case this condition is violated, our algorithm enriches the spaces $\tilde{V}^m$ and $\tilde{V}_\circ^m$ in the parameters where such condition was violated. Enriching here means evaluating FO vectors, orthogonalizing w.r.t. V^m or $V_\circ^m$, and then using POD to add some new bases to them. This procedure is then applied as long as the saturation property is violated.

Lines 8 and 17 evaluate the error measure we want to lower. If this error is higher than a certain tolerance, it means that either $\Delta_{\mathrm{y}}^{\ell}(\hat{\boldsymbol{\mu}}) > \tau$ or $\Delta_{\mathrm{q}}^{\ell}(\hat{\boldsymbol{\mu}}) > \tau$ (or, eventually, both). Then the enrichment is done by evaluating the states' snapshots, orthogonalizing them w.r.t. the bases we already have and then using POD to get new elements. We can then add as many elements as needed in order to guarantee $\Delta_{\mathrm{y}}^{\ell}(\hat{\boldsymbol{\mu}}) \leq \tau$ and $\Delta_{\mathrm{q}}^{\ell}(\hat{\boldsymbol{\mu}}) \leq \tau$.

We then check $\tilde{\sigma}_{\mathrm{y}}$ and $\tilde{\sigma}_{\mathrm{q}}$ again. Since V^{ℓ} and $V_{\circ}^{\ell}$ could have been enriched, we need to orthogonalize $\tilde{V}^{m}$ and $\tilde{V}_{\circ}^{m}$ with respect to them. It still holds that $V^{\ell} \subset V^{m}$ and $V_{\circ}^{\ell} \subset V_{\circ}^{m}$, but if the saturation property fails, we enforce it by enriching $\tilde{V}^{m}$ or $\tilde{V}_{\circ}^{m}$.

Example 3.16. Let us take into consideration the problem and discretization introduced in Example 3.7, and let us see the performance of the greedy algorithm on this model case. Parameter bounds are chosen as $\mu_{\mathsf{a},i} = 1$ and $\mu_{\mathsf{b},i} = 5$ for $i = 1, \ldots, 4$. The set $\mathcal{P}_{\mathsf{discr}}$ is chosen as a uniform grid $5 \times 5 \times 5 \times 5$ on $\mathcal{P}_{\mathsf{ad}}$, such that $\mathcal{P}_{\mathsf{discr}} = \{\boldsymbol{\mu} \in \mathbb{R}^4 \,|\, \mu_i \in \{1, 2, 3, 4, 5\} \text{ for } i = 1, 2, 3, 4\}$. Furthermore, we choose as greedy parameters $\tau = 10^{-4}$ and $L_{\mathsf{max}} = 50$. We analyze the results of the greedy algorithm and the generated RB spaces using three different input functions chosen arbitrarily, namely

$$\begin{aligned} u_1(t) &:= 1, \\ u_2(t) &:= -\mathbf{1}_{[0,0.75)}(t) + \mathbf{1}_{[0.75,1]}(t), \\ u_3(t) &:= 0.5 \cdot \cos(10t) + 0.4 \cdot \sin(20t), \end{aligned}$$

for $t \in [0, T]$, where $\mathbf{1}_I(t)$ is the indicator function of interval I. We will compare the overall execution time of the Algorithm 5, the number of bases generated in all spaces, the approximated values $\tilde{\sigma}_{\mathrm{y}}$ and $\tilde{\sigma}_{\mathrm{q}}$ and the respective approximated efficiency, $\tilde{\eta}_{\mathrm{y}}$ and $\tilde{\eta}_{\mathrm{q}}$ given as (3.41); cf. Table 3.1. Furthermore, given a set $\mathcal{P}_{\mathsf{discr}}$ of 100 randomly chosen parameters in $\mathcal{P}_{\mathsf{ad}}$, we evaluate the average times of evaluating FO and RB states, maximum test errors

$$\hat{E}_{\mathrm{y}}^{\ell} = \max_{\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{discr}}} E_{\mathrm{y}}^{\ell}(\boldsymbol{\mu}) \qquad \text{and} \qquad \hat{E}_{\mathrm{q}}^{\ell} = \max_{\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{discr}}} E_{\mathrm{q}}^{\ell}(\boldsymbol{\mu})$$

and maximum test efficiencies

$$\hat{\eta}_{\mathrm{y}} = \max_{\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{discr}}} \eta_{\mathrm{y}}(\boldsymbol{\mu}) \qquad \text{and} \qquad \hat{\eta}_{\mathrm{q}} = \max_{\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{discr}}} \eta_{\mathrm{q}}(\boldsymbol{\mu}),$$

where $\eta_{\mathrm{y}}(\boldsymbol{\mu})$ and $\eta_{\mathrm{q}}(\boldsymbol{\mu})$ are defined in (3.40).

	u_1	u_2	u_3
Time greedy	5201 s	5311 s	5796 s
Dimensions L_{y}, L_{q}	8, 4	8, 4	7, 4
Dimensions m_{y}, m_{q}	10, 6	10, 6	9, 6
Approximated σ's: $\tilde{\sigma}_{\mathrm{y}}$, $\tilde{\sigma}_{\mathrm{q}}$	0.37, 0.32	0.38, 0.16	0.18, 0.71
Approx. max. efficiencies $\tilde{\eta}_{\mathrm{y}}$, $\tilde{\eta}_{\mathrm{q}}$	1.23, 2.16	2.21, 1.37	1.43, 6.07
Avg. time FO	2.38 s	2.42 s	2.56 s
Avg. time RB	0.10 s	0.10 s	0.15 s
Max. test errors: $\hat{E}_{\mathrm{y}}^{\ell}$, $\hat{E}_{\mathrm{q}}^{\ell}$	7.89e-06, 1.27e-05	7.11e-06, 2.20e-05	7.91e-06, 2.19e-05
Max. test efficiencies: $\hat{\eta}_{\mathrm{y}}$, $\hat{\eta}_{\mathrm{q}}$	1.61, 1.46	1.75, 1.19	1.22, 3.53

Table 3.1: Comparison of the test runs for three different control inputs u.

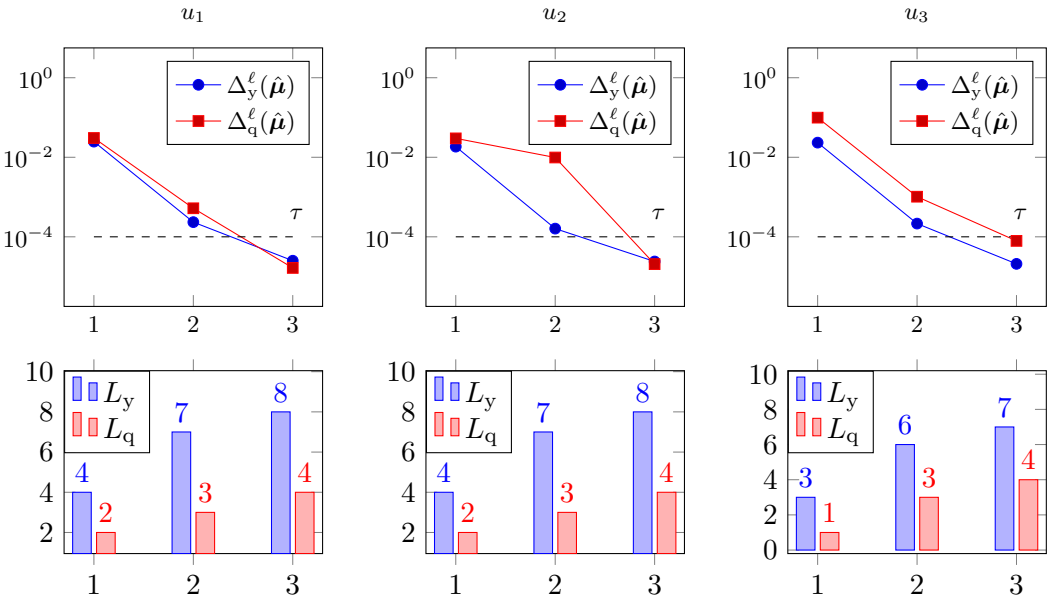


Figure 3.2: Visual comparison of the Greedy results for the three different control inputs. In both rows, the current greedy iteration is indicated on the x-axis: in the first row we can see how the convergence of the RB error estimators under tolerance is reached in three iterations, while in the second row we see the growing of the dimensions of the RB spaces.

In Figure 3.2 we also plot visual results of the greedy algorithm in the three cases, showing the real and approximated error, and the RO dimensions. The overall algorithm time does not change significantly for our different inputs. The evaluation of $\hat{e}$ in lines 8 and 17 does not take much CPU time; on the contrary, the evaluation of (3.42) is necessarily more expensive, because the evaluation of $\tilde{\sigma}_y$ and $\tilde{\sigma}_q$ uses the FO solutions, while the evaluation of $e^\ell(\boldsymbol{\mu})$ does not. We can see that the bigger spaces V^m and $V_\circ^m$ are in fact of dimensions $M_y = L_y + 2$ and $M_q = L_q + 2$, meaning that once $\tilde{V}^m$ and $\tilde{V}_\circ^m$ are created at the beginning they are not enriched anymore, namely no more bases are added to them since the conditions on lines 4 and 13 of Algorithm 5 are never true. The value of $\tilde{\sigma}_y$ and $\tilde{\sigma}_q$ change significantly in the different tests, but in the end we see that they hold good approximated max efficiencies $\tilde{\eta}_y$ and $\tilde{\eta}_q$, which we can read as "efficiency estimators" for the test efficiencies $\hat{\eta}_y$ and $\hat{\eta}_q$. ◊

We conclude the discussion on the state variable estimates with a comment on the hierarchical estimators. Even if, as we have stressed, these are just approximations of the estimators, they seem tight even with a relatively small effort; indeed, if the time of evaluation of an RB solution amounts to about 5% of the time evaluating an FO solution, we can expect the time to evaluate the estimator to total about 10% of the time evaluating the true error, since the error estimator consists in evaluating two RB solutions. Once defined the estimates for the state errors $E_y^\ell(\boldsymbol{\mu})$ and $E_q^\ell(\boldsymbol{\mu})$, we will now turn our attention to the cost function, and we will define an estimator for the error of the RO cost (3.31), i.e. for

$$E_J^\ell(\boldsymbol{\mu}) = \hat{J}^h(\boldsymbol{\mu}) - \hat{J}^\ell(\boldsymbol{\mu}).$$

The approximated a-posteriori error estimator for the cost is defined in the next proposition.

Proposition 3.17. *For all $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$, the following estimate on the RB error of the cost functional holds:*

$$\left|E_J^\ell(\boldsymbol{\mu})\right| \leq \Delta_J^\ell(\boldsymbol{\mu}), \tag{3.44}$$

where

$$\Delta_J^\ell(\boldsymbol{\mu}) := \frac{\alpha_J L^4}{2\pi^4}\Delta_{\mathrm{q}}^\ell(\boldsymbol{\mu})^2 + \frac{\alpha_J L^2}{\pi^2}\Delta_{\mathrm{q}}^\ell(\boldsymbol{\mu})\sqrt{\tilde{J}^\ell(\boldsymbol{\mu})} \tag{3.45}$$

and $\tilde{J}^\ell(\boldsymbol{\mu}) := \sum_{k=0}^K \alpha_k \left\|\mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\right\|_{\mathrm{M_q}}^2$.

Proof. As a first step, the following equality holds

$$E_J^\ell(\boldsymbol{\mu}) = \frac{\alpha_J}{2}\sum_{k=0}^K \alpha_k \left(\langle \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{w}^{h,k}, \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{w}^{h,k}\rangle_{\mathrm{M_q}} - \langle \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}, \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\rangle_{\mathrm{M_q}}\right). \tag{3.46}$$

Then, adding and subtracting the term $\langle \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{w}^{h,k}, \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\rangle_{\mathrm{M_q}}$ to each element in the sum in (3.46) we can write

$$\begin{aligned}
&\langle \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{w}^{h,k}, \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{w}^{h,k}\rangle_{\mathrm{M_q}} - \langle \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}, \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\rangle_{\mathrm{M_q}} \\
&= \langle \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{w}^{h,k}, \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}\rangle_{\mathrm{M_q}} + \langle \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\rangle_{\mathrm{M_q}} \\
&= \langle \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{q}_{\boldsymbol{\mu}}^{h,k} + \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - 2\mathrm{w}^{h,k}\rangle_{\mathrm{M_q}} \\
&= \langle \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}\rangle_{\mathrm{M_q}} + 2\langle \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\rangle_{\mathrm{M_q}}.
\end{aligned}$$

Hence,

$$|E_J^\ell(\boldsymbol{\mu})| \leq \frac{\alpha_J}{2}\sum_{k=0}^K \alpha_k \left\|\mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}\right\|_{\mathrm{M_q}}^2 + \alpha_J \sum_{k=0}^K \alpha_k \left|\langle \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\rangle_{\mathrm{M_q}}\right|. \tag{3.47}$$

Using (3.3), (3.35b) and (3.39) we can estimate

$$\sum_{k=0}^K \alpha_k \left\|\mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}\right\|_{\mathrm{M_q}}^2 \leq c_{\mathsf{P}}^2 E_{\mathrm{q}}^\ell(\boldsymbol{\mu})^2 \leq c_{\mathsf{P}}^2 \Delta_{\mathrm{q}}^\ell(\boldsymbol{\mu})^2. \tag{3.48}$$

On the other hand, using the Cauchy-Schwarz inequality and (3.48) we get

$$\begin{aligned}
&\sum_{k=0}^K \alpha_k \left|\langle \mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}, \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\rangle_{\mathrm{M_q}}\right| \\
&\qquad \leq \left(\sum_{k=0}^K \alpha_k \left\|\mathrm{q}_{\boldsymbol{\mu}}^{h,k} - \mathrm{q}_{\boldsymbol{\mu}}^{\ell,k}\right\|_{\mathrm{M_q}}^2\right)^{1/2} \left(\sum_{k=0}^K \alpha_k \left\|\mathrm{q}_{\boldsymbol{\mu}}^{\ell,k} - \mathrm{w}^{h,k}\right\|_{\mathrm{M_q}}^2\right)^{1/2} \leq c_{\mathsf{P}} \Delta_{\mathrm{q}}^\ell(\boldsymbol{\mu})\sqrt{\tilde{J}^\ell(\boldsymbol{\mu})}.
\end{aligned} \tag{3.49}$$

Inserting (3.48) and (3.49) into (3.47) we get

$$|\hat{J}^h(\boldsymbol{\mu}) - \hat{J}^\ell(\boldsymbol{\mu})| \leq \frac{\alpha_J c_{\mathsf{P}}^2}{2}\Delta_{\mathrm{q}}^\ell(\boldsymbol{\mu})^2 + \alpha_J c_{\mathsf{P}} \Delta_{\mathrm{q}}^\ell(\boldsymbol{\mu})\sqrt{\tilde{J}^\ell(\boldsymbol{\mu})}.$$

Using $c_{\mathsf{P}} = (L/\pi)^2$ the optimal Poincaré constant in $[0, L]$, we finally get (3.45). □

3.5 MOR-based parameter optimization

We solve the optimization problem $(\hat{\mathrm{P}}_2)$ in a trust-region (TR) framework using the algorithm discussed in the previous chapter. The only difference now is that we are using non-certified error estimates, since as we discussed σ_{y} and σ_{q} are approximated. In particular, they might change at each trust-region iteration n, since we define $\sigma_{\mathrm{y}} := \sigma_{\mathrm{y}}^{(n)} := E_{\mathrm{y}}^m(\boldsymbol{\mu}^{(n)})/E_{\mathrm{y}}^\ell(\boldsymbol{\mu}^{(n)})$ and $\sigma_{\mathrm{q}} := \sigma_{\mathrm{q}}^{(n)} := E_{\mathrm{q}}^m(\boldsymbol{\mu}^{(n)})/E_{\mathrm{q}}^\ell(\boldsymbol{\mu}^{(n)})$ at each iteration. Hence, while $\sigma_{\mathrm{q}}^{(n)}$ is exact in $\boldsymbol{\mu}^{(n)}$, we expect the estimate (3.44) to hold in a neighborhood of such parameter.

Another difference with the algorithm previously discussed is that we do not build RO spaces progressively, by adding spaces at each step. On the opposite, we build completely new RO spaces of fixed dimension $L_{\mathrm{py}} = L_{\mathrm{pq}} = 60$ that overwrite the previous RO spaces. As discussed in [6], in each step of the trust region algorithm we are obliged to evaluate some of the FO quantities, for example the adjoint variable (in order to evaluate the FO gradient and the exit criterion $g^h(\boldsymbol{\mu}) \leq \tau$). Hence, we make best use of the FO variables and for a little more effort (the application of the POD method), we evaluate new RO adjoint spaces and matrices.

Some of the modification – such as the approximated error estimate or the use of the inexact RO gradient – might break the convergence of the algorithm. However, we have always noticed to local optima and the FO exit condition $g^h(\boldsymbol{\mu}) \leq \tau$ is satisfied on exit.

We will now show the accuracy and the speedup of the TR optimization algorithm using the hierarchical error estimation. Let $\Omega = (0,1)$, $T = 1$, $\kappa_1(x) \equiv \kappa_2(x) \equiv 1$, $y_\circ(x) \equiv 6$ and $\mathcal{P}_{\mathsf{ad}}$ defined as in Example 3.16. Furthermore, let $\alpha_J = 10^6$, $\beta_J = 10^{-7}$, $\boldsymbol{\mu}^{(0)} = \hat{\boldsymbol{\mu}} = (3,3,3,3)$ (i.e. the middle point of $\mathcal{P}_{\mathsf{ad}}$) and let $u(t) = -3 \cdot \mathbf{1}_{[0,0.75)}(t) + 3 \cdot \mathbf{1}_{[0.75,1]}(t)$. As discretization dimensions we choose $N = 201$ (linear Lagrangian elements on 200 spatial nodes) and $K = 201$, and as data we use a noisy virtual measurement of the q state variable corresponding to the hidden parameter $\boldsymbol{\mu}^* = (2,3,4,5)$ similarly to (2.31), namely

$$\mathrm{w}_j^{h,k} = \mathrm{q}_{\boldsymbol{\mu}^* j}^{h,k} + \varepsilon_j^k, \qquad \text{where } \varepsilon_j^k \sim \mathcal{N}(0, \sigma_{\mathsf{d}}^2) \quad \text{for any } j = 1, \dots, N,\ k = 0, \dots, K,$$

where $\sigma_{\mathsf{d}} = 10^{-3}$ is the simulated measurement standard deviation.

As in the previous chapter, we compare the performance of the trust region optimization algorithm with the results of the FO optimization problem. The results are shown in Table 3.2. The speedup of the TR optimization algorithm in the overall CPU time is about 12, and we can see how the number of iterations and FO evaluations are much smaller. In the last two columns we can see the norms of the absolute and relative error between the true hidden parameter and the solution of the parameter optimization algorithm, i.e. $\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{abs}} := \|\boldsymbol{\mu}^* - \boldsymbol{\mu}^{\mathsf{opt}}\|_2$ and $\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{rel}} := \|\boldsymbol{\mu}^* - \boldsymbol{\mu}^{\mathsf{opt}}\|_2 / \|\boldsymbol{\mu}^*\|_2$. As shown, both optimization results are accurate enough. To show the accuracy of the RB approximation, in Figure 3.3 the errors between the FO and RO approximations corresponding to the underlying hidden parameter $\boldsymbol{\mu}^*$ are plotted.

The speedup given by the TR optimization, shown in Table 3.2 can be even higher when the advantage given by the evaluation of the RB solution in comparison with the FO

	Time	Iterations	FO evaluations	$\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{abs}}$	$\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{rel}}$
FO optimization	290 s	36	51	0.074	0.010
TR-RB	23 s	26	3	0.0706	0.0096

Table 3.2: Results for $\boldsymbol{\mu}^* = (2,3,4,5)$ and $u(t) = -3 \cdot \mathbf{1}_{[0,0.75)}(t) + 3 \cdot \mathbf{1}_{[0.75,1]}(t)$ using linear Lagrangian elements.

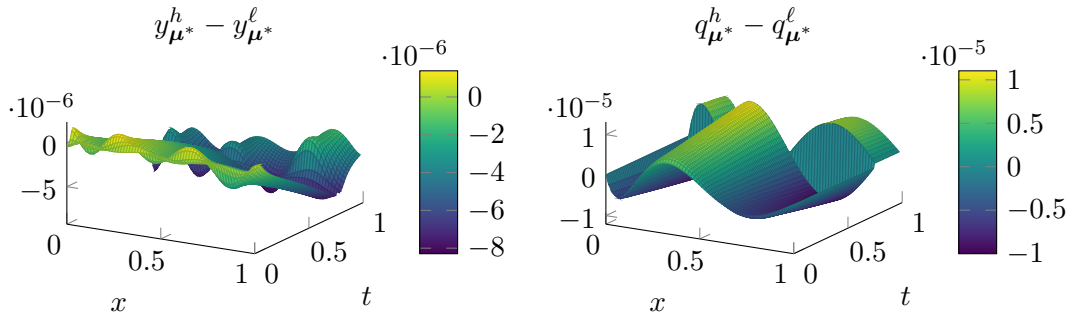


Figure 3.3: Errors between FO and RO solutions for $\boldsymbol{\mu}^* = (2,3,4,5)$ and $u(t) = -3 \cdot \mathbf{1}_{[0,0.75)}(t) + 3 \cdot \mathbf{1}_{[0.75,1]}(t)$ using linear Lagrangian elements. We observe a shock (i.e., a sudden change) in the error, deriving from the sudden change in the input function at $t = 0.75$. However, the overall errors stay under the threshold $\Delta^\ell_{\mathrm{y}}, \Delta^\ell_{\mathrm{q}} \leq \tau_{\mathrm{y,q}} := 10^{-4}$ (enforced at each RB enrichment).

solution is higher. For example, using quadratic Lagrangian elements (so that $N = 401$) the computational time for evaluating a FO solution is a bit higher (3 seconds) and the optimization algorithm takes longer. The results are shown in Table 3.3. Therefore in this case the speedup of the overall optimization with respect to the CPU time is about 18, and it is reasonable to believe that it can get even higher for more accurate discretizations, i.e. when the FO approximation are more expensive.

	Time	Iterations	FO evaluations	$\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{abs}}$	$\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{rel}}$
FO optimization	993 s	36	44	0.055	0.007
TR-RB	55 s	25	3	0.048	0.007

Table 3.3: Results for $\boldsymbol{\mu}^* = (2,3,4,5)$ and $u(t) = -3 \cdot \mathbf{1}_{[0,0.75)}(t) + 3 \cdot \mathbf{1}_{[0.75,1]}(t)$ using quadratic Lagrangian elements.

As a further example, we simply change the hidden parameter to $\boldsymbol{\mu}^* = (4,4,2,1.5)$ and use linear Lagrangian elements. As we can see in Table 3.4, the results do not change significantly with respect to results seen in Table 3.2, both in the overall time and in number of iterations.

On the other hand, if we change the input u to, say, $u(t) = 0.5 \cdot \cos(10t) + 0.4 \cdot \sin(20t)$, then the hidden parameter (in this example again $\boldsymbol{\mu}^* = (2,3,4,5)$) is not well approximated both for the FO and the RO approximations; cf. Table 3.5.

This failed optimization is what motivates the input design problem. If we are not careful, using an input could lead to convergence to a local optimum that is not the true hidden

	Time	Iterations	FO evaluations	$\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{abs}}$	$\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{rel}}$
FO optimization	338 s	40	46	0.034	0.0054
TR-RB	51 s	33	5	0.034	0.0054

Table 3.4: Results for $\boldsymbol{\mu}^* = (4, 4, 2, 1.5)$ and $u(t) = -3 \cdot \mathbf{1}_{[0,0.75)}(t) + 3 \cdot \mathbf{1}_{[0.75,1]}(t)$ using linear Lagrangian elements.

	Time	Iterations	FO evaluations	$\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{abs}}$	$\mathsf{e}_{\boldsymbol{\mu}}^{\mathsf{rel}}$
FO optimization	90 s	14	15	1.0150	0.1381
TR-RB	46 s	15	7	1.0180	0.1385

Table 3.5: Results for $\boldsymbol{\mu}^* = (2, 3, 4, 5)$ and $u(t) = 0.5 \cdot \cos(10t) + 0.4 \cdot \sin(20t)$ using linear Lagrangian elements.

parameter. Noticing how strong the impact of the input is on the parameter estimation, the next chapter will address exactly this: even if we start from a "bad" input, i.e. one that gives such bad results in the parameter optimization, how can we retrieve a "good" one, i.e. one whose parameter optimization converges to the true hidden parameter?

Chapter 4

Optimal input design

In the dealing of non-convex optimization problems, it is possible to converge to a local minimum, failing to approximate the true hidden parameter. As observed out in the last example of Section 3.5 and summarized in Table 3.5, the result of the parameter estimation can depend strongly on the chosen control, even leading to convergence to the true or wrong results. In order to gain confidence in the accuracy of the optimized parameter, we want to find (or *design*) an "optimal" control (or input), with the purpose of minimizing, at least qualitatively, the uncertainty of the estimated parameter.

A concept that we are going to use and better define later is the *observability* of a parameter, which quantifies how much information we can gather of a parameter given the current settings – or, as in this case, given a certain input function. For example, some of the parameters are more sensitive to extreme *states of charge* (namely, when the battery charge is almost full or empty), or on the other hand some parameters are more sensitive to fast changes of applied current (the so-called *high frequency dynamics*).

Designing the optimal input revolves around maximizing the observability of the true battery parameters in order to make its estimation easier. However, since we assume them to be unknown, we can only increase the observability of arbitrary given parameters. Following standard ideas of experimental design ([25, 50, 3]), we use an adaptive algorithm that, starting from some initial guesses for the parameter $\boldsymbol{\mu}^0$ and sequentially generate new inputs $u^{[n]}$ at each iteration $n > 0$. Each of these inputs is designed in order to maximize the observability of the current parameter, and with it we generate data and optimize the parameter $\boldsymbol{\mu}^n$; the inputs are specifically chosen in order to improve or facilitate the parameter optimization and the convergence to the true, hidden parameter $\boldsymbol{\mu}^*$. Hence, the result of the input design algorithm is not intended to be a single optimal input, but rather a collection of inputs that collects the most information.

In industrial applications the standard applied currents are many hours long, in order to observe the battery as long as possible as collect enough information to infer the tens of parameters. We are going to try to create input current profiles much shorter than these, lasting just a few minutes. One reason to consider the result of the design as an optimal collection of inputs is that in such a complex setting we cannot expect to maximize the observability of all the parameters using just a small amount of data; on the opposite, we

try to capture different pieces of information with different settings – or inputs.

In this chapter we introduce the input design algorithm and its motivation, to then apply this algorithm to the nonlinear coupled model previously defined and afterwards to an industrial application. We present the input design framework following the notation from Chapter 3: to each input $u^{[n]}$ is associated a collection of objects depending on the index n, indicated by the superscript $[n]$: for example $\mathrm{y}_{\boldsymbol{\mu}}^{[n],h,k} \in \mathbb{R}^{n+1}$ and $\mathrm{q}_{\boldsymbol{\mu}}^{[n],h,k} \in \mathbb{R}^n$ are the state variable FO vectors (marked by the superscript h) at OID iteration n and at time step $k = 1, \dots, K$. The FO data discrepancy corresponding only to OID iteration n is given as

$$\tilde{J}^{[n],h}(\boldsymbol{\mu}) := \sum_{k=0}^{K} \alpha_k \| q_{\boldsymbol{\mu}}^{[n],h,k} - w^{[n],h,k} \|_H^2,$$

and as a cost function for the parameter optimization problem we should "sum" all information and optimize the discrepancy between observable and data relative to each input, namely

$$\hat{J}^h(\boldsymbol{\mu}) := 1 + \frac{\alpha_J}{2N_\mathsf{e}} \sum_{n=1}^{N_\mathsf{e}} \hat{J}^{[n],h}(\boldsymbol{\mu}) + \frac{\beta_J}{2} \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2^2, \tag{4.1}$$

where N_e is the number of inputs. We then define the gradient of $\hat{J}^{[n],h}(\boldsymbol{\mu})$ as

$$\nabla \hat{J}^{[n],h}(\boldsymbol{\mu}) = \begin{pmatrix} \displaystyle\sum_{k=0}^{K} \alpha_k \left\langle \mathrm{y}_{\boldsymbol{\mu}}^{[n],h,k}, \mathrm{p}_{\mathrm{y}\boldsymbol{\mu}}^{[n],h,k} \right\rangle_{\mathrm{A_y}} \\ -\displaystyle\sum_{k=0}^{K} \alpha_k \left\langle \mathrm{f}_{\mathrm{y}}^h(\mathrm{y}_{\boldsymbol{\mu}}^{[n],h,k}, \mathrm{q}_{\boldsymbol{\mu}}^{[n],h,k}), \mathrm{p}_{\mathrm{y}\boldsymbol{\mu}}^{[n],h,k} \right\rangle_{\mathrm{M_y}} \\ \displaystyle\sum_{k=0}^{K} \alpha_k \left\langle \mathrm{q}_{\boldsymbol{\mu}}^{[n],h,k}, \mathrm{p}_{\mathrm{q}\boldsymbol{\mu}}^{[n],h,k} \right\rangle_{\mathrm{A_q}} \\ \displaystyle\sum_{k=0}^{K} \alpha_k \left\langle \mathrm{f}_{\mathrm{q}}^h(\mathrm{y}_{\boldsymbol{\mu}}^{[n],h,k}, \mathrm{q}_{\boldsymbol{\mu}}^{[n],h,k}), \mathrm{p}_{\mathrm{q}\boldsymbol{\mu}}^{[n],h,k} \right\rangle_{\mathrm{M_q}} \end{pmatrix},$$

where the FE matrices $\mathrm{A_y}, \mathrm{M_y}, \mathrm{A_q}, \mathrm{M_q}$ are given in Section 3.2, and finally

$$\nabla \hat{J}^h(\boldsymbol{\mu}) = \frac{\alpha_J}{N_\mathsf{e}} \sum_{n=1}^{N_\mathsf{e}} \nabla \hat{J}^{[n],h}(\boldsymbol{\mu}) + \beta_J(\bar{\boldsymbol{\mu}} - \hat{\boldsymbol{\mu}}).$$

This chapter is developed as follows. In Section 4.1 we will talk about measuring information and define the Fisher information matrix, or observability matrix, and its use in order to define the collected information. Then, in Section 4.2 we discuss the OID algorithm in details and in Section 4.3 we show numerical results on the coupled model defined in the previous chapter. In the last section, Section 4.4, we discuss the industrial application and show the results of the application of our OID algorithm to a real-world problem.

4.1 Measuring information

In this section we define the Fisher information matrix (FIM), a common statistical object used to quantify observability of a parameter in relation to a system, by observing the so-called *sensitivities*, namely the variations of the system outputs with respect to variations of the parameters. Using this matrix we are then able to approximate a measure of information that we will maximize.

4.1.1 Fisher's information matrix: motivation

Let us suppose that our data vector $\hat{\mathrm{w}} \in \mathbb{R}^m$ is a random variable in some arbitrary dimension $m > 0$, realization of the nonlinear model

$$\hat{\mathrm{w}} = \mathcal{F}(\boldsymbol{\mu}^*) + \nu, \tag{4.2}$$

where $\nu \sim \mathcal{N}(0, \sigma_{\mathsf{d}}^2 \mathbb{I}_m)$ is an m-dimensional Gaussian random variable, $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^m$ is a nonlinear function, and $\boldsymbol{\mu}^* \in \mathbb{R}^d$ is an unknown parameter vector (also called parameter for simplicity).

To quantify the information of the unknown parameter $\boldsymbol{\mu}^*$ carried by the random variable $\hat{\mathrm{w}}$, the Fisher information matrix (FIM) is often used, and it is defined as the variance of the score. The *score* is defined as the partial derivative with respect to $\boldsymbol{\mu}^*$ of the natural logarithm of the likelihood function p. Since $\hat{\mathrm{w}}|\boldsymbol{\mu}^* \sim \mathcal{N}(\mathcal{F}(\boldsymbol{\mu}^*), \sigma_{\mathsf{d}}^2 \mathbb{I}_m)$, we can write

$$p(\hat{\mathrm{w}}|\boldsymbol{\mu}^*) = ((2\pi)^m \sigma_{\mathsf{d}}^2)^{-1/2} \exp\left(-\frac{1}{2\sigma_{\mathsf{d}}^2} \|\hat{\mathrm{w}} - \mathcal{F}(\boldsymbol{\mu}^*)\|_2^2\right),$$

and

$$\log p(\hat{\mathrm{w}}|\boldsymbol{\mu}^*) = -\frac{1}{2} \log((2\pi)^m \sigma_{\mathsf{d}}^2) - \frac{1}{2\sigma_{\mathsf{d}}^2} \|\hat{\mathrm{w}} - \mathcal{F}(\boldsymbol{\mu}^*)\|_2^2.$$

Then, the score is

$$\left.\frac{\partial \log p(\hat{\mathrm{w}}|\boldsymbol{\mu})}{\partial \boldsymbol{\mu}}\right|_{\boldsymbol{\mu}^*} = \frac{1}{\sigma_{\mathsf{d}}^2} (\hat{\mathrm{w}} - \mathcal{F}(\boldsymbol{\mu}^*))^\top \mathrm{J}(\boldsymbol{\mu}^*),$$

where

$$\mathrm{J}(\boldsymbol{\mu}^*) = \left.\frac{\partial \mathcal{F}(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}}\right|_{\boldsymbol{\mu}^*} \in \mathbb{R}^{m \times d}.$$

Hence, the FIM in the parameter $\boldsymbol{\mu}^*$ is

$$\begin{aligned}
\mathrm{I}^{\mathrm{F}}_{\boldsymbol{\mu}^*} &:= \mathbb{E}_{\hat{\mathrm{w}}|\boldsymbol{\mu}^*} \left[\left(\left.\frac{\partial \log p(\hat{\mathrm{w}}|\boldsymbol{\mu})}{\partial \boldsymbol{\mu}}\right|_{\boldsymbol{\mu}^*} \right)^\top \left(\left.\frac{\partial \log p(\hat{\mathrm{w}}|\boldsymbol{\mu})}{\partial \boldsymbol{\mu}}\right|_{\boldsymbol{\mu}^*} \right) \right] \\
&= \frac{1}{\sigma_{\mathsf{d}}^4} \left.\frac{\partial \mathcal{F}(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}}\right|_{\boldsymbol{\mu}^*}^\top \mathbb{E}_{\hat{\mathrm{w}}|\boldsymbol{\mu}^*} \left[\|\hat{\mathrm{w}} - \mathcal{F}(\boldsymbol{\mu}^*)\|_2^2 \right] \left.\frac{\partial \mathcal{F}(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}}\right|_{\boldsymbol{\mu}^*} \\
&= \frac{1}{\sigma_{\mathsf{d}}^2} \mathrm{J}(\boldsymbol{\mu}^*)^\top \mathrm{J}(\boldsymbol{\mu}^*).
\end{aligned}$$

Let us consider the data $\hat{\mathrm{w}} := \mathrm{w}^{h,k}$ at an arbitrary time step k. Then, the nonlinear model is approximated by the FO observable (in this case the state q) evaluated at $\boldsymbol{\mu}^*$ at time k, i.e.

$$\mathcal{F}^k(\boldsymbol{\mu}^*) \approx \mathrm{q}^{h,k}_{\boldsymbol{\mu}^*} \in \mathbb{R}^N. \tag{4.3}$$

The derivative of the nonlinear map with respect to $\boldsymbol{\mu}$ evaluated in $\boldsymbol{\mu}^*$ is a vector in $\mathbb{R}^m$, and in this case its j-th component can be approximated using the j-th (namely, with respect to the j-th parameter) *sensitivity variable* $s^{\mathrm{q},h,k}_{j,\boldsymbol{\mu}}$ of the state q, defined and discretized in Section A.2.

Therefore, from (4.3), we deduce that the derivative of the nonlinear map with respect to $\boldsymbol{\mu}$ evaluated in $\boldsymbol{\mu}^*$ is approximated by the FO sensitivity, i.e. the j-th column of $\mathrm{J}^k(\boldsymbol{\mu}^*)$ is approximated as

$$\mathrm{J}^k(\boldsymbol{\mu}^*)_j := \left.\frac{\partial \mathcal{F}^k(\boldsymbol{\mu})}{\partial \mu_j}\right|_{\boldsymbol{\mu}^*} \approx \mathrm{s}^{\mathrm{q},h,k}_{j,\boldsymbol{\mu}^*} \in \mathbb{R}^N,$$

and hence we can write

$$\left(\mathrm{J}^k(\boldsymbol{\mu}^*)^\top \mathrm{J}^k(\boldsymbol{\mu}^*)\right)_{j_1,j_2} \approx \mathrm{s}^{\mathrm{q},h,k\top}_{j_1,\boldsymbol{\mu}^*}\, \mathrm{s}^{\mathrm{q},h,k}_{j_2,\boldsymbol{\mu}^*} \qquad \text{for } j_1, j_2 = 1, \dots, d.$$

In order to get a model of the form (4.2), we stack all observables into a single vector, obtaining

$$\mathcal{F}(\boldsymbol{\mu}^*) = \begin{pmatrix} \mathcal{F}^0(\boldsymbol{\mu}^*) \\ \vdots \\ \mathcal{F}^K(\boldsymbol{\mu}^*) \end{pmatrix} \approx \begin{pmatrix} \mathrm{q}^{h,0}_{\boldsymbol{\mu}^*} \\ \vdots \\ \mathrm{q}^{h,K}_{\boldsymbol{\mu}^*} \end{pmatrix}$$

and hence

$$\left(\mathrm{J}(\boldsymbol{\mu}^*)^\top \mathrm{J}(\boldsymbol{\mu}^*)\right)_{j_1,j_2} \approx \sum_{k=0}^{K} \mathrm{s}^{\mathrm{q},h,k\top}_{j_1,\boldsymbol{\mu}^*}\, \mathrm{s}^{\mathrm{q},h,k}_{j_2,\boldsymbol{\mu}^*} \qquad \text{for } j_1, j_2 = 1, \dots, d. \tag{4.4}$$

We now make two modifications to (4.4): as a first step, we apply a weighting on the sum, in order to resemble a time integral approximation:

$$\left(\mathrm{J}(\boldsymbol{\mu}^*)^\top \mathrm{J}(\boldsymbol{\mu}^*)\right)_{j_1,j_2} \approx \sum_{k=0}^{K} \alpha_k\, \mathrm{s}^{\mathrm{y},h,k\top}_{j_1,\boldsymbol{\mu}^*}\, \mathrm{s}^{\mathrm{q},h,k}_{j_2,\boldsymbol{\mu}^*} \qquad \text{for } j_1, j_2 = 1, \dots, d.$$

Secondly, we also apply a weighting for the product $\langle \mathrm{s}^{\mathrm{q},h,k}_{j_1,\boldsymbol{\mu}^*}, \mathrm{s}^{\mathrm{q},h,k}_{j_2,\boldsymbol{\mu}^*} \rangle_{\mathbb{R}^N}$, in order to obtain a discretized V-product, and consider instead $\langle \mathrm{s}^{\mathrm{q},h,k}_{j_1,\boldsymbol{\mu}^*}, \mathrm{s}^{\mathrm{q},h,k}_{j_2,\boldsymbol{\mu}^*} \rangle_{\mathrm{W_q}} = \mathrm{s}^{\mathrm{y},h,k\top}_{j_1,\boldsymbol{\mu}^*} \mathrm{W_q}\, \mathrm{s}^{\mathrm{q},h,k}_{j_2,\boldsymbol{\mu}^*}$.

Hence, we consider the following discretized information matrix I, seen as an approximation of an $L^2(0,T;V)$ inner product:

$$(\mathrm{I}_{\boldsymbol{\mu}^*})_{j_1,j_2} := \frac{1}{\sigma_{\mathsf{d}}^2} \sum_{k=0}^{K} \alpha_k\, \mathrm{s}^{\mathrm{q},h,k\top}_{j_1,\boldsymbol{\mu}^*} \mathrm{W_q}\, \mathrm{s}^{\mathrm{q},h,k}_{j_2,\boldsymbol{\mu}^*} \approx \frac{1}{\sigma_{\mathsf{d}}^2} \langle s^{\mathrm{q}}_{j_1,\boldsymbol{\mu}^*}, s^{\mathrm{q}}_{j_2,\boldsymbol{\mu}^*} \rangle_{L^2(0,T;V)}. \tag{4.5}$$

Again, we stress the purpose of this matrix. It quantifies the influence of the unknown parameter on the model observable. We observe that if this matrix has rank less than d, it

means that one or more parameter do not influence the output at all, hence making them *not observable*. Therefore, our purpose will be making this matrix as well-conditioned as possible.

Let us comment on another aspect mentioned, among others, in [65, Section 2.5]: we can interpret the FIM as the Hessian of the least-squares problem given by the cost function (2.11). Hence, improving the well-conditioning of the approximated information matrix can also help in making the cost function more convex around the unknown parameter – especially useful given the general non-convexity of the estimation problem.

4.1.2 Approximating the information function

Let us now make some observations: firstly, we have assumed that we do not have a-priori knowledge of the underlying true parameter $\boldsymbol{\mu}^*$, hence we cannot evaluate formula (4.5). For this reason the information matrix is usually evaluated for other parameters we have access to. This means both an arbitrary parameter, but also one chosen carefully along the optimization path. The idea is to add as much information as we can, hoping that this will help in the next parameter optimization. Hence, even adding information related to an other parameter might serve this purpose.

Another motivation driving this choice regards the convexity of the problem. Even if the parameter optimization falls into a local minimum far from the true parameter, improving the convexity of the least squares problem might push the parameter out of the basin of attraction. Hence, we will use the information matrix $\mathrm{I}_{\boldsymbol{\mu}} \in \mathbb{R}^{d\times d}$ dependent on the parameter $\boldsymbol{\mu}$:

$$(\mathrm{I}_{\boldsymbol{\mu}})_{j_1,j_2} := \frac{1}{\sigma_{\mathsf{d}}^2}\sum_{k=0}^{K} \alpha_k \, \mathrm{s}_{j_1,\boldsymbol{\mu}}^{\mathrm{q},h,k\top} \mathrm{W}_{\mathrm{q}} \, \mathrm{s}_{j_2,\boldsymbol{\mu}}^{\mathrm{q},h,k}, \qquad 1 \le j_1, j_2 \le d.$$

As a second observation we notice how the sensitivities, as well as the states, depend implicitly on the input u. Hence, we might want to find the u that gives us a relevant information matrix; in this fundamental idea lies the core of the input design algorithm. Let us write $\mathrm{I}_{\boldsymbol{\mu}}^{u}$ to explicit the dependence on the input u, and let $\mathrm{I}_{\boldsymbol{\mu}}^{[n]}$ be, in particular, the information matrix corresponding to the input $u^{[n]}$:

$$\left(\mathrm{I}_{\boldsymbol{\mu}}^{[n]}\right)_{j_1,j_2} := \frac{1}{\sigma_{\mathsf{d}}^2}\sum_{k=0}^{K} \alpha_k \, \mathrm{s}_{j_1,\boldsymbol{\mu}}^{\mathrm{q},[n],h,k\top} \mathrm{W}_{\mathrm{q}} \, \mathrm{s}_{j_2,\boldsymbol{\mu}}^{\mathrm{q},[n],h,k}, \qquad 1 \le j_1, j_2 \le d. \tag{4.6}$$

In order to define the input design optimization problem, we define a function that, given an information matrix, quantifies the uncertainty – in this case seen as a lack of information. We later refer to such object as the *uncertainty function*. In particular we take the D-criterion from the literature (see, e.g., [25, 3]), where the uncertainty function is given as

$$\phi(u) := -\log_{10}\left(\det\left(\mathrm{I}_{\boldsymbol{\mu}}^{u}\right)\right). \tag{4.7}$$

We can also define the uncertainty function by using the eigenvalues of I – especially convenient when the dimension d is high and evaluating the determinant gets expensive.

Indeed, if $\Lambda := \{\lambda_1, \ldots, \lambda_d\}$ is the set of eigenvalues of I, then

$$\phi(u) = \log_{10}\left(\prod_{\lambda_j \in \Lambda} \frac{1}{\lambda_j}\right)$$

and the D-criterion is equivalent to minimizing the generalized variance of the parameters (as discussed in [3]). Let us observe that we can also interpret this criterion as minimization of the volume of the confidence ellipsoid for the parameters.

Example 4.1. Using the problem and discretization as in Section 3.5, let us compare the uncertainty measure for the two different input functions $u_1(t) = -3\cdot\mathbf{1}_{[0,4/3)}(t)+3\cdot\mathbf{1}_{[4/3,2]}(t)$ and $u_2(t) = 0.5 \cdot \cos(10t) + 0.4 \cdot \sin(20t)$. We get

$$\phi(u_1) = 16.08613, \qquad \phi(u_2) = 28.57648.$$

This uncertainty quantities corresponds to the ideas discussed in this chapter: in the example of convergence to a local optimum, the information is lower. Hence minimizing the uncertainty function can indeed improve the parameter convergence. □

Remark 4.2. We can even assume the noise variance to be unknown. For example, we can evaluate the information matrix $\tilde{\mathrm{I}}_{\boldsymbol{\mu}} := \sigma_{\mathsf{d}}^2 \mathrm{I}_{\boldsymbol{\mu}}$ without knowing σ_{d}^2. As a matter of fact,

$$\left(\tilde{\mathrm{I}}_{\boldsymbol{\mu}}\right)_{j_1,j_2} = \sum_{k=0}^{K} \alpha_k \, \mathrm{s}_{j_1,\boldsymbol{\mu}}^{\mathrm{q},h,k\top} \mathrm{W}_{\mathrm{q}} \, \mathrm{s}_{j_2,\boldsymbol{\mu}}^{\mathrm{q},h,k} \qquad 1 \le j_1, j_2 \le d$$

for any parameter $\boldsymbol{\mu}$ and input u. Under these assumptions, its determinant does not get too big and problematic when σ_{d}^2 is very small and the parameter dimension grows. Observe that

$$\phi(u) = -\log_{10}\left(\det\left(\mathrm{I}_{\boldsymbol{\mu}}\right)\right) = -d\log_{10}\left(\frac{1}{\sigma_{\mathsf{d}}^2}\right) - \log_{10}\left(\det\left(\tilde{\mathrm{I}}_{\boldsymbol{\mu}}\right)\right).$$

Hence, optimizing $\tilde{\phi}(u) := \log_{10}\left(\det\left(\tilde{\mathrm{I}}_{\boldsymbol{\mu}}\right)\right)$ is equivalent to optimizing $\phi(u)$ and more convenient on the numerical point of view.

4.2 Optimal input design algorithm

The adaptive algorithm we use starts from arbitrary parameter $\boldsymbol{\mu}^0$. Then, for $n \ge 1$ (called OID iteration), the algorithm iterates on three phases:

- In the first phase we generate a new input $u^{[n]}$.
- In the second phase we read and store a data vector $\mathrm{w}^{[n]}$ coming from the control $u^{[n]}$ and the true parameter $\boldsymbol{\mu}^*$.
- The last phase is the parameter optimization/estimation, from which we obtain the parameter $\boldsymbol{\mu}^n$, approximation of $\boldsymbol{\mu}^*$.

There are different stop criteria. One possible way is to terminate the algorithm as soon as the last generated input is too close to one of the inputs already in our collection, i.e. using as a stop criterion

$$\min_{u\in\{u^{[1]},\dots u^{[n-1]}\}} \|u^{[n]} - u\| < \tau_{\mathsf{u}}. \tag{4.8}$$

However, a known limitation is that we have no guarantees of convergence and this can potentially generate an infinite number of inputs; when we use this criterion, we also set a maximum number of iterations as a safeguard.

An alternative termination is to force our algorithm to generate a fixed number of inputs and forcing these inputs to be sufficiently far from each other, by meaning of penalization of closeness. We will describe this "forced" generation of inputs in the section relative to the industrial application.

First phase

As discussed in Section 4.1, the measure of information to maximize – or better, the measure of uncertainty to minimize – is given by (4.7), evaluated in the available parameter $\boldsymbol{\mu}^{n-1}$. However, we want to maximize the total information, even considering the previous inputs and data. This can be done simply by summing the information matrices, thus at iteration n we define the uncertainty function

$$\phi(u) := -\log_{10}\left(\det\left(\sum_{m=1}^{n-1} \mathrm{I}^{[m]}_{\boldsymbol{\mu}^{n-1}} + \mathrm{I}^{u}_{\boldsymbol{\mu}^{n-1}}\right)\right). \tag{4.9}$$

To discretize this optimization problem we consider as input only step functions, to each input u is associated the vector of jumps $\mathrm{u} = (\mathrm{u}_1, \dots, \mathrm{u}_{n_\mathrm{u}}) \in \mathbb{R}^{n_\mathrm{u}}$. In particular, if $\tau := (T/n_\mathrm{u})$,

$$u(t) := \begin{cases} \mathrm{u}_1 & 0 \le t < \tau, \\ \vdots & \\ \mathrm{u}_{n_\mathrm{u}} & \tau(n_\mathrm{u}-1) \le t \le T. \end{cases}$$

The discretized optimization function is then defined as

$$\Phi(\mathrm{u}) := \phi(u) + \gamma\|\mathrm{u}\|^2_{\mathbb{R}^{n_\mathrm{u}}}, \tag{4.10}$$

where ϕ is defined in (4.9) and $\gamma := 10^{-4}$ accompanies a regularization term.

Hence, to find a new input we solve

$$\min_{\mathrm{u}\in\mathbb{R}^{n_\mathrm{u}}} \Phi(\mathrm{u}) \qquad \text{such that } u_{\mathsf{min}} \le \mathrm{u}_i \le u_{\mathsf{max}} \tag{4.11}$$

using the routine `fmin_l_bfgs_b` from `scipy` [66].

Second phase

As data we use a virtual noisy measurement of the q state variable corresponding to the hidden parameter $\boldsymbol{\mu}^*$ analogous to (4.2), i.e.

$$\mathrm{w}_i^{[n],h,k} = \mathrm{q}^{[n],h,k}_{\boldsymbol{\mu}^* i} + \varepsilon_i^k, \quad \text{where } \varepsilon_i^k \sim \mathcal{N}(0, \sigma^2_{\mathsf{d}}) \quad \text{for } k = 1, \dots, K, \quad i = 1, \dots, N. \tag{4.12}$$

Furthermore, as standard deviation of the noise we use $\sigma_{\mathsf{d}} := 10^{-3}$.

Third phase

For the parameter optimization we use the trust region algorithm described in Section 3.5. At each OID iteration, the new input might change the dynamics completely, hence we cannot rely on the reduced order model of the previous iteration.

At iteration n we need to well approximate the variables of iterations $1, \dots, n-1$ as well, since they appear in the cost function. Thus, it would seem meaningful to keep all the basis elements from previous iterations. However, we have decided to build a new basis organically, defining a different ROM at each OID iteration; rather than dividing the basis elements on the ground of relevance with the dynamics and "forgetting" the ones that do not fit, we have decided in favor of a total basis reset.

4.3 Numerical results

Once again, we work in the same setting and optimization parameters of Section 3.5. For parameters only relevant to the input design algorithm, we fix the input boundaries to $-3 \leq u(t) \leq 3$ for $t \in [0, T]$, $\tau_{\mathrm{u}} := 10^{-5}$ in (4.8), impose a maximum of 5 OID iterations, and $n_{\mathrm{u}} := 5$ for the discretization of the input function.

The inputs generated from this algorithm and numerical results are shown in Table 4.1. In this table we include a plot of the new input, time of the different phases, iterations, uncertainty functions ϕ before and after input optimization, cost function in the optimal parameter (as in (4.1)) and the norm of the difference between such parameter and the hidden parameter (as defined in Section 3.5). In particular, we notice that the uncertainty decreases monotonously, unlike the cost error. Rather, what decreases (almost) monotonously is the distance from the hidden parameter, as shown in Figure 4.1. So, our input collection seems to really improve the estimation of the parameter.

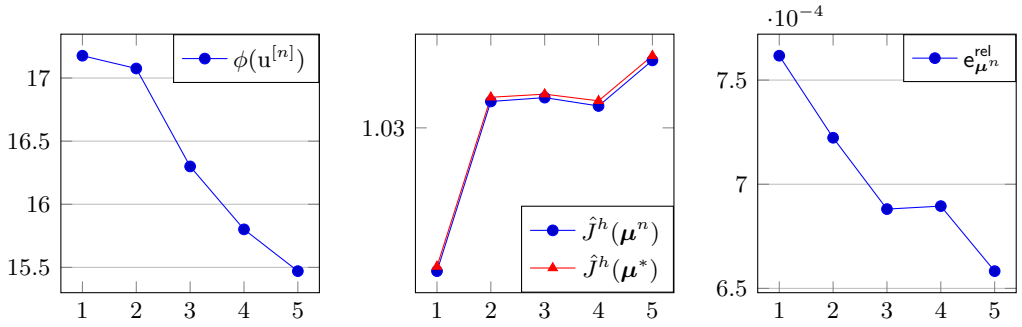


Figure 4.1: Left: uncertainty functions at each iteration. Center: plot of the FO cost evaluated in the optimal parameter and in the hidden parameter. Right: relative norm of the distance between the optimal parameter in each iteration and the true parameter. For all these plots, on the x-axis we find the OID iteration n.

Let us observe that in this particular case we could chose a maximum of 5 OID iterations because there exist some input that maximize the observability of all the parameters at the same time. However, we have already mentioned that it is also common that different

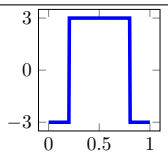
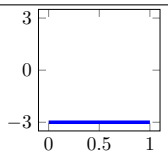
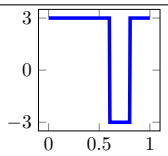
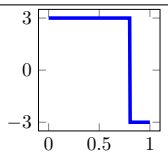
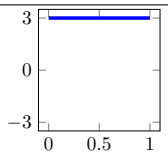
OID iteration	New input	Phase 1	Phase 2+3
$n = 1$		Time: 185 s Iterations: 4 $\phi(0)$: ∞ $\phi(\mathsf{u}_{\mathsf{opt}})$: 17.18	Time: 26 s Iterations: 24 FOM evaluations: 3 $\hat{J}^h(\boldsymbol{\mu}^n)$: 1.3318 $\mathsf{e}^{\mathsf{abs}}_{\boldsymbol{\mu}^n}$: 0.01375 $\mathsf{e}^{\mathsf{rel}}_{\boldsymbol{\mu}^n}$: 0.00187
$n = 2$		Time: 309 s Iterations: 6 $\phi(0)$: 18.84 $\phi(\mathsf{u}_{\mathsf{opt}})$: 17.08	Time: 34 s Iterations: 6 FOM evaluations: 3 $\hat{J}^h(\boldsymbol{\mu}^n)$: 1.33113 $\mathsf{e}^{\mathsf{abs}}_{\boldsymbol{\mu}^n}$: 0.00958 $\mathsf{e}^{\mathsf{rel}}_{\boldsymbol{\mu}^n}$: 0.00130
$n = 3$		Time: 354 s Iterations: 5 $\phi(0)$: 16.90 $\phi(\mathsf{u}_{\mathsf{opt}})$: 16.30	Time: 45 s Iterations: 2 FOM evaluations: 3 $\hat{J}^h(\boldsymbol{\mu}^n)$: 1.33115 $\mathsf{e}^{\mathsf{abs}}_{\boldsymbol{\mu}^n}$: 0.00958 $\mathsf{e}^{\mathsf{rel}}_{\boldsymbol{\mu}^n}$: 0.00130
$n = 4$		Time: 317 s Iterations: 6 $\phi(0)$: 16.12 $\phi(\mathsf{u}_{\mathsf{opt}})$: 15.80	Time: 70 s Iterations: 2 FOM evaluations: 3 $\hat{J}^h(\boldsymbol{\mu}^n)$: 1.33126 $\mathsf{e}^{\mathsf{abs}}_{\boldsymbol{\mu}^n}$: 0.00957 $\mathsf{e}^{\mathsf{rel}}_{\boldsymbol{\mu}^n}$: 0.00130
$n = 5$		Time: 470 s Iterations: 5 $\phi(0)$: 15.62 $\phi(\mathsf{u}_{\mathsf{opt}})$: 15.47	Time: 92 s Iterations: 2 FOM evaluations: 3 $\hat{J}^h(\boldsymbol{\mu}^n)$: 1.33149 $\mathsf{e}^{\mathsf{abs}}_{\boldsymbol{\mu}^n}$: 0.00955 $\mathsf{e}^{\mathsf{rel}}_{\boldsymbol{\mu}^n}$: 0.00130

Table 4.1: Results of the input design algorithm applied to the nonlinear coupled model.

parameters are more sensitive to different inputs, hence in general it is not possible to improve the estimation accuracy with a small number of inputs. This will be more evident in the industrial application discussed in the next section: the model parameters are 9, and it is not possible to find short inputs such that all of the model parameters are well-observable, and we should expect a higher number of inputs to maximize the observability of all of them.

In order to carefully analyze the results of the input design, we may argue that the inputs generated are only useful for the specific starting parameter $\boldsymbol{\mu}^0$. To show how the input design is effective for any starting point, we run a qualitative statistical analysis of the results of 100 parameter optimizations before and after the input design. For the "before" we use a low-observability input defined before, $u_\circ(t) = 0.5 \cdot \cos(10t) + 0.4 \cdot \sin(20t)$. In each of these optimization we randomize the starting parameter $\boldsymbol{\mu}^0$ and the virtual noise. We can see the results in Figure 4.2.

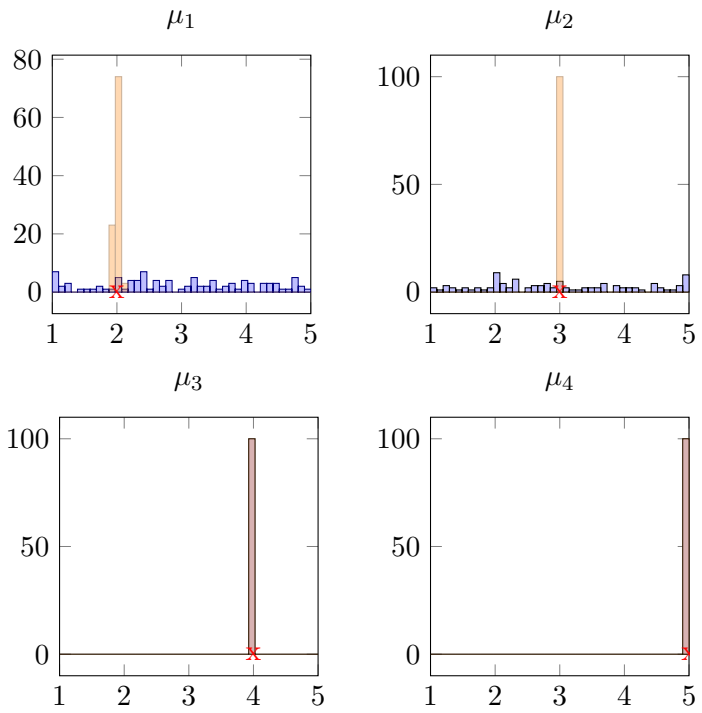


Figure 4.2: Qualitative statistical study for the input design of the nonlinear coupled model. In blue: the results of 100 parameter estimations without optimal input design. In orange: the results after. The red crosses mark the hidden parameters.

We observe that while using $u_\circ$ we can well-approximate the parameters μ_3, μ_4, for the other two parameters the situation is not that easy. This is probably due to the fact that the parameters μ_3, μ_4 appear in the state equation for the state q, namely the observable, and hence they can influence it in a stronger way. Using the collection of inputs generated by our algorithm, we can see how the parameter estimation closes in on the real values of μ_1 and μ_2. So, at least in a qualitative way, we can conclude that the input design does aid effectively the parameter estimation, improving the convergence.

4.4 An industrial application

Let us now turn our attention to an industrial application. Here we have worked with an industrial battery approximation model [57] in collaboration with our project partner Virtual Vehicle GmbH. We use this model in a black-box style: we can provide the program with an array of battery physical parameters, an array of time steps, an array of current inputs corresponding to the time steps, and the initial voltage of the battery. As output, we receive the array of voltages corresponding to the same time steps we have used. Details about the numerical model are available in [46].

As already mentioned before, the parameter estimation for such problems is usually extremely time consuming. One needs very long input currents and optimization time; for

example, in [57], in order to approximate 44 battery parameters an input current longer than 200,000 seconds (approximately 55 hours) is used, and even the optimization method lasts for several hours. On the other hand, we work with much fewer parameters and shorter time intervals: 9 parameters and 60 seconds. We cannot expect to optimize all parameters in such short time, hence the input design come into play. This will produce a collection of input variables, whose overall information will help the parameter optimization.

The parameters taken into consideration are described in Table 4.2, together with their chosen boundaries.

Parameter	Meaning	Lower bound	Upper bound
D_C	Cathode diffusion coefficient	0.0001	0.01
D_A	Anode diffusion coefficient	0.0001	0.01
ξ_A	Anode initial state of charge	0.007	0.2
R_I	Inner resistance	0.003	0.07
m_C	Cathode active mass	0.01	0.052
m_A	Anode active mass	0.006	0.034
$\tilde{k}_C$	Cathode reaction rate	-20	-1
$\tilde{k}_A$	Anode reaction rate	-23	10
U_0	Cathode 0th RK parameter	3	4

Table 4.2: Unscaled parameter names and bounds

Given the different orders of magnitude, we apply the following scaling.

$$\mu_1 := \log_{10}\left(\frac{D_C}{D_C^L}\right), \qquad \mu_2 := \log_{10}\left(\frac{D_A}{D_A^L}\right), \qquad \mu_3 := \frac{\xi_A}{(\xi_A^L + \xi_A^U)/2},$$
$$\mu_4 := \frac{R_I}{(R_I^L + R_I^U)/2}, \qquad \mu_5 := \frac{m_C}{(m_C^L + m_C^U)/2}, \qquad \mu_6 := \frac{m_A}{(m_A^L + m_A^U)/2},$$
$$\mu_7 := k_C - (k_C^L + k_A^U)/2, \qquad \mu_8 := k_A - k_C, \qquad \mu_9 := \frac{U_0}{(U_0^L + U_0^U)/2}.$$

Hence, the parameter bounds $\boldsymbol{\mu}^L$ and $\boldsymbol{\mu}^U$ are given in Table 4.3.

$\boldsymbol{\mu}^L$	$\boldsymbol{\mu}^U$
0	2
0	2
0.0676328502415459	1.93236714975845
0.0821917808219178	1.91780821917808
0.32258064516129	1.67741935483871
0.3	1.7
-9.5	9.5
-3	11
0.857142857142857	1.14285714285714

Table 4.3: Bounds for the unscaled and scaled parameters

Next, we will explain how we have adapted and used the input design algorithm in this

case: we discuss in particular the time discretization, the parameter optimization and the approximation of the uncertainty function.

The discretization

We fix the time step to $\delta = 0.1$, so that we have $K = 600$ time steps in a 60-seconds interval. Let us define the time array $\mathrm{t} := [t_0, \dots, t_K]$, where $t_k := k\delta$ for $k = 0, \dots, K$. Similarly to what done previously, we discretize all current profiles as step functions, so that all input information is compressed into the the *input array* (or, more simply, input): the array

$$\mathrm{u} := [u_1, \dots, u_{n_\mathrm{u}}, v_0] \in \mathbb{R}^{n_\mathrm{u}+1} \tag{4.13}$$

corresponds to the current function

$$i_\mathrm{u}(t) := \begin{cases} u_1 & 0 \le t < \tau, \\ u_2 & \tau \le t < 2\tau, \\ \vdots & \\ u_{n_\mathrm{u}} & (n_\mathrm{u}-1)\tau \le t \le n_\mathrm{u}\tau := T, \end{cases}$$

and the last component, v_0, is the initial voltage of the battery (which is, again, required). The current array $\mathrm{i_u} \in \mathbb{R}^K$ is then given by the discretization of i_u, evaluated on the fixed time array t. Finally, with $\mathrm{v}_{\boldsymbol{\mu},\mathrm{u}} \in \mathbb{R}^K$ we indicate the output voltage given the parameter $\boldsymbol{\mu}$ and input u (hence given current $\mathrm{i_u}$) evaluated on the time vector t.

Input design problem

In order to approximate the uncertainty function, we need to define the sensitivities, and consequently the information matrix. However, not having access to the state equations, we approximate sensitivities using finite differences: the i-th sensitivity $\mathrm{s}^i \in \mathbb{R}^K$ is evaluated as

$$\mathrm{s}^i_{\boldsymbol{\mu},\mathrm{u}} = \frac{\mathrm{v}_{\boldsymbol{\mu}+\nu\mathrm{e}_i,\mathrm{u}} - \mathrm{v}_{\boldsymbol{\mu},\mathrm{u}}}{\nu} \in \mathbb{R}^K,$$

where e_i is the i-th Euclidean basis element, and $\nu = 10^{-3}$.

The information matrix $\mathrm{I}^\mathrm{u}_{\boldsymbol{\mu}} \in \mathbb{R}^{9\times 9}$ dependent on parameter $\boldsymbol{\mu}$ and input array u is then defined using $\mathrm{s}^i = \mathrm{s}^i_{\boldsymbol{\mu},\mathrm{u}}$ as

$$\left(\mathrm{I}^\mathrm{u}_{\boldsymbol{\mu}}\right)_{i,j} := \sum_{k=0}^{K-1} \frac{\mathrm{s}^i_k \mathrm{s}^j_k + \mathrm{s}^i_{k+1}\mathrm{s}^j_{k+1}}{2}(t_{k+1} - t_k), \qquad 1 \le i, j \le 9, \tag{4.14}$$

which is an approximation of the product $\langle s^i, s^j \rangle_{L^2(0,T;\mathbb{R}^K)}$. We mention that this could be optimized by using other (and more accurate) methods to evaluate the partial derivatives, such as automatic differentiation (see, e.g., [55]).

As for the input boundaries u^L and u^U we choose the following:

$$\begin{aligned} -8.8 =: u^L_k \le u_k \le u^U_k := 8.8, &\qquad k = 1, \dots, n_\mathrm{u}, \\ 3.3 =: v^L_0 \le v_0 \le v^U_0 := 4.1. & \end{aligned}$$

As before, we optimize the discretized OID objective function $\Phi(\mathrm{u})$ constrained in $\mathrm{u}^L \leq \mathrm{u} \leq \mathrm{u}^U$ using the D-criterion. Wanting the algorithm to generate as much information as possible, we force it to produce 10 experiments in total. Since there is no information gain when the input generated is equal to one we already have, we modify the OID optimization function again, in order to penalize the distance from other inputs. The new optimization function is then

$$\min \hat{\Phi}(\mathrm{u}) \qquad \text{such that } \mathrm{u}^L \leq \mathrm{u} \leq \mathrm{u}^U,$$

where

$$\hat{\Phi}(\mathrm{u}) := \Phi(\mathrm{u}) + \sum_{\substack{\mathrm{u}^k \text{ previous} \\ \text{controls}}} \frac{1}{1 + 100 \cdot \|\mathrm{u} - \mathrm{u}^k\|_{L^\infty}}, \tag{4.15}$$

The penalizing function, as well as the scalar 100 have been chosen in an intuitive way in order to guarantee that we stop our algorithm provided our current control is sufficiently far from the already computed controls, but also not to modify the initial optimization problem too strongly.

Parameter Estimation

Assuming now that u (hence the current i) is fixed, we want to optimize the parameter vector $\boldsymbol{\mu}$ by minimizing the discrepancy with some data. Let us assume that the data is given by the voltage evaluated on some unknown parameter $\boldsymbol{\mu}^*$ without any measurement error or noise, i.e. $\mathrm{w_u} := \mathrm{v}_{\boldsymbol{\mu}^*,\mathrm{u}} \in \mathbb{R}^K$. For the numerical optimization we use a least squares method (from `scipy.optimize`, see [9]) to fit $\mathrm{v}_{\boldsymbol{\mu},\mathrm{u}}$ to $\mathrm{w_u}$. In particular, we fit the relative error

$$\mathrm{e_u}(\boldsymbol{\mu}) := (\mathrm{v}_{\boldsymbol{\mu},\mathrm{u}} - \mathrm{w_u})/\mathrm{w_u}$$

(where the division is intended component-wise) to 0. In particular, we solve

$$\min_{\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}} J(\boldsymbol{\mu}) := \frac{1}{2} \left\| \frac{\mathrm{v}_{\boldsymbol{\mu},\mathrm{u}} - \mathrm{w_u}}{\mathrm{w_u}} \right\|_2^2,$$

where $\mathcal{P}_{\mathsf{ad}} \subset \mathbb{R}^d$ is the closed, convex set of admissible parameters – in our case the box-constrained set defined by $\boldsymbol{\mu}^L \leq \boldsymbol{\mu} \leq \boldsymbol{\mu}^U$ in Table 4.3.

In the framework of multiple data, we can simply merge information by stacking them into a single vector. In particular, given $\mathrm{u}^{[1]}, \ldots, \mathrm{u}^{[n]}$ control vectors generated by the algorithm and $\mathrm{w}^{[1]}, \ldots, \mathrm{w}^{[n]}$ data vectors (with $\mathrm{w}^{[n]} := \mathrm{w}_{\mathrm{u}^{[n]}} = \mathrm{v}_{\boldsymbol{\mu}^*,\mathrm{u}^{[n]}}$), then we want to fit $\mathrm{v}_{\boldsymbol{\mu},\mathrm{u}^{[1]}}$ to $\mathrm{w}^{[1]}$ and so on. We concatenate all vectors of relative error into $[\mathrm{e}_{\mathrm{u}^{[1]}}(\boldsymbol{\mu}), \ldots, \mathrm{e}_{\mathrm{u}^{[n]}}(\boldsymbol{\mu})] \in \mathbb{R}^{nK}$ and we fit it to zero (in dimension nK). The parameter generated at OID iteration n is then the optimizer $\boldsymbol{\mu}^n$. Lastly, in order to optimize computing time, we evaluate voltage vectors in a parallel way for all different current inputs.

4.4.1 Numerical results

We have applied the input design algorithm with piecewise constant inputs consisting of 24 jumps, therefore each continuous sub-interval is 2.5 seconds long and the input dimension

is 25. We furthermore fix the maximum number of inputs to 10 and decide to speed up the first input generation by fixing it arbitrarily as

$$\mathrm{u}^{[1]} := [-1, +1, -1, +1, \ldots, -1, +1, 3.7] \in \mathbb{R}^{25}. \tag{4.16}$$

This corresponds to a rather "jumpy" – or *bang-bang* – input applied to a roughly mid-charge battery. We have chosen to do so in order to evaluate a parameter with an uninformative input and effectively see that such input estimates the parameter too poorly; the effect that an uninformative input has on the parameter estimation is shown later in Figure 4.5.

The algorithm takes around 4042 seconds to execute 9 iterations. In each iteration, the input design takes between 118 seconds and 1082 seconds, with an average of 413 seconds per iteration, while the parameter estimation takes between 134 seconds (with longer time especially at the first iterations) and 1.33 seconds (especially in the last iterations, where the parameter is already well approximated and it does not find better points), with an average of 34 seconds per iteration.

Various results of the 9 iterations of the input design algorithm, such as the variation of uncertainty, cost function and distance from the hidden parameter (analogously to Figure 4.1), are plotted in Figure 4.3, and in Figure 4.4 the generated inputs are shown. Although some of the inputs look very similar, the penalization function introduced in (4.15) guarantees that we do not use the exactly same input array.

The uncertainty does not decrease monotonously; the OID algorithm take some preliminary steps before converging to the true parameter, in particular reaching the neighborhood of $\boldsymbol{\mu}^*$ in the fourth iteration. We can interpret these steps as somehow a preparation stage, in which the algorithm tries to convexify the cost function in order to fall into the basin of attraction of the hidden parameter more easily in the next iterations.

Similarly to what we did for the nonlinear coupled model, we want to test the accuracy of the input design for randomly chosen initial parameters. We will therefore plot the result of 100 parameter optimizations before the input design, namely using the arbitrarily

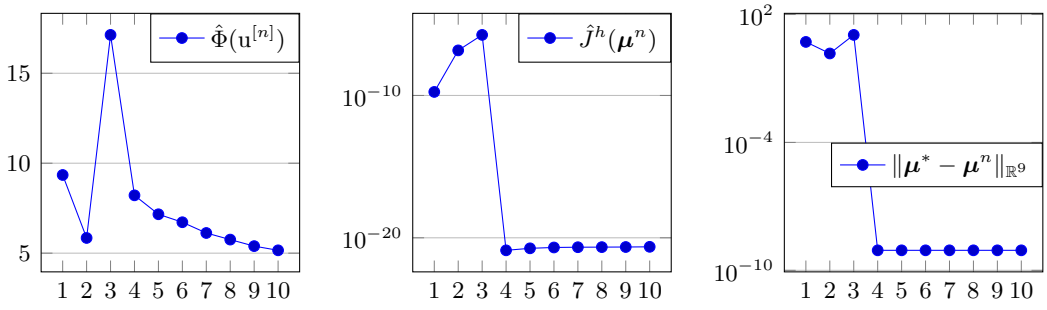


Figure 4.3: Left: uncertainty functions at each iteration. Center: plot of the FO cost evaluated in the optimal parameter. Right: Euclidean norm of the error between the optimal parameter in each iteration and the true parameter. For all these plots, on the x-axis we find the OID iteration n.

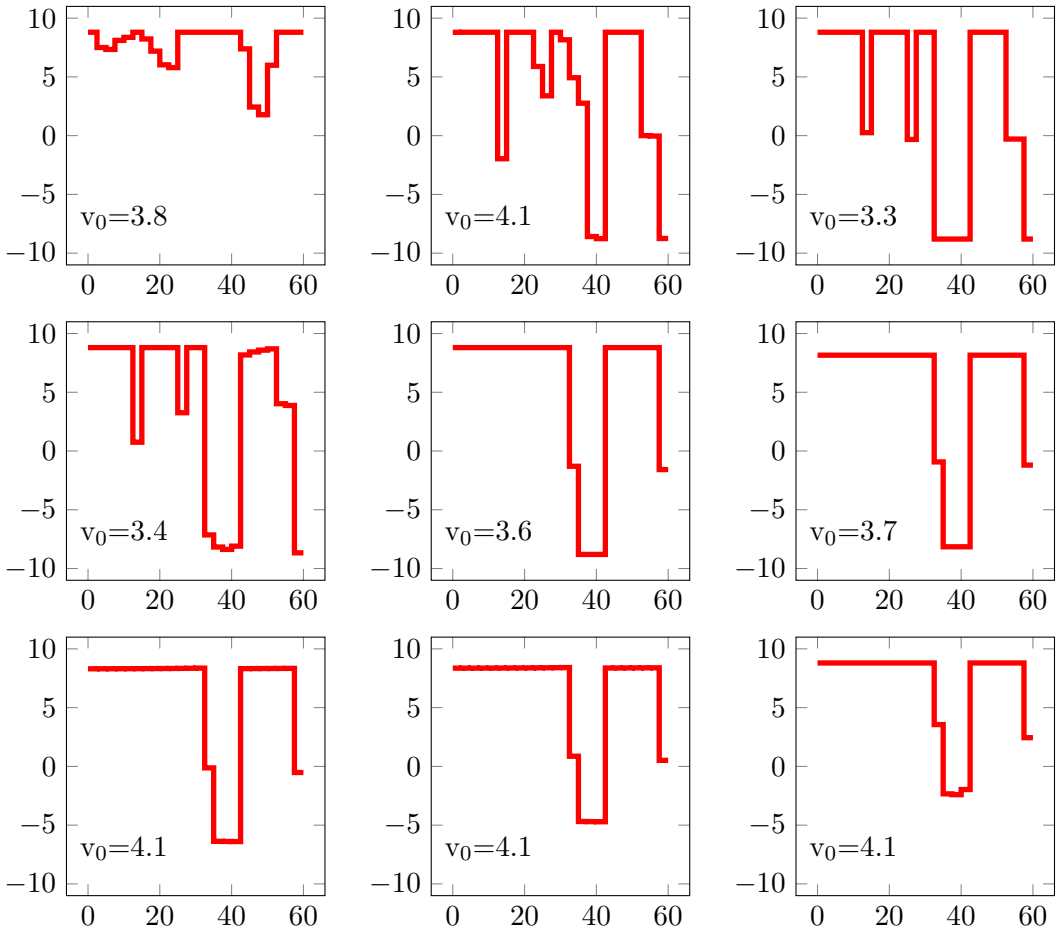


Figure 4.4: Input functions generated by the OID algorithm.

chosen input defined in (4.16), and after it, using all ten of them. The results are plotted in Figure 4.5. As in the case of the nonlinear coupled model, the input design improves the parameter estimation significantly. While a non-informative input the parameter estimation fails almost every time (shown in blue bars), an optimal collection of inputs results into the accumulation of the optimization results around the hidden parameters.

Another positive outcome is the optimization time: on average, an optimal parameter is found in 85 seconds. This is of course a rather fast convergence time, but in a very restricted scenario, such as the low dimension of the parameter and the noiseless framework. Nonetheless, although this convergence is not certain (i.e. it does not happen always), we can still see a degree of bias coming out of these plots – namely, we can deduce the hidden parameter in a easier way. Thus, we can see the point and the effectiveness of such input design. However, this framework might find some unexpected counter-effects in its applicability, how we discuss in the next session.

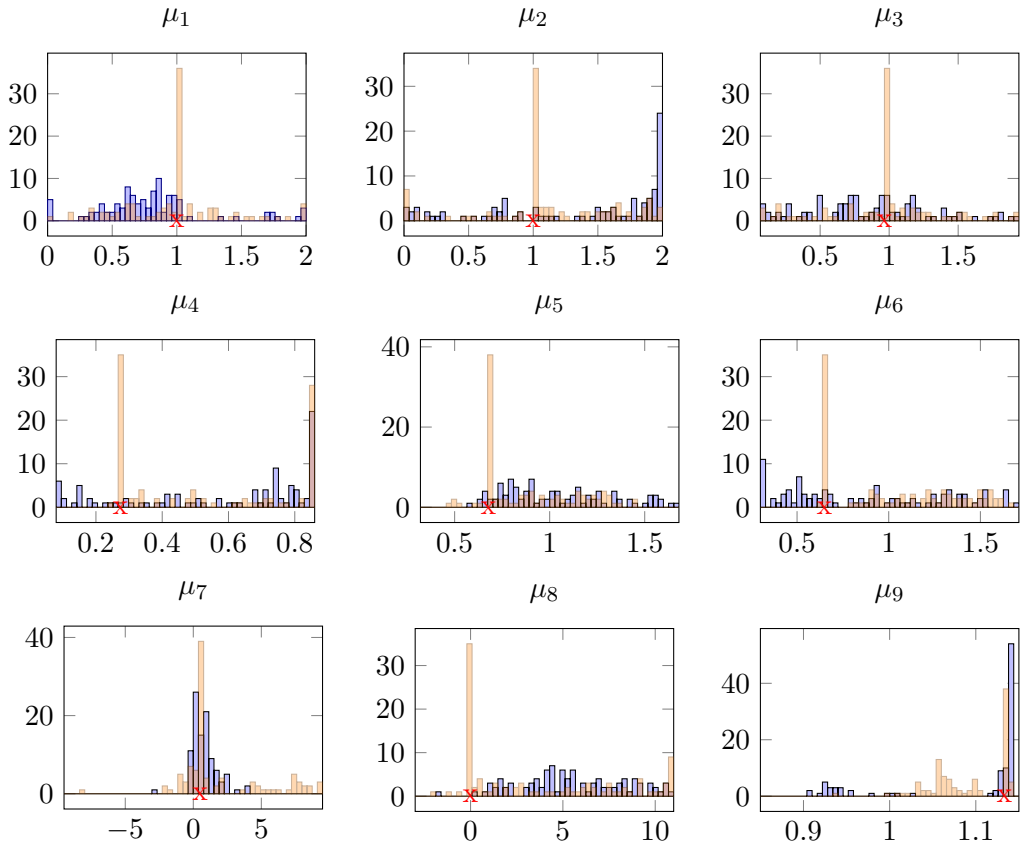


Figure 4.5: Qualitative statistical study for the input design of the battery model. In blue: the results of 100 parameter estimations using only the input given by $u^{[1]}$. In orange: the results using the optimal collection of inputs. The red crosses mark the hidden parameters.

4.4.2 A different design

Generating ten different short inputs sounds like a good idea if we want to look at the behavior of a battery cell in varied states of charge (SOCs): starting from disparate initial voltages serves exactly this purpose. But there might be a problem in the actual use of these different current profiles, since we cannot evaluate all inputs at the same time in a parallel way (as we do in the numerical model). On the contrary, before and after every input we need a phase of preparation to bring the battery cell at the right initial voltage, and a long phase of rest to let the internal electrochemical dynamics to stabilize. For this reason, even if the ten inputs together sum up to 600 seconds, actually setting up and running the experiment can take more than 60,000 seconds, namely 100 times more.

Hence, we have developed a modification of the OID algorithm and the design structure that would avoid this problem. The idea is to generate a sequence of new input intervals to concatenate one after the other, rather then to use them as totally different current profiles. As a consequence, we should not choose the initial voltage of each interval, because this will be given by the ending voltage of the previous interval. We can only choose the first

initial voltage, namely the starting voltage of the first interval; this is chosen as 3.9 V, equivalent to about 75% state of charge.

We also realize that we need a resting interval in order to capture the long-term dynamics. Hence, our input is translated into six jumps of a 120 seconds high-frequency time interval followed by a 600 seconds resting phase. As the time discretization keeps the step $\delta = 0.1$, we have 7200 time steps for each concatenated interval. An example of the first input with arbitrary jumps is plotted in Figure 4.6. This time we are not fixing arbitrarily the first input $\mathrm{u}^{[1]} = (u_1^{[1]}, \dots, u_6^{[1]})$, but it is found in the first step of the OID algorithm.

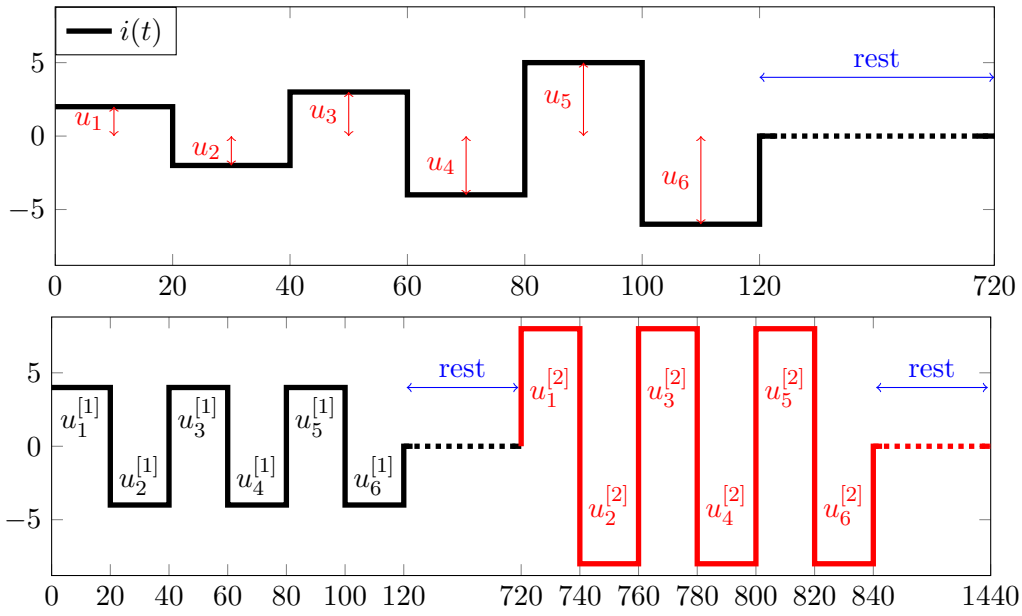


Figure 4.6: Visual explanation of the concatenated-inputs setting. Above: Example of the first input, showing the jumps and rest phase. Below: concatenation of the first two inputs $\mathrm{u}^{[1]}$ and $\mathrm{u}^{[2]}$ (in this plot chosen arbitrarily).

In the following iterations, we find input array $\mathrm{u}^{[n]}$ by optimizing the uncertainty of the full-concatenated current input, i.e. we add the new input to the old one and hence consider only one longer input. See Figure 4.6 for a visual explanation.

In this setting, we generate 9 inputs: therefore, concatenating the respective jumps and rest phases together we reach an "optimal" input lasting 6480 seconds, i.e. 108 minutes. This optimal input is plotted in Figure 4.7 together with the corresponding data $\mathrm{w_u} = \mathrm{v}_{\boldsymbol{\mu}^*,\mathrm{u}}$, i.e. the voltage function produced by the optimal input and the hidden parameter $\boldsymbol{\mu}^*$. The overall design takes 3458 seconds: the input design phases last between 100 and 833 seconds per iteration, with an average of 282 seconds, while the time for the parameter optimization varies between 9 and 486 seconds, with an average of 102 seconds per iteration. In comparison with the previous design (the "collection of inputs"), the input design phases is faster while, on the other hand, the parameter estimation takes longer. We can argue that this is expected, since the amount of data to optimize rises at each iteration and makes the least squares method more and more expensive to solve. Further information for each

OID iteration are plotted in Figure 4.8.

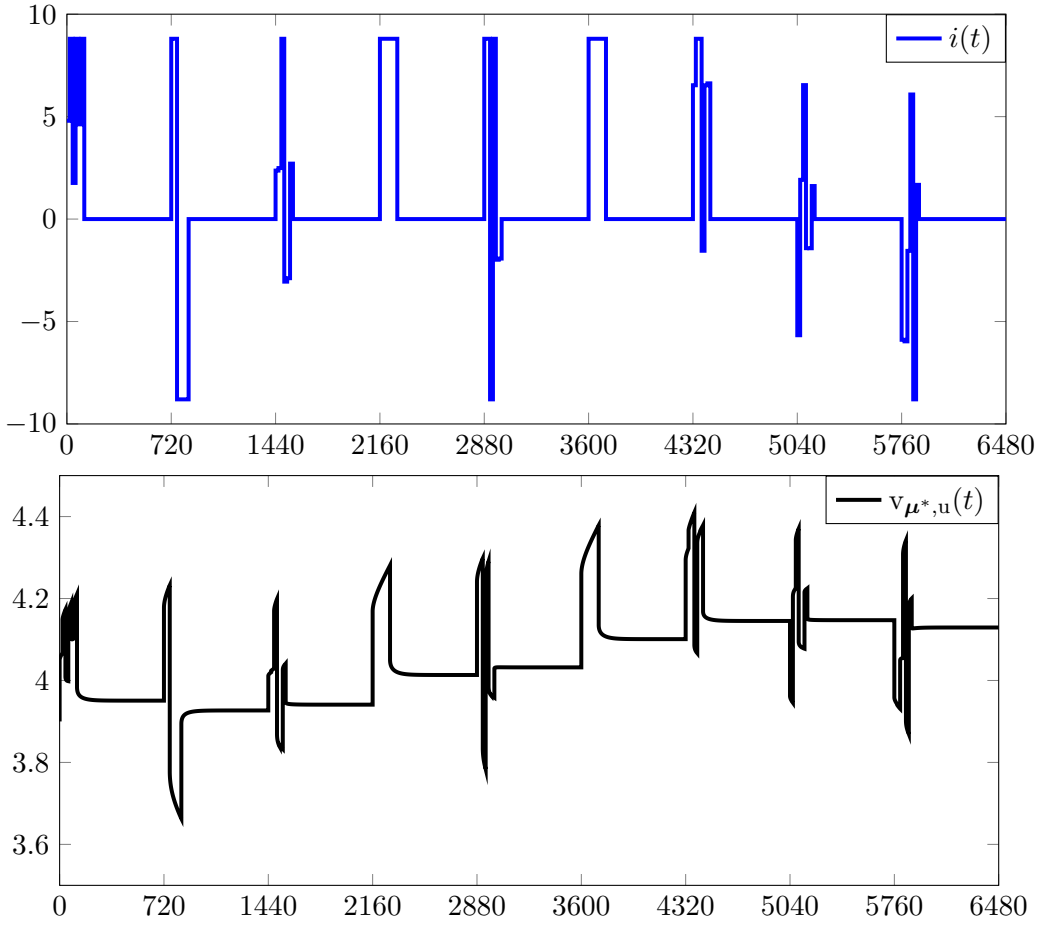


Figure 4.7: Optimal input generated by the algorithm in the concatenated setting.

As done previously, we plot the result of 100 parameter optimizations in order to show the estimation error for randomized initial parameters. This is shown in Figure 4.9, without showing a contrast before/after, but only the results corresponding to the optimal input. We observe an even better consistence and accuracy in the convergence to the true parameter; and considering the fact that we have, in fact, reduced the overall experiment time by a factor of 4, we regard this as a great improvement of the input design presented in the previous section.

4.4.3 Real battery data

While we have evaluated these optimal inputs using virtual data $w_u = v_{\mu^*,u}$ in the algorithm, we want to check how beneficial they are in a real-world parameter estimation problem. To this end, with the support of our partner Virtual Vehicle GmbH, we have run experiments on a real battery cell and collected the measurement data. We want to use these real-world data to measure the effect of the input design in a similar way to what we did for the nonlinear coupled model. Let us mention that clearly we do not randomize

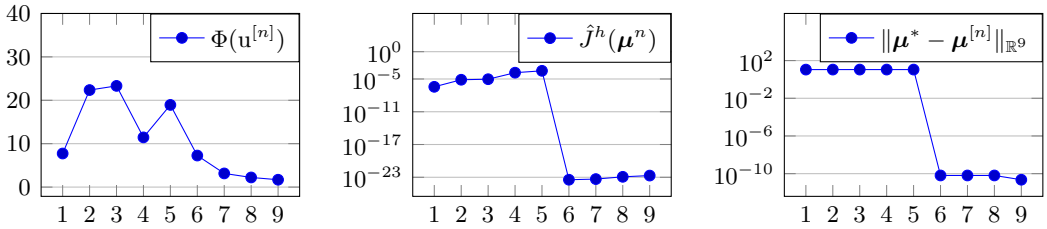


Figure 4.8: Information corresponding to each OID iteration n. Left: uncertainty functions. Center: FO cost. Right: Euclidean norm of the error.

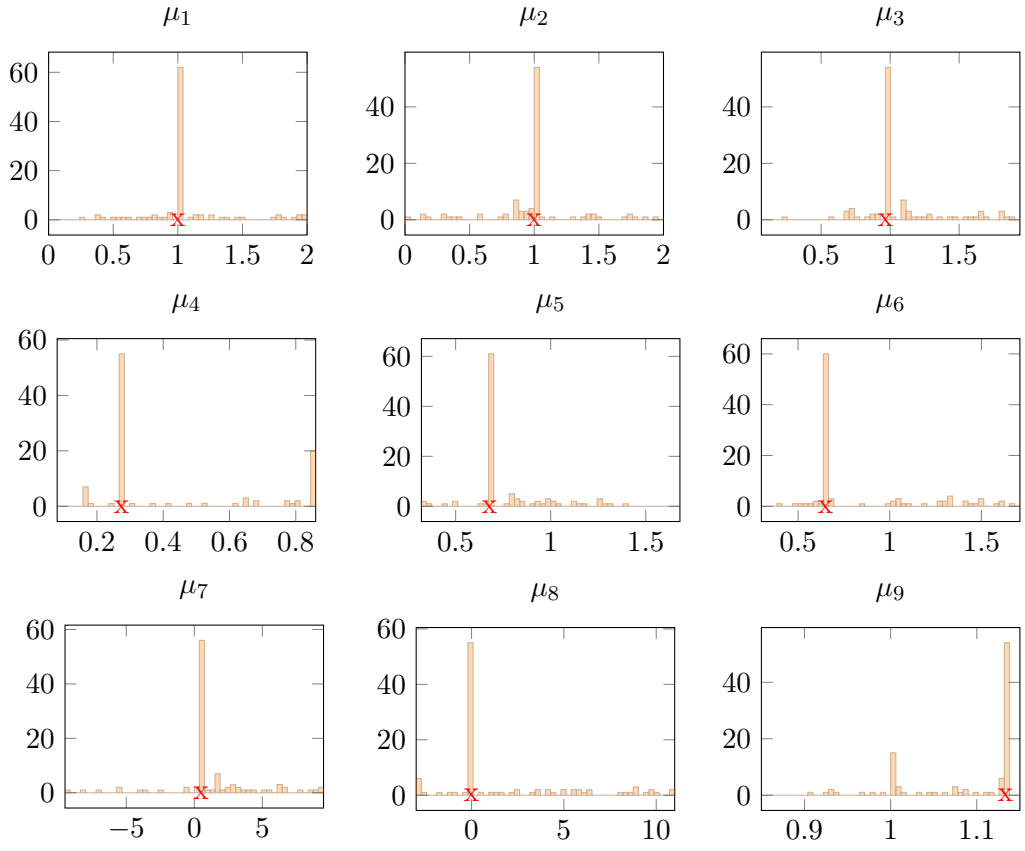


Figure 4.9: Qualitative statistical study for the input design in the concatenated setting. In orange: the results using the concatenated optimal input. The red crosses mark the true underlying parameters.

data since it would be expensive to collect a lot of data from real battery cells; instead, we only randomize the starting parameters.

This data includes also preparation steps at the beginning and the resting phase at the beginning, and depending on the test we will use information coming from the extra phases or not. Furthermore, the time stepping was chosen depending on the situation, varying between a minimum of 0.0002 (used in high-frequency dynamics) and a maximum of 1.0

(used in the resting phases). We consider four different situations, depending on two different factors: whether we use the collection of inputs derived in Section 4.4.1 or the concatenated input derived in Section 4.4.2, and whether we consider the additional data provided by the preparation and resting phases, or not. These choices are schematized in Table 4.4. Furthermore, additional plots of the inputs and data used are in Section A.3.

	Cut data: only impulses	Full data: impulses + rest phases
Collection of inputs	Test 1	Test 2
Concatenated input	Test 3	Test 4

Table 4.4: Considered tests for the analysis of real data.

	TESTS			
	1	2	3	4
Experiment time (s)	600	63,362	6,480	14,199
Data points	20,350	83,160	216,002	223,727
Optimization time (s)	800	154	550	450

Table 4.5: First line: total input time t_{f}; second line: number of data points; third line: average time for the parameter optimization.

In each of these situations we run 100 parameter optimizations in order to infer the battery cell parameters; if we notice a trend or an accumulation point, we select a *proposed optimal parameter vector*. Since we do not know this true parameter, we will lastly do some qualitative error analysis on the model output of the proposed optimal parameters given in each test.

In particular, we will propose three parameters $\boldsymbol{\mu}^{\mathsf{opt},2}$, $\boldsymbol{\mu}^{\mathsf{opt},3}$ and $\boldsymbol{\mu}^{\mathsf{opt},4}$ inferred from test 2, 3 and 4, respectively. For each of these parameters we plot the output voltage produced by the numerical model $v_{\boldsymbol{\mu},\mathrm{u}}(t)$, as well as the relative error given by the discrepancy with the real battery data $w(t)$, namely

$$e^{\mathsf{rel},[n]}_{\boldsymbol{\mu}^{\mathsf{opt},k}}(t) := (w^{[n]}(t) - v_{\boldsymbol{\mu}^{\mathsf{opt},k},\mathrm{u}^{[n]}}(t))/w^{[n]}(t) \qquad \text{for } n = 2,3,4 \text{ and } k = 2,3,4.$$

The results are shown in the upcoming figures.

TEST 1

With 10 intervals of 60 seconds and 2035 time steps each, this test amounts to a total of 20,350 data points. The parameter optimization takes in average 800 seconds, with observed minimums around 80 and maximums over 2800. The histograms of the results of the estimation are plotted in Figure 4.10. Although we can guess some of the true parameters, we have to admit that these results do not look as good as in Figure 4.5, even though we are using essentially the same input collection. In this case we can unfortunately conclude that the input design is not successful, as we derive no information about where the hidden parameters might be.

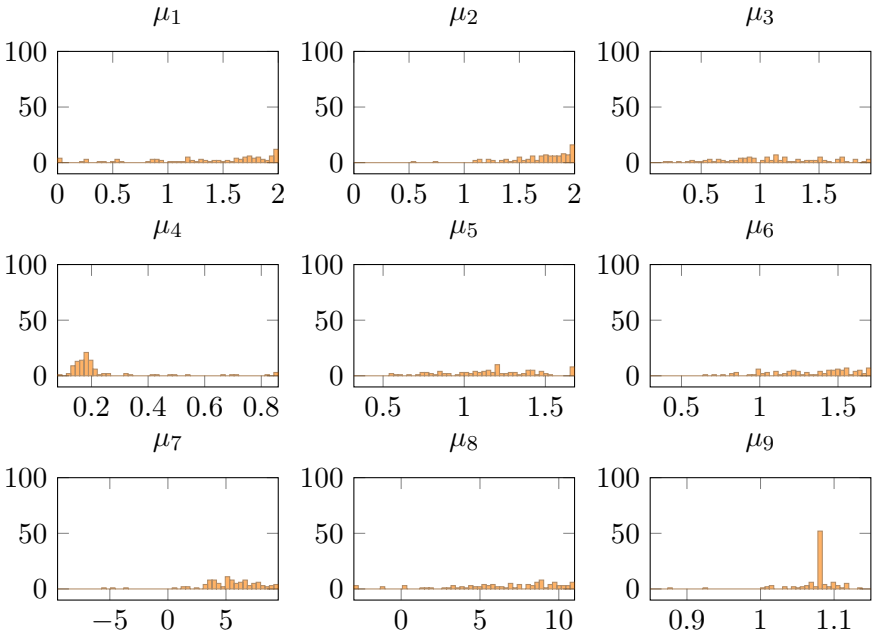


Figure 4.10: Results for Test 1. Bar plots of qualitative analysis of 100 parameter estimations starting from randomized initial guess.

TEST 2

Since the production of the data necessary for TEST 1 produces, by default, the data used in TEST 2, we might think that test is somehow more informative and hence the estimation of the true parameter is more successful. As showed in Figure 4.11, this holds true. Although the experiment duration and data amount are significantly higher, i.e. 63,362 seconds for 83,160 points, the parameter optimization is in average much faster, with 154 seconds per iteration. This might be a symptom of a generally more convex objective function and seemingly a success for the optimal input design. Let us define $\boldsymbol{\mu}^{\mathsf{opt},2}$ the most common optimizer, indicated with the blue mark.

The analysis of the model error given by the proposed parameter confirms the quality of such parameter: for the inputs $\mathrm{u}^{[2]}$ and $\mathrm{u}^{[3]}$ the relative error oscillates in the order of 10^{-3}, while for the input $\mathrm{u}^{[4]}$ the order of the relative error is 10^{-2}. The numerical model tuned with such parameter is therefore an accurate simulator of the battery cell taken into consideration.

TEST 3

Trying to reduce the experiment duration, we end up with a time period of 6480 seconds (108 minutes) and 216,002 time steps. The parameter optimization takes between 450 and 650 seconds, and in Figure 4.12 we can observe that the optimization results gravitate around different local minima. With $\boldsymbol{\mu}^{\mathsf{opt},3}$ we have indicated the most common optimizer, shown in Figure 4.12 with the blue mark.

This parameter produces accurate results for the data $w^{[3]}$, but accumulates a too high

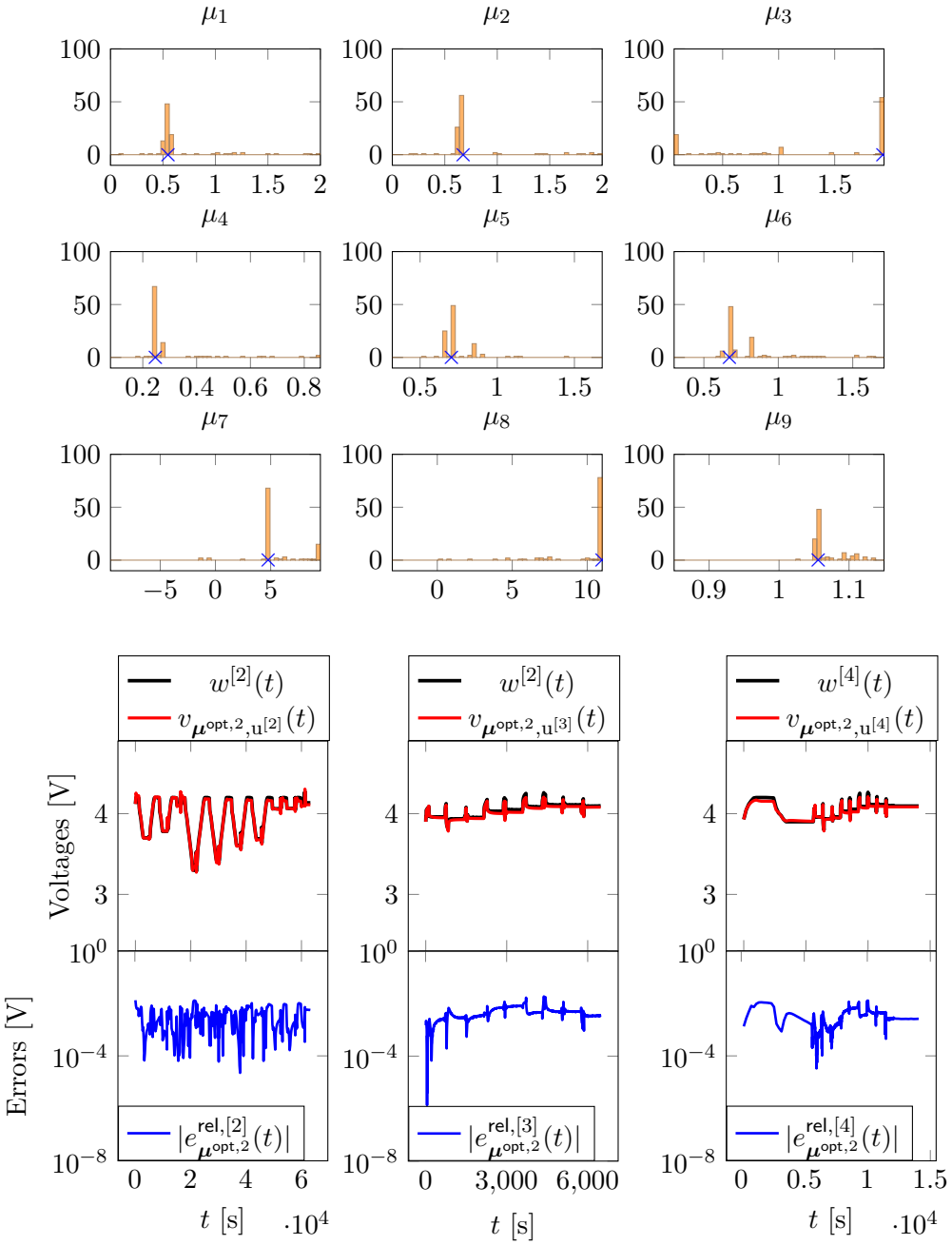


Figure 4.11: Results for Test 2. Top: bar plots of qualitative analysis of 100 parameter estimations starting from randomized initial guess; the blue cross finds the proposed parameter $\boldsymbol{\mu}^{\mathsf{opt},2}$; bottom: data, output and error generated by the proposed parameter.

error in the higher voltage changes occurring in data observations $w^{[2]}$ and $w^{[4]}$, hence not capturing very well the full range of battery cell dynamics.

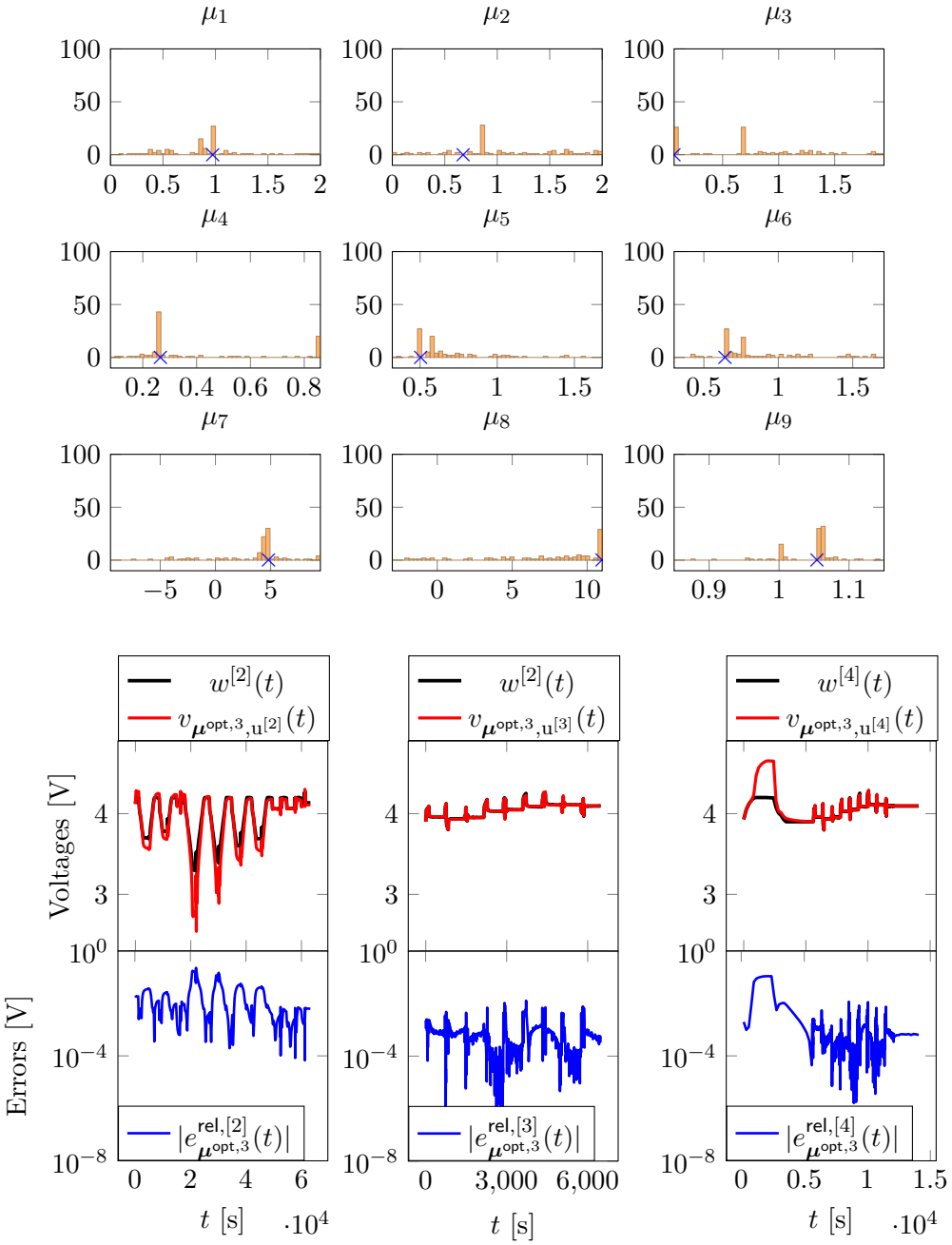


Figure 4.12: Results for Test 3. Top: bar plots of qualitative analysis of 100 parameter estimations starting from randomized initial guess; the blue cross finds the proposed parameter $\boldsymbol{\mu}^{\mathsf{opt},3}$; bottom: data, output and error generated by the proposed parameter.

TEST 4

Including the experiment and data set of TEST 3, this test lasts for 14,199 seconds and collects 223,727 time steps. Similarly to TEST 3, the optimization algorithm takes between

300 and 600 seconds to converge. The results look definitely more certain and focus around a single parameter vector $\boldsymbol{\mu}^{\mathsf{opt},4}$, marked in Figure 4.13 in blue.

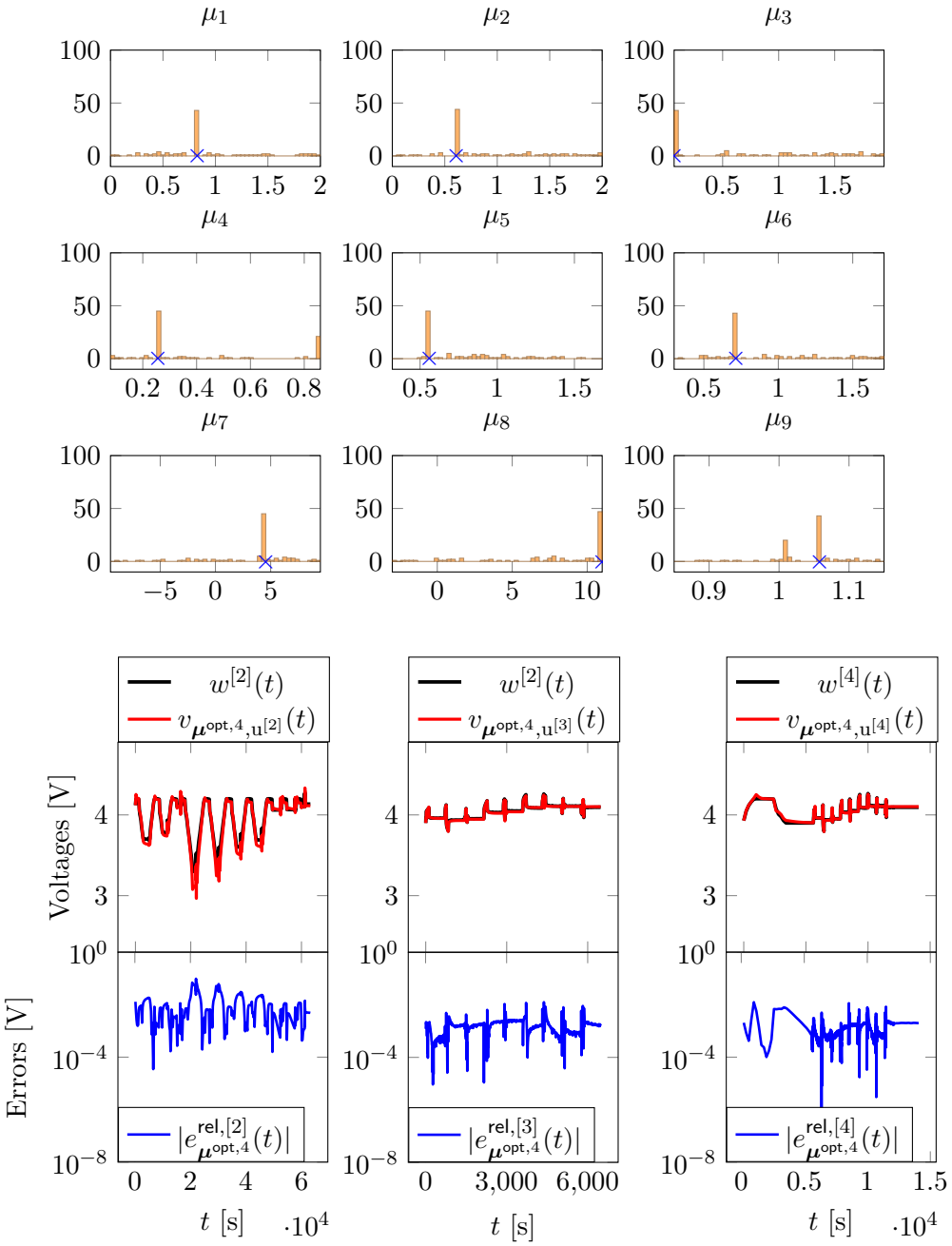


Figure 4.13: Results for Test 4. Top: bar plots of qualitative analysis of 100 parameter estimations starting from randomized initial guess; the blue cross finds the proposed parameter $\boldsymbol{\mu}^{\mathsf{opt},4}$; bottom: data, output and error generated by the proposed parameter.

Comments on the accuracy of the optimizers

At a first glance, it seems that we cannot find a parameter to perfectly replicate the data. This might depend on different factors; the most obvious consideration comes from the fact that we are running simulations with a simplified numerical model with a reduced number of active, unknown parameters, while the data is measured from real battery cells. It is possible that adjusting a number of parameters which now are fixed (for example, the RK parameters) we can generate outputs that resemble the data more closely.

On the bright side, we can make discuss the merits of the input design; it has in fact helped the parameter estimation, because it has made it harder, or rarer, to fall in other local optima. While it is possible that a better fitting parameter is hiding in another local optimum, it is now highly unlikely.

It looks like the parameter inferred from Test 2, i.e., the longest experiment, is the one better approximating the data. Test 4, which is over 4 times shorter also produces a parameter that approximates well the real data dynamics. As usual, using more expensive experiments produces data of better quality, while using shorter experiments might save time by sacrificing some of the accuracy. We believe the main advantage of using the input design algorithm is the enforcement of the well-posedness of the Hessian matrix corresponding to the parameter optimization problem. The short optimization times might be the best experimental benefit, as many optimization methods work very well with convex problems.

The input design should be applied to more complex numerical models, able to accurately represent the battery cell in all conditions: it can indeed happen that our single particle model fails in settings chosen by the input design (for example, very high or very low state of charge) where valuable information lies.

We can, of course, also argue that the parameter estimations has failed because the input design was badly set up. We have selected inputs with very restricted shapes and lengths in an arbitrary and intuitive way. For example, with longer current jumps, or longer pauses, or in general longer experiment, the input design could have found the truly optimal parameter. Of course, the possibilities of generalization are endless, and we encourage future works to pursue better results.

Chapter 5

Conclusions and outlook

In this thesis, we have worked on two fronts. On the one side, we have further developed and used reduced basis methods in the framework of trust region optimization algorithm. Here we have worked at first with an abstract linear parabolic problem, and then with a nonlinear parabolic-elliptic coupled system, where the complexity of the model has required us to develop approximated hierarchical a-posteriori error estimates for the state variables and the cost function. With these error bounds we have applied a trust region optimization algorithm already studied and discussed in previous publications. We have, however, showed qualitatively the effectivity of our approximated estimates and the positive impact on the optimization results.

On the other hand, we have built a framework for the input design problem, motivated the use of such field, and applied a pretty standard adaptive algorithm. Then, we have used this algorithm on an actual industrial problem, and when we did so we realized that changes in the design had to be made, hence exploring another design philosophy. The results of the input design algorithm showed indeed the wanted results: by choosing a specific input, or collection of inputs, we were able to better infer the underlying parameters.

Of course, the possibilities for extensions, generalizations, or steps forward are plentiful. One theoretical achievement would be to exactly quantify the approximation error of the hierarchical error estimation in order to obtain certified estimates.

Regarding the input design, one fundamental change would be to consider the Bayesian design of experiment (see, e.g., [25, 3, 1] for Bayesian design of experiment for linear and nonlinear problems). Although the nonlinearity complicates the theoretical framework of Bayesian inference, it would be of interest to change the prospective into improving robustly the statistical properties of the parameter distribution.

Moreover, we could certainly make progress on the industrial application. The main simplification was the complexity of the approximated battery cell system and the number of parameters. Generalizing the battery model, and enlarging the complexity dimension would be beneficial for the estimation and optimization of real batteries, even though an optimal input could result in a more complex experiment, with a much longer current profile and of course a higher resource cost.

The field of input/experimental design is a field that offers a valuable point of view in the manage of resources. In a world where the ethic aspect of science appears more and more important, a way of reducing the effort derived, for example, from time waste and energy consumptions, is welcome. As such, we see the potential value of using the design techniques in all the field that require the use of industrial and experimental setups.

Appendix A

Additional proofs

A.1 Existence of a minimizer for (P_1) and optimality conditions

In this section we will discuss existence of a minimizer for (2.2), and define the necessary first-order optimality conditions. Following [31, Section 1.5.2], we consider the nonlinear problem

$$\min_{(y,\boldsymbol{\mu})\in\mathcal{Y}\times\mathcal{P}} J(y,\boldsymbol{\mu}) \quad \text{subject to} \quad e(y,\boldsymbol{\mu})=0, \quad \boldsymbol{\mu}\in\mathcal{P}_{\mathsf{ad}}. \tag{A.1}$$

We suppose

Assumption 4.

1. $\mathcal{P}_{\mathsf{ad}}\subset\mathcal{P}$ is convex, bounded and closed, such that (A.1) has a feasible point.
2. $J:\mathcal{Y}\times\mathcal{P}\to\mathbb{R}$ and $e:\mathcal{Y}\times\mathcal{P}\to\mathcal{Z}^*$ are continuously Fréchet differentiable and $\mathcal{P},\mathcal{Y},\mathcal{Z}$ are Banach spaces.
3. For all $\boldsymbol{\mu}\in V$ in a neighborhood $V\subset\mathcal{P}$ of $\mathcal{P}_{\mathsf{ad}}$, the state equation $e(y,\boldsymbol{\mu})=0$ has a unique solution $y=y_{\boldsymbol{\mu}}\in\mathcal{Y}$.
4. $e_y(y_{\boldsymbol{\mu}},\boldsymbol{\mu})\in\mathcal{L}(\mathcal{Y},\mathcal{Z}^*)$ (namely the partial derivative of e w.r.t. y) has a bounded inverse for all $\boldsymbol{\mu}\in V\supset\mathcal{P}_{\mathsf{ad}}$.

Existence of the optimal solution $(\bar{y},\bar{\boldsymbol{\mu}})$ is given in [31, Theorem 1.45] as soon as Assumption 4 is satisfied. In the rest of this section we will show how this assumption is satisfied for the type of problem introduced in Chapter 2. We will not include the study for the nonlinear system of Chapter 3, but it is similar. Let us furthermore mention that Assumption 4 is also used in [31, Corollary 1.3] to define necessary the first-order optimality conditions and the adjoint equation that we have defined in this work.

Let us start by introducing an equivalent form (cf. [31, (1.66)]) for the abstract parabolic evolution problem (2.2):

$$\int_0^T \langle y_t(t),\varphi(t)\rangle_{V^*,V}\,\mathrm{d}t+\int_0^T a_{\boldsymbol{\mu}}(y(t),\varphi(t))\,\mathrm{d}t=\int_0^T \langle f_{\boldsymbol{\mu}}(t;u),\varphi(t)\rangle_{V^*,V}\,\mathrm{d}t$$
$$y(0)=y_\circ \quad \text{in } H$$

for all $\varphi \in L^2(0,T;V)$. Let us also define the spaces $\mathcal{Z} := \mathcal{Z}_1 \times \mathcal{Z}_2 := L^2(0,T;V) \times H$, with $\mathcal{Z}^* \cong L^2(0,T;V^*) \times H =: \mathcal{Z}_1^* \times \mathcal{Z}_2^*$, and the constraint operator $e := (e_1, e_2) : \mathcal{X} \to \mathcal{Z}^*$ be defined for every $(y, \boldsymbol{\mu}) \in \mathcal{X}$ as

$$\begin{aligned}
&\langle e(y,\boldsymbol{\mu}), (\varphi,\psi)\rangle_{\mathcal{Z}^*,\mathcal{Z}} = \langle e_1(y,\boldsymbol{\mu}),\varphi\rangle_{\mathcal{Z}_1^*,\mathcal{Z}_1} + \langle e_2(y,\boldsymbol{\mu}),\psi\rangle_H \\
&\langle e_1(y,\boldsymbol{\mu}),\varphi\rangle_{\mathcal{Z}_1^*,\mathcal{Z}_1} = \int_0^T \Big[\langle y_t(t),\varphi(t)\rangle_{V^*,V} + a_{\boldsymbol{\mu}}(y(t),\varphi(t)) - \langle f_{\boldsymbol{\mu}}(t;u),\varphi(t)\rangle_{V^*,V}\Big]\,\mathrm{d}t \\
&\langle e_2(y,\boldsymbol{\mu}),\psi\rangle_H = \langle y(0)-y_\circ,\psi\rangle_H \iff e_2(y,\boldsymbol{\mu}) = y(0)-y_\circ \in H.
\end{aligned}$$

The next propositions prove the Fréchet differentiability of e (which implies also continuity) and the continuous invertibility of e_y, hence falling in Assumption 4.

Proposition A.1. *The operator $e : \mathcal{X} \to \mathcal{Z}^*$ is Fréchet differentiable.*

Proof. Let $x^\delta = (y^\delta, \boldsymbol{\mu}^\delta) \in \mathcal{X}$ be a given direction, and $\tau > 0$. For every $x = (y,\boldsymbol{\mu}) \in \mathcal{X}$, $(\varphi,\psi) \in \mathcal{Z}$ and $t \in [0,T]$ f.a.a. we find:

$$\begin{aligned}
&\frac{1}{\tau}\Big(a_{\boldsymbol{\mu}+\tau\boldsymbol{\mu}^\delta}(y(t)+\tau y^\delta(t),\varphi(t)) - a_{\boldsymbol{\mu}}(y(t),\varphi(t))\Big) \\
&= \frac{1}{\tau}\left(\sum_{l=1}^{m_a}\vartheta_l^a(\boldsymbol{\mu}+\tau\boldsymbol{\mu}^\delta)\hat{a}_l(y(t)+\tau y^\delta(t),\varphi(t)) - \sum_{l=1}^{m_a}\vartheta_l^a(\boldsymbol{\mu})\hat{a}_l(y(t),\varphi(t))\right) \\
&= \sum_{l=1}^{m_a}\frac{\vartheta_l^a(\boldsymbol{\mu}+\tau\boldsymbol{\mu}^\delta)-\vartheta_l^a(\boldsymbol{\mu})}{\tau}\hat{a}_l(y(t),\varphi(t)) + \sum_{l=1}^{m_a}\vartheta_l^a(\boldsymbol{\mu}+\tau\boldsymbol{\mu}^\delta)\hat{a}_l(y^\delta(t),\varphi(t)) \\
&\xrightarrow{\tau\to 0} \sum_{l=1}^{m_a}\nabla_{\boldsymbol{\mu}^\delta}\vartheta_l^a(\boldsymbol{\mu})\hat{a}_l(y(t),\varphi(t)) + \sum_{l=1}^{m_a}\vartheta_l^a(\boldsymbol{\mu})\hat{a}_l(y^\delta(t),\varphi(t)) \\
&\qquad = \nabla_{\boldsymbol{\mu}^\delta} a_{\boldsymbol{\mu}}(y(t),\varphi(t)) + a_{\boldsymbol{\mu}}(y^\delta(t),\varphi(t)),
\end{aligned} \tag{A.2}$$

where $\nabla_{\mathbf{v}} f(\mathbf{x}) = \mathbf{v}\cdot\nabla f(\mathbf{x})$ is the directional derivative, and

$$\begin{aligned}
&\frac{1}{\tau}\Big(\langle f_{\boldsymbol{\mu}+\tau\boldsymbol{\mu}^\delta}(t;u),\varphi(t)\rangle_{V^*,V} - \langle f_{\boldsymbol{\mu}}(t;u),\varphi(t)\rangle_{V^*,V}\Big) \\
&= \langle\sum_{l=1}^{m_f}\vartheta_l^f(\boldsymbol{\mu}+\tau\boldsymbol{\mu}^\delta)\hat{f}_l(t;u),\varphi(t)\rangle_{V^*,V} - \langle\sum_{l=1}^{m_f}\vartheta_l^f(\boldsymbol{\mu})\hat{f}_l(t;u),\varphi(t)\rangle_{V^*,V} \\
&= \sum_{l=1}^{m_f}\frac{\vartheta_l^f(\boldsymbol{\mu}+\tau\boldsymbol{\mu}^\delta)-\vartheta_l^f(\boldsymbol{\mu})}{\tau}\langle\hat{f}_l(t;u),\varphi(t)\rangle_{V^*,V} \\
&\xrightarrow{\tau\to 0} \sum_{l=1}^{m_f}\nabla_{\boldsymbol{\mu}^\delta}\vartheta_l^f(\boldsymbol{\mu})\langle\hat{f}_l(t;u),\varphi(t)\rangle_{V^*,V} = \langle\nabla_{\boldsymbol{\mu}^\delta} f_{\boldsymbol{\mu}}(t;u),\varphi(t)\rangle_{V^*,V}.
\end{aligned} \tag{A.3}$$

Hence, using (A.2) and (A.3) we get

$$\begin{aligned}
&\left\langle\frac{1}{\tau}\Big[e(x+\tau x^\delta)-e(x)\Big],(\varphi,\psi)\right\rangle_{\mathcal{Z}^*,\mathcal{Z}} \\
&\xrightarrow{\tau\to 0} \int_0^T\Big[\langle y_t^\delta(t),\varphi(t)\rangle_{V^*,V} + \nabla_{\boldsymbol{\mu}^\delta}a_{\boldsymbol{\mu}}(y(t),\varphi(t)) + a_{\boldsymbol{\mu}}(y^\delta(t),\varphi(t)) \\
&\qquad - \langle\nabla_{\boldsymbol{\mu}^\delta} f_{\boldsymbol{\mu}}(t;u),\varphi(t)\rangle_{V^*,V}\Big]\mathrm{d}t + \langle y^\delta(0),\psi\rangle_H.
\end{aligned}$$

Let us hence define $e^* := (e_1^*, e_2^*) : \mathcal{X} \to \mathcal{Z}^*$ as

$$\begin{aligned}\langle e^*(x)x^\delta, (\varphi,\psi)\rangle_{\mathcal{Z}^*,\mathcal{Z}} &:= \langle e_1^*(x)x^\delta, \varphi\rangle_{Z_1^*,Z_1} + \langle e_2^*(x)x^\delta, \psi\rangle_H \\ &:= \int_0^T \Big[\langle y_t^\delta(t), \varphi(t)\rangle_{V^*,V} + \nabla_{\boldsymbol{\mu}^\delta} a_{\boldsymbol{\mu}}(y(t),\varphi(t)) + a_{\boldsymbol{\mu}}(y^\delta(t),\varphi(t)) \\ &\qquad - \langle \nabla_{\boldsymbol{\mu}^\delta} f_{\boldsymbol{\mu}}(t;u), \varphi(t)\rangle_{V^*,V}\Big]\,\mathrm{d}t + \langle y^\delta(0),\psi\rangle_H.\end{aligned}$$

for $(\varphi,\psi) \in \mathcal{Z}$, $x = (y,\boldsymbol{\mu}) \in \mathcal{X}_{\mathsf{ad}}$ and $x^\delta = (y^\delta, \boldsymbol{\mu}^\delta)$. To prove that e^* is the Fréchet derivative of e, we consider:

$$\begin{aligned}&\langle e(x+x^\delta) - e(x) - e^*(x)x^\delta, (\varphi,\psi)\rangle_{\mathcal{Z}^*,\mathcal{Z}} \\ &= \int_0^T \Big[a_{\boldsymbol{\mu}+\boldsymbol{\mu}^\delta}(y(t)+y^\delta(t),\varphi(t)) - a_{\boldsymbol{\mu}}(y(t),\varphi(t)) - \nabla_{\boldsymbol{\mu}^\delta} a_{\boldsymbol{\mu}}(y(t),\varphi(t)) - a_{\boldsymbol{\mu}}(y^\delta(t),\varphi(t)) \\ &\qquad - \langle f_{\boldsymbol{\mu}+\boldsymbol{\mu}^\delta}(t;u),\varphi(t)\rangle_{V^*,V} + \langle f_{\boldsymbol{\mu}}(t;u),\varphi(t)\rangle_{V^*,V} + \langle \nabla_{\boldsymbol{\mu}^\delta} f_{\boldsymbol{\mu}}(t;u),\varphi(t)\rangle_{V^*,V}\Big]\,\mathrm{d}t \\ &= \int_0^T \Bigg[\sum_{l=1}^{m_a}\Big(\vartheta_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta) - \vartheta_l^a(\boldsymbol{\mu})\Big)\hat{a}_l(y^\delta(t),\varphi(t)) \\ &\qquad + \sum_{l=1}^{m_a}\Big(\vartheta_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta) - \vartheta_l^a(\boldsymbol{\mu}) - \nabla_{\boldsymbol{\mu}^\delta}\vartheta_l^a(\boldsymbol{\mu})\Big)\hat{a}_l(y(t),\varphi(t)) \\ &\qquad - \sum_{l=1}^{m_f}\Big(\vartheta_l^f(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta) - \vartheta_l^f(\boldsymbol{\mu}) - \nabla_{\boldsymbol{\mu}^\delta}\vartheta_l^f(\boldsymbol{\mu})\Big)\langle \hat{f}_l(t;u),\varphi(t)\rangle_{V^*,V}\Bigg]\,\mathrm{d}t.\end{aligned} \tag{A.4}$$

Then, since the functions ϑ_l^a and ϑ_l^f are continuously differentiable, the norms $\|\nabla\vartheta_l^a\|_2$ and $\|\nabla\vartheta_l^f\|_2$ are bounded for any l, and there exist function h_l^a and h_l^f with

$$\lim_{\boldsymbol{\mu}^\delta\to 0} h_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta) = 0 \qquad \text{and} \qquad \lim_{\boldsymbol{\mu}^\delta\to 0} h_l^f(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta) = 0,$$

such that

$$\vartheta_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta) - \vartheta_l^a(\boldsymbol{\mu}) = \nabla_{\boldsymbol{\mu}^\delta}\vartheta_l^a(\boldsymbol{\mu}) + h_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta)\|\boldsymbol{\mu}^\delta\|_2, \quad \text{for } l = 1,\dots,m_a \tag{A.5}$$

and

$$\vartheta_l^f(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta) - \vartheta_l^f(\boldsymbol{\mu}) = \nabla_{\boldsymbol{\mu}^\delta}\vartheta_l^f(\boldsymbol{\mu}) + h_l^f(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta)\|\boldsymbol{\mu}^\delta\|_2, \quad \text{for } l = 1,\dots,m_f. \tag{A.6}$$

Furthermore, we can write

$$\big|\nabla_{\boldsymbol{\mu}^\delta}\vartheta_l^a(\boldsymbol{\mu})\big| \le \|\nabla\vartheta_l^a(\boldsymbol{\mu})\|_2\|\boldsymbol{\mu}^\delta\|_2 \qquad \text{and} \qquad \Big|\nabla_{\boldsymbol{\mu}^\delta}\vartheta_l^f(\boldsymbol{\mu})\Big| \le \|\nabla\vartheta_l^f(\boldsymbol{\mu})\|_2\|\boldsymbol{\mu}^\delta\|_2. \tag{A.7}$$

From the continuity of $\hat{a}_l$ for all $l = 1,\dots,m_a$ we also get

$$\int_0^T |\hat{a}_l(y^\delta(t),\varphi(t))|\mathrm{d}t \le c_a\int_0^T \|y^\delta(t)\|_V\|\varphi(t)\|_V\mathrm{d}t \le c_a\|y^\delta\|_{\mathcal{Y}}\|\varphi\|_{L^2(0,T;V)} \tag{A.8}$$

for all $l = 1,\dots,m_a$, $y^\delta \in \mathcal{Y}$ and $\varphi \in \mathcal{Z}_1$, where c_a only depends on the continuity constants.

Hence, using (A.5), (A.7) and (A.8) we get

$$\begin{aligned}
&\frac{1}{\|x^\delta\|_{\mathcal{X}}} \sup_{\|(\varphi,\psi)\|_{\mathcal{Z}}=1} \left| \int_0^T \left[\sum_{l=1}^{m_a} \left(\vartheta_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta) - \vartheta_l^a(\boldsymbol{\mu}) \right) \hat{a}_l(y^\delta(t),\varphi(t)) \right] \mathrm{d}t \right| \\
&\leq \frac{c_a}{\|x^\delta\|_{\mathcal{X}}} \sum_{l=1}^{m_a} \left(\|\nabla \vartheta_l^a(\boldsymbol{\mu})\|_2 + |h_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta)| \right) \|\boldsymbol{\mu}^\delta\|_2 \sup_{\|(\varphi,\psi)\|_{\mathcal{Z}}=1} \|y^\delta\|_Y \|\varphi\|_{\mathcal{Z}_1} \\
&\leq \frac{c_a}{\|x^\delta\|_{\mathcal{X}}} \sum_{l=1}^{m_a} \left(\|\nabla \vartheta_l^a(\boldsymbol{\mu})\|_2 + |h_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta)| \right) \|\boldsymbol{\mu}^\delta\|_2 \|y^\delta\|_{\mathcal{Y}} \\
&\leq \frac{c_a}{\|x^\delta\|_{\mathcal{X}}} \sum_{l=1}^{m_a} \left(\|\nabla \vartheta_l^a(\boldsymbol{\mu})\|_2 + |h_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta)| \right) \frac{1}{2} \left(\|\boldsymbol{\mu}^\delta\|_2^2 + \|y^\delta(t)\|_{\mathcal{Y}}^2 \right) \\
&\leq \frac{c_a}{\|x^\delta\|_{\mathcal{X}}} \sum_{l=1}^{m_a} \left(\|\nabla \vartheta_l^a(\boldsymbol{\mu})\|_2 + |h_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta)| \right) \frac{1}{2} \|x^\delta\|_{\mathcal{X}}^2 \to 0 \qquad \text{for } \|x^\delta\|_{\mathcal{X}} \to 0. \qquad \text{(A.9)}
\end{aligned}$$

Here we have used the fact that the assumption on the differentiability of ϑ_l^a for all $l = 1,\dots,m_a$ implies that $\|\nabla \vartheta_l^a(\boldsymbol{\mu})\|_2$ is bounded, that $1 = \|(\varphi,\psi)\|_Z = \sqrt{\|\varphi\|_{Z_1}^2 + \|\psi\|_H^2}$ implies $\|\varphi\|_{\mathcal{Z}_1} \leq 1$ and that the limit $\|x^\delta\|_{\mathcal{X}} \to 0$ implies $\boldsymbol{\mu}^\delta \to 0$ and thus $|h_l^a(\boldsymbol{\mu}+ bmu^\delta)| \to 0$.

Similarly, we get

$$\begin{aligned}
&\frac{1}{\|x^\delta\|_{\mathcal{X}}} \sup_{\|(\varphi,\psi)\|_{\mathcal{Z}}=1} \left| \int_0^T \left[\sum_{l=1}^{m_a} \left(\vartheta_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta) - \vartheta_l^a(\boldsymbol{\mu}) - \nabla_{\boldsymbol{\mu}^\delta} \vartheta_l^a(\boldsymbol{\mu}) \right) \hat{a}_l(y(t),\varphi(t)) \right] \mathrm{d}t \right| \\
&\leq \frac{c_a}{\|x^\delta\|_{\mathcal{X}}} \sum_{l=1}^{m_a} \left(|h_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta)| \right) \|\boldsymbol{\mu}^\delta\|_2 \sup_{\|(\varphi,\psi)\|_{\mathcal{Z}}=1} \|y\|_Y \|\varphi\|_{\mathcal{Z}_1} \\
&\leq \frac{c_a}{\|x^\delta\|_{\mathcal{X}}} \sum_{l=1}^{m_a} \left(|h_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta)| \right) \|\boldsymbol{\mu}^\delta\|_2 \|y\|_{\mathcal{Y}} \to 0 \qquad \text{for } \|x^\delta\|_{\mathcal{X}} \to 0, \qquad \text{(A.10)}
\end{aligned}$$

since in the limit $\|x^\delta\|_{\mathcal{X}} \to 0$ we have $\|\boldsymbol{\mu}^\delta\|_2 / \|x^\delta\|_{\mathcal{X}} \to 1$ and $|h_l^a(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta)| \to 0$. For the same reason and using the continuity of $\hat{f}_l(t;u)$ for $l = 1,\dots,m_f$ and f.a.a. $t \in [0,T]$, we have

$$\begin{aligned}
&\frac{1}{\|x^\delta\|_{\mathcal{X}}} \sup_{\|(\varphi,\psi)\|_{\mathcal{Z}}=1} \left| \int_0^T \left[\sum_{l=1}^{m_f} \left(\vartheta_l^f(\boldsymbol{\mu}+\boldsymbol{\mu}^\delta) - \vartheta_l^f(\boldsymbol{\mu}) - \nabla_{\boldsymbol{\mu}^\delta} \vartheta_l^f(\boldsymbol{\mu}) \right) \langle \hat{f}_l(t;u), \varphi(t) \rangle_{V^*,V} \right] \mathrm{d}t \right| \\
&\qquad\qquad \to 0 \qquad \text{for } \|x^\delta\|_{\mathcal{X}} \to 0. \qquad \text{(A.11)}
\end{aligned}$$

Finally, using (A.4), (A.9), (A.10) and (A.11) we have

$$\frac{\|e(x+x^\delta) - e(x) - e^*(x)x^\delta\|_{\mathcal{Z}^*}}{\|x^\delta\|_{\mathcal{X}}} = \sup_{\|(\varphi,\psi)\|_{\mathcal{Z}}=1} \frac{\langle e(x+x^\delta) - e(x) - e^*(x)x^\delta, (\varphi,\psi) \rangle_{\mathcal{Z}^*,\mathcal{Z}}}{\|x^\delta\|_{\mathcal{X}}} \to 0$$

for $\|x^\delta\|_{\mathcal{X}} \to 0$, i.e. e is Fréchet-differentiable and its Fréchet derivative coincides with the directional derivative derived above. □

Proposition A.2. *Let $x = (y, \boldsymbol{\mu}) \in \mathcal{X}_{\mathsf{ad}}$ be a given point. Then, the operator $e_y(x) : \mathcal{Y} \to \mathcal{Z}^*$ is continuously invertible.*

Proof. For a given $F \in \mathcal{Z}^*$ (that is, $F := (F_1, F_2) \in L^2(0,T;V^*) \times H$) the equation $e_y(x)y^\delta = F$ in $\mathcal{Z}^*$ is equivalent to the variational formulation:

$$\langle y_t^\delta(t), \varphi\rangle_{V^*,V} + a_{\boldsymbol{\mu}}(y^\delta(t), \varphi) = \langle F_1(t), \varphi\rangle_{V^*,V} \qquad \forall \varphi \in V, \text{ a.e. in } [0,T], \tag{A.12a}$$

$$y^\delta(0) = F_2, \qquad \text{in } H. \tag{A.12b}$$

As mentioned in Chapter 2, due to the properties of the bilinear form $a_{\boldsymbol{\mu}}$, we can state that (A.12) has a unique solution $y^\delta \in \mathcal{Y}$ for every $\boldsymbol{\mu} \in \mathcal{P}_{\mathsf{ad}}$. Then, the energy estimate gives

$$\|y^\delta\|_{\mathcal{Y}} \le C\|F\|_{\mathcal{Z}^*}$$

for a constant $C > 0$. Since $y^\delta = e_y(x)^{-1}F \in \mathcal{Y}$ holds, we can write:

$$\|e_y(x)^{-1}F\|_{\mathcal{Y}} \le C\|F\|_{\mathcal{Z}^*} \qquad \text{for any } F \in \mathcal{Z}^*,$$

which shows that the operator $e_y(x)^{-1}$ is bounded. □

A.2 Sensitivity equations for the coupled model

Both in Section 3.4 and in Section 4.1.2 we make use of the *sensitivity variables* (called *sensitivities* for simplicity), that quantify how state variables are impacted by the singular model parameters. In particular, they are found as solutions of weak *sensitivity equations*, obtained by deriving the state equations with respect to the parameters. In this section we define and discretize the sensitivity equations for the nonlinear coupled model of Chapter 3.

For all $1 \le j \le d$, the j-th sensitivities $s^{\mathrm{y}}_{j,\boldsymbol{\mu}} \in \mathcal{Y}$ and $s^{\mathrm{q}}_{j,\boldsymbol{\mu}} \in \mathcal{Q}$, respectively of y and q, are the solution of the j-th weak *sensitivity equations*, obtained by differentiating the weak state equations (3.5) with respect to μ_j:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\langle s^{\mathrm{y}}_{j,\boldsymbol{\mu}}(t), \varphi^y\rangle_H + \langle g^{\mathrm{sy},j}_{\boldsymbol{\mu}}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t), s^{\mathrm{y}}_{j,\boldsymbol{\mu}}(t), s^{\mathrm{q}}_{j,\boldsymbol{\mu}}(t)], \varphi^y\rangle_{V^*,V} &= 0, \\ s^{\mathrm{y}}_{j,\boldsymbol{\mu}}(0) &= 0 \qquad \text{in } H, \\ \langle g^{\mathrm{sq},j}_{\boldsymbol{\mu}}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t), s^{\mathrm{y}}_{j,\boldsymbol{\mu}}(t), s^{\mathrm{q}}_{j,\boldsymbol{\mu}}(t)], \varphi^q\rangle_{V_\circ^*,V_\circ} &= 0, \end{aligned}$$

for all $\varphi^y \in V$, $\varphi^q \in V_\circ$ and f.a.a. $t \in (0,T]$, where

$$\begin{aligned} &\langle g^{\mathrm{sy},j}_{\boldsymbol{\mu}}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t), s^{\mathrm{y}}_{j,\boldsymbol{\mu}}(t), s^{\mathrm{q}}_{j,\boldsymbol{\mu}}(t)], \varphi^y\rangle_{V^*,V} \\ &\quad := a^1_{\boldsymbol{\mu}}(s^{\mathrm{y}}_{j,\boldsymbol{\mu}}(t), \varphi^y) + \frac{\partial}{\partial\mu_j}a^1_{\boldsymbol{\mu}}(y_{\boldsymbol{\mu}}(t), \varphi^y) + \langle\frac{\partial}{\partial\mu_j}g^1_{\boldsymbol{\mu}}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)], \varphi^y\rangle_{V^*,V} \\ &\qquad - \mu_2\langle g^{\partial}_j[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t), s^{\mathrm{y}}_{j,\boldsymbol{\mu}}(t), s^{\mathrm{q}}_{j,\boldsymbol{\mu}}(t)], \varphi^y\rangle_H \end{aligned}$$

and

$$\begin{aligned} &\langle g^{\mathrm{sq},j}_{\boldsymbol{\mu}}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t), s^{\mathrm{y}}_{j,\boldsymbol{\mu}}(t), s^{\mathrm{q}}_{j,\boldsymbol{\mu}}(t)], \varphi^q\rangle_{V_\circ^*,V_\circ} \\ &\quad := a^2_{\boldsymbol{\mu}}(s^{\mathrm{q}}_{j,\boldsymbol{\mu}}(t), \varphi^q) + \frac{\partial}{\partial\mu_j}a^2_{\boldsymbol{\mu}}(q_{\boldsymbol{\mu}}(t), \varphi^q) + \langle\frac{\partial}{\partial\mu_j}g^2_{\boldsymbol{\mu}}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)], \varphi^q\rangle_{V_\circ^*,V_\circ} \\ &\qquad + \mu_4\langle g^{\partial}_j[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t), s^{\mathrm{y}}_{j,\boldsymbol{\mu}}(t), s^{\mathrm{q}}_{j,\boldsymbol{\mu}}(t)], \varphi^q\rangle_H. \end{aligned}$$

The term g_j^{∂} is given as

$$\begin{aligned}&\langle g_j^{\partial}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t), s^{\mathrm{y}}_{j,\boldsymbol{\mu}}(t), s^{\mathrm{q}}_{j,\boldsymbol{\mu}}(t)], \varphi\rangle_H \\ &\qquad := \langle \partial_y f(y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)) s^{\mathrm{y}}_{j,\boldsymbol{\mu}}(t), \varphi\rangle_H + \langle \partial_q f(y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)) s^{\mathrm{q}}_{j,\boldsymbol{\mu}}(t), \varphi\rangle_H,\end{aligned}$$

where $\partial_y f$ and $\partial_q f$ are the partial derivatives of f. The sensitivity equations are given explicitly in Section A.2, where we also define the FE approximations $\mathrm{s}^{\mathrm{y},h,k}_{j,\boldsymbol{\mu}}$ and $\mathrm{s}^{\mathrm{q},h,k}_{j,\boldsymbol{\mu}}$ for $j = 1, \dots, 4$ and $k = 0, \dots, K$.

We now include the sensitivity equations for the coupled model, for $j = 1, \dots, 4$. For simplicity, we hide the subscript $\boldsymbol{\mu}$ for all variables.

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}\langle s^{\mathrm{y}}_1(t), \varphi^y\rangle_H + \mu_1 \hat{a}^1(s^{\mathrm{y}}_1(t), \varphi^y) + \hat{a}^1(y(t), \varphi^y) \\ &\qquad -\mu_2\langle \partial_y f(y(t), q(t)) s^{\mathrm{y}}_1(t), \varphi^y\rangle_H - \mu_2\langle \partial_q f(y(t), q(t)) s^{\mathrm{q}}_1(t), \varphi^y\rangle_H = 0, \\ &\qquad\qquad s^{\mathrm{y}}_1(0) = 0, \\ &\mu_3 \hat{a}^2(s^{\mathrm{q}}_1(t), \varphi^q) + \mu_4\langle \partial_y f(y(t), q(t)) s^{\mathrm{y}}_1(t), \varphi^q\rangle_H + \mu_4\langle \partial_q f(y(t), q(t)) s^{\mathrm{q}}_1(t), \varphi^q\rangle_H = 0.\end{aligned}$$

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}\langle s^{\mathrm{y}}_2(t), \varphi^y\rangle_H + \mu_1 \hat{a}^1(s^{\mathrm{y}}_2(t), \varphi^y) - \langle \hat{g}[y(t), q(t)], \varphi^y\rangle_H \\ &\qquad -\mu_2\langle \partial_y f(y(t), q(t)) s^{\mathrm{y}}_2(t), \varphi^y\rangle_H - \mu_2\langle \partial_q f(y(t), q(t)) s^{\mathrm{q}}_2(t), \varphi\rangle_H = 0, \\ &\qquad\qquad s^{\mathrm{y}}_2(0) = 0, \\ &\mu_3 \hat{a}^2(s^{\mathrm{q}}_2(t), \varphi^q) + \mu_4\langle \partial_y f(y(t), q(t)) s^{\mathrm{y}}_2(t), \varphi^q\rangle_H + \mu_4\langle \partial_q f(y(t), q(t)) s^{\mathrm{q}}_2(t), \varphi^q\rangle_H = 0.\end{aligned}$$

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}\langle s^{\mathrm{y}}_3(t), \varphi^y\rangle_H + \mu_1 \hat{a}^1(s^{\mathrm{y}}_3(t), \varphi^y) \\ &\quad -\mu_2\langle \partial_y f(y(t), q(t)) s^{\mathrm{y}}_3(t), \varphi^y\rangle_H - \mu_2\langle \partial_q f(y(t), q(t)) s^{\mathrm{q}}_3(t), \varphi^y\rangle_H = 0, \\ &\qquad\qquad s^{\mathrm{y}}_3(0) = 0, \\ &\quad \mu_3 \hat{a}^2(s^{\mathrm{q}}_3, \varphi^q) + \hat{a}^2(y(t), \varphi^q) \\ &\qquad +\mu_4\langle \partial_y f(y(t), q(t)) s^{\mathrm{y}}_3(t), \varphi^q\rangle_H + \mu_4\langle \partial_q f(y(t), q(t)) s^{\mathrm{q}}_3(t), \varphi^q\rangle_H = 0.\end{aligned}$$

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}\langle s^{\mathrm{y}}_4(t), \varphi^y\rangle_H + \mu_1 \hat{a}^1(s^{\mathrm{y}}_4(t), \varphi^y) \\ &\quad -\mu_2\langle \partial_y f(y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)) s^{\mathrm{y}}_4(t), \varphi\rangle_H - \mu_2\langle \partial_q f(y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)) s^{\mathrm{q}}_4(t), \varphi\rangle_H = 0, \\ &\qquad\qquad s^{\mathrm{y}}_4(0) = 0, \\ &\quad \mu_3 \hat{a}^2(s^{\mathrm{q}}_4, \varphi^q) + \langle \hat{g}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)], \varphi^q\rangle_H \\ &\qquad +\mu_4\langle \partial_y f(y(t), q(t)) s^{\mathrm{y}}_4(t), \varphi^q\rangle_H + \mu_4\langle \partial_q f(y(t), q(t)) s^{\mathrm{q}}_4(t), \varphi^q\rangle_H = 0,\end{aligned}$$

where $\langle \hat{g}[y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)], \varphi\rangle_H := \langle f(y_{\boldsymbol{\mu}}(t), q_{\boldsymbol{\mu}}(t)), \varphi\rangle_H = \int_\Omega f(y_{\boldsymbol{\mu}}(t,x), q_{\boldsymbol{\mu}}(t,x))\varphi(x)\mathrm{d}x$.

The discretization is done following Section 3.2 on the FE spaces V^h and $V^h_\circ$ and using the implicit Euler scheme in time. Skipping the details, for $j = 1, 2, 3, 4$ we find coefficient

vectors $\{\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{y},h,k}\}_{k=0}^{K} \subset \mathbb{R}^{N+1}$ and $\{\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{q},h,k}\}_{k=0}^{K} \subset \mathbb{R}^{N}$, solutions of

$$\begin{aligned}
&\mathrm{M_y}\left(\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{y},h,k} - \mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{y},h,k-1}\right) + \mu_1\delta\mathrm{A_y}\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{y},h,k} - \mu_2\delta\mathrm{G}_{\mathrm{yy}}^{k}\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{y},h,k} \\
&\qquad - \mu_2\delta\mathrm{G}_{\mathrm{yq}}^{k}\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{q},h,k} - \delta\,\mathrm{c}_{j,\boldsymbol{\mu}}^{\mathrm{y},h,k} = 0, \qquad k = 1,\dots,K \\
&\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{y},h,0} = 0, \\
&\mu_3\mathrm{A_q}\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{q},h,k} + \mu_4\mathrm{G}_{\mathrm{qy}}^{k}\,\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{y},h,k} + \mu_4\mathrm{G}_{\mathrm{qq}}^{k}\,\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{q},h,k} - \mathrm{c}_{j,\boldsymbol{\mu}}^{\mathrm{q},h,k}, \qquad k = 0,\dots,K,
\end{aligned}$$

where the matrices G^k are defined in (3.27), and

$$\begin{aligned}
&\mathrm{c}_{1,\boldsymbol{\mu}}^{\mathrm{y},h,k} = -\mathrm{A_y}\mathrm{y}_{\boldsymbol{\mu}}^{h,k}, && \mathrm{c}_{1,\boldsymbol{\mu}}^{\mathrm{q},h,k} = 0, \\
&\mathrm{c}_{2,\boldsymbol{\mu}}^{\mathrm{y},h,k} = \mathrm{M_y}\mathrm{f}_{\mathrm{y}}^{h}(\mathrm{y}_{\boldsymbol{\mu}}^{h,k}, \mathrm{q}_{\boldsymbol{\mu}}^{h,k}), && \mathrm{c}_{2,\boldsymbol{\mu}}^{\mathrm{q},h,k} = 0, \\
&\mathrm{c}_{3,\boldsymbol{\mu}}^{\mathrm{y},h,k} = 0, && \mathrm{c}_{3,\boldsymbol{\mu}}^{\mathrm{q},h,k} = -\mathrm{A_q}\mathrm{q}_{\boldsymbol{\mu}}^{h,k}, \\
&\mathrm{c}_{4,\boldsymbol{\mu}}^{\mathrm{y},h,k} = 0, && \mathrm{c}_{4,\boldsymbol{\mu}}^{\mathrm{q},h,k} = -\mathrm{M_q}\mathrm{f}_{\mathrm{q}}^{h}(\mathrm{y}_{\boldsymbol{\mu}}^{h,k}, \mathrm{q}_{\boldsymbol{\mu}}^{h,k}).
\end{aligned}$$

Remark A.3. While for $k = 0$ and $j = 1, 2, 3, 4$ we just need to solve the linear system

$$\left(\mu_3\mathrm{A_q} + \mu_4\mathrm{G}_{\mathrm{qq}}^{k}\right)\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{q},h,0} = \mathrm{c}_{j,\boldsymbol{\mu}}^{\mathrm{q},h,0},$$

for $k = 1,\dots,K$ and $j = 1, 2, 3, 4$, we can define and solve the $(2N+1)$-dimensional coupled linear system $\tilde{\mathrm{M}}_{\boldsymbol{\mu}}^{k}\mathrm{z}_{j,\boldsymbol{\mu}}^{k} = \tilde{\mathrm{b}}_{j,\boldsymbol{\mu}}^{k}$, where $\mathrm{z}_{j,\boldsymbol{\mu}}^{k} := (\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{y},h,k\top}, \mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{q},h,k\top})^{\top}$,

$$\tilde{\mathrm{M}}_{\boldsymbol{\mu}}^{k} := \begin{bmatrix} \mathrm{M_y} + \mu_1\delta\mathrm{A_y} - \mu_2\delta\mathrm{G}_{\mathrm{yy}}^{k} & -\mu_2\delta\mathrm{G}_{\mathrm{yq}}^{k} \\ \mu_4\mathrm{G}_{\mathrm{qy}}^{k} & \mu_3\mathrm{A_q} + \mu_4\mathrm{G}_{\mathrm{qq}}^{k} \end{bmatrix}, \text{ and } \tilde{\mathrm{b}}_{j,\boldsymbol{\mu}}^{k} := \begin{bmatrix} \mathrm{M_y}\mathrm{s}_{j,\boldsymbol{\mu}}^{\mathrm{y},h,k-1} + \delta\mathrm{c}_{j,\boldsymbol{\mu}}^{\mathrm{y},h,k} \\ \mathrm{c}_{j,\boldsymbol{\mu}}^{\mathrm{q},h,k} \end{bmatrix}.$$

$\Diamond$

A.3 Real data plots

TEST 1

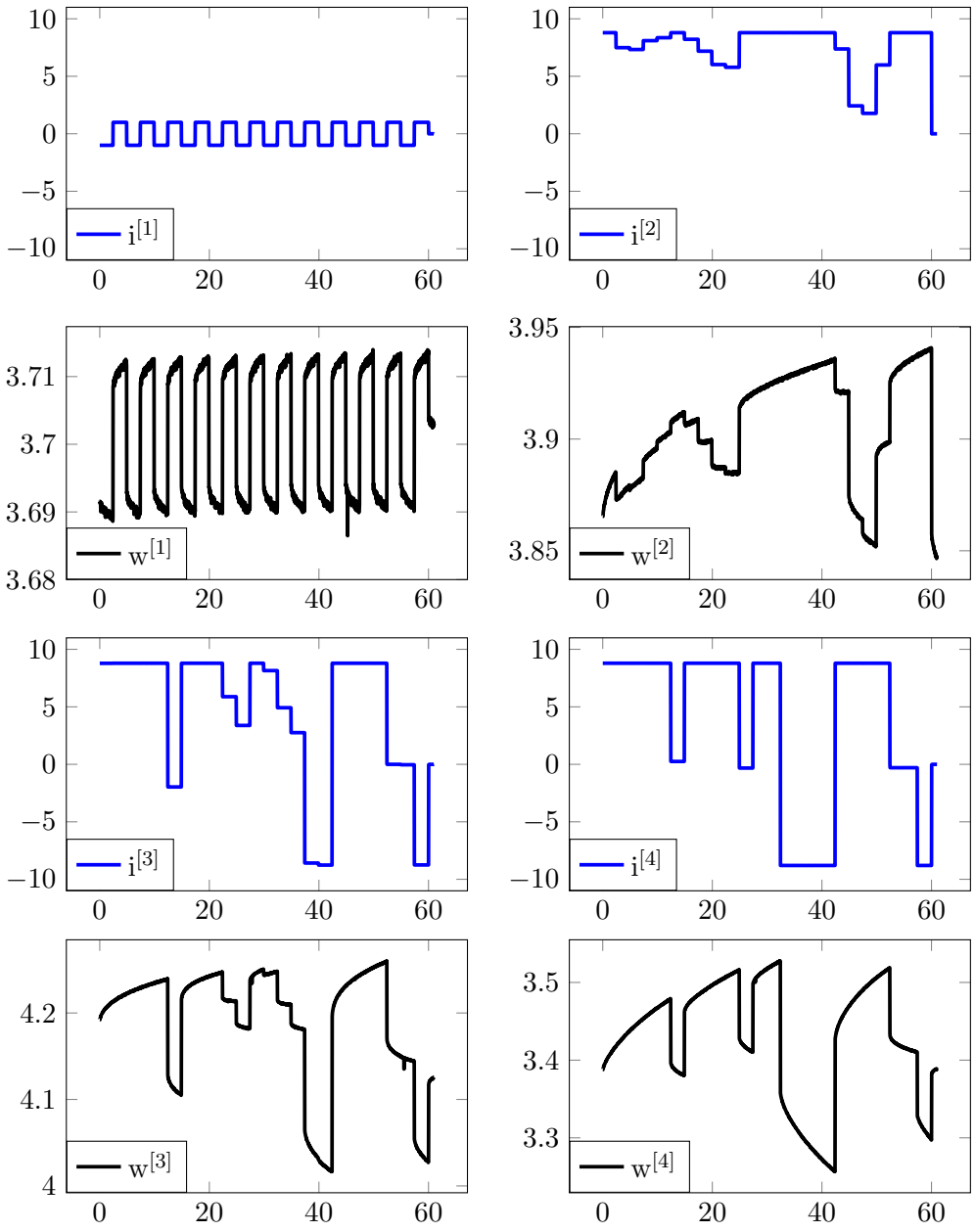


Figure A.1: Input i (in Ampere) and data w (in Volt) for TEST 1.

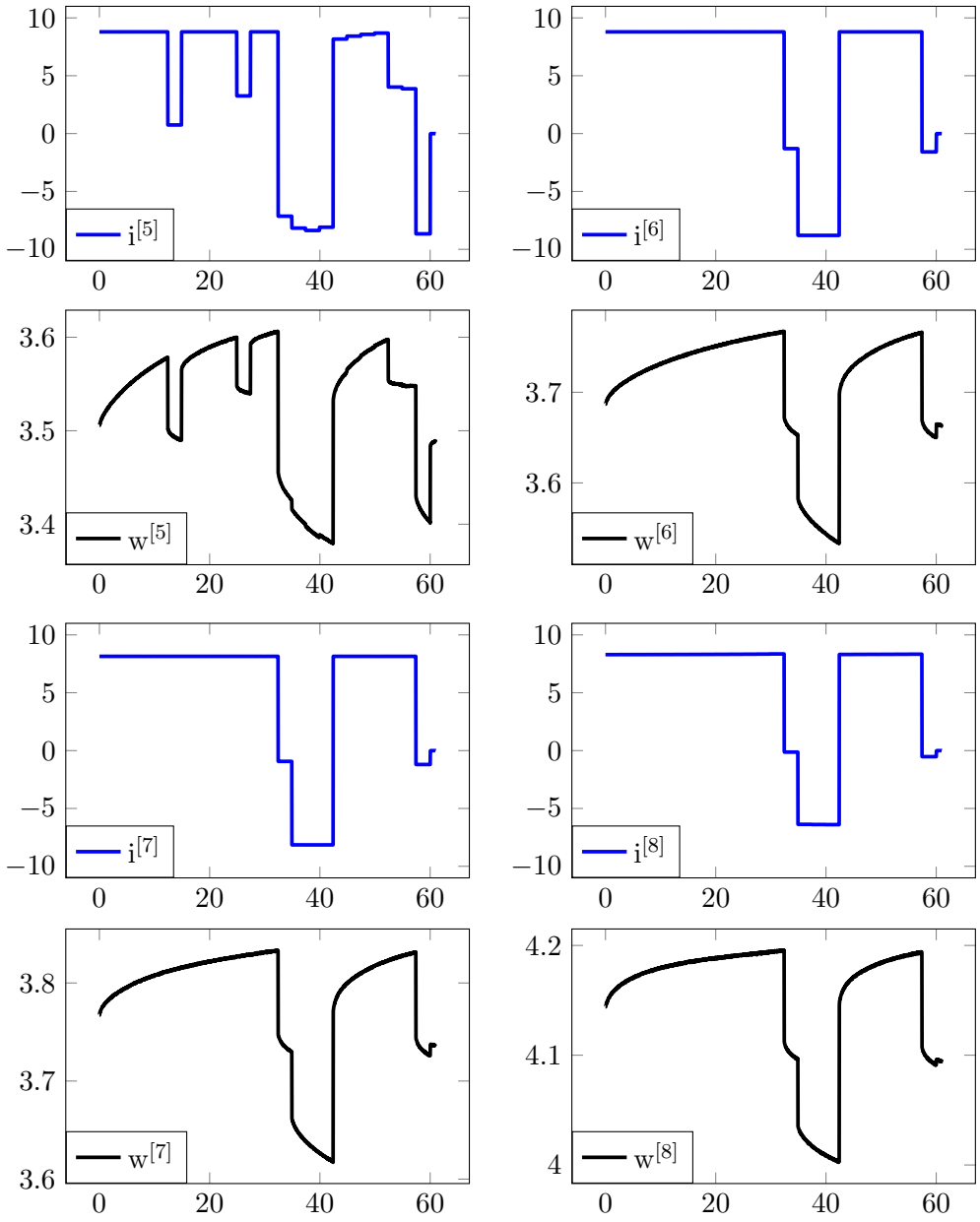


Figure A.1: Input i (in Ampere) and data w (in Volt) for TEST 1.

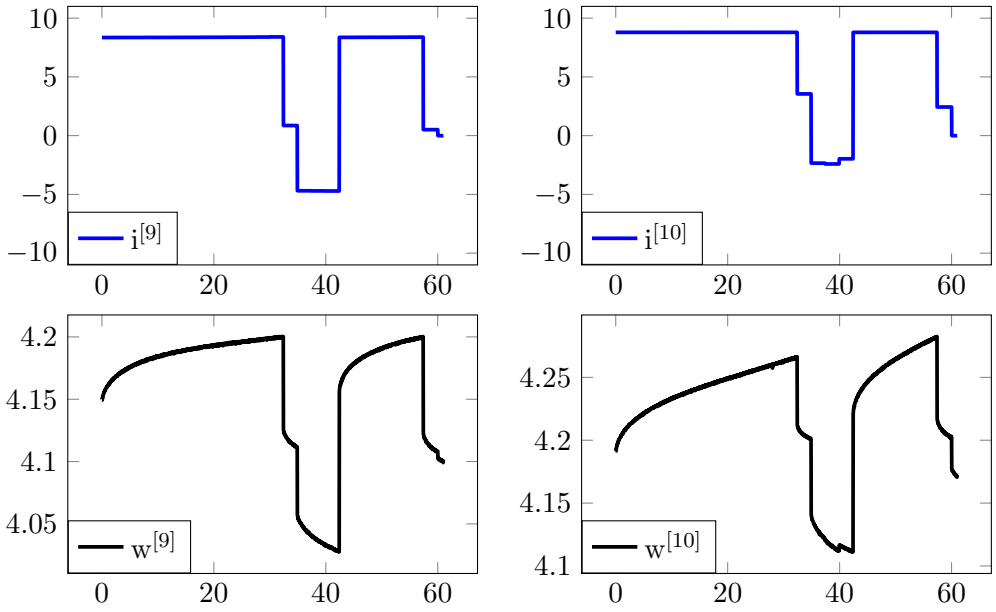


Figure A.1: Input i (in Ampere) and data w (in Volt) for TEST 1.

TEST 2

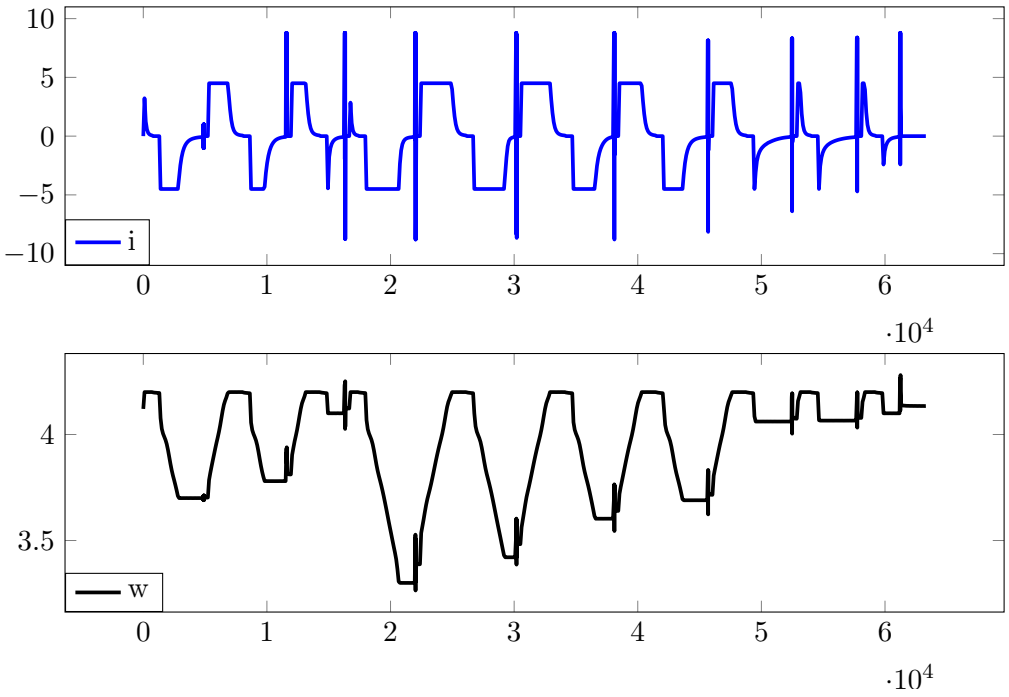


Figure A.2: Input i (in Ampere) and data w (in Volt) for TEST 2.

TEST 3

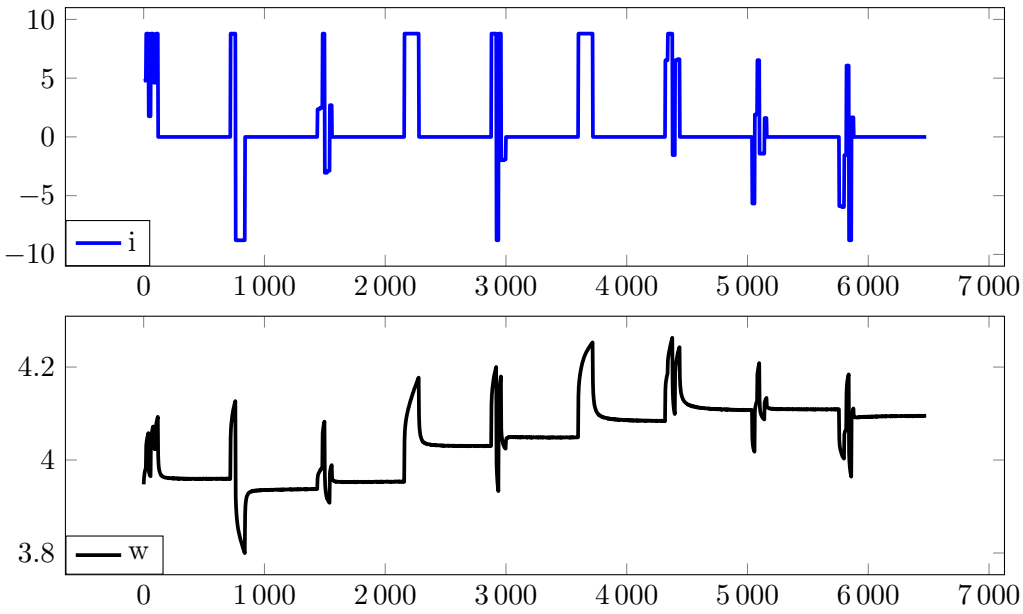


Figure A.3: Input i (in Ampere) and data w (in Volt) for TEST 3.

TEST 4

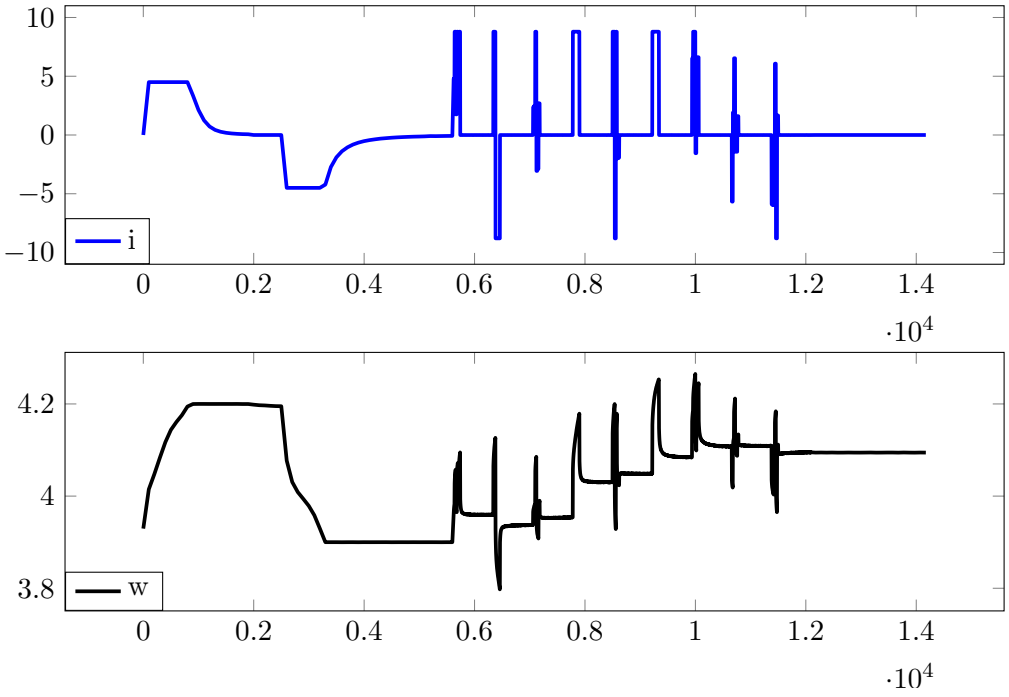


Figure A.4: Input i (in Ampere) and data w (in Volt) for TEST 4.

Bibliography

[1] A. Alexanderian et al. "A fast and scalable method for A-optimal design of experiments for infinite-dimensional Bayesian nonlinear inverse problems". In: *SIAM J. Sci. Comput.* 38.1 (2016), A243–A272. DOI: 10.1137/140992564.

[2] E. Arian, M. Fahl, and E.W. Sachs. *Trust-region proper orthogonal decomposition for flow control*. Tech. rep. 2000-25. ICASE, 2000.

[3] A.C. Atkinson, A.N. Donev, and R.D. Tobias. *Optimum experimental designs, with SAS*. Vol. 34. Oxford Statistical Science Series. Oxford University Press, Oxford, 2007, pp. xvi+511. ISBN: 978-0-19-929660-6.

[4] B. Azmi, A. Petrocchi, and S. Volkwein. *Adaptive Parameter Optimization For An Elliptic-Parabolic System Using The Reduced-Basis Method With Hierarchical A-Posteriori Error Analysis*. 2023. arXiv: 2307.12723 [math.NA].

[5] S. Banholzer, L. Mechelli, and S. Volkwein. "A trust region reduced basis Pascoletti-Serafini algorithm for multi-objective PDE-constrained parameter optimization". In: *Mathematical and Computational Applications* 27 (2022), p. 39.

[6] S. Banholzer et al. "An adaptive projected Newton non-conforming dual approach for trust-region reduced basis approximation of PDE-constrained parameter optimization". In: *Pure and Applied Functional Analysis* 7 (2022), pp. 1561–1596.

[7] R.E. Bank and R.K. Smith. "A posteriori error estimates based on hierarchical bases". In: *SIAM Journal on Numerical Analysis* 30 (1993), pp. 921–935. DOI: 10.1137/0730048.

[8] M. Barrault et al. "An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations". In: *Comptes Rendus Mathematique* 339.9 (2004), pp. 667–672.

[9] M.A. Branch, T.F. Coleman, and Y. Li. "A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems". In: *SIAM Journal on Scientific Computing* 21.1 (1999), pp. 1–23. DOI: 10.1137/S1064827595289108.

[10] R.H. Byrd et al. "A Limited Memory Algorithm for Bound Constrained Optimization". In: *SIAM Journal on Scientific Computing* 16.5 (1995), pp. 1190–1208. DOI: 10.1137/0916069.

[11] S. Chaturantabut and D.C. Sorensen. "A state space estimate for POD-DEIM nonlinear model reduction". In: *SIAM Journal on Numerical Analysis* 50.1 (2012), pp. 46–63.

[12] S. Chaturantabut and D.C. Sorensen. "Nonlinear model reduction via discrete empirical interpolation". In: *SIAM Journal on Scientific Computing* 32 (2010), pp. 2737–2764. DOI: 10.1137/090766498.

[13] J.R. Cho and J.T. Oden. "A priori modeling error estimates of hierarchical models for elasticity problems for plate- and shell-like structures". In: *Mathematical and Computer Modelling* 23 (1996), pp. 117–133. DOI: 10.1016/0895-7177(96)00058-1.

[14] R. Dautray and J.L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology. Volume 5 Evolution Problems I.* Berlin, Heidelberg: Springer, 2000. DOI: 10.1090/gsm/019.

[15] P. Deuflhard. "Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms". In: 2011. URL: https://api.semanticscholar.org/CorpusID:116962126.

[16] C. Domínguez, E.P. Stephan, and M. Maischak. "AFE-BE coupling for a fluid-structure interaction problem: hierarchical a posteriori error estimates". In: *Numerical Methods for Partial Differential Equations* 28 (2012), pp. 1417–1439. DOI: 10.1002/num.20686.

[17] M. Doyle, T.F. Fuller, and J. Newman. "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell". In: *Journal of The Electrochemical Society* 140 (1993), pp. 1256–1533. DOI: 10.1149/1.2221597.

[18] S.L. Eskew and J.R. Singler. "A new approach to proper orthogonal decomposition with difference quotients". In: *Advances in Computational Mathematics* 49 (2021), pp. 1–33. URL: https://api.semanticscholar.org/CorpusID:235485221.

[19] L.C. Evans. *Partial Differential Equations.* Graduate Studies in Mathematics. Providence, Rhode Island: American Mathematical Society, 2010. DOI: 10.1090/gsm/019.

[20] J. Feinauer et al. "MULTIBAT: Unified workflow for fast electrochemical 3D simulations of lithium-ion cells combining virtual stochastic microstructures, electrochemical degradation models and model order reduction". In: *Journal of Computational Science* 31 (2019), pp. 172–184. ISSN: 1877-7503. DOI: https://doi.org/10.1016/j.jocs.2018.03.006.

[21] J.C. Forman et al. "Genetic identification and fisher identifiability analysis of the Doyle–Fuller–Newman model from experimental cycling of a LiFePO4 cell". In: *Journal of Power Sources* 210 (2012), pp. 263–275.

[22] A. Friedman. *Partial Differential Equations of Parabolic Type.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1964.

[23] T.F. Fuller, M. Doyle, and J. Newman. "Simulation and optimization of the dual lithium ion insertion cell". In: *Journal of The Electrochemical Society* 141 (1994), pp. 1–10. DOI: 10.1149/1.2054684.

[24] P.M. Gomadam et al. "Mathematical modeling of lithium-ion and nickel battery systems". In: *Journal of Power Sources* 110 (2002), pp. 267–284. DOI: 10.1016/S0378-7753(02)00190-8.

[25] G.C. Goodwin and R.L. Payne. *Dynamic system identification.* Vol. 136. Mathematics in Science and Engineering. Experiment design and data analysis. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York-London, 1977, pp. x+291. ISBN: 0-12-289750-1.

[26] M.A. Grepl and A.T. Patera. "A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations". In: *M2AN Math. Model. Numer. Anal.* 39.1 (2005), pp. 157–181.

[27] M. Gubisch and S. Volkwein. "Proper Orthogonal Decomposition for Linear-Quadratic Optimal Control". In: 2013. URL: https://api.semanticscholar.org/CorpusID:1742092.

[28] B. Haasdonk. "Convergence rates of the POD-greedy method". In: *ESAIM Math. Model. Numer. Anal.* 47.3 (2013), pp. 859–873.

[29] S. Hain et al. "A hierarchical a posteriori error estimator for the reduced basis method". In: *Advances in Computational Mathematics* 45 (2019), pp. 2191–2214. DOI: 10.1007/s10444-019-09675-z.

[30] J.S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. SpringerBriefs in Mathematics. Cham: Springer, 2016. DOI: 10.1007/978-3-319-22470-1.

[31] M. Hinze et al. *Optimization with PDE constraints*. Vol. 23. Mathematical Modelling: Theory and Applications. Springer, New York, 2009.

[32] T. Keil and M. Ohlberger. *A Relaxed Localized Trust-Region Reduced Basis Approach for Optimization of Multiscale Problems*. 2022. arXiv: 2203.09964 [math.NA].

[33] T. Keil et al. "A non-conforming dual approach for adaptive trust-region reduced basis approximation of PDE-constrained parameter optimization". In: *ESAIM Math. Model. Numer. Anal.* 55.3 (2021), pp. 1239–1269.

[34] C.T. Kelley. *Iterative methods for optimization*. Vol. 18. Frontiers in Applied Mathematics. SIAM, Philadelphia, PA, 1999, pp. xvi+180.

[35] K. Kunisch and S. Volkwein. "Galerkin proper orthogonal decomposition methods for parabolic problems". In: *Numerische Mathematik* 90.1 (2001), pp. 117–148. DOI: 10.1007/s002110100282.

[36] O. Lass. "Reduced order modeling and parameter identification for coupled nonlinear PDE systems". See http://nbn-resolving.de/urn:nbn:de:bsz:352-272816. PhD thesis. University of Konstanz, 2014.

[37] O. Lass and S. Volkwein. "POD Galerkin schemes for nonlinear elliptic-parabolic systems". In: *SIAM Journal on Scientific Computing* 35.3 (2013), A1271–A1298. DOI: 10.1137/11084841.

[38] A. Latz and J. Zausch. "Thermodynamic consistent transport theory of Li-ion batteries". In: *Journal of Power Sources* 196.1 (2011), pp. 3296–3302. DOI: 10.1016/j.jpowsour.2010.11.088.

[39] A. Latz, J. Zausch, and O. Iliev. "Modeling of species and ccharge Transport in li-ion batteries based on non-equilibrium thermodynamics". In: *Numerical Methods and Applications*. Ed. by I. Dimov, S. Dimova, and N. Kolkovska. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 329–337. DOI: 10.1007/978-3-642-18466-6_39.

[40] X. Li, M. Xiao, and S.Y. Choe. "Reduced order model (ROM) of a pouch type lithium polymer battery based on electrochemical thermal principles for real time applications". In: *Numerische Mathematik* 90.1 (2001), pp. 117–148. DOI: 10.1007/s002110100282.

[41] C. López et al. "A computational framework for identifiability and ill-conditioning analysis of lithium-ion battery models". In: *Industrial & Engineering Chemistry Research* 55.11 (2016), pp. 3026–3042.

[42] J.S. Newman. *Electrochemical systems*. Englewood Cliffs: Prentice Hall, 1973.

[43] M. Ohlberger, S. Rave, and F. Schindler. "Model reduction for multiscale lithium-ion battery simulation". In: vol. 112. Springer, Cham, 2016, pp. 317–331. DOI: 10.1007/978-3-319-39929-4_31.

[44] A. Pázman. *Foundations of optimum experimental design*. Mathematics and its applications. D. Reidel, 1986. URL: https://cir.nii.ac.jp/crid/1130282269915354752.

[45] A. Petrocchi, M.K. Scharrer, and S. Volkwein. "Trust-Region RB Methods for PDE-Constrained Optimization and Optimal Input Design". In: *IFAC-PapersOnLine* 55.26 (2022). 4th IFAC Workshop on Control of Systems Governed by Partial Differential Equations CPDE 2022, pp. 149–154. DOI: https://doi.org/10.1016/j.ifacol.2022.10.392.

[46] A. Petrocchi et al. *Optimal Experimental Design for Large-Scale Inverse Problems via Multi-PDE-constrained Optimization*. 2024. arXiv: 2404.15797 [math.ST].

[47] P. Popov et al. "Finite volume discretization of equations describing nonlinear diffusion in li-ion batteries". In: *Numerical Methods and Applications*. Ed. by I. Dimov, S. Dimova, and N. Kolkovska. Springer Berlin Heidelberg, 2011, pp. 338–346. DOI: 10.1007/978-3-642-18466-6_40.

[48] A. Pozzi et al. "Optimal design of experiments for a lithium-ion cell: parameters identification of an isothermal single particle model with electrolyte dynamics". In: *Industrial & Engineering Chemistry Research* 58.3 (2018), pp. 1286–1299.

[49] L. Pronzato and A. Pázman. "Design of experiments in nonlinear models". In: *Lecture notes in statistics* 212.1 (2013).

[50] F. Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

[51] E. Qian et al. "A certified trust region reduced basis approach to PDE-constrained optimization". In: *SIAM J. Sci. Comput.* 39.5 (2017), S434–S460.

[52] A. Quarteroni. *Numerical models for differential problems*. third. Vol. 16. MS&A. Modeling, Simulation and Applications. Springer, Cham, 2017. DOI: 10.1007/978-88-470-5522-3.

[53] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*. UNITEXT – La Matematica per il 3+2. Cham: Springer, 2016. DOI: 10.1007/978-3-319-15431-2.

[54] E. Rafajłowicz. "Optimum choice of moving sensor trajectories for distributed-parameter system identification". In: *International Journal of Control* 43.5 (1986), pp. 1441–1451. DOI: 10.1080/00207178608933550.

[55] L.B. Rall. *Automatic Differentiation: Techniques and Applications*. Vol. 120. Lecture Notes in Computer Science. Berlin: Springer, 1981. DOI: 10.1007/3-540-10861-0.

[56] A.M. Ramos. "On the well-posedness of a mathematical model for lithium-ion batteries". In: *Applied Mathematical Modelling* 40 (2016), pp. 115–125. DOI: 10.1016/j.apm.2015.05.006.

[57] S. Reddy et al. "Accelerating parameter estimation in Doyle–Fuller–Newman model for lithium-ion batteries". In: *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering* 38.5 (2019), pp. 1533–1544.

[58] S. Rogg, S. Trenz, and S. Volkwein. *Trust-region POD using a-posteriori error estimation for semilinear parabolic optimal control problems*. Tech. rep. Konstanzer Schriften in Mathematik No. 359. University of Konstanz, 2017.

[59] E.W. Sachs, M. Schneider, and M. Schu. "Adaptive trust-region POD methods in PIDE-constrained optimization". In: *Numerische Mathematik* 165 (2014), pp. 327–342.

[60] S. Santhanagopalan, Q. Guo, and R.E. White. "Parameter Estimation and Model Discrimination for a Lithium-Ion Cell". In: *Journal of The Electrochemical Society* 154.3 (2007), A198. DOI: 10.1149/1.2422896.

[61] T. Seger. "Elliptic-Parabolic Systems with Applications to Lithium-Ion Battery Models". See http://nbn-resolving.de/urn:nbn:de:bsz:352-241413. PhD thesis. University of Konstanz, 2013.

[62] K. Smith and C.Y. Wang. "Power and thermal characterization of a lithium- ion battery pack for hybrid-electric vehicles". In: *Journal of Power Sources* 160 (2006), pp. 662–673. DOI: 10.1016/j.jpowsour.2006.01.038.

[63] M. Torchio et al. "LIONSIMBA: A Matlab Framework Based on a Finite Volume Model Suitable for Li-Ion Battery Design, Simulation, and Control". In: *Journal of The Electrochemical Society* 163.7 (2016), A1192. DOI: 10.1149/2.0291607jes.

[64] F. Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*. Vol. 112. Graduate Studies in Mathematics. Translated by J. Sprekels. Springer Cham, 2010. DOI: 10.1090/gsm/112.

[65] D. Uciński. *Optimal measurement methods for distributed parameter system identification*. Systems and Control Series. CRC Press, Boca Raton, FL, 2005, pp. xviii+371. ISBN: 0-8493-2313-4.

[66] P. Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[67] S. Volkwein and A. Wesche. "The reduced basis method applied to transport equations of a lithium-ion battery". In: *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* 32 (2013), pp. 1760–1772.

[68] J. Wu, J. Xou, and H. Zou. "On the well-posedness of a mathematical model for lithium-ion battery systems". In: *Methods and Applications of Analysis* 13 (2006), pp. 275–298. DOI: 10.4310/MAA.2006.v13.n3.a4.

[69] Y. Yue and K. Meerbergen. "Accelerating Optimization of Parametric Linear Systems by Model Order Reduction". In: *SIAM Journal on Optimization* 23.2 (2013), pp. 1344–1370. DOI: 10.1137/120869171.